Politecnico di Torino

ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Computer and Control Engineering (34^{th}cycle)

Artificial Intelligence methodologies to early predict student outcome and enrich learning material

By

Lorenzo Canale

******

Supervisor(s):
Prof. Laura Farinetti

Doctoral Examination Committee:
Prof. Marina Marchisio, Referee, Università di Torino
Prof. Tiziana Margaria, Referee, University of Limerick
Prof. Pietro Michiardi, EURECOM
Prof. Wolfgang Müller, University of Education of Weingarten

Politecnico di Torino
2022

I hereby declare that the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

. . . . . .

Lorenzo Canale

Turin, June 9, 2022

Acknowledgements

I would like to acknowledge Professor Laura Farinetti for offering me support, mentorship and opportunity to teach her courses. In addition, I thank her, along with Professor Luca Cagliero for the research articles we carried out together. Their guidance helped me to become a better researcher.

Special thanks to all the current and past members of DAUIN Lab 6 and to all the researchers who have provided me with research insights.

Finally, thank you to my parents, Anna and Ubaldo, for their support.

Abstract

From the emergence of the Web onward new learning modes such as distance learning and blended learning have developed. At the same time, *Artificial Intelligence* (AI) has taken hold in most areas including education.

This dissertation focuses on the application of AI (i) to predict the student's exam outcome and (ii) to enrich learning resources.

Student outcome prediction is a topic of great interest in the learning analytics literature. Two methodologies were proposed to address (a) *early* prediction, i.e. make predictions at different time instants from the beginning of the course, and derive (b) *explainable* models, i.e. its output can be explained in a way that "makes sense" to a human being at an acceptable level. The combination of these two characteristics provides insight into why a student is at risk of failure and enables to intervene as soon as possible. The first methodology is based on *Lazy Associative Classifier* L^3; associative algorithms enable the derivation of human-readable rules to explain the prediction reasons and extract student profiles. The second methodology named *VESPE* integrates different machine learning models and allows determining the impact of each variable through *SHapley Additive exPlanations (SHAP)*, a state-of-the-art explainable method based on game theory. Both techniques were validated in two case studies involving university courses. In addition *UNIFORM*, a method for integrating different educational datasets, used by other studies for student outcome prediction, has been proposed.

To *enrich learning resources material*, two AI techniques based on named entity linking were discussed; the first one, called *VISA*, performs *video-lecture indexing* with semantic annotations enabling students to search more easily for specific content, a practice especially beneficial for reviewing before the exams. *VISA* outperformed competing algorithms on a dataset inherent to an undergraduate Database course. The second technique, called *TVREM*, address *video to text and text to video retrieval*

of educational resources: it allows searching for a video from a text and vice-versa to align different resources related to the same topic. *TVREM* was validated with two datasets containing educational vid +s and textual content achieving significantly higher results than baselines and competitors. In addition, an application based on *TVREM* has been proposed to automatically derive educational Youtube videos from textbook paragraphs. The introduction of *VISA* and *TVREM* is beneficial to learning since literature revealed that both video-lecture indexing and cross-media retrieval of educational resources increase student engagement and lead to higher achievement on exams.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The pre-digital era is mainly characterized by the traditional educational model where instructors share new knowledge with students who possess basic competencies; learning was mainly verified through exams and homework. The only modes of distance education involved TV programs and recorded audio/videos. Professors provide assistance to students via phone and email.

One of the earliest use of computers for teaching dates back to 1954 for an educational program implemented at Harvard university by F. Skinner and J. G. Holland, where students experienced self-directed and self-administered instruction.

A gradual emphasis on the investigation of learning patterns emerged shortly after. In 1963, the term "criterion-referenced measures" was introduced by Robert Glaser to evaluate learners' behavior according to the identified learning objectives.

Yet the emergence of the Internet leads to enrichment and diversification in learning modes since the 1990s. Different modalities, such as e-learning and blended learning, have been adopted for university courses; for example, the first open-source learning management system, i.e. Moodle, was introduced in 2002. Shortly thereafter large-scale online education arose in 2008 with Massive Open Online Courses (MOOCs), e.g. Udacity, EdX, and Coursera; they reinforce traditional learning practices such as mastery learning with interactive exercises. Learners started to dispose of a wide variety of channels and workspaces to share information, encouraging peer-to-peer collaboration. Professors in turn enjoyed new ways to engage students, e.g., discussion forums, video pills, and social networks.

The rise of online learning and digital support facilitated the collection of data about student learning and contributed to the emergence of two research communities: Educational Data Mining (EDM) and Learning Analytics and Knowledge (LAK). The former defined *Educational Data Mining* on its website [1] as "a discipline that aims at developing methods for better understanding learners and the learning context by exploring unique and large scale educational data." The first international conference on educational data mining was held in 2008.

A few years later, in 2011, the first International Conference on Learning Analytics & Knowledge took place in Canada and *Learning Analytics* (LA) was defined as "the measurement, collection, analysis and reporting of data about learners and their contexts, for the purposes of understanding and optimizing learning and the environments in which it occurs."

EDM and LA have the same goal: "improving education quality by analyzing huge amounts of data to extract useful information for stakeholders" [152]. The overlap between these two issues is significant, and the communities were encouraged to converge since there is little support for a clear demarcation between the two disciplines [122].

However, some distinctions can be outlined(see Table 1.1) [127, 192]. EDM aims more at automatic discovery, while LA aims more at exploiting human judgment using automated discovery in the service of informing instructors/learners who make final decisions. In addition, the origin of learning analytics is more related to the semantic web.

The presented methodologies make extensive use of *Artificial Intelligence* (AI) which has been gradually applied in more and more areas over the past century. In education, it has been adopted mainly to support (i) administration tasks, (ii) instruction, and (ii) learning [48]. Table 1.2 shows some usage scenarios for each of the three categories.

A large proportion of LA topics rely on AI, including the automatic prediction of student outcomes which is the first main topic of this dissertation; Chapter 2 presents (i) a detailed literature review and (ii) the proposal of *artificial intelligence explainable methodologies to early predict student outcome* validated in two case studies.

[1] www.educationaldatamining.org

Table 1.1 Main distinctions between Learning Analytics (LA) and Educational Data Mining (EDM)

The content of this table is derived from [127, 192]

	LA	**EDM**
Discovery	Leveraging human judgement is key; automated discovery is a tool to accomplish this goal	Automated discovery is key; leveraging human judgment is a tool to accomplish this goal
Reduction & Holism	Stronger emphasis on understanding systems as wholes, in their full complexity	Stronger emphasis on reducing to components and analyzing individual components and relationships between them
Origins	LAK has stronger origins in semantic web, "intelligent curriculum", outcome prediction, and systemic interventions	EDM has strong origins in educational software and student modeling, with a significant community in predicting course outcomes
Adapation & Personalization	Greater focus on informing and empowering instructors and learners	Greater focus on automated adaption (e.g. by the computer with no human in the loop)
Techniques & Methods	Social network analysis, sentiment analysis, influence analytics, discourse analysis, learner success prediction, concept analysis, sensemaking models	classification, clustering, Bayesian modeling, relationship mining, discovery with models, visualization

The second contribution of this dissertation (Chapters 3 and 4) concerns the adoption of *artificial intelligence to enrich learning resources*. In particular, Chapter 3 presents a method for *automatic indexing of video lectures* that can help students safeguard time in searching for the desired content, an action especially helpful in a pre-exam review. Chapter 4, on the other hand, focuses on the *video-to-text and text-to-video retrieval* for creating educational resources that link educational media of different types (e.g., e-book chapters with videos); numerous studies have investigated cross-media retrieval, however only a couple have focused on its application in the educational domain.

The literature states that learners can benefit from both the indexing of video lectures and the use of multimodal educational resources to improve their engagement and their performance. Consequently, the combination of the two previously mentioned contributions converges in the title of this dissertation: *Artificial Intelligence methodologies to early predict student outcomes and improve learning resources*.

Finally, the conclusions (Chapter 5) detail a summary of the main findings of the thesis and present a methodology to be tested in future work to combine video indexing, cross-media retrieval for educational resources and a student alert system.

Table 1.2 Classification of Artificial Intelligence uses for education

The content of this table is derived from [48]

Administration	• Perform the administrative tasks faster that consume much of instructors' time, such as grading exams and providing feedback. • Identify the learning styles and preferences of each of their students, helping them build personalized learning plans. • Assist instructors in decision support and data-driven work. • Give feedback and work with students timely and directly.
Instruction	• Anticipate how well a student exceeds expectations in projects and exercises and the odds of dropping out of school. • Analyze the syllabus and course material to propose customized content. • Allow the instruction beyond the classroom and into the higher-level education, supporting collaboration. • Tailor teaching method for each student based on their personal data. • Help instructors create personalized learning plans for each student.
Learning	• Uncover learning shortcomings of students and address them early in education. • Customize the university course selection for students. • Predict the career path for each student by gathering studying data. • Detect learning state and apply the intelligent adaptive intervention to students.

To facilitate readers' understanding, Appendix A includes background knowledge related to the semantic web, evaluation metrics and explainable AI.

Chapter 2

Student Outcome Prediction in higher education

In higher education, a growing number of studies in the area of learning analytics are focusing on student outcome prediction using machine learning. This allows to derive models able to forecast students' performance with varying accuracy according to the point of time at which the prediction occurs; indeed, early prediction allows for intervention in advance, e.g. alerting the students at risk.

Caring model explainability, i.e. examining the features that contributed the most to the forecast, is also a major advantage in understanding students' most influential behaviors and providing accurate interventions.

This chapter focuses on these issues; section 2.1 reviews various studies in the literature by answering some key questions to understand the developments in the research topic. Section 2.1 and 2.3 present two case studies in which explainable ML models have been applied to predict student outcomes; the first focuses on the first year of university to identify the key factors for success/failure, the second examines a programming course to analyze how tracking student activity through Version Control Systems is beneficial to early predict dropout, success/failure and grade. Section 2.4 outlines a method for automatically aligning different open learning datasets that may be used to assess the transferability of prediction models and to help distinguish context-specific from general findings. Finally, Section 2.5 draws conclusions and specifies future directions.

2.1 Literature review

This section considers several studies from the literature. Most of them have been extracted from the following surveys: [56, 166] reviews the most important research about dropout prediction, while [11, 121] focus on student exam grades.

Further literature was extracted by looking at cross citations and querying Google Scholar [1] with the following keywords : "student outcome prediction", "student dropout prediction", "student performance prediction" and "student grade prediction". The results were limited to the first two pages for each query.

Table 2.1 Student Outcome Prediction literature by year

Year	Papers
2003	[113]
2005	[116]
2006	[7, 97]
2009	[59, 131, 132]
2010	[112, 114, 117, 209]
2012	[1, 74, 77, 155, 200]
2013	[15, 21, 115, 187, 235]
2014	[13, 17, 19, 90, 96, 107, 137, 175, 189, 193, 194, 208]
2015	[3, 34, 37, 45, 60, 69, 91, 106, 108, 204, 218, 225, 239]
2016	[5, 12, 14, 57, 124, 125, 126, 169, 179, 202, 221, 231, 232, 240]
2017	[8, 9, 53, 121, 147, 163, 181, 223, 226, 227, 237]
2018	[6, 56, 63, 80, 84, 129, 140, 148, 162, 174, 241]
2019	[28, 38, 44, 52, 71, 83, 151, 161, 170, 172]
2020	[51, 166, 210, 217, 222]
2021	[11, 128, 234]
2022	[149, 150, 168]

The complete list of papers (see Table 2.1) was examined from different perspectives by answering the following questions:

Q1) Which targets (outcomes) are predicted?

[1] https://scholar.google.com/

Q2) Which learning modes were considered in the analyzed studies?

Q3) Which is the deeper sense of "prediction"? Which data are used to train models, and which for testing?

Q4) Which student features are used to predict their achievement?

Q5) Which are the most widely adopted AI algorithms?

Q6) In which ways could the contribution of features be established?

Q7) Which studies address early prediction?

Q8) Which are the major challenges of student outcome prediction?

Each of the following subsections focuses on one of these issues.

2.1.1 Predicted outcomes

Two main targets related to exam outcome can be identified: *success/failure* target is a binary classification problem that distinguishes students who *pass* the exam from those who *fail* it, while the *grade* target determines the student's grade; in most cases, grades are discretized to form grade bands turning the task from regression into a multiclass classification. Consider a grade scale from 1 to 21 where 1 to 9 denote a failure (class *fail*) and grades 10 and up a success (class *pass*); it could be uniformly split into the following ranges: *poor* (10 to 13), *average* (14 to 17), *good* (18 to 21). Uniform discretization is only one of the possible strategies used to define the bins' widths.

A further target is *dropout* that identifies who abandons an attempt, activity, or chosen path; more specifically in higher education it refers to either leaving the entire learning pathway [4, 28, 59, 196], such as leaving study, or quitting a single class, i.e. *course dropout*; only the latter is considered in this dissertation.

Since a minority of studies have directly gathered students' responses that stated whether or not they had left a class [106], *dropout* lacks of a strict and unique definition ([230]). Most research [21, 21, 34, 107, 175, 189, 194, 208, 231, 235, 235] defined it as *stout*, i.e. lack of interaction from a certain point (e.g. week) onwards in the course; other studies [13, 69], on the other hand, denoted it as a momentary

departure, i.e. students don't do activities in the next phase of the course regardless of whether they take it back in the future. The measure of the activity varies according to the context; [194] takes into account the viewing of video lectures, [235] the participation in discussion forums and [12, 208] the submission of assignments and quizzes.

Other less common dropout definitions emerged in other research. In [90] students marked as dropout have been absent from the course for more than 1 month and/or have fewer than 50% of the videos in the course. In [108] stop submitting quizzes or non-participation in the final exam has been accounted for by the authors as a dropout indicator. Other studies have looked at non-completion of the certification module [179, 223], sometimes in conjunction with a lack of interaction in the final phase of the course [225]. Finally [8] held future educational advances, labeling as dropouts students who did not take any more MOOCs courses in the future after the one considered.

It is also common in the literature to find the label *at-risk*, which in most cases indicates students predicted to fail the exam, but it sometimes merges dropout students as well.

2.1.2 Learning modes

The considered studies cover three learning modes:

- *In-class learning*: it refers to courses in which the lesson is typically delivered via a speech or presentation by the instructor.
- *Blended learning*: it combines the traditional frontal classroom method with computer-mediated activity. However, just using the Internet or technology in some way does not mean the learning is blended [82]: assessments and modality of learning has to align with the course's learning objectives [219].
- *Massive Open Online Courses (MOOCs)*: free online courses available for anyone to enroll. MOOCs provide an affordable and flexible way to learn new skills, advance your career and deliver quality educational experiences at scale.

2.1.3 Prediction meaning

[34] presents two main formulations to generate models for dropout prediction that are here generalized to student outcome prediction, that is, targets success/failure and grade are also included:

- *Entire history*: all the information available regarding the student up until time instant t_i for making predictions beyond t_i is used.
- *Moving window*: a fixed amount of historical information of the learner (parameterized by the window size) is used to make predictions. That is, if window size is set to 2, only the information from time instants t_{i-1} and t_{i-2} will be used for the prediction at time instant t_i.

Since most studies use supervised algorithms, a further distinction is related to the data used for training [226]:

- Train on *same course offering*: the model is trained and tested using features and target labels from the same course instance; hence data are split into training and test sets not taking into account time constraints since training target labels become available only after the course ends. This approach implies that the future course offering has the same or similar distribution of data. This is a strong assumption that is not always true or verifiable. Nevertheless, the vast majority of considered approaches adopt this technique as shown in Figure 2.2.
- Train on a *different offering of the same course*: the model is trained on a past course offering and tested on the current one. In this case, the time constraints of prediction are met since the training labels are available before the course starts. Consequently, this training mode is a more reliable simulation of a real condition than the previous one.
- Train on a *different course*: the model is trained on another course (usually from the same field) and tested on the one considered.
- Train on *multiple different courses*: multiple different courses from the same field are considered and one model is trained for each of them finally averaging the classifiers' hyperplanes together.

- Train using proxy labels (*in situ*): this is valid only for the dropout target as it means reaching the end of the course and obtaining a certification. When predicting for a given week w_i which students from the course will drop out, train using proxy labels corresponding to whether each student persisted within the previous week w_{i-1}. The test and train labels have different semantic meanings and this may impact the prediction performance on the test set.

Table 2.2 Student Outcome Prediction literature categorized by data used for training

Target / **Training mode**	*Dropout*	*Success/Failure*	*Grade*
Same course offering	[8, 9, 12, 13, 21, 28, 44, 45, 52, 58, 59, 69, 106, 107, 140, 147, 148, 150, 161, 162, 169, 189, 194, 208, 223, 225, 226, 227, 231, 235]	[38, 51, 96, 112, 117, 129, 155, 172, 202, 239, 240, 241]	[1, 3, 5, 6, 7, 14, 15, 17, 19, 37, 53, 63, 74, 83, 106, 115, 125, 128, 151, 163, 168, 174, 187, 209, 218, 221, 222, 232, 234, 237]
Different offering of the same course	[34, 132, 226]	[91]	
Different course	[90, 225, 226, 227]		[7, 106]
Multiple different courses	[226]		
In situ	[226, 227]		

2.1.4 Feature families

Because the studies analyzed consider various educational settings, many different features are adopted for student outcome prediction (see Table 2.3). Indeed, it is very unlikely to find a pair of studies that use exactly the same features, although some predominant practices can be identified regarding MOOCs, such as the adoption of clickstream data related mostly to videos, quizzes, and discussion forums as well as other navigation indicators [9, 166, 194, 225, 227]. In many research, regardless of the delivery mode, demographic data are used [3, 7, 69, 155, 169, 187, 208, 231], sometimes coupled with social characteristics [13, 162, 174].

The other feature families have been used in fewer studies, although they contributed the most to prediction since studies address specific behaviors and, given

the variety of learning contexts, it is critical to focus on the course domain to exploit all available information.

Table 2.3 Features employed for Student Outcome Prediction in the literature

Feature family	**Feature**
Demographics	Gender, Age, Sex, Race, First-generation immigrant, Second-generation immigrant, Occupation, Residence, Accommodation type, Working experience, Health insurance, Taken care by, Cohabitation status, Family size, Family expenditure, Family income, Family assets, Father' higher education qualification, Mother' higher education qualification, Father's occupation, Mother's occupation, Parents' annual income, Parental status
Social	Have mobile, Computer/laptop at home, Net access, Social network id, Volunteer work, Travelling way, Travelling time, Number of children, Planned and unplanned pregnancies, Maleness / Feminism, Bulling, Vices of the student, Commitment for being a firstborn child
Prior education	Entrance qualification, Grade of entrance qualification, City of entrance qualification, English language level, High school grade, High school guidance/type, Prior experience with the topic, Prior experience with learning modality, Grades in other courses
Current career	Grade point average (GPA), Number of exams taken, Number of exams not participated, Number of exams succeeded, Preliminary test grade, Reason to choose this college, Enrollment year, Study Interest, Field of study, Department, Career development, Fund Funding, Study hours, Perception of the student about the insertion into the labor field
In-class	Class absence, Class early leave, Lateness
Connection	Web connection country, Browser, Number of devices, Device, OS
Navigation	Number of requests, Browser opening count, Number of clicks, Sessions count, Number of module/chapters views, Number of page views, Number of times the course progress page was checked, Number of forwards, Number of backwards, Number of touches, Number of active days, Total time spent, Last interaction
Video-lectures	Number of streaming plays, Percentage of watched video, Number of rewatched, Number played videos without any jump, Number of starts/stops during video playing, Number of pauses, Number of skip ahead, Number of skips, Number of relisten/check back, Number of seek forward/ seek backward, Seeks time, Number of show/hide subtitle actions, Speed changes, Number of slow/high play rate use, Total watch time, Number of downloads
Intermediate quizzes	Number of views, Number of submissions, Number of correct submissions, Resolution time, Resolution time per correct submissions
Assignments	Number of submissions, Number of correct submissions, Predeadline submission time, Number of homework views, Assignment grade, Resolution time per correct submissions, Project grade, Project submission date, Number of logical lines for each submitted code, Laboratory grade
Forum activity	Number of views, Number of threads, Number of posts, Number of replies received from well-performed students, Average post length, Number of comments
Books	Total time spent
External resources	Wiki view count, Wiki edits, Time spent on Wiki resources, Extra college support, Extracurricular activities
Peer collaboration	Total count of collaborations, Club activity, Number of edges on social network the students have each week
Lecturer	Lecturer department, Commitment of the teacher to the student, Instructors ability to awaken your interest, Instructor enthusiasm, Instructor facilitation, Evaluation fairness, Feedback promptness, Feedback usefulness in clarifying debts
Impressions about the course	Course intellectually stimulating, Clarification of evaluation criteria, Self-confidence improvement, Communication skills improvements, Course satisfaction, Degree aspiration improvement, Sentiment

Table 2.4 Artificial Intelligence algorithms employed for Student Outcome Prediction in the literature

Target / **Algorithm**	*Dropout*			*Success/Failure*			*Grade*		
	In-class	Blended	MOOCs	In-class	Blended	MOOCs	In-class	Blended	MOOCs
Linear Regression								[53, 234]	
Logistic Regression	[140, 148, 162]	[210]	[34, 80, 126, 179, 208, 225]	[241]	[217, 239]	[80, 91]	[128]	[53, 151]	
K-Nearest Neighbors					[38]			[151, 172]	[37]
Bayesian Network	[58, 150]			[241]			[74]		
Naive Bayes	[58, 148, 162]		[113, 124, 137]	[241]	[38]	[77, 116, 155]	[1, 3, 17, 19, 202]		
Support Vector Machines	[58, 140]		[13, 51, 107]	[241]		[51]	[128, 221]	[172]	[232]
Decision Tree	[59, 140, 148, 162]	[44, 161]	[147, 241]	[112, 241]	[44]	[117, 155]	[3, 5, 6, 7, 14, 17, 19, 128, 187, 202]	[115, 234]	
Random Forests	[52, 148, 150, 162]			[38, 83, 112]	[217]		[83, 128]	[151]	[163]
Associative Classifiers		[44]			[44]		[1, 3, 15, 17, 19]		
Probabilistic Soft Logic			[175]						
Survival Analysis			[235]						
Neural Networks	[148, 150]	[210]	[69, 71, 170]	[241]	[172, 240]	[155]	[17, 74, 128, 202]	[151, 172, 218, 234]	[63, 125, 163, 237]
AdaBoost	[148, 150]		[28, 241]		[96]		[128]		
Other ensemble			[131]	[112]			[128, 174]		
Other					[129]			[222]	

2.1.5 AI algorithms

Table 2.4 shows the ML algorithms used in the reviewed literature categorized by target and learning mode. The most common techniques are *Logistic Regression*, *Naive Bayes*, *Decision Tree*, *Random Forests*, *Support Vector Machines* and *Neural Networks*. In particular for dropout prediction *Logistic Regression* is the most widely adopted approach, especially in MOOCs, for success and failure detection there are no approaches that clearly outperform the others, and *Neural Networks* and *Decision Trees* are the most widely used algorithms for grade forecasting, although the latter is especially employed for the classroom mode and not for MOOCs.

2.1.6 Model explainability

Many approaches have tried to analyze the contribution of variables on the target at different levels:

- *Correlation of features with output*: several studies have determined correlation scores (e.g. Pearson P-value and Chi-square) [117, 155, 162, 232, 234].
- *Feature salience by feature selection*: other research has adopted feature selection methods before training models [83, 225, 241].
- *Train algorithms with different feature sets*: some approaches train and test models with varying numbers of features, often breaking them down into families (see Table 2.3) to assess which types of data are most useful in predicting the target [9, 12, 13, 21, 52, 58, 169, 221].
- *Feature importance*: the feature contributions were determined based on their impact on predictions [28, 91, 106, 147, 148, 151, 161, 189, 217, 218, 222, 237, 239, 240]; e.g. decision trees based methods allow to determine Information Gain, Gain Ratio and Gini Index of each feature [28, 96, 147], while logistic and hierarchical regression enables derive coefficients significance [162, 222].
- *Explainable AI*: some algorithms such as decision trees and associative classifiers are inherently explainable because allow rules derivation [1, 3, 5, 6, 7, 14, 15, 17, 19, 96, 115, 117, 140, 155, 162, 174, 187, 202, 209]. The rules are

automatically extracted from a labeled dataset, filtered and sorted by relevance, and then applied to unlabeled data. Since the rule related to a given data sample can be deduced, decision trees and associative classifiers provide local explanations. Due to their readability rules can be manually explored and validated by domain experts, who, looking simultaneously at the predictions' accuracy and explanation, could decide whether or not to trust the data-driven model, to choose whether to collect new data or not, and tailor the subsequent actions to specific student profiles.

2.1.7 Early prediction

Early prediction implies deriving models at different time instants $t_1, t_2 ... t_n$ from the beginning of the course onward and not only at the end, in order to early intervene. For each time instant, only features derived from the data collected up to that point are available for prediction.

Early prediction allows to establish how the prediction accuracy varies over time and, accompanied by model explainability, it gives the educator a comprehensive view of the major determinants of student outcomes.

For example, consider two models m_i and m_j, m_i related to time instant t_i and m_j related to time instant t_j, where $i > j$. Assume that the two models achieve the same prediction performance. However, they employed different features both by values, e.g., counts of a given student action change between t_i and t_j, and by type, e.g., a given course activity is started after instant t_j. The instructor can detect at-risk students as early as possible, that is, at instant t_j, and alert them based on the causes that influenced their classification, estimated by model explainability. Then she/he can examine the predictions at instant t_i to determine whether students are classified again as at-risk or not, revealing whether the alerts were helpful in stimulating them to greater effort. In addition, the m_i explanations can reveal which behaviors the student has adopted to improve her/his outcome.

Table 2.5 distinguishes papers that address student outcome prediction from those that do not. Only 30% of them address early prediction while the other limit prediction to the end of the lectures using all data collected over the entire course duration.

Table 2.5 *Early* prediction of student outcome literature

Papers that address early prediction	[13, 21, 28, 53, 69, 83, 91, 96, 107, 112, 129, 132, 147, 172, 189, 194, 217, 221, 225, 227, 231, 234]
Papers that do not address early prediction	[1, 3, 5, 6, 7, 8, 9, 12, 14, 15, 17, 19, 34, 38, 45, 51, 52, 58, 59, 63, 74, 106, 106, 115, 117, 128, 140, 148, 150, 151, 155, 161, 162, 168, 169, 174, 187, 202, 208, 209, 218, 222, 223, 232, 235, 237, 239, 241]

2.1.8 Major challenges

Based on [56] and [166] that focus only on dropout prediction, the main challenges of students outcome prediction are here listed:

- *Lack of enough data*: the majority of datasets are small and related to a single course offering; as a result, the findings are limited to individual case studies.
- *Data heterogeneity*: generalizing strategies and models turns out to be tricky because different studies consider different learning contexts, hence the available data and features differ; models that fit on one dataset are generally not applicable to the others, revealing limited transfer learning applicability.
- *High classes imbalance*: in most cases, the total number of samples between classes differs greatly; for example, the ratio between students who pass and fail an exam could be unbalanced; some algorithms, e.g. Support Vector Machines and Naive Bayes, suffer more than others unbalanced classes.
- *High feature values variance*: students have the freedom to decide if, what, when, and how to study. This might lead to considerable data variance, which may produce less accurate and reliable ML models.
- *Unstructured data*: detecting and recording student activity (e.g. clickstreams) lack of pre-defined data model. For example, filling missing values is essential since most approaches do not handle it; however, the proper manner of achieving this depends on the given variable, that may be specific to a particular study.
- *Unavailability of publicly accessible dataset*: most datasets are private; in other cases, they omit user-provided data that are, however, indispensable for replicating experiments or studying patterns.

- *Lack of standard benchmark*: there are no standard formats for sharing data and naming features, as a result, it is tricky to automatically align different datasets even when they consider the same features.

2.2 Predicting student academic performance by *Lazy Associative Classifier*

Given the high explainability of associative models demonstrated in [1, 3, 15, 17, 19, 44], this section presents the first attempt to apply the *Lazy Associative Classifier* L^3 [22] to early predict students' performance in the 1st-year bachelor's degree courses in engineering. The primary goal is to identify at-risk students in order to understand the causes of their failure and to early take action.

The following research questions were addressed:

RQ1) Are associative models as accurate as the best performing classifiers in predicting the exam success of university-level students?

RQ2) What are the most discriminating features to forecast exam success at different time points?

RQ3) Which combinations of feature values have frequently been used to assign the exam success?

An extended version of this section's content is published in [41].

2.2.1 Learning context

This study was conducted at the Polytechnic University of Turin, considering students enrolled in the first year of a bachelor's degree in the year 2018-2019.

All mandatory courses in the first year of study were included: Mathematical Analysis (MA), Chemistry (CH), Computer Science (CS), Linear Algebra (LA), and Physics (PH). Besides them, students attend an elective course which is not considered in the present analysis.

The course was held through in-class lectures. Learning Management System (LMS) provides students with the following resources:

- *In-class lectures video recordings* which can be either related to the current instance of the course or to that of previous years; they can be either streamed or downloaded to their personal computers.
- *Educational materials* (e.g., slides, lecture notes, exam simulations) that can be downloaded by students.

MA, CH, and CS courses were offered in the first semester (October 1, 2018 to January 15, 2019), while LA and PH courses were held in the second semester (March 1, 2019 to June 15, 2019).

Three examination sessions were scheduled within the academic year: (i) the winter session, which is held at the end of the first semester (i.e., from 22nd January 2019 to 28th February 2019), (ii) the summer session, which is held at the end of the second semester (i.e., from 16th June 2019 to 22nd July 2019), (iii) the autumn session, which is held after the summer break (i.e., from 1st September 2019 to 30th September 2019).

In the winter session, students can only register for exams related to first semester courses, namely MA, CH and CS, while in the following sessions they can attempt all exams.

Students could choose which exam session to attend, with the option to reject the grade and to re-register in a later exam session as long as the grade has not been already accepted and recorded.

2.2.2 Predicted targets

The target considered in this study is *success/failure*: students are classified as either *pass* or *fail*. Note that students who did not register or did not attend the exam were labeled as *fail*.

The prediction occurred at different time instants, listed in Table 2.6.

More specifically, the success/failure of the winter session exam was predicted from t_0 to t_5, the success/failure of the summer session exam was predicted from t_6 to t_9 and the success/failure of the autumn session was predicted at t_{10} and t_{11}.

The target values per time point are reported in Table 2.6.

Table 2.6 Time points regarded for predicting student academic performance

Id	Time Point	Description
t_0	31 August 2018	Before entry test
t_1	7 September 2018	After entry test
t_2	30 October 2018	Early 1st semester
t_3	31 November 2018	Mid-way 1st semester
t_4	15 January 2019	Close to 1st semester exams
t_5	22 January 2019	Start of 1st semester exam session
t_6	28 February 2019	End of 1st semester exam session
t_7	31 March 2019	Early 2nd semester
t_8	30 April 019	Mid-way 2nd semester
t_9	15 June 2019	Start of 2nd semester exam session
t_{10}	22 July 2019	End of 2nd semester exam session
t_{11}	31 August 2019	After summer break

Table 2.7 Target values per time point regarded for predicting student academic performance

	$t_0 - t_5$		$t_6 - t_9$		$t_{10} - t_{11}$	
	Pass	**Fail**	**Pass**	**Fail**	**Pass**	**Fail**
MA	1515	2577	1183	332	1035	148
CS	1786	2307	1427	359	1127	300
CH	2697	1394	2397	300	2135	262
PH	2823	1270	2823	1270	2431	392
LA	1245	2848	1245	2848	1018	227

2.2.3 Features engineering

Based on the data recorded on the Learning Management System and those released by the institution were considered both features related to the student *demographics* (Gender, Age, BH-loc, HM-loc), *prior education* (HS-loc, HS-gr), *current career* (GRE-gr, BS course) and features dependent on the student course activity as the *educational materials downloads* (C-mat) and the *video-lectures activity* (C-down, C-str). They are respectively listed in Table 2.8 and Table 2.9.

As the course progresses, the number of student interactions with the LMS varies; hence the values of features related to course activity change at each time instant.

To predict summer session outcomes the success/failure outcome of the winter session exams outcomes was added to the features set: MA-gr, CH-gr and CS-gr respectively indicate whether the student passed or failed the Mathematical Analysis (MA), Chemistry (CH) and Computer Science (CS) exams. Students' performance on previous exams may be helpful in predicting future exam outcomes.

Table 2.8 Student *background* features employed for predicting student academic performance

Feature	Description	Data Type	Domain
Gender	gender	categorical	{M = male,F = female}
Age	student's age—average students' age	ordinal	{−1,1,2,3}
BH-loc	country of birth identifier	categorical	{AF,AL,...}
HM-loc	home country identifier	categorical	{AF,AL,...}
HS-loc	high school country identifier	categorical	{AF,AL,....,}
HS-gr	high school grade band	ordinal	{1 = low, 2 = low average, 3 = average, 4 = average high, 5 = high}
GRE-gr	entry test grade	ordinal	{1 = low, 2 = low average, 3 = average, 4 = average high, 5 = high}
BS course	bachelor's degree track	categorical	{mechanical engineering, computer engineering...}

2.2.4 Associative Model Learning

The Live and Let Live (L^3) classifier [22], an associative algorithm, was employed to derive the student outcome predictions. The use of L^3 is a novel contribution of this work with respect to the previously mentioned studies since none of them apply this algorithm for student performance prediction.

L^3 consists of a subset of high-quality association rules, hereafter denoted as *strong classification rules*.

Table 2.9 *Course activity* features employed for predicting student academic performance

Feature	Description	Data Type	Domain
C-Mat	discretized frequency of learning material' downloads normalized to the maximum number of downloads made up to that point in time for course C	categorical	{H = high, F = average, L = little, N = no use}
C-down	discretized frequency of video lectures' downloads normalized to the maximum number of downloads made up to that point in time for course C	categorical	{H = high, F = average, L = little, N = no use}
C-str	discretized frequency of video lectures' accesses normalized to the maximum number of accesses up to that point in time for course C	categorical	{H = high, F = average, L = little, N = no use}

An association rule is an implication $X \rightarrow Y$, where X and Y are denoted as antecedent and consequent of rule $X \rightarrow Y$. .

For example, {(*Entry test*, [60,70]), (*Video-lectures accessed*, <5%)} → (*Outcome, fail*) is an association rule where {(*Entry test*, [60,70]), (*Video-lectures accessed*, <5%)} is the antecedent and (*Outcome, fail*) is the consequent. It indicates that the co-occurrence of two specific conditions, i.e., passing the entry test with a grade between 60 and 70 and accessing less than 5% of the video-lectures, is correlated with an exam fail.

Association rule extraction is commonly driven by support (sup), confidence (conf), and correlation (corr) quality indexes [2].

The support (sup) of a rule R is defined as $sup(R) = sup(X \cup Y)$ and indicates its frequency of occurrence in the source dataset; $sup(X)$ refers to the frequency of occurrence of the antecedent, while $sup(Y)$ to the frequency of occurrence of the consequent.

The confidence (sup) of a rule R is defined as $conf(R) = \frac{sup(X \cup Y)}{sup(X)}$ and indicates the rules strength.

For example, the association rule {(*Entry test*, [60,70]), (*Video-lectures accessed*, <5%)} → (*Outcome, fail*) has support equal to 33% and confidence equal to 100%, because in all the records in which the antecedent occurs the consequent occurs as well, i.e. if the entry test grade is between 60 and 70 and the number of video-lectures accessed is very low, the outcome is always *fail*.

When the rule consequent is characterized by relatively high support value, the confidence could be high even if its actual strength is relatively low [205]. For this reason correlation (corr) is considered; it is the ratio between how often the antecedent and the consequent are observed together and how often they would be expected to be observed together, given their individual support: $\text{corr}(X, Y) = \frac{\text{conf}(X \rightarrow Y)}{\text{sup}(Y)} = \frac{\text{sup}(X \rightarrow Y)}{\text{sup}(X)\text{sup}(Y)}$.

If corr(X,Y) is equal to or close to 1, itemsets X and Y are not correlated with each other. Correlation values significantly below 1 show negative correlation, whereas values significantly above 1 indicate a positive correlation between itemsets X and Y, i.e., X and Y co-occur more than expected.

For example, the correlation of rule {(*Entry test*, [60,70]), (*Fraction of video-lectures accessed*, <5%)} → {(*Outcome, fail*)} is $\frac{\frac{2}{6}}{\frac{2}{6}*\frac{3}{6}} = 2$. Hence, the rule correlation is positive.

A classification rule is *strong* if its support, confidence, and correlation values are above (analyst-provided) thresholds.

2.2.5 Profile Extraction and Ranking

The associative models generated by the L^3 classifier at different time points are collected and analyzed to gain knowledge about the classifiers' decisions.

Classification rules related to rate *fail* describe at-risk student profiles. For example, rule {(*Entry test*, [60,70]), (*Fraction of video-lectures accessed*, <5%)} → {(*Outcome, fail*)} describes a profile of students who have achieved fairly good test outcomes and who have not downloaded the video-recordings of the in-class lectures. Conversely, classification rules related to rate *pass* describe successful student profiles.

Profiles can be classified as (i) at-risk profiles, if they are peculiar to the success *fail*, or (ii) successful profiles, if they are peculiar to success *pass*. Note that student profiles are related to a given course and period of time. Hence, they may change while considering different courses and periods.

Another advantage of using an associative classifier is that it deals with missing data since even when some features values are null the rules that do not contain them can be used to derive profiles.

2.2.6 Experimental settings

1.2.6.1 Competitors

To answer RQ1, the following classifiers were considered as competitors: *Decision Trees*(`DT`), *Multi-Layer Perceptron* (`MLP`), *Support Vector Machines* (`SVM`), *Naive Bayes* (`NB`), *K-Nearest Neighbors* (`K-NN`) and *Random Forests* (`RF`).

Decision trees, as well as L^3, allow the derivation of association rules. However while associative classifiers perform a global search to extract rules satisfying some

quality constraints (i.e., minimum support), DT relies on a greedy (local) search that selects the most important attributes in turn based on the Information Gain or Gini Index and may discard important rules [220]. In addition, while DT derives the rules a posteriori by retracing the path of the tree (implying a hierarchy between items), associative classifiers extract, filter and order rules before including them in the classification model.

1.2.6.2 Training specifications

The data were trained using the *same course offering* and applying a stratified 5-fold cross-validation strategy.

The time complexity for training and testing the classification models ranged from a few seconds on simpler datasets to approximately one hour in the worst cases. However, most prediction models were generated in less than 60s.

1.2.6.3 Evaluation metrics

The models were evaluated by computing (i) Precision, (ii) Recall and (iii) F1-Score of the class *fail* and (iv) Balanced Accuracy. All considered metrics, except Balanced Accuracy, are specific to the class fail, since the main goal of this case study is to early detect at-risk students.

2.2.7 Results

RQ1) Are associative models as accurate as the best performing classifiers in predicting the success of university-level students?

Figure 2.1 shows for each course the scores achieved by the algorithms at various time instants. Examination sessions are denoted by the vertical dashed lines.

Algorithms scores do not increase before the autumn session for both first-semester (MA, CS, CH) and second-semester exams (PH, LA); both features related to educational material accesses and video-lectures streaming/downloads do not improve performance and predictions primarily rely on entry test and high school grades. At the beginning of the second semester, the performance trends experience

sharp growth for CS, MA and LA and lighter improvement for CH and PH because the autumn session exam outcomes are helpful in predicting summer session exam outcomes. Similarly spring session LA outcomes predictions benefit from summer session exam outcomes.

In order to assess the statistical significance of the performance variations, the Wilcoxon signed-rank t-test [88] was applied using a significance level equal to 0.5%. The results show that L^3 performed significantly better than `DT` at specific time points for the majority of the analyzed courses, while it performed as well as the best performing approaches (`K-NN`, `MLP`). Hence, the L^3 associative model could be deemed as a reliable model for early predicting student performance.

RQ2) What are the most discriminating features to forecast exam success at different time points?

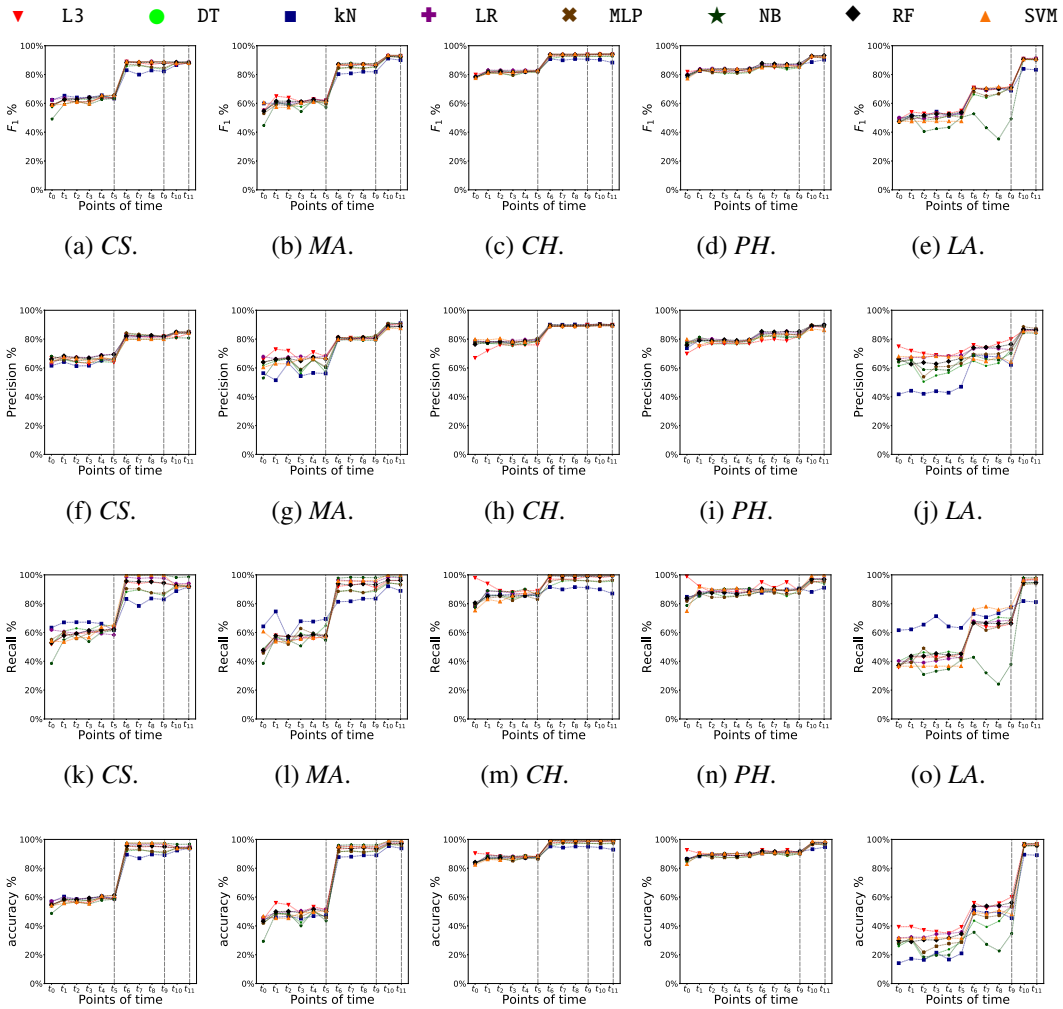
The frequency of occurrence of the single features was inspected.

Figure 2.2 shows the percentage of rules including specific features in their antecedent at three representative time points: t_0, before the entry test when only features related to students' background are available, t_1, before the beginning of the semester when students have already taken the entrance test and chosen the course of study, and t_5, at the end of the first semester when data on students' activity on the educational material and video-lectures are also available.

Figure 2.3 instead focuses on features contribution to LA course in the second half of the year grouping them into different categories (personal data and scholastic history, entry test and BS course choice, activity in two sample courses, MA and CS, educational material download and exam success) and their frequency of occurrence is compared at all the time points (from 0 to 9).

Some major findings were derived by looking at the figures:

- *The high school degree heavily influences students' performance*: the high school grade (HS-gr) is the most important feature at t_0 and it continues to be relevant during the whole academic year for most courses.
- *Age has a strong impact at the beginning of the year but gradually decreases*: this is due to a correlation with high school grades since older students generally have a lower school grade and they are also more likely to work while



(a) *CS*. (b) *MA*. (c) *CH*. (d) *PH*. (e) *LA*.

(f) *CS*. (g) *MA*. (h) *CH*. (i) *PH*. (j) *LA*.

(k) *CS*. (l) *MA*. (m) *CH*. (n) *PH*. (o) *LA*.

(p) *CS*. (q) *MA*. (r) *CH*. (s) *PH*. (t) *LA*.

Fig. 2.1 Algorithms' comparison in terms of F1-Score, Precision, Recall of class *fail* and Balanced Accuracy for predicting student academic performance

studying. The influence of this feature decreases during the time because motivated students learn to react by putting extra effort into the study and features related to student course interactions, as the use of video lectures or educational material and previous exams, grades, impact predictions.

- *The entry test grade (GRE-gr) is another determining factor*: from time instant t_1 onward is one of the most relevant features. The entry test assesses whether the student has basic skills in logic, mathematics and physics. Its impact on passing first-year exams may lead the university to alarm students who

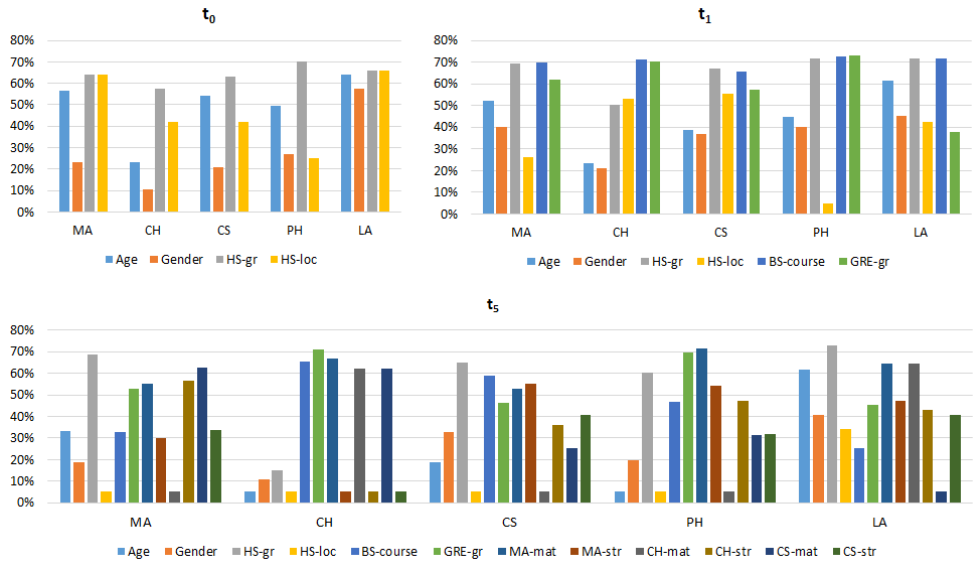


Fig. 2.2 Frequency of occurrence of the features appearing in the rule antecedents at different time points for predicting student academic performance

show more gaps, suggesting that they participate in remedial classes from the beginning of the year. In turn, professors can intervene by preparing teaching materials consisting of theoretical background and exercises on the prerequisites of their courses.

- *The BS course choice is also relevant*: the attitude of the students towards the different disciplines may depend on the perceived importance in their future.
- *The activities carried out in one course may influence other courses*: for example, activity in the MA course affects both the CS exam grade and second-semester exams grades, such as PH and LA.
- *The outcome of winter session exams affects the results of summer session exams*: for example, graph (f) from Figure 2.3 shows that passing the 1st-semester exams (MA-gr, CH-gr, CS-gr) has a strong influence on passing LA in the second semester.

Table 2.10 High-quality rules for predicting student academic performance

The rules are mined from Mathematical Analysis (MA) course training L^3 with the following parameters: minsup = 1%, minconf = 50%, mincorr = 2. Support and confidence values of each rule are averaged over the 5 cross-validation folds. "Before entry test", "Early 1st semester", "Mid-way 1st semester", "Close to 1st semester" exams indicate when the prediction had been carried out.

Num	Time ID	Body	Head	Support (%)	Confidence (%)	Lift	Description
			Before entry test				
1	t_0	HS-loc = Italy, HS-gr = 5, gender = F	pass	10.0 ± 0.3	86.5 ± 0.7	7	Very good high school grade, high school in Italy, female (independently of age)
2	t_0	HS-loc = Italy, HS-gr = 4, age = 0	pass	30.4 ± 0.2	79.2 ± 1.3	8	Good high school grade, high school in Italy, average age
3	t_0	HS-gr = 5, gender = M, age = −1	pass	14.1 ± 0.2	87.9 ± 1.0	4	Good high school grade, male, younger than average (independently of the high school country)
4	t_0	HS-gr = 1, gender = M, age = 3	fail	3.0 ± 0.1	90.7 ± 1.2	3	Very low high school grade, male, much older than average
5	t_0	HS-gr = 2, age = 1	fail	3.9 ± 0.1	89.7 ± 1.3	8	Low grade, older than average (independently of gender and high school country)
			Early first semester				
6	t_2	MA-mat = F	pass	14.9 ± 0.3	75.2 ± 0.7	8	Average use of MA material
7	t_2	MA-str=L	pass	24.2 ± 0.2	70.0 ± 0.7	6	Little use of MA videos, but soon (october), coherent with MA material
8	t_2	CH-str = L	pass	20.1 ± 0.2	71.7 ± 1.2	4	Streaming of other courses has positive impact even if no MA videos (shows students' engagement)
		MA-str = N, CH-str = L		7.6 ± 0.3	72.5 ± 0.3	5	
		MA-str = N, CS-str = L		5.5 ± 0.2	70.9 ± 0.9	8	
			Mid-way 1st semester				
9	t_3	MA-mat = H	pass	7.5 ± 0.3	78.9 ± 1.3	7	High use of MA material
10	t_3	MA-mat = F	pass	12 ± 0.2	76.1 ± 0.8	7	Average use of MA materials, confirms t_2
11	t_3	MA-mat = L	fail	14 ± 0.2	64.4 ± 0.5	7	Little use of MA material is not enough now (cfr t_2)
12	t_3	MA-mat = N, CS-mat = N, CH-mat = N	fail	17.2 ± 0.2	87.4 ± 1.3	8	No use of material (inactive) , confirms t_2
13	t_3	MA-str = L, CH-str = L	pass	15.2 ± 0.1	70.6 ± 0.9	7	Little of MA videos is enough, with or without other courses. Different from MA material: just-enough approach for video streaming
		MA-str = L, CS-str = L, CH-str = N		10.4 ± 0.1	70.2 ± 0.2	7	
14	t_3	CH-str = L	pass	24.9 ± 0.3	70.6 ± 1.3	4	Streaming of other courses has positive impact, confirms t_2
			Close to 1st semester exams				
15	t_4	MA-mat = F	pass	19 ± 0.1	78.2 ± 1.1	7	Average use of MA material, confirms t_2 and t_3
16	t_4	MA-mat = L	fail	21 ± 0.1	73.9 ± 0.8	8	Little use of MA material, confirms t_3
17	t_4	MA-mat = N, CS-mat = N, CH-mat = N	fail	13.2 ± 0.1	95.7 ± 0.6	7	No use of material (inactive) , confirms t_2 and t_3
18	t_4	CH-mat = H	fail	3.9 ± 0.1	78.8 ± 0.8	8	High use of another course material
19	t_4	MA-str = F, CH-str = F	pass	28.9 ± 0.2	70.2 ± 0.7	4	Streaming of other courses has positive impact, confirms t_2 and t_3
20	t_2	MA-mat = N, MA-str = L	pass	7.5 ± 0.1	90.2 ± 0.9	45	Streaming is effective even without access to material

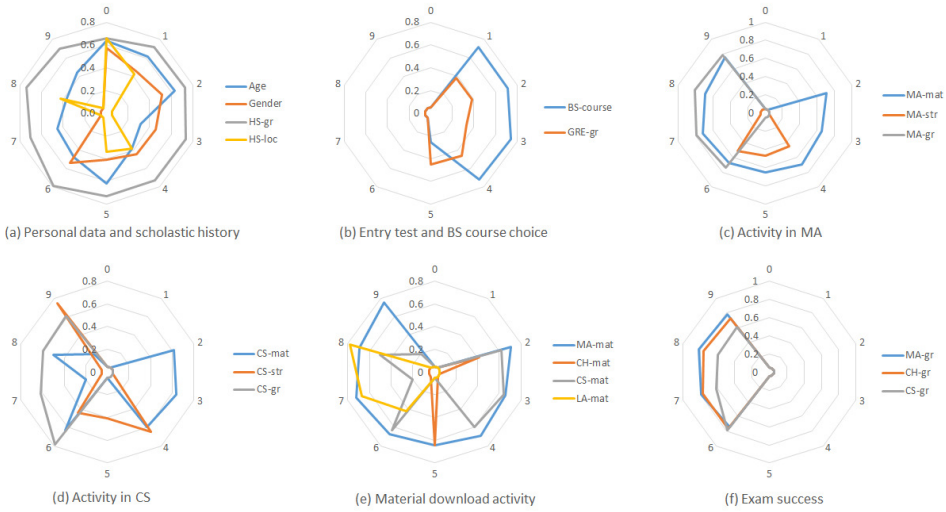


Fig. 2.3 Analysis of the relevance of single features for predicting student academic performance

RQ3) Which combinations of feature values have frequently been used to assign the exam success rates?

While analyzing the recurrence of features individually is useful to learn which ones impacted the prediction models the most, the rules inspection allows in understanding which prediction classes the feature values are associated with and in co-presence with which other features.

Table 2.10 shows samples of rules for predicting student outcomes in the first exam session of the Mathematical Analysis (MA) course.

Some rules reinforce and deepen previous findings: rules 1-5 confirm that the *higher the high school grade the greater the chance of passing the exam*, as well as *the older the student the greater the risk of failure*; rule 8, 14 and 22 support that *the activities carried out in one course may influence other courses* clarifying that *putting effort on other subjects is a strategy that pays at the beginning of the semester but not close to the exam session*: rule 8 highlights that working on other courses at t_2,$t-3$ increases the chance to pass the MA course, while rule 22 shows that the same behavior at t_4 yields opposite effects. Students should therefore be invited to work hard from the beginning of the semester, but they should also be warned that they should focus on a specific course when they are close to the exam.

In addition, new discoveries can be inferred:

1. *Students who start from the beginning of the course using the educational material have a good chance of passing the exam* (rule 6), while *those in the later stages of the course who do not use or poor use it are likely to fail* (rules 15,16,17). However, *if the lack of access to lecture material is compensated by streaming video lectures the student has a good chance of passing the exam* (rule 20).

2. *The use of the video-lecture streaming service is always positive*, even if limited. This is valid for the use of MA video-lectures (rule 13), but also for the use of other course video-lectures (rules 8). Encouraging students to actively use the service is another fruitful action to prevent failure and dropout.

Rule analysis also allows for specific intervention in even less representative student profiles (that is, when the rule support is low) when the confidence and correlation of the rule are high (e.g. rules 4,18,20).

2.2.8 Remarks and future improvements

This section demonstrates that the use of associative algorithms can help accurate early prediction of student outcomes, revealing which are the most successful and most at-risk behaviors. Some additional factors could be analyzed in the future:

- Test other associative algorithms to assess which is best at employing rules for classification.
- Set higher confidence values for rule selection to increase precision even if some students are unclassified.
- Apply the method to other courses in the following years to check whether the impact of the entry test and high school grade is still that significant.
- Test the quality of predictions on the same courses in different offerings in successive academic years to test whether performance remains the same.

Finally, since this study looks at several heterogeneous courses, features peculiar to the course topics were not taken into account; in the next section, on the contrary, a study will be presented where features used for prediction are related to the

students' activity on Version Control Systems, that is specific to the case study, i.e. a Java programming course, and to learning settings since data were recorded from laboratory sessions.

2.3 Time dependence analysis of exam performance predictors based on Version Control System features

Version control systems (VCS) are tools responsible for managing changes to computer programs and are used by programmers to share and track code changes over time.

GitHub collaborative platform is one of the most popular VCSs. Its potential to enrich the educational experience has already been established [242]. It can foster collaboration among students during university projects [70, 244] as well as individual learning [26, 85].

VCS are also excellent resources for collecting data on student activity (e.g., number of commits, number of days on which there was at least one commit operation, the average number of commit operations per date, number of lines of code added during the assignment completion) to derive statistical correlations with exam grades [95, 199, 214].

Other research [86, 87, 142, 253] formed VCS-based features from the log files containing students' interactions with VCS and used them as input for ML algorithms to predict students' performance.

Similarly, this section presents a case study on a university-level Object-Oriented Programming course in which students' activities in lab assignments were tracked via Github and the recorded data were used to early predict student outcomes.

The main contributions of this work are here summarized:

- **Time-dependent ML model training.** existing approaches are *time-invariant*,i.e. they are trained on all data recorded during the course. However, the student-VCS interactions are inherently time-dependent. To effectively support the

early prediction, different ML models were trained at various points in time on an intermediate set of statistics.

- **ML models' comparison based on visual explanations.** ML models trained on VCS usage data are typically used as black boxes. Conversely, the ML models were inspected by exploiting a state-of-the-art explainability AI method, namely SHapley Additive exPlanations (SHAP) [130]. The adopted implementation of SHAP provides a visual explanation.

2.3.1 Learning context

The data were derived from an Object-Oriented (OOP) programming course in the second year of the Computer Engineering B.S. degree, held at the Polytechnic University of Turin in the spring 2020. The lecturer chose Java as the programming language.

Most students are first-time attendees.

The course consists of over 70 hours in the classroom, including both lectures introducing the topics and live coding sessions presenting and discussing programming assignment solutions, and 20 hours in the lab dedicated to the development of programming assignments.

During the semester, students are presented with intermediate lab assignments, which are not graded. The intermediate assignments exhibit an increasing difficulty level and focus on the topics covered up to that moment in the lectures. The evaluation process adopted for such assignments includes the following steps:

1. The teacher prepares an initial project and uploads it on the VCS.
2. The student develops a small program at the end of which they must submit written code on the VCS.
3. Tests prepared by professors are automatically performed on the submitted code via JUnit automated testing framework. The following outcomes are possible for each test:
    - Success: the execution of the tests method reached the end and all assertions were verified.

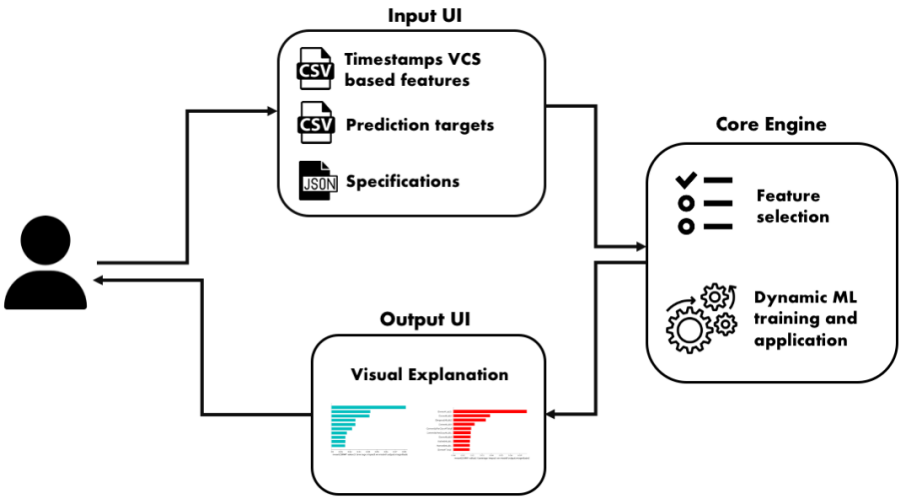
- Failure: one assertion was not verified and the execution of the test method is then interrupted.
- Error: during the execution of the tests method an unexpected exception is thrown (e.g. `NullPointerException` that signals the attempt to use a null reference), also in this case the execution is interrupted.

Similarly, the student's course outcome was determined by a final computer-based exam in which students submit an initial version of the code after their work in class; then they receive test output and are given the opportunity to review the code at home before the final submission. The teacher assigns grades ranging from 18 to 30; for this study, they were discretized into three categories: C (18 to 21), B (22 to 26), and A (27 to 30).

In the considered course offering, four exam calls were available. Students can reject an exam grade and retake it at a later call. Students enrolled in the course who do not participate in any exam calls are labeled as dropout.

2.3.2 Research Methodology

Fig. 2.4 The *VESPE* architecture



Prediction of student outcome is achieved through *Visual Explainable Student performance PrEdictor* (*VESPE*) [2], a Python-based application that allows end-users to automatically train different machine learning models at different course stages.

VESPE architecture is depicted in Figure 2.4.

It consists of two main user interfaces (UI), namely the *input UI* and *output UI*, and of a *core engine*.

Input UI

Input UI allows end-users to specify via a csv file the input features and the prediction targets and to set via a json file the specification of the algorithm setup, i.e.:

- *Target type*, which details whether the input data has to be employed for classification or regression.
- *Feature categorization*, which allows different feature sets to be specified to train the prediction model both for each of them and overall. Features sets serve to differentiate features both semantically (in this study, for example, those related to the number of commits from those related to the commit quality) and temporally (to specify those to be considered for a given time instant). Categories are not exclusive, i. e. a feature may belong to more than one category.
- A set of *hyper-parameters* that are used to set up the ML algorithms; if they are not specified an automatic optimization is performed in the core engine phase.

Core engine

The core engine consists of two components:

- A feature selection step in which k best features are selected using various criteria: Chi-2 [157], Mutual Information [118] and F-test ANOVA [201]. k ranges from 5 to N with an offset of 5, where N is the total number of features.

[2] https://github.com/Loricanal/VESPE.git

- The classification phase that relies on various ML algorithms with automatic hyper-parameters optimization using grid search. Only the following classifiers were considered in this study: *K-Nearest Neighbors* (`K-NN`), *Support Vector Machines* (`SVM`), *Decision Tree* (`DT`), *Random Forest* (`RF`), *Gaussian Process* (`GP`), *Multilayer Perceptron*(`MLP`), *Logistic Regression* (`LR`), *Linear Regression* (`LNR`), and *Gaussian Naive Bayes* (`GNB`). The list of hyper-parameters used for grid search is reported in Table 2.11.

Table 2.11 Hyper-parameters for grid search in *VESPE*.

The following algorithms are implemented with the Python library Scikit-learn (link:https://scikit-learn.org/stable/)

Hyper-parameter	**Set of possible values**
K-Nearest Neighbors	
number of neighbors	2, 3, 4, 5, 6, 7, 8, 9
Decision Tree	
maximum depth	2, 3, 5, 7, 10, None
minimum number of samples split	2, 4, 6
minimum number of samples leaf	1, 3, 5
maximum number of features for split	None, auto, sqrt, log2
Random Forest	
maximum depth	2, 3, 5, 7, 10, None
minimum number of samples split	2, 4, 6
minimum number of samples leaf	1, 3, 5
maximum number of features for split	None, auto, sqrt, log2
number of estimators	10, 50, 100
Support Vector Machines	
kernel	RBF, linear
regularizer	0.025, 0.05, 0.01, 1, 10, 100, 1000
Default configurations for *Gaussian Naive Bayes, Gaussian Process*, *Multilayer Perceptron*, *Logistic Regression, Linear Regression*	

Output UI

VESPE employs SHapley Additive exPlanations (SHAP) [130] to explain the models since SHAP Values measures the features contributions on predictions [3]. The output

[3]Further clarifications on SHAP are given in the Appendix A.

UI provides end-users with visual explanations of the ML outcomes including multiple plots, of which the following were considered for this study:

- **Summary bar plot:** It shows the impact of the twenty features with the highest impact on the predicting model. It averages the SHAP Values achieved across all students thus providing a *global explanation* of the ML model. If the color of the bars is red then the impact of the feature is positive with respect to the target considered. Conversely, if the color is blue the impact is negative.
- **Force plot:** it supports the explanation of classification for a single sample data (one student) and hence it provides a *local explanation*. It shows the model output value for a specific class, the base value, i.e. the value that would have been predicted if we did not know any features for the current output, and the impact of each feature on the output. More specifically, it draws red and blue arrows associated with the features. Each of these arrows indicates how much the feature impacts the model (the longer the arrow, the bigger the impact) and how the feature impacts the model (a red arrow increases the model output value while a blue arrow decreases the model output value) for a specific class.

2.3.3 Predicted targets

Table 2.12 Target values for student performance prediction using Version Control System features in an Object-Oriented programming course

Target	Exam 1	Exam 2	Exam 3	Exam 4	Summary
Registered	379	196	107	42	650
Dropout	-	-	-	-	173
Pass	364	166	103	42	636
Fail	15	30	4	0	14
Grade A	164	64	21	8	251
Grade B	147	47	39	24	246
Grade C	53	55	43	10	139
Reject	35	20	13	0	-

The considered targets are listed here:

- **Success**: it discriminates between the students who passed the exam in at least one of the four calls (*Pass* class) and those who did not (*Not Pass* class); the

latter may be divided into two subclasses: *Fail* class formed by students have never passed the exam and *Dropout* class comprising students who performed a course dropout, i.e. who have never registered for any examination call. Note that failure indicates that the student at least attempted to complete the course while dropout who "threw in the towel".

- **Success Exam 1**: it differentiates the students who passed the exam at the first call (*Pass Exam 1* class) from those who did not pass (*Not Pass Exam 1* class), i.e. those who failed it or did not register.

- **Success Exam 2**: it differentiates the students who passed the exam at the second call (*Pass Exam 2* class) from those who did not pass (*Not Pass Exam 2* class), i.e. those who failed it or did not register. This target is useful to determine the contribution of the first exam attempts, and specifically in determining the contribution of the attempts with respect to the features derived from the labs.

- **Grade**: it takes into account students who passed the exam in at least one of the calls (636 students out of 823) and it denotes the higher grade (A, B, or C) achieved, distinguishing the following classes: *Grade A*, *Grade B*, *Grade C*.

Table 2.12 shows some statistics related to exam outcomes.

2.3.4 Feature engineering

The data recorded through the VCSs describe the student's activity in the labs (*Lab* features). In addition, attempts to succeed in the first examination were considered (*Attempts* features) to assess whether they contribute to improvements in the prediction for the second call. Tables 2.13 and 2.14 provide the complete list of features and their categorization. The categorization is useful for training *VESPE* core engine with different feature sets in order to understand which features have the greatest impact on predictions. Conversely, training the algorithms with all features allows deriving models that combine features from different categories and evaluating which features individually impact the most. *Commit count*, *Commit quality*, *Commit frequency*, *Submitted labs* and *Active days* categories have already been adopted in previous research (see, for example, [86] for further details).+ Conversely,

categories *Lab dropout* and *Exam attempts* were defined for the first time in this study. *Lab dropout* should not be confused with *course dropout*. The former (e.g. *Dropout#LabN*) is a persistence indicator that the student stopped taking labs after the *N*-th; the latter indicates lack of willingness to complete the course and earn certification [51, 179, 223].

Table 2.13 *Lab* features recorded through Version Control System employed for predicting student exam performance in an Object-Oriented programming course

Feature	Description
Commit numbers ($D = 6$)	
CommitCount#Lab{1,2,3,4,5}	Number of commits for each lab
CommitCount#Total	Total number of commits
Commit quality ($D = 18$)	
Passed#Lab{1,2,3,4,5}	Number of tests passed for each lab
Passed#Total	Total number of passed test
Error#Lab{1,2,3,4,5}	Number of tests that raised errors for each lab
Error#Total	Total number of tests that raised errors
Failed#Lab{1,2,3,4,5}	Number of tests failed for each lab
Failed#Total	Total number of failed tests
Commit frequency ($D = 6$)	
CommitsPerDay#Lab{1,2,3,4,5}	Average number of commits per day for each lab
CommitsPerDay#Total	Average number of commits per day
Active days ($D = 6$)	
ActiveDays#Lab{1,2,3,4,5}	Number of days in which at least 1 commit was done for each lab
ActiveDays#Total	Total number of days in which at least 1 commit was done
Submitted labs ($D = 6$)	
Done#Lab{1,2,3,4,5}	Lab were submitted or not
DoneLab#Total	Total number of submitted lab
Dropout ($D = 4$)	
Dropout#Lab{2,3,4,5}	Dropout#LabN: the student submitted labs 1...N-1

Table 2.14 *Exam attempts* features employed for predicting student exam performance in an Object-Oriented programming course

Feature	Description
PreviousAttempt	The feature indicates whether the examination was already attempted by the student in the first call; it takes the value 1 if it was attempted (regardless of the outcome), and takes the value 0 if it was not attempted.
PreviousAttemptPass	The feature indicates whether the examination had already been attempted by the student in the first call and passed (0 if it was not attempted or failed, 1 if it was attempted and passed). In practice, this indicates the students who refused the grade.
PreviousAttemptFail	The feature indicates whether the examination had already been attempted by the student in the first call and failed (0 if not attempted or attempted and passed, 1 if attempted and failed).

2.3.5 Experimental settings

Time-dependent ML training

Separate ML models were trained after each laboratory session. The available feature set changes over time because the values of some of the VCS-based features in Tables 2.13 are missing (e.g., *Passed#Lab5* after the first laboratory session), whereas other features take temporary values (e.g., *Failed#Total* derive the total count after the second laboratory session, *Lab dropout* features after Lab *n* consider the course to be finished after that session). For each target, the data were split in half to form the training and test sets using the *same course offering*.

Evaluation metrics

The classifier performance in predicting an arbitrary class c was evaluated using the following metrics: (i) Precision (Pr), (ii) Recall (Rc), F1-Score ($F1$) and (iv) Balanced Accuracy (Ab).of class c.

2.3.6 Results and discussion

The results were presented by answering the following questions:

RQ1) At which course stage does the exam outcome get predictable? What are the most discriminating VCS-based features?

RQ2) Is it possible to predict the course dropout based on VCS usage data, to prevent it?

RQ3) What is the impact of laboratory activities on the exam grades?

RQ4) What is the impact of the previous exam attempts on the upcoming exam success?

RQ5) Which strategies can educators put into practice based on the results achieved on different targets?

RQ1) At which course stage does the exam outcome get predictable? What are the most discriminating VCS-based features?

Table 2.15 Prediction performance for class *Pass* using Version Control System features in an Object-Oriented programming course

Features category	**Algorithm**	*Pr*	*Rc*	*F1*	*Ab*
Commit count	GP	0.98	0.95	0.96	0.94
Commit quality	RF	0.98	0.95	0.96	0.94
Commit frequency	DT	0.98	0.95	0.97	0.94
Active days	GNB	0.98	0.95	0.97	0.94
Submitted labs	GNB	0.98	0.77	0.80	0.86
Dropout labs	GNB	0.97	0.11	0.20	0.55
Lab	MLP	0.98	0.95	0.97	0.94
Lab with feature selection	GNB	0.85	1.00	0.92	0.98

features in an Object-Oriented programming course.

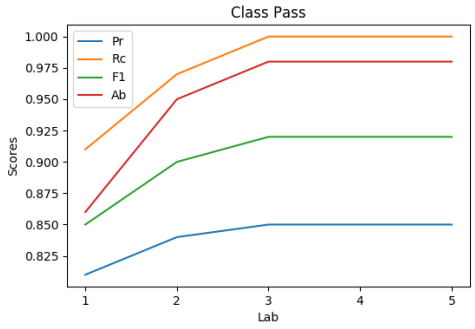
Classifier performance for class *Pass* on targets *Success* and *Success Exam 1* were reported in Tables 2.15 and 2.16 .

High-quality performance was achieved by ML algorithms for the target *Success* (e.g., balanced accuracy above 90% with all the features), whereas it was fairly high on target *Success Exam 1* (e.g., balanced accuracy above 70% with all features).

Table 2.16 Prediction performance for class *Pass Exam 1* using Version Control System features in an Object-Oriented programming course

Features category	**Algorithm**	*Pr*	*Rc*	*F1*	*Ab*
Commit count	RF	0.83	0.57	0.66	0.71
Commit quality	DT	0.75	0.65	0.69	0.70
Commit frequency	MLP	0.81	0.58	0.67	0.71
Active days	DT	0.82	0.54	0.65	0.70
Submitted labs	RF	0.95	0.46	0.62	0.71
Dropout labs	GP	0.60	0.79	0.68	0.59
Lab	SVM	0.86	0.51	0.63	0.70
Lab with feature selection	SVM	0.86	0.51	0.63	0.70

(a) Early prediction for class *Pass*.



(b) Early prediction for class *Pass Exam 1*.

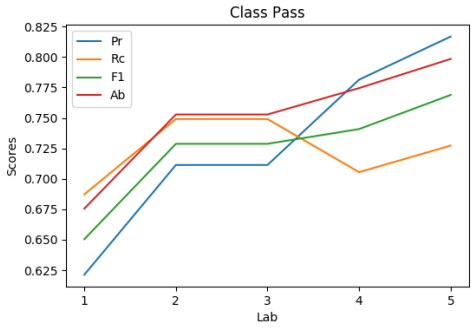
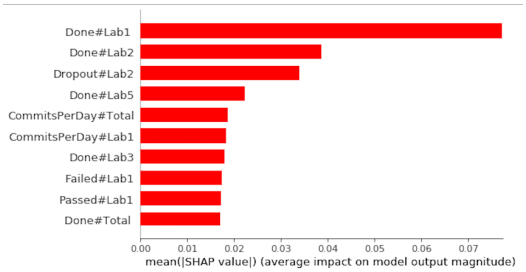


Fig. 2.5 Early prediction performance for classes *Pass* and *Pass Exam 1* using Version Control System features in an Object-Oriented programming course

(a) Summary bar plot for class *Pass*.



(b) Summary bar plot for class *Pass Exam 1*.

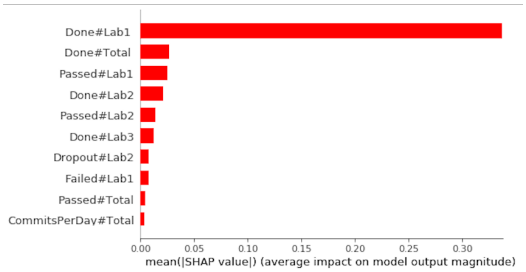
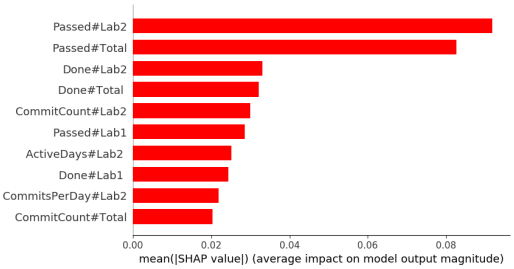


Fig. 2.6 Classes *Pass* and *Pass Exam 1* global explanation with the model estimated before the first exam call for student exam performance prediction using Version Control System features in an Object-Oriented programming course

Classifiers' performances appear to be rather similar for all the considered feature sets, except *Dropout labs* features which achieved the worst performance and are unsuitable for discriminating the exam success by itself.

(a) Summary bar plot for class *Pass*.



(b) Summary bar plot for class *Pass Exam 1*.

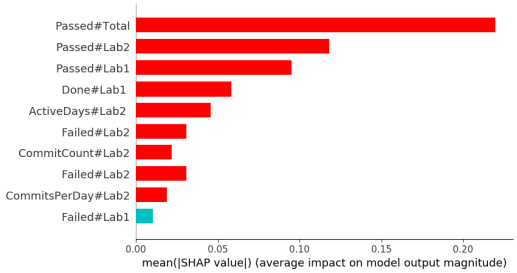


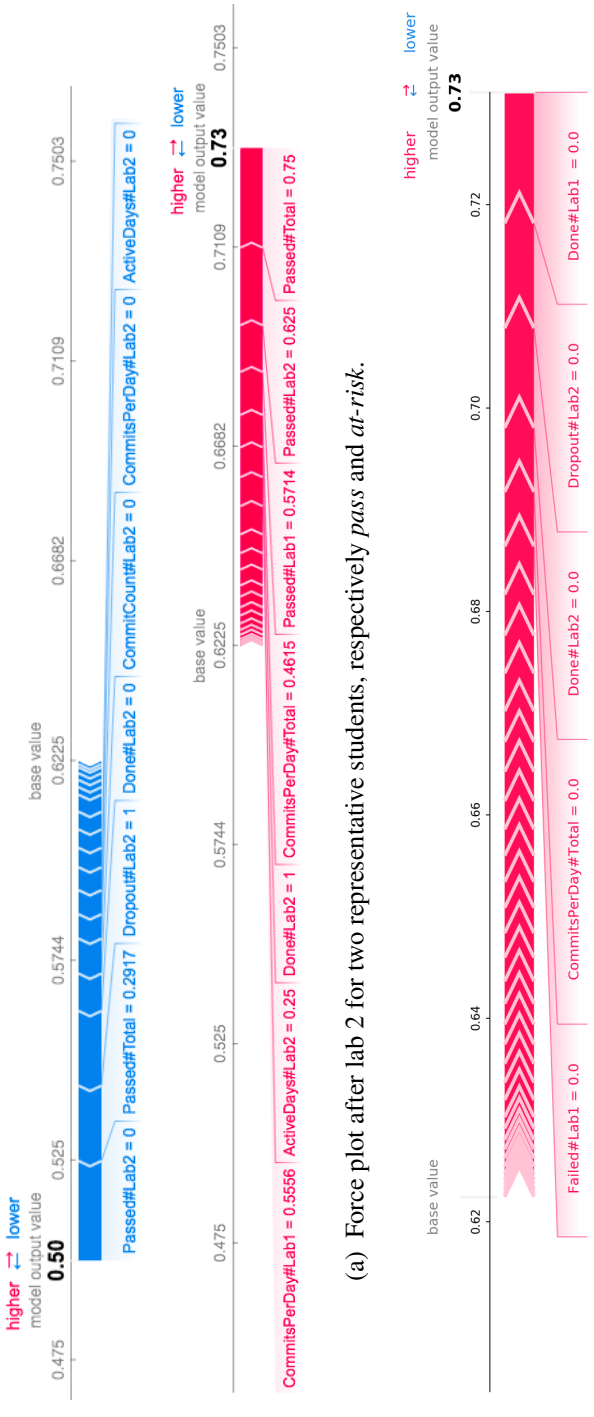
Fig. 2.7 Classes *Pass* and *Pass Exam 1* global explanation with the models estimated after lab 2 for student exam performance prediction using Version Control System features in an Object-Oriented programming course

The quality of the predictions is fairly high even at the early course stages (see Figures 2.5a and 2.5b).

Figures 2.5a and 2.5b show the time dependence of the performance measures. The quality of the predictions is fairly high even at the early course stages.

To better inspect this phenomenon, feature contributions were examined both on the model obtained after finishing all labs (Figure 2.6) and on the model derived after the second lab (Figure 2.7a). In both cases, the most significant features are related to the first two labs: however, the explanation related to the last time point reveals that it is enough to carry out the first two labs independently of past tests (almost only *Submitted labs* features related to the first two labs); moreover, features related to later time instants appear with minor contributions (i.e. *Done#Lab3*, *Done#Lab5*, *Done#Total*). *Dropout#Lab2*, which identifies the students who stopped doing the labs starting from the second one, has also a significant positive impact. The reason is that *Dropout#Lab2* is implicitly related to the student participation in *Lab1*, which appeared to be the most discriminating feature (*Done#Lab1*). In contrast, the model derived in the second time instant performed the prediction relying more on the *Commit quality* features related to the first two labs, i.e. *Passed#Lab2* and *Passed#Total* (note that *Passed#Total* considers only the first two labs at this time instant).

Hence, it is enough to do a few labs to pass the exam (sometimes only the first with many tests passed).



(a) Force plot after lab 2 for two representative students, respectively *pass* and *at-risk*.

(b) Force plot for a *dropout* student after lab 3.

Fig. 2.8 Individual explanation for student exam performance prediction using Version Control System features in an Object-Oriented programming course

Figure 2.8a compares two examples of students after laboratory 2: the first one is an *at-risk* student: she/he was marked as *Not Pass* for target *Success* because she/he is a *moderately engaged* student who only submitted the first lab passing a few tests. The instructor may act preventively at this point of time by alerting the student and inviting her/him to continue with the labs, putting higher effort. The second one was classified as *Pass* for target *Success Exam 1* and shows an average positive attitude: she/he is *highly engaged* up to this point of the course because he took both the first two labs passing an average high number of tests, working an average number of days with a medium-high frequency of commits.

RQ2) Is it possible to predict the course dropout based on VCS usage data, to prevent it?

Table 2.17 Prediction performance for class *Dropout* using Version Control System features in an Object-Oriented programming course

Features category	**Algorithm**	*Pr*	*Rc*	*F1*	*Ab*
Commit count	GP	0.84	1.00	0.91	0.97
Commit quality	RF	0.84	1.00	0.91	0.97
Commit frequency	DT	0.85	1.00	0.92	0.98
Active days	GNB	0.85	1.00	0.92	0.98
Submitted labs	GNB	0.84	1.00	0.91	0.97
Dropout labs	GNB	0.29	1.00	0.46	0.68
Lab	MLP	0.85	1.00	0.92	0.98
Lab with feature selection	GNB	0.85	1.00	0.92	0.98

Table 2.17 shows the classifier achieved good results on class *Dropout*. The performance was fairly good even at the early course stages (see Figure 2.9), then they increased until the third laboratory. After that, we did not observe any further performance improvements.

The visual explanations for dropout are depicted in Figure 2.10. All lab features are negatively correlated with the classifier outcomes; hence students who do not attend any laboratory or only the first laboratory with little effort are marked as dropout.

This assumption is confirmed by the sample student in Figure 2.8b. She/he was classified as *Dropout* because she/he was *not engaged*: she/he did not take the first two laboratories.

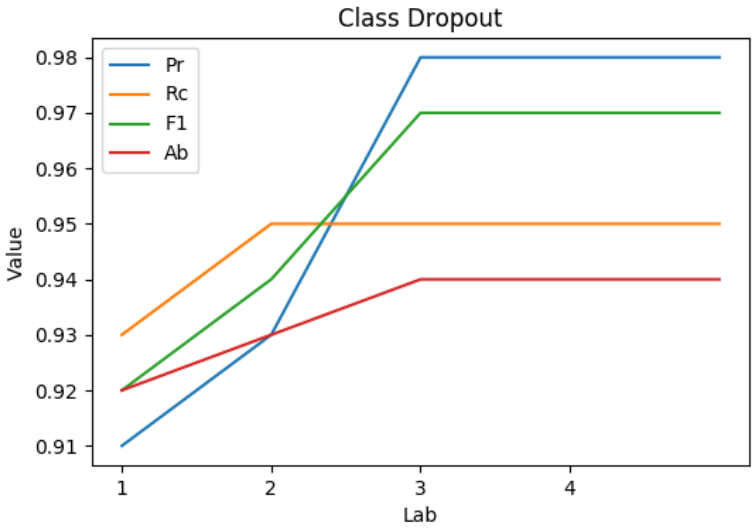


Fig. 2.9 Early prediction performance for class *Dropout* using Version Control System features in an Object-Oriented programming course

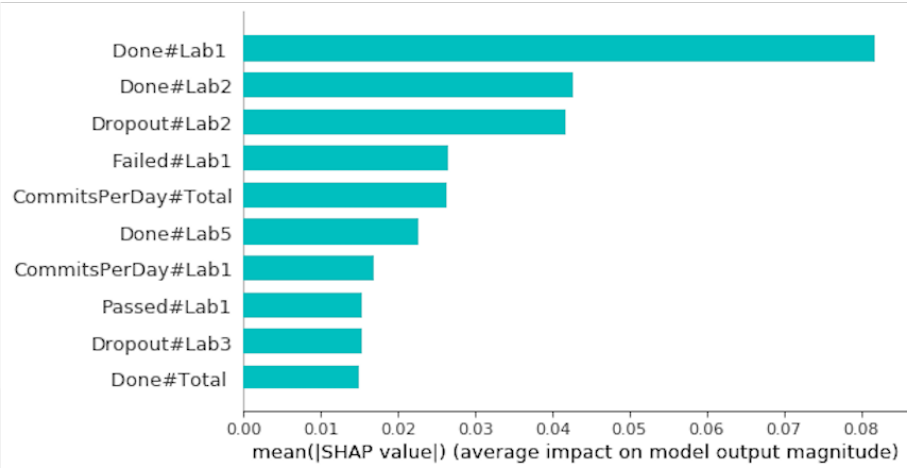


Fig. 2.10 Class *Dropout* global explanation for student exam performance prediction using Version Control System features in an Object-Oriented programming course

RQ3) What is the impact of laboratory activities on the exam grades?

Classifier performance on classes *Grade* was fairly low (see Table 2.18). As for previous classes, the contributions of the features were inspected to figure out which factors were used by the model to discriminate between classes (Figures 2.11). They are quite diversified and noisy:

Table 2.18 Prediction performance for class *Grade* using Version Control System features in an Object-Oriented programming course

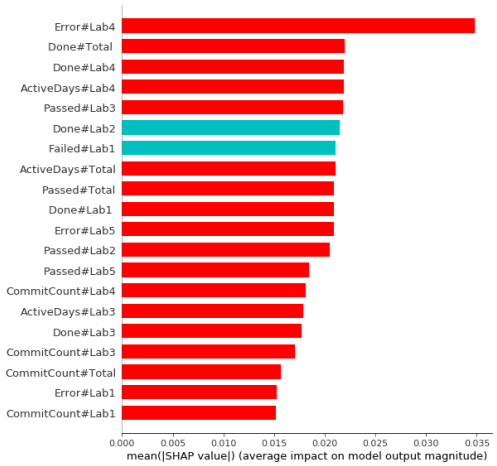
Features category	Algorithm	Class *Grade C*				Class *Grade B*				Class *Grade A*			
		Pr	*Rc*	*F1*	*Ab*	*Pr*	*Rc*	*F1*	*Ab*	*Pr*	*Rc*	*F1*	*Ab*
Commit count	RF	0.77	0.49	0.59	0.72	0.58	0.70	0.63	0.69	0.68	0.66	0.67	0.73
Commit quality	GNB	0.58	0.60	0.56	0.73	0.66	0.59	0.61	0.69	0.68	0.68	0.68	0.73
Commit frequency	KN	0.55	0.65	0.59	0.75	0.61	0.63	0.62	0.69	0.66	0.57	0.61	0.69
Active days	GNB	0.54	0.71	0.60	0.76	0.65	0.57	0.60	0.68	0.67	0.59	0.63	0.70
Submitted labs	GNB	0.54	0.71	0.62	0.77	0.89	0.50	0.63	0.73	0.60	0.75	0.66	0.71
Dropout labs	SVM	1.00	0.43	0.60	0.71	0.97	0.34	0.50	0.66	0.51	0.99	0.67	0.68
Lab	GNB	0.49	0.74	0.59	0.76	0.75	0.48	0.58	0.69	0.65	0.68	0.66	0.72
Lab(feature selection)	GNB	0.49	0.74	0.59	0.76	0.75	0.48	0.58	0.69	0.65	0.68	0.66	0.72

- Students labeled as *Grade A* submitted many labs (*Done#Total* has a positive impact) with an effort specifically in Lab 4. Note that the feature *Error#Lab4* (that represents the commit errors related to Lab 4) has a positive impact on the *A* class: the students adopted a trial-and-error approach (*ActiveDays#Lab4* has also positive impact), which is quite common in programming language courses. The other features that are positively correlated with this target group are those highlighting intense and high-quality commits (students who made many commits - *CommitCount#Total*, and who have passed many tests, especially in laboratory 3 - *Passed#Total* and *Passed#Lab3*), and those emphasizing durable effort, especially for laboratories 3 and 4 (*ActiveDays#Total*, *ActiveDays#Lab3*, *ActiveDays#Lab4*).

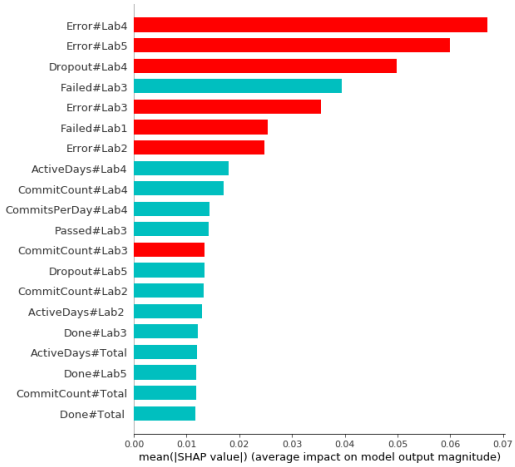
- Students labeled as *Grade B* consists of (i) students who have done at least the first 3 labs and have not done the remaining ones (feature *Dropout#Lab4* has a positive impact) and (ii) students who did activities on the last but making a relatively high number of errors (*Error#Lab4* and *Error#Lab5* have a positive impact on this class) and did not show much effort on lab 4 (*CommitPerDay#Lab4* and *CommitCount#Lab4* have a negative impact).

- Students labeled as *Grade C* consists of (i) students who did few laboratories (*Done#Total* has a negative impact), especially in the first part of the course (*Done#Lab5*, *Done#Lab4*, *Done#Lab3* have a negative impact, whereas *Dropout#Lab5* has a positive impact). In case they did the last lab, they were not very active in it: they devoted just a few days to it and failed many tests (*Passed#Lab5*, *ActiveDays#Lab5*).

(a) Summary bar plot for *Grade A*



(b) Summary bar plot for *Grade B*



(c) Summary bar plot for *Grade C*

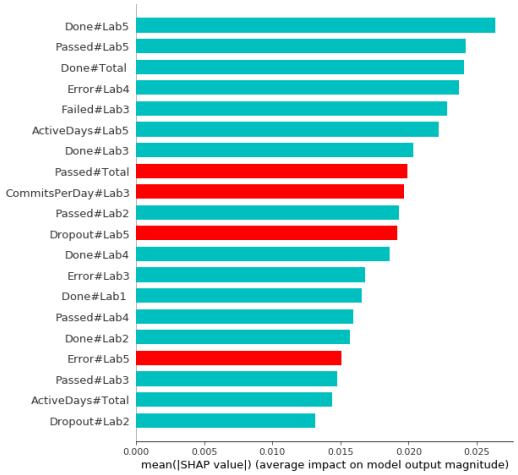


Fig. 2.11 Class *Grade* global explanation for student exam performance prediction using Version Control System features in an Object-Oriented programming course

RQ4) What is the impact of the previous exam attempts on the upcoming exam success?

Table 2.19 shows the impact of the *Exam attempts* features category on the classifier performance. The quality of the ML model trained only on *Exam attempts* was rather low. Tracing past exam attempts is not sufficient to predict the outcome of the next exams. Conversely, the results greatly increased by mixing *Exam attempts* and *Lab*

Table 2.19 Prediction performance for class *Success Exam 2* using Version Control System features in an Object-Oriented programming course

Features category	**Algorithm**	*Pr*	*Rc*	*F1*	*Ab*
Attempt	GNB	0.00	0.00	0.00	0.50
Lab	GNB	0.72	0.51	0.60	0.70
Lab + *Attempt*	RF	0.56	0.98	0.72	0.79
Lab + *Attempt* with feature selection	RF	0.56	0.98	0.72	0.79

features compared to the model trained on *Lab* features alone; hence attempts data are useful for complementing the other VCS-based features.

Table 2.20 Recommended strategies for student exam performance prediction in an Object-Oriented programming course

Class	**Early prediction time**	**Classification reliability**	**Behavioural features**	**Educator suggestion**
Pass	At the end of lab 2	Very High	Students who did the first two labs passing most tests	Good job, but keep working to get a high-grade
Pass Exam 1	At the end of lab 2	High	Students who did the first two labs passing most tests	Good job, but keep working to get a high-grade
Not Pass	At the end of lab 2	Very High	Students who have taken only one of the first two labs or who have taken both but with little effort	Warning! You are not putting enough effort into this course
Not Pass Exam 1	At the end of lab 2	High	Students who have taken only one of the first two labs or who have taken both but with little effort	Warning! You are not putting enough effort into this course
Dropout	At the end of lab 3	High	Student that has not taken any labs or has taken only the first one with little effort	Warning! Do not forget this course!
Grade A	At the end of all labs	Fair	Many submitted laboratories, intense and high quality commits, durable effort in terms of active days	Revise the work you have done, paying particular attention to labs 2 and 5
Grade B	At the end of all labs	Fair	Students who dropout labs after midcourse or who did all labs but made many errors in the last two or who did not show much effort on lab 4	Don't be satisfied with the work done in the first few lab sessions but do the others, paying particular attention to the last labs.
Grade C	At the end of all labs	Fair	Students who did not take labs or put little effort from the third lab onward	Don't be satisfied with the work done in the first few lab sessions but do the others as well, paying particular attention to the last labs.

RQ5) Which strategies can educators put into practice based on the results achieved on different targets?

Table 2.20 summarizes the targeted interventions that teachers can put into practice for each student category. Specifically, it reports the earliest prediction time, the classifier reliability estimated through standard metrics (see previous research questions), the behavioral features associated with the particular student type, and the suggested recommendation.

The recommended strategies summarize the main achievements of the empirical study and provide actionable knowledge about how to use *VESPE* in a real-life scenario. Notice that educators can conveniently exploit *VESPE* to tailor the level of complexity of both the laboratory activities and the exam. For example, if the ML model trained on target *Grade A* turns out to be highly accurate after the first laboratory then the difficulty of the past exam needs to be revised.

2.3.7 Remarks and future improvements

This section presented *VESPE*, a new method for deriving explainable models to early predict student outcomes in order to offer customized support depending on the predicted target. The method was tested in an Object-Oriented programming course. The results revealed that students who fail can be early detected based on the activity in the labs (e.g. commit count, commit quality...). The use of SHAP further enabled the derivation of feature importance on each individual prediction and outcome class. In the future work the feature set could be enriched, for example, JUnit tests in the labs may be categorized based on the learning topic, to derive features that take into account how many tests students passed for each topic. This will allow establishing which course contents are most critical in achieving a successful outcome and early alert students who show little effort on it.

2.4 *UNIFORM*: Automatic Alignment of Open Learning Datasets

Public educational datasets suffer from data heterogeneity, i.e. they hold different types of data (see the previous section 2.1.8). For example, some of them mainly focus on the students' interactions with the Learning Management System, others on the exam outcomes, still others on the student-teacher or peer-to-peer interactions. In addition, even when they are homogeneous, automatic integration is often not feasible, since even when two features from two different datasets have the same semantics they can be named differently. This section describes *UNIFORM*, already presented in [40], an integrated relational database schema that includes tables and

attributes able to handle heterogeneous data and automatically align new datasets via machine learning support. *UNIFORM* was evaluated on 11 open learning datasets.

2.4.1 Datasets description

The public educational datasets are listed below.

- OULAD[4] (Open University Learning Analytics Dataset), which contains data about student interactions with the learning management system. It was used for student *dropout*, *at-risk* and *grade* prediction [101].
- HARVARDX[5] and MITX[6], which contain the descriptions of the student activities in one edX platform course. They were used for student *dropout* prediction [206].
- COURSERA [7], which contains discussion threads presented in the forums of Coursera MOOCs. It was used for student *at-risk* prediction [92].
- PORT dataset[8], which collects student behavioral and lifestyle information as well as parent education level to perform student *grade* prediction in two secondary schools in Portugal.
- xAPI-Edu-Data[9] (XAPI), which consists of data about student behavior acquired in the University of Jordan and is used for student *grade* prediction.
- EPM[10], which contains information about student interactions with the online resources at the University of Genova and the exam grades.
- EDSA[11], which contains data about students' interactions with the online resources of the European Data science Academy portal.

[4] https://bit.ly/2m4a0NF
[5] https://bit.ly/2FLEz3f
[6] https://bit.ly/314niIv
[7] https://bit.ly/2mVuOas
[8] https://bit.ly/2lmoFDC
[9] https://bit.ly/2lmp2y0
[10] https://bit.ly/2ltgwgU
[11] https://bit.ly/2mc0NTG

- ISTM[12], which contains students' answers to survey questions about time management at Nottingham Trent International College.
- UoJ[13], which contains data about student performance.
- OUD (Our Institution Dataset), i.e. the one adopted in 2.2 for students academic performance prediction.

Details of the features included in each dataset as well as some relevant statistics about data size and schema complexity are given in Table 2.21.

Table 2.21 Statistics of public available educational datasets

The datasets are analyzed by exploiting the following data descriptors: (a) SPD (Student Personal Data), e.g. personal ID, age, gender, ethnicity; (b) SCD (Student Career Data), e.g. school degrees, entry test grades, educational modules enrollment; (c) EMD (Educational Module Data), e.g. available courses, course description, course prerequisites; (d) SAD (Student Assessment Data), e.g. exam grades, intermediate assessment evaluations; (e) ERA (Educational Resource Access), e.g. activities within a learning management system, online resources access, video-lectures streaming; (f) IAD (Interaction Activity Data), e.g. forum posts, peer-to-peer interactions, student-teacher interactions.

	OID (1)	**EDSA (2)**	**EPM (3)**	**HARV (4)**	**ISTM (5)**	**MITX (6)**	**OULAD (7)**	**COURSERA (8)**	**PORT (9)**	**XAPI (10)**	**UOJ (11)**
Data types	SPD, SCD, EMD, SAD, ERA	ERA	SAD, ERA	SPD, SCD, SAD, ERA, IAD	SPD, SCD, SAD	SPD, SCD, SAD	SPD, SCD, SAD, ERA	IAD	SPD, SCD, SAD	SPD, SCD	SPD, SCD
Dimensions (MB)	122.6	7.7	19.3	70.2	0.2	12.5	464.4	70.5	0.1	0.1	5.0
Number of tables	7	1	5	1	2	1	7	3	2	1	13

2.4.2 The *UNIFORM* schema

UNIFORM generalizes the data types provided by the open learning datasets previously discussed. Only a portion of them were looked at during the schema design: OID, EPM, HARVARDX, OULAD, COURSERA, PORT, xAPI-Edu-Data. The complete list of tables is reported in Table 2.22. The remaining ones (i.e., EDSA, ISTM, MITX, UOJ) will be employed to test the ability of the schema to handle new information; to guarantee the generality and flexibility, some attributes are supersets of the attributes in the original tables.

The USER table describes the demographics (e.g., gender, age, place of birth) including also the free time activities (e.g. alcohol week consumption). To discern

[12] https://bit.ly/2me1HyT
[13] https://bit.ly/2mxrq5L

Table 2.22 The *UNIFORM* schema

Table name	Attributes
INSTITUTE	**Institute_Id**, EduLevel, EntryGradeBase, FinalGradeBase, Name, Place, Type
USER	**User_Id**, AlchoolWeekendConsuption, AlchoolWorkdayConsuption, Birth_Place, Birth_Place_Type, Birth_Time, Disability, Education_Level, FamilyRelQuality, Familysize_Count, Father_Education_Level, Father_Job, FreeTimeQuantity, Gender, GoingOut_Duration, HealtStatus, ImdBand, InternetHomeAccess, Mother_Education_Level, Mother_Job, Nationality, NurseryAttendence, ParentStatus, Residence_Place, Residence_Place_Type, RomanticStatus
USER-INSTITUTE	**Institute_Id**, **User_Id**, Cds, ChoiceReason, Entry_Grade, ExtraEduSupport, Familysupport, Final_Grade, Guardian, HToSTravel_Duration, Higher, ParentAnsweringSurvey, ParentschoolSatisfaction, Registration_Time, StudentLevel, StudiedCredits, Unregistration_Time, User_Grade, User_Type
USER-COURSE	**Course_Id**, **User_Id**, Certified, DiscussionGroups_Count, Events_Count, Failures_Count, ForumPosts_Count, InteractingChapters_Count, InteractingDays_Count, MandatoryPosts_Count, PlayVideo_Count, ViewedAnnouncements_Count, ViewedCourseContent_Count, ViewedDashboard
USER-PRESENTATION	**Presentation_Id**, **User_Id**, Absences_Count, DiscussionGroups_Count, Events_Count, Explored, ExtraCVActivitiies, ExtraPaidClasses, ForumPosts_Count, Group, InteractingChapters_Count, InteractingDays_Count, LastInterction_Time, PartecipationSessions_Array, PlayVideo_Count, Registration_Time, Unregistration_Time, ViewedAnnouncements_Count, ViewedCourseContent_Count, ViewedDashboard, WeeklyStudy_Duration
COURSE	**Course_Id**, Credits, Institute_Id, Name, Typology
PRESENTATION	**Presentation_Id**, Course_Id, Duration, End_Time, Lang, Lectures_Count, Semester, Start_Time, User_Id
ASSESSMENT	**Assessment_Id**, Course_Id, Expiration_Time, GradeBase, Institute_Id, Lecture_Id, Presentation_Id, Start_Time, Type, Weight
USER-ASSESSMENT	**Assessment_Id**, **User_Id**, Grade, IsBanked, Submission_Time
USER-EXERCISE	**Exercise_Id**, **User_Id**, Grade
EXERCISE	**Exercise_Id**, Assessment_Id, GradeBase
LECTURE	**Lecture_Id**, Lecture_Type, Order, Presentation_Id, User_Id
USER-LECTURE	**Lecture_Id**, **User_Id** Participation, RaisedHands_Count
VIDEOLECTURE	**Videolecture_Id**, Lecture_Id, Presentation_Id, Recording_Time, User_Id
FORUM	**Forum_Id**, Course_Id, Depth, File_Id, Forum_Chain, Lecture_Id, OgForum_Id, Og_Forum_Title, ParentForum_Id, ParentForum_Title, Presentation_Id, Threads_Count, Title, TitleTags_Count, Users_Count, Videolecture_Id
THREAD	**Thread_Id**, Forum_Id, Views_Count
POST	**Post_Id**, NormalizedPost_Time, Order, ParentPost_Id, Post_Time, Thread_Id, User_Id, Votes_Count, Words_Count
FILE	**File_Id**, Course_Id, Format, Lecture_Id, Presentation_Id, Title, User_Id
ACTIVITY	**Activity_Id**, ActionType, Activity_Time, Assessment_Id, End_Time, Exercise_Id, File_Id, Forum_Id, Idle_Time, Keystroke, Lecture_Id, Mouse_Click_Left, Mouse_Click_Right, Mouse_Movement, Mouse_Wheel, Mouse_Wheel_Click, Post_Id, Start_Time, Sum_Click, Thread_Id, Type, User_Id, Videolecture_Id

between student users, teacher users, or others the attribute *User_Type* was introduced. Table COURSE saves course information (e.g. number of credits), while table PRESENTATION svaes data related to course offering (e.g semester, start

time, duration, language). Student assessment data are stored in ASSESSMENT that generalizes different types of assessments (e.g, final exams, ongoing tests, etc.). Table LECTURE records data related to the exercises assigned during an assessment procedure. The video-lectures and the other related teaching materials data are respectively saved in VIDEOLECTURE and FILE, while information related to forums and posts in FORUM, THREAD, POST. Finally, the online activities indicators (e.g. clicks, mouse movements) are saved in Table ACTIVITY.

2.4.3 Manual alignment

Each attribute in the original dataset was linked to an attribute from *UNIFORM*. The percentage of matched features per *UNIFORM* table is reported in Table 2.23. The results show that *UNIFORM* integrates most of the original data attributes, but the percentage of matching per facet is relatively low due to the high heterogeneity of the input data.

2.4.4 Automatic alignment

a_x denotes an attribute of the dataset x; a_o is an attribute in the original dataset, while a_u is an attribute in the uniform one. Each attribute is described by:

- l_{a_x} : the attribute name.
- d_{a_x} : a small description.
- W_{a_x} : a bag-of-words related to the attribute.
- E_{a_x} : a set of Wikipedia pages links related to the attribute.

l_{a_o}, d_{a_o} are given by the original dataset authors, while W_{a_o} was derived by extracting the keywords from d_{a_o} using *TextRank* [24, 143]. The same information for *UNIFORM* (i.e. l_{a_u}, d_{a_u} and W_{a_u}) were manually set. In both cases E_{a_x} is computed from d_{a_x} using a variation of *Ensemble Nerd* [42] which keeps all named entities extracted from the NEL extractors used in *Ensemble Nerd*, maximizing recall.

Table 2.23 Comparison of publicly available educational datasets based on the percentage of matched attributes per *UNIFORM*'s table

	OID (1)	EDSA (2)	EPM (3)	HARV (4)	ISTM (5)	MITX (6)
LECTURE	60.0%	0.0%	40.0%	0.0%	0.0%	0.0%
PRESENTATION	55.6%	0.0%	22.2%	22.2%	0.0%	33.3%
USER-EXERCISE	0.0%	0.0%	100.0%	0.0%	100.0%	0.0%
POST	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
ASSESSMENT	50.0%	0.0%	40.0%	40.0%	30.0%	40.0%
EXERCISE	0.0%	0.0%	66.7%	0.0%	100.0%	0.0%
THREAD	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
USER-ASSESSMENT	60.0%	0.0%	60.0%	60.0%	40.0%	60.0%
USER-LECTURE	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
ACTIVITY	21.7%	26.1%	65.2%	0.0%	0.0%	0.0%
COURSE	80.0%	0.0%	40.0%	40.0%	60.0%	40.0%
VIDEOLECTURE	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%
USER	19.2%	3.8%	3.8%	19.2%	15.4%	19.2%
FORUMs	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
USER_INSTITUTE	31.6%	15.8%	10.5%	10.5%	15.8%	10.5%
INSTITUTE	42.9%	14.3%	14.3%	14.3%	14.3%	14.3%
USER-COURSE	14.3%	0.0%	14.3%	21.4%	14.3%	14.3%
FILE	42.9%	28.6%	0.0%	0.0%	0.0%	0.0%
USER-PRESENTATION	19.0%	0.0%	14.3%	52.4%	9.5%	52.4%

	OULAD (7)	COURSERA (8)	PORT (9)	XAPI (10)	UOJ (11)
LECTURE	0.0%	0.0%	0.0%	40.0%	0.0%
PRESENTATION	33.3%	66.7%	22.2%	33.3%	55.6%
USER-EXERCISE	0.0%	0.0%	0.0%	0.0%	0.0%
POST	0.0%	100.0%	0.0%	0.0%	0.0%
ASSESSMENT	70.0%	0.0%	40.0%	0.0%	60.0%
EXERCISE	0.0%	0.0%	0.0%	0.0%	0.0%
THREAD	0.0%	100.0%	0.0%	0.0%	0.0%
USER-ASSESSMENT	100.0%	0.0%	60.0%	0.0%	60.0%
USER-LECTURE	0.0%	0.0%	0.0%	75.0%	0.0%
ACTIVITY	21.7%	0.0%	0.0%	0.0%	0.0%
COURSE	40.0%	80.0%	40.0%	40.0%	60.0%
VIDEOLECTURE	0.0%	0.0%	0.0%	0.0%	0.0%
USER	26.9%	3.8%	73.1%	19.2%	34.6%
FORUMs	0.0%	75.0%	0.0%	0.0%	0.0%
USER_INSTITUTE	21.1%	15.8%	42.1%	36.8%	10.5%
INSTITUTE	14.3%	14.3%	28.6%	14.3%	28.6%
USER-COURSE	21.4%	21.4%	21.4%	28.6%	14.3%
FILE	57.1%	0.0%	0.0%	0.0%	0.0%
USER-PRESENTATION	19.0%	14.3%	28.6%	33.3%	9.5%

In order to automatically align a new dataset with the UNIFORM schema, all attributes pairs <a_o,a_u> were represented by a similarity vector $\vec{f_{a_o,a_u}}$:

$$[S_{fuzz}(l_{a_o}, l_{a_u}), S_{cos}(\beta(d_{a_o}),\beta(d_{a_u})), |(W_{a_o} \cap W_{a_u}|, |E_{a_o} \cap E_{a_u}|]$$

Table 2.24 Hyper-parameters for grid search in *UNIFORM*

The following algorithm were implemented with the Python library Scikit-learn (link:https://scikit-learn.org/stable/)

Hyper-parameter	**Set of possible values**
Random Forest	
maximum depth	2, 3, 5, 7, 10, None
minimum number of samples split	2, 4, 6
minimum number of samples leaf	1, 3, 5
maximum number of features for split	None, auto, sqrt, log2
number of estimators	10, 50, 100
Default configurations for *Multilayer Perceptron*	

$S_{fuzz}(l_{a_o}, l_{a_u})$ denotes the similarity measure between attributes names computed using *Token Set Ratio* metric defined in *Fuzzywuzzy* [14]. $S_{cos}(\beta(d_{a_o}),\beta(d_{a_u}))$ is the *cosine similarity* between descriptions' BERT embeddings $\beta(d_{a_o})$ and $\beta(d_{a_u})$ [61]. $|(W_{a_o} \cap W_{a_u}|$ and $|(E_{a_o} \cap E_{a_u}|$ correspond respectively to the cardinalities of the bag-of-words' intersection and the Wikipedia links intersection.

Using these features vectors, *Multilayer Perceptron* (`MLP`) and *Random Forest* (`RF`) classifiers were trained for each pair of attributes $<a_o, a_u>$. Records are labeled as 1 if two attributes have the same meaning (i.e., they represent the same knowledge) as 0 otherwise.

Default hyper-parameters were used for `MLP`, while they were optimized with grid search for `RF` (see Table 2.24).

2.4.5 Classifier evaluation

A 70%-30% hold-out validation with oversampling of class 1 (i.e., the minority class) was carried out to evaluate classifier performance in predicting attribute alignment.

The evaluation was conducted on the following (manually aligned) datasets: OID, EPM, HARVARDX, OULAD, COURSERA, PORT and XAPI.

Table 2.25 shows the classifiers performance in terms of classifier (i) Accuracy, (ii) Precision, (iii) Recall and (iv) F1-Score of class 1.

[14] https://github.com/seatgeek/fuzzywuzzy

Table 2.25 Classification evaluation scores to assess the *UNIFORM* ability to automatic align new attributes

	OID		EPM		HARV		OULAD		COURSERA		PORT		XAPI	
	MLP	RF	MLP	RF	MLP	RF	MLP	RF	MLP	RF	MLP	RF	MLP	RF
Accuracy	0.90	0.61	0.76	0.91	1.00	0.60	0.96	0.42	0.97	0.73	0.86	0.94	0.87	0.53
F1-Score(1)	0.07	**0.74**	0.35	**0.94**	0.08	**0.72**	0.03	**0.58**	0.06	**0.81**	0.35	**0.97**	0.12	**0.70**
Precision(1)	0.04	0.94	0.18	0.96	0.04	0.90	0.02	0.92	0.03	0.91	0.22	1.00	0.06	1.00
Recall(1)	0.90	0.61	0.76	0.91	1.0	0.60	0.96	0.42	0.97	0.72	0.86	0.94	0.87	0.53

The *Multilayer Perceptron* model is slightly more accurate than *Random Forest*, but the precision is fairly low. Hence, the performance of *Random Forest* is globally superior in terms of F1-Score.

2.4.6 Automatic alignment of new open datasets

To evaluate the ability of the classifiers to automatically align new datasets, the classifier was trained on the seven aligned datasets and tested on the four datasets excluded from the previous evaluation (i.e., MIXT, EDSA, ISTM, UOJ).

The classifier results are summarized in Table 2.26. Instead, Figure 2.12 shows the accuracy values achieved by the Random Forest classifier on each test dataset by varying the number of aligned datasets in the training set. As expected, the accuracy increases while enriching the classification model with newly labeled data. An 80% accuracy was reached by using all the seven aligned datasets in the training set.

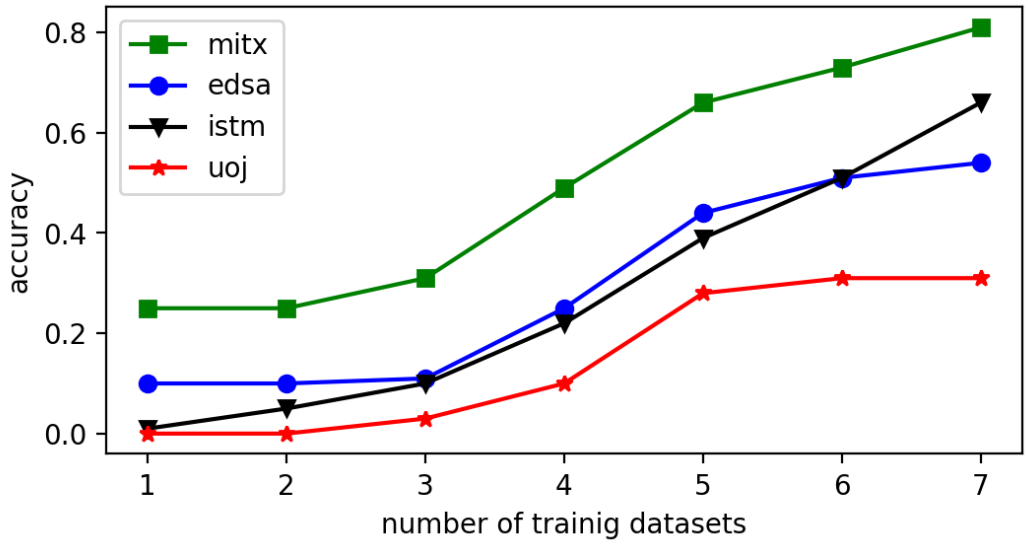
Table 2.26 Classification evaluation scores to assess the *UNIFORM* ability to automatic align new datasets

	EDSA		ISTM		MITX		UOJ	
	MLP	RF	MLP	RF	MLP	RF	MLP	RF
Accuracy	0.58	0.54	0.68	0.66	0.84	0.81	0.30	0.31
F1-Score(1)	0.02	**0.60**	0.02	**0.72**	0.16	**0.85**	0.03	**0.42**
Precision(1)	0.01	0.69	0.01	0.8	0.09	0.91	0.02	0.65
Recall(1)	0.57	0.53	0.68	0.65	0.82	0.80	0.29	0.31

2.4.7 Remarks and future improvements

The section proposes *UNIFORM*, a new data model integrating various open learning datasets that relies on ML models. Automated integration capabilities are promising

Fig. 2.12 Variation in Random Forest accuracy as the number of datasets used for training *UNIFORM* increases



since Random Forest Classifier reached an F1-Score on the minority class equal to or above 70% on 6 out of 7 datasets.

The current project leaves room for further extensions, e.g. the integration of multimodal data in the schema (e.g., video-lectures, slides). This opens new research challenges regarding the way to process and automatically integrate data sources in different formats and acquired from different media.

2.5 Discussion and guidance for future research

This chapter focused on student outcome prediction. While the prediction performance is promising in most studies, the meaning of "prediction" requires clarification. Most studies derived the supervised prediction models employing the outcomes of a one-course offering and tested them on the same offering; hence they derive statements related to the future using future information. This implies that the models may have real use in a later course offering or in similar courses. The two case studies presented demonstrate the ability of AI algorithms to recognize different classes of students. Training explainable models at different time instants allow for

identifying at-risk students and early intervention to alert them. In addition, this chapter presented *UNIFORM*, an integrated relational database schema that enables the automatic alignment of educational datasets attributes.

The literature still lacks analysis of the prediction models' portability and standards. Note that sharing models and source code does not imply that the data should also be espoused, overcoming potential privacy issues. A detailed description of data semantics is enough to figure out what type of information the models were generated from.

With this in mind some advice for future research reproducibility is summarized below:

- Releases data, source code, and trained models used for your experiments.
- Data must be released accompanied by the following metadata (i) a detailed description of their semantics (ii) the type of each variable (e.g., continuous, discrete, categorical) (iii) the maximum and minimum values of each feature and (iv) any pre-processing information (e.g., discretization, normalization..).
- If the data cannot be provided in full release them at least partially with all metadata even of the missing variables.
- If no data can be published release the metadata.
- Release source code via Version Control Systems (e.g. Github).
- If the code consists of many files use a simple organizational structure; alternatively, think about using Python or R notebooks.
- If you don't have time to tidy up it, better to release the dirty code than nothing at all.
- Trained models must be downloadable via a storage repository.
- Trained models must be accompanied by specifications on how to reuse them (e.g., language, libraries).

Chapter 3

Video-lecture Indexing

Video lessons are increasingly adopted in learning, either as a recording of classroom lectures or as additional support (e.g. in blended learning), or as the main learning resource (e.g. in massive open courses). In order to become an effective learning resource, video lectures must respect some guidelines and requirements including video indexing, i.e. the process of providing users a way to access and navigate video content easily[190]; it enables quick access to the content of interest in a long video lecture or in a whole course.

In most cases video indexing is a preliminary step for *search functions*, i.e. retrieving a portion of interest from a video through a textual query. Students are interested in watching small video portions to review specific topics, especially in preparation for the final exam.

This chapter focuses on the value of indexing video lectures (Section 3.1), on the need to make it automatic (Section 3.2), and on procedures for achieving it (Section 3.3), presenting a new one called *VISA* (Section 3.4).

Table 3.1 Summary of major studies proving the educational value of video-lecture indexing

Paper	Experiment Year	University	Course	Video Lectures	Number of students	Student Characteristics	Video indexing value
[248]	2006	Large university located in the southwest of the United States.	Introductory course in Management Information Systems (MIS)	1	138	Undergraduate students from seven departments across the campus, such as MIS, electrical engineering, communications, and arts. 92% freshmen, 8% sophomores. 59% male, 41% female.	Students who use video indexing environment will achieve perfomance improvement and higher satisfaction levels than those that presents non-interactive video
[212]	2010, 2011	University of Houston	Courses in Biology, Computer Science,Chemistry, Geology, and Mathematics	314	1167	612 from spring 2010, 555 from spring 2011. 53% of students from Biology, other students from Computer Science,Chemistry, Geology, and Mathematics. 17% freshmen, 23% sophomores, 31% juniors, and 29% seniors. 40% male, 60% female	Students felt that video indexing was helpful x
[23]	2009 - 2011	University of Houston	Courses in Biology, Computer Science,Chemistry, Geology, and Mathematics	between 7 and 50 videos per course (mean = 25.13, S = 9.89)	2300	Data were collected at the end of each of five semesters between spring 2009 and spring 2011. 73.6% of students from Biology, 13% from Computer Science, 5.8% from Geology, 3.9% from Chemistry and 3.8% from Mathematics or Physics. 23% freshmen, 23% sophomores, 29% juniors, 25% seniors. 39% male, 61% female	Students felt that search engine was helpful most of the time in jumping to segments of the video they needed
[213]	2010 - 2015	University of Houston	73 courses in Biology, Computer Science, Chemistry, Geology, and Mathematics	1602	>4000		Students state that indexing and search features were considered very helpful and easy to use and lead them to get the grade they hoped for (performance satisfaction)

3.1 The learning value of video-lectures indexing

Both older and newer studies collected positive student feedback on video-indexing systems and demonstrated that their use causes a positive impact on grades.

In [248] LBA (Learning By Asking) system was presented: it provides users with a hierarchical content index for the video lecture being examined, allowing them to directly jump to any particular video clip/slide/note by clicking a sub-topic. Students were divided into two groups; the former operated a system with indexing capabilities available, the latter without. Both course satisfaction and final test scores were higher for students who have been provided with indexing features.

Similar discoveries have been presented in later studies: [212] emphasizes students' satisfaction with indexing, [23] evidences the search engine value and finally [213] highlights the benefits in student learning outcomes pointing out that students agree that video indexing was helpful for reviewing and getting the grade they hoped for.

Table 3.1 summarizes the key statistics and major findings of these studies, that demonstrated the usefulness of video-indexing systems for educational purposes.

3.2 The need to automate Video-lecture Indexing

Manual indexing of content is often a cumbersome process. Providing automated support for administrative tasks is a growing need in universities [153], considering the high number of work roles that are generally carried out by a few people, leading to an excessive workload of distance educators [29]. Time saved through the automation of tasks could free up them time to invest in other aspects of teaching.

Since video-lecture indexing in education requires high accuracy to discern two video portions related to two different facets of the same macro topic, the contribution of the following focuses more on the analysis of video-lectures methodologies and on the proposal of a new approach based on semantic annotations.

3.3 Video-lecture automatic indexing methodologies

Approaches to indexing video-lectures content can be manual [23, 73, 212], assisted [78, 134, 207], or automatic [10, 16, 20, 25, 72, 79, 99, 110, 135, 136, 156, 177, 195, 211, 213]. This dissertation focuses mostly on the latter. Some of them also integrate *search functions* [16, 195, 211].

The majority of automatic indexing procedures consist of a preliminary step of extraction of a textual document from the video and in a core engine to detect the indices from the text.

Text recognition can be accomplished through the audio channel, using Speech-to-Text technologies [72, 135, 136, 177, 195] or manually generated transcripts/captions [20, 25], or through the video channel, using Optical Character Recognition (OCR) [16, 110] to extract content from slides [211, 213] or blackboard [99] frames. The most used technology for Speech-to-Text is YouTube Data API [I] while Tesseract [II], JOCR [III] and MODI [IV] for OCR.

The semantic units forming the index can be different in nature. Most studies use *keywords*, i.e. the most significant words or n-grams, as semantic units [16, 110, 195, 211].

A common approach selects the most frequently repeated words in the text as keywords [195].

In [211] instead the set of keywords is formed of the words extracted from transition points; successive frames constitute a transition point if the fraction of pixels that are different based on the RGB criteria exceeds a minimum threshold. In [16] the authors adopt Jaccard similarity coefficient to eliminate the duplicate text frames, and then stop word removal and stemming algorithms are applied to get meaningful keywords from the scene. They were finally indexed via single-pass in-memory indexing (SPIMI).

A more complex approach simultaneously keeps into account different variables such as Term Frequency (TF), Inverse Document Frequency (IDF), font size, time on screen, domain importance and rare word analysis [110].

Other works [25, 79] index the content by *topic*. [25] uses topic modeling to integrate videos and blogs in a common semantic space of topics.

A more elaborate perspective is presented in [79]; the authors don't index content by topics of the considered video lecture as usual but link off-topics concepts to relevant video lecture segments to furnish a basic understanding of the concerned concepts; in this way students can catch up on the basic knowledge essential to understand the lecture contiient and on additional material to stimulate curiosity. The methodology consists of a (i) previous step to determine a coherence score of a given segment by cosine similarity followed by (ii) the use of TAGME [V] service to link the important phrases to the associated Wikipedia articles and finally (iii) the generation of concept similarity network to identify off-topic concepts. The conceptual similarity has been defined using two Wikipedia-based semantic relatedness measures: Dice coefficent and Normalized Google distance.

Video-lecture indexing can also be carried out via *keyphrases* [10, 20].

In [20] the authors define a features set considering dispersion, local span, C-value, cue words and Term Frequency–Inverse Document Frequency (TF-IDF) and use them as input for a Naive Bayes classifier to extract relevant key phrases.

[10] presents SemKeyphrase, an unsupervised cluster-based approach for keyphrase extraction from MOOC video lectures, and a ranking algorithm called PhraseRank that (i) calculates the importance score of each candidate with regard to its subsuming subtopic, (ii) computes the significance score of each cluster with regard to the MOOC video lecture and (iii) determines the "semi-final" list of top candidates from the ranked clusters of candidate keyphrases.

Some studies combine the previous approaches; [177] forms both topic-based and keyphrases based indexes by using an IT-specific thesaurus as support, while [156] presenting a complete set of model metadata for video-based learning objects: some metadata are manually provided (Title, Authors, Language, Intended User Role, Context, Learner's Age Range), others are automatically inferred (Subject, Topics, Key-terms, Semantic density, Interactivity type, Interactivity level, Learning style, Learner suitability, Difficulty, Typical learning time, Learning resource type) through appropriate domain ontologies.

Another promising direction is to use *tag clouds* as an index. For instance, in [72] the authors proposed VLB (Video Lecture Browsing), a video lectures indexing system based on timed tag-clouds.

Finally [135, 136] use *named entities* to index lectures [1].

The main advantage of using named entities rather than keywords is that they allow identifying different terms that refer to the same concept.

In [135] different annotators are compared: AutoMeta [VI] [144], CSO-Classifier [VII] [183], NCBO Annotator [VIII] [102] and OntoText [IX]. The authors conduct the analysis on three Computer Science courses: Computer Network, Computer Architecture, and Data Structure. In addition, the study considers different knowledge bases for both general purposes (DBpedia [X]) and domain-specific (Computer Network Ontology [XI] and Computer Science Ontology [XII] [182]) . A similar method was adopted in [136] with the only use of AutoMeta in Portuguese language but considering both DBpedia and Computer Science Ontology. In both studies an ontology related to the specific domain achieved more precise results; however, further analysis on more courses is required to confirm these findings. The authors [136] state that "there is still a lack of research in specialized ontologies in the field of computer science that adequately organize the concepts in this area".

Given this research direction, the next section describes a new video-lecture indexing method that relies on semantic annotations via named entities.

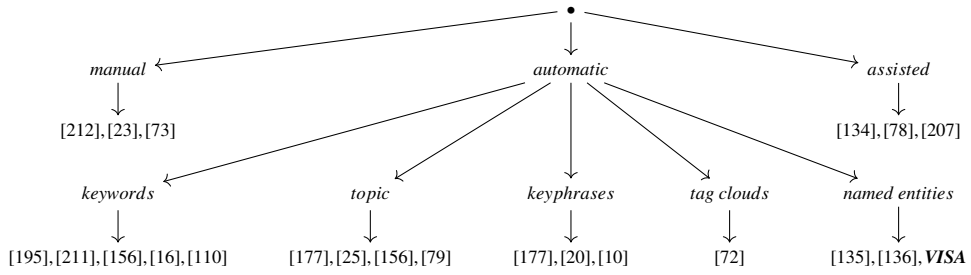


Fig. 3.1 Methodologies classification for Video-lecture Indexing

3.4 *VISA*: A supervised approach to indexing video lectures with semantic annotations

The approach described in this section has been previously published in [39] and implemented in our university for video lecture indexing.

[1] See Appendix A to clearify the meanings of named entity (NE), named entity recognition (NER) and named entity linking (NEL).

The proposed system, called *VISA* (*Video Lecture Indexing based on Supervised Approach*), enriches segments of video lectures with semantic annotations extracted from a knowledge base. The key characteristics of *VISA* can be summarized as follows:

- It processes video recordings of the face-to-face lectures in a semi-automatic way to generate video segments through recognition of slide changes.
- It analyzes both the text recognized in the video and the speech transcriptions in segments.
- It relies on multilingual knowledge bases thus enabling the indexing of video lectures in different languages.
- The supervised approach to extracting named entities combines (i) the syntactic properties of the text, (ii) the similarity between the content extracted from the text and the descriptors of the entities in the knowledge base, and (iii) the pertinence of the concepts to the main subject of the video lecture (to avoid selecting out-of-the-scope entities).
- To produce contextualized semantic annotations, the disambiguation process relies on a supervised approach that considers not only textual similarity but also the pertinence of the semantic concept with the main subject covered in the video lecture.
- It combines multiple semantic models to disambiguate text meaning and to perform Named Entity Recognition (NER) and Linking (NEL) tasks.

The method has been tested for the indexing of a database course in Italian. The language peculiarity is not negligible, since the majority of NER and NEL research is based on the English language, hence some findings are language-dependent and do not necessarily lead to better results when applied to other languages [165].

The performance of the proposed system was validated on a ground truth against the techniques available in the general entity annotation system GERBIL [216]. None of the studies from the previous session on video-lectures indexing compared their algorithms with state of art NEL extractors and GERBIL benchmarks.

The preliminary *VISA* results demonstrate the effectiveness and applicability of the proposed approach.

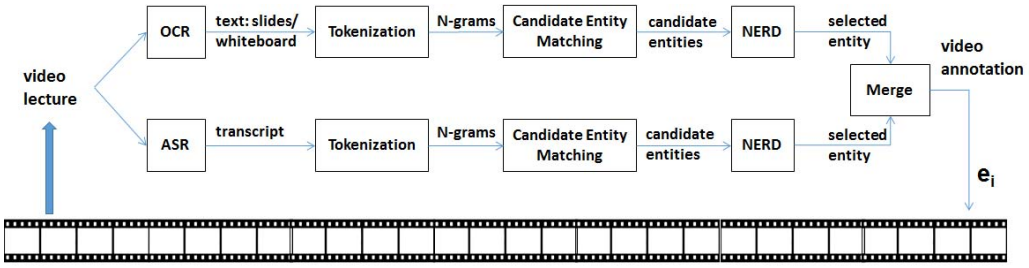
In addition to the contribution [39], a search engine was also tested, evaluating the retrieval of video lesson segments by MAP score.

The remainder of this section is divided as follows: subsection 3.4.1 details the methodology used, subsection 3.4.2 summarizes the preliminary experimental results and subsection 3.4.3 reports the results obtained. Finally subsection 3.5 analyzes possible future improvements.

3.4.1 Methodology

VISA is composed of different steps: segmentation, text extraction from the video channel, text extraction from the audio channel, text tokenization, candidate entity matching, Named Entity Recognition and Linking, and video annotation. The entire pipeline of the proposed approach is outlined in 3.2 and each step is detailed in the following.

Fig. 3.2 The *VISA* architecture



2.3.1.1 Segmentation

Slides transitions were considered to derive video segments through the following process:

- One frame per second was extracted from the video.
- Extracted frames were reprocessed; for each of them, only the part of the image corresponding to the paper used by the professor to write, or to the projected slide, has been extracted.
- The RGB color difference between consecutive frames was calculated; when a threshold value was exceeded, the slides were considered different.

2.3.1.2 Text extraction

The OCR-based *Tesseract* [II] library was adopted to extract text from the video channel.

The Google service *Youtube Data API* [5.2] was used to generate the speech transcription from the audio channel.

2.3.1.3 Text manipulation and tokenization

The extracted text is tokenized using the *Natural Language ToolKit* (NLTK) [XIII] in order to split the text into units, called tokens. In our context, a token is a single word occurring in the text. To filter out the words with little semantic meaning, words in the NLTK stopword list are removed prior to tokenization. Furthermore, punctuation was removed before processing the text. Since concepts can be described by a sequence of tokens (e.g., "New York City"), n-grams were derived from the text. N-grams are contiguous sequences of *n* tokens occurring in the text [32]. The n-grams with *n* between 1 and 5 were extracted.

2.3.1.4 Candidate entity matching

The proposed approach relies on the use of *Wikidata* (WD) knowledge base to infer semantic annotations [2].

Since *VISA* was tested for a database course, Computer Science Ontology (CSO) would have apparently been a viable alternative. However, it contains only high-level entities and more course-specific concepts are not included. In addition, the aim of this work is to propose a framework that can be tested in multiple domains and is not limited to computer science subjects.

[2] see Appendix A for an overview of Wikidata

All Wikidata triples containing the following predicates were extracted: *instance of* [3], *subclass of* [4], *part of* [5]. The *description* [6] and associated *labels* [7] were then extracted for each entity. The corresponding DBpedia and Wikipedia ones were retrieved via *sameAs* [8] property and *Wikimedia API*, respectively.

The entire Wikidata dump has been indexed by label using Elastic Search (ES) [XIV].

The candidate entity matching step identifies the candidate entities describing the n-grams in the text. N-grams can match an arbitrary number of entities (eventually zero if n-gram underlying information is irrelevant). To early discard misleading entity matches, candidate entities should satisfy a minimal quality constraint. Specifically, the similarity between the entity label and the n-gram should exceed a minimum similarity threshold. The adopted similarity σ is based on Levenshtein distance, i.e. the minimum number of single-character edits (insertions, deletions, or substitutions) required to change one string into the other. The formula for determining σ is defined below:

$$\sigma = 0.8 \bullet ratio(a,b) + 0.2 \bullet token\_sort\_ratio(a,b)$$

ratio is calculated by dividing the Levenshtein distance by the maximum of the length between string a and string b, *token_sort_ratio* is a variation of *ratio* where the strings are first tokenized, converted to lower case, stripped of punctuation, sorted alphabetically and joined together.

Similarity has been implemented by defining a custom metric for ES.

2.3.1.5 Named Entity Recognition and Disambiguation

In this study, the DBpedia ontology types [XV] were employed as reference types for Named Entity Recognition.

[3] http://www.wikidata.org/prop/direct/P31
[4] http://www.wikidata.org/prop/direct/P279
[5] http://www.wikidata.org/prop/direct/P361
[6] https://schema.org/description
[7] The main label http://www.w3.org/2000/01/rdf-schema#label and the alternative ones https://www.w3.org/2009/08/skos-reference/skos.html#altLabel are join to form the label sets related to an entity
[8] http://www.w3.org/2002/07/owl#sameAs

This ontology is generated from the manually created specifications in the DBpedia Mappings Wiki. Each release of this ontology corresponds to a new release of the DBpedia data set which contains instance data extracted from the different language versions of Wikipedia. For information regarding changes in this ontology, please refer to the DBpedia Mappings Wiki.

For each n-gram that is associated with multiple candidate entities, the disambiguation step aims at selecting the best candidate entity in the knowledge base. A decision tree model was adopted to perform this task. The model is trained on a training dataset collecting all the pairs (candidate entity, token) for which a correct matching is known. The class label attribute of the dataset indicates whether the entity assignment for the token is correct or not. To take its decision, the classification model analyzes the values of a set of additional features describing (i) the entity characteristics, (ii) the context of use of the token in the text, (iii) the similarity between token and label of the candidate entity. More specifically, the additional features used to characterize entities and tokens are summarized in Table 3.2.

Data features are classified into the following categories:

1. **Similarity**: textual similarity between n-grams and candidate entity label;
2. **Pertinence**: pertinence of the candidate entity to the main subject of the video lecture;
3. **Overlap degree**: overlap between the candidate entities of the tokens in the same n-gram or in close n-grams;
4. **POS property**: property of the token as a Part-Of-Speech [31].

Features of category *Similarity* are used to measure the similarity between the textual content of the tokens in the n-grams and the textual content of the entity labels. The more similar the token with textual information related to the entity, the more appropriate the matching with the entity. For each language used in the video lecture and supported by the knowledge base, a distinct copy of each Similarity feature is available.

Features of category *Pertinence* indicate the extent to which the candidate entity is semantically related to the context of the video lecture. To this aim, a *reference* entity describing the main subject of the lecture was selected in a semi-automatic

Table 3.2 Features characterizing the token-candidate entity relationship in *VISA* system

FEATURE ID	FEATURE NAME	DESCRIPTION
Category 1: Similarity between n-gram and candidate entity label		
1	*Label similarity*	Similarity score between the entity label and the n-gram
2	*Alternative label similarity*	Similarity score between the alternative label most similar to the n-gram and the n-gram
3	*Matching label similarity*	Similarity score between the matching label and the n-gram. The matching label is the most similar to the n-gram and it could be the entity label or one of the alternative
Category 2: Pertinence of the candidate entity with the subject		
4	*Wikidata graph similarity*	`1/dist(wkdt)` where `dist(wkdt)` is the distance between the reference and candidate entities in the Wikidata entity graph
5	*Wikipedia graph distance*	`1/dist(wkp)` where `dist(wkp)` is the distance between the reference and candidate entities in the Wikipedia content graph
Category 3: Overlap between candidate entities		
6	*Under*	number of candidate entities whose matching n-gram is nested into the n-gram under analysis
7	*Over*	number of candidate entities whose matching n-gram include the n-gram under analysis
8	*Concurrency*	total number of concurrent candidate entities
Category 4: Token-specific properties		
9	*Token position*	relative position of the token in the n-gram
10	*POS1*	NLTK POS tagger of the considered token
11	*POS2*	Polyglot POS tagger of the considered token
12	*POS1 after*	NLTK POS tagger of the following token
13	*POS2 after*	Polyglot POS tagger of the following token
14	*POS1 before*	POS tagger of the previous token computed with NLTK
15	*POS2 before*	Polyglot POS tagger of the previous token

way. Specifically, the teacher submits a keyword-based query (e.g., "Introduction to databases") through Wikidata or Wikipedia and automatically retrieves a list of the most pertinent entities, among which she/he can choose the reference one.

Given a reference Wikidata entity, a graph is built linking the reference entity to the other Wikidata entities through their *part of*, *instance of*, and *subclass of* properties. Similarly, using the Wikipedia encyclopedia a graph considering the links to other pages mentioned in the text was built. For instance if the page of "Structured Query Language" [9] contains a link to the page related to the JOIN statement [10] an arc in the graph has been added. The Wikipedia pages were aligned with Wikidata entities by means of the Wikimedia API [XVI]. For both graphs, the recovered entities at a distance greater than a specific threshold from the reference entity have been cut out because they would have been so many that the graph would not fit in RAM.

The Wikimedia REST API offers access to Wikimedia's content and metadata in machine-readable formats. Focused on high-volume use cases, it tightly integrates with Wikimedia's globally distributed caching infrastructure. As a result, API users benefit from reduced latencies and support for high request volumes.

The *pertinence* relationship between a candidate entity and a reference entity is modeled as a distance between the corresponding nodes in the graphs (expressed in terms of the number of ops to move from one entity to the other). The higher the distance value, the less similar are the candidate and reference entities. As an example, a teacher of a database course may select the Wikidata entity labeled as "Database" [11] as a reference entity. Candidate entity with labels "SQL" and "Structured Query Language" [12] has a distance equal to 2 in the Wikidata graph, while the entity labeled as "Programming language" [13] has a distance equal to 4 (similarity 0.6) from the reference entity. Hence, the former candidate entity is deemed as more pertinent than the latter one to the main subject of the course.

Features of category *Overlap degree* consider the textual overlap between multiple n-grams (e.g., "New York" and "New York City"). Since nested n-grams may be associated with different (potentially overlapped) sets of candidate entities, the degree of overlap between nested entities was taken into account. For example,

[9] https://it.wikipedia.org/wiki/Structured_Query_Language
[10] https://it.wikipedia.org/wiki/Join_(SQL)
[11] https://www.wikidata.org/wiki/Q8513
[12] https://www.wikidata.org/wiki/Q47607
[13] https://www.wikidata.org/wiki/Q9143

Table 3.3 reports the feature values related to the phrase *The Freddy Mercury Tribute Concert*.

Table 3.3 Example of overlapped n-grams

N-GRAM	WIKIDATA ENTITY	OVER	UNDER	CONCURRENCY
The Freddy Mercury Tribute Concert	0	0	4	4
Freddy Mercury	2	2	1	1
Mercury	3	3	0	4
Tribute	1	1	0	4

Features of category *POS property* indicate the information about part of speech corresponding to the token (e.g., noun, verb, adjective) and the relative position of the token in the n-gram. POS information is commonly used in Natural Language Processing to identify the parts of the text that are most likely to be correlated with semantically relevant concepts [31]. The output of the classification algorithm is a set of (candidate entity, token) labeled as Correct or Incorrect. For each n-gram, the candidate entities associated with the maximal number of correct tokens were selected as the most pertinent. For example, let us consider the n-gram "Query language".

Let us suppose that its corresponding tokens, i.e., "Query" and "Language", have two candidate entities each, i.e., the entities labeled as "SQL" and as "Programming language", respectively. The classifier assigns label *Correct* to the following pairs: (token "Query", entity labeled as "SQL"), (token "language", entity labeled as "Programming language"), (token "language", entity labeled as "SQL") while it assigns label *Incorrect* to the pair (token "Query", entity labeled as "Programming language"). Hence, the entity labeled as "SQL" gets three *Correct* labels, while the entity labeled as "Programming language" just one. Therefore, entity labeled as "SQL" is used to annotate the n-gram "Query language".

2.3.1.6 Video annotation

Each segment of the video recordings is annotated with the semantic information extracted from the knowledge base. Specifically, the annotated text was aligned to the video. In this way, the student can browse the semantic annotation of the video lectures to choose which segments of the video are worth considering in her/his study or revision. For example, if she/he is interested in the part of the lecture covering the

SQL topic, she/he can navigate the semantic annotation and watch only the segment annotated with the corresponding entity. In addition, they can query for a specific concept and can retrieve segments in which the corresponding entity is mentioned from the entire corpus of video lectures.

Note that the same entity may be present at several points, as the same concept may be taken up multiple times throughout the video lectures. When it is mentioned so many times in close proximity the student can infer that that part of the video lecture is narrowly focused on that content. Conversely, when a single occurrence of an entity is mentioned, it does not correspond to the main content of that portion of the video; however, students could be interested in understanding how the entity relates to other topics.

2.3.1.7 Search function

The system allows for a text query to enable students to retrieve video snippets of interest. The NEL algorithm previously described is employed to extract the entities from the query text; they are ordered on the basis of the value assumed by the Matching Label Similarity feature defined in Table 3.2. For each entity, the system returns the list of all the video fragments in which they appear.

3.4.2 Experimental settings

2.3.2.1 Competitors

The competitor frameworks considered in this study are *AGDIST* [215], *AIDA* [94], *Babelfy* [146], *DBpedia Spotlight* [55], *FOX* [197], *PBOH* [75]. These state-of-the-art NEL extractors provide multilingual support (including Italian) and had never been considered in previous studies that indexed video lessons with named entities, e.g. [135, 136].

2.3.2.2 Dataset

A preliminary evaluation of the effectiveness of the video indexing was carried out on a set of the 43 video lectures of a Database course. Each video has a duration of 70 minutes approximately. To generate the ground truth, 10 video segments were

randomly picked, the corresponding text (consisting of approximately 30 tokens each) was automatically extracted via Youtube API, and domain experts were asked to annotate it with Wikidata entities.

Overall, videos in the training data were enriched with 353 different entities (2,148 annotations overall, 50 annotations per lesson on average). The entities considered for ground truth, identified by the name DMBS-LARGE, include all concepts that find a match in Wikidata, not just those related to course content. From these only those related to database context were extracted, i.e. 84 different entities (880 annotations overall, 20 annotations per lesson on average), forming a second ground truth named DMBS-FOCUS.

For both ground truths, of the 10 segments per lecture, 7 segments were picked for train and 3 for test (overall 300 training segments and 130 test segments) by performing 10-fold cross-validation.

The following SQL statements and subtopics were considered to evaluate the search engine: “basi di dati” [14], “SELECT”, “WHERE”, “LIKE”, “ORDER BY”, “LIMIT”, “COUNT”, “JOIN”, “INNER JOIN”, “SELF JOIN”, “NON EQUI JOIN”, “GROUP BY”, “HAVING”, “HAVING COUNT”, “HAVING SUM”, “HAVING MAX”, “HAVING MIN”, “DISTINCT”, “IN”, “NOT IN”, “EXIST”, “NOT EXIST”, “query annidata” [15], “query correlata” [16]. For each of the previous keywords, snippets from the video lectures corpus were manually searched and used as ground truth for the retrieval task.

2.3.2.1 Evaluation metrics

Fewer papers assess indexing only qualitatively, relying on the opinions of a few experts [99, 195]. The most common evaluation approaches are divided between *survey-based* [72, 79] and *supervised metrics* [10, 16, 20, 99, 110, 177, 211, 213]. In the former teachers ask students to evaluate the system generally through surveys, analyzing their responses; in the latter, the automatically generated index was compared to a ground truth manually generated.

[14] “basi di dati” is the Italian translation of “database”
[15] “query annidata” is the Italian translation of “nested query”
[16] “query correlata” is the Italian translation of “correlated query”

Table 3.4 *VISA* Named Entity Linking performance on DMBS-LARGE dataset

Strategy	Precision	Recall	F1-Score
VISA	0.32	0.20	0.25
AGDIST	0.04	0.04	0.04
AIDA	0.034	0.11	0.02
Babelfy	0.08	0.20	0.05
DBpedia Spotlight	0.06	0.10	0.04
FOX	0	0	0
PBOH	0.17	0.17	0.17

The supervised metrics most frequently adopted are Precision, Recall and F1-Score [10, 16, 20, 99, 110, 177]. In this study, they were computed using GERBIL [216], an evaluation framework for semantic entity annotation. GERBIL also integrates public instances of previously discussed competitors. The knowledge base used for the evaluation has been DBpedia, hence the extracted entities links to DBpedia.

To retrieve DBpedia entities from the Wikidata ones extracted by *VISA*, the alignment between Wikidata and DBpedia previously discussed was used.

The search engine was evaluated using Mean Average Precision (MAP) (see Appendix A to further clarification).

3.4.3 Results

2.3.3.1 Video indexing

The performance of *VISA* was compared to the other approaches integrated in GERBIL; the final result is shown in Table 3.4 for DMBS-LARGE. *VISA* achieves significantly better performance than all competitors.

To gain insights into the process of disambiguation, the characteristics of the generated decision tree were explored. Figure 3.3 displays an example of decision tree trained from the video lectures used for the preliminary evaluation. The Gini impurity index indicates the quality of a feature domain split. A perfect separation

results in a Gini score equal to zero, while an equal distribution of the target value is achieved when the Gini index is equal to 1. In the training model, all the generated partitions have Gini index below or equal to 0.5.

The model explanation (see Figure 3.3) shows that the most discriminative data feature (i.e., the feature considered first in the top-down visit of the tree) is *Wikidata Graph Similarity*, i.e. a measure of the pertinence of the candidate entity to the main subject (Database), obtained through the Wikipedia graph. Hence, considering just textual similarity measures would be inappropriate for accurately selecting the assigned entities.

The results obtained considering only the entities related to "database" (i.e. DMBS-FOCUS dataset) are shown in Table 3.5. The system performances are lower than for DMBS-LARGE, revealing a poor ability to disambiguate the entities in the course domain, but still higher than the other extractors that reach in most cases null values. The strategy named *VISA-FILTERED* is a modification of the basic version that requires more manual effort: the educator should link each course topic to an appropriate Wikipedia page and only the corresponding entities are retained for annotating the text, while the others are discarded. With this modification, the precision increases sharply and consequently F1. This demonstrates that the precision of the system reached specifying only the main course entity was low because many entities not related to the Database course were recognized and not because the domain entities were recognized in incorrect moments of the lecture. Therefore the instructor has to consider the trade-off between choosing a single Wikipedia page and getting an annotation with out domain entities or manually specifying multiple pages and getting a context focused indexing.

This difference is not significant if the indexing phase is only developed to support search function because the student filters through the query the entities to search for.

2.3.3.2 Search function

As with video-lecture indexing, the performance of *VISA* was compared against GERBIL competitors, updating the semantic annotations used for retrieval depending on the algorithm. The complete list of results is shown in Table 3.6; *VISA* outperforms the other approaches.

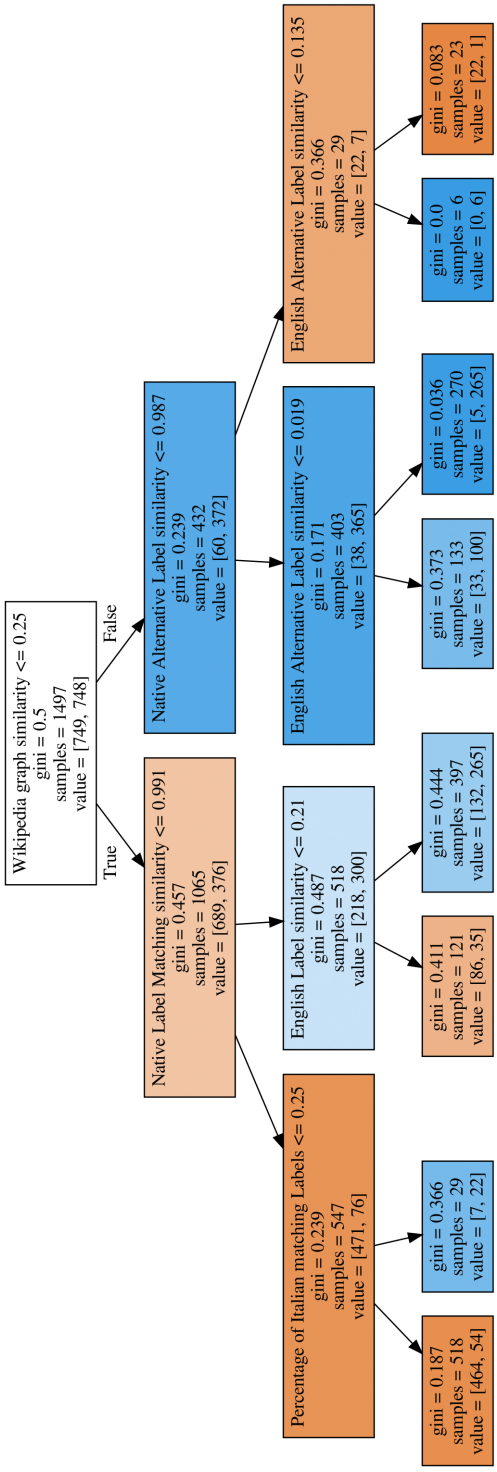


Fig. 3.3 Visual explanation of *VISA* decision tree

Table 3.5 *VISA* Named Entity Linking performance on DMBS-FOCUS dataset

Strategy	Precision	Recall	F1-Score
VISA	0.08	0.78	0.14
VISA-FILTERED	0.97	0.78	0.86
AGDIST	0.01	0.01	0.01
AIDA	0	0	0
Babelfy	0	0	0
DBpedia Spotlight	0	0	0
FOX	0	0	0
PBOH	0.02	0.02	0.02

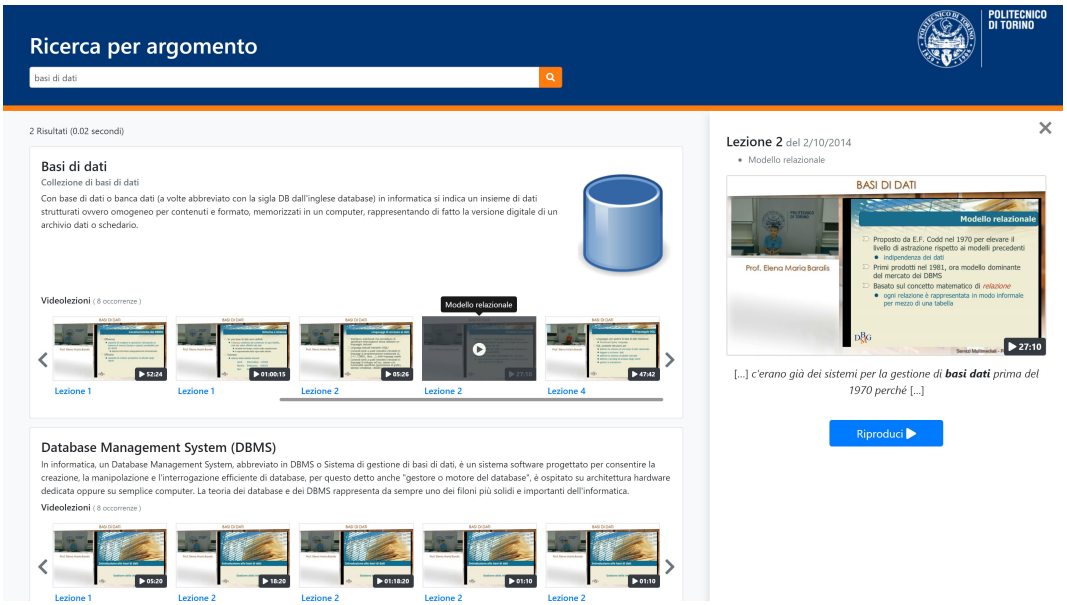
Table 3.6 *VISA* Recommendation performance in a Database course

Strategy	MAP
VISA	0.8112
AGDIST	0.0037
AIDA	0.0034
Babelfy	0.0021
DBpedia Spotlight	0.0061
FOX	0.0001
PBOH	0.1293

In particular, the system returns the following data:

- An information box with the description of all the entities recognized in the query; the description is derived from Wikidata and it illustrates the potential for using knowledge bases to provide at a glance a clear idea of the content of interest.
- The recovery of all the inherent video fragments extracted from the video lessons corpus.
- The ability to click on a fragment and start the video clip showing below it the highlighted words linked to named entities .

(a) Desktop version



(b) Mobile version

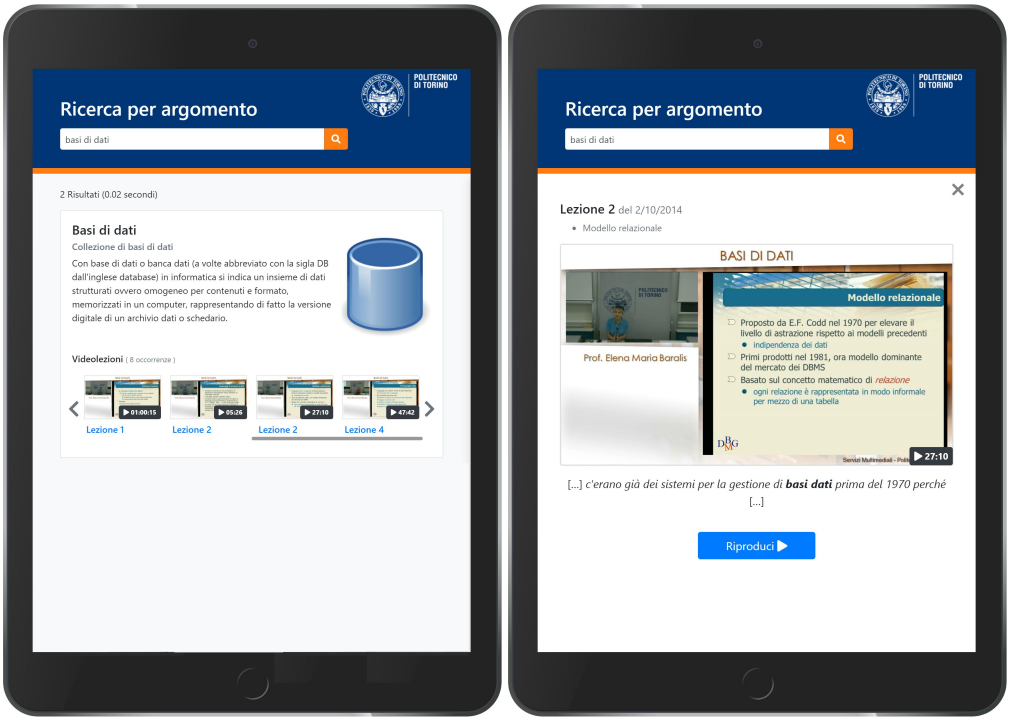


Fig. 3.4 Example of a query through *VISA*'s search engine for the n-gram "basi dati"

3.5 Discussion and guidance for future research

This chapter highlights the main advantages of indexing video lectures, namely:

- Increased student engagement in the use of an e-learning environment, allowing more comfortable browsing of video material.
- Facilitating study by allowing faster retrieval of the content of interest, a feature that is particularly useful for review before the exam in order to improve performance.

A large collection of video lecture indexing approaches from the literature were examined with particular focus on the techniques and the type of semantic annotations: *keywords, topic, keyphrases, tag clouds, and named entities*. Named entities have the inherent advantage of allowing additional information to be retrieved from the knowledge base to enrich the user interface, allowing the learner further clarification of the concept of interest.

With this in mind a new supervised semi-automatic multilingual algorithm, called *VISA*, was presented that indexes video lectures by named entities and provides a search function. *VISA* was compared with state-of-the-art NEL algorithms, achieving better results on a dataset related to a database course in Italian. This is mainly due to the ability of the methodology to consider the domain of interest of the course with minimal teacher effort. The presented work still requires future work to address some major issues:

- *VISA* needs to be validated with other courses and in multiple languages.
- *VISA* needs to be applied in a real-world scenario to assess through a survey the students' satisfaction with using it and to estimate its benefits on the students' exam outcome.

Semantic annotations are a prominent direction for research; for example, other instructional materials may be employed to provide students with automatic alignment of different educational sources. In this regard, the following chapter will focus on a new methodology to address the cross-media linking of educational material that still relies on named entity annotations.

Chapter 4

Cross-media Retrieval for multimodal learning

Technical advances allowed the availability of a large quantity of material of different nature, such as text, images, maps, audio (speech and music) and video. The Computer Science discipline that deals with the process of searching and retrieving multimedia documents is called Multimedia Information Retrieval [67]. The sub-branch of information retrieval designated for scenarios where queries and retrieval results are of different media types is called cross-media retrieval [159]. This chapter discusses the adoption of cross-media retrieval for learning to align educational materials of different types based on semantics. Specifically, the content is organized as follows: Section 4.1 outlines the benefits of using different types of educational materials, Section 4.2 inspects automatic cross-media retrieval procedures from the literature, Section 4.3 presents *TVREM*, a new method to perform video-to-text and text-to-video retrieval designed for educational resources and Section 4.4 draws out the major lessons learned and future directions.

4.1 The learning value of using educational materials of different nature

The rapid growth of the web and multimedia data has impacted education, complementing traditional resources (e.g. books, paper notes...) with materials of different

nature such as slides, ebooks, educational videos, images, discussion forums, chats, social network groups, podcasts, etc.. [33, 36, 54, 103]. The value of unconventional resources is still debated in literature; for example, there are different views related to Youtube videos:

- [203] discusses that although students are consuming more online content such as YouTube videos, their skills and self-esteem in integrating these materials are still far from the utopian vision of independent learning, and they often need authoritative guidance such as that of professors.
- [111] revealed that the use of *discovery learning* [1] and YouTube videos as educational technology tools in primary school science lessons helped students enhance cognitive achievement.
- [98] focus on Youtube tutorials videos as a resource to improve problem-solving skills for academic development.
- [139] details best practices for academic content-creators to succeed in engaging students with Youtube videos, since such casual learners are curious, intrigued by novel ideas, and actively seeking new knowledge, insights and skills.

Therefore, even if there is no unified opinion on Youtube usage, similar to many other digital resources, they certainly allow cross-checking across multiple sources of learning content. However, educational resources from different contexts and/or of different types are in most cases not linked and the learner must often search alone for resources of interest, filtering out those that are more or less suitable.

Although searching is an excellent way to achieve critical and creative learning [178], students with lower versus higher topic knowledge exhibited different patterns of navigation within and across mediums [164]; different learning styles may appeal to different modal preferences [30].

Information search, i.e. seeks relevant information from other information resources, should not be confused with *help-seeking*, i.e. asking for help from

[1] Discovering learning is a technique in which students discover knowledge without guidance, developing their own understanding. The role of instruction is merely to provide a suitable environment, which in software might be a microworld or simulation. Discovery learning involves hypothesis formulation and testing [81, 191].

someone more competent (e.g., the teacher). [167] examines the boundaries between *information search* and *help-seeking* by proposing a framework that integrates them and thus provides for the advice of a facilitator in the selection of content at either the preliminary or final stages. Similarly in [120] authors conclude that students need support in searching on the Web as well as in developing information literacy.

Other research indicates that it is increasingly a duty for teachers to dispense rich educational material from a variety of sources. Students have been shown to learn more deeply from a combination of words and pictures than from words alone [138]. The authors in [35] state: "teachers and lecturers have to deal with a much greater range of information processing styles, cultural backgrounds and styles of learning. As a result, the ideal for teaching in higher education is now recognized to involve much more than lectures as the means of information provision". [186] highlights the value of providing instructional materials in various forms as a facilitator of metacognition and surveyed most of the previous research.

[145, 186, 252] inspected the role of multimodal learning on students' performance. Findings have shown that the use of different instructional resources depends on the student and the context, i.e. *lower-achieving students benefit more than others*.

[47, 186] looked at narrow temporal contexts, such as subjecting students to the use of diverse material during a lecture and subsequently testing them; within such temporal constraints, the risk is that cognitive overload may arise. Conversely, providing diverse instructional material in broader temporal windows can motivate students to question and understand educational content in greater detail, helping them to focus more on the concepts rather than the presenting mode. Nowadays it is challenging to keep track of all the different support materials used by students, for example, the use of the web, or the exchange of information among peers (e.g. via messaging apps like Telegram); hence it might be helpful to provide students with an environment where multiple educational resources are presented and aligned by educational concept or topic in order to track directly through the system the operations performed by users in order to understand their most significant behaviors. In conclusion, the use of multimodal educational resources is a clear advantage for the student's engagement and improving performance; therefore, *the ability to automatically link different educational resources semantically related may be both an advantage for the student to discover new material to improve understanding and for the teacher as a support tool.*

4.2 Cross-media Retrieval of educational resources

A detailed review of cross-media retrieval is presented in [160]. The authors examined more than 100 references distinguishing between *common space learning methods* and *cross-media similarity measurement methods*; the former measure the similarities among items in a common space, and the latter directly compute the cross-media similarities by analyzing the known data relationships without an explicit common space. Both common space learning and cross-media similarity measurement methods are in turn divided into a wide variety of categories (Figure 4.1).

Of particular interest are graph approaches such as *Graph Regularization Methods* [27], a semi-supervised learning technique for labeling a partially labeled graph; their goal is to predict the labels of unlabeled vertices. This category includes JRL [246], JGRHML [247], S 2 UPG [158]; these approaches achieved the highest performances in [160] for *video-to-text* and *text-to-video* retrieval tasks, which are of greatest interest since the method proposed in Section 4.3 has been tested on datasets that mix video and textual resources. Other methods for this task are listed on the following Github page: https://github.com/danieljf24/awesome-video-text-retrieval. They are all *DNN-based methods*.

The starting point is CLIP [171], an image-to-text retrieval approach that instead of recognizing predetermined categories of objects in images, learns directly from raw data about alignment with subtitles. CLIP became the foundation for a large number of subsequent works for the video-to-text and text-to-video retrieval tasks:

- [68] presents CLIP2Video, a network that intends to transfer the learning capability of the image representation introduced by CLIP to conversely retrieve video from text queries.
- CAMoE [50] improves CLIP by introducing Mixture-of-Experts (MoE) to extract multi-perspective video representations (e.g. action, entity, scene) to align them separately with the corresponding part of the text and Dual Softmax Loss (DSL) to force that when a text-to-video or video-to-text pair reaches the optimal match, the symmetric video-to-text or text-to-video is the highest.

- Whereas video-to-text retrieval frameworks use transformers for video and encoders for text, CLIP2TV [76] aims at exploring where the critical elements lie in the transformer.

Other interesting approaches that do not rely on CLIP include the following:

- [64] proposes a dual encoding for video retrieval by text, rather than the single-lever encoder previously applied, secondly, they introduce hybrid space learning which combines the high performance of the latent space and the good interpretability of the concept space.
- [65] considers the domain gap problem between training data and testing data, instead of just the semantic and modality gap, proposing a new model called MAP that exhibits greater generalization capabilities.
- TACo [236] improves contrastive learning using a token-aware contrastive loss which is computed by taking into account the syntactic classes of words, such as nouns and verbs.

The approaches analyzed thus far, particularly *DNN-based methods*, are not extensible across much of the educational domain for the following reasons:

- The datasets used (Table 4.1), although ranging in different domains, are composed of short videos, whose useful content is usually present in the audio channel, and texts in the size of the paragraph (e.g. descriptions, captions, imperative English sentences); on the contrary, in educational video clips, important information can be hidden both in the video (e.g. blackboard, slides or animations) and in the audio (the professor's speech) channels and they can also last for a long time (e.g. a classroom lesson filmed and subsequently uploaded on an E-learning platform); moreover, textual resources can also be long: pages of notes, chapters of textbooks, Powerpoint presentations containing a large number of slides.
- *DNN-based methods* require plenty of time and loads of hardware (e.g., GPUs); the need for resources increases exponentially for very long videos.

Similarly, audio-to-text retrieval approaches, which could be useful for linking the audio channel of educational videos with educational texts, are also tested with datasets in which the audio clips are brief [109, 154].

Table 4.1 Analysis on available datasets for Cross-media Retrieval

Dataset	Retrieval type	Video/audio clips duration	Text content
MSVD [46]	Video-to-text	usually less than 10 seconds long	captions
MSR-VTT [233]		between 10 and 30 seconds	descriptions
TRECVID 2016 [18]		source clips between 10 e 60 seconds, target clips between 10 and 120 seconds	descriptions
VATEX [224]		around 10 seconds	descriptions
LSMDC [180]		2 seconds at most	audio descriptions
ActivityNet [119]		the dataset contains videos as long as 10 minutes	captions
DiDeMo [93]		short clips, not specified	descriptions
HowTo100M [141]		2000 seconds	descriptions
YooCook,YouCook2 [249]		average length of 5.26 minutes	sentences
Cross-task [251]		not specified but small clips	annotation
AudioCaps [105]	Audio to text	10 seconds	audio captions
Clotho [66]		between 15 and 30 seconds	audio captions

In the educational field, some studies inspect the role of Linked Data technologies to improve educational resources [62, 238]. [238] proposes Annomation, a system to manually perform semantic annotations on educational videos, and SugarTube, a platform to browse semantically linked educational video resources. The authors highlight two major benefits of using semantic annotations for cross-media retrieval: (i) collecting related learning resources (ii) the ability to share, reuse and semantically connect the educational resource from different educational institutions.

In [62] the authors survey approaches for Linked Education, i.e. education that exploits educational Web data. They focused in particular on integrating services able to merge data from heterogeneous educational repositories; for example, MERLOT [2] allows searching simultaneously in 20 partner collections and digital libraries. However, the interlinking task is different from cross-media retrieval since the MERLOT interface enables running a text query to retrieve educational materials from different sites and not fetch learning resources from others of a different type. Interlinking is most useful for searching starting without a source resource to form a set of different materials related to the same topic; on the other hand, whether students have not understood a specific educational resource, cross-media retrieval allows them to search from it for material about the same knowledge presented in a different format.

[2] https://www.merlot.org

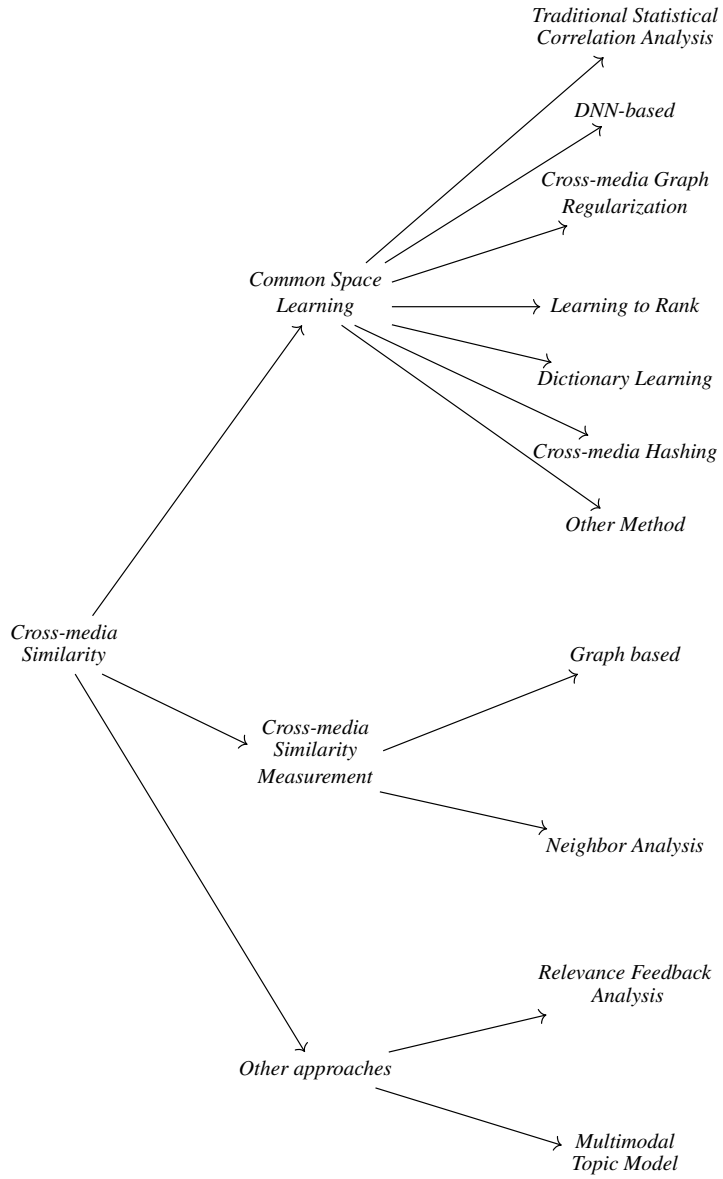


Fig. 4.1 Methodologies classification for Cross-media Retrieval

Although there may be overlap between the two tasks, cross-media retrieval should also not be confused with the augmentation of educational resources. [49] propose augmenting paper-based reading activity with direct access to digital materials and scaffolded questioning using Smartphone; however, in this case, the additional material was prepared manually and was not retrieved automatically. The unique example of cross-media retrieval similar to the approach presented in the

next section is [184]; here the authors described HealthRecSys, a content-based recommender system that links YouTube videos about health to reputable health educational websites from MedlinePlus [3]. The designed pipeline consists of the following steps:

- Video metadata (title,description and subtitles) are used as possible terms.
- The natural processing system cTAKES™ [XVII] is applied to metadata to extract SNOMED-CT health terms from text from SNOMED-CT [XVIII] clinical healthcare ontology.
- Bio-ontology API [XIX] is employed to find synonymous MedlinePlus terms [XX] from the SNOMED-CT terms.

The final output is a Youtube video under which several links to MedlinePlus pages are suggested, allowing the user to reduce the burden when searching for reliable additional content.

Given the use of semantic enrichment for cross-media retrieval in education next section proposes an approach that relies on named entity linking using Wikidata knowledge base.

4.3 *TVREM*: a new method for text-to-video and video-to-text retrieval for educational material

[43] partially presents the content of this section; the major contribution of this work is to propose a methodology that splits the cross-media retrieval task into two subtasks:

1. extraction of named entities (NEs) from original educational resources to represent each file as a set of NEs linked to a knowledge base;
2. calculating a similarity score between sets of entities belonging to different media resources to determine whether they relate to the same content.

[3] https://medlineplus.gov

In [43] the method has been validated with a newly created dataset called BookToYout consisting of ebooks sections related to Computer Science and Youtube videos. The proposed approach performed better than state-of-the-art solutions. In this section the following changes have been made:

- The method was named *TVREM*: text-to-video and video-to-text retrieval for educational material.
- It has been tested with a larger number of configurations to assess the best way to derive the similarity score.
- The set of features used to train machine learning models has been reduced since unnecessary features were removed.
- A new dataset, called EDUCA, was created consisting of lecture notes and educational videos both from MIT OpenCourseWare [XXI];
- Results were compared for both datasets against more baselines and competitors.
- The *TVREM* performance on BookToYout improved and for both datasets BookToYout and EDUCA the algorithm outperformed competitors.

The remainder of this section is broken down as follows: subsection 4.3.1 details the methodology used, subsection 4.3.2 summarizes the experimental settings and subsection 4.3.3 reports the results obtained. Finally subsection 4.3.4 details how to reproduce the experiments.

4.3.1 Methodology

The proposed approach receives as input PDF files containing text and video in MP4 format; it consists of the following steps, which are also summarized in Figure 4.2:

1. *Text extraction*: it focuses on the extraction of textual content from PDFs and of audio from videos in MP4;
2. *Named Entity Linking*: two sets of entities are derived from the video transcript and the PDF text;

3. *Entity sets expansion*: it allows to amplify the original entity sets with parents and separate instances from classes;

4. *Feature engineering*: it refers to the creation of a feature set consisting of Cardinalities and similarity metrics computed between the two entities sets derived from video and PDF;

5. *Similarity computation and ranking*: the final similarity between the two resources is determined via machine learning algorithms.

Each of these steps is discussed in detail in the following section.

3.3.1.1 Text extraction

The automatic transcription from the video is formed by two steps: (i) audio extraction from the video was performed using *MoviePy* Python library [XXII], thereafter (ii) the audio transcript was derived through *Cloud Natural Language APIs* [XXIII].

The plain text from slides was derived by using the *ConvertApi* service [XXIV].

3.3.1.2 Named Entity Linking

Named entities were extracted from the plain text by applying the following extractors:

- *TextRazor*[XXV] extracts entities by linking them with *Wikidata*.
- *Babelfy* [XXVI] [146] returns *DBpedia* entities.
- *Google Cloud Speech API* [XXVII] gives back *Wikipedia* entities.

To standardize the different responses, the entities have been aligned with *Wikidata* using *sameAs* [4] property and subsequently merged. *Wikidata* achieved higher-quality standards compared to alternative solutions [245]. Notice that *Wikidata* content curation relies on a voluntary basis, requires community approval prior to adding new content, and supports data ingestion from external data sources.

[4] http://www.w3.org/2002/07/owl#sameAs

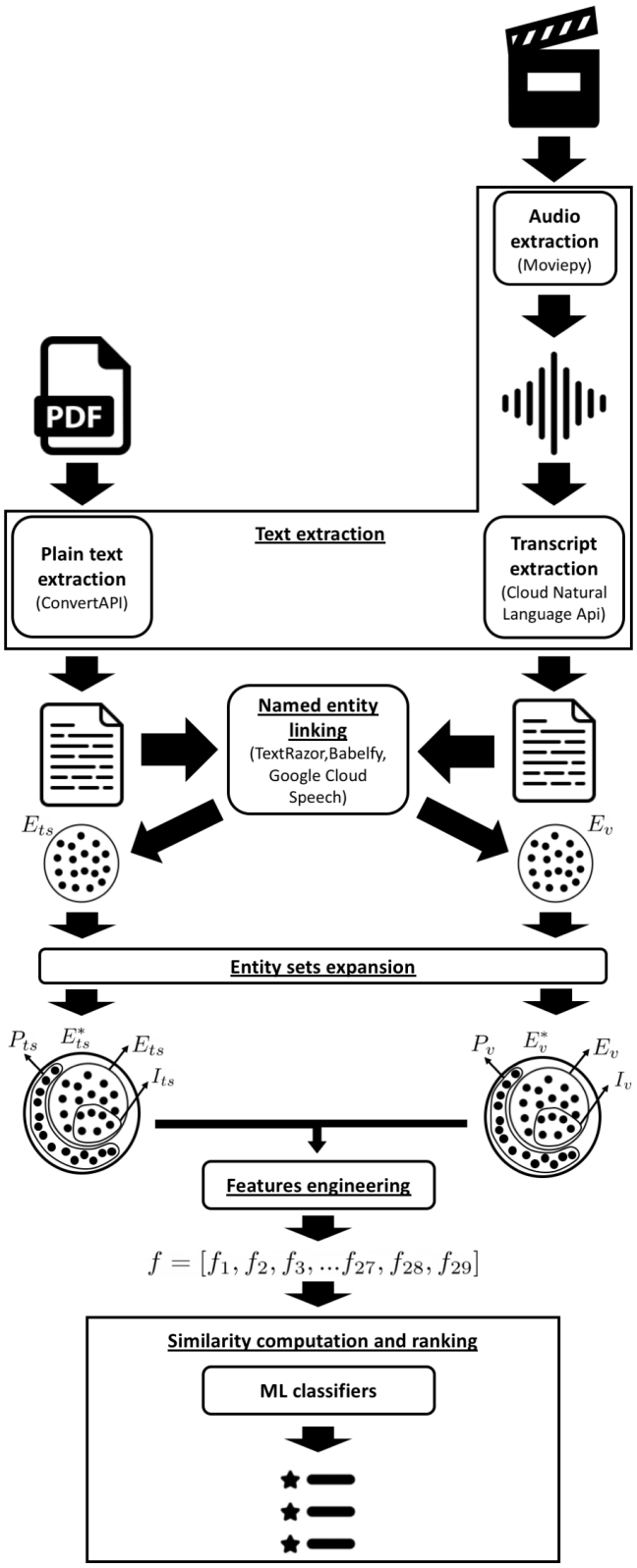


Fig. 4.2 The *TVREM* architecture

TVREM is modular, i. e. each of its blocks can be replaced with a different implementation as long as the output is of the same form. For example, ConvertApi could be replaced by Tesseract for text extraction from PDF files or a different named entity linking algorithm could be applied instead of the current one.

3.3.1.3 Entity sets expansion

Wikidata organizes the large set of available entities into complex hierarchies [198], providing the ability to expand the original set of entities to classes and parents.

Let E_{ts} and E_v be the set of KB entities associated with the PDF text snippet ts and video transcript v.

For both entities sets the instances were distinguished from other entity types by querying the Wikidata Sparql endpoint [5] as follows.

```
ASK {
  <ENTITY URI> wdt:P31 ?o.
}
```

where `ENTITY URI` indicates the URI of the entity involved, whereas `wdt:P31` [6] is the *instance of* predicate in the KB.

Let I_{ts} and I_v be the entity sets consisting of all the instances in E_{ts} and E_v, respectively. In order to semantically enrich the E_{ts} and E_v contextual descriptions, a query to retrieve the corresponding parents has been performed by using the predicates `wdt:P279` (*subclass of*) [7] and `wdt:P361` (*part of*) [8].

```
SELECT ?o {
  <ENTITY URI> wdt:P279|wdt:P361 ?o.
}
```

Let P_{ts} and P_v be the parent entities related to E_{ts} and E_v, respectively. The extended entity sets E^*_{ts} and E^*_v are obtained by the union of the respective child and parent entities P_{ts} and P_v, i.e., $E^*_{ts} = E_{ts} \cup P_{ts}$, $E^*_v = E_v \cup P_v$.

Suppose know that two compared resources a are related to the same content; in the NEL step the entity set E_a is derived from a and the entity set E_b is derived from b. The instance i is one of the entities belonging to E_a, while its class c is belong to

[5] https://query.wikidata.org/
[6] http://www.wikidata.org/prop/direct/P31
[7] http://www.wikidata.org/prop/direct/P279
[8] http://www.wikidata.org/prop/direct/P361

E_b Amplifying E_a and E_b, c will belong to E_a^* and will be part of the intersection between E^*a and E^*b.

Table 4.2 Features characterizing the $\langle ts,v \rangle$ pair in *TVREM*

Index	Feature name	Method of determination	Description
Cardinalities			
1	$card(E_{ts})$	$\lvert E_{ts}\rvert$	cardinality of set E_{ts}
2	$card(E_v)$	$\lvert E_v\rvert$	cardinality of set E_v
3	$card(E_{ts} \cap E_v)$	$\lvert E_v \cap E_{ts}\rvert$	cardinality of the intersection between E_{ts} and E_v
4	$card(E_v \cup E_{ts})$	$\lvert E_v \cup E_{ts}\rvert$	cardinality of the union between E_{ts} and E_v
5	$card(I_{ts})$	$\lvert I_{ts}\rvert$	cardinality of set I_{ts}
6	$card(I_v)$	$\lvert I_v\rvert$	cardinality of set I_v
7	$card(I_{ts} \cap I_v)$	$\lvert I_{ts} \cap I_v\rvert$	cardinality of the intersection between I_{ts} and I_v
8	$card(I_{ts} \cup I_v)$	$\lvert I_{ts} \cup I_v\rvert$	cardinality of the union between I_{ts} and I_v
9	$card(P_{ts} \cap P_v)$	$\lvert P_{ts} \cap P_v\rvert$	cardinality of the intersection between P_{ts} and P_v
10	$card(P_{ts} \cup P_v)$	$\lvert P_{ts} \cup P_v\rvert$	cardinality of the union between P_{ts} and P_v
11	$card(E_{ts}^*)$	$\lvert E_{ts}^*\rvert$	cardinality of set E_{ts}^*
12	$card(E_v^*)$	$\lvert E_v^*\rvert$	cardinality of set E_v^*
13	$card(E_{ts}^* \cap E_v^*)$	$\lvert E_{ts}^* \cap E_v^*\rvert$	cardinality of the intersection between E_{ts}^* and E_v^*
14	$card(E_{ts}^* \cup E_v^*)$	$\lvert E_{ts}^* \cup E_v^*\rvert$	cardinality of the union between E_{ts}^* and E_v^*
Similarities			
15	$\mathcal{N}(E_{ts}, E_v)$	$\frac{\lvert E_ts \cap E_v\rvert}{max(\lvert E_{ts}\rvert,\lvert E_v\rvert)}$	normalized weighted intersection between E_{ts} and E_v
16	$\mathcal{O}(E_{ts}, E_v)$	$\frac{\lvert E_ts \cap E_v\rvert}{min(\lvert E_{ts}\rvert,\lvert E_v\rvert)}$	overlap coefficient between E_{ts} and E_v
17	$\mathcal{J}(E_{ts}, E_v)$	$\frac{\lvert E_{ts} \cap E_v\rvert}{\lvert E_{ts} \cup E_v\rvert}$	Jaccard similarity between E_{ts} and E_v
18	$\mathcal{N}(I_{ts}, I_v)$	$\frac{\lvert I_{ts} \cap I_v\rvert}{max(\lvert I_{ts}\rvert,\lvert I_v\rvert)}$	normalized weighted intersection between I_{ts} and I_v
19	$\mathcal{O}(I_{ts}, I_v)$	$\frac{\lvert I_{ts} \cap I_v\rvert}{min(\lvert I_{ts}\rvert,\lvert I_v\rvert)}$	overlap coefficient between I_{ts} and I_v
20	$\mathcal{J}(I_{ts}, I_v)$	$\frac{\lvert I_{ts} \cap I_v\rvert}{\lvert I_{ts} \cup I_v\rvert}$	Jaccard similarity between I_{ts} and I_v
21	$\mathcal{N}(P_{ts}, P_v)$	$\frac{\lvert P_{ts} \cap P_v\rvert}{max(\lvert P_{ts}\rvert,\lvert P_v\rvert)}$	normalized weighted intersection between P_{ts} and P_v
22	$\mathcal{O}(P_{ts}, P_v)$	$\frac{\lvert P_{ts} \cap P_v\rvert}{min(\lvert P_{ts}\rvert,\lvert P_v\rvert)}$	overlap coefficient between P_{ts} and P_v
23	$\mathcal{J}(P_{ts}, P_v)$	$\frac{\lvert P_{ts} \cap P_v\rvert}{\lvert P_{ts} \cup P_v\rvert}$	Jaccard similarity between P_{ts} and P_v
24	$\mathcal{N}(E_{ts}^*, E_v^*)$	$\frac{\lvert E_{ts}^* \cap E_v^*\rvert}{max(\lvert E_{ts}^*\rvert,\lvert E_v^*\rvert)}$	normalized weighted intersection between E_{ts}^* and E_v^*
25	$\mathcal{O}(E_{ts}^*, E_v^*)$	$\frac{\lvert E_{ts}^* \cap E_v^*\rvert}{min(\lvert E_{ts}^*\rvert,\lvert E_v^*\rvert)}$	overlap coefficient between E_{ts}^* and E_v^*
26	$\mathcal{J}(E_{ts}^*, E_v^*)$	$\frac{\lvert E_{ts}^* \cap E_v^*\rvert}{\lvert E_{ts}^* \cup E_v^*\rvert}$	Jaccard similarity between E_{ts}^* and E_v^*
Target			
class	$sim(ts,v)$		probability value

3.3.1.4 Feature engineering

The features are derived from the previous formed entity sets: P_{ts}, E_{ts}, I_{ts}, E_{ts}^*, P_v, E_v, I_v, E_v^*. They are split into two groups: *Cardinalities* compute intersections and unions between sets; *Similarities* determine likeness between them according to

the similarity measures described in [89] and [133]. The considered feature set is summarized in Table 4.2.

For pair of resources $\langle ts,v \rangle$, a feature vector was derived.

3.3.1.5 Similarity computation and ranking

TVREM determines similarity by testing different machine learning algorithms. In particular, receiving the data divided into train and validation set, it selects the algorithm that, towed with the train split, achieves the best performance on the validation set with respect to *the target score*: (i) Mean Average Precision (MAP) [250], (ii) Precision at k (P@K) and (iii) Recall at k (R@K) [100, 185].

For each algorithm different hyper-parameters are tested choosing the best by performing a grid search on the validation set; the complete list of algorithms and hyper-parameters is provided in Table 4.3.

Since training all these models can be time-consuming, the framework allows to freely remove algorithms or hyper-parameters.

To avoid introducing a bias in the learning phase, feature values were preemptively normalized using a min-max scaler:

$$X_{norm} = 2\left(\frac{X - X_{min}}{X_{max} - X_{min}}\right) - 1$$

Cross-media content retrieval relies on the network outputs (O) produced by taking the queried resource (Q) combined with any candidate resource of a different type. The final ranking consists of the candidates sorted in order of decreasing output probability.

4.3.2 Experimental settings

Datasets

Since no datasets for this task were present in the literature, two datasets were created: BookToYout and EDUCA. Both consist of PDF files and MP4 videos and are both suitable for the video-to-text and text-to-video retrieval tasks.

Table 4.3 Hyper-parameters for grid search in *TVREM*

Hyper-parameter	**Set of possible values**
Neural network [a]	
activation hidden layers	linear, selu, relu, elu, sigmoid
activation output layer	linear,sigmoid
loss function	cosine similarity, mse, binary cross entropy
optimizer	adam
dropout	0.05, 0.10, 0.15, 0.20, 0.25, 0.30
batch size	1,5,10
number of hidden layers	2
units per layer	(15,7),(10,5),(30,15)
K-Neighbors Classifier [b]	
number of neighbors	2, 3, 4, 5, 6, 7, 8, 9
Decision Tree Classifier [b]	
maximum depth	2, 3, 5, 7, 10, None
minimum number of samples split	2, 4, 6
minimum number of samples leaf	1, 3, 5
maximum number of features for split	None, auto, sqrt, log2
Decision Tree Regressor [b]	
maximum depth	2, 3, 5, 7, 10, None
minimum number of samples split	2, 4, 6
minimum number of samples leaf	1, 3, 5
maximum number of features for split	None, auto, sqrt, log2
Random Forest Classifier [b]	
maximum depth	2, 3, 5, 7, 10, None
minimum number of samples split	2, 4, 6
minimum number of samples leaf	1, 3, 5
maximum number of features for split	None, auto, sqrt, log2
number of estimators	10, 50, 100
Random Forest Regressor [b]	
maximum depth	2, 3, 5, 7, 10, None
minimum number of samples split	2, 4, 6
minimum number of samples leaf	1, 3, 5
maximum number of features for split	None, auto, sqrt, log2
number of estimators	10, 50, 100
Epsilon-support vector Regressor (SVR) [b]	
kernel	RBF, linear
regularizer	0.025, 0.05, 0.01, 1, 10, 100, 1000
Default configurations for *Gaussian Naive Bayes Classifier, Gaussian Process Classifier*, *Gaussian Process Regressor, Multilayer Perceptron Classifier*, *Multilayer Perceptron Regressor, Logistic Regression, Linear Regression* [b] and *XGBoost* [c]	

[a] Implemented with the Python library Keras (link:https://keras.io/)
[b] Implemented with the Python library Scikit-learn (link:https://scikit-learn.org/stable/)
[c] Implemented with the Python library Xgboost (link:https://xgboost.readthedocs.io/en/stable/)

Table 4.4 Statistics on new datasets defined for Cross-Media Retrieval in education: BookToYout and EDUCA

Property	**Value**					
	BookToYout			EDUCA		
	Min	**Max**	**Avg**	**Min**	**Max**	**Avg**
Snippets length (word count)	2	34359	2959.6	1	821.68	6326
No. of relevant videos per snippet	1	10	4.7	1	1	
Video length	2min	1h21min	21min	21sec	33min	1h

BookToYout includes academic instructional ebooks and Youtube educational videos. It was formed starting from ebooks covering the following Computer Science topics: *machine learning*, *pattern recognition*, *control system engineering*, *java programming*, *SQL (Structured Query Language)*, *semantic web*, *probability*. A few chapters have been chosen for each book; the complete list is reported in the appendix B. For each chapter title, a query was performed on *Youtube Data API* [5.2] and the top 10 URLs related to English language videos were saved. A panel of experts subsequently reviewed each of the videos indicating whether or not they contained similar content to the original book chapter (0=unbound content, 1=bound content). The final dataset is composed of 92 book chapters and 753 Youtube videos; more detailed information is provided in Table 4.4.

EDUCA consists of 223 video lectures recorded in class and 223 students' notes. Both materials have been extracted from the MIT OpenCourseWare [5.2] website[9] that publishes most MIT course content. In this case, the materials were already aligned on the website and no subsequent annotation effort was required. The detailed list of considered courses is shown in the Table 4.5. These are related to various educational subjects accordingly to the International Standard Classification of Education (ISCED) [XXVIII].

Although both datasets are related to university-level topics, there are some major differences between them:

- In EDUCA the materials are generated by students (the notes) and teachers (the video lectures) while in BookToYout the materials are generated by experts, not necessarily teachers.

[9] https://ocw.mit.edu/about/

- BookToYout textual resources are discursive and explanatory while EDUCA's lecture notes generally outline summaries of what was explained in class.
- BookToYout's video resources are in most cases home-recorded, edited, and intended for a web audience, while EDUCA's video resources are recorded in-class lectures where the professor speaks in front of the students.
- Educational materials of different types and from different contexts are aligned in BookToYout, the educational materials are always of different types but from the same context in EDUCA.
- More than one video resource (5 on average) can be linked for each textual material in BookToYout, while a textual resource always corresponds to just one video resource in EDUCA.

These distinctions allow validating *TVREM* for slightly different contexts.

Implementation settings

The experiments were conducted strictly following the pipeline shown in Figure 4.2 and the details explained in Section 4.3.1. A single change was accomplished for EDUCA where text extraction was carried out using *Tesseract* rather than *ConvertAPI* since ebooks are subject to copyright restrictions and cannot be uploaded to third-party services.

Both datasets are highly unbalanced in that the number of pairs that do not match is much higher than those that match in the training data: 1 out of 128 for EDUCA, and 1 out of 131 for BookToYout. To address this problem the training dataset is randomly resampled. Three different techniques were tested:

- *Oversampling*: it extracts multiple copies from the minority class in the training dataset.
- *Undersampling*: it randomly deletes samples from the majority class in the training dataset.
- *Dynamic undersampling*: same as *undersampling* but the samples chosen for the majority class vary at each training epoch; this technique has only been validated with neural networks.

Table 4.5 Summary of the MIT courses employed in EDUCA

Course name	ISCED level	Broad fields of education in ISCED-F 2013	Number of lectures
Innovation Systems for Science, Technology, Energy, Manufacturing, and Health [a]	6	Social sciences, journalism and information	7
The Film Experience [b]	6	Humanities and Arts	20
Principles of Chemical Science	6	Natural sciences, mathematics and statistics (Physical science - Chemistry) [c]	35
Introduction to algorithm [d]	6	Information and Communication Technologies	24
Machine Learning for Healthcare [e]	7	Information and Communication Technologies	25
Foundations of Computational and Systems Biology [f]	6-7	Information and Communication Technologies	20
Blockchain and Money	7	Social sciences, journalism and information (Economics) [g]	22
Energy Decisions Markets and Policies [h]	7	Social sciences, journalism and information (Economics)	22
Probabilistic Systems Analysis and Applied Probability [i]	7	Natural sciences, mathematics and statistics (Mathematics and statistics)	25
String Theory and Holographic Duality [j]	7	Natural sciences, mathematics and statistics (Physical science - Physics)	24

[a] https://rb.gy/4vxqkx
[b] https://rb.gy/n1l18e
[c] https://rb.gy/d0d4eo
[d] https://rb.gy/itluom
[e] https://rb.gy/8fvkcf
[f] https://rb.gy/1mfthe
[g] https://rb.gy/z6aewy
[h] https://rb.gy/d9dwtp
[i] https://rb.gy/5odrpp
[j] https://rb.gy/np1sg8

Baselines and competitors

The method presented has been evaluated against some baselines and competitors.

The baselines use video transcripts and text extracted from PDFs as input documents:

- *TFIDF*: the documents are represented as *TFIDF* (term frequency-inverse document frequency) features [123] and the document similarity is evaluated using *cosine similarity* [173]; this approach is widely adopted in literature [188, 246]. The features were computed using the TFIDF Scikit-learn Python implementation [10] and the following hyper-parameters were optimized via grid search:
  - *ngram_range*: it determines lower and upper boundary of the range of n-values for different n-grams to be extracted; the following values were tested: $(1,1),(2,2),(3,3),(1,3),(1,2),(2,3)$.
  - *max_df*: when building the vocabulary ignore terms that have a document frequency strictly higher than the given threshold; the following values were tested: $0.7, 0.8, 0.9, 1.0$.
  - *min_df*: when building the vocabulary ignore terms that have a document frequency strictly lower than the given threshold; the following values were tested: $1, 3, 5, 7, 10$.
- BERT transformers: based on [176] different models for document embeddings have been tested and the final ranking was estimated by *cosine similarity*; the complete list of tested sentence transformers is provided here: *distiluse-base-multilingual-cased* , *paraphrase-MiniLM-L6-v2*, *bert-large-nli-stsb-mean-tokens* , *distilbert-base-nli-stsb-mean-tokens*, *bert-base-nli-stsb-mean-tokens*, *bert-base-nli-mean-tokens*, *roberta-base-nli-stsb-mean-tokens*, *xlm-r-large-en-ko-nli-ststb*, *bert-large-nli-mean-tokens*, *xlm-r-base-en-ko-nli-ststb*, *roberta-large-nli-stsb-mean-tokens*, *roberta-large-nli-mean-tokens*, *distilbert-base-nli-mean-tokens*, *roberta-base-nli-mean-tokens*. The transformers names are derived from the HugginFace implementation [11].

[10] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

[11] https://huggingface.co/sentence-transformers

The video-to-text and text-to-video algorithms used as competitors are *JRL* [246], *JGRHML* [247] and *S 2 UPG* [158] . The implementations described in the [158, 246, 247] for XMedia dataset were used for all three algorithms.

In addition to baselines and competitors, similarities were determined computing the final ranking using the following features individually: $\mathcal{O}(E_{ts}, E_v)$, $\mathcal{J}(E_{ts}, E_v)$, $card(E_v \cap E_{ts})$, $\mathcal{N}(E_{ts}, E_v)$, $\mathcal{J}(I_{ts}, I_v)$, $\mathcal{O}(E_{ts}, E_v)$, $card(E_v \cap E_{ts})$, $\mathcal{J}(E_{ts}, E_v)$, $\mathcal{N}(I_{ts}, I_v)$, $\mathcal{O}(I_{ts}, I_v)$, $\mathcal{O}(P_{ts}, P_v)$, $card(P_{ts} \cap P_v)$, $\mathcal{J}(P_{ts}, P_v)$,$card(I_{ts} \cap I_v)$, $\mathcal{N}(E_{ts}, E_v)$, $\mathcal{N}(P_{ts}, P_v)$. This was beneficial to figuring out the impact of machine learning algorithms to combine similarities and cardinalities.

Metrics

(i) Precision at k, (ii) Recall at k and (iii) Mean Average Precision (MAP) were adopted for evaluation.

For EDUCA dataset and in the video-to-text retrieval task for BookToYout dataset $P@1$ and $R@5$ have been calculated since for each query there is only one correct resource and consequently $P@1 = R@1$ determines how many times this resource is in the first position while $R@5$ determines if it is in the first top-5 returned resources. For the text-to-video task for BookToYout $P@5$ and $R@10$ were computed because the correct resources are on average five for each query.

4.3.3 Results

Tables 4.6,4.7,4.9,4.8 show the results of the video-to-text and text-to-video retrieval tasks on EDUCA and BookToYout datasets for the train, validation and test splits. Specifically, each table shows the scores of the top 5 machine learning algorithms in *TVREM* ,both for models estimated using the entire feature set, only *Similarities* and only *Cardinalities*) and performances achieved by features taken individually, baselines, and competitors. For each of these 4 categories, the algorithms are ordered by a weighted average of the 3 scores on the test set: $W = 0.4 \cdot MAP + 0.4 \cdot P@k + 0.2 \cdot R@k$. The best models for each <dataset,task> and their scores are highlighted in bold and underlined. In addition, for each of them, the contribution of each feature was computed by determining mean absolute Shapley Additive exPlanations(SHAP) [130] values (Figures 4.3).

The following conclusions can be derived:

- *TVREM achieved the best scores in both video-to-text and text-to-video retrieval for both datasets*. The scores are significantly high for EDUCA; for BookToYout video-to-text retrieval task instead the *Precison*@1 is lower than EDUCA while similar scores were reached for the other metrics. The lowest scores were obtained for BookToYout text-to-video retrieval task; in this case, more video resources are associated with a query and therefore only one of them not present in the first ranks will cause the recall decrease, while the lower precision is because there are more not matching video resources semantically close to the matching ones than in EDUCA.

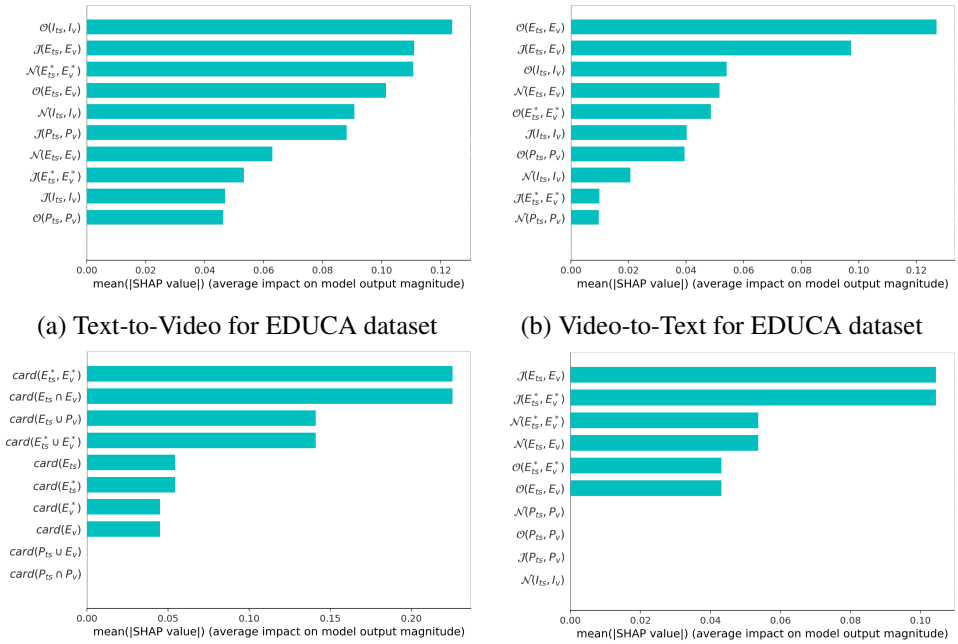
- *Among the various machine learning algorithms adopted there is not one that has outperformed the others for all tasks*, on the contrary, *they have often achieved similar results* demonstrating that by combining features with different strategies and even using simpler approaches good performance can be reached.

- *Both Similarities and Cardinalities features were beneficial*; in fact although in most cases all features or *Similarities* alone led to the highest results, the model that performed best when performance was lowest on average (i.e. for BookToYout in text-to-video retrieval task) adopted only the *Cardinalities* features.

- *Changing the data sampling mode (oversampling or undersampling) for training didn't affect the performance.*

- *Some features considered individually achieved slightly lower results but in the same order of magnitude as the machine learning models.* Generally, these are also the ones that have the highest SHAP values in the machine learning models. This suggests that the key to the proposed methodology is the adoption of entity sets to represent resources and that the algorithms to compute similarity only serve to slightly improve the final rankings.

- *Baselines underperformed machine learning models and individual features*; among them, transformers outperformed TFIDF approaches: the model that averaged the best scores (3 out of 4 cases) was *distiluse-base-multilingual-cased*.



(a) Text-to-Video for EDUCA dataset

(b) Video-to-Text for EDUCA dataset

(c) Text-to-Video for BookToYout dataset

(d) Video-to-Text for BookToYout dataset

Fig. 4.3 Feature relevance analysis for models computed with *TVREM*

- *Competitors achieved the worst scores* probably because they are designed to deal with videos that contain meaningful content in frames rather than audio, while in the given datasets most of the relevant information is in the audio track.

4.3.4 Reproducibility

The results achieved with the proposed method are fully reproducible by attending the instructions to the following Github repository: https://github.com/Loricanal/TVREM. It contains (i) the complete code of *TVREM* implemented in Python language (ii) a link to download the EDUCA dataset including videos and lecture notes and (iii) the intermediate files for both BookToYout and EDUCA datasets. For BookToYout, for privacy reasons, only the sets of entities extracted from the videos and texts have been released, therefore the experiments can be reproduced from the *Entity sets expansion* step.

Table 4.6 Text-to-Video Retrieval scores for EDUCA dataset

Algorithm	***Features***	**MAP**			**Precision@1**			**Recall@5**		
		Train	**Val**	**Test**	**Train**	**Val**	**Test**	**Train**	**Val**	**Test**
TVREM										
Multilayer Perceptron Regressor	***Cardinalities***, ***Similarities***	95.66	95.69	**100.0**	93.02	95.56	**100.0**	100.0	95.56	**100.0**
	Similarities	93.89	95.67	97.45	89.15	95.56	95.92	100.0	95.56	100.0
	Cardinalities	91.64	95.68	96.94	87.6	95.56	93.88	99.22	95.56	100.0
Gaussian Process Classifier	*Cardinalities*, *Similarities*	94.91	95.79	98.98	91.47	95.56	97.96	100.0	95.56	100.0
	Similarities	94.64	95.7	98.37	90.7	95.56	97.96	100.0	95.56	100.0
	Cardinalities	93.49	95.9	96.84	89.15	95.56	95.92	100.0	95.56	100.0
Linear Regression	*Cardinalities*, *Similarities*	95.74	95.72	98.98	92.25	95.56	97.96	100.0	95.56	100.0
	Similarities	95.03	95.67	98.64	90.7	95.56	97.96	100.0	95.56	100.0
	Cardinalities	93.06	95.67	97.14	89.15	95.56	95.92	97.67	95.56	97.96
Neural network	*Cardinalities*, *Similarities*	96.15	96.02	98.64	93.02	95.56	97.96	100.0	95.56	100.0
	Similarities	95.09	95.71	97.62	92.25	95.56	95.92	98.45	95.56	100.0
	Cardinalities	93.23	95.85	92.6	89.15	95.56	87.76	98.45	95.56	97.96
Multilayer Perceptron Classifier	*Cardinalities*, *Similarities*	95.76	95.81	96.94	92.25	95.56	93.88	100.0	95.56	100.0
	Similarities	95.25	95.67	98.64	92.25	95.56	97.96	99.22	95.56	100.0
	Cardinalities	87.37	95.67	95.92	81.4	95.56	91.84	95.35	95.56	100.0
Individual features										
$O(E_{ts}, E_v)$		92.27	93.46	98.37	88.37	91.11	97.96	97.67	95.56	100.0
$\mathcal{J}(E_{ts}, E_v)$		95.27	95.67	97.96	91.47	95.56	95.92	100.0	95.56	100.0
$card(E_v \cap E_{ts})$		92.27	93.44	96.44	88.37	91.11	95.92	97.67	95.56	97.96
$\mathcal{N}(E_{ts}, E_v)$		89.04	91.22	95.58	82.17	86.67	91.84	97.67	95.56	100.0
$\mathcal{J}(I_{ts}, I_v)$		90.17	93.45	93.1	83.72	91.11	89.8	97.67	95.56	97.96
$O(E_{ts}, E_v)$		85.16	92.07	93.03	77.52	91.11	87.76	96.12	93.33	100.0
$card(E_v \cap E_{ts})$		85.18	92.04	92.05	77.52	91.11	87.76	96.9	93.33	97.96
$\mathcal{J}(E_{ts}, E_v)$		90.4	94.56	92.52	85.27	93.33	85.71	96.9	95.56	100.0
$\mathcal{N}(I_{ts}, I_v)$		84.33	91.22	90.48	75.97	86.67	85.71	95.35	95.56	95.92
$O(I_{ts}, I_v)$		88.7	93.92	90.14	82.17	93.33	81.63	97.67	95.56	100.0
$O(P_{ts}, P_v)$		74.01	81.87	89.59	63.57	73.33	81.63	89.15	91.11	100.0
$card(P_{ts} \cap P_v)$		74.01	81.85	88.61	63.57	73.33	81.63	89.15	91.11	97.96
$\mathcal{J}(P_{ts}, P_v)$		85.19	89.74	88.78	79.07	84.44	79.59	94.57	95.56	100.0
$card(I_{ts} \cap I_v)$		88.7	92.78	88.35	82.17	91.11	79.59	97.67	95.56	97.96
$\mathcal{N}(E_{ts}, E_v)$		75.13	86.96	86.9	63.57	80.0	77.55	89.15	95.56	100.0
$\mathcal{N}(P_{ts}, P_v)$		65.97	78.52	79.83	53.49	68.89	69.39	79.07	95.56	100.0
Baselines										
distiluse-base-multilingual-cased		37.88	48.35	44.46	24.81	35.56	28.57	52.71	64.44	57.14
paraphrase-MiniLM-L6-v2		36.62	43.13	42.66	23.26	33.33	30.61	47.29	48.89	53.06
bert-large-nli-stsb-mean-tokens		29.41	25.44	35.2	19.38	13.33	24.49	39.53	31.11	44.9
distilbert-base-nli-stsb-mean-tokens		24.51	30.94	32.46	14.73	22.22	22.45	30.23	40.0	42.86
bert-base-nli-stsb-mean-tokens		33.49	30.87	32.27	20.93	22.22	20.41	45.74	35.56	38.78
bert-base-nli-mean-tokens		25.19	24.75	28.87	16.28	13.33	18.37	33.33	35.56	34.69
roberta-base-nli-stsb-mean-tokens		18.25	32.63	29.45	10.85	20.0	14.29	25.58	40.0	44.9
xlm-r-large-en-ko-nli-ststb		21.69	26.11	27.2	12.4	15.56	14.29	30.23	28.89	40.82
bert-large-nli-mean-tokens		23.61	23.69	25.79	13.95	15.56	14.29	27.91	28.89	32.65
xlm-r-base-en-ko-nli-ststb		24.83	28.58	24.48	15.5	20.0	14.29	31.78	28.89	28.57
roberta-large-nli-stsb-mean-tokens		19.82	30.5	23.08	12.4	17.78	10.2	23.26	42.22	38.78
roberta-large-nli-mean-tokens		17.87	25.51	22.5	10.85	17.78	12.24	23.26	28.89	26.53
distilbert-base-nli-mean-tokens		16.6	24.9	21.47	10.08	13.33	10.2	19.38	31.11	32.65
roberta-base-nli-mean-tokens		10.37	20.92	14.15	5.43	13.33	4.08	12.4	20.0	18.37
TFIDF		3.34	12.12	13.5	0.0	6.67	2.04	2.33	11.11	18.37
Competitors										
JRL		10.12	12.42	11.67	4.59	4.61	7.82	11.11	13.21	8.67
JGRHML		12.31	11.09	10.20	11.10	9.88	13.41	18.91	21.33	23.9
S 2 UPG		13.76	15.49	12.44	14.62	12.69	14.95	20.0	17.86	19.0

Table 4.7 Video-to-Text Retrieval scores for EDUCA dataset

Algorithm	Features	MAP			Precision@1			Recall@5		
		Train	**Val**	**Test**	**Train**	**Val**	**Test**	**Train**	**Val**	**Test**
TVREM										
Logistic Regression	*Cardinalities*, *Similarities*	93.57	93.91	**96.6**	89.15	93.33	**93.88**	98.45	93.33	**100.0**
	Similarities	93.57	94.19	96.6	89.15	93.33	93.88	98.45	95.56	100.0
	Cardinalities	91.76	93.64	90.48	88.37	93.33	81.63	96.9	93.33	100.0
Multilayer Perceptron Classifier	*Cardinalities*, *Similarities*	93.05	93.96	96.6	88.37	93.33	93.88	97.67	95.56	100.0
	Similarities	93.83	94.7	96.6	89.92	93.33	93.88	98.45	95.56	100.0
	Cardinalities	90.55	91.43	89.63	86.05	88.89	81.63	96.12	93.33	100.0
Epsilon-support vector Regressor	*Cardinalities*, *Similarities*	93.44	94.68	96.6	89.15	93.33	93.88	98.45	95.56	100.0
	Similarities	93.56	95.78	96.6	89.92	95.56	93.88	97.67	95.56	100.0
	Cardinalities	91.14	93.61	87.76	87.6	93.33	77.55	96.12	93.33	100.0
Neural network	*Cardinalities*, *Similarities*	93.92	94.7	95.58	89.92	93.33	91.84	97.67	95.56	100.0
	Similarities	92.77	95.78	96.6	88.37	95.56	93.88	97.67	95.56	100.0
	Cardinalities	89.4	94.56	91.75	83.72	93.33	85.71	95.35	95.56	97.96
Multilayer Perceptron Regressor	*Cardinalities*, *Similarities*	91.91	94.46	92.52	86.05	93.33	85.71	98.45	95.56	100.0
	Similarities	93.35	94.28	95.24	89.15	93.33	91.84	98.45	95.56	100.0
	Cardinalities	90.73	94.57	95.24	86.05	93.33	91.84	96.12	97.78	100.0
Individual features										
$O(E_{ts}, E_v)$		89.44	89.68	95.92	83.72	84.44	91.84	96.9	95.56	100.0
$O(E_{ts}, E_v)$P		79.45	72.61	91.56	70.54	60.0	85.71	91.47	88.89	100.0
$O(I_{ts}, I_v)$		81.37	82.51	88.44	72.09	73.33	79.59	95.35	95.56	100.0
$\mathcal{J}(I_{ts}, I_v)$		83.21	88.85	83.94	75.19	84.44	73.47	92.25	93.33	97.96
$O(P_{ts}, P_v)$		71.84	68.79	82.5	58.91	57.78	73.47	86.82	86.67	97.96
$\mathcal{J}(E_{ts}, E_v)$		87.67	90.05	83.48	82.17	86.67	71.43	94.57	93.33	97.96
$card(E_v \cap E_{ts})$		55.85	65.1	76.99	44.19	55.56	65.31	65.12	77.78	91.84
$\mathcal{N}(E_{ts}, E_v)$		71.47	75.11	77.03	60.47	68.89	65.31	82.17	84.44	91.84
$\mathcal{J}(E_{ts}, E_v)$P		74.27	82.34	68.72	67.44	77.78	57.14	82.95	86.67	85.71
$card(I_{ts} \cap I_v)$		50.71	60.29	69.61	37.21	48.89	53.06	61.24	71.11	91.84
$\mathcal{N}(I_{ts}, I_v)$		66.67	72.92	69.61	55.04	64.44	53.06	79.07	86.67	91.84
$\mathcal{J}(P_{ts}, P_v)$		61.14	73.82	57.13	51.16	68.89	42.86	73.64	77.78	67.35
$\mathcal{N}(E_{ts}, E_v)$P		39.51	52.13	38.77	27.91	44.44	14.29	48.84	55.56	65.31
$card(E_v \cap E_{ts})$P		15.9	41.66	38.77	1.55	31.11	14.29	19.38	53.33	65.31
$card(P_{ts} \cap P_v)$		11.76	35.64	33.11	0.78	24.44	12.24	10.85	44.44	59.18
$\mathcal{N}(P_{ts}, P_v)$		25.45	45.67	31.92	13.95	37.78	10.2	35.66	48.89	59.18
Baselines										
distiluse-base-multilingual-cased		27.64	46.02	41.45	17.83	31.11	26.53	37.98	57.78	61.22
paraphrase-MiniLM-L6-v2		26.65	44.09	39.83	16.28	33.33	24.49	36.43	57.78	57.14
bert-base-nli-stsb-mean-tokens		9.28	26.6	24.53	3.1	17.78	16.33	10.08	31.11	26.53
bert-large-nli-mean-tokens		20.38	30.16	23.63	12.4	20.0	14.29	25.58	42.22	32.65
xlm-r-large-en-ko-nli-ststb		14.01	26.04	22.37	6.98	13.33	10.2	16.28	33.33	34.69
roberta-large-nli-stsb-mean-tokens		13.14	29.64	23.26	4.65	20.0	14.29	19.38	37.78	26.53
distilbert-base-nli-stsb-mean-tokens		12.78	28.67	20.06	3.88	17.78	10.2	20.16	40.0	26.53
bert-large-nli-stsb-mean-tokens		12.39	20.33	18.08	6.2	8.89	10.2	13.95	28.89	20.41
bert-base-nli-mean-tokens		14.98	24.74	14.93	6.98	13.33	4.08	20.93	33.33	26.53
xlm-r-base-en-ko-nli-ststb		9.91	20.51	15.9	3.88	8.89	4.08	10.85	31.11	24.49
distilbert-base-nli-mean-tokens		14.16	22.68	15.77	5.43	6.67	2.04	20.16	37.78	30.61
roberta-large-nli-mean-tokens		8.19	19.46	13.69	3.1	8.89	4.08	9.3	28.89	18.37
roberta-base-nli-mean-tokens		12.61	22.5	12.56	5.43	11.11	2.04	15.5	28.89	20.41
roberta-base-nli-stsb-mean-tokens		6.06	20.18	11.27	2.33	11.11	2.04	5.43	24.44	16.33
TFIDF		4.49	13.32	9.23	0.78	8.89	2.04	5.43	8.89	10.2
Competitors										
JRL		11.35	10.91	11.58	9.11	8.74	9.92	9.32	14.43	12.12
JGRHML		12.21	12.04	11.39	12.32	9.97	12.27	17.93	19.42	22.1
S 2 UPG		13.01	14.62	11.21	13.43	11.97	13.53	19.08	17.00	19.21

Table 4.8 Text-to-Video Retrieval scores for BookToYout dataset

Algorithm	Features	MAP			Precision@5			Recall@5		
		Train	**Val**	**Test**	**Train**	**Val**	**Test**	**Train**	**Val**	**Test**
TVREM										
Gaussian Process Classifier	*Cardinalities, Similarities*	80.85	73.57	71.39	51.39	30.0	35.71	61.35	72.22	60.55
	Similarities	58.2	79.25	59.99	22.22	45.0	28.57	28.57	71.58	37.86
	Cardinalities	81.03	75.54	**86.97**	52.78	30.0	**64.29**	58.46	68.38	63.06
Epsilon-support vector Regressor	*Cardinalities, Similarities*	82.53	75.89	81.54	50.0	35.0	64.29	62.45	66.45	65.41
	Similarities	80.85	78.56	76.14	55.56	45.0	57.14	61.19	69.66	61.47
	Cardinalities	79.73	75.14	77.49	48.61	40.0	50.0	58.37	67.74	63.71
Decision Tree Regressor	*Cardinalities, Similarities*	80.81	74.74	79.6	56.94	45.0	64.29	57.01	56.84	63.56
	Similarities	79.66	75.24	82.68	45.83	45.0	64.29	60.04	60.68	62.52
	Cardinalities	73.93	71.11	79.12	50.0	30.0	57.14	47.44	70.3	65.29
Gaussian Process Regressor	*Cardinalities, Similarities*	78.85	73.9	80.4	48.61	25.0	64.29	56.41	65.81	61.59
	Similarities	81.31	73.62	79.65	59.72	40.0	50.0	57.07	62.61	65.29
	Cardinalities	81.0	48.37	80.4	58.33	10.0	57.14	56.81	17.74	65.29
Neural network	*Cardinalities, Similarities*	78.38	82.12	78.85	50.0	50.0	64.29	59.52	72.22	64.25
	Similarities	79.69	78.85	72.32	48.61	50.0	57.14	59.56	70.3	50.36
	Cardinalities	83.59	80.95	76.8	59.72	55.0	42.86	61.64	64.53	62.52
Individual features										
$card(E_v \cap E_{ts})$P		80.06	75.25	82.23	48.61	25.0	**71.43**	61.59	71.58	57.77
$card(E_v \cap E_{ts})$		80.06	75.25	82.23	48.61	25.0	71.43	61.59	71.58	57.77
$O(E_{ts}, E_v)$		79.92	76.04	84.79	47.22	30.0	64.29	60.43	65.17	56.58
$O(E_{ts}, E_v)$P		79.92	76.04	84.79	47.22	30.0	64.29	60.43	65.17	56.58
$\mathcal{N}(E_{ts}, E_v)$		81.94	74.55	77.26	56.94	20.0	50.0	60.28	75.43	63.18
$\mathcal{N}(E_{ts}, E_v)$P		81.94	74.55	77.26	56.94	20.0	50.0	60.28	75.43	63.18
$\mathcal{J}(E_{ts}, E_v)$P		80.62	72.71	75.91	51.39	15.0	50.0	60.99	67.74	60.4
$\mathcal{J}(E_{ts}, E_v)$		80.62	72.71	75.91	51.39	15.0	50.0	60.99	67.74	60.4
$O(I_{ts}, I_v)$		24.99	38.42	22.87	1.39	0.0	7.14	0.77	3.85	4.82
$\mathcal{N}(I_{ts}, I_v)$		24.99	38.42	22.87	1.39	0.0	7.14	0.77	3.85	4.82
$\mathcal{J}(P_{ts}, P_v)$		24.99	38.42	22.87	1.39	0.0	7.14	0.77	3.85	4.82
$\mathcal{N}(P_{ts}, P_v)$		24.99	38.42	22.87	1.39	0.0	7.14	0.77	3.85	4.82
$O(P_{ts}, P_v)$		24.99	38.42	22.87	1.39	0.0	7.14	0.77	3.85	4.82
$\mathcal{J}(I_{ts}, I_v)$		24.99	38.42	22.87	1.39	0.0	7.14	0.77	3.85	4.82
$card(I_{ts} \cap I_v)$		24.99	38.42	22.87	1.39	0.0	7.14	0.77	3.85	4.82
$card(P_{ts} \cap P_v)$		24.99	38.42	22.87	1.39	0.0	7.14	0.77	3.85	4.82
Baselines										
distiluse-base-multilingual-cased		48.37	66.2	49.32	33.33	35.0	42.86	27.49	50.43	36.88
paraphrase-MiniLM-L6-v2		49.67	66.31	50.82	31.94	45.0	35.71	30.88	40.6	28.75
bert-base-nli-stsb-mean-tokens		36.54	55.24	40.6	16.67	30.0	35.71	13.31	38.46	22.55
bert-large-nli-mean-tokens		32.44	46.88	40.19	15.28	10.0	28.57	10.98	20.51	22.85
xlm-r-large-en-ko-nli-ststb		32.77	49.34	38.8	15.28	15.0	28.57	9.36	24.57	21.11
roberta-large-nli-stsb-mean-tokens		35.45	58.67	39.86	20.83	30.0	28.57	11.87	45.3	19.2
distilbert-base-nli-stsb-mean-tokens		35.5	54.34	39.01	13.89	25.0	21.43	12.96	22.44	23.15
bert-large-nli-stsb-mean-tokens		34.42	51.39	39.47	18.06	30.0	21.43	11.72	27.35	20.3
bert-base-nli-mean-tokens		32.01	46.32	37.9	12.5	15.0	21.43	9.1	12.61	19.42
xlm-r-base-en-ko-nli-ststb		30.96	50.1	36.58	11.11	20.0	21.43	8.14	28.42	18.5
distilbert-base-nli-mean-tokens		32.06	48.13	36.98	13.89	20.0	14.29	8.72	22.22	26.76
roberta-large-nli-mean-tokens		31.66	49.64	36.54	13.89	10.0	14.29	8.8	29.27	25.72
roberta-base-nli-mean-tokens		30.04	42.34	29.19	16.67	5.0	21.43	6.8	6.62	14.17
roberta-base-nli-stsb-mean-tokens		34.29	48.87	33.7	19.44	15.0	14.29	11.76	17.31	16.04
TFIDF		26.28	43.21	27.33	1.39	10.0	0.0	2.12	9.62	10.56
Competitors										
JRL		14.2	24.6	19.1	5.1	4.06	7.11	8.31	6.86	6.88
JGRHML		33.12	38.0	32.1	9.55	8.0	8.47	10.52	10.22	12.54
S 2 UPG		21.31	17.7	31.04	8.81	7.62	9.03	11.2	12.45	11.92

Table 4.9 Video-to-Text Retrieval scores for BookToYout dataset

Algorithm	Features	MAP			Precision@1			Recall@5		
		Train	**Val**	**Test**	**Train**	**Val**	**Test**	**Train**	**Val**	**Test**
TVREM										
Epsilon-support vector Regressor	*Cardinalities, Similarities*	99.21	100.0	99.09	51.69	29.81	61.86	98.87	100.0	97.95
	Similarities	99.55	100.0	**100.0**	51.69	29.81	**61.86**	100.0	100.0	**100.0**
	Cardinalities	99.15	100.0	99.36	51.69	29.81	61.86	98.87	100.0	100.0
Gaussian Process Regressor	*Cardinalities, Similarities*	99.58	100.0	100.0	51.52	29.81	61.86	100.0	100.0	100.0
	Similarities	99.66	100.0	100.0	51.69	29.81	61.86	100.0	100.0	100.0
	Cardinalities	99.66	84.71	100.0	51.69	11.8	61.86	100.0	54.17	100.0
Random Forest Classifier	*Cardinalities, Similarities*	98.99	100.0	100.0	50.34	29.81	61.86	100.0	100.0	100.0
	Similarities	99.66	100.0	100.0	51.69	29.81	61.86	100.0	100.0	100.0
	Cardinalities	99.66	100.0	100.0	51.69	29.81	61.86	100.0	100.0	100.0
Random Forest Regressor	*Cardinalities, Similarities*	99.66	100.0	100.0	51.69	29.81	61.86	100.0	100.0	100.0
	Similarities	99.34	100.0	99.36	51.69	29.81	61.86	100.0	100.0	100.0
	Cardinalities	99.66	100.0	100.0	51.69	29.81	61.86	100.0	100.0	100.0
Gaussian Process Classifier	*Cardinalities, Similarities*	99.34	100.0	99.36	51.69	29.81	61.86	100.0	100.0	100.0
	Similarities	99.66	100.0	100.0	51.69	29.81	61.86	99.52	100.0	100.0
	Cardinalities	99.58	100	99.58	51.69	29.81	61.86	99.52	100.0	100.0
Individual features										
$O(E_{ts}, E_v)$		99.25	100.0	99.36	51.69	29.81	61.86	99.35	100.0	100.0
$card(E_v \cap E_{ts})$P		99.18	100.0	99.24	51.69	29.81	61.86	98.87	100.0	100.0
$card(E_v \cap E_{ts})$		99.18	100.0	99.24	51.69	29.81	61.86	98.87	100.0	100.0
$\mathcal{J}(E_{ts}, E_v)$P		99.54	100.0	99.36	51.69	29.81	61.86	100.0	100.0	100.0
$O(E_{ts}, E_v)$P		99.25	100.0	99.36	51.69	29.81	61.86	99.35	100.0	100.0
$\mathcal{J}(E_{ts}, E_v)$		99.54	100.0	99.36	51.69	29.81	61.86	100.0	100.0	100.0
$\mathcal{N}(E_{ts}, E_v)$		99.49	100.0	99.24	51.69	29.81	61.86	99.52	100.0	100.0
$\mathcal{N}(E_{ts}, E_v)$P		99.49	100.0	99.24	51.69	29.81	61.86	99.52	100.0	100.0
$O(I_{ts}, I_v)$		51.02	76.02	57.18	0.51	2.48	6.78	6.61	22.92	53.42
$\mathcal{N}(I_{ts}, I_v)$		51.02	76.02	57.18	0.51	2.48	6.78	6.61	22.92	53.42
$\mathcal{J}(P_{ts}, P_v)$		51.02	76.02	57.18	0.51	2.48	6.78	6.61	22.92	53.42
$\mathcal{N}(P_{ts}, P_v)$		51.02	76.02	57.18	0.51	2.48	6.78	6.61	22.92	53.42
$O(P_{ts}, P_v)$		51.02	76.02	57.18	0.51	2.48	6.78	6.61	22.92	53.42
$\mathcal{J}(I_{ts}, I_v)$		51.02	76.02	57.18	0.51	2.48	6.78	6.61	22.92	53.42
$card(I_{ts} \cap I_v)$		51.02	76.02	57.18	0.51	2.48	6.78	6.61	22.92	53.42
$card(P_{ts} \cap P_v)$		51.02	76.02	57.18	0.51	2.48	6.78	6.61	22.92	53.42
Baselines										
paraphrase-MiniLM-L6-v2		74.67	92.41	75.88	21.45	19.25	27.12	62.42	89.58	86.99
distiluse-base-multilingual-cased		71.86	92.0	71.01	17.91	18.63	22.03	59.19	87.5	77.4
distilbert-base-nli-stsb-mean-tokens		60.87	83.22	62.03	7.6	8.07	12.71	34.52	58.33	61.64
roberta-large-nli-stsb-mean-tokens		61.43	86.22	63.54	8.78	10.56	15.25	33.87	77.08	54.11
bert-large-nli-stsb-mean-tokens		60.5	83.39	62.76	7.09	8.07	13.56	33.06	62.5	59.59
roberta-base-nli-stsb-mean-tokens		61.76	85.64	62.61	9.29	9.94	16.1	33.55	77.08	50.0
distilbert-base-nli-mean-tokens		59.38	80.41	62.22	6.93	4.35	14.41	28.55	54.17	52.05
bert-base-nli-stsb-mean-tokens		63.06	84.34	61.37	9.46	8.7	11.02	39.19	62.5	53.42
bert-base-nli-mean-tokens		58.92	81.89	60.07	5.74	6.83	11.86	28.71	56.25	54.79
xlm-r-large-en-ko-nli-ststb		58.7	81.69	60.3	6.76	6.83	11.02	25.65	52.08	54.79
xlm-r-base-en-ko-nli-ststb		59.43	82.28	58.85	6.93	6.83	11.02	30.0	58.33	54.79
TFIDF		53.21	80.85	61.44	1.69	6.83	11.86	13.23	47.92	49.32
bert-large-nli-mean-tokens		56.39	81.42	58.8	3.55	6.21	9.32	21.94	54.17	57.53
roberta-large-nli-mean-tokens		57.72	84.08	60.57	5.57	7.45	12.71	26.13	72.92	49.32
roberta-base-nli-mean-tokens		57.54	79.2	57.42	6.42	2.48	10.17	22.58	56.25	38.36
Competitors										
JRL		12.47	11.82	11.00	5.13	4.81	8.28	19.21	41.48	39.11
JGRHML		27.67	24.12	29.00	8.44	6.37	11.02	33.47	41.89	50.74
S 2 UPG		31.12	33.02	31.04	7.11	6.03	9.32	36.17	42.42	49.0

4.4 Discussion and guidance for future research

This chapter discussed the usefulness of cross-media retrieval for learners to improve engagement and mastery of subject matter; this is useful for both improving performance in college courses and enhancing problem-solving skills.

Greater focus was placed on video-to-text and text-to-video retrieval by considering as a case study the retrieval of unconventional educational resources (e.g. Youtube videos) from traditional educational resources (e.g. slides of course material or textbooks). For this purpose, a new method called *TVREM* was proposed. It outperformed current approaches on two educational datasets.

Some future improvements can be proposed:

- To refine the current ground truth, it might be useful to move from a binary annotation to a multi-level annotation both to differentiate more precisely the similarity of the resources and to distinguish the value of the returned resource, e.g. the same content explained in a different way, additional content to satisfy curiosity and enrich knowledge, etc.

- To reach more general findings, it may be useful to create other datasets and test the transfer learning potential of models trained with one dataset on others.

- To study the flexibility of the algorithm to adapt to different ground truths for the same dataset since they depend on the annotator and the final ranking of the resources could vary according to the educational level, learning interests or training mode preferences. For example in an academic course, the professor could determine the right ranking of the educational resources according to the competencies she/he wants the students to acquire; she/he could start from a textbook or from the self-produced educational material to search for similar resources in a video collection, look at the automatic ranking obtained with a pre-computed *TVREM* model and manually reorder the resources in case of error, modifying the ground truth and fine-tuning the model. A different professor might prefer different resources and therefore annotate the dataset differently, resulting in a different final model.

  Further, the same learner could establish the resources of greater interest and refine the tuning of the model because she/he is driven by interests of personal learning (e.g. to realize an own idea of which it does not possess the

competencies) or because she/he prefers educational material of a determined type. Taking into account the case study of retrieving videos from textbooks and Youtube as a collection, different users could prefer different channels that produce videos with different styles (e.g. animations or more traditional videos): starting from the model fine-tuned by the professor they could see the ranking returned and reorder it by searching for video resources with the same semantics but with a more preferential look.

Chapter 5

Conclusions

5.1 Summary of Contributions

Since the early 1990s, there has been an increasing diversification of learning modes; in particular, the digital age resulted in the increasing use of electronic materials in education up to fully online courses such as MOOCs. This led to greater facility in recording student data and the emergence of Learning Analytics (LA) in 2011.

This dissertation focused on the application of artificial intelligence to support LA and education with two main objectives: (i) early predict students' exams outcome figuring out why they succeed or fail in order to early take action (e.g. warning at-risk profiles) (ii) offer richer educational materials to students by indexing video lectures, to facilitate their review, and aligning multimodal resources, to foster their comprehension.

With respect to the first objective, the following contributions were achieved:

- A methodology that relies on *Lazy Associative Classifier* L^3 [22] for the prediction of students' academic performance has been presented; this associative algorithm enables the derivation of human-readable rules to explain the reasons for classification. Its prediction capabilities have been validated in 1st-year Bachelor's Degrees courses in Engineering; the results demonstrate that associative models are as accurate as the other best non-explainable classifiers. In addition, from the inspection of the rules, several profiles of students were derived; these enable personalized intervention.

- *VESPE*, *Visual Explainable Student performance PrEdictor*, has been introduced; it allows deriving explainable models to early predict student outcomes by combining the most popular machine learning algorithms with a state-of-the-art explainability AI method, namely *SHapley Additive exPlanations (SHAP)* [130]. The method achieves high performance in an Object-Oriented programming course using as features the laboratory activity recorded through Version Control System. Based on the model explanation targeted interventions were proposed for each outcome category.

- Since there are no benchmarks for sharing data and naming features, *UNIFORM*, an open relational database integrating various learning data sources, was presented. It allows for automatically extending the integrated dataset as soon as new data sources become available through a machine learning classifier that detects attribute alignments based on the correlations among the corresponding textual attribute descriptions. The integration phase has reached a promising quality level on most of the analyzed benchmark datasets.

With respect to the second objective, the following contributions were achieved:

- *VISA*, a supervised approach to indexing video lectures with semantic annotations, was presented. The method automatically extracts n-grams from the video transcripts and compares them with Wikidata entity labels, deriving a list of matching candidates. These are filtered considering not only text similarity measures but also the semantic pertinence of the candidate to the main subject of the video lectures. The performance of the proposed system was validated on a ground truth against the techniques available in the general entity annotation system GERBIL. The preliminary results demonstrate the effectiveness of the proposed approach since it outperformed competitors for both named entity disambiguation and search function tasks.

- A new method for text-to-video and video-to-text retrieval for educational material named *TVREM* has been introduced. Its key feature is the representation of each resource, independent of whether it is textual or visual, as a set of entities to determine the similarity between two resources with sets of similarity metrics (e.g. Jaccard similarity). *TVREM* has been validated on two datasets achieving very high scores and outperforming baselines and competitors.

5.2 Future Works

Three macro topics were discussed in this thesis, namely student outcome prediction, video-lecture indexing and cross-media retrieval of educational resources. In future work, these could be integrated into a single learning context. Starting from the video-lecture indexing system created for the databases course using *VISA* in Chapter 3.4, *TVREM* could be used to automatically retrieve other educational resources, such as:

- The slides and handouts of the course prepared by the professor.
- Youtube videos concerning the same knowledge.
- Discussions about course exercises (e.g. SQL queries) in Telegram chats; this technology is increasingly used in our university, and students created group chats where they exchange doubts related to course exercises. Extracting discussions in the chats, anonymizing and linking them to the points in the video lectures where the same knowledge is covered would provide students with the ability to access such resources directly through the learning management system.

The learning management system would then embed both video-lecture indexing, search functions, and a wide range of different educational materials. Students' interactions with the system could be tracked to derive features as input for *VESPE*. The SHAP explanations will help to clarify if and how the system usage affects the student performance.

Notes

I. YouTube Data API integrates several facilities for developers, including Automatic caption generation and Search (link: https://developers.google.com/youtube/)

II. Tesseract is a technology for Optical Character Recognition (link: https://github.com/tesseract-ocr/tesseract)

III. JOCR is a program for Optical Character Recognition (link: http://jocr.sourceforge.net/)

IV. MODI is a software for Optical Character Recognition (link: https://www.moditrace.net/en/add-ons/ocr-clear-character-recognition/)

V. TAGME is a powerful tool that is able to identify on-the-fly meaningful short-phrases (link: https://tagme.d4science.org/tagme/)

VI. Autometa is an automated binning pipeline for extraction of microbial genomes from individual shotgun metagenomes (link: https://github.com/celsowm/AutoMeta).

VII. CSO-Classifier is an unsupervised approach for automatically classifying research papers according to the Computer Science Ontology (link:https://github.com/angelosalatino/cso-classifier)

VIII. NCBO Annotator is an ontology-based web service for annotation of textual biomedical data with biomedical ontology concepts (link: https://github.com/ncbo/ncbo_annotator)

IX. Ontotext provides a complete set of semantic technologies transforming how organizations identify meaning across diverse databases and massive amounts of unstructured data. (link:https://www.ontotext.com/)

X. DBpedia is a project aiming to extract structured content from the information created in the Wikipedia project (link: https://www.dbpedia.org/)

XI. Computer Network Ontology is an ontology for the categorization of computer networks domain. https://bioportal.bioontology.org/ontologies/CN

XII. Computer Science Ontology is a large-scale ontology of research areas based on 16 million publications, mainly in the field of Computer Science. https://cso.kmi.open.ac.uk/home

XIII. Natural Language ToolKit is a Python library that provides support to work with human language data (link:https://www.nltk.org/)

XIV. Elasticsearch is a search engine based on the Lucene library. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents. (link:https://www.elastic.co/elasticsearch/)

XV. The DBpedia ontology is a formal definition of categories, properties, and relations between the concepts, data, and entities forming DBpedia knowledge base. (link: http://mappings.dbpedia.org/server/ontology/classes/).

XVI. The Wikimedia REST API offers access to Wikimedia's content and metadata in machine-readable formats, in this thesis, it was mainly used to link Wikidata entities with Wikipedia pages and vice versa (link:https://www.mediawiki.org/wiki/Wikimedia_REST_API)

XVII. Apache cTAKES™ is a natural language processing system for extraction of information from electronic medical record clinical free-text (link: websitehttp://ctakes.apache.org/)

XVIII. SNOMED-CT is a multilingual clinical healthcare ontology (link:http://www.ihtsdo.org/snomed-ct)

XIX. Bio-ontology API is comprised of a set of biomedical resources that are connected together via links. (link: http://data.bioontology.org/documentation)

XX. MedlinePlus is an online information service produced by the United States National Library of Medicine. The service provides curated consumer health information MedlinePlus provides encyclopedic information on health and drug issues and provides a directory of medical services (link:https://en.wikipedia.org/wiki/MedlinePlus)

XXI. MIT OpenCourseWare is a web-based publication of virtually all MIT course content. (link:https://ocw.mit.edu/

XXII. MoviePy is a Python library for video editing (link:https://pypi.org/project/moviepy/)

XXIII. The Cloud Natural Language API provides developers with natural language understanding technologies, including sentiment analysis, entity sentiment analysis, content classification, and syntax analysis (link:https://cloud.google.com/natural-language/docs/reference/rest/).

XXIV. ConvertApi is an API that enables converting files to many different formats (link:https://www.convertapi.com/pdf-to-txt)

XXV. TextRazor combine state-of-the-art natural language processing techniques with a comprehensive knowledge base of real-life facts to help rapidly extract named entities from documents, tweets, or web pages (link:https://www.textrazor.com/).

XXVI. Babelfy is a unified graph-based approach to multilingual Entity Linking and Word Sense Disambiguation (link:http://babelfy.org/)

XXVII. Google Cloud Speech API provides natural language functionalities including named entity linking (link:https://cloud.google.com/speech-to-text/)

XXVIII. The International Standard Classification of Education provides a comprehensive framework for organising education programmes and qualification by applying uniform and internationally agreed definitions to facilitate comparisons of education systems across countries.The is available at this (link : http://uis.unesco.org/en/topic/international-standard-classification-education-isced)

References

[1] Mohammed Abuteir and Alaa El-Halees. Mining educational data to improve students' performance: A case study. *International Journal of Information and Communication Technology Research*, 2:140–146, 01 2012. URL https://www.acit2k.org/ACIT2006/Proceeding/131.pdf.

[2] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international Conference on Management of data*, page 207–216. Association for Computing Machinery, 1993. doi: 10.1145/170035.170072.

[3] Fadhilah Ahmad, Nur Hafieza Ismail, and Azwa Abdul Aziz. The prediction of students' academic performance using classification data mining techniques. *Applied mathematical sciences*, 9:6415–6426, 2015. doi: 10.12988/ams.2015.53289.

[4] Carmen Aina, Eliana Baici, Giorgia Casalone, and Francesco Pastore. The determinants of university dropout: A review of the socio-economic literature. *Socio-Economic Planning Sciences*, 79:101102, 2022. doi: 10.1016/j.seps.2021.101102.

[5] Mashael Al-Barrak and Muna Al-Razgan. Predicting students final gpa using decision trees: A case study. *International Journal of Information and Education Technology*, 6:528–533, 2016. doi: 10.7763/IJIET.2016.V6.745.

[6] Feras Al-Obeidat, Abdallah Tubaishat, Anna Dillon, and Babar Shah. Analyzing students' performance using multi-criteria classification. *Cluster Computing*, 21:623–632, 2018. doi: 10.1007/s10586-017-0967-4.

[7] Qasem Al-Radaideh, Emad Al-Shawakfa, and Mustafa Al-Najjar. Mining student data using decision trees. In *Proceedings of the 2006 International Arab Conference on Information Technology*, pages 126–129. IEEE Computer Society, 2006. doi: 10.1109/PDGC.2014.7030728.

[8] Raghad Al-Shabandar, Abir Hussain, Andy Laws, Robert Keight, and Janet Lunn. Towards the differentiation of initial and final retention in massive open online courses. In *Intelligent Computing Theories and Application*, pages 26–36. Springer, 2017. doi: 10.1007/978-3-319-63309-1_3.

[9] Raghad Al-Shabandar, Abir Hussain, Andy Laws, Robert Keight, Janet Lunn, and Naeem Radi. Machine learning approaches to predict learning outcomes in massive open online courses. In *Proceedings of the 2017 International Joint Conference on Neural Networks*, pages 713–720, 2017. doi: 10.1109/IJCNN.2017.7965922.

[10] Abdulaziz Albahr, Dunren Che, and Marwan Albahar. Semkeyphrase: An unsupervised approach to keyphrase extraction from mooc video lectures. In *Proceedings of the 2019 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 303–307. Association for Computing Machinery, 2019. doi: 10.1145/3350546.3352535.

[11] Balqis Albreiki, Nazar Zaki, and Hany Alashwal. A systematic literature review of student' performance prediction using machine learning techniques. *Education Sciences*, 11(9), 2021. doi: 10.3390/educsci11090552.

[12] Gloria Allione and Rebecca Stein. Mass attrition: An analysis of drop out from principles of microeconomics mooc. *The Journal of Economic Education*, 47: 174–186, 2016. doi: 10.1080/00220485.2016.1146096.

[13] Bussaba Amnueypornsakul, Suma Bhat, and Phakpoom Chinprutthiwong. Predicting attrition along the way: The UIUC model. In *Proceedings of the 2014 Empirical Methods In Natural Language Processing Workshop on Analysis of Large Scale Social Interaction in MOOCs*, pages 55–59. Association for Computational Linguistics, 2014. doi: 10.3115/v1/W14-4110.

[14] Elaf Abu Amrieh, Thair Hamtini, and Ibrahim Aljarah. Mining Educational Data to Predict Student's academic Performance using Ensemble Methods. *International Journal of Database Theory and Application*, 9(8):119–136, 2016. doi: 10.14257/ijdta.2016.9.8.13.

[15] D. Magdalene Delighta Angeline. Association rule generation for student performance analysis using apriori algorithm. In *Proceedings of the 2013 International Conference on Computer Supported Education*, pages 639–646, 2013. doi: 10.9756/SIJCSEA/V1I1/01010252.

[16] P. M. Ashok Kumar, Rami Reddy Ambati, and L. Arun Raj. An efficient scene content-based indexing and retrieval on video lectures. In *Intelligent System Design*, pages 521–534. Springer, 2021. doi: 10.1007/978-981-15-5400-1_53.

[17] Raheela Asif, Agathe Merceron, and Mahmood Pathan. Predicting student academic performance at degree level: A case study. *International Journal of Intelligent Systems and Applications*, 7:49–61, 2014. doi: 10.5815/ijisa.2015.01.05.

[18] George Awad, Jonathan Fiscus, David Joy, Martial Michel, Alan F Smeaton, Wessel Kraaij, Maria Eskevich, Robin Aly, Roeland Ordelman, Gareth J. F Jones, Benoit Huet, and Martha Larson. Trecvid 2016: Evaluating video search, video event detection, localization, and hyperlinking. In *Proceedings*

of the 2016 International Workshop on Video Retrieval Evaluation. National Institute of Standards and Technology, 2016. URL http://hdl.handle.net/2066/163213.

[19] Azwa Abdul Aziz and Norashikin Ahmad. First semester computer science students ' academic performances analysis by using data mining classification algorithms. In *Proceedings of the 2014 International Conference on Intelligence and Computer Science*. IEEE Computer Society, 2014. URL http://eprints.unisza.edu.my/id/eprint/471.

[20] Arun Balagopalan, Lalitha Lakshmi Balasubramanian, Vidhya Balasubramanian, Nithin Chandrasekharan, and Aswin Damodar. Automatic keyphrase extraction and segmentation of video lectures. In *Proceedings of the 2012 IEEE International Conference on Technology Enhanced Education*, pages 1–10. IEEE Computer Society, 2012. doi: 10.1109/ICTEE.2012.6208622.

[21] Girish Balakrishnan. Predicting student retention in massive open online courses using hidden markov models. Master's thesis, EECS Department, University of California, Berkeley, 2013. URL http://www2.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-109.html.

[22] Elena Baralis, Silvia Chiusano, and Paolo Garza. A lazy approach to associative classification. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):156–171, 2008. doi: 10.1109/TKDE.2007.190677.

[23] Lecia Barker, Christopher Lynnly Hovey, Jaspal Subhlok, and Tayfun Tuna. Student perceptions of indexed, searchable videos of faculty lectures. In *Proceedings of the 2014 IEEE Frontiers in Education Conference*, pages 1–8. IEEE Computer Society, 2014. doi: 10.1109/FIE.2014.7044189.

[24] Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. Variations of the similarity function of textrank for automated summarization. *CoRR*, 2016. URL http://arxiv.org/abs/1602.03606.

[25] Subhasree Basu, Yi Yu, Vivek K. Singh, and Roger Zimmermann. Videopedia: Lecture video recommendation for educational blogs using topic modeling. In Qi Tian, Nicu Sebe, Guo-Jun Qi, Benoit Huet, Richang Hong, and Xueliang Liu, editors, *Proceedings of the 2016 MultiMedia Modeling Conference*, pages 238–250. Springer, 2016. doi: 10.1007/978-3-319-27671-7_20.

[26] Lewis Baumstark and Michael Orsega. Quantifying introductory cs students' iterative software process by mining version control system repositories. *J. Comput. Sci. Coll.*, 31(6):97–104, 2016. doi: 10.5555/2904446.2904470.

[27] Mikhail Belkin, Irina Matveeva, and Partha Niyogi. Regularization and semi-supervised learning on large graphs. In John Shawe-Taylor and Yoram Singer, editors, *Proceedings of the 2004 Annual Conference on Learning Theory*, pages 624–638. Springer, 2004. doi: 10.1007/978-3-540-27819-1_43.

[28] Johannes Berens, Kerstin Schneider, Simon Gortz, Simon Oster, and Julian Burghoff. Early detection of students at risk - predicting student dropouts using administrative student data from german universities and machine learning methods. *Journal of Educational Data Mining*, 11(3):1–41, 2019. doi: 10.5281/zenodo.3594771.

[29] Adéle Bezuidenhout. Implications for academic workload of the changing role of distance educators. *Distance Education*, 36(2):246–262, 2015. doi: 10.1080/01587919.2015.1055055.

[30] Dawn Birch. Factors influencing academics' development of interactive multimodal technology-mediated distance higher education courses, 2008. URL https://eprints.qut.edu.au/16698/1/Dawn_Birch_Thesis.pdf.

[31] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly, 2009. URL https://www.oreilly.com/library/view/natural-language-processing/9780596803346/.

[32] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly, 2009. URL http://www.nltk.org/book.

[33] Stephanie J. Blackmon. Outcomes of chat and discussion board use in online learning: A research synthesis. *Journal of Educators Online*, 9, 2012. doi: 10.9743/JEO.2012.2.4.

[34] Sebastien Boyer and Kalyan Veeramachaneni. Transfer learning for predictive models in massive open online courses. In *Artificial Intelligence in Education*, pages 54–63. Springer, 2015. doi: 10.1007/978-3-319-19773-9_6.

[35] Peter Bradwell. The edgeless university: why higher education must embrace technology, 2010. URL https://www.demos.co.uk/files/Edgeless_University_-_web.pdf.

[36] Edna Bravo, Beatriz García, Pep Simo, Mihaela Enache, and Vicenc Fernandez. Video as a new teaching tool to increase student motivation. In *Proceedings of the 2011 IEEE Global Engineering Education Conference*, pages 638 – 642. IEEE Computer Sociey, 05 2011. doi: 10.1109/EDUCON.2011.5773205.

[37] Christopher G. Brinton and Mung Chiang. Mooc performance prediction via clickstream data and social learning networks. In *Proceedings of the 2015 IEEE Conference on Computer Communications*, pages 2299–2307. IEEE Computer Society, 2015. doi: 10.1109/INFOCOM.2015.7218617.

[38] Luiz Antonio Buschetto Macarini, Cristian Cechinel, Matheus Francisco Batista Machado, Vinicius Faria Culmant Ramos, and Roberto Munoz. Predicting students success in blended learning—evaluating different interactions inside learning management systems. *Applied Sciences*, 9(24), 2019. doi: 10.3390/app9245523.

[39] Luca Cagliero, Lorenzo Canale, and Laura Farinetti. Visa: A supervised approach to indexing video lectures with semantic annotations. In *Proceedings of the 2019 IEEE Annual Computer Software and Applications Conference*, pages 226–235. IEEE Computer Society, 2019. doi: 10.1109/COMPSAC.2019.00041.

[40] Luca Cagliero, Lorenzo Canale, and Laura Farinetti. Uniform: Automatic alignment of open learning datasets. In *Proceedings of the 2020 IEEE Annual Computers, Software, and Applications Conference*, pages 95–102. IEEE Computer Society, 2020. doi: 10.1109/COMPSAC48688.2020.00022.

[41] Luca Cagliero, Lorenzo Canale, Laura Farinetti, Elena Baralis, and Enrico Venuto. Predicting student academic performance by means of associative classification. *Applied Sciences*, 11(4), 2021. doi: 10.3390/app11041420.

[42] Lorenzo Canale, Pasquale Lisena, and Raphaël Troncy. A novel ensemble method for named entity recognition and disambiguation based on neural network. In *Proceedings of the 2018 International Semantic Web Conference*, pages 91–107. Springer, 2018. doi: 10.1007/978-3-030-00671-6_6.

[43] Lorenzo Canale, Laura Farinetti, and Luca Cagliero. From teaching books to educational videos and vice versa: a cross-media content retrieval experience. In *Proceedings of the 2021 IEEE Annual Computers, Software, and Applications Conference*, pages 115–120. IEEE Computer Society, 2021. doi: 10.1109/COMPSAC51774.2021.00027.

[44] Wilson Chango, Rebeca Cerezo, and Cristóbal Romero. Predicting academic performance of university students from multi-sources data in blended learning. In *Proceedings of the 2019 Second International Conference on Data Science, E-Learning and Information Systems*. Association for Computing Machinery, 2019. doi: 10.1145/3368691.3368694.

[45] Devendra Chaplot, Eunhee Rhim, and Jihie Kim. Predicting student attrition in moocs using sentiment analysis and neural networks. In *AIED workshop*, 2015. URL http://ceur-ws.org/Vol-1432/islg_pap2.pdf.

[46] David Chen and William Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 2011 Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 190–200. Association for Computational Linguistics, 2011. URL https://aclanthology.org/P11-1020.

[47] Gongxiang Chen and Xiaolan Fu. Effects of multimodal information on learning performance and judgment of learning. *Journal of Educational Computing Research*, 29(3):349–362, 2003. doi: 10.2190/J54F-B24D-THN7-H9PH.

[48] Lijia Chen, Pingping Chen, and Zhijian Lin. Artificial intelligence in education: A review. *IEEE Access*, 8:75264–75278, 2020. doi: 10.1109/ACCESS.2020.2988510.

[49] Nian-Shing Chen, Daniel Chia-En Teng, Cheng-Han Lee, and Kinshuk. Augmenting paper-based reading activity with direct access to digital materials and scaffolded questioning. *Computers & Education*, 57(2):1705–1715, 2011. doi: 10.1016/j.compedu.2011.03.013.

[50] Xing Cheng, Hezheng Lin, Xiangyu Wu, Fan Yang, and Dong Shen. Improving video-text retrieval by multi-stream corpus alignment and dual softmax loss. *CoRR*, 2021. URL https://arxiv.org/abs/2109.04290.

[51] Kwok Tai Chui, Dennis Chun Lok Fung, Miltiadis D. Lytras, and Tin Miu Lam. Predicting at-risk university students in a virtual learning environment via a machine learning algorithm. *Computers in Human Behavior*, 107:105584, 2020. doi: 10.1016/j.chb.2018.06.032.

[52] Jae Young Chung and Sunbok Lee. Dropout early warning systems for high school students using machine learning. *Children and Youth Services Review*, 96:346–353, 2019. doi: 10.1016/j.childyouth.2018.11.030.

[53] Rianne Conijn, Chris Snijders, Ad Kleingeld, and Uwe Matzat. Predicting student performance from lms data: A comparison of 17 blended courses using moodle lms. *IEEE Transactions on Learning Technologies*, 10(1):17–29, 2017. doi: 10.1109/TLT.2016.2616312.

[54] Nada Dabbagh and Anastasia Kitsantas. Personal learning environments, social media, and self-regulated learning: A natural formula for connecting formal and informal learning. *The Internet and Higher Education*, 15(1):3–8, 2012. doi: 10.1016/j.iheduc.2011.06.002.

[55] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 2013 International Conference on Semantic Systems*, page 121–124. Springer, 2013. doi: 10.1145/2506182.2506198.

[56] Fisnik Dalipi, Ali Shariq Imran, and Zenun Kastrati. Mooc dropout prediction using machine learning techniques: Review and research challenges. In *Proceedings of the 2018 IEEE Global Engineering Education Conference*, pages 1007–1014, 2018. doi: 10.1109/EDUCON.2018.8363340.

[57] Mihai Dascalu, Elvira Popescu, Alex Becheru, and Stefan Trausan-Matu. Predicting academic performance based on students' blog and microblog posts. In *Proceedings of the 2016 European Conference on Technology Enhanced Learning*, pages 370–376. Springer, 2016. doi: 10.1007/978-3-319-45153-4_29.

[58] Distinguished Professor Dr. Ali Daud, Naif Aljohani, Rabeeh Abbasi, Miltiadis Lytras, Farhat Abbas, and Jalal Alowibdi. Predicting student performance using advanced learning analytics. In *Proceedings of the 2007 International World Wide Web Conferences Steering Committee*, page 415–421. International World Wide Web Conferences Steering Committee, 2017. doi: 10.1145/3041021.3054164.

[59] Gerben Dekker, Mykola Pechenizkiy, and Jan Vleeshouwers. Predicting students drop out: A case study. In *Computers, Environment and Urban Systems*, pages 41–50. Elsevier, 2009. doi: 10.1109/IBDAP50342.2020.9245457.

[60] Nick Deschacht and Katie Goeman. The effect of blended learning on course persistence and performance of adult learners: A difference-in-differences analysis. *Computers & Education*, 87:83–89, 2015. doi: 10.1016/j.compedu.2015.03.020.

[61] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, 2018. URL https://arxiv.org/abs/1810.04805.

[62] Stefan Dietze, Salvador Sanchez-Alonso, Hannes Ebner, Hong Qing Yu, D. Giordano, Ivana Marenzi, and Bernardo Pereira Nunes. Interlinking educational resources and the web of data – a survey of challenges and approaches. *Program Electronic Library and Information Systems*, 47, 02 2013. doi: 10.1108/00330331211296312.

[63] Mucong Ding, Kai Yang, Dit-Yan Yeung, and Ting-Chuen Pong. Effective feature learning with unsupervised learning for improving the predictive models in massive open online courses. *CoRR*, 2018. URL http://arxiv.org/abs/1812.05044.

[64] Jianfeng Dong, Xirong Li, Chaoxi Xu, Gang Yang, and Xun Wang. Hybrid space learning for language-based video retrieval. *CoRR*, 2020. URL https://arxiv.org/abs/2009.05381.

[65] Jianfeng Dong, Zhongzi Long, Xiaofeng Mao, Changting Lin, Yuan He, and Shouling Ji. Multi-level alignment network for domain adaptive cross-modal retrieval. *Neurocomputing*, 440:207–219, 2021. doi: 10.1016/j.neucom.2021.01.114.

[66] Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. Clotho: An audio captioning dataset. *CoRR*, 2019. URL http://arxiv.org/abs/1910.09387.

[67] Horst Eidenberger. *Fundamental media understanding: the common methods of audio retrieval, biosignal processing, content-based image retrieval, face recognition, genome analysis, music genre classification, speech recognition, technical stock analysis, text retrieval and video surveillance*. Books on Demand, 2 edition, 2011. URL https://www.goodreads.com/book/show/12947368-fundamental-media-understanding.

[68] Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. Clip2video: Mastering video-text retrieval via image clip. *CoRR*, 2021. doi: 10.48550/ARXIV.2106.11097.

[69] Mi Fei and Dit-Yan Yeung. Temporal models for predicting student dropout in massive open online courses. In *Proceedings of the 2015 IEEE International*

Conference on Data Mining Workshop, pages 256–263. IEEE Computer Society, 2015. doi: 10.1109/ICDMW.2015.174.

[70] J. Feliciano, M. Storey, and A. Zagalsky. Student experiences using github in software engineering courses: A case study. In *Proceedings of the 2016 International Conference on Software Engineering Companion*, pages 422–431, 2016. doi: 10.1145/2889160.2889195.

[71] Wenzheng Feng, Jie Tang, and Tracy Xiao Liu. Understanding dropouts in moocs. *Proceedings of the 2019 AAAI Conference on Artificial Intelligence*, 33(1):517–524, 2019. doi: 10.1609/aaai.v33i01.3301517.

[72] Marco Furini. On introducing timed tag-clouds in video lectures indexing. *Multimedia Tools and Applications*, 77:967–984, 2018. doi: 10.1007/s11042-016-4282-5.

[73] Marco Furini, Silvia Mirri, and Manuela Montangero. Taglecture: The gamification of video lecture indexing through quality-based tags. In *Proceedings of the 2017 IEEE Symposium on Computers and Communications*, pages 122–127. IEEE Computer Society, 2017. doi: 10.1109/ISCC.2017.8024516.

[74] Craig S. Galbraith, Gregory B. Merrill, and Doug M. Kline. Are student evaluations of teaching effectiveness valid for measuring student learning outcomes in business related classes? a neural network and bayesian analyses. *Research in Higher Education*, 53(3):353–374, 2012. URL http://www.jstor.org/stable/41475395.

[75] Octavian-Eugen Ganea, Marina Ganea, Aurelien Lucchi, Carsten Eickhoff, and Thomas Hofmann. Probabilistic bag-of-hyperlinks model for entity linking. *CoRR*, 2015. doi: 10.48550/ARXIV.1509.02301. URL http://arxiv.org/abs/1509.02301.

[76] Zijian Gao, Jingyu Liu, Sheng Chen, Dedan Chang, Hao Zhang, and Jinwei Yuan. CLIP2TV: an empirical study on transformer-based methods for video-text retrieval. *CoRR*, 2021. URL https://arxiv.org/abs/2111.05610.

[77] Elena Gaudioso, Miguel Montero, and Felix Hernandez del Olmo. Supporting teachers in adaptive educational systems through predictive models: A proof of concept. *Expert Systems with Applications*, 39(1):621–625, 2012. doi: 10.1016/j.eswa.2011.07.052.

[78] Merzougui Ghalia, Mahieddine Djoudi, and Amel Behaz. Conception and use of ontologies for indexing and searching by semantic contents of video courses. *International Journal of Computer Science Issues*, 8:59–67, 2012. URL https://arxiv.org/abs/1201.5102.

[79] Krishnendu Ghosh, Sharmila Reddy Nangi, Yashasvi Kanchugantla, Pavan Gopal Rayapati, Plaban Kumar Bhowmick, and Pawan Goyal. Augmenting Video Lectures: Identifying Off-topic Concepts and Linking to Relevant

Video Lecture Segments. *International Journal of Artificial Intelligence in Education*, 2021. doi: 10.1007/s40593-021-00257-z.

[80] Niki Gitinabard, Farzaneh Khoshnevisan, Collin Lynch, and Yuan Wang. Your actions or your associates? predicting certification and dropout in moocs with behavioral and social features. *CoRR*, 2018. URL http://arxiv.org/abs/1407.1520.

[81] Peter Goodyear, Melanie Njoo, Hans Hijne, and Jos J.A. van Berkum. Learning processes, learner attributes and simulations. *Education and Computing*, 6(3):263–304, 1991. doi: 10.1016/0167-9287(91)80005-I.

[82] Charles Graham, W. Woodfield, and J.B. Harrison. A framework for institutional adoption and implementation of blended learning in higher education. *The Internet and Higher Education*, 18:4–14, 2012.

[83] Cameron C. Gray and Dave Perkins. Utilizing early engagement and machine learning to predict student outcomes. *Computers & Education*, 131:22 – 32, 2019. doi: 10.1016/j.compedu.2018.12.006.

[84] Pablo Gregori, Vicente Martínez, and Julio José Moyano-Fernández. Basic actions to reduce dropout rates in distance learning. *Evaluation and Program Planning*, 66:48–52, 2018. doi: 10.1016/j.evalprogplan.2017.10.004.

[85] Terry Griffin and Shawn Seals. Github in the classroom: not just for group projects. *Journal of Computing Sciences in Colleges*, 28:74–74, 2013. doi: 10.5555/2458539.2458551.

[86] Ángel GuerreroHigueras, Camino Llamas, Lidia Sánchez, Alexis Fernández, Gonzalo Costales, and Miguel Conde-González. Academic success assessment through version control systems. *Applied Sciences*, 10:1492, 2020. doi: 10.3390/app10041492.

[87] Ángel Manuel GuerreroHigueras, Noemi DeCastro-García, Vicente Matellán, and Miguel Á. Conde. Predictive models of academic success: A case study with version control systems. In *Proceedings of the 2018 International Conference on Technological Ecosystems for Enhancing Multiculturality*, page 306–312. Association for Computing Machinery, 2018. doi: 10.1145/3284179.3284235.

[88] Shuxia Guo, Thomas Bocklitz, Ute Neugebauer, and Jürgen Popp. Common mistakes in cross-validating classification models. *Anal. Methods*, 9:4410–4417, 2017. doi: 10.1039/C7AY01363A.

[89] Marios Hadjieleftheriou and D. Srivastava. Weighted set-based string similarity. *IEEE Data Engineering Bulletin*, 33:25–36, 2010. URL https://www.researchgate.net/publication/220282889_Weighted_Set-Based_String_Similarity.

[90] Sherif A. Halawa, Daniel K. Greene, and John Mck. Mitchell. Dropout prediction in moocs using learner activity features. *Journal of Educational Computing Research*, 57:073563311875701, 2014. doi: 10.1177/0735633118757015.

[91] Jiazhen He, James Bailey, Benjamin I. P. Rubinstein, and Rui Zhang. Identifying at-risk students in massive open online courses. In *Proceedings of the 2015 AAAI Conference on Artificial Intelligence*, page 1749–1755. AAAI Press, 2015.

[92] Jiazhen He, James Bailey, Benjamin IP Rubinstein, and Rui Zhang. Identifying at-risk students in massive open online courses. In *Proceedings of the 2015 AAAI Conference on Artificial Intelligence*, page 1749–1755, 2015. doi: 10.5555/2886521.2886563.

[93] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan C. Russell. Localizing moments in video with natural language. *CoRR*, 2017. URL http://arxiv.org/abs/1708.01641.

[94] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust Disambiguation of Named Entities in Text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792. Association for Computational Linguistics, 2011. URL https://aclanthology.org/D11-1072.

[95] Courtney Hsing and Vanessa Gennarelli. Using github in the classroom predicts student learning outcomes and classroom experiences: Findings from a survey of students and teachers. In *Proceedings of the 2019 ACM Technical Symposium on Computer Science Education*, page 672–678. Association for Computing Machinery, 2019. doi: 10.1145/3287324.3287460.

[96] Ya-Han Hu, Chia-Lun Lo, and Sheng-Pao Shih. Developing early warning systems to predict students' online learning performance. *Computers in Human Behavior*, 36:469–478, 2014. doi: 10.1016/j.chb.2014.04.002.

[97] Jiayuan Huang, Alexander Smola, Arthur Gretton, Karsten Borgwardt, and Bernhard Schölkopf. Correcting sample selection bias by unlabeled data. In *Proceedings of the 2006 Conference on Advances in Neural Information Processing Systems*, pages 601–608. MIT Press, 2006.

[98] Moneeba Iftikhar, Sohail Riaz, and Zahid Yousaf. Impact of youtube tutorials in skill development among university students of lahore. *Pakistan Journal of Distance and Online Learning*, 5:125–138, 2019. URL https://files.eric.ed.gov/fulltext/EJ1266671.pdf.

[99] Ali Shariq Imran, Laksmita Rahadianti, Faouzi Alaya Cheikh, and Sule Yildirim Yayilgan. Semantic tags for lecture videos. In *Proceedings of the 2012 IEEE Sixth International Conference on Semantic Computing*, pages 117–120. IEEE Computer Society, 2012. doi: 10.1109/ICSC.2012.36.

[100] Kalervo Järvelin and Jaana Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 2000 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 41–48. Association for Computing Machinery, 2000. doi: 10.1145/345508.345545.

[101] Nikhil Indrashekhar Jha, Ioana Ghergulescu, and Arghir-Nicolae Moldovan. Oulad mooc dropout and result prediction using ensemble, deep learning and regression techniques. In *Proceedings of the 2019 International Conference on Computer Supported Education*, pages 154–164, 2019. doi: 10.5220/0007767901540164.

[102] Clement Jonquet, Nigam Shah, Cherie Youn, Mark Musen, Chris Callendar, and Margaret-Anne Storey. Ncbo annotator: Semantic annotation of biomedical data. In *Proceedings of the 2009 ACM International Symposium on Wearable Computers*. Association for Computing Machinery, 2009. doi: 10.5555/1636706. URL http://www.lirmm.fr/~jonquet/publications/documents/Demo-ISWC09-Jonquet.pdf.

[103] Robin Kay. Exploring the use of video podcasts in education: A comprehensive review of the literature. *Computers in Human Behavior*, 28:820–831, 2012. doi: 10.1016/j.chb.2012.01.011.

[104] Mahboob Khalid, Valentin Jijkoun, and Maarten Rijke. The impact of named entity normalization on information retrieval for question answering. In *Proceedings of the 2008 European Conference on Information Retrieval*, volume 4956, pages 705–710. Springer, 2008. doi: 10.1007/978-3-540-78646-7_83.

[105] Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. AudioCaps: Generating captions for audios in the wild. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 119–132. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1011.

[106] René Kizilcec and Sherif Halawa. Attrition and achievement gaps in online learning. In *Proceedings of the 2015 ACM Conference on Learning @ Scale*, pages 57–66. Association for Computing Machinery, 2015. doi: 10.1145/2724660.2724680.

[107] Marius Kloft, Felix Stiehler, Zhilin Zheng, and Niels Pinkwart. Predicting MOOC dropout over weeks using machine learning methods. In *Proceedings of the 2014 Empirical Methods In Natural Language Processing Workshop on Analysis of Large Scale Social Interaction in MOOCs*, pages 60–65. Association for Computational Linguistics, 2014. doi: 10.3115/v1/W14-4111.

[108] Kenneth R. Koedinger, Jihee Kim, Julianna Zhuxin Jia, Elizabeth A. McLaughlin, and Norman L. Bier. Learning is not a spectator sport: Doing is better

than watching for learning from a mooc. In *Proceedings of the 2015 ACM Conference on Learning @ Scale*, page 111–120. Association for Computing Machinery, 2015. doi: 10.1145/2724660.2724681.

[109] A. Sophia Koepke, Andreea-Maria Oncescu, Joao Henriques, Zeynep Akata, and Samuel Albanie. Audio retrieval with natural language queries: A benchmark study. *IEEE Transactions on Multimedia*, pages 1–1, 2022. doi: 10.1109/tmm.2022.3149712.

[110] Raga Shalini Koka, Farah Naz Chowdhury, Mohammad Rajiur Rahman, Thamar Solorio, and Jaspal Subhlok. Automatic identification of keywords in lecture video segments. In *Proceedings of the 2020 IEEE International Symposium on Multimedia*, pages 162–165. IEEE Computer Society, 2020. doi: 10.1109/ISM.2020.00035.

[111] Irwan Koto. Teaching and learning science using youtube videos and discovery learning in primary school. *Mimbar Sekolah Dasar*, 7(1):106–118, 2020. doi: 10.53400/mimbar-sd.v7i1.22504.

[112] S. Kotsiantis, K. Patriarcheas, and M. Xenos. A combinational incremental ensemble of classifiers as a technique for predicting students' performance in distance education. *Knowledge-Based Systems*, 23(6):529–535, 2010. doi: 10.1016/j.knosys.2010.03.010.

[113] Sotiris Kotsiantis, Christos Pierrakeas, and P. Pintelas. Preventing student dropout in distance learning using machine learning techniques. In *Proceedings of the 2003 Conference on Knowledge-Based Intelligent Information and Engineering Systems*, pages 267–274. Springer, 2003. doi: 10.1007/978-3-540-45226-3_37.

[114] Sotiris Kotsiantis, Kiriakos Patriarcheas, and Michalis Xenos. A combinational incremental ensemble of classifiers as a technique for predicting student's performance in distance education. *Knowledge-Based Systems*, 23: 529–535, 2010. doi: 10.1016/j.knosys.2010.03.010.

[115] Sotiris Kotsiantis, Nikolaos Tselios, Andromahi Filippidi, and Vassilis Komis. Using learning analytics to identify successful learners in a blended learning course. *International Journal of Technology Enhanced Learning*, 12 2013. doi: 10.1504/IJTEL.2013.059088.

[116] Sotiris B. Kotsiantis, Christos Pierrakeas, Ioannis D. Zaharakis, and Panayiotis E. Pintelas. Efficiency of machine learning techniques in predicting students' performance in distance learning systems, 2005. URL https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.100.7185&rep=rep1&type=pdf.

[117] Zlatko J. Kovacic. Early prediction of student success: Mining students enrolment data. In *Proceedings of the 2010 Informing Science & IT Education Conference*. AAAI Press, 2010. doi: 10.28945/1281.

[118] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E*, 69:066138, 2004. doi: 10.1103/PhysRevE.69.066138.

[119] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. *CoRR*, 2017. doi: 10.48550/ARXIV.1705.00754. URL https://arxiv.org/abs/1705.00754.

[120] Els Kuiper, Monique Volman, and Jan Terwel. The web as an information resource in k–12 education: Strategies for supporting students in searching and processing information. *Review of Educational Research*, 75(3):285–328, 2005. doi: 10.3102/00346543075003285.

[121] Mukesh Kumar, A Singh, and Disha Handa. Literature survey on student's performance prediction in education using data mining techniques. *International Journal of Education and Management Engineering*, 6:40–49, 2017. doi: 10.5815/ijeme.2017.06.05.

[122] David J. Lemay, Clare Baek, and Tenzin Doleck. Comparison of learning analytics and educational data mining: A topic modeling approach. *Computers and Education: Artificial Intelligence*, 2:100016, 2021. doi: 10.1016/j.caeai.2021.100016.

[123] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of Massive Datasets*. Cambridge University Press, 2nd edition, 2014. doi: 10.5555/2787930.

[124] Wentao Li, Min Gao, Hua Li, Qingyu Xiong, Junhao Wen, and Zhongfu Wu. Dropout prediction in moocs using behavior features and multi-view semi-supervised learning. In *Proceedings of the 2016 International Joint Conference on Neural Networks*, pages 3130–3137. IEEE Computer Society, 2016. doi: 10.1109/IJCNN.2016.7727598.

[125] Xiu Li, Lulu Xie, and Huimin Wang. Grade prediction in moocs. In *Proceedings of the 2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES)*, pages 386–392. IEEE Computer Society, 2016. doi: 10.1109/CSE-EUC-DCABES.2016.213.

[126] Jiajun Liang, Chao Li, and Li Zheng. Machine learning application in moocs: Dropout prediction. In *Proceedings of the 2016 International Conference on Computer Science Education*, pages 52–57. IEEE Computer Society, 2016. doi: 10.1109/ICCSE.2016.7581554.

[127] Laura Calvet Liñán and Angel A. Juan Pérez. Educational data mining and learning analytics: differences, similarities, and time evolution. *International Journal of Educational Technology in Higher Education*, 12:98–112, 2015. doi: 10.7238/rusc.v12i3.2515.

[128] Shantanu Lokhande and Vedant Bahel. Effect of non-academic parameters on student's performance, 08 2021.

[129] Owen H. T. Lu, Anna Y. Q. Huang, Jeff C.H. Huang, Albert J. Q. Lin, Hiroaki Ogata, and Stephen J. H. Yang. Applying learning analytics for the early prediction of students' academic performance in blended learning. *Journal of Educational Technology & Society*, 21(2):220–232, 2018. URL https://www.jstor.org/stable/26388400.

[130] Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2(1):2522–5839, 2020.

[131] Ioanna Lykourentzou, Ioannis Giannoukos, Vassilis Nikolopoulos, George Mpardis, and Vassili Loumos. Dropout prediction in e-learning courses through the combination of machine learning techniques. *Computers & Education*, 53(3):950–965, 2009. doi: 10.1016/j.compedu.2009.05.010.

[132] Ioanna Lykourentzou, Ioannis Giannoukos, Vassilis Nikolopoulos, Giorgos Mpardis, and Vassilis Loumos. Dropout prediction in e-learning courses through the combination of machine learning techniques. *Computers & Education*, 53:950–965, 2009. doi: 10.1016/j.compedu.2009.05.010.

[133] Vijaymeena M K and Kavitha K. A survey on similarity measures in text mining. *Machine Learning and Applications: An International Journal*, 3: 19–28, 03 2016. doi: 10.5121/mlaij.2016.3103.

[134] Wei-Hsiu Ma, Yen-Jen Lee, D.H.C. Du, and M.P. McCahill. Video-based hypermedia for education-on-demand. *IEEE MultiMedia*, 5(1):72–83, 1998. doi: 10.1109/93.664744.

[135] Marcos Vinicius Macedo Borges, Julio Cesar dos Reis, and Guilherme Pereira Gribeler. Empirical analysis of semantic metadata extraction from video lecture subtitles. In *Proceedings of the 2019 IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 301–306. IEEE Computer Society, 2019. doi: 10.1109/WETICE.2019.00069.

[136] Marcos Vinícius Macêdo Borges and Julio Cesar dos Reis. Semantic-enhanced recommendation of video lectures. In *Proceedings of the 2019 IEEE International Conference on Advanced Learning Technologies*, volume 2161-377X, pages 42–46. IEEE Computer Society, 2019. doi: 10.1109/ICALT.2019.00013.

[137] Laci Mary Barbosa Manhães, Sérgio Manuel Serra da Cruz, and Geraldo Zimbrão. Wave: An architecture for predicting dropout in undergraduate courses using edm. In *Proceedings of the 2014 Annual ACM Symposium on Applied Computing*, page 243–247. Association for Computing Machinery, 2014. doi: 10.1145/2554850.2555135.

[138] Richard Mayer. Elements of a science of e-learning. *Journal of Educational Computing Research*, 29:297–313, 12 2003. doi: 10.2190/YJLG-09F9-XKAX-753D.

[139] Andrew D. Maynard. How to succeed as an academic on youtube. *Frontiers in Communication*, 5, 2021. doi: 10.3389/fcomm.2020.572181.

[140] Alban Mayra and David Mauricio. Factors to predict dropout at the universities: A case of study in ecuador. In *Proceedings of the 2018 IEEE Global Engineering Education Conference*, pages 1238–1242. IEEE Computer Society, 2018. doi: 10.1109/EDUCON.2018.8363371.

[141] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. *CoRR*, 2019. URL http://arxiv.org/abs/1906.03327.

[142] Keir Mierle, Kevin Laven, Sam Roweis, and Greg Wilson. Mining student cvs repositories for performance indicators. *SIGSOFT Softw. Eng. Notes*, 30 (4):1–5, 2005. doi: 10.1145/1082983.1083150.

[143] Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411. Association for Computational Linguistics, 2004. URL https://aclanthology.org/W04-3252.

[144] Ian J Miller, Evan R Rees, Jennifer Ross, Izaak Miller, Jared Baxa, Juan Lopera, Robert L Kerby, Federico E Rey, and Jason C Kwan. Autometa: automated extraction of microbial genomes from individual shotgun metagenomes. *Nucleic Acids Research*, 47(10):e57–e57, 03 2019. doi: 10.1093/nar/gkz148.

[145] Roxana Moreno and Richard Mayer. A coherence effect in multimedia learning: The case for minimizing irrelevant sounds in the design of multimedia instructional messages. *Journal of Educational Psychology*, 92:117–125, 03 2000. doi: 10.1037/0022-0663.92.1.117.

[146] Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244, 2014. doi: 10.1162/tacl_a_00179.

[147] Saurabh Nagrecha, John Z. Dillon, and Nitesh V. Chawla. Mooc dropout prediction: Lessons learned from making pipelines interpretable. In *Proceedings of the 2017 International Conference on World Wide Web Companion*, page 351–359. International World Wide Web Conferences Steering Committee, 2017. doi: 10.1145/3041021.3054162.

[148] Marcell Nagy and Roland Molontay. Predicting dropout in higher education based on secondary school performance. In *2018 IEEE 22nd International*

Conference on Intelligent Engineering Systems (INES), pages 000389–000394. IEEE Computer Society, 2018. doi: 10.1109/INES.2018.8523888.

[149] Jovial Niyogisubizo, Lyuchao Liao, Eric Nziyumva, Evariste Murwanashyaka, and Pierre Claver Nshimyumukiza. Predicting student's dropout in university classes using two-layer ensemble machine learning approach: A novel stacked generalization. *Computers and Education: Artificial Intelligence*, 3:100066, 2022. doi: 10.1016/j.caeai.2022.100066.

[150] Jovial Niyogisubizo, Lyuchao Liao, Eric Nziyumva, Evariste Murwanashyaka, and Pierre Claver Nshimyumukiza. Predicting student's dropout in university classes using two-layer ensemble machine learning approach: A novel stacked generalization. *Computers & Education: Artificial Intelligence*, 3:100066, 2022. doi: 10.1016/j.caeai.2022.100066.

[151] Jalal Nouri, Mohammed Saqr, and Uno Fors. Predicting performance of students in a flipped classroom using machine learning: towards automated data-driven formative feedback. *Journal of Systemics, Cybernetics and Informatics*, 08 2019. URL http://www.iiisci.org/Journal/pdv/sci/pdfs/EB614LI19.pdf.

[152] Sandra Nunn, John Avella, Therese Kanai, and Mansureh Kebritchi. Learning analytics methods, benefits, and challenges in higher education: A systematic literature review. *Online Learning*, 20(2), 2016. doi: 10.24059/olj.v20i2.790.

[153] Ogechi Ohadomere and Ikedinachi K. Ogamba. Management-led interventions for workplace stress and mental health of academic staff in higher education: a systematic review. *Journal of Mental Health Training, Education and Practice*, 16(1):67–82, 2021. doi: 10.1108/JMHTEP-07-2020-0048.

[154] Andreea-Maria Oncescu, A. Sophia Koepke, João F. Henriques, Zeynep Akata, and Samuel Albanie. Audio retrieval with natural language queries. *CoRR*, 2021. URL https://arxiv.org/abs/2105.02192.

[155] Edin Osmanbegović and Mirza Suljic. Data mining approach for predicting student performance. *Journal of Economics & Business/Economic Review*, 10: 3–12, 2012. doi: 10.1007/s10639-018-9839-7.

[156] Saurabh Pal, Pijush Dutta Pramanik, Tripti Majumdar, and Prasenjit Choudhury. A semi-automatic metadata extraction model and method for video-based e-learning contents. *Education and Information Technologies*, 24, 11 2019. doi: 10.1007/s10639-019-09926-y.

[157] Karl Pearson. *On the Criterion that a Given System of Deviations from the Probable in the Case of a Correlated System of Variables is Such that it Can be Reasonably Supposed to have Arisen from Random Sampling*, pages 11–28. Springer, 1992. doi: 10.1007/978-1-4612-4380-9_2.

[158] Yuxin Peng, Xiaohua Zhai, Yunzhen Zhao, and Xin Huang. Semi-supervised cross-media feature learning with unified patch graph regularization. *IEEE*

Transactions on Circuits and Systems for Video Technology, 26(3):583–596, 2016. doi: 10.1109/TCSVT.2015.2400779.

[159] Yuxin Peng, Xin Huang, and Yunzhen Zhao. An overview of cross-media retrieval: Concepts, methodologies, benchmarks and challenges. *CoRR*, 2017. doi: 10.48550/ARXIV.1704.02223.

[160] Yuxin Peng, Xin Huang, and Yunzhen Zhao. An overview of cross-media retrieval: Concepts, methodologies, benchmarks, and challenges. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(9):2372–2385, 2018. doi: 10.1109/TCSVT.2017.2705068.

[161] Filipe D. Pereira, Elaine Oliveira, Alexandra Cristea, David Fernandes, Luciano Silva, Gene Aguiar, Ahmed Alamri, and Mohammad Alshehri. Early dropout prediction for programming courses supported by online judges. In *Artificial Intelligence in Education*, pages 67–72. Springer, 2019. doi: 10.1007/978-3-030-23207-8_13.

[162] Boris Pérez, Camilo Castellanos, and Darío Correal. Predicting student dropout rates using data mining techniques: A case study. In Alvaro David Orjuela-Cañón, Juan Carlos Figueroa-García, and Julián David Arias-Londoño, editors, *Applications of Computational Intelligence*, pages 111–125. Springer, 2018. doi: 10.1007/978-3-030-03023-0_10.

[163] Ángel Pérez-Lemonche, Gonzalo Martínez-Muñoz, and Estrella Pulido-Cañabate. Analysing event transitions to discover student roles and predict grades in moocs. In *Proceedings of the 2017 International Conference on Artificial Neural Networks*, pages 224–232. Springer, 2017. doi: 10.1007/978-3-319-68612-7_26.

[164] Emily Grossnickle Peterson and Patricia A. Alexander. Navigating print and digital sources: Students' selection, use, and integration of multiple sources across mediums. *The Journal of Experimental Education*, 88(1):27–46, 2020. doi: 10.1080/00220973.2018.1496058.

[165] Gerald Petz, Werner Wetzlinger, and Dietmar Nedbal. Improving language-dependent named entity detection. In Andreas Holzinger, Peter Kieseberg, A Min Tjoa, and Edgar Weippl, editors, *Proceedings of the 2017 Machine Learning and Knowledge Extraction Conference*, pages 330–345. Springer, 2017. doi: 10.1007/978-3-319-66808-6_22.

[166] Bardh Prenkaj, Paola Velardi, Giovanni Stilo, Damiano Distante, and Stefano Faralli. A survey of machine learning approaches for student dropout prediction in online courses. *ACM Computer Survey*, 53(3), 2020. doi: 10.1145/3388792.

[167] Minna Puustinen and Jean-François Rouet. Learning with new technologies: Help seeking and information searching revisited. *Computers & Education*, 53(4):1014–1019, 2009. doi: 10.1016/j.compedu.2008.07.002.

[168] Feiyue Qiu, Guodao Zhang, Xin Sheng, Lei Jiang, Lijia Zhu, Qifeng Xiang, Bo Jiang, and Ping-kuo Chen. Predicting students' performance in e-learning using learning process and behaviour data. *Scientific Reports*, 12(1):453, 2022. doi: 10.1038/s41598-021-03867-8.

[169] Jiezhong Qiu, Jie Tang, Tracy Xiao Liu, Jie Gong, Chenhui Zhang, Qian Zhang, and Yufei Xue. Modeling and predicting learning behavior in moocs. In *Proceedings of the 2016 ACM International Conference on Web Search and Data Mining*, page 93–102. Association for Computing Machinery, 2016. doi: 10.1145/2835776.2835842.

[170] Lin Qiu, Yanshen Liu, Quan Hu, and Yi Liu. Student dropout prediction in massive open online courses by convolutional neural networks. *Soft Computing*, 23:10287–10301, 2019. doi: 10.1007/s00500-018-3581-3.

[171] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, 2021. URL https://arxiv.org/abs/2103.00020.

[172] Rodolfo C. Raga and Jennifer D. Raga. Early prediction of student performance in blended learning courses using deep neural networks. In *2019 International Symposium on Educational Technology (ISET)*, pages 39–43, 2019. doi: 10.1109/ISET.2019.00018.

[173] Faisal Rahutomo, Teruaki Kitasuka, and Masayoshi Aritsugi. Semantic cosine similarity. In *Proceedings of the 2012 International Student Conference on Advanced Science and Technology*. Springer, 2012. URL https://www.researchgate.net/profile/Faisal_Rahutomo/publication/262525676_Semantic_Cosine_Similarity/links/0a85e537ee3b675c1e000000.pdf.

[174] Kaviyarasi Ramanathan and T. Balasubramanian. Exploring the high potential factors that affects students' academic performance. *International Journal of Education and Management Engineering*, 8:15–23, 11 2018. doi: 10.5815/ijeme.2018.06.02.

[175] Arti Ramesh, Dan Goldwasser, Bert Huang, Hal Daumé, and Lise Getoor. Learning latent engagement patterns of students in online courses. In *Proceedings of the 2014 AAAI Conference on Artificial Intelligence*, page 1272–1278. AAAI Press, 2014. doi: 10.5555/2893873.2894071.

[176] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, 2019. URL http://arxiv.org/abs/1908.10084.

[177] S. Repp and M. Meinel. Semantic indexing for recorded educational lecture videos. In *Proceedings of the 2006 IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 245–250. IEEE Computer Society, 2006. doi: 10.1109/PERCOMW.2006.122.

[178] Soo Young Rieh, Kevyn Collins-Thompson, Preben Hansen, and Hye-Jung Lee. Towards searching as a learning process: A review of current perspectives and future directions. *Journal of Information Science*, 42(1):19–34, 2016. doi: 10.1177/0165551515615841.

[179] Carly Robinson, Michael Yeomans, Justin Reich, Chris Hulleman, and Hunter Gehlbach. Forecasting student achievement in moocs with natural language processing. In *Proceedings of the 2016 International Conference on Learning Analytics & Knowledge*, pages 383–387. Association for Computing Machinery, 2016. doi: 10.1145/2883851.2883932.

[180] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher J. Pal, Hugo Larochelle, Aaron C. Courville, and Bernt Schiele. Movie description. *CoRR*, 2016. URL http://arxiv.org/abs/1605.03705.

[181] Léon Rothkrantz. Dropout rates of regular courses and moocs. In *Computers Supported Education*, pages 25–46. Springer, 2017. doi: 10.1007/978-3-319-94640-5.

[182] Angelo A. Salatino, Thiviyan Thanapalasingam, Andrea Mannocci, Aliaksandr Birukou, Francesco Osborne, and Enrico Motta. The Computer Science Ontology: A Comprehensive Automatically-Generated Taxonomy of Research Areas. *Data Intelligence*, 2(3):379–416, 2020. doi: 10.1162/dint_a_00055.

[183] Angelo A. Salatino, Francesco Osborne, Thiviyan Thanapalasingam, and Enrico Motta. The CSO classifier: Ontology-driven detection of research topics in scholarly articles. *CoRR*, 2021. URL https://arxiv.org/abs/2104.00948.

[184] Carlos Luis Sanchez Bocanegra, Jose Luis Sevillano Ramos, Carlos Rizo, Anton Civit, and Luis Fernandez-Luque. HealthRecSys: A semantic content-based recommender system to complement health videos. *BMC Medical Informatics and Decision Making*, 17(1):63, 2017. doi: 10.1186/s12911-017-0431-7.

[185] Mark Sanderson. Test collection based evaluation of information retrieval systems. *Foundations and Trends in Information Retrieval*, 4:247–375, 01 2010. doi: 10.1561/1500000009.

[186] Michael Sankey, Dawn Birch, and Michael Gardiner. Engaging students through multimodal learning environments: The journey continues. In *Proceedings of the 2010 Australian Society for Computers in Learning in Tertiary Education Annual Conference*, pages 852–863. Australasian Society for Computers in Learning in Tertiary Education, 2010. URL https://www.learntechlib.org/p/45485.

[187] Farhana Sarker, Thanassis Tiropanis, and Hugh Davis. Students' performance prediction by using institutional internal and external open data sources. *Proceedings of the 2013 International Conference on Computer Supported Education*, pages 639–646, 2013. URL https://eprints.soton.ac.uk/353532/1/Students%2527%2520mark%2520prediction%2520model.pdf.

[188] Omid Shahmirzadi, Adam Lugowski, and Kenneth Younge. Text similarity in vector space models: A comparative study. *CoRR*, 2018. URL http://arxiv.org/abs/1810.00664.

[189] Mike Sharkey and Robert F. Sanders. A process for predicting mooc attrition. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2014. doi: 10.3115/v1/W14-4109.

[190] Huang-Chia Shih and Chung-Lin Huang. Content-based multi-functional video retrieval system. In *2005 Digest of Technical Papers - IEEE International Conference on Consumer Electronics*, pages 383–384. IEEE Computer Society, 2005. doi: 10.1109/ICCE.2005.1429878.

[191] Jeff Shrager and David Klahr. Instructionless learning about a complex device: the paradigm and observations. *International Journal of Man-Machine Studies*, 25(2):153–189, 1986. doi: 10.1016/S0020-7373(86)80075-X.

[192] George Siemens and Ryan S. J. d. Baker. Learning analytics and educational data mining: Towards communication and collaboration. In *Proceedings of the 2012 International Conference on Learning Analytics and Knowledge*, page 252–254. Association for Computing Machinery, 2012. doi: 10.1145/2330601.2330661.

[193] Vineet Kumar Singh and Maitreyee Dutta. Analyzing cryptographic algorithms for secure cloud network. *CoRR*, 2014. URL http://arxiv.org/abs/1407.1520.

[194] Tanmay Sinha, Nan Li, Patrick Jermann, and Pierre Dillenbourg. Capturing "attrition intensifying" structural traits from didactic interaction sequences of MOOC learners. In *Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs*, pages 42–49. Association for Computational Linguistics, 2014. doi: 10.3115/v1/W14-4108.

[195] Azzam Sleit, Moaath Hajaya, and Farhan Alebeisat. Video powersearcher: A text-based indexing e-learning system. In *Proceedings of the 2010 International Conference on Intelligent Semantic Web-Services and Applications*, page 23. Association for Computing Machinery, 2010. doi: 10.1145/1874590.1874613.

[196] Martin Solis, Tania Moreira, Roberto Gonzalez, Tatiana Fernandez, and Maria Hernandez. Perspectives to predict dropout in university students with machine learning. In *Proceedings of the 2018 IEEE International Work*

Conference on Bioinspired Intelligence, pages 1–6. IEEE Computer Society, 2018. doi: 10.1109/IWOBI.2018.8464191.

[197] René Speck and Axel-Cyrille Ngonga Ngomo. Ensemble learning for named entity recognition. In *The International Semantic Web Conference*, page 519–534. Springer, 2014. doi: 10.1007/978-3-319-11964-9_33.

[198] Andreas Spitz, Vaibhav Dixit, Ludwig Richter, Michael Gertz, and Johanna Geiß. State of the union: A data consumer's perspective on wikidata and its properties for the classification and resolution of entities. In *Proceedings of the 2016 International Conference on Web and Social Media Workshop*. AAAI Press, 2016. URL http://aaai.org/ocs/index.php/ICWSM/ICWSM16/paper/view/13200.

[199] Gina Sprint and Jason Conci. Mining github classroom commit behavior in elective and introductory computer science courses. *J. Comput. Sci. Coll.*, 35 (1):76–84, 2019. doi: 10.5555/3381540.3381548.

[200] Heather Staker and Michael Horn. Classifying k–12 blended learning. *Open Journal of Modern Linguistics*, 5(1), 2012. URL http://www.christenseninstitute.org/wp-content/uploads/2013/04/Classifying-K-12-blended-learning.pdf.

[201] Lars St»hle and Svante Wold. Analysis of variance (anova). *Chemometrics and Intelligent Laboratory Systems*, 6(4):259–272, 1989. doi: 10.1016/0169-7439(89)80095-4.

[202] Rajendrarao Sumitha and E. S. Vinothkumar. Prediction of students outcome using data mining techniques. *International Journal of Scientific Engineering and Applied Science*, 2, 2016. URL https://ijseas.com/volume2/v2i6/ijseas20160615.pdf.

[203] Elaine Tan. Informal learning on youtube: exploring digital literacy in independent online learning. *Learning, Media and Technology*, 38(4):463–477, 2013. doi: 10.1080/17439884.2013.783594.

[204] Mingjie Tan and Peiji Shao. Prediction of student dropout in e-learning program through the use of machine learning method. *International Journal of Emerging Technologies in Learning*, 10(1):11–17, 2015. doi: 10.3991/i-jet.v10i1.4189.

[205] Pang-Ning Tan and Vipin Kumar. Interestingness measures for association patterns : A perspective. In *Proceedings of the 2000 Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2000. URL http://cs.fit.edu/~pkc/ml/related/tan-postkdd00.pdf.

[206] Jeff KT Tang, Haoran Xie, and Tak-Lam Wong. A big data framework for early identification of dropout students in MOOC. In *Proceedings of the 2015 International Conference on Technology in Education*, pages 127–132. Springer, 2015. doi: 10.1007/978-3-662-48978-9_12.

[207] Lijun Tang and J.R. Kender. Semantic indexing for instructional video via combination of handwriting recognition and information retrieval. In *Proceedings of the 2005 IEEE International Conference on Multimedia and Expo*, pages 920–923. IEEE Computer Society, 2005. doi: 10.1109/ICME.2005.1521574.

[208] Colin Taylor, Kalyan Veeramachaneni, and Una-May O'Reilly. Likely to stop? predicting stopout in massive open online courses. *CoRR*, 2014. URL http://arxiv.org/abs/1408.3382.

[209] Nguyen Thai-Nghe, Lucas Drumond, Artus Krohn-Grimberghe, and Lars Schmidt-Thieme. Recommender system for predicting student performance. *Procedia Computer Science*, 1(2):2811–2819, 2010. doi: 10.1016/j.procs.2010.08.006.

[210] Shuo-Chang Tsai, Cheng-Huan Chen, Yi-Tzone Shiao, Jin-Shuei Ciou, and Trong-Neng Wu. Precision education with statistical learning and deep learning: A case study in taiwan. *RUSC. Universities and Knowledge Society Journal*, 17:12, 2020. doi: 10.1186/s41239-020-00186-2.

[211] Tayfun Tuna, Jaspal Subhlok, and Shishir Shah. Indexing and keyword search to ease navigation in lecture videos. In *Proceedings of the 2011 IEEE Applied Imagery Pattern Recognition Workshop*, pages 1–8. IEEE Computer Society, 2011. doi: 10.1109/AIPR.2011.6176364.

[212] Tayfun Tuna, Jaspal Subhlok, Lecia Barker, Varun Varghese, Olin Johnson, and Shishir Shah. Development and evaluation of indexed captioned searchable videos for stem coursework. In *Proceedings of the 2012 ACM Technical Symposium on Computer Science Education*, page 129–134. Association for Computing Machinery, 2012. doi: 10.1145/2157136.2157177.

[213] Tayfun Tuna, Jaspal Subhlok, Lecia Barker, Shishir Shah, Olin Johnson, and Christopher Hovey. Indexed Captioned Searchable Videos: A Learning Companion for STEM Coursework. *Journal of Science Education and Technology*, 26:82–99, 2017. doi: 10.1007/s10956-016-9653-1.

[214] Miroslav Tushev, Grant Williams, and Anas Mahmoud. Using github in large software engineering classes. an exploratory case study. *Computer Science Education*, 30(2):155–186, 2020. doi: 10.1080/08993408.2019.1696168.

[215] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Sören Auer, Daniel Gerber, Andreas Both, and Sandro Coelho. Agdistis - graph-based disambiguation of named entities using linked data. In *Proceedings of the 2014 International Semantic Web Conference*. Springer, 2014. doi: 10.1007/978-3-319-11964-9_29.

[216] Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, Paolo Ferragina, Christiane Lemke, Andrea Moro, Roberto Navigli, Francesco Piccinno, Giuseppe Rizzo, Harald Sack,

René Speck, Raphaël Troncy, Jörg Waitelonis, and Lars Wesemann. Gerbil: General entity annotator benchmarking framework. In *Proceedings of the 2015 International Conference on World Wide Web*, page 1133–1143. International World Wide Web Conferences Steering Committee, 2015. doi: 10.1145/2736277.2741626.

[217] Steven Van Goidsenhoven, Daria Bogdanova, Galina Deeva, Seppe vanden Broucke, Jochen De Weerdt, and Monique Snoeck. Predicting student success in a blended learning environment. In *Proceedings of the 2020 International Conference on Learning Analytics & Knowledge*, page 17–25. Association for Computing Machinery, 2020. doi: 10.1145/3375462.3375494.

[218] Tatjana Vasileva-Stojanovska, Toni Malinovski, Marina Vasileva, Dobri Jovevski, and Vladimir Trajkovik. Impact of satisfaction, personality and learning style on educational outcomes in a blended learning environment. *Learning and Individual Differences*, 38:127–135, 2015. doi: 10.1016/j.lindif.2015.01.018.

[219] Norman Vaughan. *Teaching in Blended Learning Environments: Creating and Sustaining Communities of Inquiry*. AU Press, 2013. URL https://www.learntechlib.org/primary/p/180547/.

[220] Adriano Veloso, Wagner Meira, and Mohammed J. Zaki. Lazy associative classification. In *Proceedings of the 2006 International Conference on Data Mining*, pages 645–654. IEEE Computer Society, 2006. doi: 10.1109/ICDM.2006.96.

[221] Carlos Villagrá, Francisco José Durán, Patricia Rosique, Faraón Llorens, and Rafael Molina-Carmona. Predicting academic performance from behavioural and learning data. *International Journal of Design & Nature and Ecodynamics*, 11:239–249, 2016. doi: 10.2495/DNE-V11-N3-239-249.

[222] Minh Hien Vo, Chang Zhu, and Anh Diep. Students' performance in blended learning: disciplinary difference and instructional design factors. *Journal of Computers in Education*, 7:487–510, 04 2020. doi: 10.1007/s40692-020-00164-7.

[223] Wei Wang, Han Yu, and Chunyan Miao. Deep model for dropout prediction in moocs. In *Proceedings of the 2017 International Conference on Crowd Science and Engineering*, page 26–32. Association for Computing Machinery, 2017. doi: 10.1145/3126973.3126990.

[224] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. VATEX: A large-scale, high-quality multilingual dataset for video-and-language research. *CoRR*, 2019. URL http://arxiv.org/abs/1904.03493.

[225] Jacob Whitehill, Joseph Williams, Glenn Lopez, Cody Coleman, and Justin Reich. Beyond prediction: First steps toward automatic intervention in mooc

student stopout. In *Proceedings of the 2015 International Conference on Educational Data Mining*. ERIC, 06 2015. doi: 10.2139/ssrn.2611750.

[226] Jacob Whitehill, K. V. Ram Mohan, Daniel T. Seaton, Yigal Rosen, and Dustin Tingley. Delving deeper into mooc student dropout prediction. *CoRR*, 2017. URL https://arxiv.org/pdf/1702.06404.pdf.

[227] Jacob Whitehill, Kiran Mohan, Daniel Seaton, Yigal Rosen, and Dustin Tingley. Mooc dropout prediction: How to measure accuracy? In *Proceedings of the 2017 ACM Conference on Learning @ Scale*, page 161–164, New York, NY, USA, 2017. Association for Computing Machinery. doi: 10.1145/3051457.3053974.

[228] Wikipedia. Named-entity recognition — Wikipedia, the free encyclopedia, 2022. URL http://en.wikipedia.org/w/index.php?title=Named-entity%20recognition&oldid=1073163208. [Online; accessed 08-April-2022].

[229] Wikipedia. Named entity — Wikipedia, the free encyclopedia, 2022. URL http://en.wikipedia.org/w/index.php?title=Named%20entity&oldid=1061073252. [Online; accessed 06-April-2022].

[230] Marlon Xavier and Julio Meneses. A literature review on the definitions of dropout in online higher education. In *Proceedings of the 2020 Annual Conference of European Distance and E-Learning Network*. European Distance and E-Learning Network, 2020. doi: 10.38069/edenconf-2020-ac0004.

[231] Wanli Xing, Xin Chen, Jared Stein, and Michael Marcinkowski. Temporal predication of dropouts in moocs: Reaching the low hanging fruit through stacking generalization. *Computers in Human Behavior*, 58:119–129, 2016. doi: 10.1016/j.chb.2015.12.007.

[232] Bin Xu and Dan Yang. Motivation classification and grade prediction for moocs learners. *Intell. Neuroscience*, 2016, jan 2016. doi: 10.1155/2016/2174613.

[233] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 5288–5296. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.571.

[234] Zhuojia Xu, Hua Yuan, and Qishan Liu. Student performance prediction based on blended learning. *IEEE Transactions on Education*, 64(1):66–73, 2021. doi: 10.1109/TE.2020.3008751.

[235] Diyi Yang, Tanmay Sinha, David Adamson, and Carolyn Penstein Rose. Turn on, tune in, drop out: Anticipating student dropouts. In *Proceedings of the 2013 NIPS Data-Driven Education Workshop*, 2013. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.681.7808&rep=rep1&type=pdf.

[236] Jianwei Yang, Yonatan Bisk, and Jianfeng Gao. Taco: Token-aware cascade contrastive learning for video-text alignment. *CoRR*, 2021. URL https://arxiv.org/abs/2108.09980.

[237] Tsung-Yen Yang, Christopher G. Brinton, Carlee Joe-Wong, and Mung Chiang. Behavior-based grade prediction for moocs via time series neural networks. *IEEE Journal of Selected Topics in Signal Processing*, 11(5):716–728, 2017. doi: 10.1109/JSTSP.2017.2700227.

[238] Hong Qing Yu, Carlos Pedrinaci, Stefan Dietze, and John Domingue. Using linked data to annotate and search educational video resources for supporting distance learning. *IEEE Transactions on Learning Technologies*, 5(2):130–142, 2012. doi: 10.1109/TLT.2012.1.

[239] Nick Z. Zacharis. A multivariate approach to predicting student outcomes in web-enabled blended learning courses. *The Internet and Higher Education*, 27:44–53, 2015. doi: 10.1016/j.iheduc.2015.05.002.

[240] Nick Z. Zacharis. Predicting student academic performance in blended learning using artificial neural networks. *International Journal of Artificial Intelligence & Applications*, 7:17–29, 2016. doi: 10.5121/IJAIA.2016.7502.

[241] Maryam Zaffar, Manzoor Ahmed Hashmani, K.S. Savita, and Syed Sajjad Hussain Rizvi. A study of feature selection algorithms for predicting students academic performance. *International Journal of Advanced Computer Science and Applications*, 9(5), 2018. doi: 10.14569/IJACSA.2018.090569.

[242] Alexey Zagalsky, Joseph Feliciano, Margaret-Anne Storey, Yiyun Zhao, and Weiliang Wang. The emergence of github as a collaborative platform for education. In *Proceedings of the 2018Conference on Computer Supported Cooperative Work & Social Computing*, page 1906–1917. Association for Computing Machinery, 2015. doi: 10.1145/2675133.2675284.

[243] Mohammed J. Zaki and Wagner Meira, Jr. *Data Mining and Machine Learning: Fundamental Concepts and Algorithms*. Cambridge University Press, 2 edition, 2020. doi: 10.1017/9781108564175.

[244] Azizah Zakiah and Mohamad Fauzan. Collaborative learning model of software engineering using github for informatics student. In *Proceedings of the 2016 International Conference on Cyber and IT Service Management*, pages 1–5. IEEE Computer Society, 2016. doi: 10.1109/CITSM.2016.7577521.

[245] Amrapali Zaveri, Dimitris Kontokostas, Sebastian Hellmann, Jürgen Umbrich, Michael Färber, Frederic Bartscherer, Carsten Menne, Achim Rettinger, Amrapali Zaveri, Dimitris Kontokostas, Sebastian Hellmann, and Jürgen Umbrich. Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago. *Semantic Web*, 9(1):77–129, 2018. doi: 10.3233/SW-170275.

[246] Xiaohua Zhai, Yuxin Peng, and Jianguo Xiao. Heterogeneous metric learning with joint graph regularization for cross-media retrieval. In *Proceedings of the 2013 AAAI Conference on Artificial Intelligence*, page 1198–1204. AAAI Press, 2013. doi: 10.5555/2891460.2891627.

[247] Xiaohua Zhai, Yuxin Peng, and Jianguo Xiao. Learning cross-media joint representation with sparse and semisupervised regularization. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(6):965–978, 2014. doi: 10.1109/TCSVT.2013.2276704.

[248] Dongsong Zhang, Lina Zhou, Robert O. Briggs, and Jay F. Nunamaker. Instructional video in e-learning: Assessing the impact of interactive video on learning effectiveness. *Information & Management*, 43(1):15–27, 2006. doi: 10.1016/j.im.2005.01.004.

[249] Luowei Zhou, Chenliang Xu, and Jason J. Corso. Procnets: Learning to segment procedures in untrimmed and unconstrained videos. *CoRR*, 2017. URL http://arxiv.org/abs/1703.09788.

[250] Mu Zhu. Recall, precision and average precision, 2004. URL https://www.researchgate.net/publication/228874142_Recall_precision_and_average_precision.

[251] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *Proceedings of the 2019 Computer Vision and Pattern Recognition Conference*, pages 3532–3540. Computer Vision Foundation, 2019. doi: 10.1109/CVPR.2019.00365.

[252] Malgorzata Zywno. A contribution to validation of score meaning for felder soloman's index of learning styles. *Proceedings of the 2003 American Society for Engineering Education Annual Conference & Exposition*, 119, 01 2003. doi: 10.18260/1-2–12424.

[253] Ángel Manuel GuerreroHigueras, Noemí DeCastroGarcía, Francisco Javier RodriguezLera, Vicente Matellán, and Miguel Ángel Conde. Predicting academic success through students' interaction with version control systems. *Open Computer Science*, 9(1):243 – 251, 2019. doi: 10.1515/comp-2019-0012.

Appendix A

Background knowledge

Semantic web basics

Knowledge base

A knowledge base (KB) is a technology used to store complex structured and unstructured information used by a computer system, that employs Resource Description Framework (RDF) data model to represent knowledge in a machine-readable way. RDF stores data in triples composed by a subject, a predicate and an object. Subject and predicate are URIs, while object be either an URI or plain text. An Uniform Resource Identifier (URI) is a string of characters that uniquely identify a name or a resource on the internet; Uniform Resource Locator (URL) is a type of URI that specifies not only a resource, but how to reach it on the internet.

To clarify RDF format, some examples of triples related to Wikidata knowledge base are presented below:

1. `wd:Q16559571 wdt:P31 wd:Q5`
2. `wd:Q16559571 rdfs:label "Giorgio Montanini"@en`
3. `wd:Q5 rdfs:label "human"@en`
4. `wdt:P31 rdfs:label "instance of"@en`

The 4 triples mean that Giorgio Montanini is a human.

They use the following list of prexifes:

- wd prefix is the shortening of <http://www.wikidata.org/entity/> that denotes Wikidata subject or objects referenced by URIs
- wdt is the shortening of <http://www.wikidata.org/prop/direct/> that denotes Wikidata predicates
- rdfs is the shortening of <http://www.w3.org/2000/01/rdf-schema#> that denotes the label related to the given subject.

@en suffix indicates that the label is in English.

Wikidata

Wikidata is a free and open knowledge base including structured content derived from Wikipedia (link:https://www.wikidata.org/wiki/Wikidata:Main_Page).

A Wikidata *entity* refers to a subject or an object identified by an URI; a Wikidata *property* refers to a predicate. Each Wikidata entity or property is associated to a web page.

A peculiar feature of Wikidata is the orderly and unambiguous organization of content; this allows to derive parental relationships between entities through the following predicates:

- http://www.wikidata.org/prop/direct/P31 (*instance of*)
- http://www.wikidata.org/prop/direct/P279 (*class of*)
- http://www.wikidata.org/prop/direct/P361 (*part of*)

Named entity recognition and linking

A *named entity* is a real-world object, such as a person, location, organization, product, etc., that can be denoted with a proper name [229]. *Named entity recognition* is a subtask of information extraction that seeks to locate and classify named entities mentioned in unstructured text into pre-defined categories such as person names,

organizations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc. [228]. *Named entity linking* is the task of assigning a unique identity to entities mentioned in text [104]; most of the cases they are linked to a knowledge base.

Evaluation metrics

This section provides a formal definition of the evaluation metrics employed in the dissertation.

Classification evaluation metrics

The metrics used to evaluate the classification are Precision, Recall, F1-Score, Accuracy and Balanced Accuracy, that rely on the concepts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The terms positive and negative refer to the classifier's prediction (the expectation), and the terms true and false refer to whether that prediction corresponds to the ground truth (the observation).

Precision

Precision is the fraction of relevant instances among the retrieved instances and it is defined by the following formula:

$$Precision = \frac{TP}{TP + FP}$$

Recall

Recall (also known as sensitivity) is the fraction of relevant instances that were retrieved and it is defined by the following formula:

$$Recall = \frac{TP}{TP + FN}$$

F1-Score

F1-Score is the harmonic mean of precision and recall:

$$F1Score = \frac{Precision \cdot Recall}{Precision + Precision}$$

Accuracy

Accuracy is the proportion of correct predictions among the total number of cases examined

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy can be a misleading metric for imbalanced data sets.

Balanced Accuracy

Balanced accuracy [243] can serve as an overall performance metric for a model. It is especially useful when the classes are imbalanced since it rewards the predictions made for the samples belonging to the minority class whereas penalizes those made for the samples of the majority class.

$$Balanced\ Accuracy = \frac{1}{2}(\frac{TP}{TP + FP} + \frac{TN}{TN + FN})$$

Retrieval evaluation metrics

The metrics utilized to evaluate retrieval task are Precision at k (P@k), Recall at k (R@k)and Mean Average Precision (MAP). While MAP is computed using the entire list of returned resources sorted by similarity score, P@k and R@k cutoff the list at the k entity.

Precision at k

$$Precision\ at\ k = P@k = \frac{recommended\ items\ at\ k\ that\ are\ relevant}{number\ of\ recommended\ items\ at\ k}$$

Recall at k

$$Recall\ at\ k = R@k = \frac{recommended\ items\ at\ k\ that\ are\ relevant}{total\ number\ of\ relevant\ items}$$

Mean Average Precision

The Mean Average Precision (MAP) is defined in [250]. Given an input query, the Average Precision (AP) is computed as follows:

$$AP = \frac{\sum_{n}^{k=1} (P(k) \times rel(k))}{num.\ of\ retrieved\ relevant\ resources}$$

where $P(k)$ is the precision at k, whereas $rel(k)$ is an indicator function that takes value one if the retrieved resource at rank k is relevant, zero otherwise.

The MAP score is the average AP over all the performed queries.

Shapley Additive Explanations

Shapley Additive Explanations (SHAP), is a method for the interpretation of predictions of ML models through Shapely values. SHAP values are based on game theory where the “game” is reproducing the outcome of the model and the “players” are the features $\vec{f}$ included in the model; to determine the importance of a single feature/-player the outcome of each possible combination (or coalition) of features/players should be considered. SHAP requires to train a distinct predictive model for each distinct coalition, i.e. $2^{\|\vec{f}\|}$ models.

The SHAP value $SHAP(f_1)$ of a feature f_1 corresponds to the weighted sum of its "marginal contributions" in the various models in which it is employed. The marginal contribution of a feature f_1 for a model $M1$ starting from a model $M2$ in which are present all the features of $M1$ except f_1 corresponds to the difference of the prediction scores between $M1$ and $M2$; it represents the effect of that additional feature on the outcome.

SHAP values are determined for each data sample (one game:one observation), hence they offer local explainability; to estimate the global explainability (i.e., over all data) of a feature f_1, mean of absolute shap values is taken into account:

$\frac{\sum_{j=1}^{N} \|SHAP_j(f_1))\|}{N}$ where N is the total number of samples and $SHAP_j(f_1)$ corresponds to the SHAP value of feature f_1 on the jth data sample.

Appendix B

Educational content presented in BookToYout

Each educational topic in BookToYout corresponds to a chapter in an instructional book. The complete list of chapter titles is given here:

- *Elementary Graph Algorithms*
- *Medians And Order Statistics*
- *Adding Jtextfields And Jbuttons To A Jframe*
- *Boosting*
- *B-trees*
- *Steady-state Errors*
- *Transient Response Via Gain Adjustment*
- *Running A Java Application And Correcting Logic Errors*
- *Steady-state Error For Systems In State Space*
- *The Weak Law Of Large Numbers*
- *Implementing Tf-idf*
- *Using Views To Simplify Queries*
- *Inferencing In The Good Relations Ontology*
- *Improving Performance When Updating Large Tables*
- *Inference In The Semantic Web*
- *Specifying The Exceptions That A Method Can Throw*
- *Working With Cbow Embeddings*
- *Implementing An Lstm Model*
- *List Of Symbols*
- *Markov Models*
- *Training A Siamese Similarity Measure*
- *Conditioning*
- *Installation And Setup*
- *Multithreaded Algorithms*
- *All-pairs Shortest Paths*
- *Total Probability Theorem And Bayes' Rule*
- *Blank Nodes*
- *The Em Algorithm In General*
- *The Control Systems Engineer*
- *Linear Algebra*
- *Creating And Using Packages*
- *Software Development Tools*
- *Distributing Data Across The Web*
- *Creating A Table*
- *Convolutional Neural Networks*
- *Writing Fast Numpy Functions With Numba*
- *Explore The Data*
- *Understanding Computer Files*
- *The Continuous Bayes' Rule*
- *Analysis And Design Of Feedback Systems*
- *Where Are The Smarts?*
- *Distributing Tensorflow Across Devices And Servers*
- *Clustering Using K-means*
- *Hidden Markov Models*
- *Greedy Algorithms*
- *Learning The Tensorflow Way Of Linear Regression*
- *Gain Margin And Phase Margin Via The Nyquist Diagram*
- *The Z-transform*
- *Symbolic Differentiation*
- *Structured And Record Arrays*

- *Implementing A One-layer Neural Network*
- *Block Diagrams*
- *Set Complement*
- *Linear Programming*
- *Moving Window Functions*
- *Using Null To Find Rows With Missing Values*
- *Derived Distributions*
- *Expressivity In Modeling*
- *Overriding Superclass Methods*
- *Decision Theory*
- *Advanced Array Input And Output*
- *Probabilistic Models*
- *Movielens 1m Dataset*
- *Understanding Composition And Nested Classes*
- *Interfacing Between Pandas And Model Code*
- *Growth Of Functions*
- *Understanding Javafx Structure: Stage*
- *Scene*
- *Panes*
- *And Widgets*
- *Binary Hypothesis Testing*
- *Higher-order Relationships*
- *Maximum Margin Classifiers*
- *Analysis And Design Objectives*
- *Rdf As A Tell-and-ask System*
- *Approximation Algorithms*
- *Bayesian Inference And The Posterior Distribution*
- *Quicksort*
- *Using Shortcut Arithmetic Operators*
- *Hopfield Networks*
- *Exponential Family Distributions*
- *Writing Records To A Random Access Data File*
- *Data Aggregation*
- *Forward-mode Autodiff*
- *Functions Of Random Variables*
- *Limitations Of Fixed Basis Functions*
- *Implementing A Simpler Cnn*
- *Modeling For Human Communication*
- *Prerequisites*
- *Manual Differentiation*
- *Basic Sampling Algorithms*
- *Working With A Genetic Algorithm*
- *Working With A Linear Svm*
- *Classification Of States*
- *The Machine Learning Landscape*