UNIVERSIDADE DE ÉVORA

Universidade de Évora

Departamento de Informática

Educational Data Mining Applied to Moodle Data From the University of Évora

Pedro Miguel Lúcio Melgueira

Orientação *Lígia Maria Rodrigues da Silva Ferreira*

Mestrado Engenharia Informática

Dissertação

August 6, 2017

Esta dissertação inclui as críticas e sugestões feitas pelo Júri

UNIVERSIDADE DE ÉVORA

Universidade de Évora

Departamento de Informática

Educational Data Mining Applied to Moodle Data From the University of Évora

Pedro Miguel Lúcio Melgueira

Orientação *Lígia Maria Rodrigues da Silva Ferreira*

Mestrado Engenharia Informática

Dissertação

August 6, 2017

Esta dissertação inclui as críticas e sugestões feitas pelo Júri

To My Family

Acknowledgements

First, I want to thank my thesis advisor, Professor Lígia Ferreira, for first providing me with a theme for the thesis and for overseeing the development of this project. I also want to thank the University of Évora for making this project possible.

I also thank my family who supported me unconditionally.

Lastly, I thank my friends and colleagues who accompanied me during the development of this project, and who talked and challenged me about this research.

Contents

List of Figures

List of Tables

List of Acronyms

crud Create, Read, Update, and Delete

dm Data Mining

edm Educational Data Mining

lms Learning Management Systems

Sumário

Data Mining Educacional Aplicado aos Dados do Moodle da Universidade de Évora

E-Learning tem vindo a ganhar popularidade como forma de transmissão de conhecimentos a nível educacional graças aos avanços nas tecnologias, como por exemplo, a Internet. Instituições como universidades e empresas têm vindo a usar E-Learning para a transmissão de conteúdos educacionais para locais remotos estendendo o seu alcance a estudantes e colaboradores que estão fisicamente distantes.

Sistemas chamados "Learning Management Systems", como o Moodle, existem para organizar E-Learning. Eles oferecem plataformas online onde professores e educadores podem publicar conteúdo, organizar actividades, fazer avaliações, e outro tipo de ações relacionadas, de modo a que os estudantes possam aprender e serem avaliados.

Estes sistemas geram e guardam muitos dados relacionados não só com o seu uso, mas também relacionados com notas de estudantes. Este tipo de dados são frequentemente chamados de Dados Educacionais. Métodos de Data Mining são aplicados a estes dados de modo a fazer suposições não triviais. As técnicas aplicadas tomam inspiração de projectos semelhantes no campo da Data Mining Educacional. Este campo consiste na aplicação de métodos de Data Mining a Dados Educacionais.

Neste projecto, um repositório de dados do Moodle da Universidade de Évora foi explorado. Técnicas de aprendizagem supervisionada são aplicadas aos dados de modo a mostrar como é possível prever o sucesso de estudantes a partir do seu uso do Moodle. Métodos de aprendizagem não supervisionada são também aplicados de modo a mostrar como há divisões nos dados.

Palavras chave: Datamining, Big Data, LMS, Classificação, Árvores de Decisão

Abstract

E-Learning has been rising in popularity as a way to deliver training due to the advancements of technologies, like the Internet. Institutions such as universities and companies have been making use of E-Learning to deliver training to remote locations extending their reach to students and employees who are physically distant.

Systems called Learning Management Systems, like Moodle, exist to organize E-Learning. They provide online platforms where professors and educators can publish content, organize activities, perform evaluations, and so on, in order for students to learn and get evaluated.

These systems generate and store lots of data regarding not only their usage, but also regarding the grades of students. This kind of data is often referred too as Educational Data. Data Mining techniques are applied to this data in order to make non trivial assumptions. The techniques applied take inspiration from similar projects within the field of Educational Data Mining. This field consists in applying Data Mining Techniques to Education Data.

In this project, a data repository from the Moodle of the University of Évora is explored. Supervised learning techniques are applied to this data in order to show how it is possible to make predictions about the success of students based on their usage of Moodle. Unsupervised learning techniques are also applied in order to show how data is divisible.

Keywords: Datamining, Big Data, LMS, Classification, Decision Trees

1

Introduction

In the last few years E-Learning has been rising in popularity as a way to deliver education and training in a remote setting. Institutions such as universities and companies have been taking advantage of E-Learning in different ways. Universities have been developing remote learning courses in order to reach more people, and companies have been developing remote training solutions to reach people across different physical locations.

This rise in popularity of E-Learning is due to the some factors. The advancements of technology, such as the Internet, have not only contributed to the development of better and more complex systems, but have also made possible for those systems to reach more people and obtain greater adoption from users and institutions.

These E-Learning platforms are called Learning Management Systems (LMS). The platforms allow institutions not only to deliver training but also keep track of users, courses, content, etc. In their normal operations, theses platforms generate and store a lot of data. This data concerns the courses, the content in those courses, activities from users, and so on. In some specific cases, like Moodle in universities, the system may even keep track of the student's grades.

The data becomes complex because these systems store a lot of information regarding the users, courses, access logs, etc. Analysing this data, beyond performing some statistical analysis, is not a trivial task. Some Data Mining techniques may be applied in order to make less trivial assumptions.

The field of study which deals with applying Data Mining (DM) techniques to educational data is called Educational Data Mining (EDM). This projects applies EDM to data from the Moodle of the University of Évora. [17, 11, 19, 8, 5]

In this document we start by exploring the background of DM and, specifically, EDM in section 2. The background section also talks about some LMS and the tools used in this project.

Before applying those techniques, we explore the original datasets of the Moodle provided for this specific project. Section 3 talks about the original datasets, how they were explored, and how they were preprocessed into a data repository for use in this project.

Some statistical analysis are made to the preprocessed data repository before any DM techniques are applied. This is done in order to understand the contents of the data and know which DM techniques should be applied and to which subset of the data repository those techniques should be applied to. Section 4 talks about these statistics and draws some conclusions from them.

With the work presented here, it is shown how the data from the University of Évora is used in order for known EDM techniques to be applied.

Section 5 talks about the Data Mining techniques which are applied to the data and also draws some conclusions from them. Those techniques reference the statistics made in the previous section.

Finally, section 6 draws some conclusions.

2

Background

2.1 E-Learning and Learning Management Systems

Recently, E-Learning has been rising in popularity as a solution for the delivery of training. Institutions such as universities and companies have successfully implemented E-Learning to bring course content and evaluation to students, and to deliver training to employees remotely. This has been made possible by the advances in technology and the increasing use of the Internet.

E-Learning is usually provided over the Internet using platforms called LMS. Examples of LMS include Moodle, Blackboard Learn, D2L, among others. These platforms organize users by separating them into roles such as Professors and Students or Authors and Learners. The Professors or Authors are the users who organize courses in universities or training in companies, create and publish content, perform evaluation of students, etc. Students or Learners are the users who actually perform the training and follow courses in the platforms.

A typical LMS contains various data entries which are stored in a database. Information on users, such as their names, usage logs, roles, forum messages, are generated by the systems and kept in their databases.

Resources and activities, as well as the logs detailing their usage, are also kept in a database. The usage of these systems by professors and students generates entries. [17, 11, 19]

In the specific case of Moodle, content is divided into courses. Each Moodle course represents a course from a University or similar learning institution.

The contents of a course may be resources such as PDF files, links, media such as images, videos, etc. These resources should be guides for students to achieve their objectives in learning, or should be resources needed for the student to perform certain tasks. A course may also have activities. Activities may be, for example, projects that a student needs to complete before a given deadline. Activities may also be quizzes made to students over Moodle. These quizzes may be used for evaluation.

Course also have forums, in which professors and students can exchange messages which are relevant to the topics at hand.

A course is overseen by a group of Professors who will publish resources and create activities. Students will use the resources and perform the activities. [8, 5]

2.2 Data Mining

Current computational systems produce and store large amounts of data. This data may represent anything, such as the sales records in a large commercial chain, the usage telemetry on a popular web service, or even the registers of some scientific survey. Data is usually rich in features. For example, the data from the sales records would contain information on which items were sold and in what quantity, to whom they were sold, on what date, and so on.

Data Mining aims at finding patterns in data by looking at the different features of the data entries. These patterns allows us to make non trivial assumptions in data. [18]

Data is usually stored in databases which may follow different paradigms and have different architectures. But generally, we refer to data entries as datasets. We also refer to a group of datasets as a data repository.

To find patterns in data, models are trained either using supervised or unsupervised learning. The following sections describe these concepts and introduce models from both.

2.2.1 Supervised Learning

Supervised learning is done when data made up of a collection of labeled objects which have a list of features and a label, usually called class. The idea is that a model is trained using instances from labeled data in order to correctly classify unlabeled instances later.

There exist several supervised learning models, but in this project we use only two. They are Decision Trees and Naive Bayes Classifiers. [10, 18]

Decision Trees

Decision trees are a model that make use of a tree to classify an object. The model works by sorting an object down the tree, taking a single feature into account on each node of the tree. The branches on each node are the possible values of the feature being taken into account on that node. The leafs of the tree

assign a class to the object.

Decision trees are generally best suited for situations where the data has discrete values. This is the case in this project, since many features are either integer numbers with a limited range of values, or strings and enumerated types which are nominal in nature.

There are a few algorithms used to train these model. They are ID3, C4.5 which is an optimization of ID3, and CART. [10, 18]

Naive Bayes Classifiers

Naive Bayes Classifiers are probabilistic models which are based on Bayesian Networks. A Bayesian Network is a graph that describes directional the dependencies between features of objects. Those features are described as random variables.

The models that make up a Naive Bayes Classifier are a particular case of Bayesian Networks. In these models there is a single node of the network which is the class of the object. Every feature depends directly on the class and only on the class.

Each model essentially describes a conditional probability distribution like:

$$P(\, x_1, \, \ldots, \, x_n \mid C), \tag{2.1}$$

where C is the class of the object and $x_1, \, \ldots, \, x_n$ are the features of it. The model is able to classify an object by directly applying the Bayes Theorem to the distribution, obtaining a value a C.

There are different types of Naive Bayes Classifier. Each type is adequate for different types of features. For example, Guassian Naive Bayes Classifier describe relations between variables, like equation 2.1, as normal distributions. Bernoulli Naive Bayes Classifier are used if the features are boolean.

Multinomial Naive Bayes Classifier are used when the data has discrete values and is nominal in nature. This model is used in this project because the data it deals with is discrete and nominal. [10, 18]

2.2.2 Unsupervised Learning

Unsupervised learning deals with data that is unlabeled. One approach to unsupervised learning is clustering. The objective of clustering is to find groups of unlabeled objects. A group would ideally have objects which features are similar to objects within that group, and dissimilar to objects in other groups.

One difficulty of clustering is to find the number of clusters that optimally divide the data. Some algorithms take a number of clusters as an argument, while others are able to find an optimal value.

By its own nature, it may not be possible to find clusters in data, that is, there may not be any great division of objects given their features in a given dataset. After executing clustering algorithms a performance measure is used to determine if that division is good or not.

In this project, the performance measure used is called Silhouette Score. Silhouette score yields a value between -1 a 1. The interpretation given to this value is that higher values mean that objects are correctly placed in their clusters, lower values mean the objects are in the wrong clusters. Values near zero indicate that the clusters overlap. [10, 18]

K-Means

K-Means is a clustering algorithm that finds K clusters in data. The K variable is given to the algorithm as an argument, hence the name K-Means. The algorithm works by creating K random points based on the features of the data.

Each point has has many coordinates as there are features, and the domains of the coordinates correspond to the domains of the features. Each object in the data gets associated with the closets point to it. A set of objects associated with the same point is called a cluster.

The algorithm then iterates updating the location of those points by calculating the center of of the cluster until convergence is reached. [10, 18]

Affinity Propagation

Affinity Propagation is another clustering algorithm, but this one does not take a number of clusters as an argument. This algorithm finds the optimal number of clusters.

To find clusters, the algorithm keeps two matrices, a responsibility matrix and an availability matrix. Each matrix relates each point in the data to every other point. Each step of the algorithm iterates by updating the information on these matrices until there is convergence.

[10, 18]

2.2.3 Cross Validation

The model used in this project and described in this section always go through two processes, training and testing. To do this, two sets of data are needed, one for each process. Data for the two processes is usually taken from the same dataset in different proportions. For example, some projects may take 75% or 90% of the data to do training, and use the remainder to test.

Testing a model with different objects than the ones used in training is very important. This is because a model, while training, will learn the very specific details of that data and may over fit. When over fitted, a model will not work for general examples of the data for which it was trained and will only "remember" training data instead.

A way of dividing a dataset into training and testing sets is to do Cross Validation. Cross Validation divides the data into N folds. From the original N folds, a single one is taken for testing and the model is trained with the remaining $N - 1$ folds.

A model is trained and tested N times, each time with a different fold taken for testing. Like this we get N different executions for the same dataset and same model and we are able to compare the best, worst, and average scenario in terms of training and testing. [18]

2.3 Educational Data Mining

Educational Data Mining (EDM), refers to the act of applying Data Mining techniques to educational data. Educational data refers to data which origin is in the act of teaching and training, in the activities of professors and students, results of evaluation, and so on. In this project, the origin of data is the University

of Évora's Moodle.

The main objective of EDM is to extract meaningful information from educational data like web logs. It is not trivial to do so without advanced Data Mining techniques. EDM, therefore, aims at training models based on education data for a variety of tasks.

One such example of an EDM task is to predict student's grades based features from the students themselves, the courses they are undertaking, and their activity in a LMS. By training such models, professors can make decisions about how courses are organized regarding students success. For example, a certain number of activities in a given course may be beneficial to its success because it keeps students engaged, but any number of activities more might not make a difference. [15, 7, 9]

These models also detect students which may have greater difficulty completing certain courses, or be used to predict how many students will complete a certain class. [13, 14].

In [15], some models are trained in order to predict student's grades. The tasks described in the paper are classification tasks, in which a student is assigned to a given class corresponding to their success in courses based on their usage of a LMS platform. The classes available are Excellent, Good, Pass, and Fail.

The dataset used in [15] contained the fields in table 2.1 and was made from registers of 438 students in 7 different courses.

Field name	Description
`course`	Identification number of the course.
`n_assigment`	Number of assignments done.
`n_quiz`	Number of quizzes taken.
`n_quiz_a`	Number of quizzes passed.
`n_quiz_s`	Number of quizzes failed.
`n_posts`	Number of messages sent to the forum.
`n_read`	Number or messages read on the forum.
`total_time_assignment`	Total time used on assignments.
`total_time_quiz`	Total time used on quizzes.
`total_time_forum`	Total time used on forum.
`mark`	Final mark the student obtained in the course.

Table 2.1: Features in experiment [15] and [12].

The used models are Statistical Classifiers, Decision Trees, Association Rules, and Neural Networks. After training, performance measures are calculated using test data. The best models were found to be decision trees according to the calculated global percentage of correctly classified students.

The same experiment described in [15] is also described in [12] with the same models, data, and results.

Similar models to before are used in [9]. It is observed again that Decision Trees have the best performance. Table 2.2 shows the data from [9]. The experiments were made on data from 824 students in 11 courses.

Neural networks are used in [16] and [6] for similar purposes. In [16] a neural network is used on a dataset with 116 entries. Each entry is made from 25 features and represents the results of placements tests made to students. There are 5 outputs to the network, one for each possible grade. The paper claims correctly classified rates of over 90% in some trained networks while most are just above 80%. This shows how neural networks can be used in this type of problems.

In [6], networks are trained from Moodle logs. The data from the logs is structured in a slightly different

Field name	Description
UserName	Name of User
CourseName	Name of the Course
ResourceView	Number of Coursware and Other Supporting Materials Views
VirtualClassroom	Number of Virtual Classroom Participations
ArchiveView	Number of Archive Views
ForumRead	Number of Forum Reads
ForumPost	Number of Forum Posts
DiscussionRead	Number of Discussion Reads
DiscussionPost	Number of Discussion Responses
AssignmentView	Number of Assignments Views
AssignmentUpload	Number of Assignment Answer Uploads
FinalGrade	Final Grades

Table 2.2: Features in experiment [9].

way than the previous studies. The features of this data are shown in table 2.3.

Description
Full name of the student.
Number of times that has been officially registered in the subject.
Number of examination sessions.
Mark in numerical format.
Total of accesses (of any type) made to the Moodle system during course 2005/2006.
Total of accesses to "resource view"
Percentage of "resource view" accesses from the total accesses.
Number of different "resource view"of each type (theoretical, examples, etc.) have been visited.
Percentage of the resources of each type (theoretical, examples, etc.) sights.
Segmentation of the number of accesses in every month of the year.
Segmentation per month of the percentage of accesses.

Table 2.3: Features in experiment [6].

The data is extracted from 240 students. The number of courses is not specified. Some trained networks are able to achieve 80% of correctly classified instances, while most achieve a value over 70%.

Beyond the classification tasks, clustering may be used in educational data. The overall objective of clustering is to find groups of similar objects within data. In the context of EDM that may mean looking for groups of students with similar characteristics without their Moodle usage habits. [13, 14]

2.4 Data Mining Tools

The practical component of this project was developed entirely using Python and some Python libraries. The version of CPython used was version 3.5.2. The python libraries Pandas, version 0.19.2, was used for statistical analysis, exploration, and preprocessing. The library SciKit-Learn, version 0.18.1, was used to perform Data Mining tasks. SciKit-Learn uses Numpy and Scikit, in versions 1.11.3 and 0.18.1 respectively. To output some graphs, the python library Mat Plot Lib, version 1.5.4, was used.

Decision trees and Bayesian Networks were used in this project. The algorithm to train decision trees,

implemented in the SciKit-Learn library, is an optimized version of the CART algorithm [4]. Bayesian Networks are used in Naive Bayes tasks. The algorithm to train such models, also implemented in SciKit-Learn, is Multinomial Naive Bayes [3].

Clustering is done using K-Means [2] and Affinity Propagation [1]. Both algorithms are implemented in SciKit-Learn.

3

University of Évora's Moodle Data

3.1 Data Exploration

To begin with, we explore the data which was initially provided. The repository is a spreadsheet book with 12 sheets. The sheets contain raw data and some calculated fields.

In order to explore the data using the bash tools (cut, grep, etc) and tools like Python with Pandas, they were exported to a CSV format. Some transformations were applied to these files in order to clean useless data points or make the data easier to process.

The provided sheets contain information of courses, students, their activities in Moodle, and their results. The data comes from Moodle databases and from the SIIUE system. SIIUE is the integrated information system of the university. It is the system which contains information on courses, students, their grades, among many other things. Both systems are integrated, making Moodle plus SIIUE the University's own LMS.

The sheets are the following:

- UCs_SIIUE_Moodle.csv
- complete_rate.csv
- completed.csv
- Notas UCS SIIUE.csv
- Profile
- ResultadosUCMoodle
- Correlação
- Alunos SIIUE E-L
- AlunosMoodle
- export_weekly
- export_activity
- CalculosAlunos.csv

Some sheets contained information that was the result of calculations from the established data. Analysis like these were removed from the data for two reasons. First, some analysis are not useful so them being in the data only make it more complex. By removing such analysis we get a simpler data model. Second, even if some analysis are in fact useful, they are still easily reproduced by the Python tools, with the difference that they are only reproduced when needed for specific experiments.

The sheets that remained after removing analysis data are:

- UCs_SIIUE_Moodle.csv
- Notas UCS SIIUE.csv
- Profile
- ResultadosUCMoodle
- Alunos SIIUE E-L
- AlunosMoodle

The rest of the sheets contains the objects which are going to be used in the Data Mining tasks. Those objects and their features are fairly unorganized in the original data, so an analysis of the available features had to be done. Section 3.2 states the objects found in the data and details their features. Section 3.3 shows how the final dataset is organized.

3.2 Objects and Features

After a first analysis of the data, the objects are identified. These objects are courses, Moodle users, results, profile, and Moodle logs. Data mining tasks are applied over these objects which are described by their features.

This section specifies objects in this data repository and states their features.

3.2.1 Courses

In the data, a course is identified by one of three unique ids. They are, the Course Id, SIIUE's Course's Code, and Moodle Course's Code. The Course Id is simply a unique positive integer number which is probably assigned as an incremental value when the courses where created in some database.

The two course's code come from their register in the SIIUE system and in the Moodle system. For the most part they are the same, when they are not the same the differences are minimal.

Table 3.1 shows an example of the three ids. Notice how `CourseCodeMoodle` is either the same as `CourseCodeSiiue` or has just minimal differences (in this case there is an underscore followed by a number).

CourseId	CourseCodeSiiue	CourseCodeMoodle
496	ARC10548	ARC10548
1392	GES7182	GES7182_2
79	GES7182	GES7182_4
1535	GES7182	GES7182_3

Table 3.1: Example of course id and codes in the `CoursesGeneral` Dataset.

Each course has the features listed in table 3.2. The table displayed the features and their data type. Each course has a name, a department to which they belong, a cycle (if Master's Degree, Phd, etc), and a degree. These features are all strings.

A course also has a regime, a semester, a season, and a type. These features are enumerated types because they have a limited domain. For example, the semester may either be "*Par*" or "*Ímpar*" (Even or Odd). The type only has one possible value which is `Normal`. Despite this, the field was kept in the data for completeness. Table 3.3 shows the domain of each enumerated type.

Finally, the credits are the only field which is a number.

Feature	Data Type
Department	String
Cycle	String
CourseName	String
Regime	Enumerated
Credits	Number
Degree	String
Semestre	Enumerated
Season	Enumerated
Type	Enumerated

Table 3.2: Data types of features of Courses as they appear in the `CoursesGeneral` dataset.

Some courses have no Moodle and, as such, have no other associated information. They are therefore useless for these experiments and were simply removed.

3.2.2 Moodle Users - Students, and Professors

The logs contain the actions of the Moodle users which role can be of a professor or a student. The data contains features for students but no features for professors. Table 3.4, contains all the features of

Regime	{ A, O, S }
Semester	{ *Par*, *Ímpar*, n/a }
Season	{ *Normal*, *Especial* }
Type	{ *Normal* }

Table 3.3: Values for enumerated types on courses.

students.

Feature	Data Type
StudentsId	Number
StudentsNumber	Number
MoodleUsername	String
Name	String
EnrollementCount	Number
StudentType	Enumerated

Table 3.4: Student features.

From table 3.4 it is observable that a student may be identified by its internal Moodle Student ID, or by its student number as given by SIIUE. Some students in this data repository do not have a *StudentsNumber* value, but those students also don't have any activity in the logs.

The Students Number has a leading character indicating the cycle of that student. For example, a Master's Degree student has a number of the form *m12345*, starting with an *m*, and a PhD student has a number *p12345* starting with a *p*. In the data, the values for *StudentsNumber* are typed as number and only contain the number part. But the field *MoodleUsername* has the complete student's number.

The Name field contains the actual name of the student. This information is useless for any analysis, but the name was kept for completeness.

The values for the *StudentType* field are *Normal* or *Interno* (Normal or Internal).

Professors have only the features listed in table 3.5. The data contains a field called *Role* which states if that Moodle user is a professor or a student. If the user is a professor the role has a value of 5. If it is a student, it has a value of 3. A few users with a value of 4 are present. These users have the role of editor, but the Moodle logs only have 10 entries for a user of this role, so they are simply discarded.

Feature	Data Type
MoodleUsername	String
Name	String
Role	Enumerated

Table 3.5: Professor features.

3.2.3 Results

The data repository contains the results of students in the courses where they participated. A listing of grades students had in particular courses is available. Each object of that dataset, referred to as a Result, has the features listing in table 3.6. The values for the enumerated types are in table 3.7.

Feature	Data Type
CourseCodeSiiue	String
StudentsNumber	Number
Grade	Number
Results	Enumerated
FinalResult	Enumerated

Table 3.6: Results features.

Results	{ *Anulado*, *Aprovado*, *Desistiu*, *Faltou*, *Reprovado* }
FinalResult	{ N, S, n/a }

Table 3.7: Values for enumerated types on results.

A student has a grade if he was approved (Results field has a value of *Aprovado*). The value for *FinalResult* is *S* if and only if the value of *Results* is *Aprovado*. Else, it will be *N*.

The fields *Results* and *FinalResult* are always empty together and in those cases there is never any grade.

3.2.4 Profile

Profile objects are objects that, for each pair of course and student, state the following indicators:

- `NumberOfResources`
- `NumberOfActivities`
- `NumberOfViews`
- `NumberOfSubmissions`

A resource in this context is any file or link that a professor may have placed in a Moodle page. Profile objects state the number of resources which were seen by a particular student in a particular course.

An activity is any interaction in a Moodle forum or quizzed. A view is a view of any page, resource, etc. A submission is when a students submits a project.

There are a total of 331 entries in the Profile dataset.

3.2.5 Moodle Logs

Moodle logs contain the actual activities of Moodle users. Each log entry states a user who performed a Create, Read, Update, and Delete (CRUD) action in a course page. The actions are aggregated per week. Table 3.8 shows the features of the Moodle logs, and table 3.9 shows the values of the enumerated fields. There are a total of 25206 entries in the Logs.

Feature	Data Type
CourseCodeMoodle	String
Role	Enumerated
MoodleUsername	String
StudentsId	Number
CRUD	Enumerated
Week	Number
NumberOfActivitiesPerWeek	Number
StudentsNumber	Number

Table 3.8: Moodle Logs features.

Role	{ 3, 5 }
CRUD	{ C, R, U, D }

Table 3.9: Values for enumerated types on Moodle Logs.

3.3 Preprocessed Datasets

From the original data we make an initial preprocessing in order to have a consistent repository of datasets ready for further analysis. The preprocessed datasets have consistent names and fields.

The data stored in CSV files from which any statistics are calculated and experiments are executed. The data is also organized following a relational paradigm. [1]

The following sections describe which datasets store the data of the various objects as described in section 3.2. In the end of this section, figure 3.1 shows the relational model of the various datasets.

3.3.1 Courses Datasets

Data on the courses is store in the following dataset files:

- `CoursesEntriesOverview`
- `CoursesGeneral`
- `CoursesNoMoodle`
- `CoursesStudents`

The first dataset contains mostly what is found in the original `ResultadosUCMoodle` dataset. The second dataset contains general information on each course, namely the three codes that identify each course, the course name, regime, credits, etc.

The `CoursesNoMoodle` dataset simply contains a listing of courses that have no moodle. These courses aren't very important since no other information on them is available. But they are still kept in the processed datasets.

[1] Despite the fact that the data is not kept in a Relational Database, having it organized in a relational paradigm simplifies the process of working with it using the Python tools

The CoursesStudents dataset contains one entry for each course. Each entry states the total number of students of that course, the total number of approved students, and the approval ration for that course. This information would easily be calculated from other datasets in this repository, but it is kept here since it was present in the original datasets as a non calculated field.

3.3.2 Moodle Users Datasets

Each Moodle user has a username, an actual name, and a role. That information is stored in MoodleUsers. This dataset contains users who are both professors and students. They are identified by their role.

Students have the fields stated in section 3.2.2, and those are stored in the Students dataset.

3.3.3 Results

Results are stored in the Results dataset. The results consist of the fields stated in table 3.6 in section 3.2.3.

3.3.4 Profiling Datasets

The Profile dataset contains the information stated in section 3.2.4. The fields are shown in table 3.10.

Feature	Data Type
CourseCodeMoodle	String
CourseId	String
StudentsId	Number
NumberOfResources	Number
NumberOfActivities	Number
NumberOfViews	Number
NumberOfSubmissions	Number

Table 3.10: Profile features.

3.3.5 Moodle Logs Dataset

The logs dataset in MoodleLogs contain the entries as described in section 3.2.5.

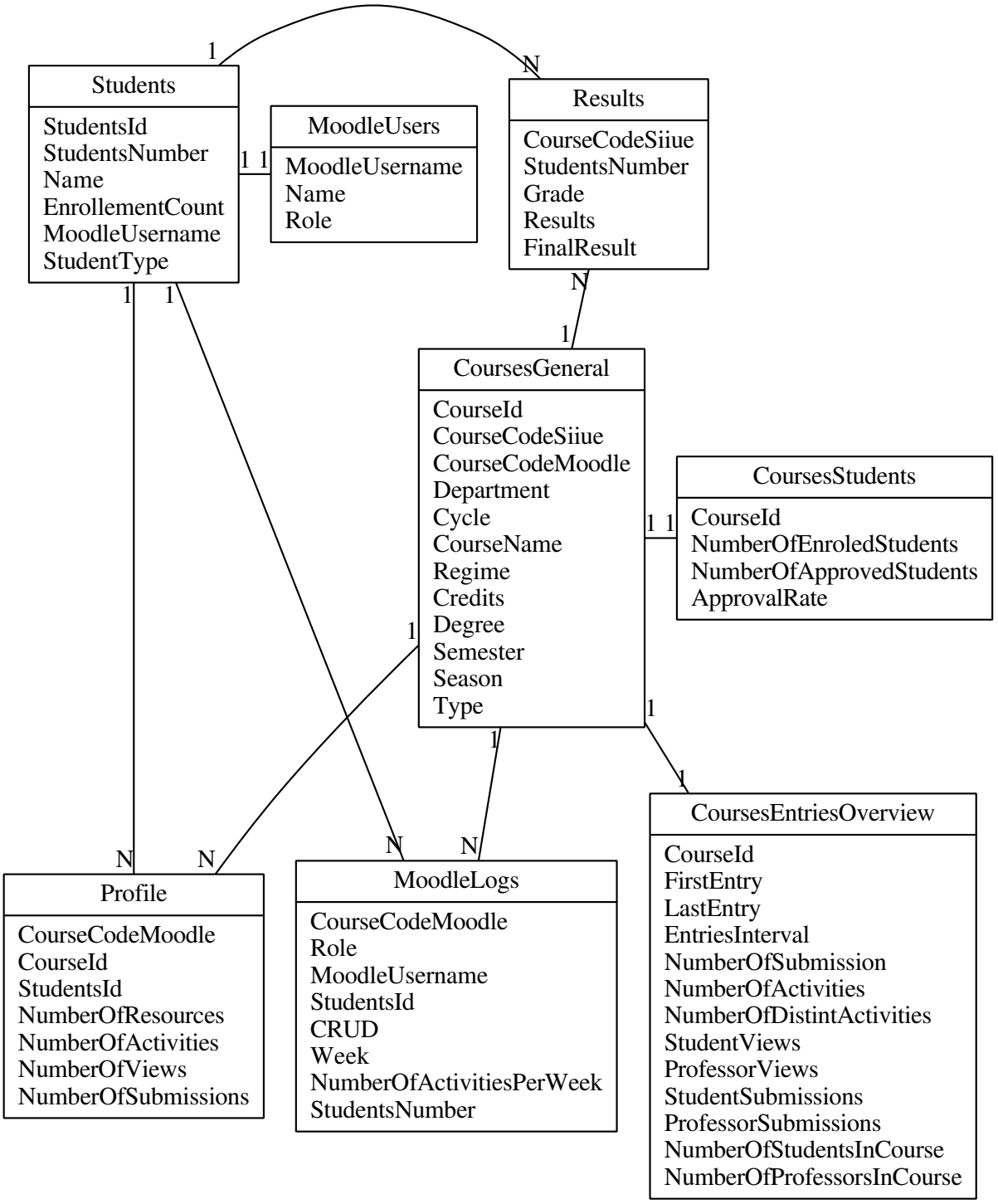

Figure 3.1: Relational model for processed data repository.

4

Statistics on the Data

Before applying any Data Mining methods, it is important to understand how the data is constituted. Knowing things like, how many courses there are, how many students, how many students per course, etc, is important in order to train models and interpret the results.

4.1 Courses

To begin exploring the courses, some counts are made. We count the number of overall courses, the number of departments, and number of degrees. Table 4.1 shows these counts. We see 77 courses which are part of 9 different departments and make up 11 degrees. While the number of courses is much greater than some experiments mentioned in the background, it is important to understand how many students each of these courses have.

If some courses only have one or two students who fail, they might not be useful for training of the model because they won't add any useful information to them. To understand the distribution of students in the courses, we plot the number of students for each course, and the number of approved students. Figure 4.1 shows that plot. Each bar is a course, and courses are sorted by total number of students.

Number of Courses	77
Number of Departments	9
Number of Degrees	11

Table 4.1: Courses in numbers.

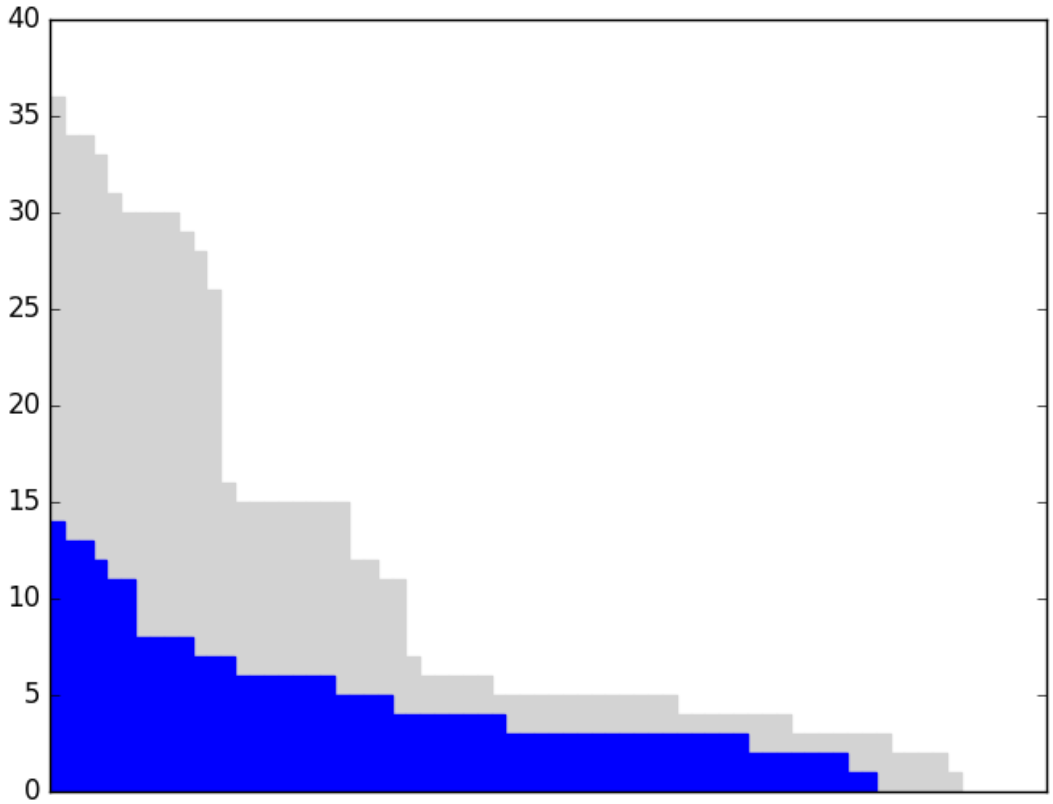

Figure 4.1: Number of Students per Course and Number of Approved Students per Course.

Courses have different values for credits. Table 4.2 shows how many courses are there for each number of credits. We see that the majority of courses has 6 credits.

Credits	Number of Courses
1	3
2	3
3	3
5	6
6	46
7.5	12
12	2
15	2

Table 4.2: Number of courses group by credits.

We take the number of courses for each degree. Table 4.3. The actual name of the degrees is not available, so we can only identify them by their codes. Looking into the counts we see that the degree with code B_M_REST_EINF_E (578) has the majority of courses by far. From this table we also see that some degrees have a low number of degrees and some even have have courses only on one semester.

It was to be expected that every degree would have a similar number of courses, or at least that every degree has courses for both semesters. It is unknown why there are degrees present that have such characteristics, whether it is just missing data or if there really are such degrees.

Degree	Semester	Number of Courses	Total
B_M_EHEA (478)	*Par*	3	7
	Ímpar	4	
B_M_REST_EINF_E (578)	*Par*	14	28
	Ímpar	14	
B_M_REST_TET (457)	*Par*	2	7
	Ímpar	5	
B_PD_M_E (580)	*Par*	3	8
	Ímpar	5	
PG_B_EGNEG (554)	*Par*	4	9
	Ímpar	5	
PG_B_AE (438)	*Par*	5	10
	Ímpar	5	
FC_EDM (564)	*Par*	1	1
CCDND_CPM (344)	*Ímpar*	1	1
FC_CVP_FDOC (513)	*Ímpar*	2	2
FC_CVP_UAM (514)	*Ímpar*	3	3

Table 4.3: Number of Courses per Degree and Semester.

4.2 Students

Table 4.4 shows some counts regarding Moodle users. Some of the explored projects listed in the background, like [15], have a number of students around 800 to 900 which is greater than the number of students in this dataset, but they also have a lower number of courses, for example [15] has 7, where this dataset has 77.

Number of Students	145
Number of Professors	87
Total	232

Table 4.4: Moodle users counts.

Observing the behaviour of students in most courses in any University, it is common to find that the number of students decreases over time as the course progresses. This is due to students giving up on that course for that particular semester for various reasons. Although the data doesn't have any explicit data point stating that a student gave up on a course, we can still observe student activity to verify if it decreases or not as the semester progresses over the weeks.

The activity on a course is measured using the Moodle logs which contain the CRUD activities. CRUD activities, as stated in 3.2.5, are aggregated by courses, students, and weeks. To have a sense of how

activity changes over a course we take the logs dataset and aggregate every object only by course. So, each entry will have the sum of the activities aggregated by course and week.

To get a sense of activities over the semester, we take six courses. Two of the courses are the ones with the greatest number of activities, other two are the ones with a median number of activities, and the last two are courses with the least number of activities.

We take the two courses with the most activity overall and plot a graph which x axis is the weeks and y axis is the number of activities. Figure 4.2 shows the graph for these courses. The number of read activities and number of total activities is much greater than the number of create, update, and delete activities. Because of this, we plot a graph which only contains create, update, and delete activities for the same courses as shown in figure 4.3.

From figures 4.2 and 4.3 we see that the activities of courses remain generally the same during the semester. With the exception of courses with the least activity, it is observable that while there are a few peaks during the semester, there is no significant drop in activities towards the end of them. The examples shown have peaks in various places during the semester, however, they don't tend to be at the start or at the end of the semester.

This leads to the conclusion not only that the levels of activity remain relatively constant during the semester, but also leads to the conclusion that many students don't quit the courses even if they fail.

To get an overall view of how activities change during the whole semester we make another plot. This time, we see the total number of activities for every course during the semester and the mean of activities. Figure 4.4 shows this plot.

The number of total activities per week drops significantly. This is because some courses are longer than others. For example, some courses may have their last evaluation in the first weeks of the exam season, while others may still have ongoing activity after exam season.

The mean of activities remains the same. The mean is calculated for each week only on courses that have any activity on that week. So looking at the last few weeks we get a mean of activities only for courses which are still active. With this in mind, we again see that courses remain relatively the same in terms of activities while the semester is going.

The concept of active course needs to be understood. A course is active from week 0 until the last week for which there is any activity. For example, if the last activity of a course is on week 20, then the course is active from week 0 until week 20.

Figure 4.5 shows a graphs bar with the number of active courses per week, and again shows the total number of activities over these bars. We see that the number of active courses fall correlates with the fall of the total number of activities. This further reinforces the notion that the activity of courses remains relatively the same as the semester moves forward.

4.3 Results

Table 4.5 shows some statistics on the results data. We see that from the available 716 entries, 334 refer to instances of a student getting approved on a course.

As seen in section 3.2.3, the dataset containing the results has the fields *Grade*, *Results*, and *FinalResult* for a combination of course and student. However, some objects have no values for those three fields. When that happens, we interpret the objects as having the student not getting approved on that course.

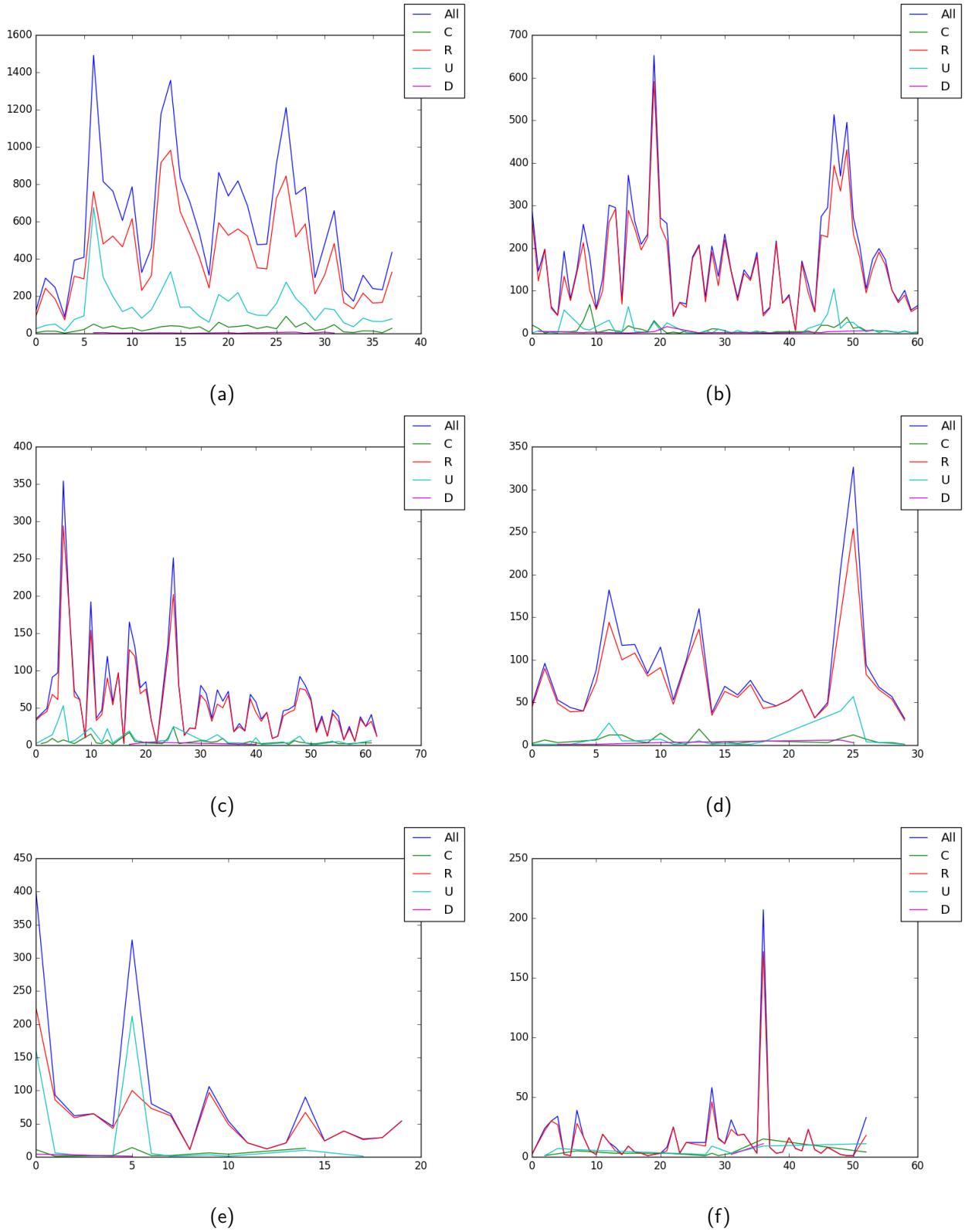

Figure 4.2: Number of CRUD activities for two courses with greater activity (4.2a and 4.2b), median activity (4.2c and 4.2d), and least activity (4.2e and 4.2f).

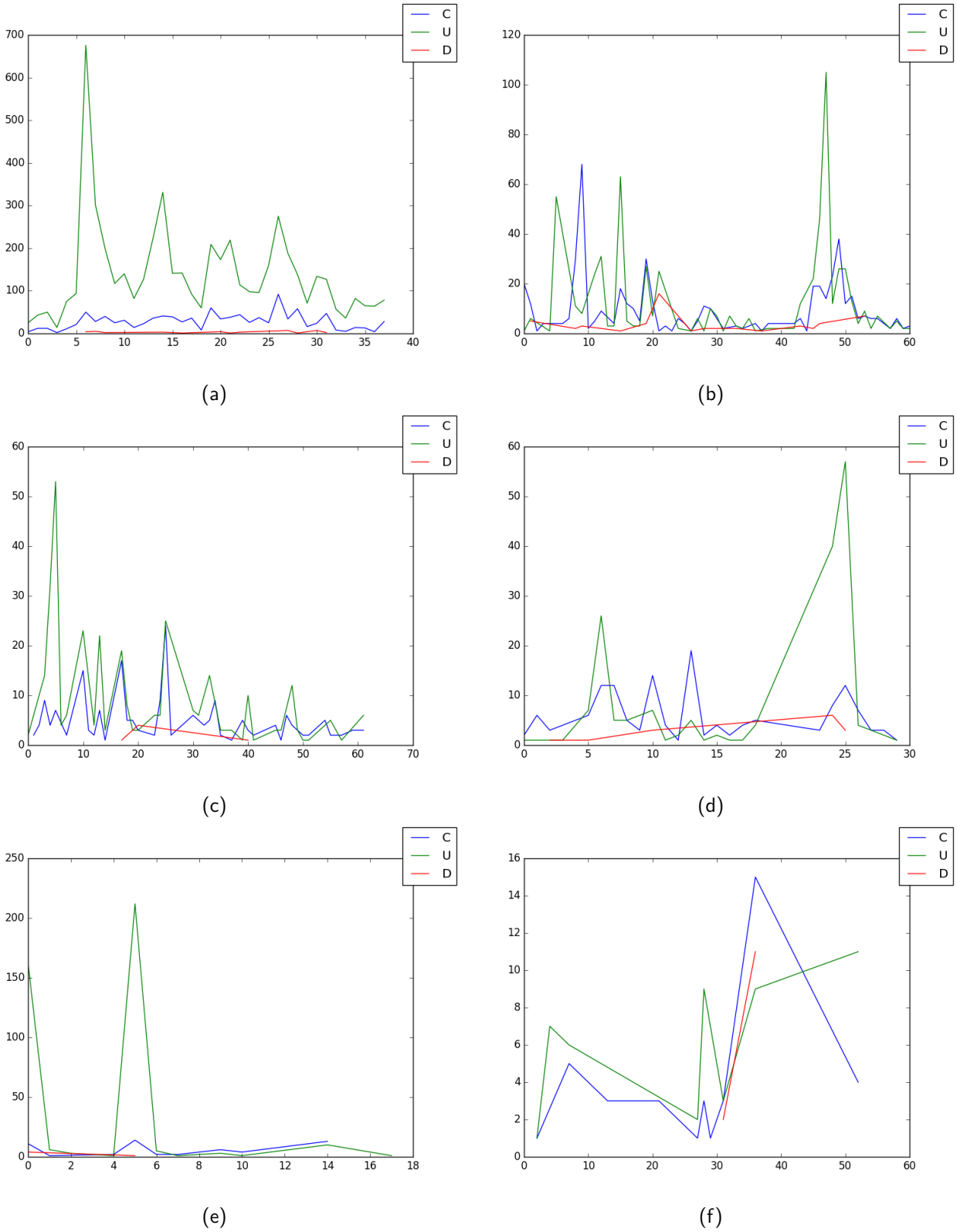

Figure 4.3: Number of Create, Update, and Delete activities for two courses with greater activity (4.3a and 4.3b), median activity (4.3c and 4.3d), and least activity (4.3e and 4.3f).

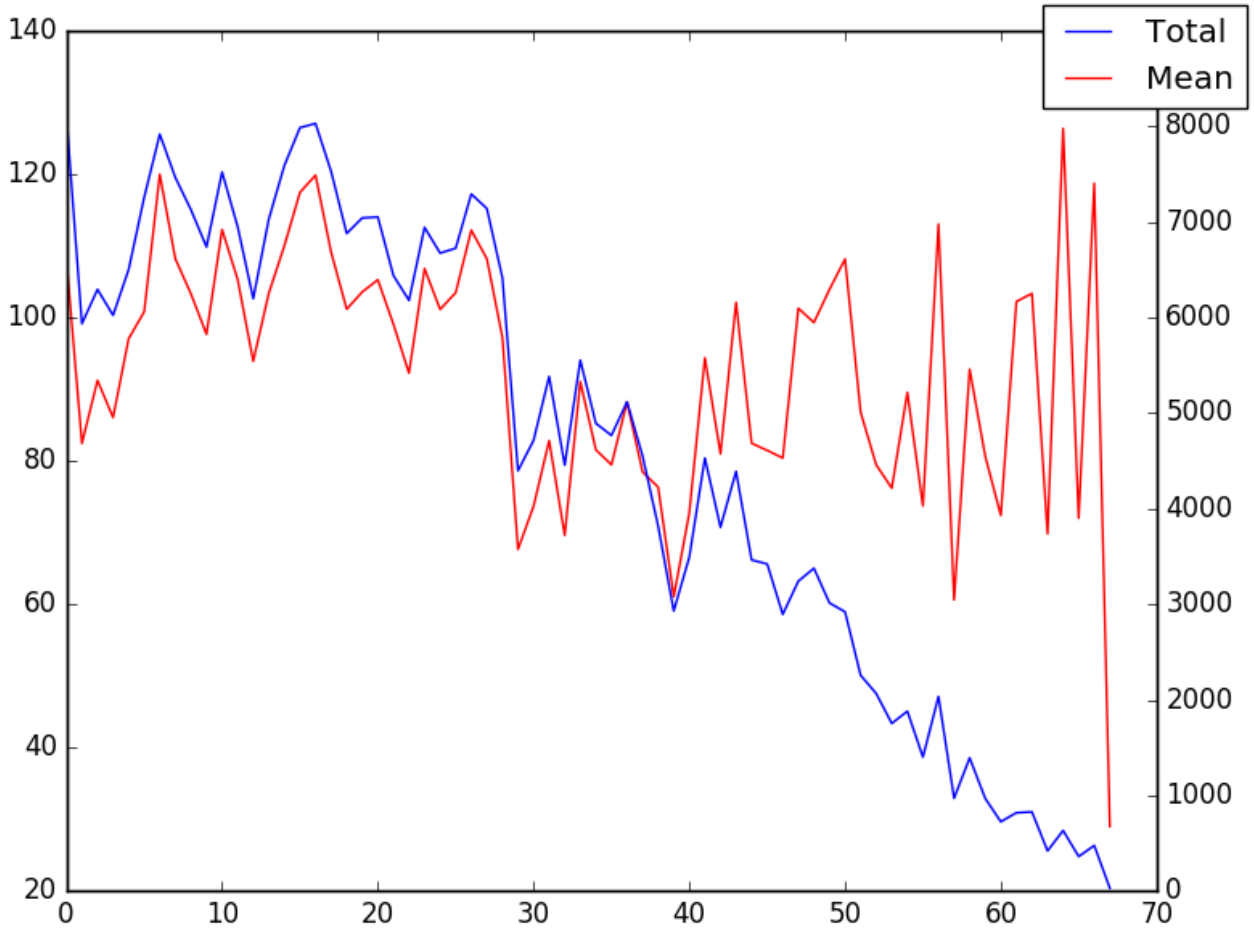

Figure 4.4: Total number of activities (Right) vs mean of activities per week (Left).

From the 716 results, 57 of them are empty.

Number of results	716
Number of non-empty results	659
Number of approved students	334

Table 4.5: Counts on the Results.

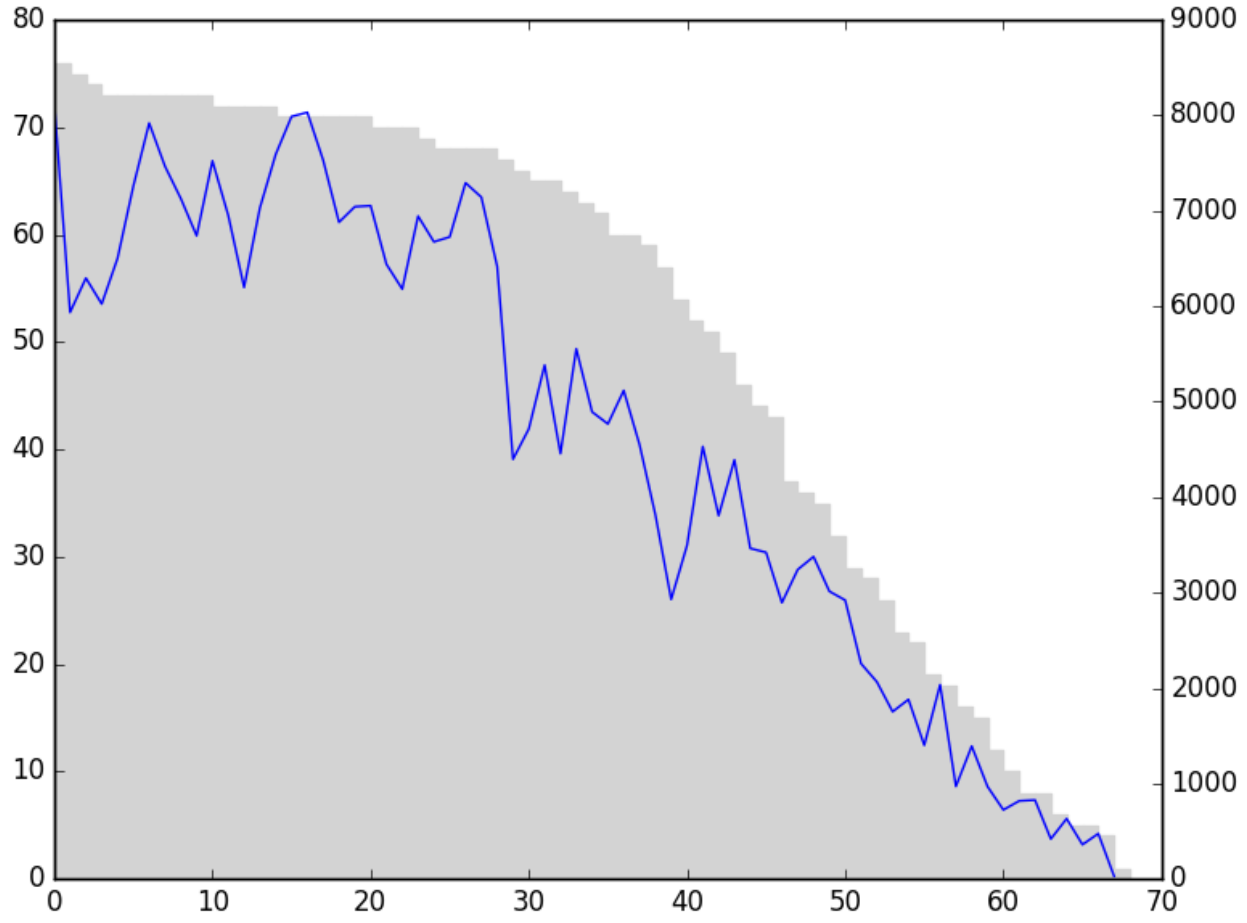

Figure 4.5: Total number of active courses per week (Bars) vs total active courses (Line).

5 Experiments

The Data Mining experiments made in this project consist in organizing the data from the processed datasets described in section 3.3 into a new dataset useful for a set of experiments. This chapter is organized by sections on each dataset. Each dataset is used in several experiments with different algorithms.

5.1 Students, Courses, Logs, and Results

In an initial analysis, we look into how Moodle usage relates with student's success. A dataset is build with the features presented in table 5.1. The new dataset takes information from the following datasets:

- `Students`
- `CoursesGeneral`
- `MoodleLogs`
- `Results`

Each object of the dataset contains information referencing a single student in a single course. The objects contain the overall enrollment count and type of student. They also contain the identification of a course along with the semester and season of that course. Finally, the object contains a count of CRUD activities of that student in that course.

The dataset can be generated with different classes. There are three options. Option one, the class is binary and has the value true if the student was approved on that course, and false otherwise. Option two, the class has one of the following values which are intervals of grades:

$$[0, 10[\ ,\ [10, 12[\ ,\ [12, 14[\ ,\ [14, 16[\ ,\ [16, 18[\ ,\ [18, 20].$$

Option three, the class has one of the following values:

$$[0, 10[\ ,\ [10, 14[\ ,\ [14, 18[\ ,\ [18, 20].$$

With option two, we get a dataset which groups every records with grades below 10, but separates the positive grades in groups of two. Option three does the same but separates grades into groups of four, with the last group having only three values instead of four.

Field	**Source Dataset**	**Type**
`EnrollementCount`	Student	Number
`StudentType`	Student	Enumerated
`CourseCodeSiiue`	CoursesGeneral	String
`Semester`	CoursesGeneral	Enumerated
`Season`	CoursesGeneral	Enumerated
`TotalActsOnCourse`	Calculated From MoodleLogs	Number
`CActsOnCourse`	Calculated From MoodleLogs	Number
`RActsOnCourse`	Calculated From MoodleLogs	Number
`UActsOnCourse`	Calculated From MoodleLogs	Number
`DActsOnCourse`	Calculated From MoodleLogs	Number
`Class`	Calculated From Results	Binary

Table 5.1: Students, Courses, Logs, and Results.

5.1.1 Predicting Student's Grades Based on Moodle Usage

A first experiment on this dataset aims at predicting the student's grades based on their Moodle usage data and in their other characteristics present on this dataset. In this experiment six tasks are made. Both Decision Trees and Naive Bayes Classifiers are trained using the dataset with binary classes, classes with intervals of two, and classes with intervals of four.

Some fields on dataset 5.1 are strings and enumerated types, therefore, these are nominal fields. Because of this characteristic, Decision Trees and Naive Bayes Classifiers are used.

The models are trained using cross-validation with 10 splits. This means that the data is divided into 10 parts. Nine parts are used to train to models and the remaining part is used to test the models.

Training is executed 10 times, each time with different splits for training and testing. The accuracy of

each model is taken. Table 5.2 shows the maximum accuracy of the trained models, the minimum, and the average.

Model	Classes	Maximum	Minimum	Average
Decision Tree	Binary Class	0.92	0.34	0.70
	Grades Classes 4 by 4	0.84	0.22	0.59
	Grades Classes 2 by 2	0.74	0.34	0.55
Naive Bayes	Binary Class	0.95	0.53	0.74
	Grades Classes 4 by 4	0.73	0.36	0.57
	Grades Classes 2 by 2	0.67	0.28	0.52

Table 5.2: Predicting student's grades based on Moodle usage results.

It is observable that for binary classes, Naive Bayes Classifiers performed slightly better than decision trees. But for grade classes two by two and four by four, decision trees were better. In particular, training models for binary classes, understandably, yields better results on average.

These results show that such models can be trained to predict grades of students with success.

5.1.2 Clustering Student's Grades Based on Moodle Usage

With the same dataset, clustering using the K-Means algorithm is performed, however, the class feature is removed. The training and testing method is similar to the previous experiment. Cross validation is used and a model is trained 10 times, each time using nine different folds for training and one fold for testing.

K-Means is trained and tested for a number of clusters between two and five. Testing is done using silhouette score. Like before, the maximum, minimum, and average score is taken. Table 5.3 shows the results.

Number of Clusters	Maximum	Minimum	Average
2	0.72	0.68	0.69
3	0.67	0.65	0.66
4	0.68	0.64	0.66
5	0.67	0.65	0.66

Table 5.3: Clustering Student's Grades Based on Moodle Usage.

In general, the model fails to get good results having a maximum score of 0.72 for two clusters. Numbers greater them five clusters were tried but they, understandably, got lower scores.

To understand what the algorithm is separating, that is, which type of objects are being placed in which cluster, we look in detail to an execution of K-Means with two clusters.

Table in attachment A has a listing of objects that are placed in each of the two clusters. It is observable that the major difference is in the number of CRUD activities. Different values for activities remain the same for different courses and enrollment numbers, but greater values are placed in one cluster while lower values are placed in another.

Besides K-Means, Affinity Propagation was executed with this dataset, but the algorithm performed poorly have scores below 0.5.

5.2 Number of Courses and Logs

In similar experiments to 5.1, we make a dataset in which we relate a students total number of enrolled courses, number of approved courses, and Moodle usage data.

Each object of this dataset will have information on a single student. As mentioned, an object has the number of enrolled and approved courses and the sum of each CRUD activity for every course of that single student. Each entry does not have the identification of the student to which it refers too because that information is irrelevant for our learning objectives.

This dataset only contains features which have a number type. The records come from the `Results` and `MoodleLogs` datasets. Table 5.4 shows the features of this dataset.

Field	Source Dataset	Type
`EnrolledCourses`	Calculated From Results	Number
`ApprovedCourses`	Calculated From Results	Number
`ActSumAll`	Calculated From MoodleLogs	Number
`ActSumC`	Calculated From MoodleLogs	Number
`ActSumR`	Calculated From MoodleLogs	Number
`ActSumU`	Calculated From MoodleLogs	Number
`ActSumD`	Calculated From MoodleLogs	Number

Table 5.4: Number of courses and logs. Each object in this dataset contains the number of enrolled courses, number of approved courses and Moodle usage data for a single student.

Some classification tasks use the `ApprovedCourses` feature as a class. In this case, feature may be left in its original form or it may be replaced by classes. There are three ways to do this replacement.

The maximum number of approved courses is 12 and minimum is 0, so the possible divisions are with two classes:

$$[0, 6[\ ,\ [6, 12],$$

Three classes:

$$[0, 4[\ ,\ [4, 8[\ ,\ [8, 12],$$

And four classes:

$$[0, 3[\ ,\ [3, 6[\ ,\ [6, 9[\ ,\ [9, 12].$$

5.2.1 Predicting Number of Approved Courses Based on Moodle Usage

Like before both Decision Trees and Naive Bayes Classifiers are used. Table 5.5 shows the maximum, minimum, and average accuracy for each pair of model and classes.

The models are trained using cross validation with 10 splits like before.

It is observable that Decision Trees reached an accuracy of 1 producing, no false negatives or false positives

Model	Classes	Maximum	Minimum	Average
Decision Tree	2 Classes	1.00	0.64	0.83
	4 Classes	1.00	0.29	0.68
	6 Classes	1.00	0.21	0.68
	Original Classes	0.86	0.14	0.52
Naive Bayes	2 Classes	1.00	0.47	0.70
	4 Classes	0.87	0.29	0.56
	6 Classes	0.86	0.07	0.43
	Original Classes	0.86	0.00	0.39

Table 5.5: Predicting number of approved courses based on Moodle usage results. These are the results of two models being trained 10 times for 4 different variants of the same dataset. The maximum, minimum, and average accuracy is shown.

in testing, for all datasets except for the one with the original classes. On average, Decision Trees, models achieve high levels of accuracy for every dataset.

Naive Bayes Classifiers are achieve hight level of accuracy, but not as high as Decision Trees. For the dataset with the original classes, Naive Bayes Classifiers perform similarly in the best case scenario, given that the values for maximum accuracy are similar. But the values for minimum and average are noticeably lower.

Because the models trained with the original classes achieve high values for accuracy we conclude that it is possible to predict the exact number of approved courses with great accuracy given only the initial number of enrolled courses and the Moodle usage data, especially if such tasks are executed with Decision Trees.

5.2.2 Clustering Number of Approved Courses Based on Moodle Usage

Clustering methods are applied to look for groups of similar objects within this data. In this experiment, every field is a feature, so the `ApprovedCourses` field is a feature and not a class like before.

Number of Clusters	Maximum	Minimum	Average
2	0.83	0.80	0.81
3	0.78	0.75	0.76
4	0.78	0.57	0.73
5	0.68	0.55	0.61
6	0.57	0.55	0.56

Table 5.6: Clustering number of approved courses based on Moodle usage results. The table displays the results for the silhouette score for clustering. The values for every variant of the dataset are approximately the same, so only one table is shown.

Table 5.6 shows the results of this experiment. The silhouette score shown is approximately the same for each variant of the dataset, so only one table is shown for every variant.

It is observable that for two clusters, the score is around 0.8, and for three clusters the score is around 0.75. This shows how these objects are able to be separated in groups of two or three without having those groups overlapping.

For a number of clusters equal to or greater than four, we start getting scores bellow 0.7.

The results shown were obtained with the K-Means algorithm. Affinity Propagation was also executed, but the results obtained were not good, having scores within 0.0 and 0.5.

5.3 Student/Course Profile and Results

In this dataset we relate the profiling of a student in a course and his results on that course. To begin constructing this dataset we use the Profiling dataset, as shown in section 3.2.4.

The Profiling dataset contains a results field, however, the meaning of each value in this field is unknown. So initially, this field is dropped. To get the actual results of students we need the Results dataset.

Each object of Profile contains the features `CourseCodeMoodle`, `CourseId`, and `StudentsId`. But the Results dataset has `CourseCodeSiiue` and `StudentsNumber`. In order to relate both, the Students dataset and the CoursesGeneral dataset are needed.

These datasets are related and some fields are dropped in order to produce a dataset with the features listed in table 5.7.

Like in dataset 5.1, the class of this dataset may be binary, in which the student is approved or not, or divided in groups of 2 or 4.

Field	Source Dataset	Type
`CourseCodeSiiue`	CoursesGeneral	String
`NumberOfResources`	Profile	Number
`NumberOfActivities`	Profile	Number
`NumberOfViews`	Profile	Number
`NumberOfSubmissions`	Profile	Number
`Class`	Results	Binary

Table 5.7: Student/Course Profile and Results.

5.3.1 Predicting Student's Grades From Student/Course Profiling

Using the described dataset, we train models to predict student's grades based on profiling data. Table 5.8 shows the results of six tasks, three for each Model. Each task is done in a variant of the dataset.

Model	Classes	Maximum	Minimum	Average
Decision Tree	Binary Class	0.91	0.43	0.73
	Grades Classes 4 by 4	0.65	0.31	0.44
	Grades Classes 2 by 2	0.56	0.18	0.34
Naive Bayes	Binary Class	0.97	0.49	0.77
	Grades Classes 4 by 4	0.91	0.31	0.55
	Grades Classes 2 by 2	0.71	0.26	0.46

Table 5.8: Predicting student's grades from student/course profiling results. The results show the maximum, minimum, and average for each trained model and each dataset variant.

We observe from the results that Naive Bayes Classifiers perform better than Decision Trees achieving greater levels of accuracy for each dataset variant.

5.3.2 Clustering Student's Grades From Student/Course Profiling

Taking the same dataset, we remove the field `CourseCodeSiiue` and apply clustering like in previous experiments. Table 5.9 shows the results of this experiment.

Number of Clusters	Maximum	Minimum	Average
2	0.64	0.60	0.63
3	0.58	0.56	0.57
4	0.57	0.53	0.54
5	0.51	0.50	0.50

Table 5.9: Clustering Student's Grades From Student/Course Profiling. The values for every variant of the dataset are approximately the same, so only one table is shown.

Having two clusters we get the higher values for silhouette score, but those values are still far from 1, meaning that the executed algorithms do not find any meaningful separation of objects into groups.

6

Conclusions and Future Work

In this project EDM techniques were applied to the University of Évora's Moodle data. The tasks applied to the data were based on similar projects shown in the background section which also applied EDM techniques to Moodle data of other learning institutions.

In this project it was shown how the original data from the University of Évora's Moodle was preprocessed into a data repository useful for analysis. Some statistics over that data were then shown. The experiments were broken down by dataset. Each dataset was build from the preprocessed original data.

The techniques applied are divided into Supervised and Unsupervised learning. In supervised learning, Decision Trees and Naive Bayes Classifiers were applied. In unsupervised learning, clustering was done using the K-Means and the Affinity Propagation algorithms.

6.1 Supervised Learning Tasks

Supervised learning techniques were applied to the different variants of the three described datasets. For every one of them, Decision Trees and Naive Bayes Classifiers were trained using the cross validation technique described in section 2.2.3.

It was shown how these models were generally able to achieve high values of accuracy. This means that it is possible to:

1. Predict student's grades from their usage of Moodle; (Section 5.1.1)

2. Predict how many courses a student is going to complete given his usage of Moodle and initial number of enrolled courses; (Section 5.2.1)

3. Predict student's grades from his usage of features in Moodle. (Section 5.3.1)

Following cross validation, each model was trained ten times, each training is done with nine folds from the dataset, leaving one fold for testing. Despite having high values for maximum accuracy, and relatively high values for average accuracy, some models show very low values for minimum accuracy. This means that some instances of training have testing folds which are too different from the folds used in training, and therefore, the testing folds may contain outlier objects.

Continuing the analysis in this project, it might be important to understand which are these outlier cases in order to achieve better accuracies in further supervised learning tasks.

6.2 Unsupervised Learning Tasks

Clustering tasks were done and it was shown that some separation of objects exists in the explored datasets. Results for experiment 5.2.2, which are about course completion given the number of enrolled courses and Moodle usage data, were particularly promising because the results show a maximum and average silhouette score above 0.70 for two, three, and four clusters.

Clustering was also done by dividing the data into folds of ten and excluding one fold for each training execution, and then using that fold to calculate the silhouette score. The difference between the maximum and minimum silhouette scores using this division method were noted to be minimal, sometimes not exceeding a value of 0.05.

In these unsupervised learning tasks, the K-Means algorithm was used along with Affinity Propagation. However, only K-Means produced good results for silhouette scores.

From these results, two things can be further done. First, the clustering results may be analysed in order to understand what are the characteristics of each cluster of the more successful tasks. In section 5.1.2, for example, we see that data seems to be divisible into two clusters. An hypothesis for this binary division may be that one cluster contains objects in which students get approved in courses while the other cluster contains the opposite.

In section 5.2.2, like mentioned above, we see good scores for two, three, and four clusters. However, without further analysis, no hypothesis is presented at this time as to what are the reasons we see such good separation of objects in this dataset.

Aside from those analysis, one second thing to do is to use different clustering algorithms. One algorithm to use would be K-Modes, which is a variant of K-Means useful for nominal data, which is present in this data.

6.3 Applications

The results of an EDM study may have many applications. Experiments such as the ones made to dataset 5.1 and 5.3 yield insight into how the basic usage of Moodle is a great indicator of the success of students. From the trained models, future applications may be developed which look into students and their usage of Moodle and finally predict their success even before a student as finished a course. With such applications we can detect student behaviour that might lead to his failure of a course and signal that student to take action to avoid failure.

From the Moodle data described in this project, further classification and clustering analysis may also be done. More datasets may be built and analysed from the original data and more algorithms may be used. For example, a dataset may be built that joins Moodle usage data (as described in sections 5.1 and 5.3) and profiling data (as described in section 5.2). A dataset with all these fields might yield better accuracies in the prediction of grades or more ways to do clustering.

Finally, other types of Data Mining analysis may be done. For example, in the background section, Neural Networks are mentioned to be a usual model in EDM for grade prediction. Association Rule Analysis, which is a topic not discussed in this project, may also be used with this data. Possibly, dependencies between course completion features might be detected.

A

Clustering Student's Grades Based on Moodle Usage Sample

Tables A.1 and A.2 show a sample of 30 objects from the clustering done in experiment 5.1.2. The features are the ones in table 5.1 but without the class.

Table A.1 shows objects placed in cluster 0, and A.2 show objects placed in cluster 1.

14.00	10.00	0.00	451.00	19.00	424.00	8.00	0.00	0.00	1.00
54.00	10.00	0.00	802.00	62.00	717.00	22.00	1.00	0.00	1.00
55.00	10.00	0.00	699.00	32.00	651.00	12.00	4.00	0.00	1.00
51.00	1.00	0.00	1585.00	45.00	1386.00	151.00	3.00	2.00	1.00
51.00	1.00	0.00	1087.00	50.00	828.00	209.00	0.00	2.00	1.00
51.00	1.00	0.00	551.00	25.00	431.00	95.00	0.00	2.00	1.00
51.00	1.00	0.00	1174.00	42.00	911.00	221.00	0.00	2.00	1.00
21.00	11.00	0.00	492.00	49.00	432.00	11.00	0.00	0.00	1.00
10.00	11.00	0.00	992.00	36.00	941.00	15.00	0.00	2.00	1.00
17.00	11.00	0.00	708.00	59.00	642.00	5.00	2.00	0.00	1.00
20.00	11.00	0.00	647.00	54.00	587.00	6.00	0.00	2.00	1.00
51.00	1.00	0.00	535.00	18.00	338.00	179.00	0.00	2.00	1.00
10.00	11.00	0.00	1104.00	27.00	1065.00	12.00	0.00	2.00	1.00
11.00	11.00	0.00	494.00	20.00	459.00	15.00	0.00	2.00	1.00
21.00	11.00	0.00	745.00	41.00	692.00	8.00	4.00	0.00	1.00
10.00	12.00	0.00	782.00	48.00	722.00	12.00	0.00	2.00	1.00
16.00	12.00	0.00	865.00	56.00	800.00	8.00	1.00	2.00	1.00
20.00	12.00	0.00	790.00	59.00	713.00	18.00	0.00	2.00	1.00
21.00	12.00	0.00	1198.00	81.00	1084.00	33.00	0.00	0.00	1.00
42.00	1.00	0.00	446.00	16.00	424.00	6.00	0.00	0.00	1.00
29.00	1.00	0.00	635.00	26.00	490.00	119.00	0.00	2.00	1.00
29.00	1.00	0.00	548.00	28.00	395.00	125.00	0.00	2.00	1.00
29.00	1.00	0.00	753.00	30.00	593.00	130.00	0.00	2.00	1.00
40.00	8.00	1.00	784.00	83.00	671.00	30.00	0.00	2.00	1.00
40.00	2.00	1.00	1341.00	161.00	1163.00	17.00	0.00	2.00	1.00
19.00	7.00	1.00	949.00	41.00	890.00	18.00	0.00	0.00	1.00
30.00	7.00	1.00	2894.00	428.00	2192.00	152.00	122.00	2.00	1.00
31.00	7.00	1.00	1428.00	57.00	1355.00	16.00	0.00	2.00	1.00
32.00	7.00	1.00	1236.00	46.00	1175.00	15.00	0.00	0.00	1.00
33.00	7.00	1.00	1130.00	65.00	1036.00	27.00	2.00	0.00	1.00

Table A.1: Clustering results sample for 5.1.2, cluster 0.

59.00	10.00	0.00	425.00	19.00	393.00	13.00	0.00	0.00	1.00
14.00	10.00	0.00	222.00	20.00	193.00	8.00	1.00	0.00	1.00
58.00	10.00	0.00	253.00	28.00	218.00	7.00	0.00	2.00	1.00
59.00	10.00	0.00	124.00	7.00	112.00	5.00	0.00	0.00	1.00
60.00	10.00	0.00	312.00	43.00	263.00	6.00	0.00	2.00	1.00
62.00	10.00	0.00	206.00	25.00	179.00	2.00	0.00	0.00	1.00
14.00	10.00	0.00	350.00	17.00	323.00	10.00	0.00	0.00	1.00
58.00	10.00	0.00	351.00	28.00	314.00	9.00	0.00	2.00	1.00
59.00	10.00	0.00	346.00	16.00	323.00	7.00	0.00	0.00	1.00
51.00	1.00	0.00	193.00	10.00	165.00	18.00	0.00	2.00	1.00
51.00	1.00	0.00	187.00	5.00	160.00	22.00	0.00	2.00	1.00
51.00	1.00	0.00	322.00	6.00	257.00	59.00	0.00	2.00	1.00
51.00	1.00	0.00	403.00	25.00	316.00	62.00	0.00	2.00	1.00
11.00	11.00	0.00	329.00	10.00	314.00	5.00	0.00	2.00	1.00
17.00	11.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
18.00	11.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
20.00	11.00	0.00	18.00	3.00	15.00	0.00	0.00	2.00	1.00
21.00	11.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
10.00	11.00	0.00	6.00	0.00	6.00	0.00	0.00	2.00	1.00
11.00	11.00	0.00	0.00	0.00	0.00	0.00	0.00	2.00	1.00
12.00	11.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
15.00	11.00	0.00	1.00	0.00	1.00	0.00	0.00	2.00	1.00
16.00	11.00	0.00	0.00	0.00	0.00	0.00	0.00	2.00	1.00
13.00	11.00	0.00	408.00	28.00	372.00	8.00	0.00	1.00	0.00
15.00	11.00	0.00	368.00	16.00	341.00	11.00	0.00	2.00	1.00
18.00	11.00	0.00	295.00	13.00	274.00	8.00	0.00	0.00	1.00
20.00	11.00	0.00	336.00	18.00	318.00	0.00	0.00	2.00	1.00
10.00	11.00	0.00	0.00	0.00	0.00	0.00	0.00	2.00	1.00
11.00	11.00	0.00	0.00	0.00	0.00	0.00	0.00	2.00	1.00
12.00	11.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

Table A.2: Clustering results sample for 5.1.2, cluster 1.

Bibliography

[1] Affinity Propagation SciKit-Learn. http://scikit-learn.org/stable/modules/clustering.html#affinity-propagation, 2017. [Online; accessed February 28, 2017].

[2] K-Means SciKit-Learn. http://scikit-learn.org/stable/modules/clustering.html#k-means, 2017. [Online; accessed February 28, 2017].

[3] Naive Bayes SciKit-Learn. http://scikit-learn.org/stable/modules/naive_bayes.html#multinomial-naive-bayes, 2017. [Online; accessed February 28, 2017].

[4] Tree Algorithms SciKit-Learn. http://scikit-learn.org/stable/modules/tree.html#tree-algorithms, 2017. [Online; accessed February 28, 2017].

[5] Klaus Brandl. Are you ready to "moodle". *Language Learning & Technology*, 9(2):16–23, 2005.

[6] M Delgado Calvo-Flores, E Gibaja Galindo, MC Pegalajar Jiménez, and O Pérez Pineiro. Predicting students' marks from moodle logs using neural network models. *Current Developments in Technology-Assisted Education*, 1:586–590, 2006.

[7] Gwo-Dong Chen, Chen-Chung Liu, Kuo-Liang Ou, and Baw-Jhiune Liu. Discovering decision knowledge from web log portfolio for managing classroom processes by applying decision tree and data cube technology. *Journal of Educational Computing Research*, 23(3):305–332, 2000.

[8] Martin Dougiamas and Peter Taylor. Moodle: Using learning communities to create an open source course management system. 2003.

[9] Mohammad Hassan Falakmasir and Jafar Habibi. Using educational data mining methods to study the impact of virtual classroom in e-learning. In *Educational Data Mining 2010*, 2010.

[10] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[11] Sergio Rapuano and Francesco Zoino. A learning management system including laboratory experiments on measurement instrumentation. *IEEE Transactions on instrumentation and measurement*, 55(5):1757–1766, 2006.

[12] Cristobal Romero, Pedro G. Espejo, Amelia Zafra, Jose Raul Romero, and Sebastian Ventura. Web usage mining for predicting final marks of students that use moodle courses. *Computer Applications in Engineering Education*, 21(1):135–146, 2013.

[13] Cristobal Romero and Sebastian Ventura. Educational data mining: A survey from 1995 to 2005. *Expert systems with applications*, 33(1):135–146, 2007.

[14] Cristóbal Romero and Sebastián Ventura. Educational data mining: A review of the state of the art. *Trans. Sys. Man Cyber Part C*, 40(6):601–618, November 2010.

[15] Cristóbal Romero, Sebastián Ventura, Pedro G. Espejo, and César Hervás. Data mining algorithms to classify students. In Ryan Shaun Joazeiro de Baker, Tiffany Barnes, and Joseph E. Beck, editors, *Educational Data Mining 2008, The 1st International Conference on Educational Data Mining, Montreal, Québec, Canada, June 20-21, 2008. Proceedings*, pages 8–17. www.educationaldatamining.org, 2008.

[16] Xavier Sierra-Canto, Francisco Madera-Ramirez, and Victor Uc-Cetina. Parallel training of a back-propagation neural network using cuda. In *Proceedings of the 2010 Ninth International Conference on Machine Learning and Applications*, ICMLA '10, pages 307–312, Washington, DC, USA, 2010. IEEE Computer Society.

[17] Elizabeth T. Welsh, Connie R. Wanberg, Kenneth G. Brown, and Marcia J. Simmering. E-learning: emerging uses, empirical results and future directions. *International Journal of Training and Development*, 7(4):245–258, 2003.

[18] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[19] Dongsong Zhang, J Leon Zhao, Lina Zhou, and Jay F Nunamaker Jr. Can e-learning replace classroom learning? *Communications of the ACM*, 47(5):75–79, 2004.

UNIVERSIDADE DE ÉVORA
INSTITUTO DE INVESTIGAÇÃO
E FORMAÇÃO AVANÇADA

Contactos:
Universidade de Évora
Escola de Ciências e Tecnologia - Departamento de Informática
Palácio do Vimioso | Largo Marquês de Marialva, Apart. 94
7002 - 554 Évora | Portugal
Tel: (+351) 266 706 581
Fax: (+351) 266 744 677
email: iifa@uevora.pt