Using Machine Learning to Predict Student Performance

Murat Pojon

University of Tampere
Faculty of Natural Sciences
Software Development
M. Sc. Thesis
Supervisor: Jorma Laurikkala
June 2017

University of Tampere
Faculty of Natural Sciences
Software Development
Murat Pojon: Using Machine Learning to Predict Student Performance
M. Sc. Thesis, 35 pages
June 2017

This thesis examines the application of machine learning algorithms to predict whether a student will be successful or not. The specific focus of the thesis is the comparison of machine learning methods and feature engineering techniques in terms of how much they improve the prediction performance.

Three different machine learning methods were used in this thesis. They are linear regression, decision trees, and naïve Bayes classification. Feature engineering, the process of modification and selection of the features of a data set, was used to improve predictions made by these learning algorithms.

Two different data sets containing records of student information were used. The machine learning methods were applied to both the raw version and the feature engineered version of the data sets, to predict the student's success.

The thesis comes to the same conclusion as the earlier studies: The results show that it is possible to predict student performance successfully by using machine learning. The best algorithm was naïve Bayes classification for the first data set, with 98 percent accuracy, and decision trees for the second data set, with 78 percent accuracy. Feature engineering was found to be more important factor in prediction performance than method selection in the data used in this study.

Keywords and terms: student performance, machine learning, regression, naïve Bayes classification, decision trees.

Acknowledgements

I would like to express my sincere gratitude to Jorma Laurikkala for his supervision, special guidance, suggestions, and encouragement through the development of this thesis. Also, I would like to thank my colleague Hannu Pahkala for his support and ideas for the thesis.

Tampere, 5 June 2017

Murat Pojon

Contents

1. Introduction

With the wide usage of computers and internet, there has recently been a huge increase in publicly available data that can be analyzed. Be it online sales information, website traffic, or user habits, data is generated everyday. Such a large amount of data present both a problem and an opportunity. The problem is that it is difficult for humans to analyze such large data. The opportunity is that this type of data is ideal for computers to process, because it is stored digitally in a well-formatted way, and computers can process data much faster than humans.

The concept of machine learning is something born out of this environment. Computers can analyze digital data to find patterns and laws in ways that is too complex for a human to do. The basic idea of machine learning is that a computer can automatically learn from experience (Mitchell, 1997). Although machine learning applications vary, its general function is similar throughout its applications. The computer analyzes a large amount of data, and finds patterns and rules hidden in the data. These patterns and rules are mathematical in nature, and they can be easily defined and processed by a computer. The computer can then use those rules to meaningfully characterize new data. The creation of rules from data is an automatic process, and it is something that continuously improves with newly presented data.

Applications of machine learning cover a wide range of areas. Search engines use machine learning to better construct relations between search phrases and web pages. By analyzing the content of the websites, search engines can define which words and phrases are the most important in defining a certain web page, and they can use this information to return the most relevant results for a given search phrase (Witten *et al.*, 2016). Image recognition technologies also use machine learning to identify particular objects in an image, such as faces (Alpaydin, 2004). First, the machine learning algorithm analyzes images that contain a certain object. If given enough images to process, the algorithm is able to determine whether an image contains that object or not (Watt *et al.*, 2016). In addition, machine learning can be used to understand the kind of products a customer might be interested in. By analyzing the past products that a user has bought, the computer can make suggestions about the new products that the customer might want to buy (Witten *et al.*, 2016). All these examples have the same basic principle. The computer processes data and learns to identify this data, and then uses this knowledge to make decisions about future data. The increase in data has made these applications more effective, and thus more common in use.

Depending on the type of input data, machine learning algorithms can be divided into supervised and unsupervised learning. In supervised learning, input data comes with a known class structure (Mohri *et al.*, 2012; Mitchell, 1997). This input data is known as training data. The algorithm is usually tasked with creating a model that can predict one of the properties by using other properties. After a model is created, it is used to process data that has the same class structure as input data. In unsupervised learning, input data does not have a known class structure, and the task of the algorithm is to reveal a structure in the data (Sugiyama, 2015; Mitchell, 1997).

This thesis focuses on supervised learning, more specifically predictive analytics, which is the process of using machine learning to predict future outcomes (Nyce, 2007). Predictive analytics has a wide range of applications, such as fraud detection, analyzing population trends, or understanding user behavior (Sas, 2017).

The specific focus of this thesis is education. The aim is to predict student performance. Data about students is used to create a model that can predict whether the student is successful or not, based on other properties. First, the training data set is taken as input. There are two different data sets, containing different types of information. These data sets are in tabular format, where each row represents a student and each column, or variable, contains certain information about a student, such as age, gender, family background or medical information. In addition, a column representing the success of the student is used as the variable that the algorithm is trying to predict. The algorithm creates a model, which is a function that outputs success or failure of the student, using other variables as input.

This thesis evaluates the effectiveness of different machine learning algorithms and methods. While algorithms that are used in creating predictive models are numerous, this thesis focuses on three of them, which are linear regression, decision trees, and naïve Bayes classification. The thesis also measures the improvement made by feature engineering, which refers to modifying the data to make it more suitable for machine learning.

There are widely used indicators for evaluating the effectiveness of machine learning algorithms, such as precision, recall and F-measure (Powers, 2011). These are covered in detail in further chapters. These indicators can also be used in evaluating the predictive models. Algorithms were compared to each other in terms indicator values, to determine which algorithm provides the best results. In addition to the algorithm choice, the importance of feature engineering was also tested. To improve the prediction performance, the data sets were modified by variable selection and custom variable creation. Finally, improvements made by feature engineering were compared to improvements made by algorithm choice, to see if one is a more determinant factor than the other. Results of the comparison indicates that feature engineering provides better improvements than method selection.

Chapter 2 describes some of the work done in the field of student performance prediction. Chapter 3 explains the machine learning methods and the evaluation criteria used in this thesis. Chapter 4 briefly describes the two data sets used to create the prediction models. In Chapter 5, the machine learning methods are applied both to the raw and engineered versions of the data sets. Chapter 6 reports the comparison between methods, and the improvements made by feature engineering. In Chapter 7, results are discussed in detail, and, finally, in Chapter 8, future work is discussed.

2. Previous work

Student retention is an important issue in education. While intervention programs can improve retention rates, such programs need prior knowledge of students performance (Yadav *et al*., 2012). That is where performance prediction becomes important. The usage of machine learning to predict either the student performance or the student dropout is a commonly found subject in academic literature. Dropout prediction in virtual learning, or e-learning is a particularly common focus in such studies, due to both high dropout rates and easily available data (Kalles and Pierrakeas, 2006). Areas outside of virtual learning are also common contexts where dropout or performance predictions are used for research. The purpose of the research of these studies varies. In some of them, the aim is to find the best method for prediction. In others, the aim is simply to evaluate whether machine learning is a viable approach for predicting student dropout or performance.

One study evaluating the effectiveness of machine learning for dropout prediction was done at the Eindhoven University of Technology (Dekker *et al*., 2009). Basic methodology was to build multiple prediction models using different machine learning methods, such as CART, BayesNet, and Logit. Then, prediction results of different models were compared in terms of their effectiveness. Most successful model was built by using the J48 classifier. (Dekker *et al*., 2009).

A similar study was made by researchers from three different universities in India (Yadav *et al*., 2012). A data set of university students was analyzed by different algorithms, after which precision and recall values of the predictions were compared. The ADT decision tree model provided the most accurate results (Yadav *et al*., 2012).

However, predicting student performance instead of student dropouts is more related with this thesis, and there are examples of such studies as well. One of these studies, made in the Hellenic Open University, analyzed the usage of machine learning in distance education (Kalles and Pierrakeas, 2006). Genetic algorithms and decision trees were used to build a predictive model, and the results were compared in terms of accuracy. Most accurate results were provided by the GATREE (genetically evolved decision trees) model (Kalles and Pierrakeas, 2006).

Another study about performance prediction was made at the University of Jordan (Amrieh *et al.*, 2016). A data set of students from different countries was used. In addition to using individual machine learning methods, the researchers also applied ensemble methods, and compared the results between them. Decision trees provided the best results. Another area that the researchers focused on were behavioral features. A model was built with and without these features. It was found that the inclusion of behavioral features improved the prediction results (Amrieh *et al.*, 2016).

The last study reviewed here was also about performance prediction. It was done at the University of Minho, Portugal (Cortez and Silva, 2008). The data set contained information about whether the student had passed the exam in the subjects of math and Portuguese language. Decision trees, random forest, neural networks, and support vector machines were used (Cortez and Silva, 2008). These methods were compared in terms of accuracy. Another comparison was made between a data set that included the past exam results and the one that did not. Inclusion of the past grades resulted in an improved performance.

The pattern is similar in most of these studies. First, different algorithms are applied to a data set to build prediction models. Then, predictions made by these models are compared using common evaluation criteria, such as accuracy, precision, and recall. Feature selection is also a commonly compared criteria. However, what these studies are missing is a more comprehensive comparison between distinct approaches such as method selection and feature engineering. This is the part where this thesis can introduce a new approach. By comparing the effectiveness of different processes used in machine learning, this thesis can provide insight into the more efficient ways to improve predictions in student performance.

3. Methods

3.1. Machine learning basics

3.1.1. Definition

A common definition of machine learning is (Mitchell, 1997):

> "A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P* if its performance at tasks in *T*, as measured by *P*, improves with experience *E*."

Basically, machine learning is the ability of a computer to learn from experience (Mitchell, 1997). Experience is usually given in the form of input data. Looking at this data, the computer can find dependencies in the data that are too complex for a human to form. Machine learning can be used to reveal a hidden class structure in an unstructured data, or it can be used to find dependencies in a structured data to make predictions. Latter is the main focus of the thesis.

3.1.2. Predictive analytics

Predictive analytics is the act of predicting future events and behaviors present in previously unseen data, using a model built from similar past data (Nyce, 2007; Shmueli, 2011). It has a wide range of applications in different fields, such as finance, education, healthcare, and law (Sas, 2017). The method of application in all these fields is similar. Using previously collected data, a machine learning algorithm finds the relations between different properties of the data. The resulting model is able to predict one of the properties of future data based on properties (Eckerson, 2007).

Table 1 shows example data about students who passed or failed at an exam, along with other information about students.

Age	Gender	GPA	Absences	Passed
14	F	3.2	5	1
13	M	2.4	7	0
15	M	3	6	1

Table 1. Example data.

The aim is to predict if the student has passed the exam or not by looking at the other variables (the column of the table). In this case, the column "Passed" is called the dependent variable, and every other variable is called the independent variable. In the "Passed" column, "1" means student has passed the exam and "0" means failure in the exam. By applying a machine learning algorithm to this data, a function can be created, also known as the prediction model, that gives the value for the dependent variable as output, and takes every other variable as input.

The act of creating a prediction model from previously known data is called training, and such data is called the training data or a training set. After the model is created, it must be applied to another data set to test its effectiveness. Data used for such purpose is called test data or test set. The reason for using two different sets is to ensure that the model is flexible enough to be used on data sets other than the one it was built with. Otherwise, the problem of overfitting may occur, which is when a model is accurate with its original data set, but performs poorly on other data sets, because it is overly complicated (Srivastava, 2014). A common method to avoid overfitting is to divide the input data set into training and test sets.

To evaluate the model with test data, the model is used to predict the dependent variable in the test set. Then, the predicted values and actual values of the dependent variable are compared. Evaluation is more complicated than looking at the number of correct predictions. There are multiple different evaluation criteria (see Chapter 3.4).

3.2. Selected methods

There are numerous algorithms to create a prediction model. This thesis uses three different algorithms: linear regression, decision trees, and naïve Bayes classifier. While they all essentially have the same task, which is predicting a dependent variable based on independent variables, they are based on different mathematical methods.

3.2.1. Linear regression

Regression method takes a finite set relations between dependent variable and independent variables, and creates a continuous function generalizing these relations (Watt *et al.*, 2016). Table 2 shows another data set containing information about students.

Age	Passed
15	0
14	1
13	1

Table 2. Student data.

For the sake of simplicity, the data has only one independent variable. Figure 1 depicts a two dimensional graph that shows the relation between the student age and the dependent variable indicating whether they have passed the exam or not.

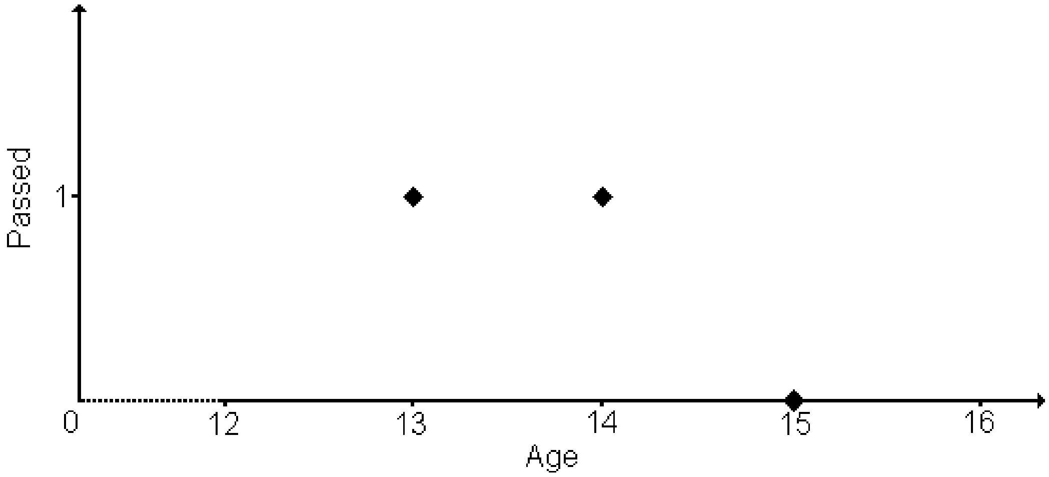

Figure 1. Graph representation of data.

Depending on the type of regression method, regression creates a straight line or a curve that fits the best to the data. Figure 2 shows the graph after the regression.

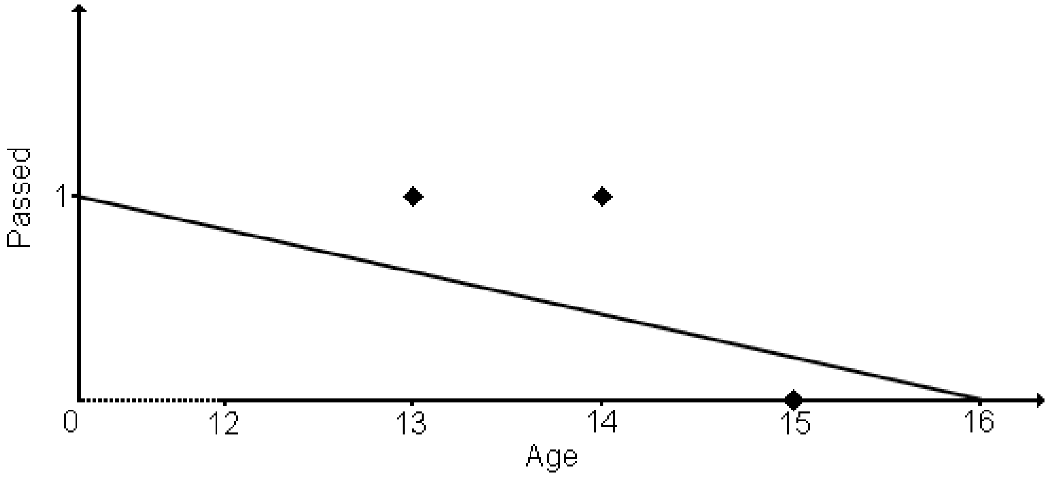

Figure 2. Graph after the regression.

After the regression model has been constructed, predictions about previously unknown cases, such as age 12 and age 16, can be made. Two things should be noted. The first is that regression does not have to cover the exact points in the previous dotted graph. For example, the function no longer has the same values for ages 13, 14 and 15. This is acceptable, because regression algorithm makes an approximation (Watt *et al.*, 2016). Another thing to note is that the function can have any value between 0 and 1 as output. Since 0 and 1 are the only acceptable values, a threshold is needed to convert any output of the function to 0 or 1. For example, threshold can be 0.5, and if the passed value is equal or greater than 0.5, it is 1, otherwise it is 0. Using such threshold, output values for ages 13 and 14 are 1, and 0 for age 15.

3.2.2. Decision trees

Decision trees are graph structures, where each potential decision creates a new node, resulting in a tree-like graph (Quinlan, 1987). Figure 3 shows an example of a decision tree.

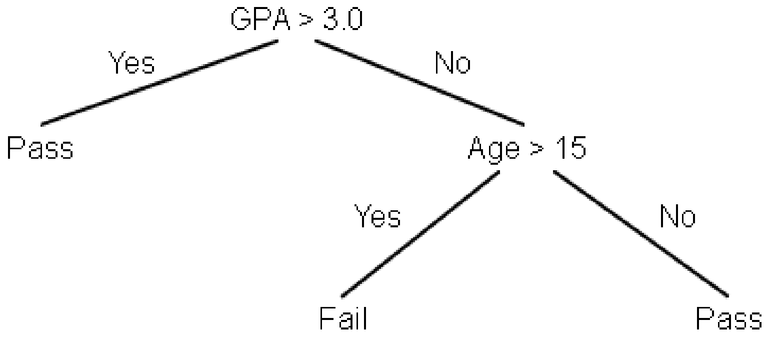

Figure 3. Decision tree example.

This tree is used to predict if a student has passed the exam by looking at GPA and age values. “Yes” and “No” in the edges indicate whether the “GPA > 3.0” and “Age > 15” conditions are met.

In machine learning, decision trees partition the data set in appropriate values until a tree structure has emerged. This process is called recursive partitioning (Strobl, 2009). Decision tree algorithm tries to find the best way to partition the data so that parts are as homogeneous as possible. If a fully homogeneous part is impossible, more common value is chosen. This process is demonstrated by creating a decision tree from student data shown in Table 3.

Age	GPA	Pass
10	3	0
10	3.6	1
12	2	0
12	3	1
13	4	1

Table 3. Decision tree data.

Again, the aim is to predict the “Pass” value using values “Age” and “GPA”. Since the data contains only two independent variables, this data set can be shown as a scatter plot (see Figure 4).

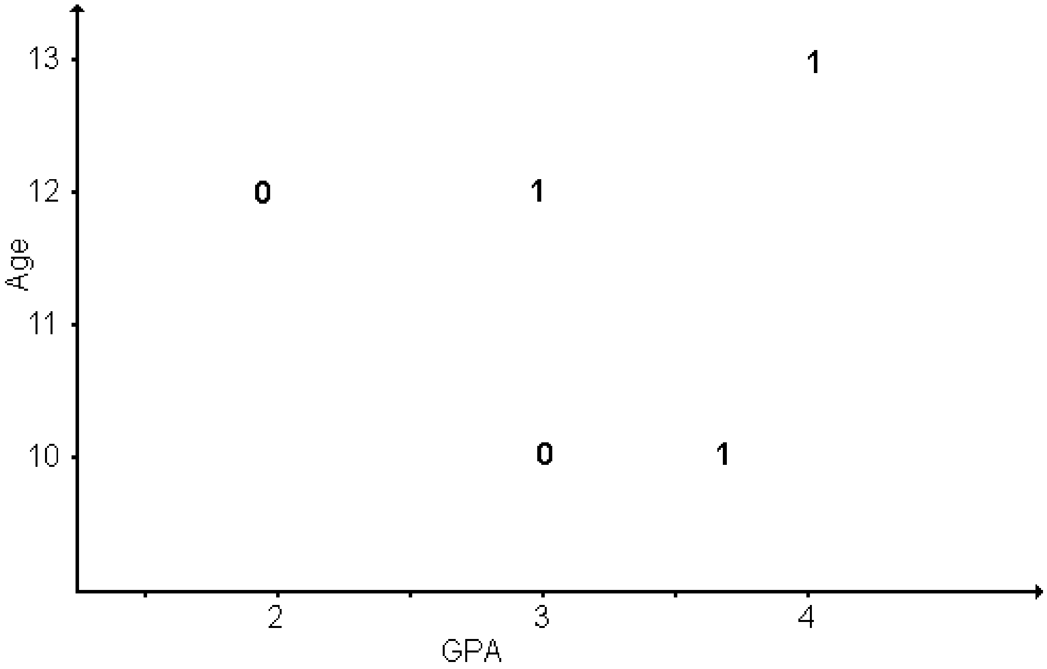

Figure 4. Scatter plot of student data.

The *X* and *Y* axes represents the independent variables, and points in the plot represent the dependent variable (Pass = 1 and Fail = 0). Using a decision tree algorithm, the plot can partitioned, as shown in the Figure 5.

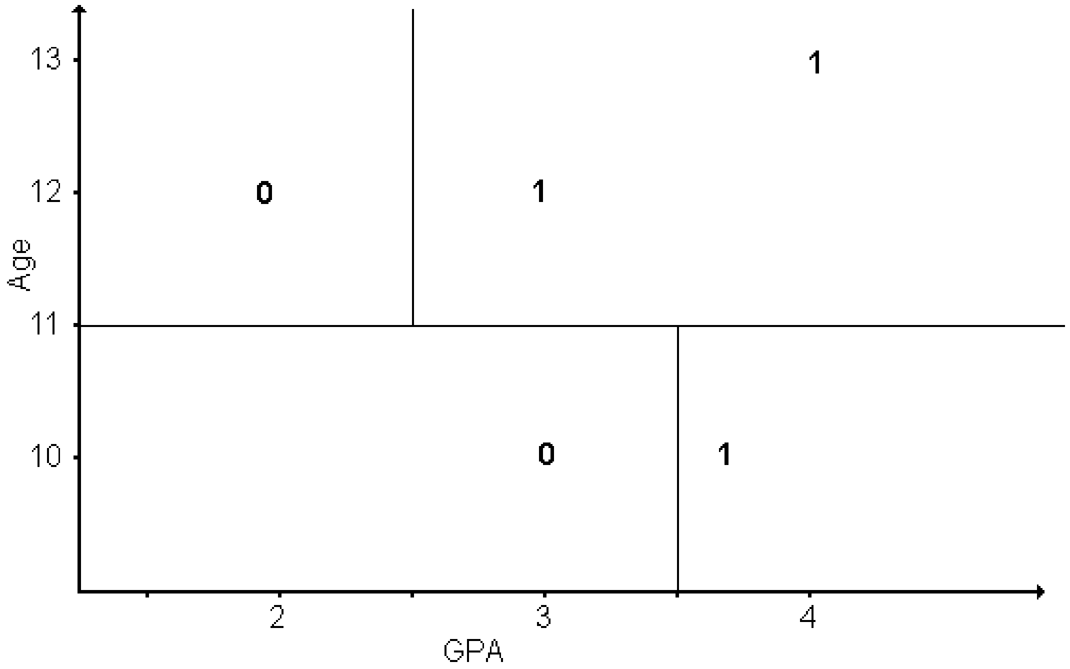

Figure 5. Partitioned student data.

The decision tree algorithm determines the partition locations and the number of partitions. Data in the partitioned plot can be shown in the form of a decision tree (see Figure 6).

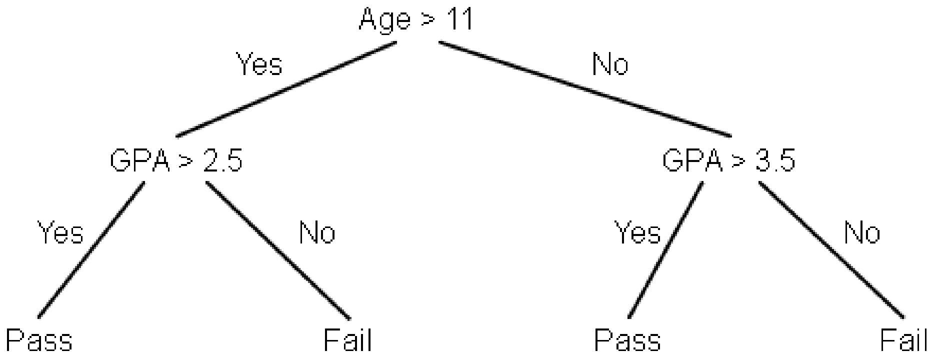

Figure 6. Decision tree version of the plot.

3.2.3. Naïve Bayes classifier

Naïve Bayes classification is a machine learning method relying on the Bayes' Theorem:

$$P(A|B)=\frac{P(B|A)P(A)}{P(B)},$$

where *A* and *B* are two different events, *P(A)* and *P(B)* are the probability of *A* and *B* occurring, respectively. *P(A|B)* is the probability of *A* occurring given that *B* has already occurred (Islam *et al*., 2007). This equation is used to calculate the probability of dependent variable having a certain value. In the following, the Bayes' theorem is applied to classify a student having 3 GPA and age of 12, using data given in Table 4.

GPA	Age	Pass
2	12	1
3	13	1
2	14	0
4	12	0
3	14	0

Table 4. Data for the naïve Bayes classification.

First, the probability of "Pass" being 1 with the specified conditions must be calculated. This is denoted by:

$$P(Pass=1|GPA=3, Age=12)$$

Using the Bayes' formula, this is equal to:

$$P(Pass=1|GPA=3, Age=12)=\frac{P(GPA=3, Age=12|Pass=1)\,P(Pass=1)}{P(GPA=3, Age=12)}.$$

Using the chain rule of conditional probability, the first part of numerator can be expanded to produce this equation:

$$P(Pass=1|GPA=3, Age=12)=\frac{P(GPA=3|Pass=1)\,P(Age=12|Pass=1)\,P(Pass=1)}{P(GPA=3, Age=12)}.$$

Now, the numerator can be calculated:

$$P(Pass=1|GPA=3, Age=12)=\frac{0.5\cdot 0.5\cdot 0.4}{P(GPA=3, Age=12)}=\frac{0.1}{P(GPA=3, Age=12)}.$$

Taking the same steps for "Pass" value being 0, resulting equation is:

$$P(Pass=0|GPA=3, Age=12)=\frac{0.33\cdot 0.33\cdot 0.6}{P(GPA=3, Age=12)}=\frac{0.067}{P(GPA=3, Age=12)}.$$

Last, the probability of "Pass" being 0 is compared to the probability of "Pass" being 1. Since the expression $P(GPA=3, Age=12)$ is the same in both formulas, comparing the numerator is sufficient. Predicted “Pass” value is 1 for the “Age” value 12 and “GPA” value 3, because the probability for passing (0.1) is greater than that of failing (0.067). This is a very basic example, where only some value combinations have a probability. In actual implementation, a distribution needs to be used.

This classification method assumes that features, in this case “Age” and “GPA”, are independent from each other, meaning that occurrence of one does not affect the probability of the other. The independence assumption is the reason for the phrase “naïve” (Friedman, 2001).

3.3. Feature engineering

In machine learning, feature engineering is the process of selecting or creating features (variables) in a data set to improve machine learning results (Domingos, 2012). Feature selection can include removing unnecessary or redundant features. The process of removing unnecessary variables requires assessing the relevance of the variable. This can be done by creating a model to test the correlation of the variable with the dependent variable. Feature creation includes modifying the variables and creating new ones by combining multiple different variables (Kern, 2014).

The first use of feature engineering in the thesis is the selection of the relevant variables. Input data may contain too many variables, some of which do not improve the prediction performance, and thus make the predictive model overly complicated. In such a case, unnecessary variables must be removed to make the model more efficient. Deciding which variable to remove can be done manually using domain knowledge or it can be done automatically (Domingos, 2012). In the case of this thesis, feature selection was done by observing the output of the linear regression model to find how much correlation each variable has with the dependent variable.

The second use of feature engineering in the thesis is the modification of variables. This can refer to combining multiple variables to create a new variable, calculating a variable differently so that it can be used better in classification, or categorizing a variable so that it has a limited range of possible values. An example of variable modification can be made with a student data set containing the native language of the student as one of the variables. Table 5 contains the the data.

Student Id	Age	Native language	Passed
1	14	Finnish	1
2	15	Finnish	0
3	13	Turkish	0
4	13	Finnish	1
5	16	Finnish	1
6	15	Arabic	0
7	14	English	1
9	14	Finnish	1
10	15	Finnish	0

Table 5. Example data for feature engineering.

In this example, the variable “Native language” has four possible values, which are “Finnish”, “Turkish”, “Arabic', and “English”. However, the vast majority of this variable has the value “Finnish”, and rest of them form a small group. In such a case, variable might be modified so that possible values are “Finnish” and “Other”. In an environment where education language is Finnish, modifying the data in such way does not affect the importance of the variable, while making it more simple. This is a manual process, and deciding the usefulness and results of such modification requires domain knowledge. In this thesis, feature modification is done by creating a new custom variable as a function of different variables.

3.4. Evaluation methods

In order to evaluate the effectiveness of a prediction model, predicted values must be compared with actual values. There are multiple criteria for prediction effectiveness. Table 6 shows the possible results of prediction for binary values.

	Predicted as True	Predicted as False
Actually True	True Positive	False Negative
Actually False	False Positive	True Negative

Table 6. Possible prediction results.

The matrix that shows the possible prediction results is called a confusion matrix (Fawcett, 2005). There are different evaluation criteria that can be obtained from these values. One is accuracy, defined as (Powers, 2011):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy is basically the ratio of correct predictions. However, accuracy has limitations in evaluating the prediction performance. Especially, accuracy does not show how the cases of minority class are classified, when the class distribution is imbalanced. As an example, a data set that contains 100 students, 90 of which has passed the exam, might be considered. A crude prediction (known as the majority rule) that does not use any machine learning method, but instead predicts that every student will pass the exam, has 90% accuracy. The model should perform better than just guessing that each case belongs to the majority class.

In this thesis, three other criteria are used. Two of them are precision and recall, which are defined as (Powers, 2011):

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

Precision and recall are used together to make a better evaluation. The main idea is that accurately predicting positive outcome is not enough. A good predictive model must have a good combination of successful positive predictions and successful negative predictions. The third criteria that is used by this thesis is called F-measure, and it is defined as (Fawcett, 2005):

$$F = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

F-measure is a way of having a single value that takes both precision and recall into account. F-measure is the final evaluation criteria for comparisons in this thesis.

4. Materials

Two different data sets were used for this research. The first data set was originally used in research made at the University of Jordan (Amrieh *et al.*, 2016). It contains information about 480 students from various countries, mostly in the Middle East. The data has a total of 17 variables (see Table 7).

Column	Description	Type
Gender	Gender of student	Nominal
Nationality	Nationality of student	Nominal
PlaceofBirth	Country of birth for student	Nominal
StageID	Educational stage, for example Middle school, high school	Nominal
GradeID	Grade level of the student	Nominal
SectionID	Classroom of the student	Nominal
Topic	Course topic	Nominal
Semester	Semester of the year	Nominal
Relation	Parent responsible for the student	Nominal
Raisedhands	Number of times the student raised hands during the class	Quantitative
VisitedResources	Number of times the student visited the course content	Quantitative
AnnouncementsView	Number of times the student checked new announcements	Quantitative
Discussion	Number of times the student joined the discussion groups	Quantitative
ParentAnsweringSurvey	Did the parent answer the school surveys	Nominal
ParentschoolSatisfaction	Parents level of satisfaction for the school	Nominal
StudentAbsenceDays	Number of days the student has been absent	Quantitative
Class	Grade of student for the course	Quantitative

Table 7. Variable descriptions for the first data set.

Variables of the data has two types. Nominal types have a specific set of values, while quantitative types can have values which can be ordered (Card, 1997). Variable “Class” is the dependent variable, meaning it is the variable that the model is trying to predict. It can have three different values, which are “L”,“M”, and “H”. Value “L” means low, which represents a grade score between 0 and 69. Value “M” means medium, which represents a grade score between 70 and 89. The final value “H” means high, and it represents a grade score between 90 and 100.

The second data set was originally used in a research done at the University of Minho, Portugal (Cortez and Silva, 2008). It contains information about 395 students amd has 31 different variables (see Table 8).

Column	Description	Type
School	Name of student's school	Nominal
Sex	Gender of student	Nominal
Age	Age of student	Quantitative
Address	Whether the student lives in urban or rural area	Nominal
Famsize	Student's family size	Nominal
Pstatus	Whether the parents are living together or apart	Nominal
Medu	Mother's education	Quantitative
Fedu	Father's education	Quantitative
Mjob	Mother's job	Nominal
Fjob	Father's job	Nominal
Reason	Reason to choose the school	Nominal
Guardian	Student's guardian	Nominal
Traveltime	Travel time between home and school	Quantitative
Studytime	Study time in a week	Quantitative
Failures	Number of times student failed in past	Quantitative
Schoolsup	Educational support from school	Nominal
Famsup	Educational support from family	Nominal
Paid	Extra paid classes	Nominal
Activites	Extra activities	Nominal
Nursery	Attended nursery school	Nominal
Higher	If the student wants to pursue higher education	Nominal
Internet	If the student has internet at home	Nominal
Romantic	Does the student have a relationship	Nominal
Famrel	Family relations quality	Quantitative
Freetime	Student's amount of free time	Quantitative
Goout	Going out with friends	Quantitative
Dalc	Alcohol take during weekdays	Quantitative
Walc	Alcohol take during weekends	Quantitative
Health	Student's health	Quantitative
Absences	Number of times student was absent	Quantitative
G3	Final grade	Quantitative

Table 8. Variable descriptions for the second data set.

Variable “G3” is the dependent variable. It can have a value between 0 and 20. Originally, the data had two other variables, “G1” and “G2”, describing the grades for the first period and the second period, respectively. They are not included in this research, because past grades would have too much prediction power over the final grade, diminishing the importance of other variables. Furthermore, a realistic case of predicting student performance require making a prediction before student has started taking exams.

Before any feature engineering, a modification of dependent variables in both data sets was made. They were converted to binary variables. For the first data set, values “M” and “H” were converted to 1, value “L” was converted to 0. For the second data set, values equal to or greater than 10 were converted to 1 and values less than 10 were converted to 0. This way, 0 and 1 mean that student performed successfully and non-successfully, respectively.

5. Implementation and results

The aim of the research was to compare different machine learning methods and feature engineering in the student performance prediction. The prediction models were created using the R language. It is a language commonly used for machine learning applications. It has built-in functions for the three methods selected for this research, which are linear regression, decision tree, and naïve Bayes classification. It also creates the necessary output for evaluating and refining the results of predictions. The code written in the R language is run on an application called R Studio.

5.1. Results from raw data

5.1.1. The first data set

The first step was to apply the machine learning methods to the raw data. In this case, the only processing done to the data was the modification of the dependent variables to make them binary. A total of 353 students out of 480 has performed well or satisfactory, and therefore, the majority rule has accuracy of 73 percent. This is the baseline accuracy for this data set, to which the accuracy of prediction models built on this data set were compared to see, if the models can make useful predictions.

After calculating the baseline accuracy, the next step was to divide the data into the training and test sets. Training set (75 percent of data) was used to build the prediction model and test set (25 percent of data) was used to test the model. While building training and test sets, an important thing to consider is that both sets must contain similar ratios of students from both classes. The R language has a built-in functionality ensuring that the cases of different classes are spread proportionally among the training and test sets.

Next step after creating the test and training sets, was to build the models. The first model was created using the linear regression method. Building a model using the R language is a straightforward process which mainly includes defining the input data, dependent variable, and independent variables. After the model is created, it is applied to the test data set. Output of this process that concerns the thesis is a confusion matrix. It contains data about predicted values and actual values. Table 9 shows the confusion matrix for the first prediction model.

	Predicted False	Predicted True
Actual False	26	6
Actual True	2	86

Table 9. Confusion matrix for linear regression used on the first raw data set.

The accuracy calculated from the confusion matrix is 93 percent. This is an improvement over the baseline accuracy of 73 percent. Results are further evaluated in next chapter.

The second model was created using the decision tree method. The R function for this model is an implementation of CART (Strobl, 2009), classification and regression tree. Apart from the function used, the procedure is the same as with the previous model. Training and test sets were created, model was built using the training set, and then applied to the test set. Table 10 shows the confusion matrix for this model.

	Predicted False	Predicted True
Actual False	24	8
Actual True	0	88

Table 10. Confusion matrix for CART used on the first raw data set.

The accuracy of this model is 93 percent which is the same as tat of the previous model. The last model for this data set is built by the naive Bayes classification method. Table 11 shows the confusion matrix for the model. This confusion matrix gives the accuracy of 95 percent.

	Predicted False	Predicted True
Actual False	28	4
Actual True	1	87

Table 11. Confusion matrix for the naïve Bayes classification used on the first raw data set.

5.1.2. The second data set

The second data set contains 395 students. In this data set, 265 students out of 395 have a passing grade. This means the baseline accuracy for this data set is 67 percent. As in the first data set, the training and test sets were created, and models were built using the machine learning methods. Then, models were tested and confusion matrices were produced as the relevant output. Tables 12-14 show the confusion matrices for the linear regression, CART, and naïve Bayes classifier methods respectively.

	Predicted False	Predicted True
Actual False	14	18
Actual True	7	59

Table 12. Confusion matrix for linear regression used on the second raw data set.

	Predicted False	Predicted True
Actual False	8	24
Actual True	7	59

Table 13. Confusion matrix for CART used on the second raw data set.

	Predicted False	Predicted True
Actual False	14	18
Actual True	8	58

Table 14. Confusion matrix for the naive Bayes classification used on the second raw data set.

Accuracy values for the models are 74 percent, 68 percent, and 73 percent respectively. Although results are further evaluated next chapter, it can be already noticed that machine learning methods does not offer much improvement over the baseline method in this data set.

5.2. Results from engineered data

To improve the prediction performance, the data sets were modified. The first modification method in the thesis was feature selection. To find the important variables in the data sets, the process of variable ranking can be used (Guyon and Elisseeff, 2003). In this thesis, it is done by using the output of linear regression model, which shows the correlation of each dependent variable with the independent variable. The process of selecting variables is done by a trial and error approach, where the machine learning model was built multiple times with different sets of relevant variables, and the best combination of variables is identified. The second method of modification was custom feature creation, where important variables are combined into custom variables to make the decision trees more efficient.

5.2.1. The first data set

The linear regression function of R language has a built-in functionality to determine each independent variable's correlation with the dependent variable. Table 15 shows the most relevant variables identified using the correlations calculated from the raw data having 16 dependent variables.

Column	Description	Type
Raisedhands	Number of times the student raised hands during the class	Quantitative
VisitedResources	Number of times the student visited the course content	Quantitative
Discussion	Number of times the student joined the discussion groups	Quantitative
ParentAnsweringSurvey	Did the parent answer the school surveys	Nominal
StudentAbsenceDays	Number of days the student is absent	Quantitative

Table 15. Relevant variables in the first data set.

The next step was to build the prediction models using only these variables. The first model was built using the linear regression method. Table 16 shows the confusion matrix created by the model.

	Predicted False	Predicted True
Actual False	27	5
Actual True	1	87

Table 16. Confusion matrix for linear regression used on first engineered data set.

The accuracy value for this confusion matrix is 95 percent which is a slight improvement over the 93 percent accuracy of the raw data.

The second model was built using the CART method. Table 17 shows the confusion matrix of the model, using only the selected variables.

	Predicted False	Predicted True
Actual False	24	8
Actual True	0	88

Table 17. Confusion matrix for CART used on first engineered data set.

The accuracy for this matrix is 93 percent which is the same as in the accuracy of the model created from the raw data. In order to improve this method, further feature engineering was required. Analyzing the decision tree created by the CART method (see Figure 6) was important to determine what type of modification could be done to improve the results.

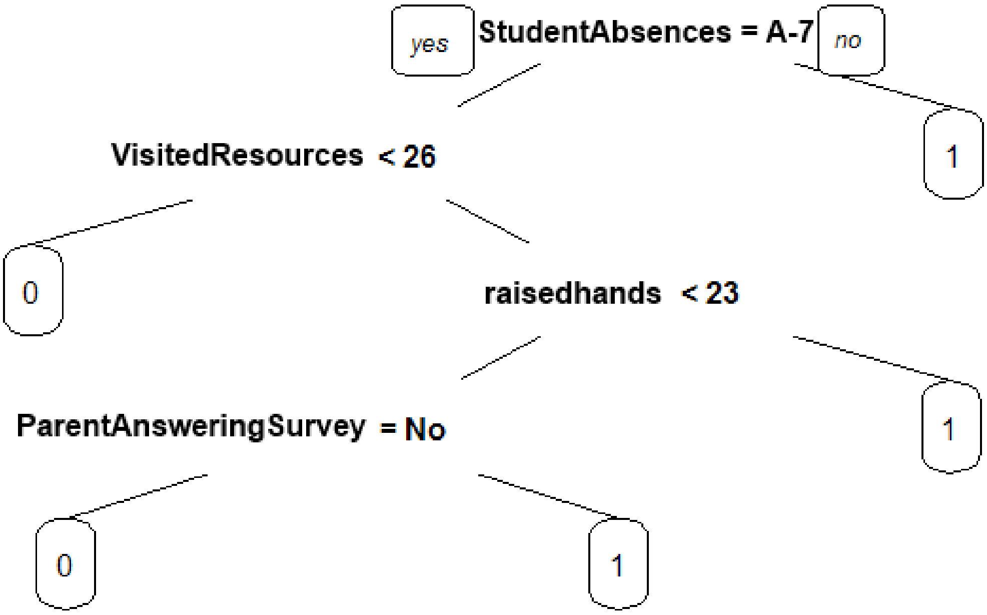

Figure 6. Decision tree created by the CART model on the first raw data set. "A-7" indicates whether the absence number is higher than 7 or not.

The CART model automatically determines the optimal number of nodes for the most effective decision tree. The problem here is that the number of nodes is less than the number of variables available. One possible approach to this problem is to combine multiple important variables to create a custom variable. A formula for the custom variable was defined as:

$$customVar = A \cdot StudentAbsences - B \cdot VisitedResources - C \cdot raisedhands$$

In this formula, symbols A, B, and C are coefficients that are determined by interpreting the data and assessing the importance of each variable. Logic of this formula is that student's success chance is proportional to “VisitedResources” and “RaisedHands” variables, but inversely proportional to “StudentAbsences” variable. For this reason, parts with B and C are subtracted. The process also includes trial and error, where model is built with different coefficients and the values are modified to improve the results.

Values of A, B, C were determined to be 30, 0.8, and 1.1 respectively. After the custom variable was created, the CART model was created again, this time including the new variable. Table 18 shows the confusion matrix from the results obtained from the further modified data.

	Predicted False	Predicted True
Actual False	29	3
Actual True	1	87

Table 18. Confusion matrix for the CART model on the first data set, with a custom variable.

The accuracy calculated from this confusion matrix is 96 percent, which is better than the 93 percent accuracy obtained from the raw data.

The third model was built with the naïve Bayes classification method. Similar to the linear regression model, only feature engineering done for this model was the variable selection. Table 19 shows the confusion matrix.

	Predicted False	Predicted True
Actual False	29	3
Actual True	0	88

Table 19. Confusion matrix for the naïve Bayes classification used on the first engineered data set.

This confusion matrix gives an accuracy of 97 percent. Model built with the raw data had 93 percent accuracy.

5.2.2. The second data set

As in the first data set, the relevant variables of the second data set were determined using correlations. Table 20 shows the six most relevant variables out of the total of 30 dependent variables.

Column	Description	Type
Failures	Number of times student failed in past	Quantitative
Age	Age of student	Quantitative
Absences	Number of times student was absent	Quantitative
Studytime	Study time in a week	Quantitative
Schoolsup	Educational support from school	Quantitative
Famsup	Educational support from family	Nominal

Table 20. Relevant variables in the second data set.

After the relevant variables were identified, the next step was to build the models, as earlier. Table 21 shows the confusion matrix of the linear regression model.

	Predicted False	Predicted True
Actual False	12	20
Actual True	2	64

Table 21. Confusion matrix for linear regression used on the second engineered data set.

The accuracy calculated from this matrix is 77 percent, which is a better accuracy compared to the 74 percent with the raw data.

The second model was built using the CART method, using only the relevant variables. Table 22 and Figure 7 show the confusion matrix and the decision tree produced by this model, respectively.

	Predicted False	Predicted True
Actual False	14	18
Actual True	8	58

Table 22. Confusion matrix for CART used on the second engineered data set.

Accuracy obtained from this matrix is 73 percent. Although it is an improvement over the 68 percent accuracy of raw data, further modifications can be made.

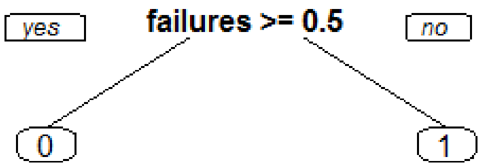

Figure 7. Decision tree created by the CART model on the second data set.

This is an overly simple tree with just one node. A custom variable was built to improve the model also in this data set. Formula of the custom variable is defined as:

$$customVar = A \cdot failures + B \cdot absences - C \cdot studytime$$

Symbols *A*, *B*, and *C* are coefficients that are determined by interpreting the data and assessing the importance of each variable. Values of *A*, *B*, and *C* were 9, 0.9, and 2 respectively. Parts with coefficient is subtracted because study time is inversely proportional to failures and absences. Using this custom variable the CART model is built again. Table 23 shows the confusion matrix of the new model.

	Predicted False	Predicted True
Actual False	13	19
Actual True	3	63

Table 23. Confusion matrix for the CART model on the second data set, with a custom variable.

Accuracy of this model is 77 percent, and it is even higher than the model created with relevant variables.

The third model was created again using the naïve Bayes classification method, using only the relevant variables. Table 24 shows the confusion matrix.

	Predicted False	Predicted True
Actual False	13	19
Actual True	5	61

Table 24. Confusion matrix for the naïve Bayes classification used on the second engineered data set.

This confusion matrix gives the accuracy of 75 percent. The model using the raw data had 73 percent accuracy.

6. Evaluation

6.1. Method comparison

The first step in evaluating the results is to compare the machine learning methods in terms of their prediction performance. Tables 25 and 26 show the prediction results of the three machine learning methods for the first data set, with the raw data and the modified data, respectively.

Method	Accuracy (%)	Precision (%)	Recall (%)	F-measure
Linear regression	93.3	93.5	97.7	0.952
Decision trees	93.3	91.7	100	0.956
Naïve Bayes classification	95.8	95.6	98.9	0.972

Table 25. Method comparison for the first data set, with raw data.

Method	Accuracy (%)	Precision (%)	Recall (%)	F-measure
Linear regression	95	94.6	98.9	0.966
Decision trees	96.7	96.7	98.9	0.978
Naïve Bayes classification	97.5	96.7	100	0.984

Table 26. Method comparison for the first data set, with engineered data. A custom variable added for decision tree.

In both raw and engineered data, the performances of different methods are similar to each other, and in both cases, the naïve Bayes classification provides the best results, followed by decision tree and linear regression. For each method, feature engineering provides an improvement in prediction performance.

The second set of tables compares machine learning methods applied to the second data set. Tables 27 and 28 show the prediction results of the three machine learning methods for the second data set, with raw data and modified data, respectively.

Method	Accuracy (%)	Precision (%)	Recall (%)	F-measure
Linear regression	74.5	76.6	89.4	0.826
Decision trees	68.4	71.1	89.4	0.792
Naïve Bayes classification	73.5	76.3	87.9	0.816

Table 27. Method comparison for the second data set, with raw data.

Method	Accuracy (%)	Precision (%)	Recall (%)	F-measure
Linear regression	77.5	76.1	97.0	0.854
Decision trees	77.6	76.8	95.5	0.853
Naïve Bayes classification	75.5	76.3	92.4	0.838

Table 28. Method comparison for the second data set, with engineered data. A custom variable added for decision tree.

Machine learning models built with this data set were not as accurate as those of the first data set. Furthermore, methods have a different order of success in this data set. Unlike the first data set, this one has linear regression as the most effective model. With the engineered data, different methods shows similar performances. However, raw data contains one exception when it comes to method performance similarity. Decision tree model for the raw data has a slightly different performance compared to linear regression and the naïve Bayes classification. For all three methods, feature engineering improves the prediction results.

6.2. Feature engineering improvements

The second step of evaluating the results is the detailed analysis of the effects of feature engineering on each method. Tables 29 and 30 show the differences between the performances of the different methods in the modified (Table 29) and the raw data sets (Table 30). The differences in accuracy, precision, and recall are percentage points. For example, the first cell of Table 29 contains the difference of accuracies obtained with linear regression from the engineered and raw data sets (95.0% - 93.3% = 1.7%). The values in the tables show that feature engineering caused a performance improvement for each method and data set.

Method	Difference in accuracy (%)	Difference in precision (%)	Difference in recall (%)	Difference in F-measure
Linear regression	1.7	1.1	1.2	0.014
Decision trees	3.4	5.0	-1.1	0.022
Naïve Bayes classification	1.7	1.1	1.1	0.012

Table 29. Feature engineering effects on the first data set.

Method	Difference in accuracy (%)	Difference in precision (%)	Difference in recall (%)	Difference in F-measure
Linear regression	3.0	0.5	7.6	0.028
Decision trees	9.2	5.7	6.1	0.060
Naïve Bayes classification	2.0	0.0	4.5	0.022

Table 30. Feature engineering effects on the second data set.

6.3. Feature engineering versus method selection

The final step of evaluation is comparing feature engineering and method selection in terms of prediction performance. The aim is to determine which one has more impact on the prediction results. For this, mean and median values of F-measure improvements were calculated. There are two data sets and three machine learning methods, which means there are six cases where improvements made by feature engineering can be observed (see Tables 29 and 30). As for the improvements made by method selection, there are a total of eight cases (see Table 31). Mean and median F-measure improvement values for feature engineering are 0.0264 and 0.022 respectively. For method selection, mean and median values are both 0.011.

Case	Low F-measure	High F-measure	F-Measure improvement
First Raw Data - First and second Method	0.956	0.972	0.016
First Raw Data - Second and third Method	0.952	0.956	0.004
First Engineered Data - First and second Method	0.978	0.984	0.006
First Engineered Data - Second and third Method	0.966	0.978	0.012
Second Raw Data - First and second Method	0.816	0.826	0.010
Second Raw Data - Second and third Method	0.792	0.816	0.024
Second Engineered Data - First and second Method	0.852	0.854	0.002
Second Engineered Data - Second and third Method	0.838	0.852	0.014

Table 31. Eight cases of method selection.

7. Discussion and conclusions

The success of machine learning in predicting student performance relies on the good use of the data and machine learning algorithms. Selecting the right machine learning method for the right problem is necessary to achieve the best results. However, the algorithm alone can not provide the best prediction results. Feature engineering, the process of modifying data for machine learning, is also an important factor in getting the best prediction results.

The aim of this thesis was to compare method selection and feature engineering, in terms of their ability to improve the prediction results. Two different data sets were analyzed with three different machine learning methods, and their results were compared using four evaluation measures. Methods used were linear regression, decision trees, and naïve Bayes classification. For the evaluation of feature engineering, machine learning methods were applied to the raw and modified versions of the data separately. The main method of feature engineering was feature selection. In the case of classification and regression trees, additional feature engineering was done in the form of custom feature creation. Feature engineering was done both with automatic functionality and manual interpretation of the data. In addition, fine tuning of features was done with a trial and error approach.

Results of both data sets show similarities and differences with their use in the original studies. In the first data set, similarity is that recall values were consistently higher than precision values. Difference was in the accuracy values. The accuracy reached in this thesis was higher than in the original research (Amrieh *et al.*, 2016). This can be attributed to the difference in dependent variables. In original research, dependent variable was not converted to binary, and it has three values instead of two. Generalizing the dependent variable might have made the predictions easier in this thesis. In the second data set, original research used additional variables that indicate the past exam grades, and achieved better accuracy than in this thesis (Cortez and Silva, 2008). However, once those variables are omitted, accuracy values were similar to those of this thesis.

The models that used the first data set gave much better results compared to the models built with the second data set. Accuracy values for the first data set ranged from 93 percent to 98 percent, while accuracy values for the second data set were between 68 percent and 78 percent. Although the second data set contained more features than the first one, results imply that features in the first data set were more related to the student success. This shows the importance of data when it comes to prediction performance. Methods used for both data sets were nearly identical, but the results where very different. This indicates that better methods can not offset the limitations of the data.

The results of this study indicate that feature engineering provides more improvement to prediction results than method selection. Despite feature engineering was done in a limited capacity, it made a bigger difference in prediction performance. Furthermore, biggest leap in improvement was made in the case of decision trees, where both feature selection and feature modification is applied to the data. When trying to improve the prediction of student performance, the modification of input data is an important factor besides selecting the right method for the data.

Although feature engineering was more effective than method selection, the combination of both approaches provided the best results. In both data sets, best possible accuracy values were a clear improvement over the baseline accuracy values. This shows that using machine learning is an effective way of predicting the student performance.

8. Future work

This research has certain limitations that must be noted. There was not an access to a dedicated student data set, and the study relies on public data sources. In addition, both data sets were small, having less than thousand records. A research that has access to more comprehensive data may offer more conclusive results.

Another area that future research can improve is the variety of the machine learning methods. This research used linear regression, decision trees, and the naïve Bayes classification. Other methods, such as clustering and artificial neural networks can be used to have a better understanding of the importance of method selection.

Final area that can be improved is the process of feature creation. Since the data is limited, the amount of feature modification that can be made is also limited. Both data sources used in this research consists of a single table, and custom variables were created using variables from the same table. With a more comprehensive data set that spans multiple tables, there will be more potential to create new custom variables, while keeping in mind that the more a custom variable is, the more difficult it is to interpret the relation between it and the dependent variable.

References

Ethem Alpaydin. 2004. *Introduction to Machine Learning*. Cambridge, MA.

Elaf Abu Amrieh, Thair Hamtini, and Ibrahim Aljarah. 2016. Mining educational data to predict student's academic performance using ensemble methods. *International Journal of Database Theory and Application* 9(8), 119-136.

S. K. Card and J. Mackinlay. 1997. The structure of the information visualization design space. In: *Proceedings of the 1997 IEEE Symposium on Information Visualization.* IEEE, 92-99.

Paulo Cortez and Alice Maria Gonçalves Silva. 2008. Using data mining to predict secondary school student performance. In: *Proceedings of 5th Annual Future Business Technology Conference, Porto,* 5-12.

G. Dekker, M. Pechenizkiy, and J. Vleeshouwers. 2009. Predicting students drop out: A case study. In: *Educational Data Mining 2009,* 41-50.

Pedro Domingos. 2012. A few useful things to know about machine learning. *Communications of the ACM* 55(10), 78-87.

Wayne W. Eckerson. 2007. Predictive analytics. *Extending the Value of Your Data Warehousing Investment. TDWI Best Practices Report* 1, 1-36.

Tom Fawcett. 2005. An introduction to ROC analysis. *Pattern Recognition Letters* 27(8), 861-874.

J. Friedman, T. Hastie, and R. Tibshirani. 2001. *The Elements of Statistical Learning*. Springer, Berlin: Springer Series in Statistics.

Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research* 3, 1157-1182.

M. J. Islam, Q. J. Wu, M. Ahmadi, and M. A. Sid-Ahmed. 2007. Investigating the performance of naive-Bayes classifiers and k-nearest neighbor classifiers. In: *International Conference on Convergence Information Technology.* IEEE*,* 1541-1546.

D. Kalles and C. Pierrakeas. 2006. Analyzing student performance in distance learning with genetic algorithms and decision trees. *Applied Artificial Intelligence* 20(8), 655-674.

Roman Kern. 2014. Feature Engineering, Knowledge Discovery and Data Mining, http://kti.tugraz.at/staff/denis/courses/kddm1/featureengineering.pdf. Retrieved May 3, 2017.

Tom M. Mitchell. 1997. *Machine Learning.* McGraw-Hill.

M. Mohri, A. Rostamizadeh, and A. Talwalkar. 2012. *Foundations of Machine Learning (Adaptive Computation and Machine Learning Series)*. MIT Press.

C. Nyce and CPCU. 2007. A. Predictive analytics white paper. *American Institute for CPCU. Insurance Institute of America*, 9-10.

David M.W. Powers. 2011. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation. *Journal of Machine Learning Technologies* 2(1), 37-63.

J. Ross Quinlan. 1987. Simplifying decision trees. *International Journal of Man-Machine Studies* 27(3), 221-234.

Sas. 2017. Predictive Analytics: What it is and why it matters, SAS. https://www.sas.com/en_us/insights/analytics/predictive-analytics.html. Retrieved April 24, 2017.

Galit Shmueli, and Otto R. Koppius. 2011. Predictive analytics in information systems research. *Mis Quarterly* 35(3), 553-572.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15(1), 1929-1958.

Carolin Strobl, James Malley, and Gerhard Tutz. 2009. An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological Methods* 14(4), 323-348.

Masashi Sugiyama. 2015. *Introduction to Statistical Machine Learning*. Morgan Kaufmann.

Jeremy Watt, Reza Borhani, and Aggelos Katsaggelos. 2016. *Machine Learning Refined: Foundations, Algorithms, and Applications*. Cambridge University Press.

Ian H. Witten, Eibe Frank, Mark A. Hall, and Christopher J. Pal. 2016. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.

Surjeet Kumar Yadav, Brijesh Bharadwaj, and Saurabh Pal. 2012. Mining education data to predict student's retention: A comparative study. *International Journal of Computer Science and Information Security* 10(2), 113-117.