Methodologies of Early Detection of Student Dropouts

Ruta Gronskyte

Kongens Lyngby 2011
IMM-M.Sc-2011-67

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

IMM-M.Sc: ISSN 1601-233X

Summary

The Danish government hopes to have more highly educated young people in the future. However very high dropout rates especial from the technical subjects makes it difficult to achieve the goal. The Technical University of Denmark is already trying to monitor the performance of the students, but the current method is not efficient enough. Monitoring students can raise an alarm to the administration about potential dropout students. Helping potential dropouts might get them back on the track for graduation.

In this master's thesis student dropouts from the Technical University of Denmark is analysed. Several methods like the logistic regression, principal component logistic regression, classification and regression trees (CART), classification and regression trees bagging (CART bagging), random forest and multivariate adaptive regression splines are investigated. With each method a dropout detection system is built and compared. For model building and testing historical data is used.

CART and CART bagging performs significantly better than the others. Further analysis must be performed for tuning the final system. In comparison to the current system all analysed models performs better.

Resumé

Den danske regering håber på at have flere højtuddannede unge i fremtiden. Store frafald fra især de tekniske fag gør det svært at indfri målet. Danmarks Tekniske Universitet prøver allerede at overvåge de studerendes resultater, men den nuværende metode er ikke effektiv nok. Overvågning af de studerende kan alamere administrationen om en studerendes mulige frafald. En mulig frafaldende studerende kan hjælpes tilbage på sporet så de kan færdiggøre uddannelsen.

I det indeværende kandidatspeciale analyseres frafaldne studerende fra Danmarks Tekniske Universitet. En række metoder som logistisk regression, principal komponent logistisk regression, klassifikations og regressionstræer (CART), klassifikations og regressionstræer *bagging* (CART bagging), tilfældig skov, multivariat adaptiv regressionskilenoter undersøges. For alle metoder bygges og sammenlignes et system til at opdage frafald. Historisk data bruges til opbygningen af modellerne.

CART og CART bagging er signifikant bedre til at opdage frafald end nogen af de andre. Yderligere analyse skal til for at fintune det endelige system. Sammenlignet med det nuværende system er alle de analyserede modeller bedre til at opdage frafald.

Preface

This thesis was prepared at Informatics Mathematical Modelling, the Technical University of Denmark in partial fulfillment of the requirements for acquiring the Master of Science in Engineering (Mathematical Modelling and Computation).

The thesis deals with different aspects of mathematical modelling of systems using data and partial knowledge about the structure of the systems. The main focus is on modelling the student dropouts for detection purpose at DTU.

Acknowledgements

This master's thesis was done in collaboration with the Department for Study Affairs at the Technical University of Denmark. I would like to thank Merete Reuss, Annette Elmue, Camilla Nørring and Christian Westrup Jensen who provided the data and fruitful insight to the current systems in use.

I would like to thank Bjarne Kjær Ersbøll who offered this topic when my initial project turned out to be infeasible. Also a thank you for the support while writing this thesis.

A special thank you to my supervisor Murat Kulahci for his invaluable discussions and help during the project.

Last but not least, thank you for Rune Juhl for his comments on my thesis and emotional support.

Contents

CHAPTER 1

Introduction

1.1 Overview of Student Drop Out in Denmark

Not all students who begin a university education graduates. According to [15] around 30-40% of the students drop out or change their subject. This high dropout rate cost 200 millions Danish kroner for the taxpayers.

Recently the reasons for dropping out have been discussed. The Danish Institute of Governmental Research (in Danish: Anvendt KommunalForskning) made a large survey [14] trying to identify the reasons of early students' drop out. Data from year 2000-2005 was analysed in this survey and several reasons were identified for early students' drop out. One of most significant reason for completing the studies is establishment status. That is married persons with children are more likely to finish their education. Also students who previously completed a higher education. However, those who previously tried and failed to complete a higher education are more likely to drop out once again. Another important reason for drop out is an ethnic minority background. According to [18] a student with an ethnic minority background is 2.6 times more likely

to drop out. The reason is discrimination at the study institutions and a lack of possibilities to study at home. The last reason is identified as performance at the beginning of studies.

Some universities are taking actions to identify students with higher risks for dropping out. One of the suggested solutions is pre-admission interviews. This allows to find the really motivated students. However, this approach is highly costly at around 3000 Danish kroner per interview [15].

The objective of this master's thesis is to research and build a model based on the general data obtained from the application and performance data from each semester to identify students who are most likely to drop out. Potential dropouts could be interviewed to reinstate their motivation. The identification of the potential dropouts could save money and provide the opportunity for more motivated students. Thus increasing the effectiveness of the universities and helping to achieve the national goal that half of the young population would earn a higher education.

1.2 Current System at Technical University of Denmark

Camilar Nørring from the Student Counselling (Studievejledningen) at the Technical University of Denmark (DTU) from the Department of Education and Study Affairs (Afdelingen for Uddannelse og Studerende) gave a short introduction to the remedies suggested by the Ministry of Science, Technology and Innovation (Ministeriet for Videnskab, Teknologi og Udvikling)(VTU) and how DTU have implemented it.

According to the VTU the universities must contact students who are 6 months (equivalent to 30 ECTS credits) behind. In other words, students who have not passed any credits for one semester. These students must be offered counselling. However it is not specified how it should be. For the students who are more than 12 months (60 ECTS credits) behind must be offered an individual meeting with a counsellor.

Students at DTU who are behind by more than 6 months and less than 12 months (30-59 ECTS credits) gets an invitation by e-mail for an individual talk with a counsellor. Those who are more than 12 months (60 ECTS credits) behind receive an official invitation by mail. Furthermore, public workshops on study planning, how to avoid delays and how to get back on track are organised every semester by the study counsellors. All students are invited to participate.

1.3 Goal of this Master's Thesis

The goal of this master's thesis is to re-evaluate the current student dropout detection system using principles of quality control, application and performance scoring techniques. To compare several performance scoring techniques and evaluate whether an improved method can be suggested. Also, identify significant characteristics for dropout student detection. Finally, compare findings with the current system.

CHAPTER 2

Principe of Quality Control

Experience have shown how important it is to keep track on the students' performance to proactively help students from delays or an eventual drop out. Student monitoring is a continuous process and the basic philosophy could be adopted from statistical process control. The basic principles of process control are discussed in [16].

2.1 Reasons for Process Variations

The are two main branches of variability appearing in a process. The first type is caused by the process itself. It does not matter how well the process is designed, there will always be natural variability. Whole of natural causes in the process is called *stable system of change causes* and the process that operates only with chance causes of variation present is said to be in *statistical quality control*. Second type of variability sources are: improperly adjusted or controlled mechanisms, administration errors or defected raw material. Whole of unnatural courses are called *assignable*

causes. Variability emerged from assignable causes usually is represented as unexceptionable outcome. A process that is operating in the presence of the assignable causes is said *to-be-out-of control*.

It is noticed, that students who wants to graduate on time tend to study in a more stable manner than those unsure about their wishes and choices. However, sometimes even good students might go through a period where the studies are quite difficult. This can be because of study, university and personal matters. Study or university related matters could be:

- changes at the university education system
- changes at the teaching methodology
- more difficult subjects in some period of studies then usual
- ...

Personal matters could be:

- changes in personal life
- health issues
- lack of concentration for any number of reasons
- ...

These reasons will degrade the overall performance of a student who will however eventually graduate. The assignable causes influence student performance much stronger and eventually the student will drop out. As previously discussed in chapter 1 on page 1 there are two types of assignable causes: one related to university and studies and the other to personal matters.

2.2 Statistical Basics of the Control

Control charts are widely used in the manufacturing industry. The idea is to take a sample of the manufactured product and compare it with the target measurements. The changes over time are observed. Using different statistical techniques bounds can be set for the measurement variations. While monitoring changes in the measurements, problems can be detected and action can be taken to prevent producing non qualitative products.

Classical control charts in student performance monitoring is not suitable. There are many variables that can be used for monitoring. This problem is quite usual in process control, thus combined variable charts are used. It is not know which performance measurements should be used in student performance monitoring. In this master's thesis a new performance monitoring system is suggested. Students performance is measured every semester, which is the *sampling time*. Instead of charts presenting the overall production performance chart status update will be used. When the model detects a student not performing good enough to graduate the student's status is changed from "pass" to "drop out". When the student is classified as "drop out" the assignable causes must be investigated so the student can be helped to perform better in the future. A simplified model scheme is presented in fig. 2.1. As it can be

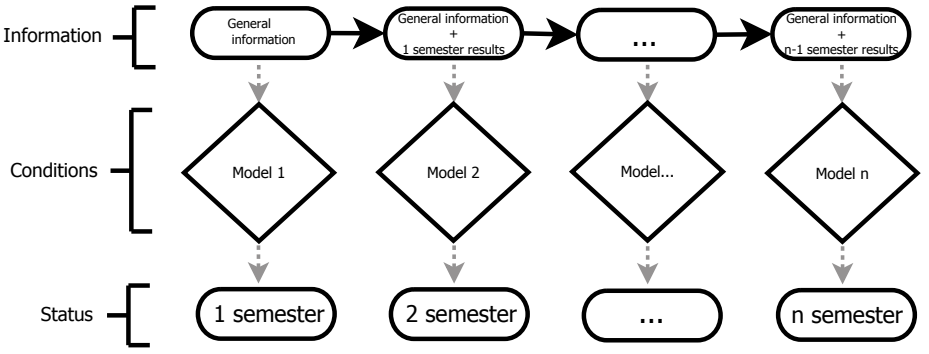


Figure 2.1: Simplified model scheme.

seen in fig. 2.1 the model consists of n models. Every model analyse the general information and student performance records from the previous

semester. The outcome of every model is the predicted student status after the mth, that $m = 1, ..., n$ semester.

In process control where classical control charts are used it is often used probabilities of error type I and II. The *type I error* is indicating the probability that the process is out of control when it is actually in control. The *type II error* is indicating the probability that the process is in control when it is actually out of control. The sum of these errors is always equal to one thus decreasing one type error will increase the other. It is always important to find the balance between these errors. For example, if the type I error is very high false alarms will be occur too often. Likewise a high type II error will give the impression that the process in control. In the suggested model, the type I error is the classification of students as dropouts when they actually do pass. This error is also called *false alarm*. Error type II is when a student is classified as passed but actually is a dropout.

The most important task of process control is to stabilise and improve the process. To achieve this three basic ideas can be used:

1 Most processes are not in total statistical control.

2 Actions must be taken as soon as it is identified that process is out of control.

3 When actions are taken after control has been lost and an investigation of the assignable causes is performed and the causes notified, a trend might be noticed.

As soon as a drop out student is identified using these three basic ideas an investigation must be performed. This will help to identity the problems the student is facing and the university may be able to help him. Collecting the assignable causes might suggest what actions could be taken to prevent students from droping out and also improve the model.

2.3 Phase I and Phase II of Control Methods Application

Model building takes to distinct phases. In *phase I* the general process behaviour is investigated. Usually it is assumed that the process is out of control and a general investigation is conducted to bring the process back to control. Usually already collected data is investigated. In *phase II* the suggested model is implemented and the process is followed online. During this phase it is assumed that the process is already in control. New adjustments might be implemented.

This master's thesis is like a phase I. Already collected data is investigated. A general model is going to be suggested. Phase II is not a part of this thesis and is suggested as future work. The phase II is as important as the phase I. In this case some of the students' behaviour might change due to the close monitoring of students. It might be necessary to re-estimate the models or even redo the entire modelling.

CHAPTER 3

Types of Scoring

3.1 Application Scoring

Application scoring is wildly used in marketing. The aim of application scoring methods are to identify customers who a company can offer their products or services to. For example resellers can target their advertisements for new products to a specific segment of the costumers. That is those most likely to buy new products. Banks can use the information from the applications for identifying those customers who are most likely to keep up with their repayments. In bank terminology these methods are usually called credit scoring. In this thesis application scoring method is used to identify students who are most likely to drop out based on their application information even though these students are already enrolled.

Usually the predictive variables are called *characteristics* and the value or class they are assigned is called the *attributes*. The aim of application scoring is to build a model that could predict attribute classes using the available characteristics. There are several techniques developed over the

years and a broad overview is presented in [6]. Classical methods used these days are *discriminant analysis* and *linear regression*. Discriminant analysis might have problems with data that do not follow the normal distribution or groups that do not have a discrete form. Yet there are some proposed solution for these problems. *Linear regression* can be used for two class identification. The benefit of linear regression is that it can be used as application scoring and *behavioural scoring* which will be more introduced in section 3.2. Closely related to linear regression is *logistic regression* that may be also used in two discrete class classification. It is noticed that it do not perform significantly better than linear regression. Another classical method is *decision trees*. The advantage of this method is that non-linearity and interaction can be included in the model. More on CART in section 4.3 on page 18.

Other methods as *mathematical programming*, *neural networks*, *nearest neighbour* can be also used. Mathematical programming suffer from problems with linear relations among the characteristics. Neural networks used with association rule is widely discussed in [9] and presented as one of the advanced classification methods. Yet using this method the interpretation of the model can be lost. Nearest neighbour avoid the distribution change problem, though this method is highly computational expensive.

3.2 Performance Scoring

In many cases it is important to follow the performance of the customers to make sure that they will perform as expected at the application scoring level. This method is called *performance* or *behavioural scoring*. Application scoring is a more general technique that is done at the first step. Performance scoring evaluates existing customers based on the similar principles as application scoring. As in application scoring customers can be classified to the same or a different performance group.

There are several common techniques between application and performance scoring: *linear/multiple discriminant analysis*, *linear regression*, *logistic regression*, *neural network* and *support vector machines* all discussed in [9] and [8]. The same techniques can be used in application scoring. In

the discussion of these papers the linear discriminant analysis do not perform any better than any other methods. Support vector machine has issues with parameter selection. As in application scoring neural networks performs best.

CHAPTER 4

Techniques of Scoring

4.1 Logistic Regression

Logistic regression (LR) is one of the basic methods for the classification problem. The method is defining a non-linear relationship between the dependent and independent variables. LR can be used as a classifier with two or more classes. In biostatistical applications as in survival analysis LR is widely used due to its good performance in two class problems. The theory is discussed in [7].

4.1.1 Principles of the Logistic Regression

LR uses posterior probabilities of the K classes via a linear function in x. The model is naturally constrained so the probabilities are in the interval $[0, 1]$ and sum to 1. The model is defined by $K - 1$ logit transformations

of the ratio of probabilities of two classes

$$\begin{aligned}
\log \frac{Pr(G=1|X=x)}{Pr(G=K|X=x)} &= \beta_{10} + \beta_1^T x \\
\log \frac{Pr(G=2|X=x)}{Pr(G=K|X=x)} &= \beta_{20} + \beta_2^T x \\
&\dots \\
\log \frac{Pr(G=K-1|X=x)}{Pr(G=K|X=x)} &= \beta_{(K-1)0} + \beta_{K-1}^T x.
\end{aligned} \tag{4.1}$$

Class K is arbitrarily chosen as a reference class and appears in the denominator in eq. (4.1). Hereafter the following notation for the probabilities is used: $Pr(G=k|X=x) = p_k(x;\theta)$, where $\theta = \{\beta_{10}, \beta_1^T, ..., \beta_{(K-1)0}, \beta_{K-1}^T\}$.

4.1.2 Estimating the Logistic Regression Model

Fitting logistic regression to data is solved by maximizing the conditional likelihood of G given X. The log-likelihood for N observations is

$$l(\theta) = \sum_{i=1}^{N} \log p_{g_i}(x_i;\theta). \tag{4.2}$$

Maximizing the log-likelihood for the two classes problem is done by the *iteratively re-weighted least squares* method. It is based on the Newton-Raphson algorithm which requires the Hessian matrix. Each iteration performs a minimization and update the β estimate

$$\beta^{new} \leftarrow \arg\min_{\beta} (z - X\beta)^T W (z - X\beta). \tag{4.3}$$

W is an NxN diagonal matrix of weights where the ith diagonal element is $p(x_i;\beta^{old})(1 - p(x_i;\beta^{old}))$. z is the adjusted response

$$z = X\beta^{old} + W^{-1}(y - p). \tag{4.4}$$

Usually the best starting values is $\beta = 0$ and in most of the cases the algorithm will converge.

4.2 Principle Component Logistic Regression

Linear regression face problems with collinearity among variables. One option is to remove insignificant variables and perform linear regression in a smaller variable space. To select variables a *principle component analysis* (PCA) can be used [4]. [1] discuss two methods *principle components logistic regression* (PC-LR) and *partial least-square logistic regression* (PLS-LR) for dimension reduction in a logistic regression setting. As it is noticed in this paper, it can be expected that PC-LR will give better estimates for regression coefficients. Thus PC-LR will be used in this thesis.

4.2.1 Principles of Principle Component Analysis

The principle components is the best linear approximation of the space $\Re^p$ in the smaller space of the dimension q such that $q \leq p$. The linear approximation can be expressed as

$$f(\lambda) = \mu + V_q\lambda. \tag{4.5}$$

μ is the *location vector* which is the origo of the new coordinate space in the original space $\Re^p$. The q orthogonal unit vectors spanning the subspace is arranged column wise in the *loading matrix* V_q. The matrix is pxq of size. It is how the PCs are weighted by the original space. λ is a vector with q elements which is the point in the subspace - also called the *score*. The scores from each observation is arranged in the *score matrix*.

The principle components are arranged by how much variance they explain with the first PC explaining the majority of the variance. Lowering the dimension is essentially selecting a number of PCs usually based on two rules: accumulated variance explained by the chosen number of PCs and if adding one more PC will not increase the explained variance significantly.

4.2.2 Principles of Principle Components Logistic Regression

As described above the loading matrix is the data representation in new - usually smaller coordinate system. The loading matrix can be used instead original data matrix in the logistic regression. Thus, the PC-LR model is build on the most important variables which are not collinear.

4.3 Classification and Regression Tree

Classification and regression trees (CART) are widely used in application scoring and survival analysis. Using trees and splines in survival analysis is discussed in [10]. The paper outlines a great advantage that survival groups are classified according to similarities which can provide some insight to the underlying reason for the classification. Thus can also be used to identify the most important variables. CART can also do variable selection based on the covariance matrix or complexity parameters and handles missing values directly. The principles of CART are presented in [11, pp. 281–313] and [7, pp. 256–346].

4.3.1 Principles of Classification and Regression Tree

CART is a non-parametric method based on a *recursive partitioning* algorithm that step-by-step constructs the decision tree. CART is a supervised learning technique asking hierarchical boolean questions. There are several ways of model presentation. Imagine the binary problem with two independent variables X_1 and X_2 and one response variable Y with two groups. Figure 4.1 on the facing page shows a scenario with three splits. The first split is called the *root node*. A *node* is subset of the set of variables. A node without a split is a *terminal node*. All terminal nodes are assigned a class label. A node with a split is consequently called a *non-terminal node*. Non-terminal nodes are also called *parent nodes* and is divided into two *daughter nodes*. A single-split CART is called *stump*. The set of all terminal nodes of the CART is called a *partition* of the data. The partition of the data in the tree in fig. 4.1 on the next page is presented

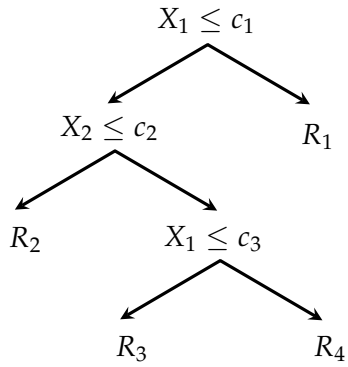


Figure 4.1: The tree example.

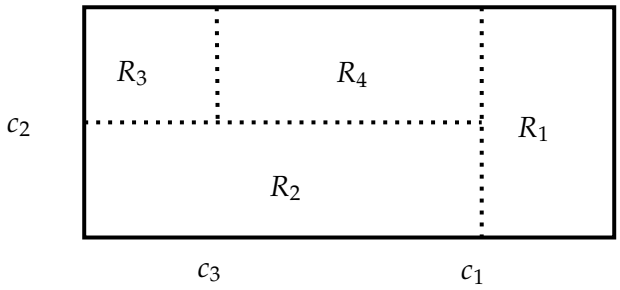


Figure 4.2: Data set division.

alternatively in fig. 4.2. Each region represents a terminal node. That is the region $R_2 = X_1 <= c_2, X_2 <= c_2$ corresponds to the terminal node R_2.

4.3.2 Growing a Tree

The principles of growing a tree is to find binary splits that will separate the data in groups. The process is continued until some minimal terminal node size is reached. Already separated regions are defined as:

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m) \tag{4.6}$$

M is the total number of regions, c_m is the average response in region m. The optimal $\hat{c}_m$ is:

$$\hat{c}_m = \text{average}(y_i | x_i \in R_m) \tag{4.7}$$

The best partition is found by solving a sum of squares minimization problem. Variable j with the split point s divides into two region k and l. These two regions can be defined as:

$$R_l(j,s) = \{X | X_j \leq s\} \quad \text{and} \quad R_k(j,s) = \{X | X_j \geq s\} \tag{4.8}$$

Then the minimization problem is:

$$\min_{j,s} \left(\min_{c_l} \sum_{x_i \in R_l(j,s)} (y_i - c_l)^2 + \min_{c_k} \sum_{x_i \in R_k(j,s)} (y_i - c_k)^2 \right) \tag{4.9}$$

Where the solution is:

$$\hat{c}_l = \text{average}(y_i | x_i \in R_l(j,s)) \quad \text{and} \quad \hat{c}_k = \text{average}(y_i | x_i \in R_k(j,s)) \tag{4.10}$$

At every node the variable split that will minimise the cost function the most is selected.

4.3.3 Node Impurity Measure

Let's denote $|T|$ as the number of terminal nodes in a sub-tree T. The sub-tree T of the initial tree T_0 can be obtained by collapsing non-terminal nodes. If

$$\hat{c}_m = \frac{1}{N_m} \sum_{x_i \in R_m} y_i$$

then squared-error node impurity measure is defined as

$$Q_m(T) = \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - \hat{c}_m)^2. \tag{4.11}$$

In the case of a categorical response variable a different impurity method should be used. The proportion of the class h in node m is used

$$\hat{p}_{hm} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = h). \tag{4.12}$$

The observations are classified as group k in node m when

$$k(m) = argmax_k \hat{p}_{mk}. \tag{4.13}$$

There are several different impurity measures used in practice: misclassification error, Gini index, cross-entropy or deviance and Twoing rule

$$\begin{aligned} Q_m(T) &= 1 - \hat{p}_{mk(m)}, \\ Q_m(T) &= \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_m k), \\ Q_m(T) &= -\sum_{k=1}^{K} \hat{p}_{mk} log \hat{p}_m k, \\ Q_m(T) &= \frac{P_l P_r}{4} (\sum_{k=1}^{K} |\hat{p}_{k|m_l} - \hat{p}_{(k|m_r)|})^2. \end{aligned} \tag{4.14}$$

P_l, P_r are the probabilities of right and left nodes. The Gini index method is searching for the largest group in the data and tries to separate it from the other classes. Twoing rule performs the separation in a different manner. It will search for two groups, that each of them will ad up to 5o % of data. Cross-entropy works as the Twoing rule by searching for similar splits. Gini index and cross-entropy are more sensitive to changes than the misclassification rate.

4.3.4 Pruning

One of CART's disadvantages is overfitting. Letting the tree grow until there are no more splits results in very large trees with many small groups. Restricting the growth in size can lead to not capturing the underlying structure. One solution is tree pruning. The pruning procedure has three main steps. First, a large tree is grown until every terminal node has no

more that specified number of observations. Next the misclassification parameter $Q(T)$ is calculated for every node. Finally the initial tree is pruned upwards towards its root node. At every stage of the pruning the estimate of the cost-complexity parameter C_α is minimized. The *cost-complexity pruning* method penalize large trees for their complexity.

$$C_\alpha = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T| \tag{4.15}$$

α is called the complexity parameter. Small values (around 0) give small or no penalty while large values give large penalty. For any α there can exist more than one minimizing subtree. However, a finite set subtrees can be obtained. Every subtree corresponds to a small subinterval of α. To find the finite set of subtrees $C_\alpha(T)$ must be minimized. To do so the method *weakest link pruning* is used. That is removing non-terminal nodes that produce the lowest increase in $\sum_m N_m Q_m(T) + \alpha|T|$. To obtain the smallest unique subtree for every α the following condition must be satisfied.

$$\texttt{if } C_\alpha(T) = C_\alpha(T(\alpha)) \texttt{ then } T > T(\alpha) \tag{4.16}$$

The solution of the conditions is finite set of complexity parameters, which corresponds to nested subtrees of maximal tree:

$$0 = \alpha_0 < \alpha_1 < \alpha_2 < ...\alpha_M$$
$$T_{Max} = T_0 > T_1 > T_2 > ... > T_M$$

Then in T_1 the weakest-link node $\tilde{m}$ is determined. Then tree T_{m_1} is pruned with the root node as $\tilde{m}$. This gives subtree T_2. This procedure is performed until T_M.

4.3.5 The Best Subtree

There are several tests to identify the best subtree. The mainly used methods are not just used in the CART for selecting the best subtree but are the general statistical ideas of *independent test set* and *cross-validation*. The idea of the independent test set is to divide the data in to two sets with the proportions of 50%/50% or 80%/20%. The larger set is used for

training the model and smaller to test the model. The overall performance of model is defined from the model results using test set. For best subtree selection a tree is build using the training set. Then the set of subtrees is defined. Using the test set the misclassification rate of every subtree is calculated. The subtree with the smallest misclassification rate is chosen.

The cross-validation method can be used even when the data set is small. Data set is divided in k subsets also called folds of equal size, usually 5-10 observation in every fold. The model is trained using $k-1$ sets and tested on the remaining set. This is performed k times so every subset would be used once for testing. The overall performance is the average of the k test errors. In the selection of the best subtree k trees are gowned using v^{th} learning set, were $k = 1, 2, ..., k$. Then the complexity parameter α values are fixed and the best pruned subtree of T^v_{max} is found. Then the v^{th} test set is used in every $T^v(\alpha)$ tree to define the misclassification ratio. Then the overall misclassification rate for every α is defined and the α with the minimal misclassification ratio is chosen.

4.3.6 Disadvantages and Advantages of CART

There are several issues with CART. One of the problems is instability of the trees due to variance. The reason lies in their hierarchical nature and even a small change in the data can result a different tree. *Bagging* is used to as a remedy by averaging the results of many trees to reduce the variance. Bagging will be discussed in section 4.4 on the next page. A second disadvantage is the complexity of the trees that may lower the prediction power. It is usually solved by pruning. The third disadvantage is lack of smoothness and difficulty in capturing additive structure. This problem also has a solution: *multivariate adaptive regression splines* (MARS). This will be further discussed in section 4.6 on page 27.

One of the main advantages of CART is interpretability. Trees are easily explained and understood by end-users. What is more, it is easy to implement in any kind of programming language only requiring the *if* statement.

4.4 CART and Bagging

As mentioned in section 4.3 on page 18 using CART on data with large variance the tree becomes very unstable. This section will discuss the method using the same classification and regression trees to stabilize the solution.

4.4.1 Principles of Bagging Using CART

The bootstrap mean can be approximated as a posterior average. Say a data set is divided into a training and a test set. The training set is denoted $Z = (z_1, z_2, ..., z_N)$ where $z_i = (x_i, y_i)$. Randomly draw B samples with replacement from the training set. B is the size as the original training set. In this way, $Z^{*b}, b = 1, 2, ..., B$ separate training sets are obtained. For each new set estimate the model to get the predictor $\hat{f}_{bag}(x)$. The average of the predictions of each training set is bagging

$$\hat{f}_{bag}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^{*b}(x). \tag{4.17}$$

It is expected for $B \to \infty$ that the estimate gets closer to the true value. However when the function is non-linear or adaptive the estimate will differ from the true value.

Bagging can be also used in CART. In the K-class case, bagging can be performed in the following way. First, number of trees with replacement are build. Second, decision of the predicted class can be estimated using: $\hat{G}_{bag}(x) = \arg\max_k \hat{f}_{bag}(x)$. This means, that class with the highest probability, is the estimated class.

4.5 Random Forest

Another technique that is closely related to CART is *Random Forest* (RF). The RF algorithm was first presented by Leo Breiman and Adele Cutler

[2] and [3].

4.5.1 Principles of the Random Forest

As with bagging, RF grows a lot of trees and each tree casts a "vote" for the class. The difference between bagging and RF is the algorithm for growing trees. The RF algorithm has three main steps:

1 Randomly draw with replacement from the training set a new set which is used for growing a tree.

2 Define $MTRY$ such that is smaller than the number of variables. In each split $MTRY$ random variables are selected. The best split variable is found among those $MTRY$ variables. $MTRY$ is constant through the procedure.

3 Repeat until reaching a pre-selected maximal number of trees $NTREE$.

RF grows many trees, but do not prune any. $MTRY$ should be around

$$mtry = \lfloor \sqrt{\text{number of variables}} \rfloor. \tag{4.18}$$

It is important not to choose $MTRY$ too big as it will increase the *correlation* between the trees and the *strength* of a tree as it may reappear. Highly correlated trees and high strength of individual trees increase the error of the random forest too.

4.5.2 The Out-Of-Bag Error

One of the main advantages of RF is that it should not overfit. It is not even necessary to use cross-validation or an independent test set in the model building process. It is built into the method by resampling the training data with replacement. One third of training data is left out and the model is build on remaining data. After the model is build it is tested

on the one third of data. After testing all the trees the element is assigned a class that got the most votes. The *out-of-bag error* (OOB error) obtained counting misclassification using different number of trees. This type of error is unbiased.

4.5.3 Variable Importance

OOB error can be used to compute the importance of the variables. The importance of the variable is calculated by changing its value in the tree. Out-of-bag data is used again to calculate changes error with changed variable value. The average changes in classification across the forest is called the *mean decrease in accuracy* (MDA) measure.

There is another variable importance measure called *mean decrease in Gini index* (MDG). This shows the average decrease of the Gini impurity measure across the forest for each variable. According to [17], when the measurements are on different scale and if there is correlations within the variables then MDA will give more stable scorings than MDG. It is noted the MDG can be better in some informatics applications.

In the case of many variables these measurements can be used to reduce the dimension. At first, build a model with all the variables. Then select the important variables and redo the model only using those variables.

4.5.4 Missing Values

There are several theoretical approaches for how to handle missing data. For example, use median of the variable in the class to fill non categorical missing values. Also, the proximity matrix can be used to replace the missing values. However, in [12] handling missing data is not implemented, but there is a workaround. A RF is basically many CART trees. CART trees has the property to “force” data points to go though the tree even with missing information.

4.6 Multivariate Adaptive Regression Splines

As mentioned in section 4.3.6 on page 23 CART lacks smoothness and thus *multivariate adaptive regression splines* (MARS) could be used. Although MARS is using a different technique for the model building it resembles CART.

4.6.1 Introduction to MARS

MARS is relating Y to X through the model

$$Y = f(X) + \epsilon \tag{4.19}$$

where ϵ is standard normally distributed and $f(x)$ is a weighted sum of M basis functions

$$f(X) = \beta_0 + \sum_{m=1}^{M} \beta_m h_m(X) \tag{4.20}$$

where h_m is a basis function in C or a product of several of these basic functions.

$$h_m(X) = (X_j - t)_+ \tag{4.21}$$

The collection of basis functions is C

$$C = \{(X_j - t)_+, (t - X_j)_+\} \tag{4.22}$$
$$t \in \{x_{1j}, x_{2j}, ..., x_{Nj}\} \quad \text{and} \quad j = 1, 2, ..., p.$$

Although every basis function only depends on a one X_j it is a function over all input space $\Re^p$. It is a hinge function.

Model building consist of two parts. First, using a forward-stepwise process large linear model is build. The process starts from the intercept β_0 ($h_0(X)$), and step by step adds another hinge function eq. (4.21) to minimize the residual error

$$MSE(M) = \sum_{i=1}^{n} (y_i - f_M(x_i))^2 \tag{4.23}$$

The full model will overfit the data. The second part is using a backwards-stepwise procedure to delete terms that gives the smallest increase in residual squared error. To find the optimal number λ of terms in the model, the generalized cross-validation can be used. The criterion is

$$GCV(\lambda) = \frac{\sum_{i=1}^{N}(y_i - \hat{f}_\lambda(x_1))^2}{(1 - M(\lambda)/N)^2}. \tag{4.24}$$

$M(\lambda)$ represents the *effective number of parameters*.

4.6.2 CART and MARS Relation

Although MARS has a different approach than CART, MARS can be seen as a smooth version of CART. Two changes must be done to make MARS be as CART. First, the hinge functions must be changed to indicator functions: $I(x - t > 0)$ and $I(x - t \leq 0)$. Second, multiplication of two terms must be replaced by interaction, and therefore further interaction are not possible. With these two changes MARS becomes CART at the tree growing phase. A consequence of the second restriction is that a node can be only have one split. This CART restriction makes it difficult to capture any additive structure.

CHAPTER 5

Data

Data from four study programs were provided by DTU. At first three programs were given

- Design and Innovation
- Mathematics and Technology
- Biotechnology

The three datasets all had different drop out rates. However, the number of dropouts per semester were too low. Therefore, the Biomedicine program was added to the analysis.

As seen in fig. 5.1 on the following page the highest drop out rates are in Mathematics and Technology as well as Biotechnology programs. The drop out rates reach around 30-40%. The lowest drop out rate is in the Design and Innovation program, around 10%.

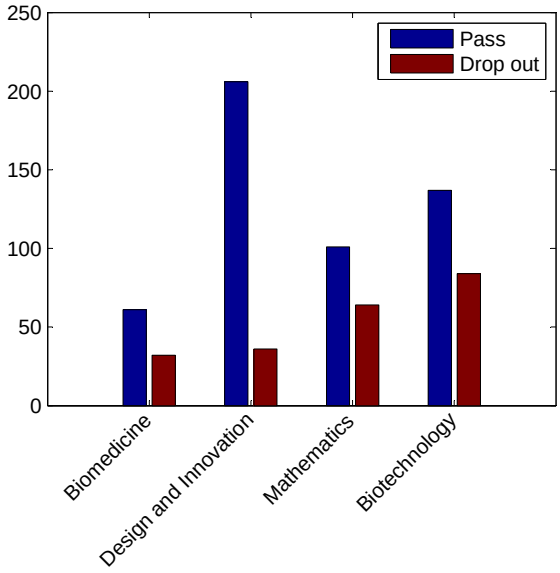


Figure 5.1: Histogram of passed and drop out students in every program.

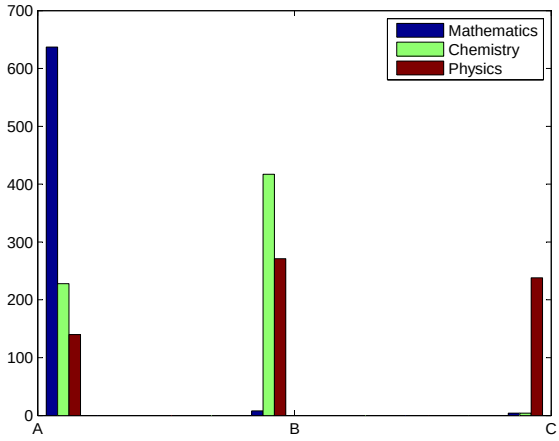
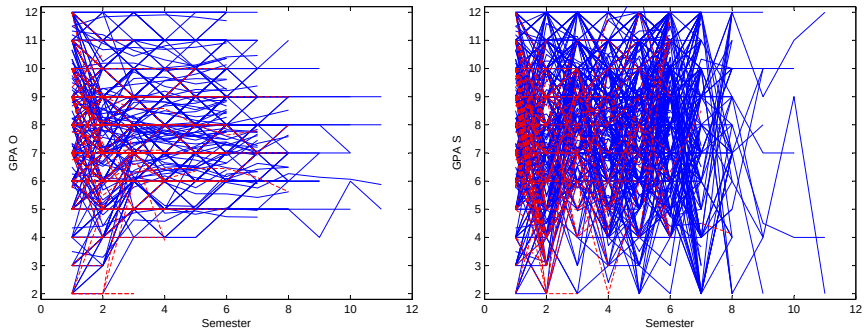


Figure 5.2: School exam level distribution.

There are two source of information about the student. One is from their application and the second is they perform after each semester. When applying at DTU a student provides the following information: age, sex, nationality, name of school, type of entrance exam, school GPA, the exam level and grade of the subjects mathematics, physics and chemistry. In fig. 5.2 on the preceding page can be seen, the histogram of chosen school exam levels. DTU's records provide information about the courses every student sign up for. For each course the mark, date of assessment, ECTS credits and at which semester the course was taken is recorded.

From the records additional performance measures were created. For every student the ECTS credits taken each semester is summed. Also the ECTS credits that student actually passed. The accumulated ECTS after every semester since enrolment is summed. In addition to the credits measurements the GPA for every semester and the overall GPA was calculated. As seen in fig. 5.3 the overall GPA becomes steady after the

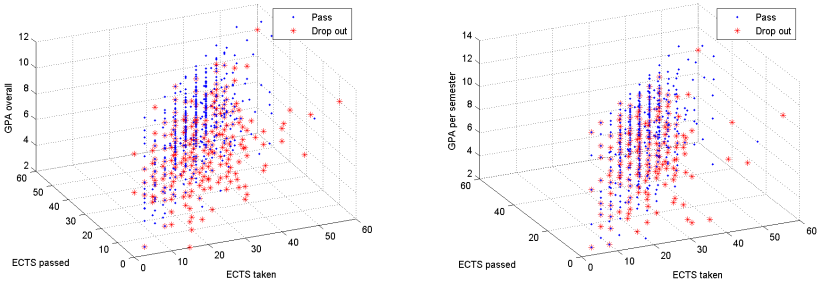


(a) GPA overall (b) GPA of every semester

Figure 5.3: GPA changes over the study period. Red - dropouts, blue - pass students.

third semester while the GPA of every semester can vary a lot. Logically the GPA of every semester depends on the specifics of the study program and the student's personal life. The specifics is how one semester can be more difficult than another. The figure also shows how students with very high grades might even drop out.

It is most natural to expect that a good student would pass all the courses they are assigned and would continue to get good marks. Equally a bad student would not be able to pass all the registered courses and consequently get poor grades from the courses they do pass. Figure



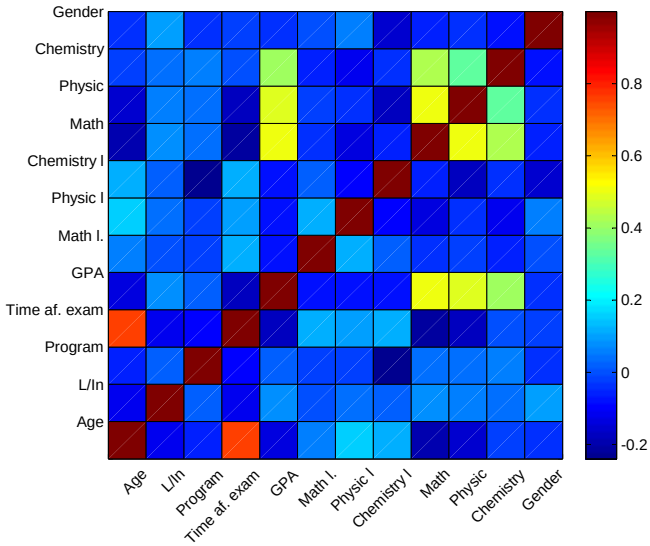
(a) GPA overall (b) GPA of every semesester

Figure 5.4: GPA measures vs. ECTS taken measures vs. ECTS passed measures. Red - dropouts, blue - pass students.

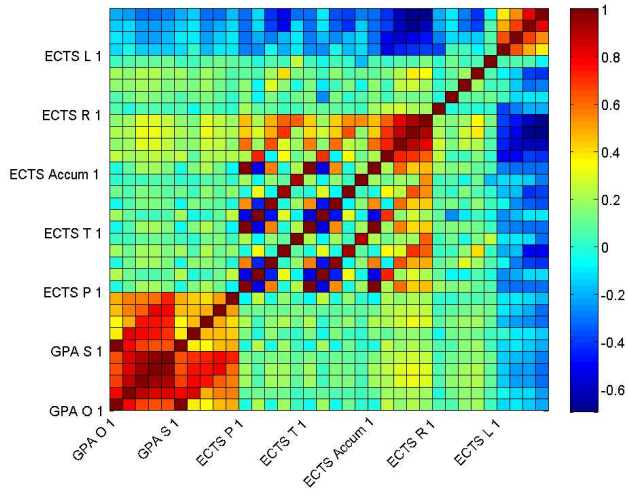
fig. 5.4 only shows the above expectation partly. On the left figure it can be seen that there is a cloud of red stars in the lower right part of the plot that represents dropouts. However, there are so many dropouts who passed all the courses they took even with high grades. In fig. 5.4b clouds of passed and drop out students are even more mixed. Though, some relation between passed and taken ECTS credits is observed. The ratio of these two measures will be included in the models.

In addition to all the performance characteristics, one more was included called ECTS L (ECTS late). It is an indicator for whether the student is behind by more than 30 ECTS credits. This indicator was included to check whether the current system is reasonable.

To get an understanding of the inter-correlation between all the indicators the correlation matrix was computed. Plotted in fig. 5.5a on the next page shows the highest correlation is between time since the qualifying exam and age. There is also a very high correlation between school GPA, chemistry, physic and chemistry exam grades. A negative correlation between age and mathematics exam grade is also observed.



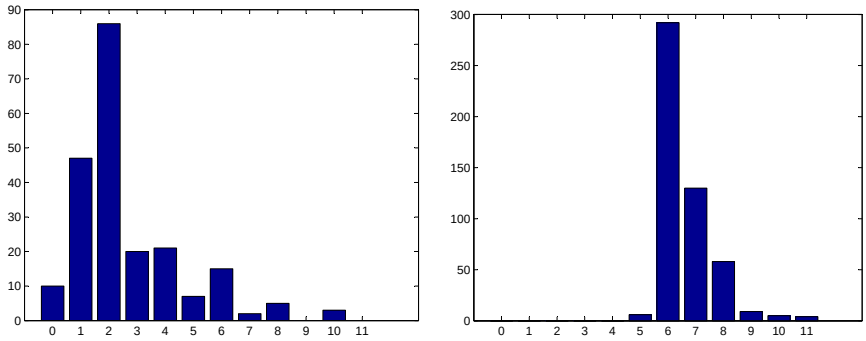
(a) Correlation among application data



(b) Correlation among performance data

Figure 5.5: Correlation among characteristics

Figure 5.5b on the preceding page shows very strong correlation between the GPA overall and GPA of each semester. Correlation of GPA overall after two semester becomes very strongly correlated indicating that GPA overall becomes stable after the second semester. Different situations occur with the GPA of every semester. It varies form semester to semester. For the passed, taken and accumulated ECTS credits measures it can be seen that the correlation varies a lot for the first three months. However, the first and second semesters are negatively correlated, while the first and third semesters are positively correlated. This represent an instability of the students progress during the first three semesters.



(a) Drop out students distribution. (b) Pass students distribution.

Figure 5.6: Distributions of drop out and pass students.

Figure 5.6 shows the highest number of dropouts occur during the first and sixth semester, while most of the students graduate after the sixth-eighth semester. In the further analysis performance data from the first to the fifth semester will be analysed.

CHAPTER 6

Modelling

6.1 Modelling Techniques and Methods

The data was divided in to three parts in two steps. First, the it was divide in two sets with the ratio 1 to 9. The sets were drawn randomly and the proportion of dropouts and passed students are approximately similar in all the sets. The smaller part was used for the final model validation using different techniques. The larger set was used for training and testing the individual semester models. This set was further divided in a training and test set with the ratio 8 to 2. The sets were draw with supervision. In every semester there was the same drop out ans pass student ratio (8:2) in training and test set.

In this thesis six techniques are compared: logistic regression, PC-LR, CART, bagging CART, RF and MARS. For each technique eight semester models were build. The first three models all aim at predicting the dropout status before the first semester.

Model 1 corresponds to application scoring. To build this model personal

information from the application was used: school GPA, level and grade from mathematics, chemistry and physic, age, gender, nationality and time since taking the last exam at school. By analysing all drop out and passed students the model can raise an alert to the university about students that in general will potentially drop out.

Model 2 is based on the same information as in model 1. Only the students who drop out before even beginning their studies or drop out after first semester were analysed together with the students who graduated.

Model 3 was build using the same information as in models above. Only the student who dropped out after the first semester of courses were analysed with the students who graduated.

The following models aim at predicting the dropout status after the second to sixth semester.

Model 4 is for status prediction after the second semester. This model was build using personal information and performance information from the first semester: GPA of the first semester, taken and passed courses and the ratio of passed and taken ECTS credits after first semester. The indicator for being behind by more than 30 ECTS credits was included. Students who dropped out after the second semester together with the students who graduated were analysed.

Model 5 is for status prediction after the third semester. This model was build using personal information and performance information from the first and second semester. Students who dropped out after third semester together with the students who graduated were analysed.

Model 6 ...

Model 7 ...

Model 8 is for status prediction after the sixth semester. This model was build using personal information and performance information from the first to fifth semester to predict status after sixth semester.

Students who dropped out after the sixth semester together with the students who graduated were analysed.

All these models were built and tested independently of each other. Then the models were tested on the training data to see how the perform in regard to each other. This means, that the models were executed in the order described above. Unique dropouts not predicted by any of the preceding models were counted. That is if student 11 was classified as a dropout by model 1 and 2 then he is only counted for model 1. The model with the highest prediction number were selected. These models constitute the final model which was tested on the small validation set created by the first split.

Models were struggling to find good separations. For this reason training data was rounded. ECTS measures of taken, passed and accumulated was rounded that the value of module after division of five would be 0. GPA measure of overall and semester were rounded to the nearest integer number.

For all techniques except the logistic modelling the predictions are grouped in four classes. For the logistic modelling in five classes. If true status is "drop out" and the predicted class is the same it is classified as "DD". If true is "pass" and classified as such then it is class "PP". If the true status is "drop out", but predicted as "pass" then it is classified as "DP". In the true status is "pass", but predicted as "drop out" then it is classified as "PD" which is a false alarm. Due to the properties of the logistic regression the students with missing values cannot be predicted. Thus there is one additional class: "Not classified".

For each semester model several ratio measurements were calculated to get an overview of the model performance. The number of predictions in each training and testing set was used for these ratios:

$$\text{Misclassification ratio} = \frac{PD + DP}{DD + DP + PP + PD} \tag{6.1}$$

$$\text{Drop out misclassification ratio} = \frac{DP}{DD + DP + PP + PD} \tag{6.2}$$

$$\text{Drop out ratio in all misclassification} = \frac{DP}{DP + PD} \tag{6.3}$$

The *Misclassification ratio* is the total number misclassification among all the predicted observations. The *Drop out misclassification ratio* and the *Drop out ratio in all misclassification* represents dropouts not detected in all observations and in all misclassifications respectively.

6.2 Logistic Regression Modelling

6.2.1 Logistic Regression Technique

The modelling was performed in MATLAB using standard linear modelling functions:

- `b = glmfit(X,y,distribution)` was used to build a model with the matrix of characteristics `X`, status vector `y` and the `distribution` parameter set to `binomial`.
- `yfit=glmval(b,X, link)` was used to predict using model `b` and input matrix `X`. The `link` option was set to `logit`.

Using the MATLAB function `glmfit` insignificant coefficients are set to zero automatically. The final model for the semester was obtained using 10 fold cross-validation and taking the average of the coefficients.

6.2.2 Overview of Individual LR Semester Models

A description of the performance for each semester models can be seen in appendix A on page 89.

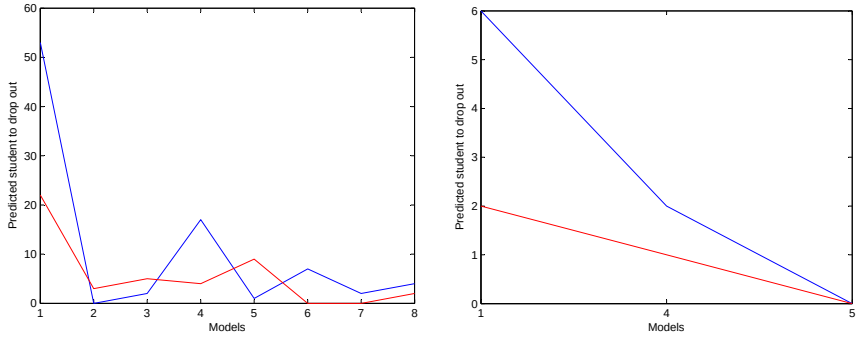
All 10 cross-validation models for every semester model were completed with one of the following warnings:

- *iterations limit reached*
- *X is ill conditioned, or the model is over parametrized, and some coefficients are not identifiable. You should use caution in making predictions*

The results of these warning are large coefficients with opposite signs. It can be expected due to the high correlation among the variables. Most of troubles were caused by the school exams level characteristics. In the next technique, principle component analysis will be used prior to LR to reduce the dimension of the characteristics to avoid the collinearity.

6.2.3 Final Model Using LR Technique

As described in section 6.1 on page 35 the first individual semester models will be analysed together. Those models that can predict additional drop out students are further selected. The models are executed in the order described in section 6.1 on page 35.



(a) Training (b) Test

Figure 6.1: Final model determination using LR. Blue - classified drop out correctly, red - falls alarms. The numbers are additional unique classifications not previously classified by the lowered numbered models.

A summary of the plot fig. 6.1 is given in tables 6.1 and 6.2 on the following page. It can be seen in fig. 6.1a that the highest number of drop out

	1	2	3	4	5	6	7	8	Ratio
Train correct	53	0	2	17	1	7	2	4	0.4388
Train falls	22	3	5	4	9	0	0	2	0.3435

Table 6.1: Important semester model selection for final LR model.

	1	4	5	Ratio
Test correct	6	2	0	0.4444
Test falls	2	1	0	0.2727

Table 6.2: Final LR model analysis.

students are predicted by model 1, 4 and 6. These three models are taken to the final model.

Table 6.2 shows only two models are significant on the validation set, but small data set is problematic. The model can identify 50% of dropouts. However, 30% of predicted dropouts are false alarms. An important property is how soon the final model it able to detect an upcoming dropout. On the training and test data the dropout notice is given 2.6860 and 2.5000 semesters in advance respectively.

6.3 Principle Component Analysis and Logistic Regression Modelling

6.3.1 Principle Component Analysis Technique

In this section the PC-LR model will be applied. For the logistic regression the same MATLAB functions as in section 6.2 on page 38 were used. To perform the principle component analysis the following was done:

- `[PCALoadings,PCAScores,PCAVar] = princomp(X)`. The function for given matrix X computes loading and scores matrices and vector with explained variance by each principle component.

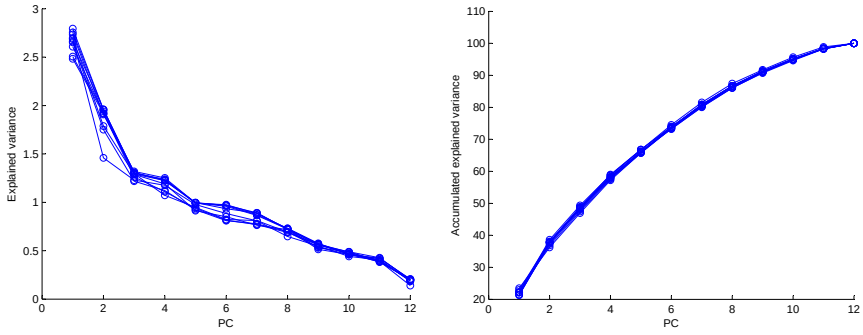
It must be noted, that PCA as logistic regression cannot work with `NaN` and `Inf` values. For this reasons, students with missing values will be removed.

As it was identified in section 6.2 on page 38 just model 1 and 4 were significant on the test set. Due to the fact, that PCA is helping the logistic regression, only those models for overall and second semester predictions will be analysed.

6.3.2 PC-LR Models

6.3.2.1 PC-LR: Model 1

PCA was performed as the first step. Figure 6.2 shows the explained and accumulated variance by the principal components.



(a) Explaned variance by each PC.

(b) Accumulated explained variance by PC's.

Figure 6.2: Variance of principle components for model 1.

As it can be seen in fig. 6.2 there is no clear cut for how many PCs should be used. In fig. 6.2a the most significant changes are at the 3rd and 8th PC. Two logistic models will be build and compared using the first 3 and 8 principal components.

As it can be seen from the analyses none of the models performed sig-

	DD	DP	PP	PD
Train(3 PC's)	90	59	203	141
Train(8 PC's)	85	62	332	113

Table 6.3: Predictions using 3 and 8 principle components.

Misclassification ratio	0.4057
Drop out misclassification ratio	0.1197
Drop out ratio in all misclassification	0.2950
Total number of PC	12
Used number of PC	3

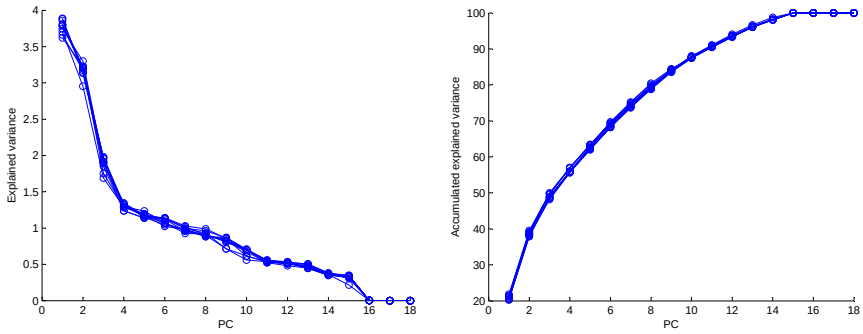
Table 6.4: Summary of the model using 3 principle components.

Misclassification ratio	0.3557
Drop out misclassification ratio	0.1260
Drop out ratio in all misclassification	0.3543
Total number of PC	12
Used number of PC	8

Table 6.5: Summary of the model using 8 principle components.

nificantly better than LR. If 3 principle components were used then the number of correctly classified drop out students is much higher than the simple logistic regression model table A.1 on page 89. However, the false alarm rate is unacceptable high. Using 8 principle components the false alarm rate had improved, but was still too high.

6.3.2.2 PC-LR: Model 4



(a) Explaned variance by each PC.

(b) Accumulated explained variance by PCs.

Figure 6.3: Variance of principle components for model 4.

Here as in section 6.3.2.1 on page 41 there is no clear cut for how many principle components should be used. Again two models will be build: one with 4 and one with 9 principle components.

	DD	DP	PP	PD
Train(4 PC's)	34	12	196	145
Train(9 PC's)	19	26	304	33

Table 6.6: Predictions using model fig. 6.3

As in the overall status prediction both models tables 6.7 and 6.8 on the following page have very high false alarm rate. The larger part of drop out predictions of model are false alarms. The performance of logistic regression is better then PC-LR. Misclassification rate of LR model 4 is around 8% while PC-LR with 4 PCs is around 40% and with 9 PCs 15%.

Misclassification ratio	0.4057
Drop out misclassification ratio	0.0310
Drop out ratio in all misclassification	0.0764
Total number of PCs	18
Used number of PCs	4

Table 6.7: Summary of the model using 4 principle components.

Misclassification ratio	0.1545
Drop out misclassification ratio	0.0681
Drop out ratio in all misclassification	0.4407
Total number of PCs	18
Used number of PCs	9

Table 6.8: Summary of the model using 9 principle components.

6.4 CART Modelling

6.4.1 CART Modelling Technique

The modelling was performed in MATLAB using standard CART functions:

- `t = classregtree(X,y)` was used for the model building, where `y` is the response variable and `X` the input matrix. Additional settings were used:
  - `categorical` to indicate which columns in matrix `X` are categorical.
  - `method` was set as classification, because `y` is categorical.
- `[c,s,n,best] = test(t,'crossvalidate',X,y)` to identify the best pruning level using cross-validation. Function provides with the results:
  - `c` is the cost vector.

  - `secost` is a vector that contains the standard error of the cost vector.
  - `n` is a vector of number of terminal nodes for each subtree.
  - `best` is the best level of pruning.

- `t2 = prune(t,'level',bestlevel)` to prune the chosen tree using the suggested best pruning level.
- `view(t)` to plot tree.
- `yfit = eval(t2,X)` to predict with tree `t2` using input matrix `X`.

The procedure of building the CART model starts with grow a large tree such that every terminal node has the minimal amount of observations, by default less than 10. MATLAB removes any observations with missing values automatically. However, when the final tree is built CART is able to predict using incomplete data. The trees were pruned using best pruning level found through cross-validation.

6.4.2 CART Models for Every Semester

6.4.2.1 CRAT: Model 1

As in section 6.2 on page 38 the first model for the overall status prediction was built.

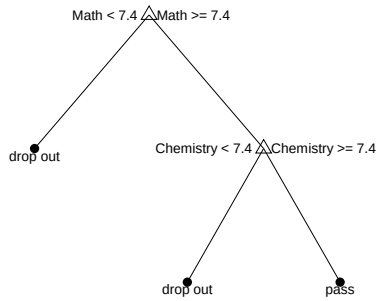


Figure 6.4: Classification tree for model 1.

CART trees are easy to interpret. Figure 6.4 on the preceding page shows that students who's mathematics and chemistry exams grades are greater or equal to 7.4 are most likely to graduate.

	DD	DP	PP	PD
Train	49	108	338	21
Test	10	31	87	4

Table 6.9: Predictions using model fig. 6.4 on the preceding page

Misclassification ratio	0.2531
Drop out misclassification ratio	0.2145
Drop out ratio in all misclassification	0.8476
Total number of levels	15
Pruned to level	3

Table 6.10: Performance information on model fig. 6.4 on the preceding page.

Tables 6.9 and 6.10 show that this model's false alarm rate might be a concern. Around 30% of all predicted dropouts might be false alarms.

No models 2 and 3 were build. When initial models were build it was used the cross validation to search for the best pruning level. In both cases it was suggested to prune to root node, for this reason no models were build.

6.4.2.2 CART: Model 4

	DD	DP	PP	PD
Train	28	34	350	9
Test	8	8	88	3

Table 6.11: Predictions using model fig. 6.5 on the next page

As it seen in fig. 6.5 on the facing page that only the ratio of passed and taken ECTS was chosen. Interpretation of this tree is that students who passed less than 87% of their chosen courses during first semester would

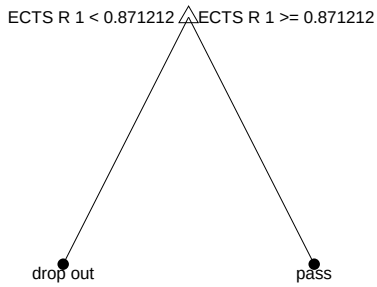


Figure 6.5: Classification tree for model 2.

Misclassification ratio	0.1023
Drop out misclassification ratio	0.0795
Drop out ratio in all misclassification	0.7778
Total number of levels	12
Pruned to level	2

Table 6.12: Performance information model fig. 6.5.

drop out after the second semester. Those who passed more than 87% would not drop out after the second semester. The misclassification rate compared to other models is not significantly higher.

6.4.2.3 CART: Model 5

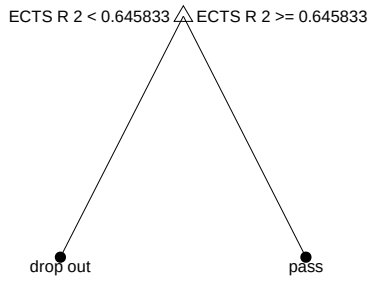


Figure 6.6: Classification tree for model 3.

As in model 4 the ratio of passed and taken ECTS credits was selected. Completing at least 65% of signed up ECTS credits is enough to not drop

	DD	DP	PP	PD
Train	9	6	357	2
Test	2	2	90	1

Table 6.13: Predictions using model fig. 6.6 on the preceding page

Misclassification ratio	0.0235
Drop out misclassification ratio	0.0171
Drop out ratio in all misclassification	0.7273
Total number of levels	4
Pruned to level	3

Table 6.14: Performance information on model fig. 6.6 on the preceding page.

out. This decrease of in required passed ECTS could be because students are more motivated to graduate being closer to graduation although they do lower the pace. The performance on the training set only had a few misclassifications. The test set is quite small so even 1 misclassification seems like a lot.

6.4.2.4 CART: Model 6

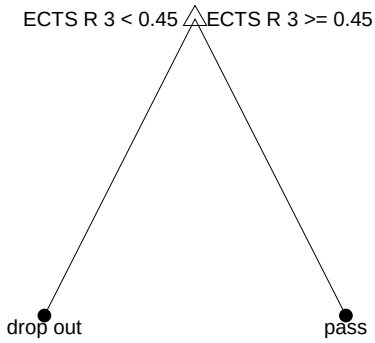


Figure 6.7: Classification tree for model 6.

The ratio of passed and taken ECTS credits suggested by the model is tendentiously decreasing. The false alarm rate, 0, for this model and is

	DD	DP	PP	PD
Train	5	9	359	0
Test	1	3	91	0

Table 6.15: Predictions using model fig. 6.7 on the facing page

Misclassification ratio	0.0256
Drop out misclassification ratio	0.0256
Drop out ratio in all misclassification	1
Total number of levels	4
Pruned to level	3

Table 6.16: Performance information on model fig. 6.7 on the facing page.

low, but this model does not catch all the dropouts.

No model for the fifth semester was created. As it was happening with model 2 and 4 the suggested pruning left only left the root node.

6.4.2.5 CART: Model 8

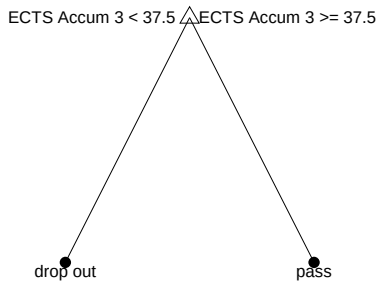


Figure 6.8: Classification tree for model 8.

Different from model 4, 5 and 6 model 8 for the sixth semester checks how many ECTS credits the students accumulated by the end of the third semester. Those students who accumulated less than 37.5 ECTS credits will drop out. Following the study plan to graduate in 3 years then by the end of third semester the student should have been accumulated 90 ECTS credits. It is interesting that this model is predicting the outcome

	DD	DP	PP	PD
Train	3	9	355	0
Test	2	1	86	4

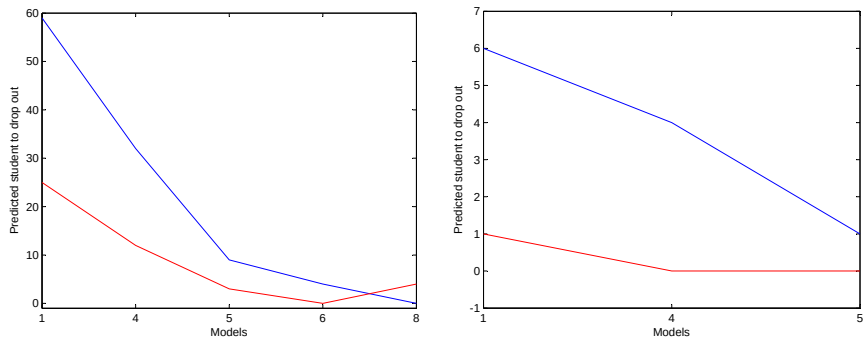
Table 6.17: Predictions using model fig. 6.8 on the preceding page

Misclassification ratio	0.0304
Drop out misclassification ratio	0.0217
Drop out ratio in all misclassification	0.7143
Total number of levels	4
Pruned to level	3

Table 6.18: Performance information on model fig. 6.8 on the preceding page.

after the sixth semester based only having at least 37.5 credits after the third semester. It is most likely due to students delaying the dropout from the university.

6.4.3 Final Model Using CART Technique



(a) Training (b) Test

Figure 6.9: Final model determination using CART. Blue - classified drop out correctly, red - falls alarms. The numbers are additional unique classifications not previously classified by the lowered numbered models.

Summary of the plots fig. 6.9 on the preceding page is presented in tables 6.19 and 6.20. Using a training set it can be seen that model 1, 4 and 6 are significant.

	1	4	5	6	8	Rate
Train correct	59	32	9	4	0	0.5253
Train false	25	12	3	0	4	0.2973

Table 6.19: Important semester model selection for final CART model.

	1	4	5	Rate
Test correct	6	4	1	0.6111
Test false	1	0	0	0.0833

Table 6.20: Final CART model analysis.

It is very unusual that model performance rates are better on the testing set than the training set. Around 52% of all dropouts were correctly identified and 29% of all dropouts predictions were incorrect (false alarms) in the training set. With the test set, 61% of all dropouts were identified and only 8% were incorrect. Also with the training data the average predicted in advance notice time is 2.8750 months and with the test set 3 month. The reason could be the noisy training set.

6.5 Bagging Modelling

6.5.1 Bagging Modelling Technique

The modelling was performed in MATLAB using standard tree bagging function:

- `B = TreeBagger(ntrees,X,Y)` was used to build `ntrees` trees with the characteristics matrix `X` and status indicators `Y`. Additional options were used:
  - `method` was set to `classification`.

  - `'oobPred'` was turned `on`. This saves for each tree information on which observations were out-of-bag (OOB).

- `oobError(B)` was used together with the MATLAB function `plot` to plot the out-of-bag classification error.

- `yfit = predict(B,X)` to predict using the bagged trees model `B` and the input matrix `X`.

The semester model building procedure followed these steps. First, five tree bagging models were built with 500 trees in each. The mean and standard deviation of the out-of-bag error was calculated. The number of required trees where the mean and standard deviation stabilises was chosen. At last, the new model with reduced number of trees was built.

6.5.2 Overview of Individual CART Bagging Semester Models

CART bagging gives almost no possibility to investigate the significance of the characteristics. One more drawback of this method is size of model. In fig. 6.10 it can be seen that the number of trees per semester model vary from 40 to 300 trees. For example, as in the previous models 1, 2 or 3 individual semester models were chosen and between 340 and 630 trees necessary for the prediction.

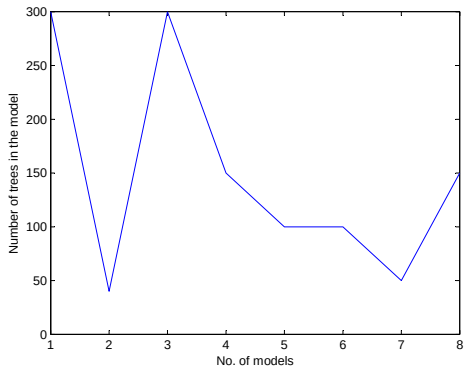
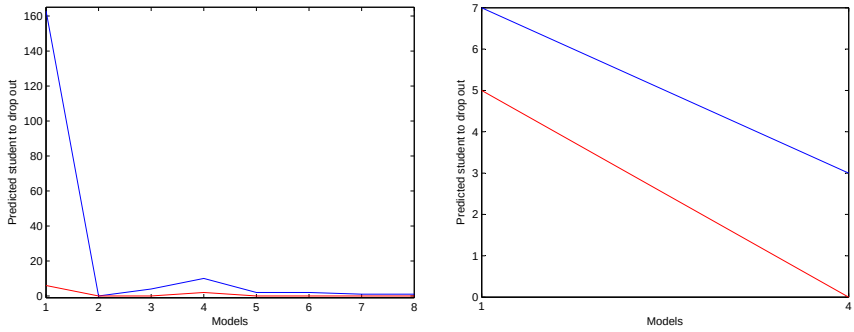


Figure 6.10: Simplified model scheme.

6.5.3 Final Model Using CART Bagging Technique



(a) Training. (b) Test.

Figure 6.11: Final model determination using CART bagging. Blue - classified drop out correctly, red - falls alarms. The numbers are additional unique classifications not previously classified by the lowered numbered models.

Figure 6.11a shows that the best predicting semester models are model 1 and 4. As it can be seen from the result, the final model has very high prediction power with low false alarm rate 0 and 4% (from the test and training sets respectively).

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Ratio**
Train correct	163	0	4	10	2	1	1	1	0.9242
Train false	6	0	0	2	0	0	0	0	0.0419

Table 6.21: Important semester model selection for final CART bagging model.

	1	**4**	**Ratio**
Test correct	7	3	0.5556
Test false	5	0	0.3333

Table 6.22: Final CART bagging model analysis.

The training data set gives a notice of dropouts 3.3934 months and the test set gives 2.6000 months in advance. Although the model is capable of predicting many of dropouts, the implementation of this type of model

is costly. 450 trees must be stored and used in the computation. Another problem of this model is lack of interpretability. It is not possible to pinpoint any characteristics as more valuable than others.

6.6 Random Forest Modelling

6.6.1 Random Forest Modelling Technique

The MATLAB code by Abhishek Jaiantilal [12] was used to build the models. This MATLAB code is based on the R implementation of Random Forest by Andy Liaw which is based on the original Fortran code by Leo Breiman and Adele Cutler. Two function were used:

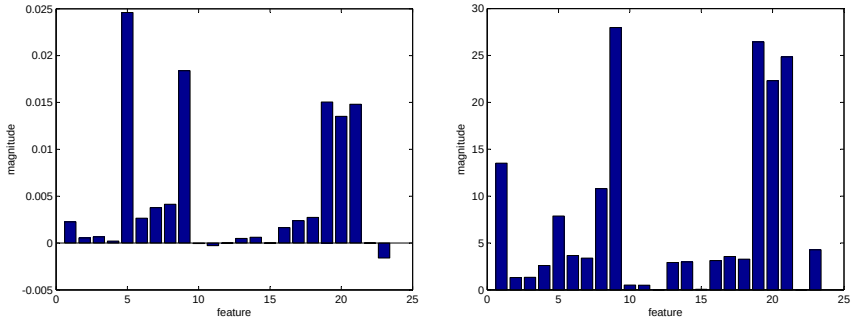
- `model = classRF_train(X,Y)` for model building with additional settings:
  - `ntree`: number of trees.
  - `mtry`: number of characteristics in X.
  - `extra_options.importance`: importance of the prediction will be assessed.
- `yfit = classRF_predict(X2,model)` for prediction. No additional option were used.

The variables *NTREE*, *MTRY* and number of important characteristics were selected using averages of several model results. Parameters of the model were chosen this way due to noise in the data. In the first step important variables were selected. It was done by creating 10 models and taking the mean of mean decrease in accuracy and mean decrease of Gini index. In the second step *MTRY* was determined. It was done using 5 fold cross-validation with *NTREE* fixed at 500 and *MTRY* varying from 1...*eq*. (4.18)*onpage* 25. In the third step, *NTREE* values were determined using 5 fold cross-validation with *MTRY* fixed.

6.6.2 RF Models for Every Semester

6.6.2.1 RF: Model 1

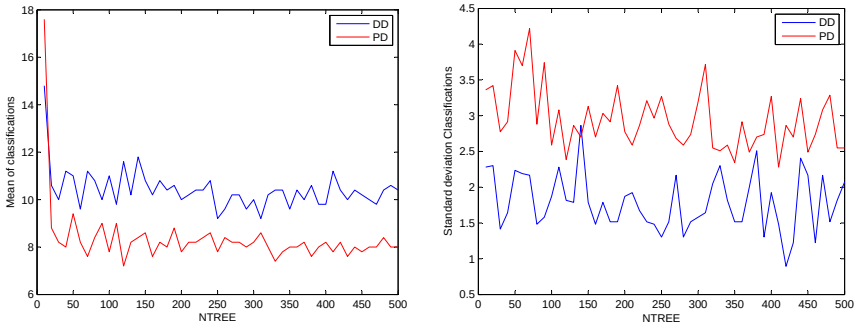
Important variables were detected using the default values: $NTREE = 500$ and $MTRY = 4$. The $MTRY$ value is calculated using eq. (4.18) on page 25.



(a) Mean decrease in accuracy.

(b) Mean decrease in Gini index.

Figure 6.12: Important variables selection for model 1.



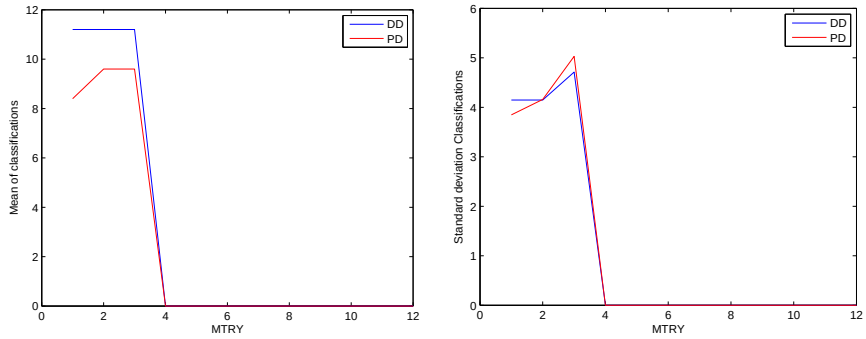
(a) Mean of classsifications.

(b) Standard deviation of classsifications.

Figure 6.13: Identification of $NTREE$, when $MTRY = 4$, for model 1.

As it was mentioned in section 4.5 on page 24 the mean decrease in the accuracy measure is more valuable than the mean decrease in the Gini index. For this reason, important features were selected on fig. 6.12a on

the preceding page. If the selection is too inaccurate the decrease Gini index is used. Important features were selected: 1 (Age), 5 (Design and Innovation programme), 6 (Mathematics and Technology programme), 7 (Biotechnology programme), 8 (Time since the last exam at school), 9 (GPA in school), 16 (Chemistry level A), 17 (Chemistry level B), 18 (Chemistry level C), 19 (Mathematics exam grade), 20 (Physics exam grade), 21 (Chemistry exam grade). The most important characteristics from the mean decrease in accuracy are 5, 9 , 19, 20, 21. These characteristics corresponds to the ones found in chapter 5 on page 29. The lowest drop out rates are in the Design and Innovation programme. And student with the highest grades are less likely to drop out.



(a) Mean of classsifications. (b) Sandard deviation of classsifications.

Figure 6.14: Identification of MTY for model 1.

It can be seen in fig. 6.14 that $MTRY$ should be 4. With $MTRY$ determined the test for $NTREE$ was performed. As seen in fig. 6.13 on the previous page the mean and standard deviation stabilises around 150 trees. Thus $NTREE$ is 150.

	DD	**DP**	**PP**	**PD**
Train	148	0	344	0
Test	13	28	78	13

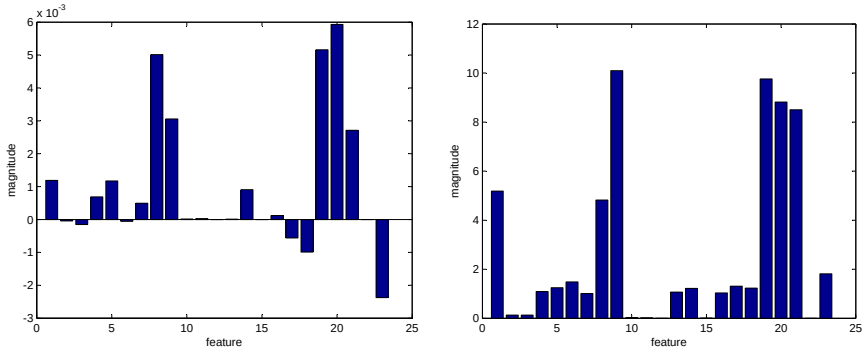
Table 6.23: RF predictions using model 1.

On the training set the model fits perfectly to the specific data. However, with the test set the false alarm rate was 50%.

Misclassification ratio	0.0657
Drop out misclassification ratio	0.0449
Drop out ratio in all misclassification	0.6829

Table 6.24: Performance information of model 1.

6.6.2.2 RF: Model 2



(a) Mean decrease in accuracy. (b) Mean decrease in Gini index.

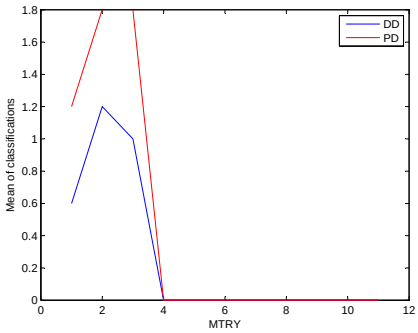
Figure 6.15: Important variables selection for model 2.

The default values for the important feature detection are: $NTREE = 500$ and $MTRY = 4$. The important features selected are: 1 (Age), 4 (Biomedicine programme), 5 (Design and Innovation programme), 8 (Time passed since school exam), 9 (school GPA), 14 (Physics level B), 18 (Chemistry level C), 19 (Mathematics exam grade), 20 (Physics exam Grade), 21 (Chemistry exam grade), 23 (Gender: male). Most of the important variables are the same as in model 1. The most significant change is the additional characteristic: the gender male. Figure fig. 6.15a shows that males are more likely to drop out.

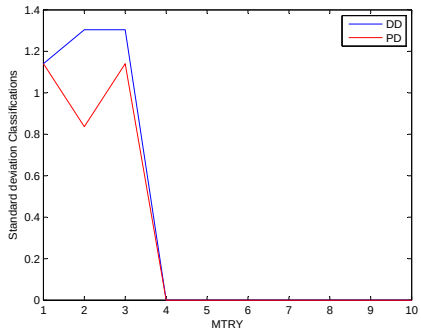
	DD	**DP**	**PP**	**PD**
Train	41	0	340	0
Test	3	8	91	0

Table 6.25: RF predictions using model 2.

$MTRY$ and $NTREE$ were chosen to be 4 and 300 respectively. The model

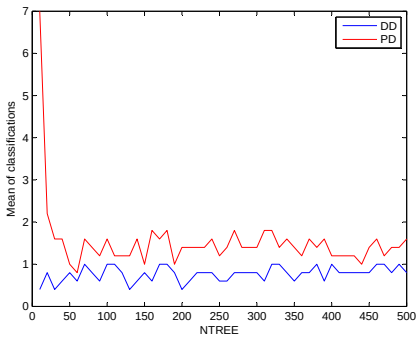


(a) Mean of classsifications.

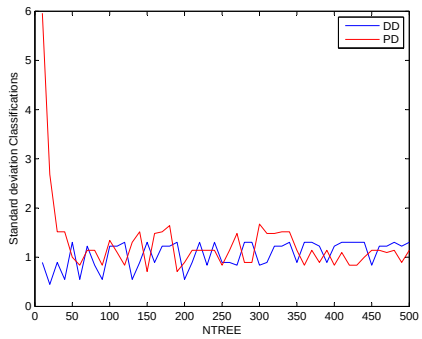


(b) Standard deviation of classsifications.

Figure 6.16: Identification of MTY for model 2.



(a) Mean of classsifications.



(b) Standard deviation of classsifications.

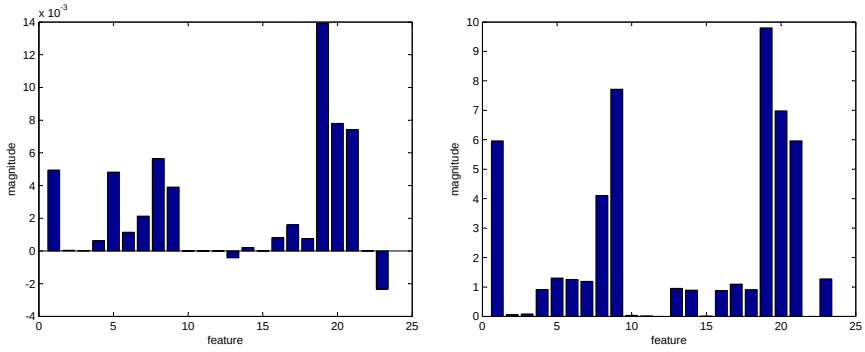
Figure 6.17: Identification of $NTREE$, when $MTRY = 4$, for model 2.

Misclassification ratio	0.0166
Drop out misclassification ratio	0.0166
Drop out ratio in all misclassification	1

Table 6.26: Performance information of model 1.

performance rates show that RF performs well in the model where CART could not be built. The only fussiness is that in the test set 8 dropouts were not identified.

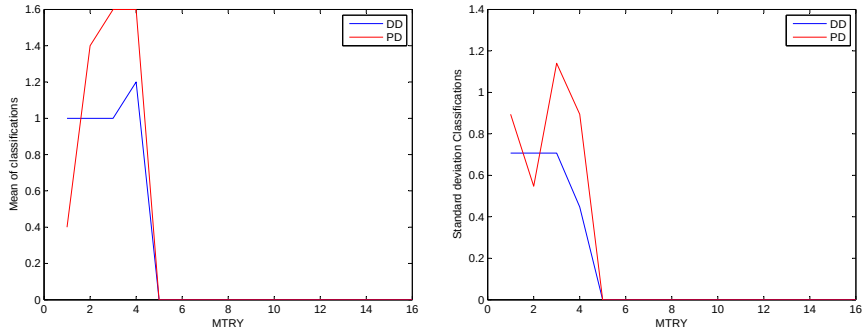
6.6.2.3 RF: Model 3



(a) Mean decrease in accuracy. (b) Mean decrease in Gini index.

Figure 6.18: Important variables selection for model 3.

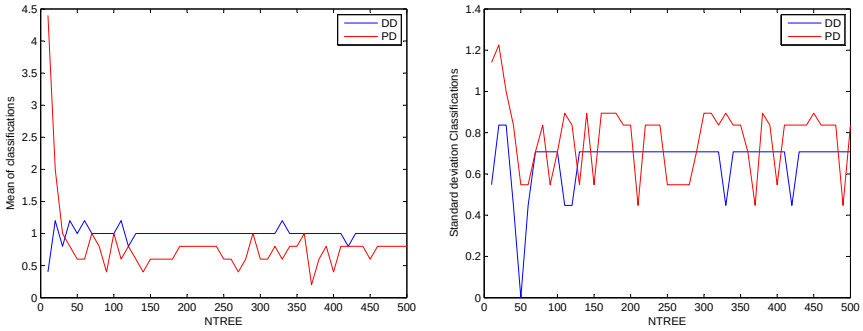
Important variables were detected using default values: $NTREE = 500$ and $MTRY = 4$. The same characteristic as in model 2 were chosen.



(a) Mean of classsifications. (b) Standard deviation of classsifications.

Figure 6.19: Identification of MTY for model 3.

The chosen values $MTRY = 5$ and $NTREE = 150$. Almost the same



(a) Mean of classsifications.

(b) Standard deviation of classsifications.

Figure 6.20: Identification of $NTREE$, when $MTRY = 5$, for model 3.

	DD	DP	PP	PD
Train	33	1	340	0
Test	1	8	91	0

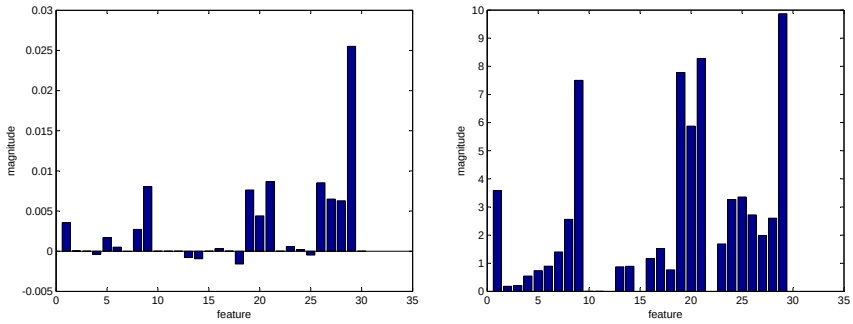
Table 6.27: RF predictions using model 3.

Misclassification ratio	0.0190
Drop out misclassification ratio	0.0190
Drop out ratio in all misclassification	1

Table 6.28: Performance information of model 3

situation with model 3 as with the model 2: CART was not capable of building the model, but random forest works very well.

6.6.2.4 RF: Model 4



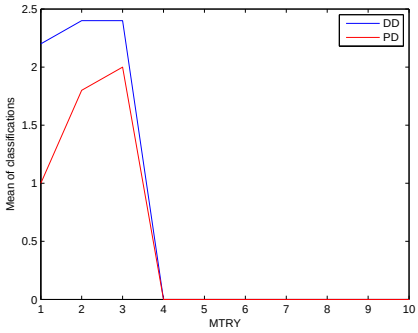
(a) Mean decrease in accuracy. (b) Mean decrease in Gini index.

Figure 6.21: Important variables selection for model 4.

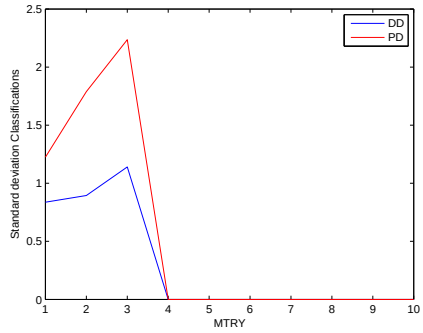
The important variables were detected using the default values: $NTREE = 500$ and $MTRY = 5$. The important features are: 1 (Age), 8 (Time after exams), 9 (school GPA), 19 (Mathematics exam grade), 20 (Physics exam grade), 21 (Chemistry exam grade), 26 (ECTS passed during first semester), 27 (ECTS taken during first semester), 28 (accumulated ECTS after first semester), 29 (ratio of passed and taken ECTS credits after first semester). The characteristics 26 - 29 are highly correlated because of their nature. The accumulated and passed ECTS (26 and 28) credits values are equal after the first semester. ECTS ratio (29) is just generalization of characteristics passed(26) and taken(27) ECTS credits after first semester. As it can be seen in fig. 6.21a ratio that is the generalization of all these correlated characteristic is most significant.

	DD	**DP**	**PP**	**PD**
Train	42	0	337	0
Test	4	12	88	3

Table 6.29: RF predictions using model 4.

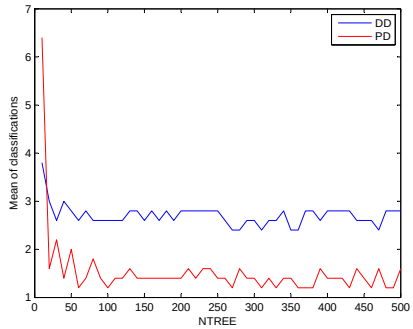


(a) Mean of classsifications.

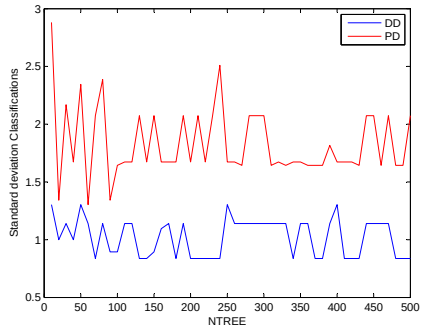


(b) Standard deviation of classsifications.

Figure 6.22: Identification of MTY for model 4.



(a) Mean of classsifications.



(b) Standard deviation of classsifications.

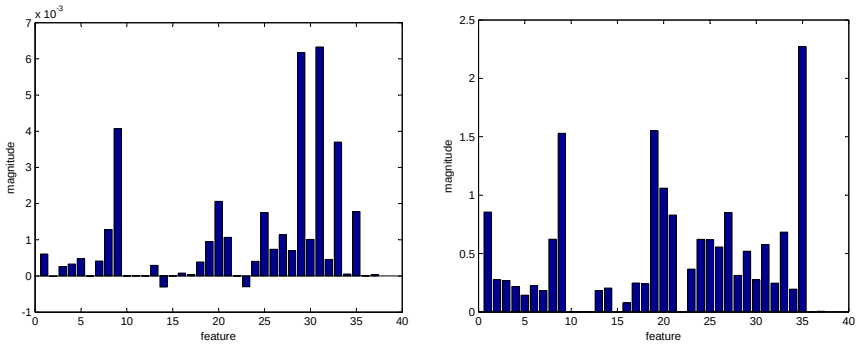
Figure 6.23: Identification of $NTREE$, when $MTRY = 5$, for model 4.

Misclassification ratio	0.0309
Drop out misclassification ratio	0.0247
Drop out ratio in all misclassification	0.8000

Table 6.30: Performance information of model 4.

The chosen values for $MTRY = 4$ and $NTREE = 100$. As seen in tables 6.29 and 6.30 on pages 61–62 the model has a very low misclassification rate. However, with the test set the model only identifies 4 out of 16 dropouts. This could be a sign of overfitting.

6.6.2.5 RF: Model 5

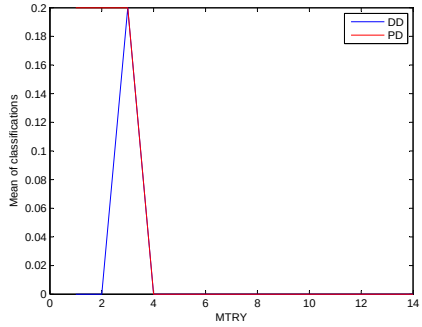


(a) Mean decrease in accuracy. (b) Mean decrease in Gini index.

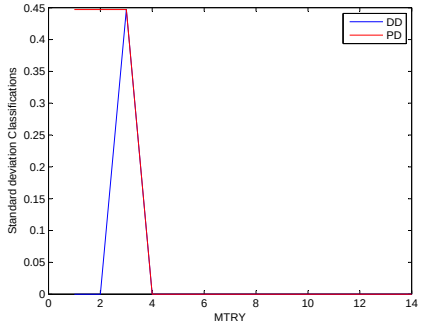
Figure 6.24: Important variables selection for model 5.

The important variables were detected using the default values: $NTREE = 500$ and $MTRY = 6$. The important features are: 1 (Age), 9 (school GPA), 19 (Mathematics exam grade), 20 (Physics exam grade), 21 (Chemistry exam grade), 25 (GPA overall after second semester), 26 (GPA of first semester), 27 (GPA of second semester), 28 (ECTS passed after first semester), 29 (ECTS passed after second semester), 30 (ECTS taken after first semester), 31 (ECTS taken after second semester), 33 (ECTS accumulated after second semester), 35 (ratio of passed and taken ECTS credits after second semester). The most important characteristics in this model are 31, 29, 9 and 33. It can be noticed that drop out students tend to fail a reasonable amount of ECTS credits during first semester. In the second semester they tend to take a large amount of courses to catch up. This could explain why the ECTS passed during first semester and ECTS taken during second semester are so important.

Chosen parameters are $MTRY = 5$ and $NTREE = 100$.

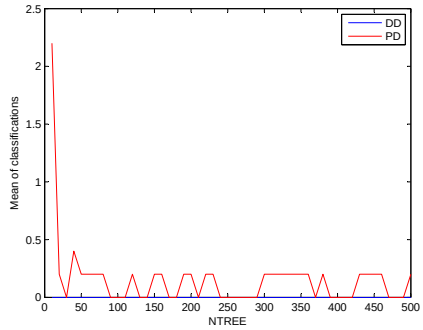


(a) Mean of classsifications.

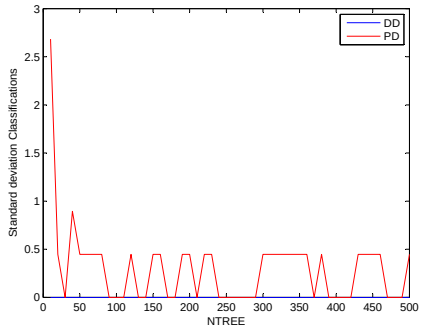


(b) Standard deviation of classsifications.

Figure 6.25: Identification of MTY for model 5.



(a) Mean of classsifications.



(b) Standard deviation of classsifications.

Figure 6.26: Identification of $NTREE$, when $MTRY = 5$, for model 5.

	DD	**DP**	**PP**	**PD**
Train	9	0	331	0
Test	0	4	91	0

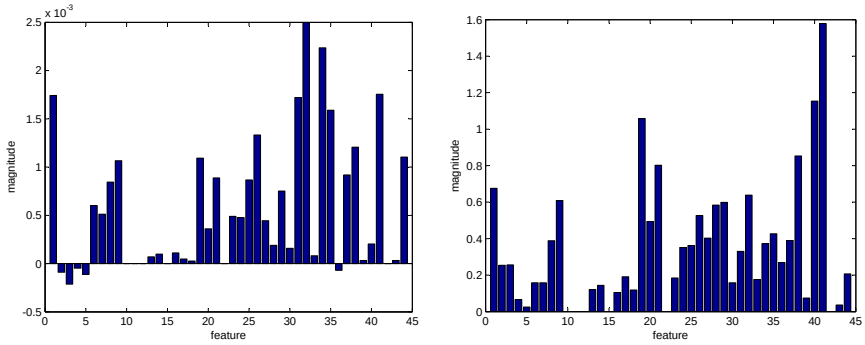
Table 6.31: RF predictions using model 5.

Misclassification ratio	0.0092
Drop out misclassification ratio	0.0092
Drop out ratio in all misclassification	1

Table 6.32: Performance information of model 5.

The misclassification rates are very low, but it seems that model overfits. Model 5 do not identify any drop out students in the test set.

6.6.2.6 RF: Model 6

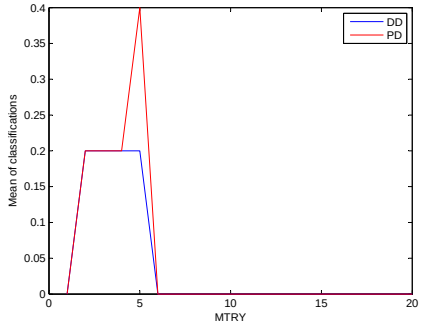


(a) Mean decrease in accuracy. (b) Mean decrease in Gini index.

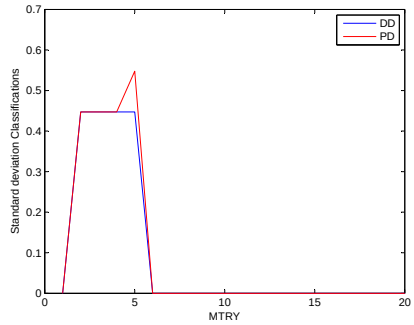
Figure 6.27: Important variables selection for model 6.

The important variables were detected using default values: $NTREE = 500$ and $MTRY = 6$. The list of the important variables is very long. The important features are: 1 (Age), 6 (Mathematics and Technology programme), 7 (Biotechnology programme), 8 (time after last school exam), 9 (school GPA), 19 (mathematics exam grade), 20 (physics exam grade), 21 (chemistry exam grade), 23 (gender:male), 24 - 26 (represents overall GPA after fist-third semester), 27 (GPA of the first semester), 29 (GPA of the third semester), 31-32 (represents ECTS passed during second and third semester), 34-35 (represents ECTS taken during second and third semester), 41 (ratio of passed and taken ECTS after third semester), 44 (indicator that the student was more then 30 ECTS credits behind study plan). This large number of important variables makes it difficult to interpret the model. This could also be an indicator that the model is unreasonable.

Here $MTRY = 6$ and $NTREE = 100$. The model performance rates confirms that the model overfits. Probably, there were to little dropouts the in the training set to catch the structure.

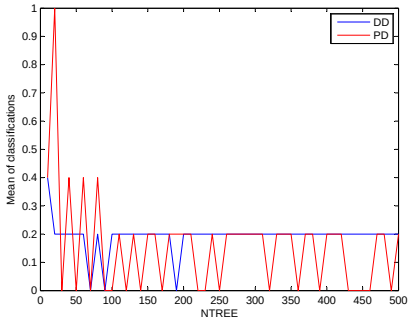


(a) Mean of classsifications.

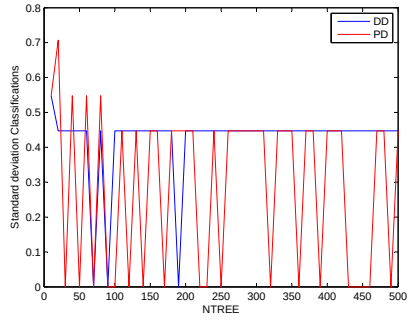


(b) Standard deviation of classsifications.

Figure 6.28: Identification of MTY for model 6.



(a) Mean of classsifications.



(b) Standard deviation of classsifications.

Figure 6.29: Identification of $NTREE$, when $MTRY = 5$, for model 6.

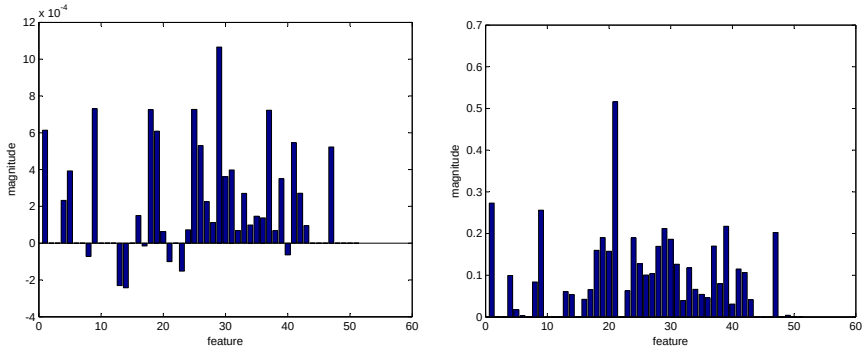
	DD	DP	PP	PD
Train	8	0	333	0
Test	1	3	91	0

Table 6.33: RF predictions using model 6.

Misclassification ratio	0.0069
Drop out misclassification ratio	0.0069
Drop out ratio in all misclassification	1

Table 6.34: Performance information of model 6.

6.6.2.7 RF: Model 7



(a) Mean decrease in accuracy. (b) Mean decrease in Gini index.

Figure 6.30: Important variables selection for model 7.

Default values: $NTREE = 500$ and $MTRY = 7$. As in model 6 the list of important features is long: 1 (age), 4 (Biomedicine programme), 5 (Design and Innovation programme), 9 (school GPA), 13 (physic level A), 14 (physic level B), 18 (chemistry level C), 19 (Mathematics exam grade), 23 (gender:male), 25-27 (overall GPA of the second-fourth semester), 29-31 (semester GPA of second-fourth semester), 33 (passed ECTS during second semester), 37 (takes ECTS during second semester), 39 (takes ECTS during fourth semester), 41-42 (accumulated ECTS during second - third semester), 47 (ratio of passed and taken ECTS during fourth semester).

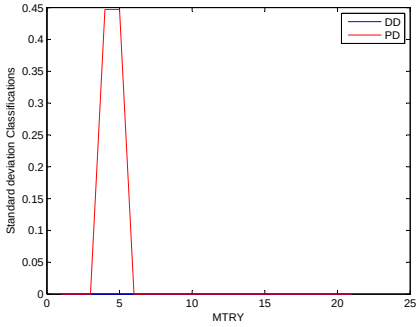
	DD	**DP**	**PP**	**PD**
Train	12	0	332	0
Test	0	1	91	0

Table 6.35: Model predictions using model 7.

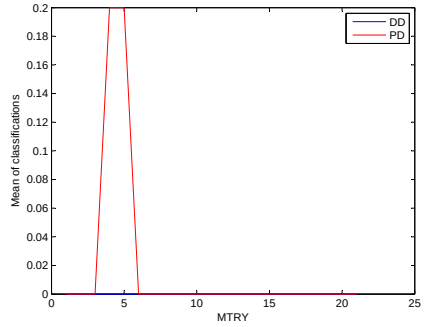
Misclassification ratio	0.0023
Drop out misclassification ratio	0.0023
Drop out ratio in all misclassification	1

Table 6.36: Performance information of model 7.

Here $MTRY = 6$ and $NTREE = 50$. As in the previous model a large

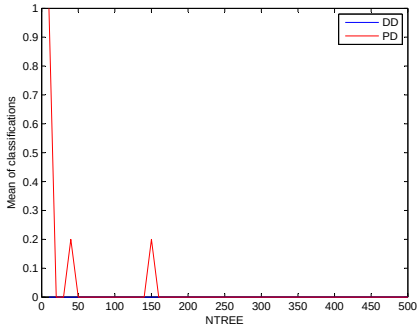


(a) Mean of classsifications.

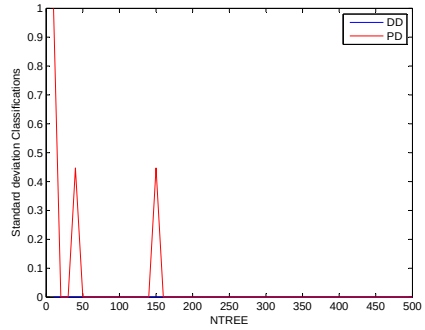


(b) Standard deviation of classsifications.

Figure 6.31: Identification of MTY for model 7.



(a) Mean of classsifications.

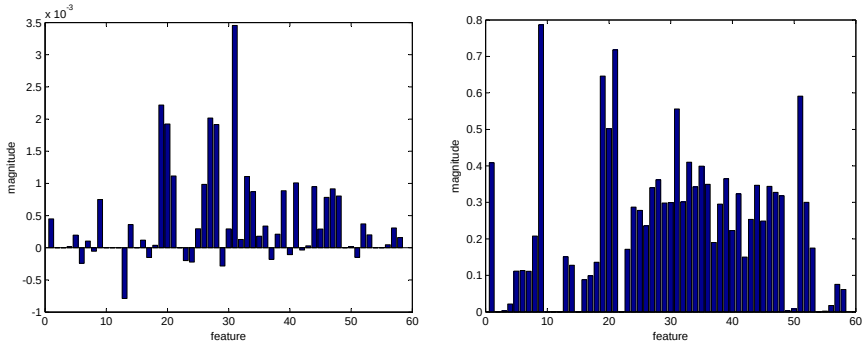


(b) Standard deviation of classsifications.

Figure 6.32: Identification of $NTREE$, when $MTRY = 5$, for model 7.

number of important characteristics were found and the small amount of drop out students made the model useless.

6.6.2.8 RF: Model 8



(a) Mean decrease in accuracy. (b) Mean decrease in Gini index.

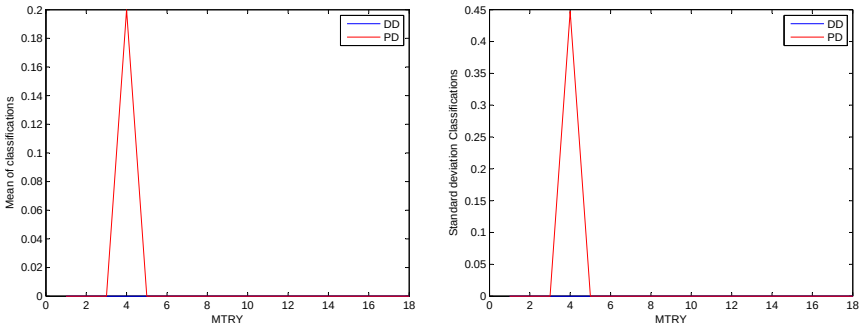
Figure 6.33: Important variables selection for model 8.

The important variables were detected using the default values: $NTREE = 500$ and $MTRY = 7$. Important features selected: 1 (age), 9 (school GPA), 13 (physics level A), 19 (mathematics exam grade), 20 (physics exam grade), 21 (chemistry exam level), 26-28 (overall GPA during third-fifth semester), 31 (third semester GPA), 33 (fifth semester GPA), 34 (passed ECTS during first semester), 39 (taken ECTS during first semester), 41 (taken ECTS during third semester), 44-45 (accumulated ECTS during first-second semester) and 47-48 (accumulated ECTS during fourth-fifth semester).

	DD	**DP**	**PP**	**PD**
Train	7	0	332	0
Test	0	3	90	0

Table 6.37: RF predictions using model 8.

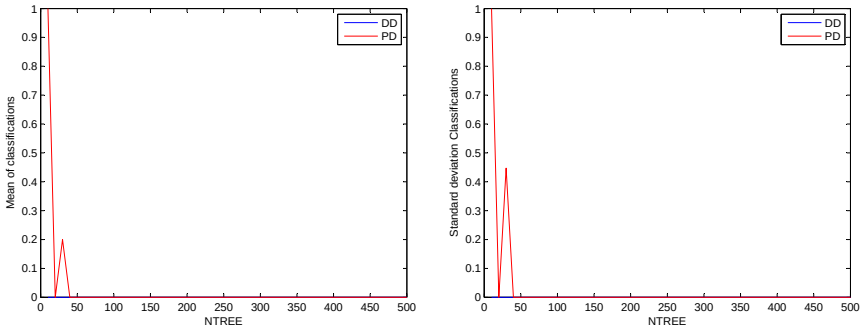
Chosen values for $MTRY = 6$ and $NTREE = 50$. As in the previous model a large number of important characteristics were found and the small amount of drop out students made the model useless.



(a) Mean of classsifications.

(b) Standard deviation of classsifications.

Figure 6.34: Identification of MTY for model 8.



(a) Mean of classsifications.

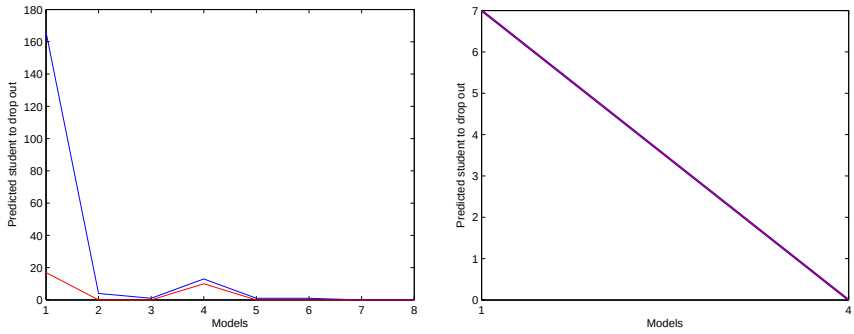
(b) Standard deviation of classsifications.

Figure 6.35: Identification of $NTREE$, when $MTRY = 5$, for model 8.

Misclassification ratio	0.0069
Drop out misclassification ratio	0.0069
Drop out ratio in all misclassification	1

Table 6.38: Performance information of model 8.

6.6.3 Final Model Using RF Technique



(a) Training (b) Test

Figure 6.36: Final model determination using RF. Blue - classified drop out correctly, red - false alarms. The numbers are additional unique classifications not previously classified by the lowered numbered models.

	1	2	3	4	5	6	7	8	Rate
Train correct	166	4	1	13	1	1	0	0	0.9394
Train false	17	0	0	10	0	0	0	0	0.1268

Table 6.39: Important semester model selection for final RF model.

	1	4	Rate
Test correct	7	0	0.3889
Test false	7	0	0.5000

Table 6.40: Final RF model analysis.

For the training data the model gives a pre-drop out notice of 3.3871 semester and for the test set 3.2857 semester in advance. No doubt that with training set model does excellent job. Just with the first model it identifies around 84% of all dropouts with a 9% false alarm rate. However, the model validation results are very disappointing. The model will identify around 39% of all dropouts with a high 50% false alarm rate.

6.7 MARS Modelling

6.7.1 MARS Modelling Technique

MATLAB does not have an implemented MARS function. However, [13] has implemented MARS building and support functions:

- `trainParams = aresparams(M, k-fold, cubic, [], mi, d)` creates a structure of MARS configuration parameter values.
  - `M` maximal number of basis function in the forward-stepwise method.
  - `cubic` if set to `false` a piecewise-linear polynomial is used.
  - `d` in case of piecewise-linear polynomial is use then it can be set to the maximum allowed iterations.
  - `mi` is the maximum degree of self interactions for the characteristics. This parameter is only used for the piecewise-linear polynomial.
  - `k-fold` number of folds to use in cross-validation.
- `[model, time] = aresbuild(X, y, trainParams)` is used to build the MARS models. `X` is the independent variable matrix, `y` is a vector of the dependent variable and finally the parameters.
- `Y = arespredict(model, X )` does prediction of the MARS model with the input `X`.
- `[avgMSE, avgRMSE, avgRRMSE, avgR2] = arescv(X, Y, trainParams, [], k)` performs cross-validation. Options were used:
  - `trainParams` model parameters.
  - `k` number of folds.

  This function results in the average mean squared error, average root mean squared error, average relative root mean squared error, average coefficient of determination and average execution time.

- `eq = areseq(model, precision)` gives the model in a mathematical form. `precision` is the number of digits in the model coefficients and knot sites.

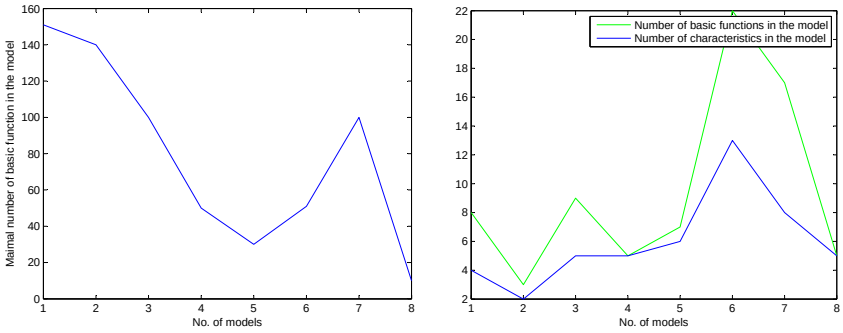
As suggested in [5] the best way to choose M, d and mi is using cross-validation. The same paper mention the values of d and mi should be from 1 to 3. The number of M depends on the number of characteristics in input matrix. The minimum number of characteristics is 23 (in application scoring model) and maximum 72 (in model 8 for the sixth semester). The chosen interval for M was from 21 to 151 by a step of 5 to save computational time. With every set of parameters `arescv(X, Y, trainParams, [], k)` computed which performed a 5 fold cross-validation. The parameters were selected by the lowest average mean square error (`avrMSE`).

6.7.2 Overview of Individual MARS Semester Models

A description of the performance for each semester models can be seen in appendix C on page 111.

For all semester models no maximum interactions and number of self-interactions were chosen. The analysis showed that increasing these two parameters would increase the mean square error variation, but remains around the same value. The maximum number and used number of basic functions in the final semester models vary from model to model.

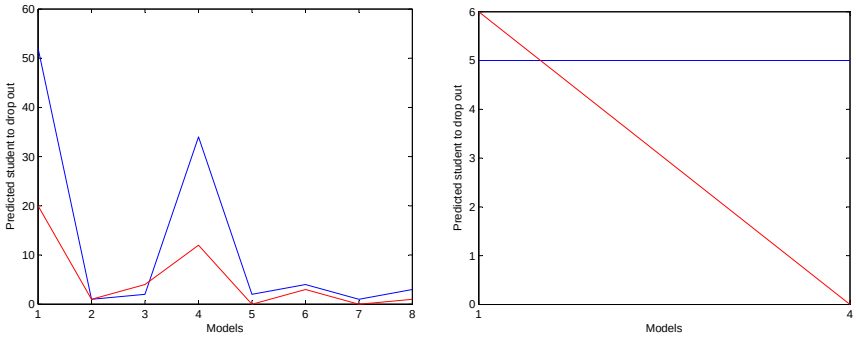
As seen in fig. 6.37 on the next page MARS struggles when searching for the best separation. For some semester models the required maximum number of basic functions can be quite high for the forward-stepwise method. The method reduces the number of basic functions but the final model remains too complex carrying too many basic functions. In the final semester model, MARS will include the same characteristics several times. This makes the model difficult to interpret.



(a) Maximum. (b) Final.

Figure 6.37: Maximal number of basic functions in the forward-stepwise process and number of basic functions in final semester model

6.7.3 Final Model Using MARS



(a) Training (b) Test

Figure 6.38: Final model determination using MARS. Blue - classified drop out correctly, red - false alarms. The numbers are additional unique classifications not previously classified by the lowered numbered models.

On the training set the model gives a pre-drop out notice of 2.7374 semesters and on test set it is 2.4000 semesters in advance. In comparison to the other models this is a bit lower. The model predicts around 50% of all dropouts, but the false alarm rate was observed at 37-59% in the training and test sets. Furthermore, the models are very complex can pose a challenge upon implementation.

	1	2	3	4	5	6	7	8	Ratio
Train correct	52	1	2	34	2	4	1	3	0.5000
Train false	20	1	4	12	0	3	0	1	0.5973

Table 6.41: Important semester model selection for final MARS model.

	1	4	Ratio
Test correct	5	5	0.5556
Test false	6	0	0.3750

Table 6.42: Final MARS model analysis.

CHAPTER 7

Result Analysis

7.1 Model Comparison

Model	Number of models	Correct class.	False alarm	Time
Logistic Regression	3	0.44	0.34	2.5
CART	3	0.61	0.08	3
CART Bagging	2	0.56	0.33	2.6
Random forest	2	0.39	0.5	3.3
MARS	2	0.56	0.38	2.4
Current system	-	0.33	0.80	2.5

Table 7.1: Model comparison table.

At the model building phase the most promising model was the random forest. On the test set RF showed very low performance. This suggest that RF overfitted. Logistic regression face problems with collinearity, that leads to low prediction rate with a high false alarm rate. MARS also fails to predict drop out students correctly. As LR, MARS is also sensitive to collinearity and noise.

Most promising methods is CART. CART predicts 61% of all dropouts with a false alarm rate of 8% based on 3 separate semester models. It can give the prediction of dropping out 3 semesters in advance before student is actually going to drop out. Second best method is CART bagging do it 2.6 semesters before and predicts 56% of all dropouts, but misclassification rate is 33%. Both methods has similar prediction power, but to implement CART bagging would need a specific programs to handle all the many trees generated while CART can be done with some 'if' conditions.

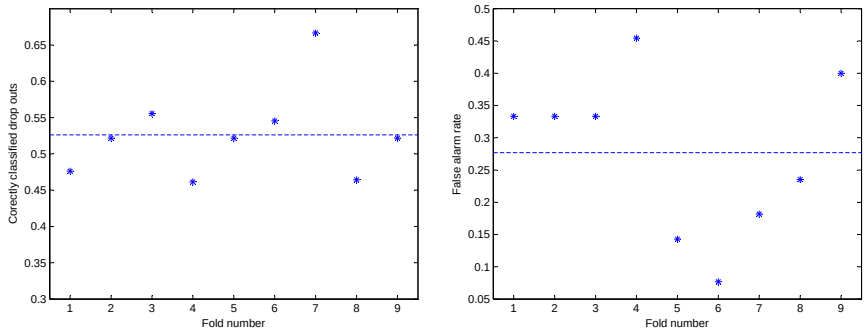
In comparison of current system to analysed methods, it performs worst. It identifies just 33% of dropouts with high false alarm rate 80%. Where the worst analysed model (LR) predicts 44% with false alarm rate 34%. This high rate of false alarm in the current student monitoring system is because of two reasons. Fist, strong relation to the previous semester. If good student skipped one semester it most likely he or she will be identified as a drop out for rest of the study time. Second, the long period of student monitoring. If analysed models suggest to monitor students for the first two or three semester, current system requires of students monitoring during all study period.

7.2 Final CART Model Stability

CART showed very good performance on the training set and even significantly better on the test set. Further analysis of CART's prediction stability will be done. The training set is divided in 9 folds of similar size as the test set. For each fold the final CART model will be applied.

As seen in fig. 7.1 on the facing page the drop out classification rate is more or less stable around the correct average drop out rate of 53%. The false alarm rate can vary from 8% to 45% with the average being 28%.

Figure 7.2 on the next page shows that model 1 is the strongest source of the false alarm. It can be due to a few reasons. First, in the first model the chemistry grade is included. Chemistry is not very relevant for the Mathematics and Technology programme. The model most likely



(a) Drop out calssifications. (b) False alarms rate.

Figure 7.1: CARTfinal model stability.

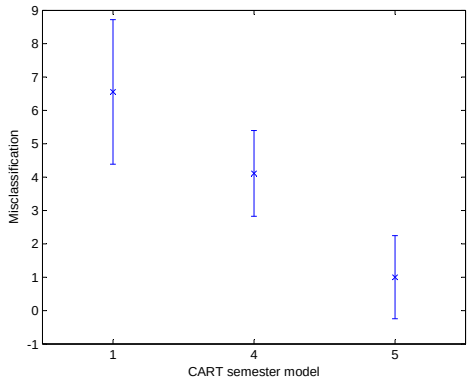


Figure 7.2: CART individual semester models false alarm stability.

captured most of the dropouts from the Biomedicine programme where chemistry is important. To avoid this all programmes with similar main topics should be grouped and analysed separately. Another reason could be that students with low grades entering the university can graduate. The motivation is not included in the model and thus there can be some misclassifications. It can be seen in fig. 7.2, that with time predictions are more accurate. It is because, more information is known about the student capabilities.

7.3 Important Variable Analysis

An additional task of this thesis was to identify the important factors causing a student to drop out. This information can be obtained from the models analyses. However, not all models are easily interpreted which is essentially the question here. The CART bagging model is a black box method. This means the characteristics that are included in the model are not really known. MARS provides information about the characteristics picked up for during the model building, but the complexity of the MARS models complicates its interpretability and the importance for the different characteristics is cluttered. Usually logistic regression can be used for interpretation, yet only few of the coefficients were set to zero and due to the highly inter-correlated characteristics some of the estimates had large values and with opposite sign - also ruining the interpretation.

CART does not provide an importance measure for the characteristics in the model, but characteristics can be seen in a hierarchical order. It can become complicated if the tree is very large. The parent node is the most important characteristic, but how does lower similar levelled characteristics rank relative to each other is not clear. Random Forest has developed a characteristic measures. That is the mean decrease in accuracy and mean decrease in Gini index. Important variables will be compared from CART and RF models.

Model 1: CART chooses mathematics and chemistry exam grades. RF chooses the most significant characteristics: Design and Innovation programme and school GPA. However, mathematics and chemistry exam grades together with physics grade are also very significant in RF.

Model 4: CART an RF choose the first semester ratio of passed and taken ECTS credits. It is worth to mention that school exam grades are also significantly important.

Model 5: CART chooses the ratio of passed and taken ECTS credits after the second semester. RF does not consider this characteristic equally

important although it is in the list of the four most important characteristics. The school exam grades are still important.

Model 6: CART chooses the ratio of passed and taken ECTS credits after the third semester. This characteristics was also one of the most important in RF. Furthermore, RF chooses the individual characteristics passed and taken ECTS credits during the second and third semester.

Model 8: CART chooses the accumulated ECTS credits after the third semester. RF also refers to the third semester, but to the GPA of the semester. RF gives additionally a lot of importance to the school exam grades.

From this comparison it can be seen that there is no doubt that the school exam grades are very important while the GPA is not too important. A high school GPA can be driven by subjects irrelevant to the DTU study programmes thus not necessarily provide any useful information. The three specific school exam results are a good representation for the students readiness for a technical university.

When a student reaches university, the best performance indicators are passed and taken ECTS credits per semester. Depending on the model it might also include the ratio of the passed and taken ECTS credits. CART uses the ratio, while RF the individual characteristics.

From this summary it can be concluded that the current dropout detection system is not optimal for DTU. The being behind by 30 credits indicator was only included in one last semester RF model and it was only medium important. This measure might be worth to check when the study time gets long.

CHAPTER 8

Conclusion

High rates of dropouts every year that cost a lot of money for the tax payers is forcing universities to search for new solutions. The current system suggests to offer consultation to a student who is behind by 30 ECTS credits or more. This master's thesis offers a different approach for drop out students detection. The new method suggest to keep track on student performance using their application and performance information over several semesters. Before the student even starts their studies the model will evaluate whether the student will drop out. After every semester the student's performance must be checked by model to make sure that the student is still performing good enough to graduate. Though some of the drop out student can not be identified due to their high performance rates, most often these students decide to leave university for personal reasons.

Six techniques were compared for every semester status identification: logistic regression (LR), principle component logistic regression (PC-LR), classification and regression trees (CART), classification and regression tree bagging (CART bagging), random forest (RF) and multivariate adaptive regression splines (MARS). After testing each model it was concluded

that LR failed to perform due to hight collinearity among the variables. Principle components analysis do not improve the performance of logistic regression. RF overfits the data which results in many misclassifications, though some of the literature suggests random forest cannot overfit due to its model building technique. MARS also failed to correctly predict drop out students. The most efficient models were build by CART and CART bagging methods. These methods could identify more than half of the drop out students with low false alarm rates. These methods showed that it is enough to keep track on students' performance for no more than the first three semesters. CART can be perform on any program supporting logic functions, but for the CART bagging it is necessary to adopt a special program.

Methods like CART and RF gave an understanding of the indicators causing the students to drop outs. It was identified that chemistry, mathematics and physics exam grades are significant indicators for a student's ability to continue. Surprisingly, school GPA was just a medium significance indicator. One of the most important performance characteristic was the ratio of passed and taken ECTS credits per semester. CART chose this ratio as a main indicator for three semester models. The indicator that student is behind by 30 ECTS credits or more was chosen once by one of the by RF for later semester models. This leads to the conclusion, that current system is not optimal for DTU.

The current student monitoring system is not better then other analysed methods. The 30 ECTS credit delay indicator makes more confusion then helps to identify drop out students. It do not show how well performs the student and student's capabilities. It indicates whether the student missed one or more semester. That has low relation to the student status. What is more, using this system student must be monitored for all study period, while with suggested methods students must be monitored just for few semesters.

The further analysis and implementation of this new system will allow DTU identify drop out students early on. This would give enough time for the university to intervene and help those troubled students. As more students actually graduates the university will be paid by the state and will help the government to achieve the national goal that half of the

young people should have the higher education.

8.1 Future Work

In the future three aspects of this topic should be investigated. First, models for groups of programmes. Second, categorization of drop out and passed students. Third, information about student performance in mandatory courses.

It has been demonstrated that the chemistry grade is chosen by many models. However, this subject does not seem all too relevant for Design and Innovation or Mathematics and Technology programmes. Chemistry was likely chosen because there are so many dropouts in the Biomedicine and Biotechnology programmes where a good chemistry exam grade seem more important. It should be investigated if specific programmes require a specific school exam. For example, the Biomedicine and Biotechnology programmes' main exam would be chemistry while Physics and Mechanics would be physics.

Another aspect that should be investigated is categorization of the students. As demonstrated in the data analysis the data is very noisy. Good students drop out and bad students graduate. A suggestion could be to group students into four groups. Two groups for pass students: passed at a high performance level, passed at a low performance level and two groups for dropouts: drop out at a high performance level and drop out at a low performance level. Students who drop out at the high performance level would be a student quitting due to personal reasons and this group is likely impossible to detect. The highest attention should be given to the group that drop out at the low performance level. This group consists of students lacking motivation and social establishment. Students who passed at the low performance level would be students either solely interested in graduating but not in the studies itself or having troubles with studying. For the model to perform better a student categorization should be examined. Their motivation, interest in the subject and their evaluation of the university should give an understanding of the group boundaries.

At DTU the students have freedom to choose their courses. Some of the courses are easier than others, yet there are some mandatory courses that must be completed during the programme. Investigating the mandatory courses should give an equal comparison among the students in the program.

Changing the study programme within DTU can be considered dropping out of the first programme. [14] suggests student who dropped out once are likely to drop out again. The effect of jumping between the programmes should be investigated.

Abbreviations

avgRRMSE	average of relative root mean squared error.
avrMSE	average of means square error.
avgR2	average of R-square measure.
avrRMSE	average of root square error.
CART	Classification and Regression Trees
d	maximal number of allowed interactions in MARS modelling.
DD	group of students who drop out and were classified as dropouts.
DP	group of students who drop out but were classified as pass students.
ECTS_A	Accumulated ECTS credits
ECTS_L	Indicator if 30 ECTS credits behind
ECTS_P	ECTS credits passed
ECTS_R	ratio of passed and taken ECTS credits
ECTS_T	ECTS credits taken
GCV	generalized cross-validation.

GPA_O overall grade point average

GPA_S semester grade point average

LR Logistic Regression

M maximal number of basic functions in forward-stepwise method of MARS modelling.

MARS Multivariate Adaptive Regression Splines.

MDA mean decrease in accuracy measure.

MDG mean decrease in Gini index measure.

mi maximal number of allowed self interactions.

MTRY the smallest number of variables rampantly selected in the random forest method.

NTREE number of tree is random forest method.

OOB error out-of-bag error.

PD group of students who passed but were classified as dropouts.

PP group of students who passed and were classified as passed students.

PC-LR Principle Component Logistic Regression.

RF Random Forest

Appendix A

LR Models for Every Semester

A.1 LR: Model 1

	Interaction	**Age**	**Lock stud.**	**In. stud.**
β	-4116.9220	0.1160	-0.5950	0
	Tech. biomed. stud.	**Design stud.**	**Mat stud.**	**Biotech. stud.**
β	0.1270	-1.0650	0.1800	-0.0930
	Time af. exam	**GPA**	**Math lev. A**	**Math lev. B**
β	-0.1240	-0.0550	-20.8000	0
	Math lev. C	**Physics lev. A**	**Physics lev. B**	**Physics lev. C**
β	0.0000	256204.7870	256204.8680	14.7730
	Chemistry lev. A	**Chemistry lev. B**	**Chemistry lev. C**	**Math grade**
β	-252063.55900	-252063.6090	252063.6090	-0.1210
	Physics grade	**Chemistry grade**	**Man**	**Woman**
β	-0.2090	-0.2440	0	-0.2200

Table A.1: Coefficient of logistic regression model 1.

Reasonable coefficient estimated for some characteristics can be noted.

Students who had mathematics at A level is less likely to drop out. It seems that physics and chemistry exam grades are more significant than mathematics exam grade. This can be due to the large data set from the Biomedicine and Biotechnology programmes. Females and students from the Design and Innovation programme have a lower drop out rate.

	DD	**DP**	**PP**	**PD**
Train	39	109	329	15
Test	15	25	82	4

Table A.2: Predictions using the model table A.1 on the preceding page

Misclassification ratio	0.2476
Drop out misclassification ratio	0.2168
Drop out ratio in all misclassification	0.8758
Not classified	30

Table A.3: Performance information for model table A.1 on the preceding page

Table A.3 shows some 87% of the misclassifications are uncaught drop out students.

A.2 LR: Model 2

Table A.4 on the facing page has almost the same important characteristics as table A.1 on the previous page. The greatest difference is the chemistry level coefficients. Table A.6 on the facing page shows this model has a low false alarm rate of 0.0936. However the model did not identify 45 out of 51 drop out students.

	Interaction	Age	Lock stud.	In. stud.
β	-143.7010	0.0650	-10.3280	-27.5760
	Tech. biomed. stud.	**Design stud.**	**Mat stud.**	**Biotech. stud.**
β	-2.2520	-4.5490	-2.2140	-3.2530
	Time af. exam	**GPA**	**Math lev. A**	**Math lev. B**
β	0.0680	-0.1480	1909.1380	-10.0420
	Math lev. C	**Physics lev. A**	**Physics lev. B**	**Physics lev. C**
β	0	-1730.2700	-1729.4840	9.7300
	Chemistry lev. A	**Chemistry lev. B**	**Chemistry lev. C**	**Math grade**
β	-17.7220	-17.5100	-17.0290	-0.2290
	Physics grade	**Chemistry grade**	**Man**	**Woman**
β	-0.1840	-0.3660	0	-0.2200

Table A.4: Coefficients for the logistic regression model 2

	DD	DP	PP	PD
Train	5	35	342	1
Test	1	10	85	2

Table A.5: Predictions using model table A.4

Misclassification ratio	0.0998
Drop out misclassification ratio	0.0936
Drop out ratio in all misclassification	0.9375
Not classified	23

Table A.6: Performance information for model table A.4

A.3 LR: Model 3

The model coefficients in table A.7 on the next page look more reasonable. However, as in earlier model there were warnings during the estimation. Still this model has low performance abilities.

	Interaction	Age	Lock stud.	In. stud.
β	-145.8170	0.0260	117.1710	-12.2550
	Tech. biomed. stud.	**Design stud.**	**Mat stud.**	**Biotech. stud.**
β	-36.5070	-38.6520	-36.1480	-37.3460
	Time af. exam	**GPA**	**Math lev. A**	**Math lev. B**
β	0.1000	-0.1950	74.3710	0
	Math lev. C	**Physics lev. A**	**Physics lev. B**	**Physics lev. C**
β	0	39.1270	39.7740	0.0000
	Chemistry lev. A	**Chemistry lev. B**	**Chemistry lev. C**	**Math grade**
β	-41.4560	-41.6390	-41.2480	-0.2380
	Physics grade	**Chemistry grade**	**Man**	**Woman**
β	-0.02200	-0.5850	0	-0.2200

Table A.7: Coefficient of logistic regression model 3.

	DD	DP	PP	PD
Train	5	28	342	2
Test	1	8	85	1

Table A.8: Predictions using model table A.7

Misclassification ratio	0.0826
Drop out misclassification ration	0.0763
Drop out ration in all misclassification	0.9231
Not classified	22

Table A.9: Model table A.7 performance information.

A.4 LR: Model 4

It can be noted in table A.10 on the facing page that the ratio of taken and passed ECTS credits is most significant among the performance characteristics from the first semester. Table A.11 on the next page shows the overall training prediction performance is much better than the test performance. It is most likely that the model did not catch the underling structure.

	Interaction	Age	Lock stud.	In. stud.
β	-28.6370	0.1280	159.4340	0
	Tech. biomed. stud.	**Design stud.**	**Mat stud.**	**Biotech. stud.**
β	-51.9410	-54.7600	-54.2770	-53.2230
	Time af. exam	**GPA**	**Math lev. A**	**Math lev. B**
β	-0.2460	-0.3780	0	0
	Math lev. C	**Physics lev. A**	**Physics lev. B**	**Physics lev. C**
β	0	-37.4600	-37.3960	0
	Chemistry lev. A	**Chemistry lev. B**	**Chemistry lev. C**	**Math grade**
β	-39.2180	-39.3280	-38.2030	-0.0530
	Physics grade	**Chemistry grade**	**Man**	**Woman**
β	0.4620	-0.5680	0.0000	-0.2200
	GPA O 1	**GPA S 1**	**ECTS P 1**	**ECTS T 1**
β	0.1950	0.1270	-0.3260	0.5130
	ECTS A 1	**ECTS R 1**	**ECTS L 1**	
β	-0.2500	-1.9980	0.0000	

Table A.10: Coefficients for the logistic regression model 4.

	DD	DP	PP	PD
Train	19	25	335	3
Test	1	11	84	1

Table A.11: Predictions using model table A.1 on page 89

Misclassification ratio	0.0835
Drop out misclassification ration	0.0752
Drop out ration in all misclassification	0.9000
Not classified	49

Table A.12: Performance information for model table A.10

A.5 LR: Model 5

From table A.13 on the following page it is difficult to interpret the result. For example the GPA overall after the second semester and GPA of second semester are highly correlated due to their nature. For this reason,

	Interaction	Age	Lock stud.	In. stud.
β	58.7380	26.9000	-93.2100	197.8790
	Tech. biomed. stud.	**Design stud.**	**Mat stud.**	**Biotech. stud.**
β	-136.2840	-292.6380	-169.8330	-142.1090
	Time af. exam	**GPA**	**Math lev. A**	**Math lev. B**
β	-82.3460	105.6830	1113.2170	0.0000
	Math lev. C	**Physics lev. A**	**Physics lev. B**	**Physics lev. C**
β	0.0000	-107.7690	-47.3640	0.0000
	Chemistry lev. A	**Chemistry lev. B**	**Chemistry lev. C**	**Math grade**
β	245.3770	600.1370	746.9250	-117.5030
	Physics grade	**Chemistry grade**	**Man**	**Woman**
β	-61.6550	-44.8300	0.0000	64.5770
	GPA O 1	**GPA O 2**	**GPA S 1**	**GPA S 2**
β	44.3950	-124.4460	0.2770	103.3640
	ECTS P 1	**ECTS P 2**	**ECTS T 1**	**ECTS T 2**
β	13.7160	-52.1930	-50.5980	19.4430
	ECTS A 1	**ECTS A 2**	**ECTS R 1**	**ECTS R 2**
β	0.0000	32.4580	-1969.5560	739.8820
	ECTS L 1	**ECTS L 2**		
β	0.0000	-506.2760		

Table A.13: Coefficients for the logistic regression model 5.

these two variables have high coefficient values but with opposite sign.

	DD	DP	PP	PD
Train	4	3	332	1
Test	0	3	81	6

Table A.14: Predictions using model fig. 6.6 on page 47

Although this model does have a low misclassification ratio it has a low prediction power.

Misclassification ratio	0.0302
Drop out misclassification ratio	0.0140
Drop out ratio in all misclassification	0.4615
Not classified	39

Table A.15: Performance information for model table A.13 on the facing page.

A.6 LR: Model 6

	Interaction	Age	Lock stud.	In. stud.
β	48.7210	-22.1190	-286.8410	-203.7010
	Tech. biomed. stud.	**Design stud.**	**Mat stud.**	**Biotech. stud.**
β	17.1800	-24.4050	60.3770	40.1240
	Time af. exam	**GPA**	**Math lev. A**	**Math lev. B**
β	18.4890	11.5320	-35.8910	0.0000
	Math lev. C	**Physics lev. A**	**Physics lev. B**	**Physics lev. C**
β	0.0000	-112.1210	-98.6510	21.8890
	Chemistry lev. A	**Chemistry lev. B**	**Chemistry lev. C**	**Math grade**
β	238.8280	110.3730	201.7730	6.6740
	Physics grade	**Chemistry grade**	**Man**	**Woman**
β	13.0800	-18.4750	0.0000	24.9890
	GPA O 1	**GPA O 2**	**GPA O 3**	**GPA S 1**
β	7.3370	28.6440	-10.0800	6.3800
	GPA S 2	**GPA S 3**	**ECTS P 1**	**ECTS P 2**
β	-17.5730	-7.7710	-35.5390	-50.3550
	ECTS P 3	**ECTS T 1**	**ECTS T 2**	**ECTS T 3**
β	-1.0820	26.7230	-2.6470	15.9420
	ECTS A 1	**ECTS A 2**	**ECTS A 3**	**ECTS R 1**
β	-36.9050	64.0060	-18.9740	301.8250
	ECTS R 2	**ECTS R 3**	**ECTS L 1**	**ECTS L 2**
β	27.4430	313.6580	0.0000	-104.3470
	ECTS L 3			
β	13.2090			

Table A.16: Coefficients for the logistic regression model 6.

	DD	DP	PP	PD
Train	6	1	335	0
Test	1	3	85	0

Table A.17: Predictions using model fig. 6.6 on page 47

Table A.16 shows that this model has the same problems as the previous models. However, it can be seen that the model emphasizes the ratio of passed and taken ECTS credits during first semesters.

Misclassification ratio	0.0093
Drop out misclassification ratio	0.0093
Drop out ratio in all misclassification	1
Not classified	37

Table A.18: Performance information for model table A.16 on the facing page.

A.7 LR: Model 7

	DD	**DP**	**PP**	**PD**
Train	2	0	335	0
Test	0	1	83	0

Table A.19: Predictions using model table A.21 on the next page

Misclassification ratio	0.0024
Drop out misclassification ratio	0.0024
Drop out ratio in all misclassification	1
Not classified	35

Table A.20: Performance information for model table A.21 on the next page.

Table A.19 shows that model identified correctly all pass students. The same plot shows that model correctly classified dropouts in training set while in test it missed. This is due to small dropouts number in the training set.

	Interaction	Age	Lock stud.	In. stud.
β	-52.6730	-4.7950	-48.6170	-58.9840
	Tech. biomed. stud.	**Design stud.**	**Mat stud.**	**Biotech. stud.**
β	0.0000	16.4000	50.0270	61.7250
	Time af. exam	**GPA**	**Math lev. A**	**Math lev. B**
β	4.0500	-18.8170	-13.5530	0.0000
	Math lev. C	**Physics lev. A**	**Physics lev. B**	**Physics lev. C**
β	0.0000	-31.5970	-39.2630	0.0000
	Chemistry lev. A	**Chemistry lev. B**	**Chemistry lev. C**	**Math grade**
β	-117.9800	-110.4200	-117.7010	5.1410
	Physics grade	**Chemistry grade**	**Man**	**Woman**
β	-11.5560	19.7970	0.0000	1.1530
	GPA O 1	**GPA O 2**	**GPA O 3**	**GPA O 4**
β	-12.4990	52.7470	-6.9470	-36.0350
	GPA S 1	**GPA S 2**	**GPA S 3**	**GPA S 4**
β	0.0000	-15.9480	11.3880	8.8080
	ECTS P 1	**ECTS P 2**	**ECTS P 3**	**ECTS P 4**
β	-2.6940	-1.1360	-5.5330	-14.9100
	ECTS T 1	**ECTS T 2**	**ECTS T 3**	**ECTS T 4**
β	2.9500	1.4370	-4.1140	5.2410
	ECTS A 1	**ECTS A 2**	**ECTS A 3**	**ECTS A 4**
β	0.0000	-13.7760	4.8270	7.7260
	ECTS R 1	**ECTS R 2**	**ECTS R 3**	**ECTS R 4**
β	85.0670	151.5130	-125.7120	136.5570
	ECTS L 1	**ECTS L 2**	**ECTS L 3**	**ECTS L 4**
β	0.0000	-103.8210	97.0010	-6.7590

Table A.21: Coefficients for the logistic regression model 7.

A.8 LR: Model 8

	Interaction	Age	Lock stud.	In. stud.
β	191.6930	5.8580	149.1860	0.0000
	Tech. biomed. stud.	**Design stud.**	**Mat stud.**	**Biotech. stud.**
β	0.0000	287.4420	186.7700	249.5920
	Time af. exam	**GPA**	**Math lev. A**	**Math lev. B**
β	-20.6310	-13.5890	3.6590	0.0000
	Math lev. C	**Physics lev. A**	**Physics lev. B**	**Physics lev. C**
β	0.0000	43.7140	-18.2440	23.5720
	Chemistry lev. A	**Chemistry lev. B**	**Chemistry lev. C**	**Math grade**
β	110.8750	67.6620	83.8060	-6.5800
	Physics grade	**Chemistry grade**	**Man**	**Woman**
β	4.8570	-21.6060	0.0000	-26.0750
	GPA O 1	**GPA O 2**	**GPA O 3**	**GPA O 4**
β	4.3560	8.4510	-33.3630	-14.4330
	GPA O 5	**GPA S 1**	**GPA S 2**	**GPA S 3**
β	42.4270	3.3510	5.8460	7.2700
	GPA S 4	**GPA S 5**	**ECTS P 1**	**ECTS P 2**
β	1.1100	-7.9150	-13.2290	-28.7930
	ECTS P 3	**ECTS P 4**	**ECTS P 5**	**ECTS T 1**
β	-79.8320	13.8150	2.6010	-18.1430
	ECTS T 2	**ECTS T 3**	**ECTS T 4**	**ECTS T 5**
β	-15.2690	18.4880	2.2630	-6.3680
	ECTS A 1	**ECTS A 2**	**ECTS A 3**	**ECTS A 4**
β	-15.8120	-8.2510	71.2860	-18.2060
	ECTS A 5	**ECTS R 1**	**ECTS R 2**	**ECTS R 3**
β	2.8250	-269.2820	-668.8170	354.6240
	ECTS R 4	**ECTS R 5**	**ECTS L 1**	**ECTS L 2**
β	128.7720	-218.6120	0.0000	148.5470
	ECTS L 3	**ECTS L 4**	**ECTS L 5**	
β	-12.5840	-96.0870	63.1400	

Table A.22: Coefficients for the logistic regression model 8.

The sixth semester model is no different from the previous models. It faces the same problems: warning during model estimation and some

	DD	DP	PP	PD
Train	6	0	333	0
Test	0	3	80	2

Table A.23: Predictions using model table A.22 on the preceding page

Misclassification ratio	0.0118
Drop out misclassification ratio	0.0071
Drop out ratio in all misclassification	0.6000
Not classified	36

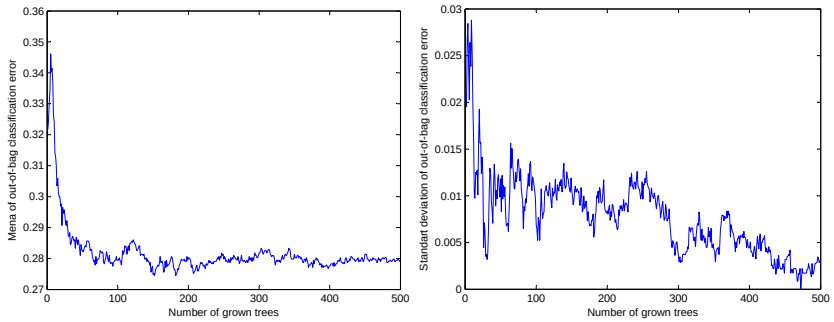
Table A.24: Performance information model table A.22 on the preceding page.

of coefficients have very high estimates. Also there are only a few coefficients set to zero, which makes the model difficult to interpret.

Appendix B

CART Bagging Models for Every Semester

B.1 CART Bagging: Model 1



(a) Mean. (b) Standard deviation.

Figure B.1: Identification of the number trees required for model 1 using out-of-bag error of five models.

Figure B.1 shows the mean stabilizes around 200 trees, but the standard

deviation only around 300. It is also important to note, that even using a large number of trees the out-of-bag error is high.

	DD	DP	PP	PD
Train	149	8	359	0
Test	8	33	85	6

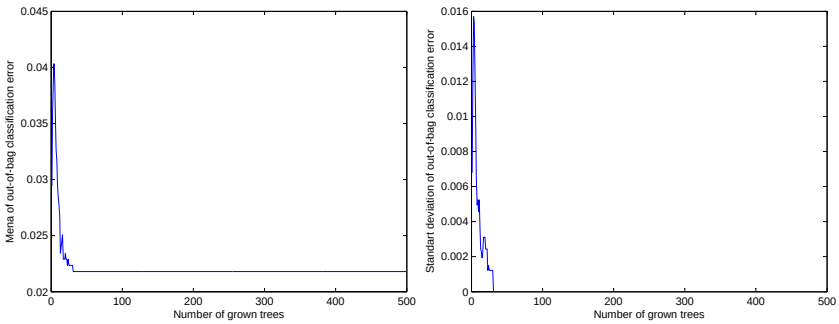
Table B.1: Predictions using model fig. B.1 on the previous page

Misclassification ratio	0.0735
Drop out misclassification ratio	0.0633
Drop out ratio in all misclassification	0.8723
Chosen number of trees	300

Table B.2: Performance information on model fig. B.1 on the previous page.

Table B.2 shows that the model performance on the training data is very good. However with the test set it is poor. This could be an indication, that the method overfits.

B.2 CART Bagging: Model 2



(a) Mean. (b) Standard deviation.

Figure B.2: Identification of the number trees required for model 2 using out-of-bag error of five models.

It is seen in fig. B.2 on the preceding page that the mean and standard deviation of the out-of-bag error stabilises when there are more than 40 trees in the model.

	DD	DP	PP	PD
Train	4	4	359	0
Test	0	2	91	0

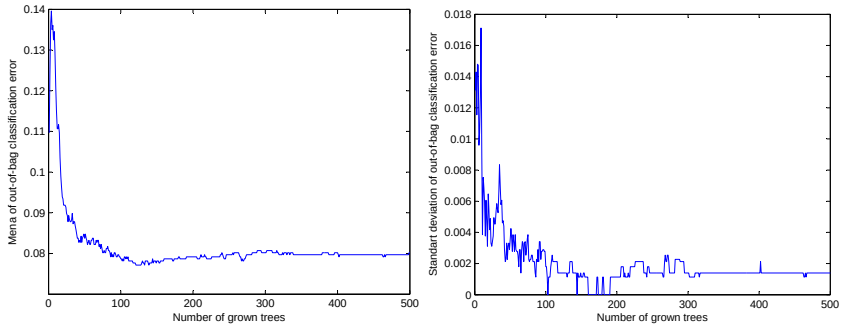
Table B.3: Predictions using model fig. B.2 on the preceding page

Misclassification ratio	0.0130
Drop out misclassification ratio	0.0130
Drop out ratio in all misclassification	1
Chosen number of trees	40

Table B.4: Performance information on model fig. B.2 on the preceding page.

As in model 1, model 2 is also overfitting. For the training set the model performs decent, but it fails to predict the dropouts in the test set.

B.3 CART Bagging: Model 3



(a) Mean. (b) Standart deviation.

Figure B.3: Identification of the number trees required for model 3 using out-of-bag error of five models..

	DD	DP	PP	PD
Train	33	2	359	0
Test	1	8	91	1

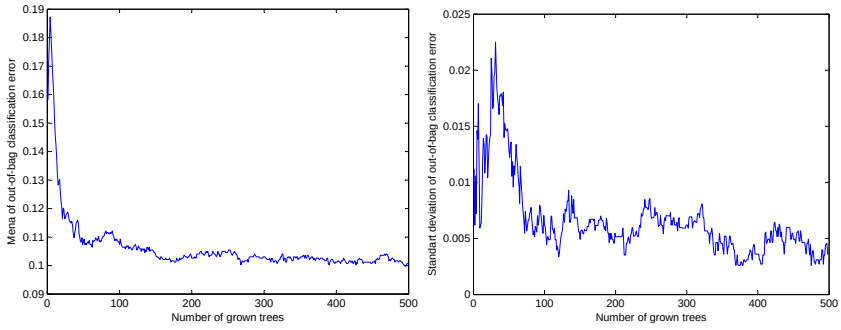
Table B.5: Predictions using model fig. B.3 on the previous page

Misclassification ratio	0.0202
Drop out misclassification ratio	0.0202
Drop out ratio in all misclassification	1
Chosen number of trees	300

Table B.6: Performance information on model fig. B.3 on the previous page.

The mean stabilizes in fig. B.3a on the preceding page between 250 and 300 trees. The standard deviation became almost constant after 300 trees. This model is also overfitting.

B.4 CART Bagging: Model 4



(a) Mean. (b) Standard deviation.

Figure B.4: Identification of the number trees required for model 4 using out-of-bag error of five models.

Model 4 performs well on the training set. It do not identify 3 dropouts. Yet on the test set it do not identify 14 dropouts.

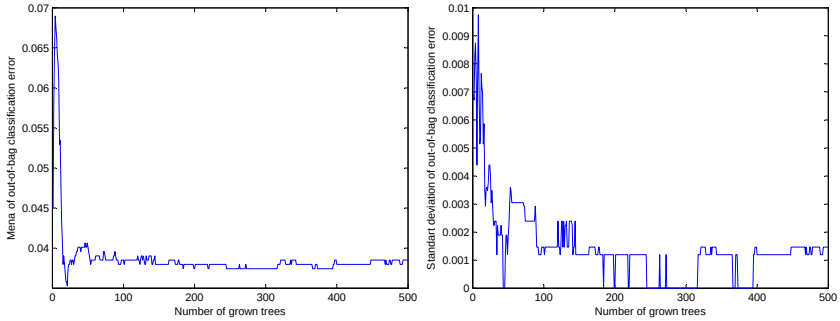
	DD	DP	PP	PD
Train	59	3	359	0
Test	2	14	89	2

Table B.7: Predictions using model fig. B.4 on the facing page

Misclassification ratio	0.0360
Drop out misclassification ratio	0.03022
Drop out ratio in all misclassification	0.8947
Chosen number of trees	150

Table B.8: Performance information on model fig. B.4 on the facing page.

B.5 CART Bagging: Model 5



(a) Mean. (b) Standard deviation.

Figure B.5: Identification of the number trees required for model 5 using out-of-bag error of five models..

Model stabilises with 100 trees.

From table B.10 on the next page it is seen that model 5 performs well with the training set. It identifies 93% of all dropouts. On the test set the model identifies only 50% of the dropouts, but raises no false alarms.

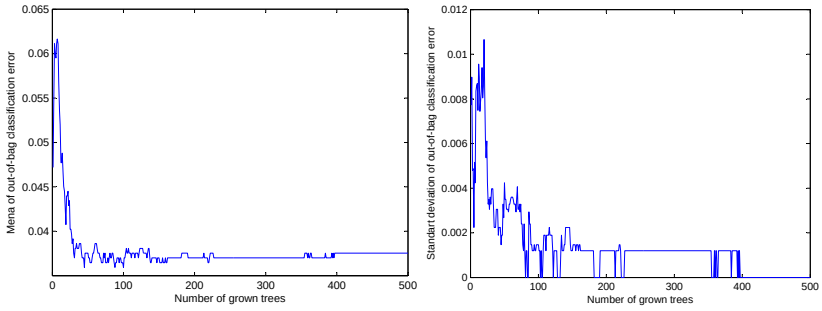
	DD	DP	PP	PD
Train	13	2	359	0
Test	2	2	91	0

Table B.9: Predictions using model fig. B.5 on the preceding page

Misclassification ratio	0.0085
Drop out misclassification ratio	0.0085
Drop out ratio in all misclassification	1
Chosen number of trees	100

Table B.10: Performance information on model fig. B.5 on the preceding page.

B.6 CART Bagging: Model 6



(a) Mean. (b) Standard deviation.

Figure B.6: Identification of the number trees required for model 6 using out-of-bag error of five models.

Both the mean and standard deviation stabilizes around 100 trees. As the previous models, this model performs well on the training set, but fails on the test set. It identified 12 out of 14 dropouts, with no false alarms on the training data set while on the test set it did not identify any dropouts.

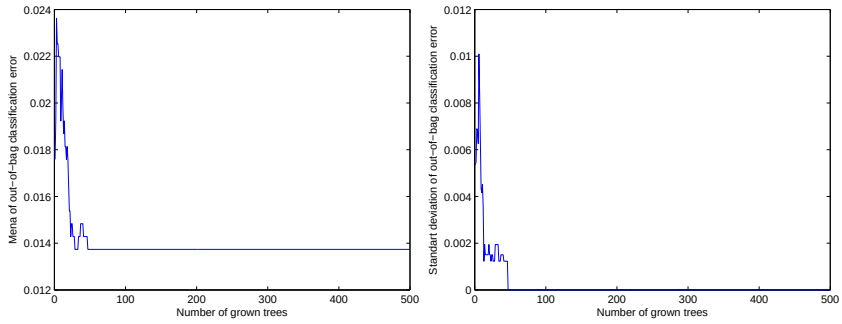
	DD	DP	PP	PD
Train	12	2	359	0
Test	0	4	91	0

Table B.11: Predictions using model fig. B.6 on the preceding page

Misclassification ratio	0.0128
Drop out misclassification ration	0.0128
Drop out ration in all misclassification	1
Chosen number of trees	100

Table B.12: Performance information on model fig. B.6 on the preceding page.

B.7 CART Bagging: Model 7



(a) Mean. (b) Standard deviation.

Figure B.7: Identification of the number trees required for model 7 using out-of-bag error of five models.

The model well identified pass students, but fails to identify dropouts. Most likely it is due to the small ratio of drop out students in the training set. There were only 5 dropouts in the training set.

	DD	DP	PP	PD
Train	3	2	359	0
Test	0	1	91	0

Table B.13: Predictions using model fig. B.7 on the previous page

Misclassification ratio	0.0066
Drop out misclassification ration	0.0066
Drop out ration in all misclassification	1
Chosen number of trees	50

Table B.14: Performance on model fig. B.7 on the previous page.

B.8 CART Bagging: Model 8

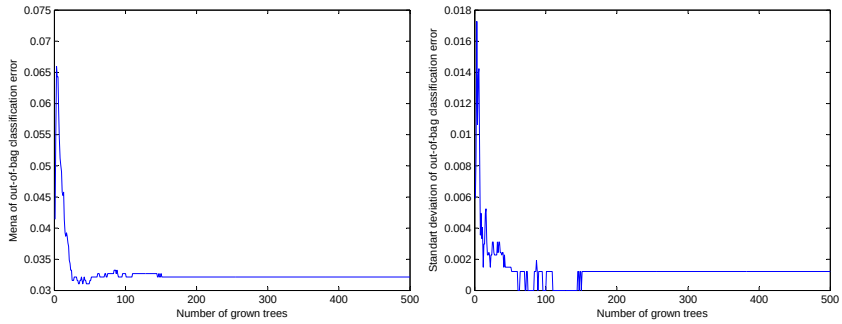
The out-of-bag error is constant in mean and standard deviation after some 150 trees. Table B.15 identifies all drop out and pass student in training set, but fails to identify the dropouts in the test set. As in the previous models it is due to the small number of dropouts (12) so the model overfits.

	DD	DP	PP	PD
Train	12	0	355	0
Test	0	3	90	0

Table B.15: Predictions using model fig. B.8 on the next page

Misclassification ratio	0.0065
Drop out misclassification ratio	0.0065
Drop out ratio in all misclassification	1
Chosen number of trees	150

Table B.16: Performance information on model fig. B.8.



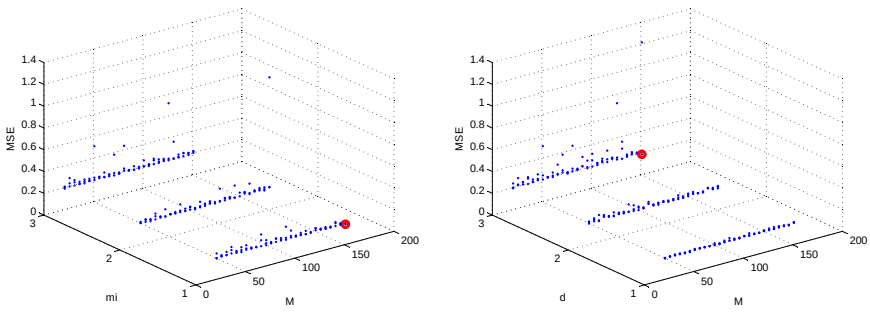
(a) Mean. (b) Standard deviation.

Figure B.8: Identification of the number trees required for model 8 using out-of-bag error of five models.

Appendix C

MARS Models for Every Semester

C.1 MARS: Model 1



(a) Parameters *mi* and *M*. (b) Parameters *d* and *M*.

Figure C.1: Variable identification for model 1.

$$
\begin{aligned}
BF1 &= max(0, 0.34232 - \textit{school GPA}) \\
BF2 &= max(0, \textit{Design and Innovation programme} + 0.70027) \\
BF3 &= max(0, \textit{chemistry exam grade} - 1.9528) \\
BF4 &= max(0, \textit{physics exam grade} - 0.67752) \\
BF5 &= max(0, \textit{physics exam grade} - 0.90619) \\
BF6 &= max(0, \textit{chemistry exam grade} - 0.91098) \\
BF7 &= max(0, \textit{chemistry exam grade} - 1.1194) \\
BF8 &= max(0, -0.46585 - \textit{physics exam grade}) \\
y &= 0.31632 + 0.082916 \cdot BF1 - 0.099606 \cdot BF2 \\
&\quad - 2.2102 \cdot BF3 - 1.034 \cdot BF4 + 1.5881 \cdot BF5 \\
&\quad - 1.2418 \cdot BF6 + 1.9649 \cdot BF7 + 0.13511 \cdot BF8
\end{aligned}
\tag{C.1}
$$

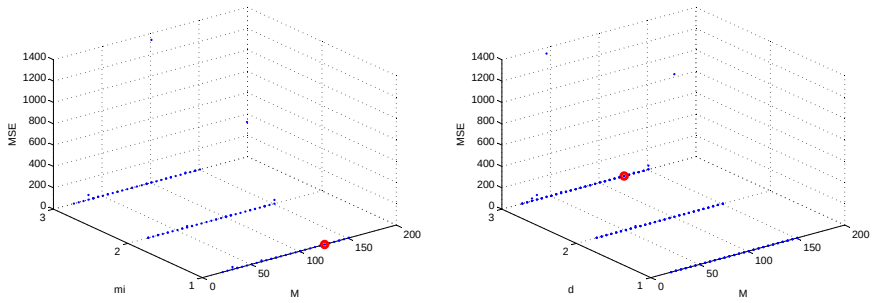
	DD	**DP**	**PP**	**PD**
Train	48	109	336	13
Test	4	37	85	6

Table C.1: Predictions using model eq. (C.1)

Misclassification ratio	0.2546
Drop out misclassification ratio	0.2253
Drop out ratio in all misclassification	0.8848

Table C.2: Model eq. (C.1) performance information.

As seen in fig. C.1 on the previous page the mean square error is almost constant for all combinations of d, mi and M. However increasing d or mi does increase the variation of the error. The optimal values are $M = 151$, $mi = 1$ and $d = 1$. As in many of the previous models the most important characteristics are exam grades, school GPA and Design and Innovation programme. Unfortunately the model performances is very poorly.



(a) Parameters *mi* and *M*. (b) Parameters *d* and *M*.

Figure C.2: Variable identification for model 2.

C.2 MARS: Model 2

$$
\begin{aligned}
BF1 &= max(0, mathematics\ exam\ grade + 1.9441) \\
BF2 &= max(0, mathematics\ exam\ grade + 1.5453) \\
BF3 &= max(0, -1.1797 - physical\ exam\ grade) \\
y &= 0.27218 - 0.60897 \cdot BF1 + 0.63589 \cdot BF2 + 0.20369 \cdot BF3 \qquad \text{(C.2)}
\end{aligned}
$$

	DD	DP	PP	PD
Train	4	39	358	1
Test	0	11	91	2

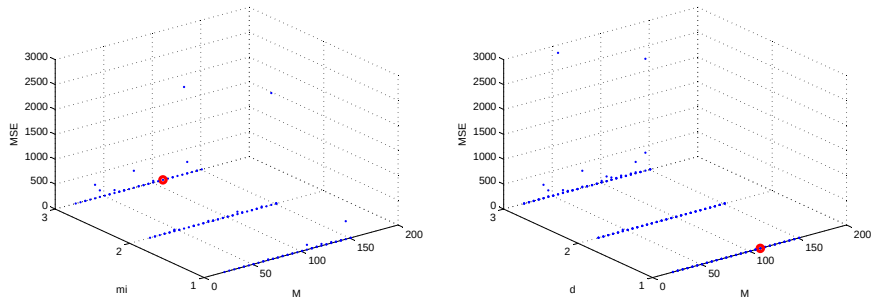
Table C.3: Predictions using model eq. (C.2)

Misclassification ratio	0.1012
Drop out misclassification ratio	0.0992
Drop out ratio in all misclassification	0.9804

Table C.4: Model eq. (C.2) performance information.

The optimal values were chosen to $M = 140$, $mi = 1$ and $d = 1$. Most of the analysed methods failed to build model 2. MARS is no exception.

C.3 MARS: Model 3



(a) Parameters *mi* and *M*. (b) Parameters *d* and *M*.

Figure C.3: Variable identification for model 3.

$$
\begin{aligned}
BF1 &= max(0, physics\ exam\ grade + 1.0746)\\
BF2 &= max(0, mathematics\ exam\ grade + 1.9439)\\
BF3 &= max(0, 1.3819 - physics\ exam\ grade)\\
BF4 &= max(0, Design\ an\ Innovation\ programme + 0.78729)\\
BF5 &= max(0, chemistry\ level\ B + 0.8302)\\
BF6 &= max(0, mathematics\ exam\ grade + 1.546)\\
BF7 &= max(0, age - 0.054309)\\
BF8 &= max(0, age - 0.082829)\\
BF9 &= max(0, age - 0.029864)\\
y &= 0.13244 + 0.10927 \cdot BF1 - 1.0078 \cdot BF2 + 0.14035 \cdot BF3 \qquad \text{(C.3)}\\
&\quad - 0.052549 \cdot BF4 - 0.037993 \cdot BF5 + 1.0464 \cdot BF6 - 40.037 \cdot BF7\\
&\quad + 17.459 \cdot BF8 + 22.577 \cdot BF9
\end{aligned}
$$

	DD	DP	PP	PD
Train	7	28	357	2
Test	1	8	98	2

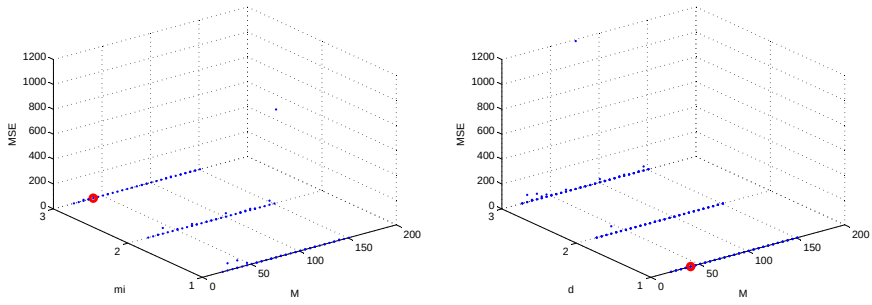
Table C.5: Predictions using model eq. (C.3)

Misclassification ratio	0.0810
Drop out misclassification ratio	0.0729
Drop out ratio in all misclassification	0.900

Table C.6: Model eq. (C.3) on the preceding page performance information.

The optimal values were chosen to $M = 100$, $m = 1$ and $d = 1$. The misclassification rate (0.0810) is very low, however the model does not identify many of true dropouts. As seen in eq. (C.3) on the facing page the model selects the age characteristic three times while mathematics and physics exam grades are selected twice. It seem that the model tries to build very precise group classification boundaries that led to poor performance.

C.4 MARS: Model 4



(a) Parameters *mi* and M.

(b) Parameters d and M.

Figure C.4: Variable identification for model 4.

$$
\begin{aligned}
BF1 &= max(0, ECTS\ R\ 1 + 0.72667) \\
BF2 &= max(0, -0.34271 - chemistry\ exam\ grade) \\
BF3 &= max(0, physics\ exam\ grade + 0.64253) \\
BF4 &= max(0, shool\ GPA + 2.1348) \\
BF5 &= max(0, 1.61 - Biotechnology\ programme) \\
y &= 0.82796 - 0.62631 \cdot BF1 + 0.098617 \cdot BF2 + 0.089562 \cdot BF3 \\
&\quad - 0.060304 \cdot BF4 - 0.045833 \cdot BF5
\end{aligned} \tag{C.4}
$$

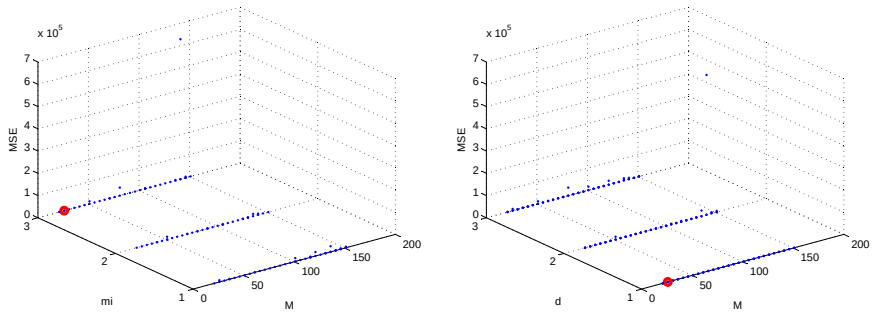
	DD	**DP**	**PP**	**PD**
Train	29	33	351	8
Test	8	8	87	4

Table C.7: Predictions using model eq. (C.4)

Misclassification ratio	0.1004
Drop out misclassification ratio	0.0777
Drop out ratio in all misclassification	0.7736

Table C.8: Model eq. (C.4) performance information.

The optimal values are $M = 50$, $mi = 1$ and $d = 1$. In this case around 50% dropouts were identified with 20-30% false alarm rate. All characteristics where chosen once and the model used 5 basic functions. Among the models for the second semester that use other techniques, the model chose school exam grades and ratio of passed and taken ECST credits after the first semester.



(a) Parameters *mi* and M. (b) Parameters d and M.

Figure C.5: Variable identification for model 5.

C.5 MARS: Model 5

$$
\begin{aligned}
BF1 &= max(0, mathematics\ exam\ grade + 0.11625)\\
BF2 &= max(0, physics\ level\ C + 0.051709)\\
BF3 &= max(0, ECTS\ R\ 1 + 1.6956)\\
BF4 &= max(0, -1.6956 - ECTS\ R\ 1)\\
BF5 &= max(0, -2.6567 - ECTS\ A\ 2)\\
BF6 &= max(0, 0.89165 - ECTS\ T\ 2)\\
BF7 &= max(0, 0.97359 - ECTS\ P\ 2)\\
y &= 0.22933 + 0.042831 \cdot BF1 - 0.019974 \cdot BF2 - 0.11515 \cdot BF3 \quad \text{(C.5)}\\
&\quad - 0.090297 \cdot BF4 + 0.24416 \cdot BF5 - 0.39499 \cdot BF6 + 0.38565 \cdot BF7
\end{aligned}
$$

	DD	DP	PP	PD
Train	9	6	359	0
Test	3	1	91	0

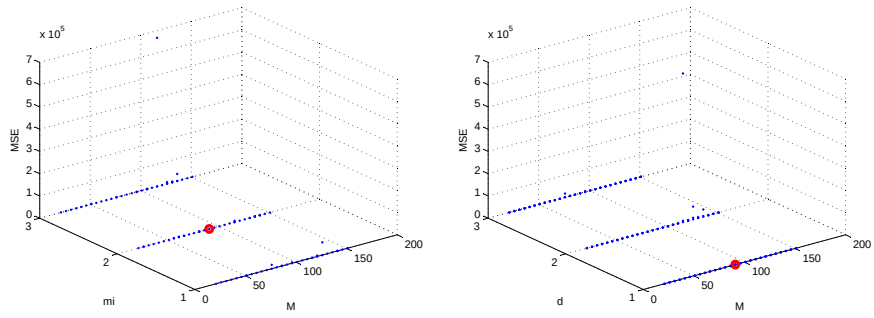
Table C.9: Predictions using model eq. (C.5)

The optimal values were chosen to $M = 30$, $mi = 1$ and $d = 1$. The model performs with a false alarm rate at zero and identifies 63% of all dropouts after third semester.

Misclassification ratio	0.0149
Drop out misclassification ratio	0.0149
Drop out ratio in all misclassification	1

Table C.10: Model eq. (C.5) on the preceding page performance information.

C.6 MARS: Model 6



(a) Parameters *mi* and *M*. (b) Parameters *d* and *M*.

Figure C.6: Variable identification for model 6.

$$
\begin{aligned}
BF1 &= max(0, -1.4015 - ECTS\ R\ 3) \\
BF2 &= max(0, ECTS\ R\ 2 + 1.2037) \\
BF3 &= max(0, mathematics\ exam\ grade - 1.8054) \\
BF4 &= max(0, ECTS\ P\ 3 + 1.8701) \\
BF5 &= max(0, mathematics\ level\ B + 0.089924) \\
BF6 &= max(0, ECTS\ R\ 2 + 1.0254) \\
BF7 &= max(0, physics\ levelB\ + 0.051778) \\
BF8 &= max(0, physics\ exam\ grade - 1.6228) \\
BF9 &= max(0, -0.97217 - ECTS\ P\ 2) \\
BF10 &= max(0, -1.0282 - ECTS\ T\ 2) \\
BF11 &= max(0, -2.0505 - school\ GPA) \\
BF12 &= max(0, -1.9469 - ECTS\ P\ 2) \\
BF13 &= max(0, ECTS\ P\ 3 + 1.2039) \\
BF14 &= max(0, -1.2039 - ECTS\ T\ 3) \\
BF15 &= max(0, GPA\ O\ 2 + 1.8835) \\
BF16 &= max(0, -1.2209 - GPA\ S\ 2) \\
BF17 &= max(0, GPA\ O\ 2 + 0.30493) \\
BF18 &= max(0, -0.30493 - GPA\ O\ 2) \\
BF19 &= max(0, ECTS\ A\ 3 + 1.7934) \\
BF20 &= max(0, ECTS\ A\ 3 + 1.0439) \\
BF21 &= max(0, -1.0439 - ECTS\ A\ 3) \\
BF22 &= max(0, -2.1681 - ECTS\ A\ 3)
\end{aligned}
$$

$$
\begin{aligned}
y = {} & 1.6134 + 0.06998 \cdot BF1 - 1.1624 \cdot BF2 + 0.55959 \cdot BF3 \\
& - 0.3315 \cdot BF4 + 0.045761 \cdot BF5 + 1.1938 \cdot BF6 - 0.02721 \cdot BF7 \\
& + 0.24682 \cdot BF8 + 1.0294 \cdot BF9 - 0.93138 \cdot BF10 - 0.15034 \cdot BF11 \\
& - 0.91951 \cdot BF12 + 0.3205 \cdot BF13 - 0.20559 \cdot BF14 - 0.22006 \cdot BF15 \\
& + 0.16247 \cdot BF16 + 0.20949 \cdot BF17 - 0.23919 \cdot BF18 - 1.1545 \cdot BF19 \\
& + 1.1628 \cdot BF20 - 0.80525 \cdot BF21 + 0.76949 \cdot BF22
\end{aligned} \tag{C.6}
$$

The optimal values are $M = 51$, $mi = 1$ and $d = 1$. The model performs decent as it identifies 58% of all dropouts with 2% misclassification rate.

	DD	DP	PP	PD
Train	9	5	358	1
Test	1	2	89	2

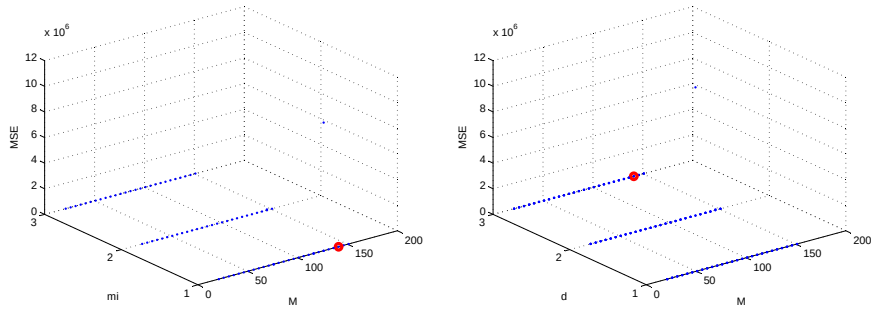
Table C.11: Predictions using model eq. (C.6) on the preceding page

Misclassification ratio	0.0235
Drop out misclassification ratio	0.0171
Drop out ratio in all misclassification	0.7273

Table C.12: Model eq. (C.6) on the preceding page performance information.

However the model complexity is just unacceptable. The model used 22 basic functions.

C.7 MASR: Model 7



(a) Parameters *mi* and *M*. (b) Parameters *d* and *M*.

Figure C.7: Variable identification for model 7.

$$
\begin{aligned}
BF1 &= max(0, physics\ exam\ grade + 1.9439)\\
BF2 &= max(0, -0.73559 - ECTS\ R\ 4)\\
BF3 &= max(0, 0.48563 - physics\ exam\ grade)\\
BF4 &= max(0, age + 0.91393)\\
BF5 &= max(0, -0.91393 - age)\\
BF6 &= max(0, -1.0295 - ECTS\ T\ 2)\\
BF7 &= max(0, ECTS\ P\ 2 + 1.0121)\\
BF8 &= max(0, age + 0.88543)\\
BF9 &= max(0, GPA\ S\ 2 + 1.7638)\\
BF10 &= max(0, 1.5915 - GPA\ S\ 2)\\
BF11 &= max(0, ECTS\ P\ 2 + 2.0065)\\
BF12 &= max(0, chemistry\ exam\ grade + 0.82668)\\
BF13 &= max(0, physics\ exam\ grade - 0.19409)\\
BF14 &= max(0, 0.19409 - physics\ exam\ grade)\\
BF15 &= max(0, physics\ exam\ grade0 - 0.68)\\
BF16 &= max(0, 0.68 - physics\ exam\ grade)\\
BF17 &= max(0, school\ GPA + 0.11428)\\
y &= 1.0553 + 0.3294 \cdot BF1 + 0.030421 \cdot BF2 - 0.67208 \cdot BF3 \qquad \text{(C.7)}\\
&\quad - 3.3647 \cdot BF4 - 0.64363 \cdot BF5 - 0.53151 \cdot BF6 + 0.46353 \cdot BF7\\
&\quad + 3.3485 \cdot BF8 - 0.11542 \cdot BF9 - 0.10965 \cdot BF10 - 0.47915 \cdot BF11\\
&\quad + 0.024294 \cdot BF12 - 1.9305 \cdot BF13 + 2.194 \cdot BF14 + 1.6114 \cdot BF15\\
&\quad - 1.2111 \cdot BF16 - 0.034379 \cdot BF17
\end{aligned}
$$

	DD	**DP**	**PP**	**PD**
Train	2	3	359	0
Test	0	1	91	0

Table C.13: Predictions using model eq. (C.7)

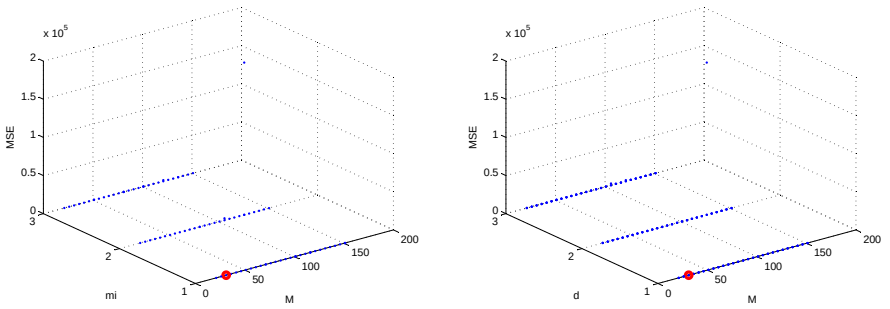
The optimal values are $M = 100$, $mi = 1$ and $d = 1$. The model identities passed students well, but it fails to identify dropouts. It is because model was trained with merely 5 dropouts in the training data set. Equation (C.7) is a complex model as it has 17 basic functions and some of

Misclassification ratio	0.0088
Drop out misclassification ratio	0.0088
Drop out ratio in all misclassification	1

Table C.14: Model eq. (C.7) on the preceding page performance information.

the characteristics were chosen several times. For example, physic exam grade was chosen 4 times.

C.8 MARS: Model 8



(a) Parameters *mi* and *M*. (b) Parameters *d* and *M*.

Figure C.8: Variable identification for model 8.

$$
\begin{aligned}
BF1 &= max(0, -2.2372 - ECTS\ A\ 3) \\
BF2 &= max(0, -2.003 - ECTS\ R\ 5) \\
BF3 &= max(0, -1.99 - ECTS\ R\ 3) \\
BF4 &= max(0, -2.7095 - ECTS\ T\ 5) \\
BF5 &= max(0, -0.65111 - ECTS\ T\ 1) \\
y &= -0.0028257 + 0.18207 \cdot BF1 + 0.11221 \cdot BF2 \\
&\quad + 0.074542 \cdot BF3 + 1.0831 \cdot BF4 + 0.095512 \cdot BF5
\end{aligned}
\tag{C.8}
$$

	DD	DP	PP	PD
Train	6	6	354	1
Test	0	3	90	0

Table C.15: Predictions using model eq. (C.8) on the preceding page

Misclassification ratio	0.0196
Drop out misclassification ratio	0.0196
Drop out ratio in all misclassification	1

Table C.16: Model eq. (C.8) on the preceding page performance information.

The optimal values are $M = 10$, $mi = 2$ and $d = 2$. Table C.15 shows the model performs well on the training set. The model eq. (C.8) on the preceding page is reasonable simple. It has just 5 basic functions. This model takes performance information from the first, third and fifth semester.

Appendix D

MATLAB Code

D.1 File: `Main.m`

```
% Read data frol .xlsx file
[num1_bio,txt1_bio,raw1_bio] =...
    xlsread('early_warning_2_tekbio.xlsx',2,'A2:V210');
[num2_bio,txt2_bio,raw2_bio] =...
    xlsread('early_warning_2_tekbio.xlsx',1,'A2:J3021');
[num1_dis,txt1_dis,raw1_dis] =...
    xlsread('early_warning_2_design.xlsx',2,'A2:V613');
[num2_dis,txt2_dis,raw2_dis] =...
    xlsread('early_warning_2_design.xlsx',1,'A2:J9360');
[num1_mat,txt1_mat,raw1_mat] =...
    xlsread('early_warning_system_2_mat.xlsx',2,'A2:V406');
[num2_mat,txt2_mat,raw2_mat] =...
    xlsread('early_warning_system_2_mat.xlsx',1,'A2:J5956');
[num1_btek,txt1_btek,raw1_btek] =...
    xlsread('early_warning_2_biotek.xlsx',1,'A2:V522');
[num2_btek,txt2_btek,raw2_btek] =...
    xlsread('early_warning_2_biotek.xlsx',2,'A2:J7323');
%% Data of BC Biotechnology programme
prog = 1;
```

```
num2_bio = SortExamps(raw2_bio,num2_bio,num1_bio,0);
[stat_num_bio_BC,stat_txt_bio_BC] = Status(raw1_bio,...
    num1_bio,num2_bio,prog);
num2_bio_BC = PerformanceInformation(num2_bio,...
    stat_num_bio_BC, prog);
info_bio_BC = PersonalInfo(raw1_bio,num1_bio,...
    stat_num_bio_BC);
info_s_bio_BC = PersonalInfo_short(raw1_bio,num1_bio,...
    stat_num_bio_BC);
[ECTS_T_bio_BC, ECTS_P_bio_BC, ECTS_R_bio_BC,...
    ECTS_A_bio_BC,ECTS_L_bio_BC, GPA_S_bio_BC,...
    GPA_O_bio_BC] = PerformanceInfo_Summury...
    (stat_num_bio_BC, num2_bio_BC,prog);
% Data of BC Design and Inovation programme
prog = 1;
num2_dis = SortExamps(raw2_dis,num2_dis,num1_dis,0);
[stat_num_dis_BC,stat_txt_dis_BC] = Status(raw1_dis,...
    num1_dis,num2_dis,prog);
num2_dis_BC = PerformanceInformation(num2_dis,...
    stat_num_dis_BC, prog);
info_dis_BC = PersonalInfo(raw1_dis,num1_dis,stat_num_dis_BC);
info_s_dis_BC = PersonalInfo_short(raw1_dis,...
    num1_dis,stat_num_dis_BC);
[ECTS_T_dis_BC, ECTS_P_dis_BC, ECTS_R_dis_BC,...
    ECTS_A_dis_BC, ECTS_L_dis_BC,GPA_S_dis_BC,...
    GPA_O_dis_BC]= PerformanceInfo_Summury...
    (stat_num_dis_BC, num2_dis_BC,prog);
% Data of BC Mathematics and Technologu programme
prog = 1;
num2_mat = SortExamps(raw2_mat,num2_mat,num1_mat,0);
[stat_num_mat_BC,stat_txt_mat_BC] =...
    Status(raw1_mat,num1_mat,num2_mat,prog);
num2_mat_BC = PerformanceInformation...
    (num2_mat, stat_num_mat_BC, prog);
info_mat_BC = PersonalInfo(raw1_mat,num1_mat,stat_num_mat_BC);
info_s_mat_BC = PersonalInfo_short...
    (raw1_mat,num1_mat,stat_num_mat_BC);
[ECTS_T_mat_BC, ECTS_P_mat_BC, ECTS_R_mat_BC,...
    ECTS_A_mat_BC, ECTS_L_mat_BC, GPA_S_mat_BC,...
    GPA_O_mat_BC] = PerformanceInfo_Summury...
    (stat_num_mat_BC, num2_mat_BC,prog);
% Data of BC Biomedicine programme
prog = 1;
num2_btek = SortExamps(raw2_btek,num2_btek,num1_btek, 1);
[stat_num_btek_BC,stat_txt_btek_BC] =...
    Status(raw1_btek,num1_btek,num2_btek,prog);
num2_btek_BC = PerformanceInformation...
```

```
67      (num2_btek, stat_num_btek_BC, prog);
68  info_btek_BC = PersonalInfo...
69      (raw1_btek,num1_btek,stat_num_btek_BC);
70  info_s_btek_BC = PersonalInfo_short...
71      (raw1_btek,num1_btek,stat_num_btek_BC);
72  [ECTS_T_btek_BC, ECTS_P_btek_BC, ECTS_R_btek_BC,...
73      ECTS_A_btek_BC,ECTS_L_btek_BC, GPA_S_btek_BC,...
74      GPA_O_btek_BC] = PerformanceInfo_Summury...
75      (stat_num_btek_BC, num2_btek_BC,prog);
76  % BC data concatenation
77  stat_num_BC= [stat_num_bio_BC;stat_num_dis_BC;...
78      stat_num_mat_BC; stat_num_btek_BC];
79  stat_txt_BC =[stat_txt_bio_BC;stat_txt_dis_BC;...
80      stat_txt_mat_BC;stat_txt_btek_BC];
81  ECTS_T_BC = [ECTS_T_bio_BC; ECTS_T_dis_BC;...
82      ECTS_T_mat_BC; ECTS_T_btek_BC;];
83  ECTS_P_BC = [ECTS_P_bio_BC; ECTS_P_dis_BC;...
84      ECTS_P_mat_BC;ECTS_P_btek_BC];
85  ECTS_A_BC = [ECTS_A_bio_BC; ECTS_A_dis_BC;...
86      ECTS_A_mat_BC;ECTS_A_btek_BC;];
87  ECTS_L_BC = [ECTS_L_bio_BC; ECTS_L_dis_BC;...
88      ECTS_L_mat_BC;ECTS_L_btek_BC];
89  ECTS_R_BC = [ECTS_R_bio_BC; ECTS_R_dis_BC;...
90      ECTS_R_mat_BC;ECTS_R_btek_BC];
91  GPA_S_BC = [GPA_S_bio_BC; GPA_S_dis_BC;...
92      GPA_S_mat_BC;GPA_S_btek_BC];
93  GPA_O_BC = [GPA_O_bio_BC; GPA_O_dis_BC;...
94      GPA_O_mat_BC;GPA_O_btek_BC];
95  info_BC=[info_bio_BC;info_dis_BC;...
96      info_mat_BC;info_btek_BC];
97  info_s_BC=[info_s_bio_BC;info_s_dis_BC;...
98      info_s_mat_BC;info_s_btek_BC];
99  %%
100 % Divides BC in to maint training and test sets (9:1)
101 % Randomize index for data division
102 index = randperm(size(info_BC,1));
103 % Counts 10proc. for test set.
104 BC_text_num = round(size(index,2)*0.0);
105 %Divides to general training and test set
106 % Test set
107 stat_num_BC_test= stat_num_BC(index(1:BC_text_num),:);
108 stat_txt_BC_test =stat_txt_BC(index(1:BC_text_num),:);
109 ECTS_T_BC_test = ECTS_T_BC(index(1:BC_text_num),:);
110 ECTS_P_BC_test = ECTS_P_BC(index(1:BC_text_num),:);
111 ECTS_A_BC_test = ECTS_A_BC(index(1:BC_text_num),:);
112 ECTS_L_BC_test = ECTS_L_BC(index(1:BC_text_num),:);
113 ECTS_R_BC_test = ECTS_R_BC(index(1:BC_text_num),:);
```

```
GPA_S_BC_test = GPA_S_BC(index(1:BC_text_num),:);
GPA_O_BC_test = GPA_O_BC(index(1:BC_text_num),:);
info_BC_test = info_BC(index(1:BC_text_num),:);
info_s_BC_test = info_s_BC(index(1:BC_text_num),:);
% Train set
stat_num_BC_train= stat_num_BC(index(BC_text_num+1:  end),:);
stat_txt_BC_train =stat_txt_BC(index(BC_text_num+1:  end),:);
ECTS_T_BC_train = ECTS_T_BC(index(BC_text_num+1:  end),:);
ECTS_P_BC_train = ECTS_P_BC(index(BC_text_num+1:  end),:);
ECTS_A_BC_train = ECTS_A_BC(index(BC_text_num+1:  end),:);
ECTS_L_BC_train = ECTS_L_BC(index(BC_text_num+1:  end),:);
```

D.2 File: `Main_LR.m`

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LR individual  semester model training and testing
% Final CART model determination and testing.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
semester2 =0; semester =20;
% Separates data set in to training and testing sets
[train_BC,records1, test_BC, records2, name] =...
    DataDivision(semester,semester2, sem_BC,...
    stat_txt_BC_train, GPA_O_BC_train, GPA_S_BC_train,...
    ECTS_A_BC_train,ECTS_P_BC_train,...
    ECTS_T_BC_train,ECTS_R_BC_train, ECTS_L_BC_train,...
    info_BC_train);
% Recoding students status in to numerical values
% 1— drop out; 0 — pass
train_coding = zeros(size(train_BC));
train_coding(strcmp(train_BC,'drop out')) = 1;
test_coding = zeros(size(test_BC));
test_coding(strcmp(test_BC,'drop out')) = 1;
% Seperates data set in to 10 folds
Indices = crossvalind('Kfold', size(train_BC,1), 10);
B = [];
% Estimates  B for every fold
for i = 1:10
    cros_train = train_coding((Indices ≠i));
    cros_rec = records1((Indices ≠i),:);
    B(:,i) = glmfit(cros_rec,...
        [cros_train ones(size(cros_train,1),1)],...
        'binomial', 'link', 'logit');
end
% Mean B
```

```
B_Mean = (round(mean(B,2)*1000))/1000;
[misclas(1,3), misclas(1,4) , misclas(1,2) , misclas(1,1),...
    notclass1] =Prediction_num(1,B_Mean, records1,...
    train_coding);
[misclas(2,3), misclas(2,4) , misclas(2,2) , misclas(2,1),...
    notclass2]=Prediction_num(1,B_Mean, records2,...
    test_coding);
Table(misclas);
%% Final model determination
S = [0 1 1 2 3 4 5 6; 20 0 1 2 3 4 5 6];
Model = [1 2 3 4 5 6 7 8];
file =  'Models_new/lr_%i.mat';
type = 3;
predict = FinalPrediction(S, Model, file, type,...
    info_BC_train, GPA_O_BC_train, GPA_S_BC_train,...
    ECTS_P_BC_train, ECTS_T_BC_train,...
    ECTS_A_BC_train,ECTS_R_BC_train,ECTS_L_BC_train);
Model = {'1';'2';'3';'4';'5';'6';'7';'8'};
[drop,false,sem_advance]=FinalEval(S(1,:),...
    stat_num_BC_train, predict, sem_BC,Model)
%% Final model testing
S = [0 2 3; 20 2 3];
Model = [1 4 5];
file =  'Models_new/lr_%i.mat';
type = 3;
```

D.3 File: Main_PCA.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PCA—LR individual  semester model training and testing.
% Final PCA—LR model determination and testing.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
semester2 =0; semester = 20;
[train_BC,records1, test_BC, records2, name] =...
    DataDivision(semester,semester2, sem_BC,...
    stat_txt_BC_train, GPA_O_BC_train, GPA_S_BC_train,...
    ECTS_A_BC_train,ECTS_P_BC_train, ECTS_T_BC_train,...
    ECTS_R_BC_train, ECTS_L_BC_train,info_s_BC_train);
% Remove student with missing data and get status
% numerical values
index=any(isnan(records1),2) ;
records1(index == 1,:) = [];
train_coding = double(strcmp(train_BC, 'drop out'));
test_coding = double(strcmp(test_BC, 'drop out'));
```

```
train_coding(index == 1,:) = [];
% Standatize data
AMean = mean(records1);
AStd = std(records1);
[n,m] = size(records1);
records1 = (records1 - repmat(AMean,[n 1]))...
    ./ repmat(AStd,[n 1]);
% Performs cros validation for PCA
k_fold = crossvalind('Kfold', size(train_coding,1), 10);
PCALoadings = cell(1,10);
PCAScores   = cell(1,10);
PCAVar      = cell(1,10);
% Ploting variance
figure(); hold on;
for i = 1:10
 [PCALoadings{i},PCAScores{i},PCAVar{i}]=...
     princomp(records1(k_fold ~= i,:));
 plot(PCAVar{i}, '-o');
end
xlabel('PC'); ylabel('Explained variance'); hold off;
figure(); hold on;
for i = 1:10
    plot(100*cumsum(PCAVar{i})/sum(PCAVar{i}), '-o');
end
xlabel('PC'); ylabel('Accumulated explained variance');
hold off;
% Number of PCA for model
number_PCA = 8;
% PCA for modeling
[PCALoadings,PCAScores,PCAVar] = princomp(records1);
Indices = crossvalind('Kfold', size(train_coding,1), 10);
B = zeros(number_PCA,10);
% Estimates  B for every fold
for i = 1:10
    cros_train = train_coding(Indices ~=i);
    cros_rec = PCAScores((Indices ~=i),1:number_PCA);
    B(:,i) = glmfit(cros_rec,...
        [cros_train ones(size(cros_train,1),1)],...
        'binomial', 'link', 'logit','constant','off');
end
% Average of betas
avrB=mean(B,2);
% Prediction
[misclas(1,3), misclas(1,4) , misclas(1,2) , misclas(1,1),...
    notclass1] =Prediction_logistic(3, avrB,PCAScores...
    (:,1:number_PCA), train_coding);
Table(misclas)
```

D.4 File: Main_CART.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  CART individual  semester model training and testing.
%  Final CART model determination and testing.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Individual semester model training
semester2 =0; semester = 20;
%Divides data in to training and testing set that
% approximately where would be the same data structure
[train_BC,records1, test_BC, records2, name] =...
    DataDivision(semester, semester2, sem_BC,...
    stat_txt_BC_train, GPA_O_BC_train, GPA_S_BC_train,...
    ECTS_A_BC_train,ECTS_P_BC_train, ECTS_T_BC_train,...
    ECTS_R_BC_train, ECTS_L_BC_train,info_s_BC_train);
% Build the CART tree
t=classregtree(records1,train_BC, 'name', name,...
    'method', 'classification','minparent', 2,...
    'splitcriterion','twoing');
view(t);
% Search for the best pruning level
[b,l] = Cost_Plot(t, records1, train_BC)
% Prunes the tree
t_prune= prune(t, 'level',l);
view(t_prune);
% Model performance table
[misclas(1,3), misclas(1,4) , misclas(1,2) , misclas(1,1)]...
    = Prediction_txt(1,t_prune, records1, train_BC);
[misclas(2,3), misclas(2,4) , misclas(2,2) , misclas(2,1)]...
   = Prediction_txt(1,t_prune, records2, test_BC);
Table(misclas);
%% Final model determination
S = [0 2 3 4 6; 20 2 3 4 6];
Model = [1 4 5 6 8];
file =  'Models_new/cart_%i.mat';
type = 1;
predict = FinalPrediction(S, Model, file, type,...
    info_s_BC_train, GPA_O_BC_train, GPA_S_BC_train,...
    ECTS_P_BC_train, ECTS_T_BC_train,...
    ECTS_A_BC_train,ECTS_R_BC_train,ECTS_L_BC_train);
Model = {'1';'4';'5';'6';'8'};
[drop,false,sem_advance]=FinalEval(S(1,:),...
    stat_num_BC_train, predic, sem_BC,Model)
%% Final model
S = [0 2 3; 20 2 3];
```

```
Model = [1 4 5];
file =  'Models_new/cart_%i.mat';
type = 1;
predict = FinalPrediction(S, Model, file, type,...
    info_s_BC_test, GPA_O_BC_test, GPA_S_BC_test,....
    ECTS_P_BC_test, ECTS_T_BC_test,...
```

D.5 File: Main_Bagging.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CART Bagging individual  semester model training and
% testing. Final CART bagging model determination
% and testing.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
semester2 = 20; semester = 0;
% Separates data set in to training and testing sets
[train_BC,records1, test_BC, records2, name,stud_id]...
    = DataDivision(semester,semester2, sem_BC,...
    stat_txt_BC_train, GPA_O_BC_train,GPA_S_BC_train,...
    ECTS_A_BC_train,ECTS_P_BC_train,ECTS_T_BC_train,...
    ECTS_R_BC_train, ECTS_L_BC_train,info_BC_train);
err = zeros(500,5);
% Model building
for i = 1 :5
    fprintf('Step %f',   i);
    B = TreeBagger(500,records1,train_BC,'OOBPred',...
        'on','Method','classification','OOBVarImp', 'on');
    err(:,i) = oobError(B);
end
figure('Name', 'Mean'); plot(mean(err,2));
xlabel('Number of grown trees')
ylabel('Mena of out-of-bag classification error')
figure('Name','STD'); plot(std(err,0,2));
xlabel('Number of grown trees')
ylabel...
    ('Standart deviation of out-of-bag classification error')
% Define minimal number of trees for the semester model
min_tree = 300;
B1 = TreeBagger(min_tree,records1,train_BC,'OOBPred',...
    'on','Method','classification');
[misclas(1,3), misclas(1,4) , misclas(1,2) , misclas(1,1)]...
    = Prediction(2,B1, records1, train_BC);
[misclas(2,3), misclas(2,4) , misclas(2,2) , misclas(2,1)]...
    = Prediction(2,B1, records2, test_BC)
```

```
Table(misclas);
%% Final model determination
S = [0 1 1 2 3 4 5 6; 20 0 1 2 3 4 5 6];
Model = [1 2 3 4 5 6 7 8];
file =  'Models_new/bagging_%i.mat';
type = 2;
predict = FinalPrediction(S, Model, file, type,...
    info_BC_train, GPA_O_BC_train, GPA_S_BC_train,...
    ECTS_P_BC_train, ECTS_T_BC_train, ECTS_A_BC_train,...
    ECTS_R_BC_train,ECTS_L_BC_train);
Model = {'1';'2';'3';'4';'5';'6';'7';'8'};
[drop,false,sem_advance]=FinalEval(S(1,:),...
    stat_num_BC_train, predict,sem_BC,Model)
%% Final model testing
S = [0 2; 20 2];
Model = [1 4];
file =  'Models_new/bagging_%i.mat';
type = 2;
predict = FinalPrediction(S, Model, file, type,...
    info_BC_test, GPA_O_BC_test, GPA_S_BC_test,...
    ECTS_P_BC_test, ECTS_T_BC_test, ECTS_A_BC_test,...
    ECTS_R_BC_test,ECTS_L_BC_test);
Model = {'1';'4'};
[drop,false,sem_advance]=FinalEval(S(1,:),...
    stat_num_BC_test, predict,sem_BC_test,Model)
```

D.6 File: Main_RF.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Random forest individual  semester model training and
% testing. Final RF model determination and testing.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
semester2 =1; semester =0;
% Separates data set in to training and testing sets
[train_BC,records1, test_BC, records2, name] = ....
    DataDivision(semester, semester2, sem_BC,...
    stat_txt_BC_train, GPA_O_BC_train, GPA_S_BC_train,...
    ECTS_A_BC_train,ECTS_P_BC_train, ECTS_T_BC_train,....
    ECTS_R_BC_train, ECTS_L_BC_train,info_BC_train);
train_coding = ones(size(train_BC));
train_coding(strcmp(train_BC,'drop out')) = -1;
train = ones(size(train_BC));
train(strcmp(train_BC,'drop out')) = -1;
test_coding = ones(size(test_BC));
```

```
test_coding(strcmp(test_BC,'drop out')) = 1;
% Removes students with missig values
records= records1;
index=any(isnan(records),2) ;
records(index == 1,:) = [];
train_coding(index == 1,:) = [];
% Setting extra_optioons for modeling
clear extra_options
extra_options.importance = 1;
extra_options.proximity  = 1;
%% Most important variables detection, using average
% of 10 models
test = 10;
mean_decrease = zeros(size(records,2),test);
gini_decrease = zeros(size(records,2),test);
for i = 1:test
   model = classRF_train(records,...
       train_coding,0,0,extra_options);
   mean_decrease(:,i) = model.importance(:,end-1);
   gini_decrease(:,i) = model.importance(:,end);
end
mean_decrease_avr = mean(mean_decrease,2);
gini_decrease_avr = mean(mean_decrease,2);
figure('Name','Mean decrease in Accuracy');
bar(model.importance(:,end-1));
xlabel('feature');ylabel('magnitude');
figure('Name','Mean decrease in Gini index');
bar(model.importance(:,end));
xlabel('feature');ylabel('magnitude');
%% Searching for the good MTRY using 5 fold cross validation
% Determining important variables
important_varb = [1 4 5 8 9 14 18 19 20 21 23];
records_red1 = records(:,important_varb);
records_red2 = records2(:,important_varb);
% Determining parameters for the test
kfold = 5 ;
Indices = crossvalind('Kfold', size(train_coding,1), kfold);
MTRY = size(records_red1,2);
dd = zeros(kfold,1); pd = zeros(kfold,1); dp = zeros(kfold,1);
pp = zeros(kfold,1); dd_avr = zeros(MTRY,1);
pd_avr = zeros(MTRY,1); dd_std = zeros(MTRY,1);
pd_std = zeros(MTRY,1);
for i = 1:MTRY
       for k = 1:kfold
            model = classRF_train(records_red1...
                (Indices ~=k,:), train_coding(Indices...
                ~=k,:),500,i,extra_options);
```

```
64              [pp, pd(k,1), dp, dd(k,1)] = Prediction_RF...
65                  (model, records_red1(Indices ==k,:),....
66                  train_coding(Indices == k,:));
67          end
68          dd_avr(floor(i/5)+1,1) = mean(dd);
69          pd_avr(floor(i/5)+1,1) = mean(pd);
70          dd_std(floor(i/5)+1,1) = std(dd);
71          pd_std(floor(i/5)+1,1) = std(pd);
72  end
73  figure('Name','Mean'); plot(1:MTRY,dd_avr, '-b');
74  hold on; plot(1:MTRY,pd_avr,'-r'); hold off;
75  xlabel('MTRY'); ylabel('Mean of classifications');
76  legend('DD', 'PD')
77  figure('Name', 'STD'); plot(1:MTRY,dd_std, '-b');
78  hold on; plot(1:MTRY,pd_std,'-r'); hold off;
79  xlabel('MTRY'); ylabel('Standard deviation Classifications');
80  legend('DD', 'PD')
81  %%  Searching for the good NTREE using 5 fold cross validation
82  MTRY = 4;
83  dd = zeros(kfold,1); pd = zeros(kfold,1); dp = zeros(kfold,1);
84  pp = zeros(kfold,1); dd_avr = zeros(50,1);
85  pd_avr = zeros(50,1); dd_std = zeros(50,1);
86  pd_std = zeros(50,1);
87  NTREE = 500;
88  for i = 1:10:NTREE
89         for k = 1:kfold
90              model = classRF_train(records_red1(Indices...
91                  ≠k,:),train_coding(Indices ≠k,:)....
92                  ,i,MTRY,extra_options);
93              [pp, pd(k,1), dp, dd(k,1)] = Prediction_RF...
94                  (model, records_red1(Indices ==k,:),...
95                  train_coding(Indices == k,:));
96          end
97          dd_avr(floor(i/10)+1,1) = mean(dd);
98          pd_avr(floor(i/10)+1,1) = mean(pd);
99          dd_std(floor(i/10)+1,1) = std(dd);
100         pd_std(floor(i/10)+1,1) = std(pd);
101 end
102 figure('Name','Mean'); plot(1:50,dd_avr, '-b');
103 hold on; plot(1:50,pd_avr,'-r'); hold off;
104 xlabel('NTREE'); ylabel('Mean of classifications');
105 legend('DD', 'PD')
106 set(gca,'XTickLabel',0:50:500,'XTick',0:5:50);
107 figure('Name', 'STD'); plot(1:50,dd_std, '-b');
108 hold on; plot(1:50,pd_std,'-r'); hold off;
109 xlabel('NTREE');
110 ylabel('Standard deviation Classifications');
```

```
legend('DD', 'PD');
set(gca,'XTickLabel',0:50:500,'XTick',0:5:100);
%% Building final semester model
NTREE = 200;
model = classRF_train(records_red1,...
    train_coding,NTREE,MTRY,extra_options);
[misclas(1,3), misclas(1,4) , misclas(1,2) , misclas(1,1)]...
    = Prediction_RF(model, records_red1, train_coding);
[misclas(2,3), misclas(2,4) , misclas(2,2) , misclas(2,1)]...
    = Prediction_RF(model, records_red2, test_coding);
Table(misclas);
%% Determing final model
S = [0 1 1 2 3 4 5 6; 20 0 1 2 3 4 5 6 ];
Model = [1 2 3 4 5 6 7 8];
file =  'Models_new/RF_%i_.mat';
type = 5;
predict = FinalPrediction(S, Model, file, type,...
    info_BC_train, GPA_O_BC_train, GPA_S_BC_train,...
    ECTS_P_BC_train, ECTS_T_BC_train,...
    ECTS_A_BC_train,ECTS_R_BC_train,ECTS_L_BC_train);
predict(predict == 1) = 0; predict(predict == -1) = 1;
Model = {'1';'2';'3';'4';'5';'6';'7';'8'};
[drop,false,sem_advance]=FinalEval(S(1,:),...
    stat_num_BC_train, predict, sem_BC,Model)
%% Final model testing
S = [0 2; 20 2];
Model = [1 4];
file =  'Models_new/RF_%i_.mat';
type = 5;
predict = FinalPrediction(S, Model, file, type,...
    info_BC_test, GPA_O_BC_test, GPA_S_BC_test,...
    ECTS_P_BC_test, ECTS_T_BC_test, ECTS_A_BC_test,...
    ECTS_R_BC_test,ECTS_L_BC_test);
predict(predict == 1) = 0; predict(predict == -1) = 1;
Model = {'1';'4'};
[drop,false,sem_advance]=FinalEval(S(1,:),...
    stat_num_BC_test, predict, sem_BC_test,Model)
```

D.7 File: Main_MARS.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MARS individual  semester model training and testing.
% Final MARS model determination and testing.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Searching for best  M, mi and d variables combination...
% that would give the smalest MSE
% Variable semeter
A = [0 1 1 2 3 4 5 6; 20 0 1 2 3 4 5 6];
% Grid for M, mi and d
M_vec = 21:5:151; mi_vec = 1:3; d_vec = 1:3;
N = length(M_vec) * 3 * 3;
[MIM,DM,MM] = meshgrid(mi_vec,d_vec,M_vec);
MARSpar = [MM(:),MIM(:),DM(:)];
% Get some MATLAB workers
matlabpool open SGE 15;
k = 1;
% Result matrix
res = zeros(N*8,9);
fprintf(1, 'Beginning MARS calculations\n\n');
% Standardization values matrix
mean_est = cell(1,8); std_est = cell(1,8);
starttime=tic;
try
    for i = 1:8
        semester2 = A(1,i); semester = A(2,i);
        % Separates data set in to training and testing sets
        [train_BC,records1, dummy1, dummy2, dummy3] =...
            DataDivision( semester, semester2, sem_BC,...
            stat_txt_BC_train, GPA_O_BC_train,...
            GPA_S_BC_train, ECTS_A_BC_train,...
             ECTS_P_BC_train, ECTS_T_BC_train,...
             ECTS_R_BC_train, ECTS_L_BC_train,info_BC_train);
        mean_est{i} = nanmean(records1);
        std_est{i} = nanstd(records1);
        cent = bsxfun(@minus,records1, mean_est{i});
        records1 = bsxfun(@rdivide,cent,std_est{i});
        train_coding=strcmp(train_BC,'drop out');
        parfor k = (1+(i-1)*N):(N+(i-1)*N)
            tmpi = k - (i-1)*N;
            % This is a var running between 1 and N
            M = MM(tmpi);
            mi = MIM(tmpi);
            d = DM(tmpi);
            fprintf(1,'i=%i\tk=%i\tM=%i\tmi=%i\td=%i\n',...
                i, k, M, mi, d);
            params = aresparams(M, 5,  false, [], mi, d);
            [taMSE,taRMSE,taRRMS,taR2] =...
                arescv(records1, train_coding,...
                params, [], 5,[],[],[],0);
            res(k,:) = [i k M mi d taMSE taRMSE taRRMS taR2];
        end
```

```
52      end
53  catch err
54      disp(err);
55  end
56  toc(starttime);
57  save(sprintf('mars_res_%s.mat',datestr(now(),...
58      'yyyymmddHHMM')), mean_est,std_est,res);
59  matlabpool close;
60  %% Individual semester model building
61  semester2 =0; semester = 20;
62  % Separates data set in to training and testing sets
63  [train_BC,records1, test_BC, records2, name] =...
64      DataDivision(semester, semester2, sem_BC,...
65      stat_txt_BC_train, GPA_O_BC_train, GPA_S_BC_train,...
66      ECTS_A_BC_train,ECTS_P_BC_train, ECTS_T_BC_train,...
67      ECTS_R_BC_train, ECTS_L_BC_train,info_BC_train);
68  % Standardization
69  mean_estm = nanmean(records1);
70  std_estm = nanstd(records1);
71  Cent = bsxfun(@minus,records1,mean_estm);
72  records1 = bsxfun(@rdivide,Cent,std_estm);
73  Cent = bsxfun(@minus,records2,mean_estm);
74  records2 = bsxfun(@rdivide,Cent,std_estm);
75  train_coding = zeros(size(train_BC));
76  train_coding(strcmp(train_BC,'drop out')) = 1;
77  test_coding = zeros(size(test_BC));
78  test_coding(strcmp(test_BC,'drop out')) = 1;
79  % Setting parameters
80  M = 10; mi = 1; d = 1;
81  params = aresparams(M, 5,  false, [], mi, d);
82  % Model building
83  model = aresbuild(records1, train_coding, params)
84  ModelF = struct('model', model,...
85      'mean_estm',mean_estm,'std_estm',std_estm);
86  % Model output in mathematical form
87  B = areseq(model, 5)
88  [misclas(1,3), misclas(1,4) , misclas(1,2) , misclas(1,1)]...
89      = Prediction_num(2,model, records1, train_coding);
90  [misclas(2,3), misclas(2,4) , misclas(2,2) , misclas(2,1)]...
91      = Prediction_num(2,model, records2, test_coding);
92  Table(misclas);
93  %% Final model determination
94  S = [0 1 1 2 3 4 5 6; 20 0 1 2 3 4 5 6];
95  Model = [1 2 3 4 5 6 7 8];
96  file =  'Models_new/mars_%i.mat';
97  type = 4;
98  predict = FinalPrediction(S, Model, file, type,...
```

```
    info_BC_train, GPA_O_BC_train, GPA_S_BC_train,...
    ECTS_P_BC_train, ECTS_T_BC_train,ECTS_A_BC_train,...
    ECTS_R_BC_train,ECTS_L_BC_train);
Model = {'1';'2';'3';'4';'5';'6';'7';'8'};
[drop,false,sem_advance]=FinalEval(S(1,:),...
    stat_num_BC_train, predict, sem_BC,Model)
%% Final model testing
S = [0 2; 20 2];
Model = [1 4];
file =  'Models_new/mars_%i.mat';
type = 4;
predict = FinalPrediction(S, Model, file, type,...
    info_BC_test, GPA_O_BC_test, GPA_S_BC_test,...
    ECTS_P_BC_test, ECTS_T_BC_test, ECTS_A_BC_test,...
    ECTS_R_BC_test,ECTS_L_BC_test);
Model = {'1';'4'};
[drop,false,sem_advance]=FinalEval(S(1,:),....
    stat_num_BC_test, predict,sem_BC_test,Model)
```

D.8 Function: Round_ECTS.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Raunding ECTS credits measurments that value of module
% after division of five would be 0.
% Adress: Main
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [rounded] = Round_ECTS(A)
for i = 1:size(A,1)
  for j = 2:size(A,2)
     if mod(A(i,j),1) < 2.5
         rounded(i,j) = A(i,j) - mod(A(i,j),5);
     else
         rounded(i,j) = A(i,j) + 5 - mod(A(i,j),5);
     end
  end
end
rounded(:,1) = A(:,1);
```

D.9 Function: SortExamps.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Updates num_[program] matrix. Output: 1 col: student ID, 2
% col: number of semester, 3 col: program level,
% 4 col: mark, 5 col: ECTS.
% Adress: Main
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [num] = SortExamps(raw2,num2,num1, type)
[num_code,txt_code,raw_code] =...
    xlsread('semester_coding.xlsx',1,'A1:B30');
num(:,1) = num2(:,1);
% Converting semester name in to digits
% Coding is in excel file: semester_coding.xlsx
for i = 1:size(raw_code,1)
    [r,c] = find(strcmp(raw2,txt_code(i)));
    if ¬isempty(r)
        num(r,2) = num_code(i,1);
        r = [];
    end
end
% Chnging Marks from string to numbers
% -20 - not attendet exam; 20 - passed exam;
% 1 - BC program;2 - MC program
for i = 1:size(raw2,1)
    if strcmp(raw2(i,2), 'CBAC04')
        num(i,3) = 1;
    elseif strcmp(raw2(i,2), 'CKAN06DK') ||...
            strcmp(raw2(i,2), 'CKAN10DK')
        num(i,3) = 2;
    end
    if strcmp(raw2(i,2), 'CBAC04') ||...
            strcmp(raw2(i,2), 'CKAN06DK')...
            || strcmp(raw2(i,2), 'CKAN10DK')
        if strcmp(raw2(i,6),'EM')
            num(i,4) = -20;
        elseif strcmp(raw2(i,6),'BE')
            num(i,4) = 20;
        else
            num(i,4) = str2double(raw2(i,6));
        end
        if type ==1
            num(i,5) = num2(i,8);
        else
            num(i,5) = str2double(raw2(i,8));
        end
    end
end
% Sorting data according semesters
for i=1:size(num1,1)
```

```
49      begin = find(num ==num1(i,1),1);
50      all = size(find(num ==num1(i,1)),1);
51      extract = num(begin:all+begin-1,:);
52      num(begin:begin+all-1,:) = sortrows(extract,[3 2]);
53      extract=[];
54  end
```

D.10 Function: Status.m

```
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Extracts student status. In stat_num first col coded
3  % status (1 - drop out, 0 - pass) and second student ID
4  % In stat_txt status in text
5  % Adress: Main
6  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7  function [stat_num, stat_txt] = Status(raw1,num1,num2, prog)
8  % prog ==1 corresponds to BSc
9  if prog == 1
10     txt1 = 'CBAC04';
11     txt2 = 'CBAC04';
12 else
13     txt1 = 'CKAN06DK';
14     txt2 = 'CKAN10DK';
15 end
16 stat_num = []; stat_txt = {};
17 for i = 1:size(raw1,1)
18   if  strcmp(raw1(i,19),txt1) || strcmp(raw1(i,19),txt2)
19       if strcmp(raw1(i,20),'afbrudt')
20           stat_num( end+1,1) = 1;
21           stat_num( end,2) = num1(i,1);
22           stat_txt( end+1,1) = {'drop out'};
23       elseif strcmp(raw1(i,20),'afsluttet')
24           stat_num( end+1,1) = 0;
25           stat_num( end,2) = num1(i,1);
26           stat_txt( end+1,1) = {'pass'};
27       end
28   end
29 end
```

D.11 Function: PersonalInfo_short.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Makes personal information matrix with out dummy
% variables for categorical variables
% Adress: Main
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function info = PersonalInfo_short(raw1,num1,stat)
info = zeros(size(stat,1),13);
for i = 1:size(stat,1)
    stud = find(num1(:,1) == stat(i,2),1);
    % Student ID
    info(i,1) = num1(stud,1);
    % Age
    info(i,2) = num1(stud,2);
    % Lockal student
    % 1- international students, 2 - lockal students
    if strcmp(raw1(stud,8),'ANDET')
        info(i,3)= 1;
    else
        info(i,3)= 2;
    end
    % Progrma
    % 1- Teknisk biomedicin, 2 - Design og innovation,
    % 3 - Matematik og teknologi
    if strcmp(raw1(stud,1),'Teknisk biomedicin')
        info(i,4)= 1;
    elseif strcmp(raw1(stud,1),'Design og innovation')
        info(i,4)= 2;
    elseif strcmp(raw1(stud,1),'Matematik og teknologi')
        info(i,4)= 3;
    else
        info(i,4)= 4;
    end
    % Time pased after shool exam
    star_date = char(raw1(stud,16));
    star_date = star_date(1:4);
    star_date = str2num(star_date);
    end_date = num1(stud,6);
    info(i,5) = star_date - end_date;
    % GPA school
    info(i,6) = str2num(char(raw1(stud,9)));
    % Math level
    % 1- level A ; 2 - level B, 3 - level C
    if strcmp(raw1(stud,14),'A')
        info(i,7)= 1;
    elseif strcmp(raw1(stud,14),'B')
        info(i,7)= 2;
    else
```

```
48          info(i,7)= 3;
49      end
50      % Physics level
51      % 1— level A ; 2 — level B, 3 — level C
52      if strcmp(raw1(stud,10),'A')
53          info(i,8)= 1;
54      elseif strcmp(raw1(stud,10),'B')
55          info(i,8)= 2;
56      else
57          info(i,8)= 3;
58      end
59      % Chemisrty level
60      % 1— level A ; 2 — level B, 3 — level C
61      if strcmp(raw1(stud,12),'A')
62          info(i,9)= 1;
63      elseif strcmp(raw1(stud,12),'B')
64          info(i,9)= 2;
65      else
66          info(i,9)= 3;
67      end
68      % Math grade
69      info(i,10) = num1(stud,14);
70      % Physics grade
71      info(i,11) = num1(stud,10);
72      % Chemistry grade
73      info(i,12) = num1(stud,12);
74      % Gender
75      % 1— men, 2— woman
76      if strcmp(raw1(i,4),'M')
77          info(i,13) = 1;
78      else
79          info(i,13) = 2;
80      end
81  end
```

D.12 Function: PersonalInfo.m

```
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Makes personal information matrix with dummy variables for
3  % categorical variables
4  % Adress: Main
5  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6  function info = PersonalInfo(raw1,num1,stat)
7  info = zeros(size(stat,1),24);
```

```
for i =1:size(stat,1)
    stud = find(num1(:,1) == stat(i,2),1);
    % Student ID
    info(i,1) = num1(stud,1);
    % Age
    info(i,2) = num1(stud,2);
     % Lockal student
  if ¬strcmp(raw1(stud,8),'ANDET')
    info(i,3)= 1;
  end
    % International student
    if strcmp(raw1(stud,8),'ANDET')
        info(i,4)= 1;
    end
    % Teknisk biomedicin progrma
    if strcmp(raw1(stud,1),'Teknisk biomedicin')
        info(i,5)= 1;
    end
    % Design og innovation program
    if strcmp(raw1(stud,1),'Design og innovation')
        info(i,6)= 1;
    end
    % Matematik og teknologi program
    if strcmp(raw1(stud,1),'Matematik og teknologi')
        info(i,7)= 1;
    end
    % Bioteknologi program
    if strcmp(raw1(stud,1),'Bioteknologi')
        info(i,8)= 1;
    end
    % Time pased after shool exam
    star_date = char(raw1(stud,16));
    star_date = star_date(1:4);
    star_date = str2num(star_date);
    end_date = num1(stud,6);
    info(i,9) = star_date - end_date;
    % GPA school
    info(i,10) = str2num(char(raw1(stud,9)));
    % Math level A
    if strcmp(raw1(stud,14),'A')
        info(i,11)= 1;
    end
    % Math level B
    if strcmp(raw1(stud,14),'B')
        info(i,12)= 1;
    end
    % Math level C
```

```
55      if strcmp(raw1(stud,14),'C')
56          info(i,13)= 1;
57      end
58      % Physics level A
59      if strcmp(raw1(stud,10),'A')
60          info(i,14)= 1;
61      end
62      % Physics level B
63      if strcmp(raw1(stud,10),'B')
64          info(i,15)= 1;
65      end
66      % Physics level C
67      if strcmp(raw1(stud,10),'C')
68          info(i,16)= 1;
69      end
70      % Chemisrty level A
71      if strcmp(raw1(stud,12),'A')
72          info(i,17)= 1;
73      end
74      % Chemisrty level B
75      if strcmp(raw1(stud,12),'B')
76          info(i,18)= 1;
77      end
78      % Chemisrty level C
79      if strcmp(raw1(stud,12),'C')
80          info(i,19)= 1;
81      end
82      % Math grade
83      info(i,20) = num1(stud,14);
84      % Physics grade
85      info(i,21) = num1(stud,10);
86      % Chemistry grade
87      info(i,22) = num1(stud,12);
88      % Gender
89      % If man
90      if strcmp(raw1(i,4),'M')
91          info(i,23) = 1;
92      else
93          % If woman
94          info(i,24)=1;
95      end
96  end
```

D.13 Function: PerformanceInformations.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Extracts performance information according program level
% Adress: Main
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function num = PerformanceInformation(num2, stat_num, prog)
num=zeros(1,size(num2,2));
num_length = 1;
for i = 1:size(stat_num,1)
    begin = find(num2(:,1) == stat_num(i,2) &...
    num2(:,3)== prog ,1);
    all = size(find(num2(:,1) == stat_num(i,2) &...
        num2(:,3)== prog),1);
    num(num_length:num_length+all-1,:) =...
        num2(begin:begin+all-1,:);
    num_length = num_length+all;
end
```

D.14 Function: `Table.m`

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Performance table
% Adress: Main_CART, Main_LR, Main_Bagging, Main_PCA, Main_RF,
% Main_MARS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Table(misclas)
% Total misclasification rate: all misclas / all observ.
ratio = sum( misclas(:,2) + misclas(:,4))/sum(sum(misclas));
% Drop out misclass in all observ:
% drop out misclas/ all observ
drop_ratio = sum( misclas(:,2))/sum(sum(misclas));
% Drop out misclas in all misclas:
% drop out misclas/ all misclas
total_drop = sum( misclas(:,2))/...
    sum( misclas(:,2) + misclas(:,4));
cnames = {'DD ','DP','PP','PD'};
rnames = {'Train','Test'};
f = figure('Position',[500 500 700 200]);
tt = uitable('Parent',f,'Data',misclas,'ColumnName',cnames,...
    'RowName',rnames,'Position',[50 110 370 59]);
cnames = {'Misclass. ratio ','Drop out misclass. ratio',...
    'Drop out ratio in all misclass.'};
rnames = {''};
dat = [ratio drop_ratio total_drop];
t1 = uitable('Parent',f,'Data',dat,'ColumnName',cnames,...
```

```
26      'RowName',rnames,'Position',[50 50 552 60]);
```

D.15 Function: DataDivision.m

```
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Divides data into training and test set for semester model
3  % building.
4  % Adress: Main_CART, Main_LR, Main_Bagging, Main_PCA, Main_RF,
5  % Main_MARS
6  % Notes: If semester == 20, then it is application
7  % evaluation and it uses just information from the
8  % application and all students.
9  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10 function [train,records1, test, records2, name] =...
11     DataDivision(semester, semester2, sem,B, GPA_O,...
12     GPA_S, ECTS_A,ECTS_P, ECTS_T, ECTS_R,ECTS_L, info )
13 % B - the vector of true classes
14 % Devides in two sets, that in each semester would be
15 % 2:8 cases of original set.
16 [train_index, train, test_index, test] = SSS(B,sem);
17 % Initialize data sets
18 train = {}; train_ECTS_P = [];  train_ECTS_T = [];
19 train_GPA_O =[]; train_GPA_S = []; train_ECTS_A = [] ;
20 train_ECTS_R = []; train_ECTS_L=[]; train_in =[]; test = {};
21 test_ECTS_P = []; test_ECTS_T = [];test_GPA_O = [];
22 test_GPA_S = []; test_ECTS_A = []; test_ECTS_R = [];
23 test_ECTS_L=[]; test_in =[];
24 % Divides records in two data sets.
25  for i = 1:size(train_index,1)
26      if semester ==20
27          train( end+1,1) = B(train_index(i));
28          train_in( end+1,:) = info(train_index(i),2:  end);
29      elseif (sem(train_index(i),1) ≥ semester2 &&...
30              strcmp(B(train_index(i)), 'pass')) ||...
31              (sem(train_index(i),1) == semester2 ||...
32              sem(train_index(i),1) == semester &&...
33              strcmp(B(train_index(i)), 'drop out')) &&...
34              semester ≠ 20
35          train( end+1,1) = B(train_index(i));
36          train_ECTS_P( end+1,:) = ECTS_P(train_index(i),:);
37          train_ECTS_T( end+1,:) = ECTS_T(train_index(i),:);
38          train_GPA_O( end+1,:) = GPA_O(train_index(i),:);
39          train_GPA_S( end+1,:) = GPA_S(train_index(i),:);
40          train_ECTS_A( end+1,:) = ECTS_A(train_index(i),:);
```

```
41          train_ECTS_R( end+1,:) = ECTS_R(train_index(i),:);
42          train_ECTS_L( end+1,:)= ECTS_L(train_index(i),:);
43          train_in( end+1,:) = info(train_index(i),2:  end);
44      end
45  end
46  for i = 1:size(test_index,1)
47      if semester == 20
48          test( end+1,1) = B(test_index(i));
49          test_in( end+1,:) = info(test_index(i),2:  end);
50      elseif (sem(test_index(i),1) ≥ semester2 &&...
51              strcmp(B(test_index(i)),'pass')) ||...
52              (sem(test_index(i),1) == semester2 ||...
53              sem(test_index(i),1) == semester &&...
54              strcmp(B(test_index(i)), 'drop out')) &&...
55              semester ≠ 20
56          test( end+1,1) = B(test_index(i));
57          test_ECTS_P( end+1,:) = ECTS_P(test_index(i),:);
58          test_ECTS_T( end+1,:) = ECTS_T(test_index(i),:);
59          test_GPA_O( end+1,:) = GPA_O(test_index(i),:);
60          test_GPA_S( end+1,:) = GPA_S(test_index(i),:);
61          test_ECTS_A( end+1,:) = ECTS_A(test_index(i),:);
62          test_ECTS_R( end+1,:) = ECTS_R(test_index(i),:);
63          test_ECTS_L( end+1,:)= ECTS_L(test_index(i),:);
64          test_in( end+1,:) = info(test_index(i),2:  end);
65      end
66  end
67  % Makes racords name vector
68  records1 = Records(semester, semester2,train_in,...
69      train_GPA_O, train_GPA_S,train_ECTS_P,train_ECTS_T,...
70      train_ECTS_A, train_ECTS_R, train_ECTS_L);
71  records2 = Records(semester, semester2,test_in,test_GPA_O,....
72      test_GPA_S,test_ECTS_P,test_ECTS_T,test_ECTS_A,...
73      test_ECTS_R, test_ECTS_L);
74  in = {'age' 'L/In' 'Program' 'Time af. exam' ' GPA'...
75      'Math 1' 'Physic 1' 'Chemistry 1' 'Math' 'Physic'...
76      'Chemistry' 'Gender'};
77  if semester == 20
78      name = [in];
79  end
80  if semester ≠20
81  n_GPA_O = {'GPA O 1' ' GPA O 2 ' 'GPA O 3' 'GPA O 4'...
82      'GPA O 5' 'GPA O 6' 'GPA O 7' 'GPA O 8' 'GPA O 9'...
83      'GPA O 10' 'GPA O 11' 'GPA O 12' 'GPA O 13'};
84  n_GPA_S = {'GPA S 1' ' GPA S 2' 'GPA S 3' 'GPA S 4'...
85      'GPA S 5' 'GPA S 6' 'GPA S 7' 'GPA S 8' 'GPA S 9'...
86      'GPA S 10' 'GPA S 11' 'GPA S 12' 'GPA S 13'};
87  n_ECTS_T = {'ECTS T 1' ' ECTS T 2' 'ECTS T 3' 'ECTS T 4'...
```

```
    'ECTS T 5' 'ECTS T 6' 'ECTS T 7' 'ECTS T 8' 'ECTS T 9'...
    'ECTS T 10' 'ECTS T 11' 'ECTS T 12' 'ECTS T 13'};
n_ECTS_P = {'ECTS P 1' ' ECTS P 2' 'ECTS P 3' 'ECTS P 4'...
    'ECTS P 5' 'ECTS P 6' 'ECTS P 7' 'ECTS P 8' 'ECTS P 9'...
    'ECTS P 10' 'ECTS P 11' 'ECTS P 12' 'ECTS P 13'};
n_ECTS_A = {'ECTS Accum 1' ' ECTS Accum 2' 'ECTS Accum 3'...
    'ECTS Accum 4' 'ECTS Accum 5''ECTS Accum 6'...
    'ECTS Accum 7' 'ECTS Accum 8' 'ECTS Accum 9'...
    'ECTS Accum 10' 'ECTS Accum 11'...
    'ECTS Accum 12' 'ECTS Accum 13'};
n_ECTS_R= {'ECTS R 1' ' ECTS R 2' 'ECTS R 3' 'ECTS R 4'...
    'ECTS R 5' 'ECTS R 6' 'ECTS R 7'  'ECTS R 8' 'ECTS R 9'...
    'ECTS R 10' 'ECTS R 11' 'ECTS R 12' 'ECTS R 13'};
n_ECTS_L= {'ECTS L 1' ' ECTS L 2' 'ECTS L 3' 'ECTS L 4'...
    'ECTS L 5' 'ECTS L 6' 'ECTS L 7'  'ECTS L 8' 'ECTS L 9'...
    'ECTS L 10' 'ECTS L 11' 'ECTS L 12' 'ECTS L 13'};
name = [in n_GPA_O(1:semester2-1) n_GPA_S(1:semester2-1)....
    n_ECTS_P(1:semester2-1) n_ECTS_T(1:semester2-1)...
    n_ECTS_A(1:semester2-1) n_ECTS_R(1:semester2-1)...
    n_ECTS_L(1:semester2-1)];
end
```

D.16 Function: Cost_Plot.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Finds the best pruning level for the tree
% Adress: Main_CART
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [b,best] = Cost_Plot(tree, rec, res)
 % c - cost vector; s - vector of standart errors of the
 % cost vector, n- vector of number of teriminal nodees of
 % each subtree best - the best level to prune
 [c,s,n,best] = test(tree,'cross',rec,res);
figure;
[mincost,minloc] = min(c);
plot(n,c,'b-o',...
     n(best+1),c(best+1),'bs',...
     n,(mincost+s(minloc))*ones(size(n)),'k--')
xlabel('Tree size (number of terminal nodes)')
ylabel('Cost')
b = n(best+1);
```

D.17 Function: Prediction_txt.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Count corect and incorect predictions for CART and Bagging
% Adress: Main_CART, Main_Bagging
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [pp, pd, dp, dd] = Prediction_txt(type, model,...
    records, test_BC)
% CART prediction
if type == 1
 yfit = eval(model,records);
end
% TreeBagger prediction
if type ==2
  yfit = predict(model,records);
end
% Count
temp = [test_BC yfit];
pp = sum(strcmp(temp(:,1),temp(:,2)) &...
    strcmp(temp(:,1),'pass') );
dd= sum(strcmp(temp(:,1),temp(:,2)) &...
    strcmp(temp(:,1),'drop out'));
pd= sum(strcmp(temp(:,1),'pass') &...
    strcmp(temp(:,2),'drop out'));
dp = sum(strcmp(temp(:,1),'drop out') &...
    strcmp(temp(:,2),'pass'));
```

D.18 Function: Prediction_num.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Count corect and incorect predictions for Loggistic
% Regression, MARS and CPA.
% Adress: Main_LR, Main_MARS, Main_PCA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [pp, pd, dp, dd, notclass] = Prediction_num(type,...
      model, records, test_BC)
% Logistic regresion prediction
if type ==1
  yfit =glmval(model,records,'logit');
  yfit = round(yfit);
end
% MARS prediction
```

```
if type == 2
   yfit = round(arespredict(model, records));
end
% PCA prediction
if type == 3
    yfit =round(glmval(model,records,...
        'logit','constant','off'));
end
temp = [test_BC yfit];
pp = sum(temp(:,1) == temp(:,2) & temp(:,1) == 0 );
dd= sum(temp(:,1) == temp(:,2) & temp(:,1) == 1 );
pd= sum(temp(:,1)==0 & temp(:,2)==1);
dp = sum(temp(:,1)==1 & temp(:,2)==0);
notclass = sum(isnan(yfit))
```

D.19 Function: FinalPrediction.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Predicts step by step all models
% Adress: Main_LR, Main_CART, Main_Bagging, Main_RF, Main_mars
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% S  - semester list % Model - model list
function predict = FinalPrediction(S, Model, file, type,...
    info_s_BC, GPA_O_BC, GPA_S_BC, ECTS_P_BC, ECTS_T_BC,...
    ECTS_A_BC,ECTS_R_BC,ECTS_L_BC)
if type == 1 || type == 2
    predict = {};
else
    predict = [];
end
for i = 1:size(S,2)
 load(sprintf(file,Model(i)));
 predict(:,i)= StepPrediction(S(2,i), S(1,i),type, model,...
     info_s_BC, GPA_O_BC, GPA_S_BC, ECTS_P_BC,ECTS_T_BC,...
     ECTS_A_BC, ECTS_R_BC,ECTS_L_BC);
end
```

D.20 Function: StepPrediction.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Predicts one semester for final model evaluation.
```

```
% Adress: FinalPrediction
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function predic = StepPrediction(semester, semester2,type,...
    model, in,GPA_O, GPA_S, ECTS_P, ECTS_T, ECTS_A,...
    ECTS_R,ECTS_L)
records = Records(semester, semester2,in(:,2:  end),...
    GPA_O,GPA_S,ECTS_P,ECTS_T,ECTS_A,ECTS_R,ECTS_L);
% CART
if type == 1
 predic  = eval(model,records);
% Bagging
elseif type == 2
    predic = predict(model,records);
% LR
elseif type == 3
  yfit =glmval(model,records,'logit');
  predic = round(yfit);
% Mars
elseif type ==4
  Cent = bsxfun(@minus,records,model.mean_estm);
  records = bsxfun(@rdivide,Cent,model.std_estm);
  predic = round(arespredict(model.model, records));
% Random forest
elseif type ==5
  predic = classRF_predict(records(:,model.vari),...
      model.model);
end
```

D.21 Function: `FinalEval.m`

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Counts corect drop out prredictions, flase alarms and
% predicted drop out semesters inadvance. Plots corect
% predicted drop outs and false alarms.
% Adress: Main_CART, Main_LR, Main_Bagging, Main_PCA, Main_RF,
% Main_MARS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [count, mis_pred,sem_advance] = FinalEval...
    (S,stat,predic, sem_BC, models)
% If prediction is in text, change it in numerical
if ¬ isnumeric(predic)
   predict_num = zeros(size(predic));
   predict_num(strcmp(predic,'drop out'))=1;
   predict_num(strcmp(predic,'pass'))=0;
```

```
else
    predict_num  = predic;
end
count = zeros(size(S,2),1);
mis_pred = zeros(size(S,2),1);
sem_advance = zeros(size(S,2),1);
sem_temp = sem_BC;
% Counts corrected predicted drop outs and falls alarms
for i = 1:size(S,2)
    for j = 1:size(stat,1)
        if stat(j,1) == 0 && predict_num(j,i) == 1 &&...
                sem_temp(j,1)≥ S(i)
            mis_pred(i) = mis_pred(i)+1;
        elseif stat(j,1)== predict_num(j,i) &&...
                predict_num(j,i) ==1 && sem_temp(j,1)≥ S(i)
            count(i) = count(i)+1;
            sem_advance(i) = sem_advance(i)+...
                sem_temp(j,1) - S(i) +1;
            sem_temp(j,1) = -1;
        end
    end
end
%Plots predictions
figure();
plot(count, '-b');
 hold on;
 plot(mis_pred, '-r');
 hold off;
set(gca, 'XTick',1:size(S,2),'XTicklabel',models);
 xlabel('Models');
ylabel('Predicted student to drop out');
```

D.22 Function: Records.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Creating records martirx
% Adress: Main_CART, Main_LR, Main_Bagging, Main_PCA, Main_RF,
% Main_MARS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function records = Records(semester, semester2,in,GPA_O,...
    GPA_S,ECTS_P, ECTS_T,ECTS_A,ECTS_R,ECTS_L)
records = in(:,1:  end);
if semester ≠20
    records(:,size(records,2)+1:size(records,2)+...
```

```
        (semester2-1))= GPA_O(:,2:semester2);
    records(:,size(records,2)+1:size(records,2)+...
        (semester2-1))= GPA_S(:,2:semester2);
    records(:,size(records,2)+1:size(records,2)+...
        (semester2-1))= ECTS_P(:,2:semester2);
    records(:,size(records,2)+1:size(records,2)+...
        (semester2-1))= ECTS_T(:,2:semester2);
    records(:,size(records,2)+1:size(records,2)+...
        (semester2-1))= ECTS_A(:,2:semester2);
    records(:,size(records,2)+1:size(records,2)+...
        (semester2-1))= ECTS_R(:,2:semester2);
    records(:,size(records,2)+1:size(records,2)+...
        (semester2-1))=  ECTS_L(:,2:semester2);
end
```

D.23 Function: SSS.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Supervised set selesction. Cournts drop outs in every
% semester and divides it 2:8 for test and training set.
% Adress: DataDivision.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [tr, tr_e, te, te_e] = SSS(B,sem)
d=1; p = 1;
% Creats four set. Two for those who passed and two for
% those that drop out. Saves numbur of semesters  and
% indexes in the original set
for i = 1:size(B,1)
    if strcmp(B(i,1), 'drop out')
        drop_sem(d,1) = sem(i,1);
        drop_stud(d,1) = i;
        d = d+1;
    else
        pass_sem(p,1) = sem(i,1);
        pass_stud(p,1) = i;
        p = p +1;
    end
end
% Line represents number of semester
% First column, total amount of student at that semester
% Second colum, 80% of total amout of students at that
% semester. Rest of colums represent student idexes
for i = 0:18
    k = 3;
```

```
28      drop(i+1,1) = histc(drop_sem,i);
29      drop(i+1,2) = round(drop(i+1,1)*0.8);
30      for j = 1:size(drop_sem,1)
31          if drop_sem(j,1) == i && drop(i+1,1) ~= 0
32              drop(i+1, k) = drop_stud(j,1);
33              k = k +1;
34          end
35      end
36  end
37  for i = 1:13
38      k = 3;
39      pass(i,1) = histc(pass_sem,i);
40      pass(i,2) = round(pass(i,1)*0.8);
41      for j = 1:size(pass_sem,1)
42          if pass_sem(j,1) == i && pass(i,1) ~= 0
43              pass(i, k) = pass_stud(j,1);
44              k = k +1;
45          end
46      end
47  end
48  % Divides to training and test sets
49  k = 1; l = 1;
50  for i = 1:size(drop,1)
51      if drop(i,1) ~= 0
52          index= randperm(drop(i,1));
53          for j = 1:drop(i,2)
54              tr(k,1)= drop(i, index(j)+2);
55              tr_e(k,1) = {'drop out'};
56              k = k +1;
57          end
58          for j = drop(i,2)+1:drop(i,1)
59              te(l,1)= drop(i, index(j)+2);
60              te_e(l,1) = {'drop out'};
61              l = l +1;
62          end
63      end
64  end
65  for i = 1:size(pass,1)
66      if pass(i,1) ~= 0
67          index= randperm(pass(i,1));
68          for j = 1:pass(i,2)
69              tr(k,1)= pass(i, index(j)+2);
70              tr_e(k,1) = {'pass'};
71              k = k +1;
72          end
73          for j = pass(i,2)+1:pass(i,1)
74              te(l,1)= pass(i, index(j)+2);
```

```
75              te_e(l,1) = {'pass'};
76              l = l +1;
77          end
78      end
79  end
```

D.24 Function: NumberSemester.m

```
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Counts how many semester student was studying program.
3  % Adress: Main_CART, Main_LR, Main_Bagging, Main_PCA, Main_RF,
4  % Main_MARS
5  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6  function sem = NumberSemester(ECTS)
7  for i = 1:size(ECTS,1)
8      for j = 2:size(ECTS,2)-1
9          sem(i,2) = ECTS(i,1);
10         if ((ECTS(i,size(ECTS,2)-j))) ~= 0
11             sem(i,1) = size(ECTS,2) - j-1 ;
12             break;
13         end
14     end
15 end
```

Bibliography

[1] Ana M. Aguilera, Manuel Escabias and Mariano J. Valderrama. 'Using principal components for estimating logistic regression with high-dimensional multicollinear data'. In: *Computational Statistics & Data Analysis* 50.8 (2006), pp. 1905 –1924. ISSN: 0167-9473. DOI: DOI:10.1016/j.csda.2005.03.011. URL: http://www.sciencedirect.com/science/article/pii/S0167947305000630 (cit. on p. 17).

[2] L. Breiman and A. Cutler. *Random Forests*. 29th July 2011. URL: http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm (cit. on p. 25).

[3] Leo Breiman. 'Random Forests'. In: *JOUR*. Machine Learning 45 (1 2001), p. 532. ISSN: 08856,12515730565,10.1023/A:101093340432 (cit. on p. 25).

[4] K.H. Esbensen et al. *Multivariate data analysis: in practice : an introduction to multivariate data analysis and experimental design*. Camo, 2002. ISBN: 9788299333030 (cit. on p. 17).

[5] JH Friedman. 'Multivariate Adaptive Regression Ssplines'. In: *Annals of Statistics* 19 (1 1991), pp. 123–141. ISSN: 00905364 (cit. on p. 73).

[6] D. J. Hand and W. E. Henley. 'Statistical Classification Methods in Consumer Credit Scoring: A Review'. English. In: *Journal of the Royal Statistical Society. Series A (Statistics in Society)* 160.3 (1997), pp.

523–541. ISSN: 09641998. URL: http://www.jstor.org/stable/2983268 (cit. on p. 12).

[7] Trevor J. Hastie, Jerome. Friedman and Robert J. Tibshirani. *The elements of statistical learning : data mining, inference, and prediction*. Series: Springer series in statistics 1-745 S. New York, N.Y: Springer, 2009 (cit. on pp. 15, 18).

[8] Horng-I Hsieh, Tsung-Pei Lee and Tian-Shyug Lee. 'Data Mining in Building Behavioral Scoring Models'. In: *JOUR* (2010), pp. 1–4 (cit. on p. 12).

[9] Nan-Chen Hsieh. 'An integrated data mining and behavioral scoring model for analyzing bank customers'. In: *JOUR* 27.4 (2004), pp. 623 –633 (cit. on p. 12).

[10] O Intrator and C Kooperberg. 'Trees and splines in survival analysis'. In: *JOUR. Statistical Methods in Medical Research* 4 (3 1995). ISSN: 09622802 14770334 (cit. on p. 18).

[11] Alan Julian Izenman. *Modern multivariate statistical techniques : regression, classification, and manifold learning*. Series: Springer texts in statistics, 1431-875x. New York: Springer, 2008. Chap. 9, pp. 1–731. ISBN: 9780387781884 0387781889 (cit. on p. 18).

[12] Abhishek Jaiantilal. *Randomforest-matlab*. 27th July 2011. URL: http://code.google.com/p/randomforest-matlab/ (cit. on pp. 26, 54).

[13] G. Jekabsons. *Adaptive Regression Splines toolbox for Matlab/Octave*. Version 1.5. Institute of Applied Computer Systems, Riga Technical University. 2010. 19 pp. DOI: http://www.cs.rtu.lv/jekabsons/ (cit. on p. 72).

[14] Kræn Blume Jensen, Christophe Kolodziejczyk and Torben Pilegaard Jensen. *Student drop-out from Professional Bachelor programmes. Student retention in Danish educational institutions*. Danish Institute of Governmental Research, Oct. 2010, p. 76. URL: http://www.akf.dk/udgivelser_en/2010/2868_frafald_professionsbacheloruddannelserne/ (cit. on pp. 1, 86).

[15] JH. 'Students to face interviews for university admission'. In: *The Copenhagen Post* (16th Apr. 2010). URL: http://www.cphpost.dk/news/192-universities/48775-students-to-face-interviews-for-university-admission.html (visited on 31/05/2011) (cit. on pp. 1, 2).

[16] Douglas C. Montgomery. *Introduction to statistical quality control*. Ed. by 5th. Hoboken, NJ: Wiley, 2005. Chap. 4, 1–759 s. ISBN: 9780471656319 (cit. on p. 5).

[17] Kristin K. Nicodemus. 'etter to the Editor: On the stability and ranking of predictors from random forest variable importance measuresLetter to the Editor: On the stability and ranking of predictors from random forest variable importance measures'. In: *JOURBriefings in Bioinformatics* 12 (4 2011), p. 369 373. ISSN: 14675463 (cit. on p. 26).

[18] Magnus Brand Tingstrøm. 'Minorities drop out of university'. Trans. by Marianne Beck Hassl. In: *University Post* (24th Aug. 2009). URL: http://universitypost.dk/article/minorities-drop-out-university (cit. on p. 1).