Preventing university drop outs

Early detection of university drop outs using machine learning

Eddie Fernberg

Master's Thesis

MEng in Big Data Analytics

2025

Master's Thesis

Eddie Fernberg

Preventing university drop outs. Early detection of university drop outs using machine learning.

Arcada University of Applied Sciences: MEng in Big Data Analytics, 2025.

Identification number:

048147433

Commissioned by:

N/A

Abstract:

University drop out rates present a significant challenge for higher education institutions, leading to wasted resources and decreased graduation rates. This thesis explores the potential of machine learning to provide an early detection system for identifying students at risk of dropping out. Using academic data from SISU, a student information system used in Finland, various machine learning models were tested and optimized. Key steps included feature engineering, data balancing using SMOTE, and model evaluation to ensure reliable predictions. The results indicate that academic data alone is sufficient for creating a viable predictive system. The findings emphasize the importance of early intervention strategies and the potential for data-driven decision-making in university administration. This study demonstrates that machine learning can serve as a powerful tool for drop out prevention, enabling universities to take proactive steps toward student retention.

Keywords: Dropout detection, machine learning, SISU, feature engineering

Contents

Figures

Listings

Tables

Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
ANN	Artificial Neural Network
Hanken	Svenska handelshögskolan (Hanken School of Economics)
KNN	K-Nearest Neighbor
LightGBM	Light Gradient-Boosting Machine
LMS	Learning Management System
ML	Machine Learning
MLP	Multi Layer Perceptron
PCA	Principal Component Analysis
REST	Representational State Transfer
SaaS	Software as a Service
SIS	Student Infomation System
SISU	Student Information System for Universities
SMOTE	Synthetic Minority Oversampling Technique
SMOTETOMEK	SMOTE with Tomek links
SSMS	Microsoft SQL Server Management Studio
SVM	Support Vector Machine
SQL	Structured Query Language
UAS	University of Applied Science

Acknowledgments

I would like to express my deepest gratitude to my supervisor Truong An Pham for his guidance in the creation of this thesis.

I also could not have undertaken this journey without the support of Linda Gerkman, director for education and digital services at Hanken. Apart from allowing me access to the dataset, she spent many hours explaining it to me.

Last but not least I would like to mention Sofia Wiksten, my fiancée and rock, for her endless support of my pursuit of education.

VI UTHÄRDAR

1. Introduction

This study aims to develop a machine learning-based solution capable of accurately identifying patterns associated with student dropouts in higher education. By enabling early detection, such a system can support institutions in making timely and informed decisions to mitigate dropout rates.

Student dropouts from higher education present a range of challenges. They can lead to unnecessary administrative overhead, inefficient use of institutional resources (Huynh-Cam, Chen, and Lu), and significant emotional and academic stress for the students themselves. These consequences not only affect the individuals involved but also impact the broader educational ecosystem.

In some countries, such as Finland, the funding model for higher education institutions is closely tied to the number of graduates they produce (Ministry of Education and Culture). This creates a strong institutional incentive to reduce dropout rates and ensure students successfully complete their studies.

If institutions can identify students at risk of dropping out early in their academic journey, there may still be time to intervene effectively. In many cases, even modest adjustments—such as academic counseling, changes in course load, or targeted support—can significantly alter a student's trajectory.

This underscores the need for a method that not only detects potential dropouts early but also offers actionable insights into the factors contributing to that risk. A well-designed predictive model can serve as a valuable tool for academic advisors and decision-makers, guiding interventions based on data-driven evidence.

The central research question driving this study is:

How accurately can machine learning models predict a student's probability of dropping out?

The objective is to build a reliable binary classification model that distinguishes between students likely to drop out and those likely to persist. Furthermore, the study seeks to identify the key features and variables that most strongly influence dropout risk, providing both predictive power and interpretability.

1.1 Background

SISU (a backronym for Student Information System for Universities) is a widely used student information system adopted by many universities and universities of applied sciences in Finland. It is developed by Funidata and provided as a SaaS solution (Funidata a).

Funidata is a joint venture owned by several educational institutions and primarily develops IT systems tailored to the needs of its stakeholders (Funidata b).

Among the various features offered by SISU, one of the most central is the study registry, which records all student attainments and grades. This repository of academic performance data provides a valuable resource for analysing patterns related to student drop outs.

For the purposes of this study, I have been granted access to SISU data from Hanken Svenska handelshögskolan. In my role as a data specialist at the university, I work daily with the institution's data warehouse and am familiar with its structure and content (Hanken Svenska handelshögskolan).

2. Related Work

The first step of this study was to perform a literature study to see if and how other had solved similar problems.

I was primarily interested in seeing:

- which features were selected.
- which models were used and yielded the best result.

Other good practises, or pitfalls to avoid, were of course also of great interest.

2.1 Literature study

I began with collecting previous knowledge within the field and performed a literature study on available studies.

An initial source for finding further material was recommended by my supervisor: a literature review by Albán and Mauricio. This study shows an extensive overview of chosen features and methods for predicting and analysing them.

By the supervisor's suggestion I also turned to Google Scholar for finding further relevant work. The search term used was:

"Drop out rate detection in university"

With this search term I got many promising matches. After sifting through the top results, I proceeded with **17** papers that had studied a similar issue. I tried to focus on more recent papers, but some were a bit outdated. These were kept nevertehless, because they could give valuable insights of the development of the field.

A good systematic review was from Elza and Widyasari, a comprehensive overview which helped me find some articles not indexed by Google Scholar.

There were four (4) themes frequently mentioned in the found studies:

- Academic vs. personal data
- Data sources
- Balancing
- Models

These will be discussed in more detail in their respective sections down below.

2.1.1 Academic vs. personal data

I quickly concluded that different studies had used different feature sets: some had used only academical features while others had used personal. Some made use of combinations. I compared the information found to the features that were available to me, to see if that study was viable in my case. The dataset I have access to contains very few personal features (name, gender and nationality) and thus this study will be focused on using academical features.

Using only personal data is quite uncommon in the previous studies. One example is Huynh-Cam, Chen, and Lu who uses personal data with family background information such as parents' occupation, salary, and education level. This data is collected before the first semester.

Only using academical data for the models is a bit more common. In most cases, grades or other markers for study performance are used. Bañeres, Rodríguez, Guerrero-Roldán, and Karadeniz uses number of courses, grades and test performance. Kabathova and Drlik are interested in seeing grades on exams and projects. Cho, Yu, and Kim uses average grade, number of courses and admission status. It is the one study that closest resembles the dataset I have available. Last but not least, I have a domestic conference paper from Pesonen, Fomkin, and Jokipii which looks into the amount of study credits 18 months prior and the distance to study right validity end.

Combined data is the most common form of features. Fernandez-Garcia, Preciado, Melchor, Rodriguez-Echeverria, Conejero, and Sanchez-Figueroa uses mainly academic data but also includes birth date and municipality. An interesting aspect is that the data is collected at several stages: prior to enrolment and then at the end of the first, second, third and fourth semester respectively.

Two studies focuses on performance in one single subject (programming). These features included scoring of assignments, amount of attempts, student participation (Martinez, Sood, and Mahto), and grade (Cooper). The latter study also uses data regarding the main study field and whether the student is native american or not. This kind of ethnicity data is neither available nor interesting for my study.

Some studies relies heavier on personal data than academic dito. Examples of these includes Lee and Chung which is mainly interested in behaviour such as tendency to violate rules, relationships with friends, and maladjustment. Martins, Baptista, Machado, and Realinho uses mainly academical data such as grade and enrolled units but also personal data such as nationality, scholarships, and the parents' occupations. Segura, Mello, and Hernández also takes scholarships into consideration. Library loans are considered by Kim, Choi, Jun, and Lee. Lottering, Hans, and Lall is a case study which mainly uses data on amount of modules and credits, but also possible disability and aid.

Opazo, Moreno, Álvarez Miranda, and Pereira tries to compare different universities based on the schools' rankings and the students' scores from math tests. Some personal data in the form of home municipality and gender are also used. Bedregal-Alpaca, Cornejo-Aparicio, Zárate-Valderrama, and Yanque-Churo looks into gender and school of origin. This was interesting to read, but not relevant to my project due to lots of complicated models.

2.1.2 Data sources

A main difference between studies is from where the data is obtained. There are two (2) major sources: SIS and LMS.

The most common variety is the SIS. Cho et al.; Opazo et al.; Segura et al.; Bedregal-Alpaca et al.; Lottering et al., and Pesonen et al. are all using this kind of data source. South Korean NEIS is a SIS where student behaviour is recorded along with academic data (Lee and Chung).

An LMS is the second most used data source. That kind of data tends to be leaning more towards academic performance metrics rather than study data. I have no access to an LMS, so this is not usable in my case. This data source is utilised by Martinez et al.; Bañeres et al.; Kabathova and Drlik, and Fernandez-Garcia et al.

There are also other data sources available. Cooper uses data from a teachers gradebook for a programming course. In South Korea (Martinez et al.) and Taiwan (Huynh-Cam et al.) for example, there are systems where you can extract data about a student's parents. Finally there are some outliers: Martins et al. uses SIS data and data from external national databases; Kim et al. is based upon SIS data, but also includes information on library loans.

2.1.3 Balancing

Another important theme was the balance of the dataset: there will inevitably be many more rows of non-drop outs than drop outs, which might skew the dataset.

Several studies mentions methods to mitigate this skewness.

The most common is SMOTE. This method is used by Huynh-Cam et al.; Martinez et al.; Lee and Chung; Bañeres et al.; Martins et al., and Cho et al.

Two studies mentions also other balancing methods, namely SMOTETOMEK (Fernandez-Garcia et al.) and ADASYN (Kim et al.).

Due to its popularity in the previous studies, I decided to use SMOTE for my own solution. Initially I was not to use balancing, but after some experiments I concluded that it would be needed for the rather imbalanced dataset. Thus the results are shown before and after the balancing has been applied.

2.1.4 Models

Models are for obvious reasons an important part of every study that wants to predict drop outs.

The previous studies uses a vast array of different models. All models and their accuracy, if available, were compiled into a table and sorted based on the highest value. The ones that deemed feasible to compare with were selected for this study. See *listing 5* for the chosen models.

See *Appendix A* for a complete list of models used in the previous studies and their respective scoring.

3. Research Methodology

This study is based on testing and tuning models to see if they yield a sufficient result.

3.1 Tools

All of the data were processed in Python. Jupyter notebooks (Jupyter Team) were used to analyse the code and display interactive output, with the aid of several freely available libraries. Visualisation was handled by matplotlib (Matplotlib development team), seaborn (Waskom) and matplot2tikz (deGelder). Pylatex (Fennema) was used for creating tables and formatting straight from the notebook. See *listing 1*.

The machine learning models and related tools, such as data splitting, pipelines and score measuring were loaded from Scikit-learn (scikit-learn). See *listing 2*.

A very good instruction on how to overall perform all the steps and what configurations to use was provided by Lee. This book makes a solid starting point for all things machine learning.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib as mpl
import matplot2tikz
import seaborn as sns
import math

from pylatex import Table, LongTable, Tabular, Center, MultiRow,
    NoEscape
from pylatex.utils import bold
from os import path
from sqlalchemy import create_engine
```

Listing 1: Python libraries

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import RandomForestClassifier,
    GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.pipeline import Pipeline
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay,
    accuracy_score, precision_score, recall_score, roc_auc_score,
    roc_curve
from sklearn.inspection import permutation_importance
```

Listing 2: Scikit-learn modules

3.2 Acquiring data

The data in SISU resides on Funidata's servers, due to its SaaS nature. To be able to utilise the data the universities load it from REST API:s and then store it in some sort of data warehouse.

Hanken's data warehouse is based on SQL and thus easy to access for this project. There are several tables associated with SISU, but I chose to load the data for study rights joined with aggregated credit data. Each row thus represents a study right with several columns. Other tables, such as attainments and courses are not interesting for the analysis of study progress.

An SQL file was written for the `SELECT` statements. See *Appendix B* for the SQL file definition. This file made it possible to simultaneously inspect the rowset in SSMS and load the query from Python, all while adjusting the parameters. SQLAlchemy in conjunction with Pandas `read_sql` function was utilised in order to load the data into a dataframe that could be used in the notebook.

Initially all rows were selected, without any filtering, in order to inspect and analyse the dataset.

3.3 Dataset

The dataset contains information about study rights. Academic data regarding study progress and grading are the most important parts, but some personal info such as gender and whether the person is a Finnish citizen are included.

There are also several columns with calculated values, such as amount of terms attended, registered and absent.

The study office can into SISU enter a value whether the student is a drop out based on certain conditions. The conditions mostly requires manual evaluation, except if the student does not register for the semester, in which case it is assigned automatically. This is the target value for the predicitions.

See *Appendix C* for a complete list of the columns in the dataset.

3.4 Data preparation

Some manipulations in SQL were needed in order to make the data usable, namely casting integers and bits to their correct data types.

These changes were also written into the aforementioned SQL file (*Appendix B*). This was mainly a functional choice - it is easier to have the fields formatted correctly from the beginning of the process.

The data preparation was an iterative process where changes gradually were added to the SQL file after more insights were gained.

3.4.1 Filtering

Along the way filtering was added in order to fine tune the selection. The main reason is to make the dataset a tad lighter and more manageable - there are tens of thousands of rows in total.

Rows marked `not started` were filtered out as to not skew the predicitons. Most of these rows are missing content in the important columns. The year was selected to make the training set smaller. A future version of the machine learning solution will use a larger dataset.

Rows were filtered out with the parameters in *listing 3*.

```
1 WHERE status <> 'not started'
2 AND year = 2024;
```

Listing 3: SQL filters

3.5 Feature engineering

The dataset cannot be thrown into the models as is and thus requires some feature engineering. This became the biggest part of the project and took quite some time to get right.

This was yet again an iterative process of trial and error when loading the data and see how the models behave. By far, this step was the most time consuming.

The feature engineering consisted of the following steps:

3.5.1 Dependent variables

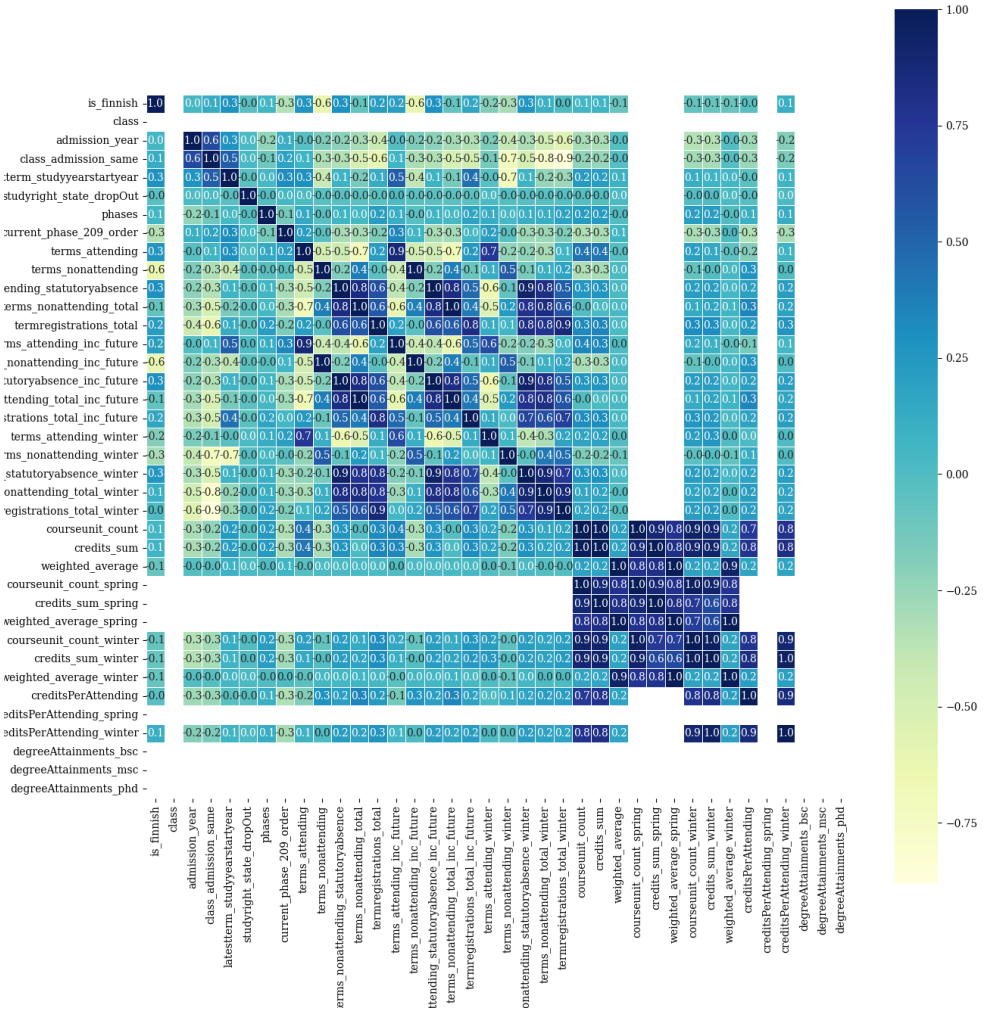

Figure 1: Correlation matrix at data load

From the correlation matrix (*figure 1*) I can see that several numerical columns share a high dependency. The darker the blue, the higher dependency. This might cause unwanted leakage and needs to be adressed to optimise the models.

Leakage becomes apparent when looking at ROC AUC for a model: a high ROC AUC might indicate that columns are dependent.

I manually iterated a `df.drop(columns=[a,b,c,etc])` with dependent columns added to the column list until both the correlation matrix and the ROC AUC yielded better results. *Listing 4* shows all columns that were added to the list.

```
#Remove leaking columns:
dependant = ["class","admission_year","latestterm_studyyearstartyear","
    studyrightState_20_9","educationLocation_sv","
    studyright_state_dropOut_integrated_bsc_en","
    studyright_state_dropOut_integrated_msc_en","
    Person_DegreeAttainment_bachelor","Person_DegreeAttainment_masters"
    ,"Person_DegreeAttainment_doctor","terms_attending_inc_future","
    terms_nonattending_inc_future","
    terms_nonattending_statutoryabsence_inc_future","
    terms_nonattending_total_inc_future","
    termregistrations_total_inc_future","terms_attending_winter","
    terms_nonattending_winter","
    terms_nonattending_statutoryabsence_winter","
    terms_nonattending_total_winter","termregistrations_total_winter","
    courseunit_count_spring","credits_sum_spring","
    studyweeks_sum_spring","weighted_average_spring","
    courseunit_count_winter","credits_sum_winter","
    studyweeks_sum_winter","weighted_average_winter","
    creditsPerAttending_spring","creditsPerAttending_winter","
    graduation_standard_period_education_en","standard_period_bsc_en","
    standard_period_msc_en","standard_period_phd_en","
    degreeAttainments_bsc","degreeAttainments_msc","
    degreeAttainments_phd"]
df = df.drop(columns=dependant)
```

Listing 4: Dropped dependent columns

3.5.2 Column completeness

Column completeness is a measure on how many rows are having values in that particular column in the dataframe. Total column completeness is calculated to: **53.74%**.

This tells us that almost half of the columns are empty and can be filtered out, which clearly can be seen as white bands in the correlation matrix (*figure 1*). Only a handful of the features were complete and could be used with the models.

3.5.3 Feature amount

The dataframe was manipulated in several steps, which altered the amount of features available:

1. Data loaded from SQL server.

2. Columns with 90% of `NULL` values were dropped.

3. Columns that were dependent were dropped.

4. Categorical features were one-hot encoded.

Table 1 shows how the amount of features changes at every step above. Figure 2 shows the completeness after empty and dependent columns were dropped. Figure 3 shows the correlation for all remaining numerical features.

Step	Function	Amount
1	`read_sql`	62
2	`drop(columns)`	25
3	`dropna`	22
4	`get_dummies`	71

Table 1: Amount of features

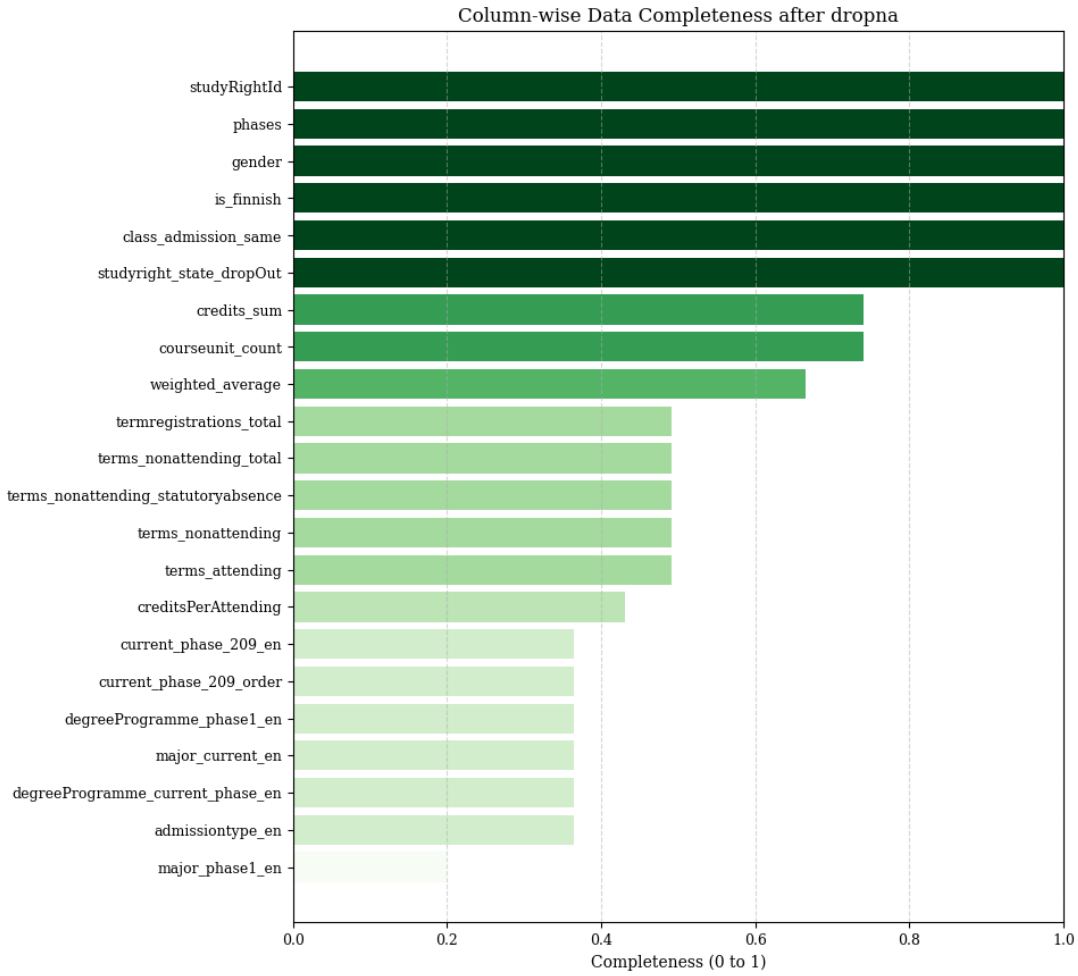

Figure 2: Column completeness after drop and dropna

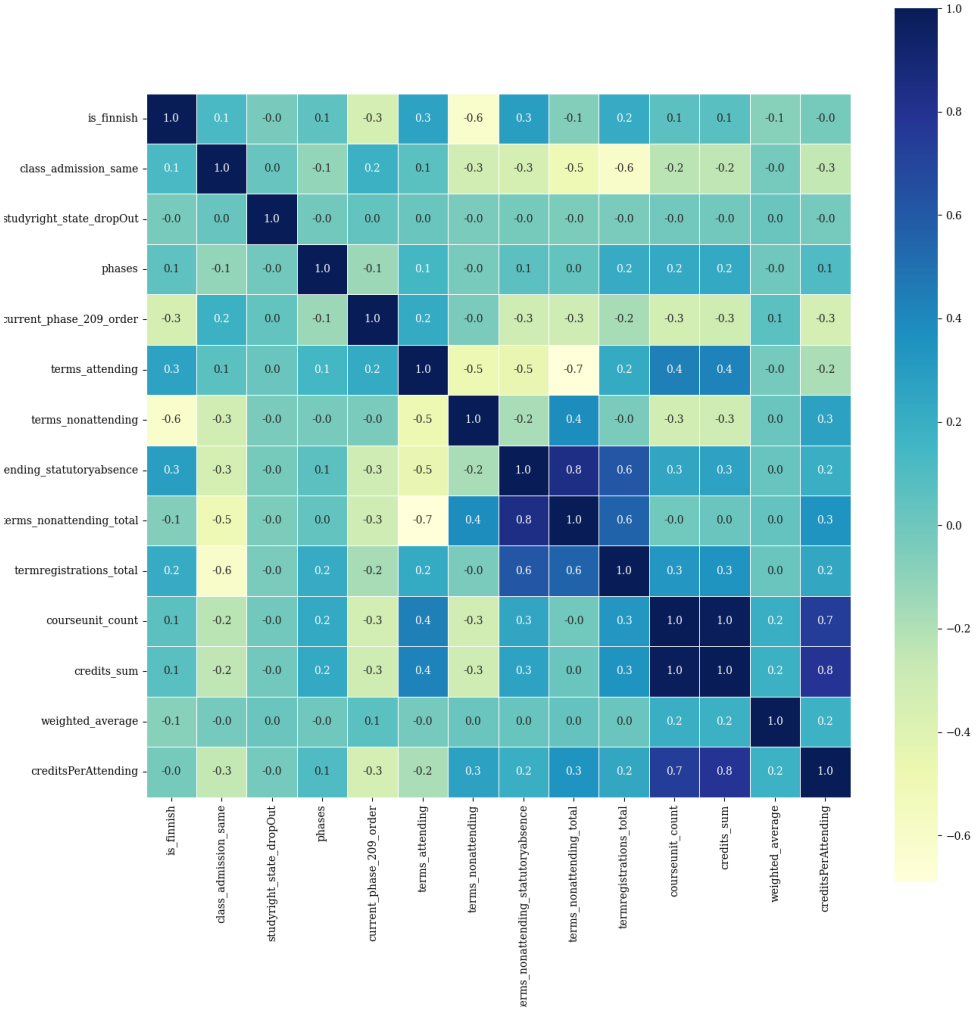

Figure 3: Correlation matrix after drop and dropna

3.5.4 Imbalances

The dataset is heavily imbalanced due to the very few rows with the drop out status. *Figure 4* shows that only a small fraction of the dataset is marked as drop out, which makes this a minority group,

Imbalance might cause heavy oversampling, which fits the model too good to a too narrow result set, especially when the classification is binary. This can be solved by balancing the dataset (Provost and Fawcett).

SMOTE is a technique mentioned in several previous studies and might help with the imbalance. It works by creating synthetic minority samples instead of ramdom replacing. The new values are produced by the help of k-distance and multiplied by a number from 0 to 1. It makes the minority class more general (Müller and Guido). Most of the SMOTE usage instructions were found from the official manual (Imbalanced-learn).

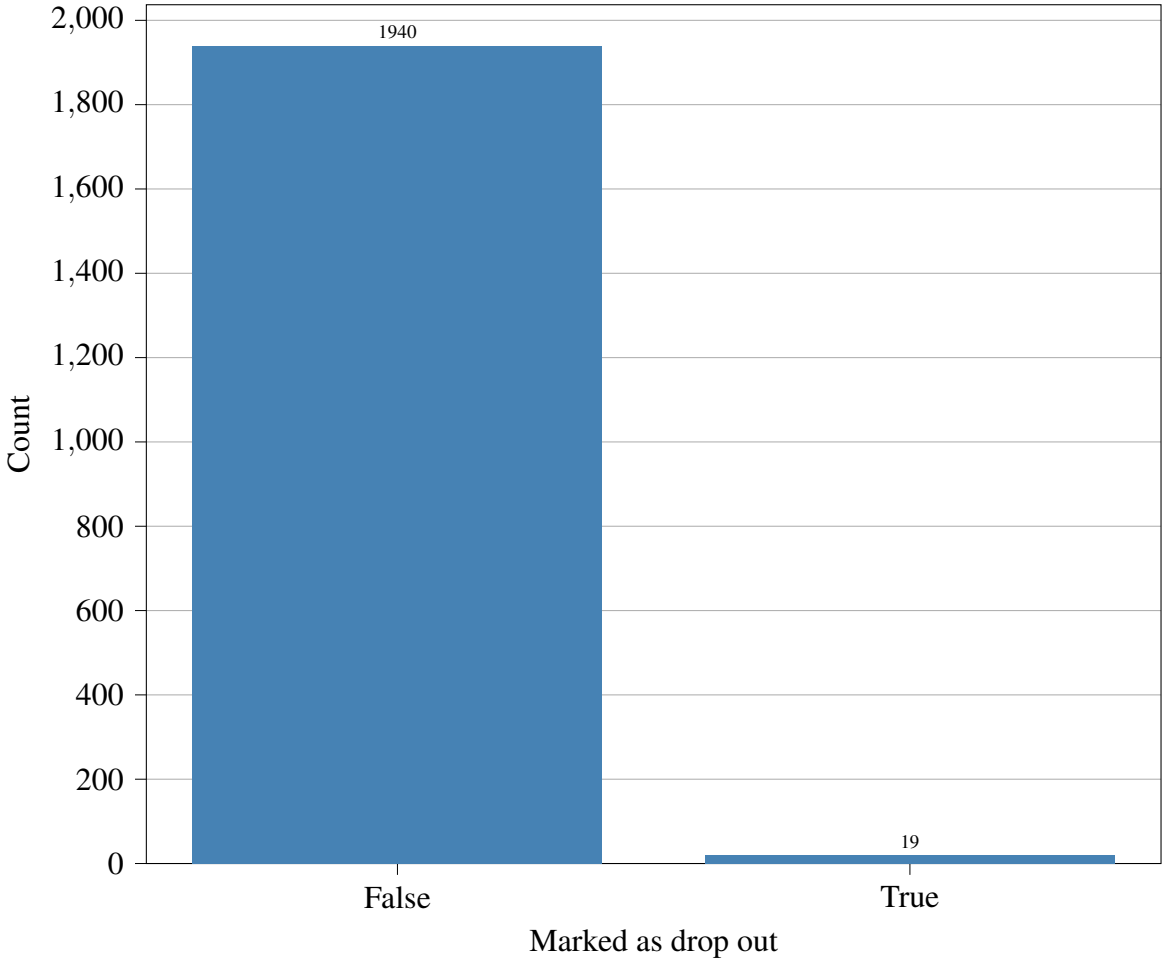

Figure 4: Drop out status distribution

4. Experiments

The experiments consisted, apart from the tedious feature engineering, mainly of trying different model configurations. I wanted to see if the prediction score could be made sufficient for usable analysis.

4.1 Models

Most of the previous studies uses different models for their machine learning in order to find the best suited. I decided go by five (5) models for the comparison. The following models were chosen because of their consistency in amount mentioned and their high ranking (see *Appendix A*):

- Decision trees
- MLP
- Random forest
- Gradient boosting
- SVM

There was no particular reason for selecting these models apart from familiarity gained in the previous machine learning courses.

Each chosen model was defined in an object to be tested for their accuracy. See *listing 5*.

All parameters were taken straight from the respective user guides and were not adjusted nor tuned in any way. Only SVM had the parameter of `kernel="linear"` added in order to plot the feature importance.

```
#Define models.
models = {
    "DecisionTree": DecisionTreeClassifier(),
    "MLP": Pipeline([
        ("scaler", StandardScaler()),
        ("mlp", MLPClassifier(max_iter=500))
    ]),
    "RandomForest": RandomForestClassifier(),
    "GradientBoosting": GradientBoostingClassifier(),
    "SVM": Pipeline([
        ("scaler", StandardScaler()),
        ("svc", SVC(probability=True, kernel="linear"))  # Enable
    probability output
    ])
}
```

Listing 5: Model definitions

4.2 Predictions

Firstly the target and features were separated into two different dataframes to create `X` and `y`. Then each dataframe was splitted into train and test sets, with a standard split of 80/20.

Figure 5 shows the distribution of drop out status in the test set. As seen, the dataset is rather skewed due to the small amount of drop outs. In this stage I will ignore the imbalance and just feed the dataframe into the models for evaluation.

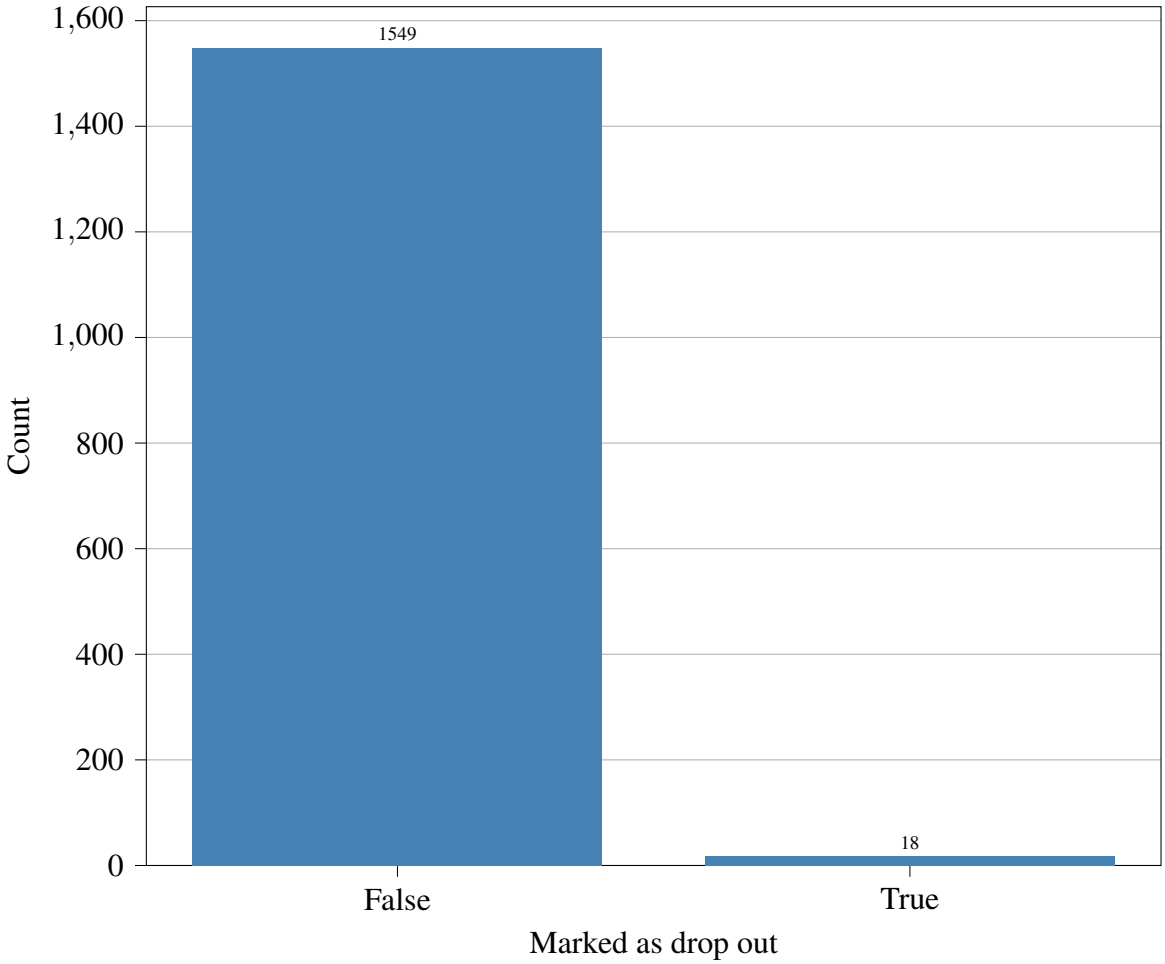

Figure 5: Drop out status distribution for test set

A loop was run to get the score for each of the models in the object in *listing 5*. The results can be seen in *table 2*. While accuracy is high, the precision, recall and ROC AUC are quite low. These numbers might indicate that the model is overfitting the dataset. This behaviour is expected due to the imbalance.

	Accuracy	**Precision**	**Recall**	**ROC AUC**
DecisionTree	0.990	0.000	0.000	0.425
MLP	0.990	0.000	0.000	0.542
RandomForest	0.990	0.000	0.000	0.946
GradientBoosting	0.990	0.000	0.000	0.921
SVM	0.995	0.000	0.000	0.665

Table 2: Metrics of models

The ROC AUC curves are plotted in *figure 6*. There is a quite large spread on the metrics in this plot. Decision trees performs worse than the chance - meaning the prediction is mere a guess. Random forest on the other hand performs quite good. Judging by the straight perpendicular lines, none of these models are actually doing a good job. This is again expected due to the skewed dataset.

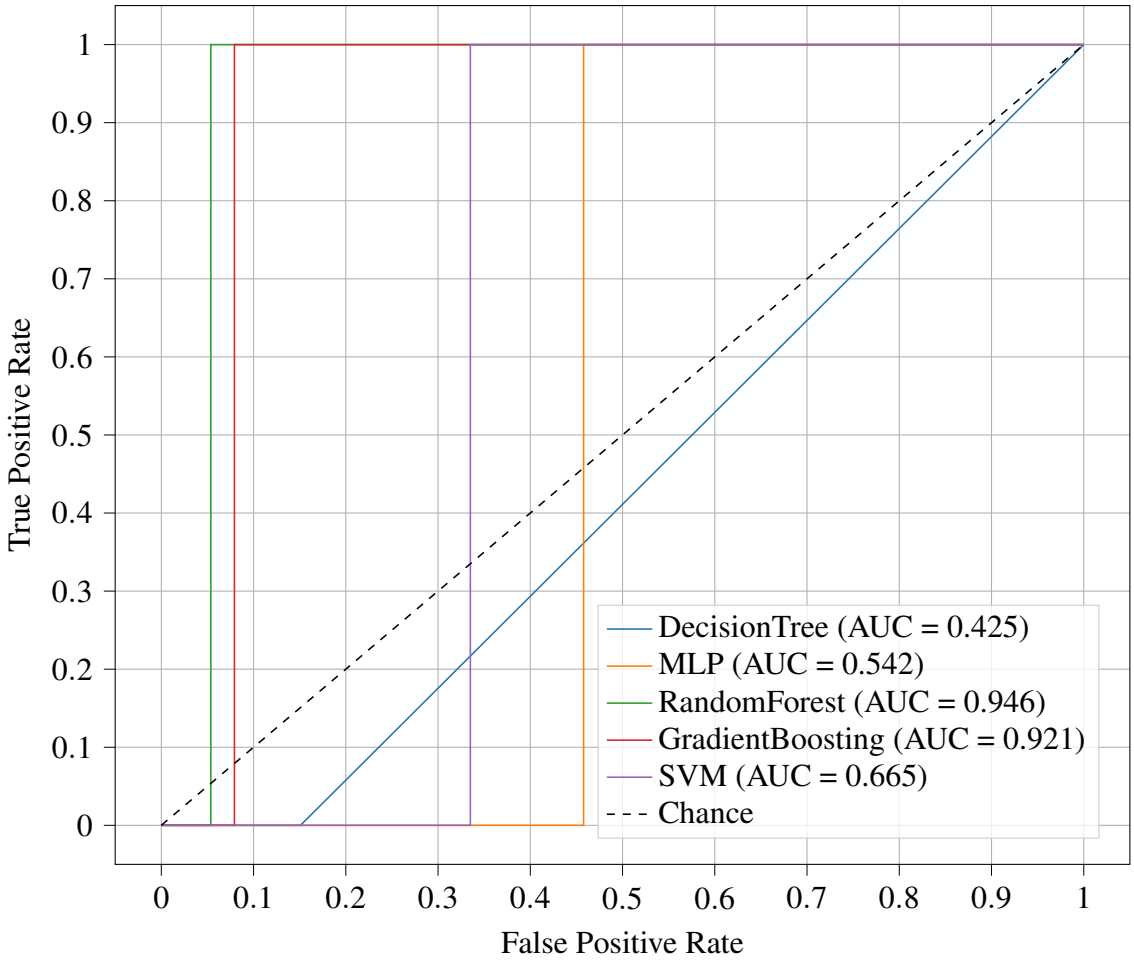

Figure 6: ROC curves for models

By looking at the confusion matrices in *figure 7*, I can see that `true/true` values are zero (0) in every model. This also tells me that the models are not doing a good job predicting the correct values. The amounts should be more symmetrical in order to be usable.

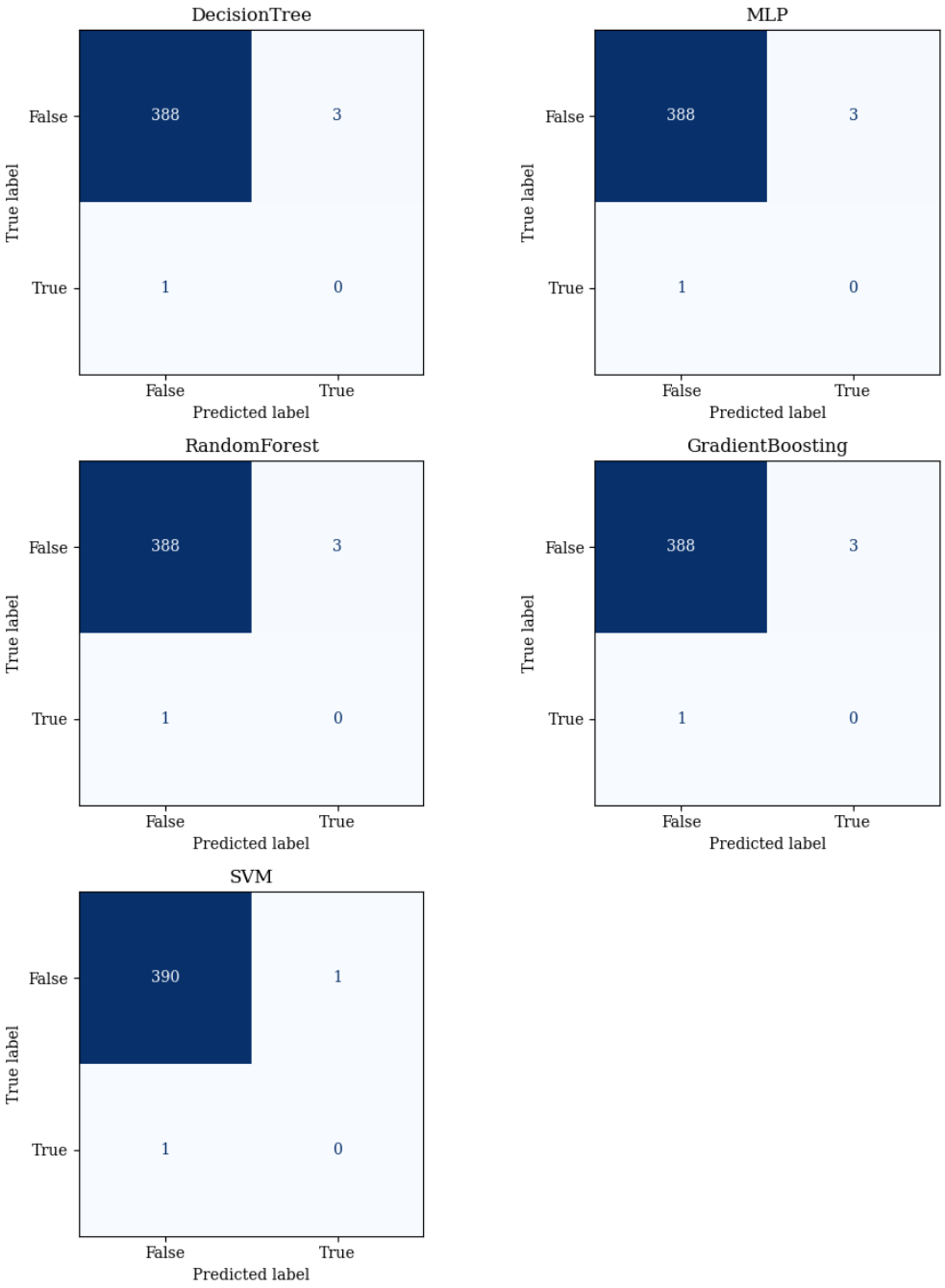

Figure 7: Confusion matrices

Finally I plotted the feature importances in *figure 8*. These bars shows what features the models have used when performing the predictions.

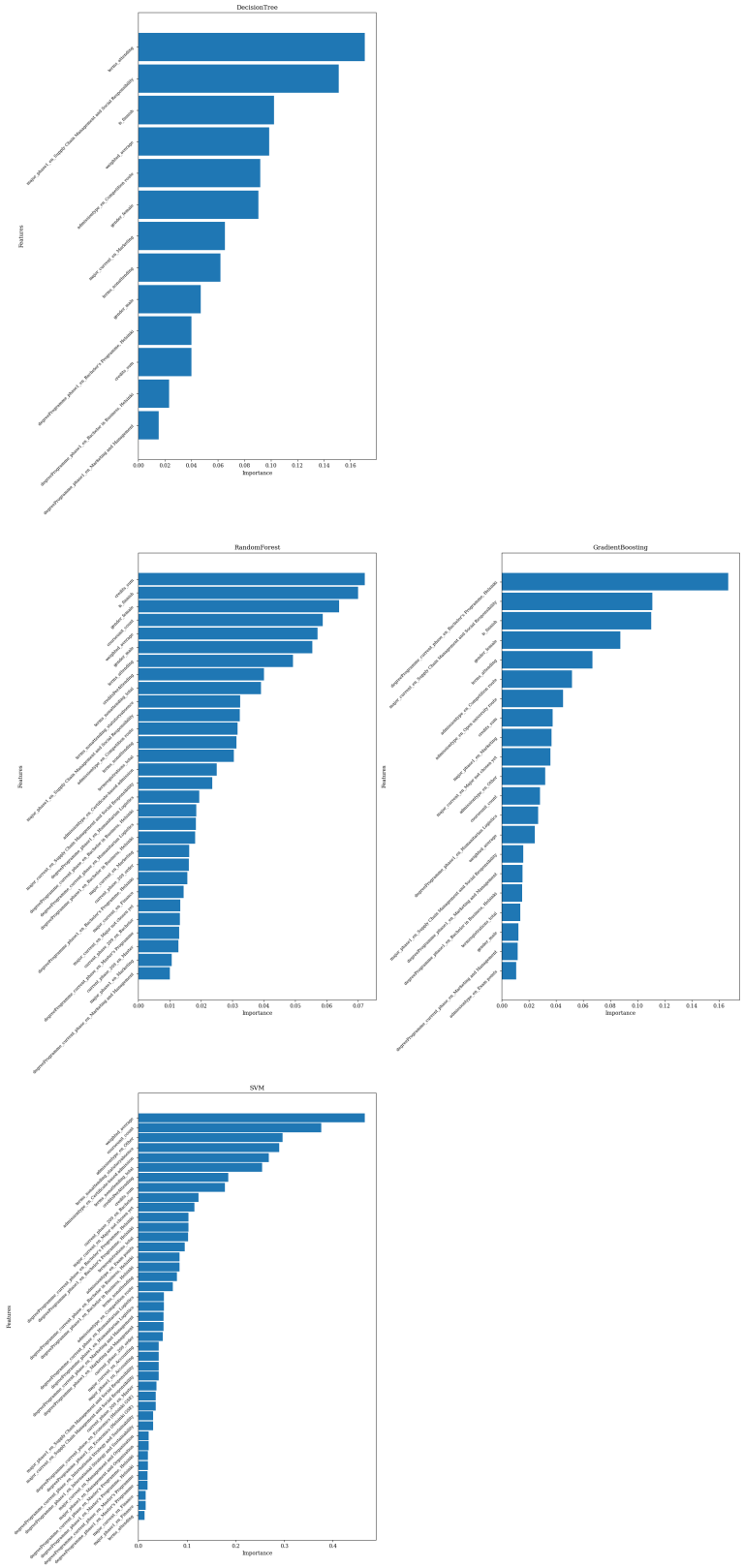

Figure 8: Feature importances

4.3 Predictions with SMOTE

Due to the imbalances, SMOTE was used to make the models less suspicible to overfitting. In the initial test I could see that the models were not performing very well. Let's try to make them more accurate.

The next thing to do is to apply SMOTE. I resample `X` and `y` with the parameter `sampling_strategy="minority"` to synthetically fill the dataset with new rows belonging to the minority class (drop out).

Figure 9 shows the distribution of drop out status in the test set after SMOTE has been applied. Now, the bars are almost of equal height, meaning that the dataset is now balanced.

After this the target and features can be splitted into two different dataframes to create train and test sets, with a split of 80/20.

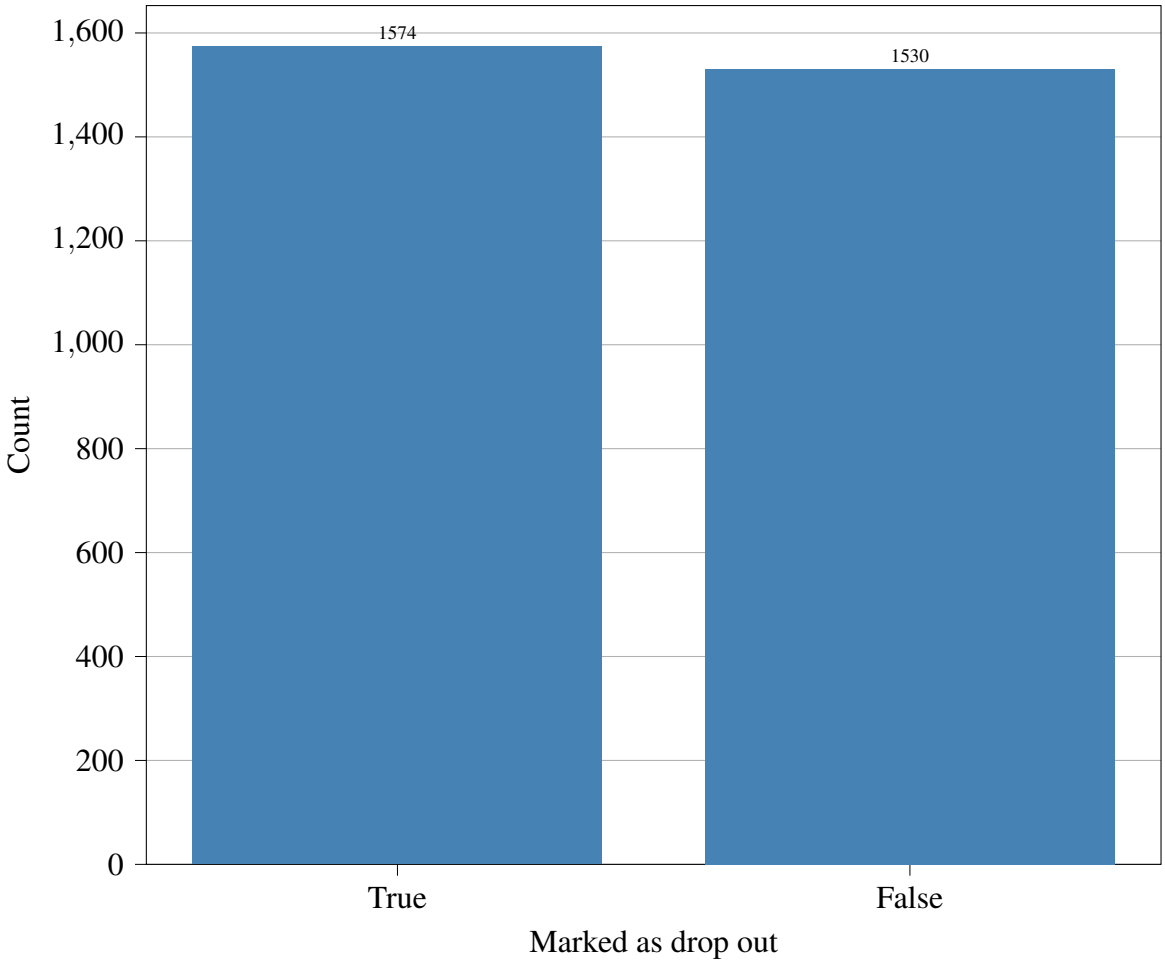

Figure 9: Drop out status distribution for test set (after SMOTE)

A loop was run again to get the score for each of the models in the object in *listing 5*. The results can be seen in *table 3*. Now all the metrics have gone up. High values on all metrics indicates that the model is actually performing well. With a ROC AUC on 97% and upwards, this is very good scoring. A too high value, such as 1, would have been suspect and probably meant that the model is overfitting. But now I can conclude that the model is rather well fitted to the dataset.

	Accuracy	Precision	Recall	ROC AUC
DecisionTree	0.936	0.882	0.997	0.981
MLP	0.932	0.943	0.910	0.987
RandomForest	0.939	0.888	0.997	0.988
GradientBoosting	0.936	0.883	0.995	0.988
SVM	0.930	0.880	0.986	0.979

Table 3: Metrics of models with SMOTE

The well fitted model is also confirmed by looking at the ROC AUC curves in *figure 10*: they are almost angular, which indicates that the models are performing really good and can predict correctly.

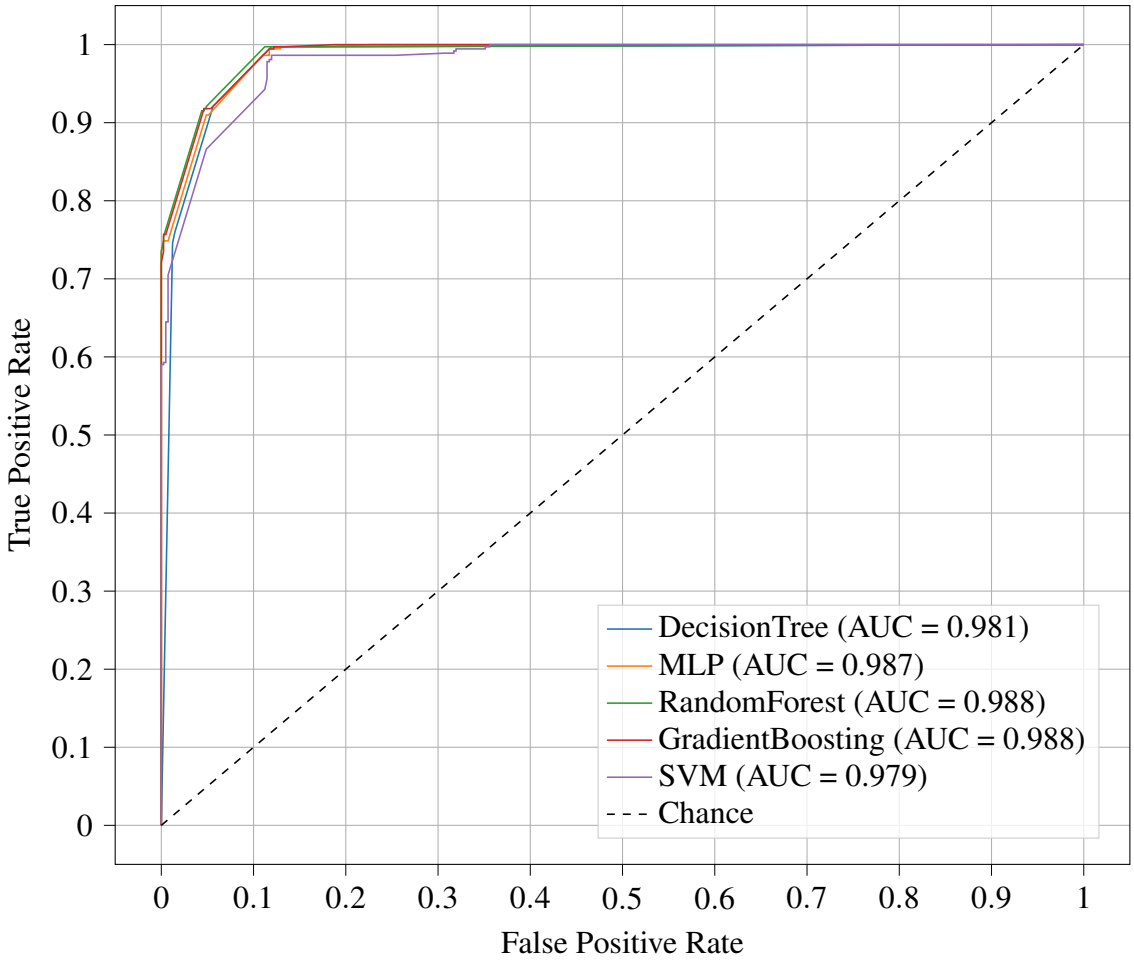

Figure 10: ROC curves for models after SMOTE

Continueing with the confusion matrices in *figure 11* there are promising numbers: the `false/false` and `true/true` predicitions are almost equally sized. This is a indicator that the models are performing well. A coloring that looks like Battenberg cake is desired.

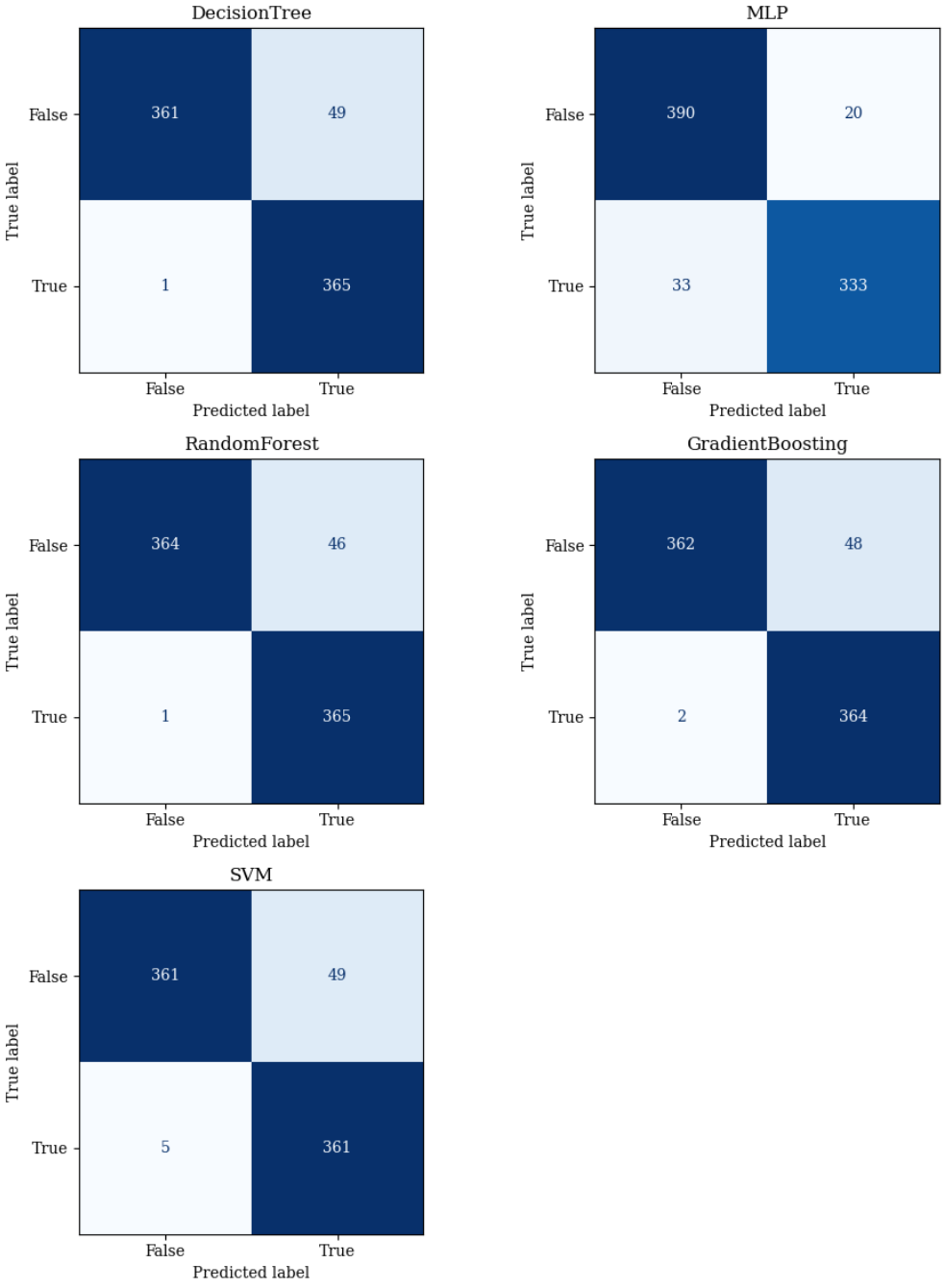

Figure 11: Confusion matrices after SMOTE

Looking at the feature importance plots in *figure 12* there are fewer feature bars. This might indicate that, while fewer, the features are more precise for predicting the desired classes.

I can conclude that balancing the dataset has huge impact on the performance of the models. SMOTE provides means to mitigate skewness and makes the models predictions more precise. The ROC AUC was greatly improved by applying this technique.

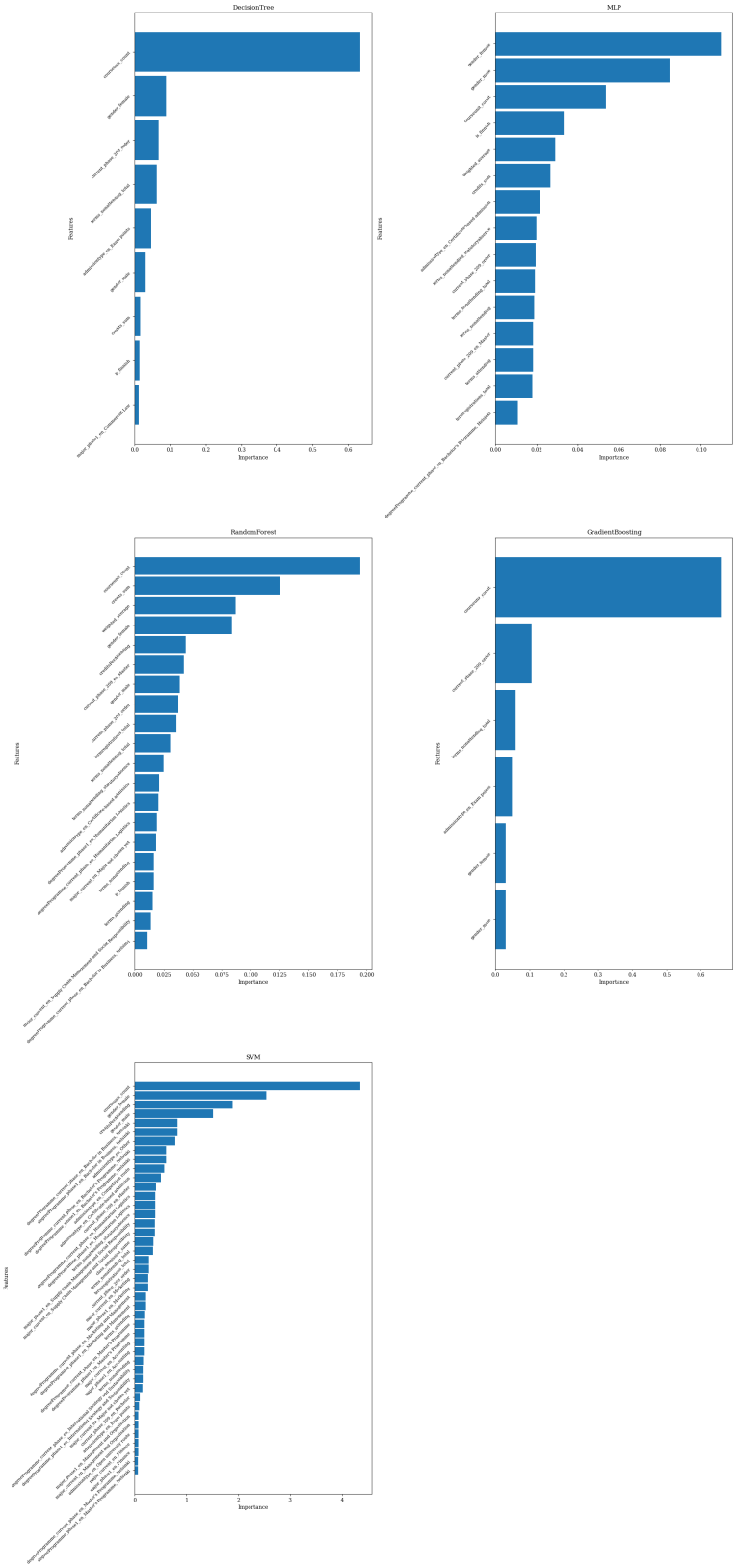

Figure 12: Feature importances after SMOTE

5. Results

The primary goal of this study was to assess whether student dropouts could be accurately predicted using data from the SISU system. A key objective was also to identify which machine learning model would deliver the most effective and reliable results for this classification task.

The experiments demonstrated that model performance varied significantly across different algorithms. All models had initially high accuracy, but very low ROC AUC. The worst performing (Decision tree) had a value of 0.42 and the best performing (Random forest) value was 0.81. Definetely not stellar numbers.

A critical factor in improving model performance was the use of data balancing techniques. Initially, the dataset was heavily skewed toward the majority class—students who did not drop out — which caused the models to overfit and perform poorly when predicting actual dropouts.

To address this, SMOTE was applied. It helped balance the dataset by synthetically generating samples of the minority class. After applying SMOTE, all models saw notable improvements in both recall and precision. Decision tree's and Random forest's accuracy went down to around 0.93 (a more accurate measure) and the ROC AUC went to 98.4 and 99.0 respectively. These findings highlight the importance of proper data preprocessing when dealing with imbalanced educational datasets.

Overall, the results suggest that not only is dropout prediction feasible using SISU data, but that model performance can be substantially enhanced through appropriate preprocessing and model selection.

6. Conclusions

6.0.1 Findings

This study demonstrates that academic data alone, without the need for personal or demographic information, can provide strong predictive power for identifying students at risk of dropping out. However, careful selection and preprocessing of features is crucial. One of the key challenges encountered was data leakage which can occur easily in this type of dataset where many variables are dependent. Mitigating this risk is essential to ensure the validity of model performance.

One of the most important enhancements to model performance was the application of SMOTE. Prior to balancing, models tended to oversample the minority (drop outs). After SMOTE was applied, model performance improved across the board, particularly in terms of recall and precision. These improvements were clearly visible in performance metrics and through visualisations comparing results before and after balancing.

After applying proper balancing techniques, all evaluated models reached high levels of accuracy, making them viable candidates for deployment into institutional usage.

6.0.2 Further Research

There are several promising directions for further development and research. Refining the feature filtering and selection process would likely yield even better results and reduce the risk of overfitting or leakage. Advanced feature engineering could also help uncover more fine grained patterns associated with drop out behavior.

In the future, it would be valuable to explore the integration of the model into a user interface, such as a dashboard. Such an interface could include visualisations and explanatory plots (such as SHAP values) to help users interpret predictions and take informed action.

Implementing this would require model persistence, so that trained models can be saved and deployed.

Overall, the findings of this study lay a strong foundation for data-driven drop out prevention in higher education and open the door for practical tools to support students more effectively.

Dixi.

References

Mayra Albán and David Mauricio. Predicting University Dropout through Data Mining: A Systematic Literature. URL https://api.semanticscholar.org/CorpusID:134497957.

David Bañeres, M. Elena Rodríguez, Ana Elena Guerrero-Roldán, and Abdulkadir Karadeniz. An Early Warning System to Detect At-Risk Students in Online Higher Education. 10(13):4427. ISSN 2076-3417. doi: 10.3390/app10134427. URL https://www.mdpi.com/2076-3417/10/13/4427.

Norka Bedregal-Alpaca, Víctor Cornejo-Aparicio, Joshua Zárate-Valderrama, and Pedro Yanque-Churo. Classification Models for Determining Types of Academic Risk and Predicting Dropout in University Students. 11(1). ISSN 21565570, 2158107X. doi: 10.14569/IJACSA.2020.0110133. URL http://thesai.org/Publications/ViewPaper?Volume=11&Issue=1&Code=IJACSA&SerialNo=33.

Choong Hee Cho, Yang Woo Yu, and Hyeon Gyu Kim. A Study on Dropout Prediction for University Students Using Machine Learning. 13(21):12004. ISSN 2076-3417. doi: 10.3390/app132112004. URL https://www.mdpi.com/2076-3417/13/21/12004.

Cameron Cooper. Using Machine Learning to Identify At-risk Students in an Introductory Programming Course at a Two-year Public College. 02(03):407–421. ISSN 25829793. doi: 10.54364/AAIML.2022.1127. URL https://www.oajaiml.com/uploads/archivepdf/97051127.pdf.

Erwin deGelder. Matplot2tikz. URL https://github.com/ErwindeGelder/matplot2tikz.

Sari Fauzia Elza and Yohana Dewi Lulu Widyasari. PRISMA-Guided Systematic Review on Machine Learning for University Student Dropout Prediction. pages 377–385.

Jelte Fennema. PyLaTeX. URL https://jeltef.github.io/PyLaTeX/current/.

Antonio Jesus Fernandez-Garcia, Juan Carlos Preciado, Fran Melchor, Roberto Rodriguez-Echeverria, Jose Maria Conejero, and Fernando Sanchez-Figueroa. A Real-Life Machine Learning Experience for Predicting University Dropout at Different Stages Using Academic Data. 9:133076–133090. ISSN 2169-3536. doi: 10.1109/ACCESS.2021.3115851. URL https://ieeexplore.ieee.org/document/9548895/.

Funidata. Sisu, a. URL https://www.funidata.fi/en/services/sisu.

Funidata. About us, b. URL https://www.funidata.fi/en/about-us.

Hanken Svenska handelshögskolan. Eddie fernberg.

Thao-Trang Huynh-Cam, Long-Sheng Chen, and Tzu-Chuen Lu. Early prediction models and crucial factor extraction for first-year undergraduate student dropouts. 17 (2):624–639. ISSN 2050-7003. doi: 10.1108/JARHE-10-2023-0461. URL https://www.emerald.com/insight/content/doi/10.1108/JARHE-10-2023-0461/full/html.

Imbalanced-learn. SMOTE. URL https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html.

Jupyter Team. Project Jupyter Documentation. URL https://docs.jupyter.org/en/latest/.

Janka Kabathova and Martin Drlik. Towards Predicting Student's Dropout in University Courses Using Different Machine Learning Techniques. 11(7):3130. ISSN 2076-3417. doi: 10.3390/app11073130. URL https://www.mdpi.com/2076-3417/11/7/3130.

Sangyun Kim, Euteum Choi, Yong-Kee Jun, and Seongjin Lee. Student Dropout Prediction for University with High Precision and Recall. 13(10):6275. ISSN 2076-3417. doi: 10.3390/app13106275. URL https://www.mdpi.com/2076-3417/13/10/6275.

Sunbok Lee and Jae Young Chung. The Machine Learning-Based Dropout Early Warning System for Improving the Performance of Dropout Prediction. 9(15):3093. ISSN 2076-3417. doi: 10.3390/app9153093. URL https://www.mdpi.com/2076-3417/9/15/3093.

Wei-Meng Lee. *Python Machine Learning*. Wiley. ISBN 978-1-119-54569-9 978-1-119-55750-0 978-1-119-54567-5. doi: 10.1002/9781119557500.

Roderick Lottering, Robert Hans, and Manoj Lall. A Machine Learning Approach to Identifying Students at Risk of Dropout: A Case Study. 11(10). ISSN 21565570, 2158107X. doi: 10.14569/IJACSA.2020.0111052. URL http://thesai.org/Publications/ViewPaper?Volume=11&Issue=10&Code=IJACSA&SerialNo=52.

Azucena L. Jimenez Martinez, Kanika Sood, and Rakeshkumar Mahto. Early Detection of At-Risk Students Using Machine Learning. URL http://arxiv.org/abs/2412.09483.

Mónica V. Martins, Luís Baptista, Jorge Machado, and Valentim Realinho. Multi-Class Phased Prediction of Academic Performance and Dropout in Higher Education. 13(8): 4702. ISSN 2076-3417. doi: 10.3390/app13084702. URL https://www.mdpi.com/2076-3417/13/8/4702.

Matplotlib development team. Using Matplotlib. URL https://matplotlib.org/stable/users/index.

Ministry of Education and Culture. Steering, financing and agreements of higher education institutions, science agencies and research institutes. URL https://okm.fi/en/steering-financing-and-agreements.

Andreas Christian Müller and Sarah Guido. *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media, first edition edition. ISBN 978-1-4493-6941-5.

Diego Opazo, Sebastián Moreno, Eduardo Álvarez Miranda, and Jordi Pereira. Analysis of First-Year University Student Dropout through Machine Learning Models: A Comparison between Universities. 9(20):2599. ISSN 2227-7390. doi: 10.3390/math9202599. URL https://www.mdpi.com/2227-7390/9/20/2599.

Joonas Pesonen, Anna Fomkin, and Lauri Jokipii. Building Data Science Capabilities into University Data Warehouse to Predict Graduation. pages 156–158. URL http://www.eunis.org/eunis2018/.

Foster Provost and Tom Fawcett. *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking*. O'Reilly, first edition edition. ISBN 978-1-4493-6132-7 978-1-4493-7429-7.

scikit-learn. User Guide. URL https://scikit-learn.org/stable/user_guide.html.

Marina Segura, Jorge Mello, and Adolfo Hernández. Machine Learning Prediction of University Student Dropout: Does Preference Play a Key Role? 10(18):3359. ISSN 2227-7390. doi: 10.3390/math10183359. URL https://www.mdpi.com/2227-7390/10/18/3359.

Michael Waskom. An introduction to seaborn. URL https://seaborn.pydata.org/tutorial/introduction.html.

Appendix A

Models used by previous studies. Listed in the order of best accuracy per study.

Authors	**Models**	**Accuracy**
Huynh-Cam, Chen, Lu	**Logistic regression**	**97.46**
	MLP	97.03
	Decision trees	95.8
Jimenez Martinez, Sood, Mahto	**Random forest**	**98.0**
	Decision trees	97.0
	SVM	94.0
	Naive Bayes	92.0
	KNN	92.0
	Logistic regression	88.0
Cooper	**Probalilistic neural network**	**91.3**
	Regression MLP	90.82
	Classification MLP	89.86
	Reg Gen Feedforward	89.86
	Classification SVM	87.44
	Logistic regression	86.96
	Reg MLP with PCA	85.99
	Radial Basis Function	85.51
	Class MLP with PCA	85.02
	Time-Lag Recurrent Network	84.06
	Linear regression	83.09
	Time-Delay Network	83.09
Fernández-Garcia, Preciado, Melchor, Rodriguez-Echeverria, Conejero, Sánchez-Figueroa	**SVM**	**80.89**
	Random forest	80.53
	Ensemble	80.39
	Gradient boosting	79.93

Authors	Models	Accuracy
Lee, Chung	**Boosted decision tree with SMOTE**	**99.1**
	Boosted decision tree	98.8
	Random forest	98.6
	Random forest with SMOTE	98.6
Bañeres, Rodríguez, Guerrero-Roldán, Karadeniz	Naive Bayes	*nan*
	Decision trees	*nan*
	KNN	*nan*
	SVM	*nan*
Kabathova, Drlik	**Logistic regression**	**93.0**
	Random forest	93.0
	SVM	92.0
	Decision tree	90.0
	Neural network	88.0
	Naive Bayes	77.0
Martins, Baptista, Machado, Realinho	**Random forest**	**71.13**
	Balanced random forest classifier	69.77
	Easy ensemble classifier	68.63
	RusBoost algorithm	64.77
Kim, Choi, Jun, Lee	Logistic regression + SMOTE	*nan*
	ANN + RandomOverSampler	*nan*
	Gradient boosting + SMOTETOMEK	*nan*
	Ensemble + SMOTETOMEK	*nan*
Cho, Yu, Kim	**LightGBM**	**95.5**
	Random forest	95.3

Authors	**Models**	**Accuracy**
	Decision tree	94.7
	Deep neural network	94.7
	SVM	94.2
	Linear regression	92.7
Opazo, Moreno, Álvarez-Miranda, Pereria	**Random forest**	**69.0**
	Gradient boosting decision tree	69.0
	Decision trees	68.0
	Naive Bayes	66.0
	Neural networks	66.0
	SVM	65.0
	Logistic regression	62.0
	KNN	62.0
Segura, Mello, Hernández	**KNN**	**85.59**
	ANN	85.11
	Logistic regression	83.37
	SVM	82.9
	Decision trees	82.05
Bedregal-Alpaca, Cornejo-Aparicio, Zárate-Valderrama, Yanque-Churo	**ANN**	**73.59**
	Decision trees	72.74
	Iterative dichotomizer 3	*nan*
	Hunt C4.5	*nan*
Lottering, Hans, Lall	**SVM**	**89.0**
	Logistic regression	88.0
	Decision trees	85.0
	KNN	84.0
	Naive Bayes	81.0
Pesonen, Fomkin, Jokipii	Logistic regression	*nan*
	Multiple linear regression	*nan*

Appendix B

SQL file for selecting the columns from the database.

```
WITH codebook (urn, name_en, name_fi) AS (
SELECT [urn] COLLATE Finnish_Swedish_CI_AS
  ,LOWER([name_en]) COLLATE Finnish_Swedish_CI_AS
  ,LOWER([name_fi]) COLLATE Finnish_Swedish_CI_AS
FROM [SA].[sa].[sisu_codebooks_codes]
)

SELECT
  sr.[studyRightId]
  ,c1.name_en as gender
  ,CAST(sr.[is_finnish] as bit) as is_finnish
  ,CAST(sr.[class] as int) as class
  ,CAST(sr.[admission_year] as int) as admission_year
  ,CAST(IIF(sr.[class]=sr.[admission_year], 1, 0) as bit) as
    class_admission_same
  ,sr.[admissiontype_en]
  ,CAST(sr.[latestterm_studyyearstartyear] as int) as
    latestterm_studyyearstartyear
  ,sr.[studyrightState_20_9]
  ,sr.[educationLocation_sv]
  ,CAST(IIF(LEFT(sr.[studyright_state_dropOut_en],1)='3',1,0) as bit)
    as studyright_state_dropOut
  ,sr.[studyright_state_dropOut_integrated_bsc_en]
  ,sr.[studyright_state_dropOut_integrated_msc_en]
  ,CAST(sr.[phases] as int) as phases
  ,sr.[current_phase_209_en]
  ,sr.[current_phase_209_order]
  ,sr.[degreeProgramme_phase1_en]
  ,sr.[degreeProgramme_phase2_en]
  ,sr.[degreeProgramme_current_phase_en]
  ,sr.[major_phase1_en]
  ,sr.[major_phase2_en]
  ,sr.[major_current_en]
  ,sr.[Person_DegreeAttainment_bachelor]
  ,sr.[Person_DegreeAttainment_masters]
```

```
  ,sr.[Person_DegreeAttainment_doctor]
  ,cs.[terms_attending]
  ,cs.[terms_nonattending]
  ,cs.[terms_nonattending_statutoryabsence]
  ,cs.[terms_nonattending_total]
  ,cs.[termregistrations_total]
  ,cs.[terms_attending_inc_future]
  ,cs.[terms_nonattending_inc_future]
  ,cs.[terms_nonattending_statutoryabsence_inc_future]
  ,cs.[terms_nonattending_total_inc_future]
  ,cs.[termregistrations_total_inc_future]
  ,cs.[terms_attending_winter]
  ,cs.[terms_nonattending_winter]
  ,cs.[terms_nonattending_statutoryabsence_winter]
  ,cs.[terms_nonattending_total_winter]
  ,cs.[termregistrations_total_winter]
  ,cs.[courseunit_count]
  ,cs.[credits_sum]
  ,cs.[studyweeks_sum]
  ,cs.[weighted_average]
  ,cs.[courseunit_count_spring]
  ,cs.[credits_sum_spring]
  ,cs.[studyweeks_sum_spring]
  ,cs.[weighted_average_spring]
  ,cs.[courseunit_count_winter]
  ,cs.[credits_sum_winter]
  ,cs.[studyweeks_sum_winter]
  ,cs.[weighted_average_winter]
  ,cs.[creditsPerAttending]
  ,cs.[creditsPerAttending_spring]
  ,cs.[creditsPerAttending_winter]
  ,cs.[graduation_standard_period_education_en]
  ,cs.[standard_period_bsc_en]
  ,cs.[standard_period_msc_en]
  ,cs.[standard_period_phd_en]
  ,cs.[degreeAttainments_bsc]
  ,cs.[degreeAttainments_msc]
  ,cs.[degreeAttainments_phd]
```

```
71 FROM [DW].[REPORT].[f_t_StudyRights_all] sr
72 LEFT JOIN [DW].[REPORT].[calc_t_studyright] cs ON sr.[studyRightId] =
      cs.[studyRightId]
73 LEFT JOIN codebook c1 ON sr.[genderUrn] = c1.urn
74 WHERE
75   sr.[class] = 2024
76   AND sr.[studyrightState_20_9] <> 'not started'
```

Listing 6: SQL SELECT statements

Appendix C

Columns and data types in the dataset (after filtering and dropping columns).

Column	Data type
studyrightid	object
gender	object
is_finnish	bool
class_admission_same	bool
admissiontype_en	object
studyright_state_dropout	bool
phases	int64
current_phase_209_en	object
current_phase_209_order	float64
degreeprogramme_phase1_en	object
degreeprogramme_phase2_en	object
degreeprogramme_current_phase_en	object
major_phase1_en	object
major_phase2_en	object
major_current_en	object
terms_attending	float64
terms_nonattending	float64
terms_nonattending_statutoryabsence	float64
terms_nonattending_total	float64
termregistrations_total	float64
courseunit_count	float64
credits_sum	float64
studyweeks_sum	float64
weighted_average	float64
creditsperattending	float64