

CS773 - Data Mining and Security

Course Project Report

Analyzing Open University Learning Analytics Dataset

Anuradha Mantena (01167161)
Triveni Sangama Saraswathi Edla (01160800)

Contents

1. Executive Summary.....	3
2. Introduction	3
3. Problem Statement	4
4. Solution Methodology.....	5
4.1. Data cleaning and Combining.....	6
4.2. Exploratory Data Analysis (EDA)	6
4.3. Predictive modeling.....	6
4.3.1. Decision Tree	7
4.3.2. Random Forest.....	7
4.3.3. Gaussian Naïve Bayes	7
4.3.4. Gradient Boosting	7
4.3.5. Performance Metrics	8
4.3.5.1 Confusion Matrix.....	8
4.3.5.3 Precision.....	8
4.3.5.4 Recall.....	9
4.3.5.5 F-score	9
4.3.5.6 Accuracy	9
4.3.6. <i>k</i>-Fold cross-validation	9
5. Experimental setup and data used	10
5.1. Exploratory Data Analysis (EDA) Using Python:	11
6. Results.....	14
6.1. Modeling using only Biographic data.....	14
6.2. Modeling using Biographic and VLE data	15
6.3. Modeling using only Biographic, VLE and assessment data	16
7. Conclusions.....	20

1. Executive Summary

Online education has become very popular in recent years with many companies starting their courses by collaborating with top universities in the world. According to the national center for education statistics, over 5 million students are currently enrolled in distance education courses. Additionally, with the new social distance norms due to the current outbreak of the Covid19 pandemic throughout the world and online learning would become much more popular. If the organization can predict the student's final grade ahead of time, the students at risk can be well informed and can improve their performance. In this project, the data provided by the Open University (OU), one of the largest distance learning universities in the United Kingdom (UK), is used to demonstrate the use of machine learning algorithms to predict the students at risk.

The open university (OU) provides biographic information, virtual learning environment (VLE) data, and assessments of the students in different files. The data from the different files are cleaned, combined, and preprocessed to find the important attributes that help in building a predictive model with good performance. Then the data is explored using exploratory data analysis (EDA). Four different classification algorithms such as Decision Tree, Random Forest, Gaussian Naïve Bayes, and Gradient boosting algorithms are used to predict the results of the students. Initially, the classification algorithms are trained using only biographic information, then the information of VLE and assessments is added to find the extent of improvement in the prediction performance. The Random Forest Algorithm returned the best accuracy of prediction. Further, the various trials were conducted to improve the accuracy of the prediction model.

2. Introduction

Online education enables the students at any location on the globe to credit the courses of interest from top universities for an affordable price. The organizations offering the online course must consider many factors to make the learning process effective along with maintaining the standards of the courses. The online course organizations must be able to predict the outcome of the student so that the student at risk will be well informed about the final grade and improve the performance. Open University (OU) is one of the largest distance learning universities in the United Kingdom (UK). It offers various courses for undergraduate and graduate students and is vastly popular among the students who cannot be on campus for various reasons and no prior education is required for the students to enroll in OU. It has more than 250,000 students enrolled making it the largest

academic institution in the country. With the advent of the internet, it has become possible and popular for distance learning universities to provide course materials online in different available formats. Students can access these study materials anywhere and even give exams online. Universities can capture and record the way students interact with the learning material.

OU provides the biographic information, that gives their socio-economic conditions and information interaction with Virtual Learning Environment (VLE), provides the proactiveness of the students. As the course progress, the student assessments provide more information of student performance. Virtual Learning Environment (VLE) provided by the open university has several factors that affect the student's performance. Many students end up failing courses or withdrawing and early detection of students at risk of failure could help the students in taking a timely action of improving their performance. If the impact on student's performance is identified correctly, better results can be obtained. Both students and faculty can be well informed about the progress based on the analysis which provides an opportunity to excel. Such data can provide useful and actionable insights into students' learning behavior, which universities can use to improve student performance by providing them with additional help wherever necessary.

The report is organized as follows. In Sec. 3, the problem statement of the project is stated. In Sec. 4 solution methodology is discussed such as data cleaning, exploratory data analysis (EDA), different machine learning algorithms used for predictive modeling and performance metrics.

3. Problem Statement

The student data for the 7 courses (called a module) with multiple offerings are provided by OU as shown in Fig. 1. The data contains the demographic data, the VLE interaction data, and the assessment data. Dataset is stored in several CSV files and it mainly contains about courses, students, and their interactions with Virtual Learning Environment (VLE) for the respective courses. Presentations of courses start in February and October - they are marked by 'B' and 'J' respectively. The dataset consists of tables connected using unique identifiers. The goal is to combine all the information from different files and build a predictive model that can classify the result of the student accurately.

Course	2013-Offering 1	2013-Offering 2	2014-Offering 1	2014-Offering 2
AAA		2013J		2014J
BBB	2013B	2013J	2014B	2014J
CCC			2014B	2014J
DDD	2013B	2013J	2014B	2014J
EEE		2013J	2014B	2014J
FFF	2013B	2013J	2014B	2014J
GGG		2013J	2014B	2014J

Fig. 1 Course modules and presentations

4. Solution Methodology

The student data is stored in seven different CSV files, connected through unique identifiers as shown in Fig. 2.

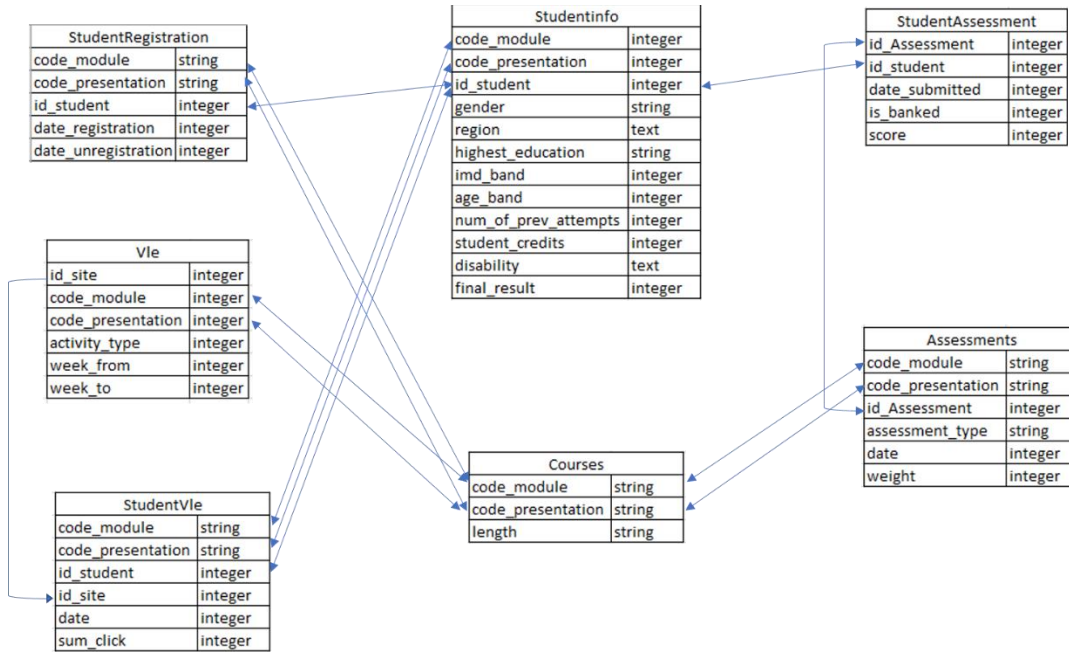


Fig. 2 Data design between different excel files

The data available in the seven CSV files is very large with many attributes and to develop the predictive model, the solution methodology is divided into different stages: 1) Data cleaning and combining 2) Exploratory Data Analysis and 3) Predictive Modeling

4.1. Data cleaning and Combining

There are missing data for some of the attributes in the CSV files and they are replaced by appropriate values. The Studentinfo.csv, StudentRegistration.csv files contain the biographic information of the students and only Studentinfo.csv file contains the final result of each student. Vle.csv, StudentVle.csv have different activity types of VLE and the number of times the student has accessed the resources on each day. A python script is written to find the sum of clicks by each student on the VLE resources at a different percentage of the course length, namely at 20%, 40%, 60%, 80%, and 100%. This information is added as an attribute to each student's data. Assesments.csv and StudentAssesment.csv files have assessment types and scores of each student and different types of scores are calculated for students at different percentage lengths of time.

4.2. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is used to analyze data sets to visualize the characteristics of the data, hypothesis task, patterns in the data, and significance of the attributes towards the results of students. In EDA, basic biographic attributes, VLE attributes, and the assessment attributes are visualized against the final result. Python is used for this purpose.

4.3. Predictive modeling

The information explored using EDA is used to decide on the attributes that needed to be considered for modeling. Four different machine learning algorithms: Decision Tree, Random Forest, Gaussian NB, and gradient boosting were used in this project. Machine learning models are trained to recognize certain types of patterns. Machine learning algorithms build a mathematical model based on sample data, known as training data in order, to make predictions or decisions. To implement a machine learning model, python is one of the best programming languages because of its simplicity and consistency. Python has access to great libraries and frameworks for machine learning models. For this project, we have first used python and towards the end, the results were compared with Weka. Weka is a tried and tested open-source machine learning software that can be accessed through a graphical user interface, standard terminal applications, or Java API. The benefit of using the Weka platform is a large number of supported machine learning algorithms.

4.3.1. Decision Tree

A decision tree is a flow chart like structure in which each internal node represents an attribute, each branch represents the outcome of the test, and each leaf node represents a decision taken after computing all attributes. A significant advantage of a decision tree is that it creates a comprehensive analysis of the consequences along each branch and identifies decision nodes that need for further analysis. It is used for both regression and classifier problems.

4.3.2. Random Forest

Random forest is a Supervised Learning algorithm which uses an ensemble learning method for both classification and regression. Random forest classifier consists of a large number of individual decision trees that operate as an ensemble. Each tree in the random forest spits out a class prediction and the class with the most votes becomes the model's prediction. It is one of the best-supervised learning algorithms with good predictive accuracy. The main objective of the Random Forest algorithm is to take a set of high variance, low-bias of decision trees, and transform them into a model that has both low variance and low bias. By aggregating the various outputs of individual decision trees, random forests reduce the variance that can cause errors in decision trees.

4.3.3. Gaussian Naïve Bayes

Gaussian Naive Bayes methods are the supervised learning algorithms based on applying Bayes theorem with the naive assumption of conditional independence between every pair of features given the value of the class variable. The advantage of Gaussian Naive Bayes is that it doesn't need much training data and it is applied for both regression and classifier problems. The main objective of Gaussian Naive Bayes is to determine the likelihood of an event A happening given B happens.

4.3.4. Gradient Boosting

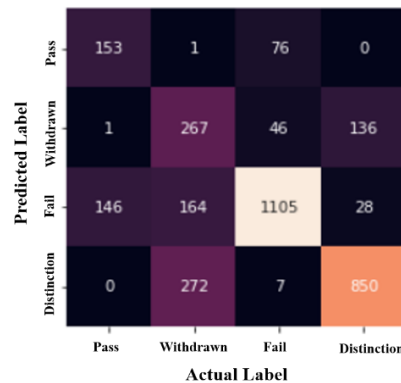
Gradient boosting is a machine learning technique that uses an ensemble learning method for both classification and regression. Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models to create a strong predictive model. The objective of Gradient Boosting classifiers is to minimize the loss or the difference between the actual class value of the training example and the predicted class value.

4.3.5. Performance Metrics

Performance metrics are used to evaluate different machine learning algorithms. Below are the various performance metrics that can be used to evaluate predictions for classification problems.

4.3.5.1 Confusion Matrix

It is the easiest way to measure the performance of a classification problem where the output can be of two or more types of classes. A confusion matrix is a table with two dimensions viz. “Actual” and “Predicted” and both the dimensions have “True Positives (TP)”, “True Negatives (TN)”, “False Positives (FP)”, “False Negatives (FN)”. A typical confusion matrix for multiclass classifiers look as shown in the below figure.



Predicted Label	Pass	Withdrawn	Fail	Distinction
Pass	153	1	76	0
Withdrawn	1	267	46	136
Fail	146	164	1105	28
Distinction	0	272	7	850
		Actual Label		

Fig. 3. Confusion matrix for multiclass classifier

4.3.5.3 Precision

Precision is the fraction of relevant instances among the retrieved instances. In simple words, it is defined as the number of true positives divided by the number of true positives plus the number of false positives.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

4.3.5.4 Recall

Recall is the number of correct results divided by the number of results that should have been returned. In binary classification, recall is also called as sensitivity. It can be viewed as the probability that a relevant document is retrieved by the query.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

4.3.5.5 F-score

The F-score is defined as the weighted harmonic mean of the test's precision and recall. This score is calculated according to the precision and recall of a test considered. In simple words, F-score is defined as two times the precision and recall divided by the sum of precision and recall.

$$\text{F-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

4.3.5.6 Accuracy

Accuracy is one metric for evaluating classification models. Accuracy is the ratio of the number of correct predictions to the total number of input samples. Below is the definition of accuracy

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

4.3.6. *k*-Fold cross-validation

k-fold cross-validation is a standard evaluation technique. It is a systematic way of running repeated percentage splits. Divide a data set into *k* pieces then hold each piece in turn for testing and remaining *k*-1 for training together. The average of the test results from *k* trained models is taken as the final output. The advantage of *k*-fold cross-validation is that every data point is tested exactly once and is used in training *k*-1 times. The variance will be reduced as the value of *k* increases. In this project, *k*=10 is considered

5. Experimental setup and data used

Below shows the number of data points in each CSV files corresponding to the Fig. 2.

```
Number of data points in courses.csv: 22
Number of data points in student studentInfo.csv: 32593
Number of data points in student studentRegistration.csv 32593

Number of data points in vle.csv 6364
Number of data points in studentVle.csv 10655280

Number of data points in assessments.csv: 206
Number of data points in student studentAssessment.csv: 173912
```

There are a total of 22 courses. studentInfo.csv, studentRegistration.csv files contain the biographic and registration information of each student, which shows that there are a total of 32593 number of students. vle.csv contains information about VLE activity types and studentVle.csv contains all records of each student on the VLE. There are a total of 6264 activity types and 10655280 data points, much higher than the total number of students. The reason for this is because studentVle.csv file records the daily clicks of each student's activity on VLE as a different instance. Similarly, studentAssessment.csv data points are higher than the total number of students because the scores of a student in different assessments are recorded separately in different rows.

To combine the VLE and assessment data with the biographic data, a python script is used, and the total data of each student is combined into a single CSV file. As the assessments and VLE are distributed over the entire length of the course, all the activities are summed up at different percentages of the course lengths (20%-100%). From the VLE data, the total clicks and mean clicks are calculated and added as an attribute for each student and saved to vle_combined.csv. In case for assessment data, weighted cumulative score (CS), percentage weighted cumulative score (PCS), the number of late submissions (LS) and average raw scores (RS) is calculated for each student at different percentage length of the course (indicated by adding percent to attribute name, eg: CS80 at 80% of the length) and stored to students_asses_combined.csv.

5.1.Exploratory Data Analysis (EDA) Using Python:

This section shows the results of EDA followed by the modeling results in python and then verification using Weka. Fig. 4a shows that there are four classes in the data: Pass, Withdrawn, Fail, and Distinction. The plot shows that count of students in each class and it shows that the total number of students with pass and distinction combined is more than other classes together. However, about 10000 out of 32593, are withdrawing from the course and a significant number of students are failing. A good classification algorithm is needed to predict different classes with good accuracy.

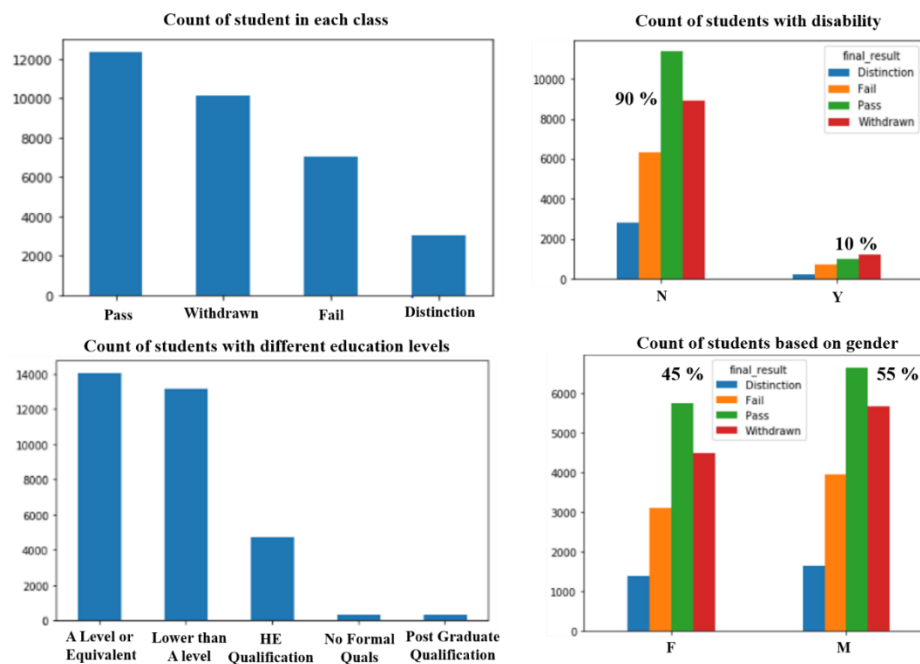


Fig. 4 Count of (a) Students in each class (b) Disability students in each class (c) Education levels of students and (d) Gender in each class

Fig. 4b shows that there is a small percentage of students with disabilities. Among the disabled students, a greater number of people are withdrawing from the course compared to other classes. Proper follow up with the withdrawn disabled students would help in understanding their issues and modifying the course structure that could better help them. Fig. 4c shows the number of students in different education levels and it is observed that the maximum number of people are from A Level or equivalent followed by lower than A level and least is by students with

postgraduates. Fig. 4d shows that the percentage of male students is slightly higher than females and the percentage of students in each class among male and female categories is almost the same.

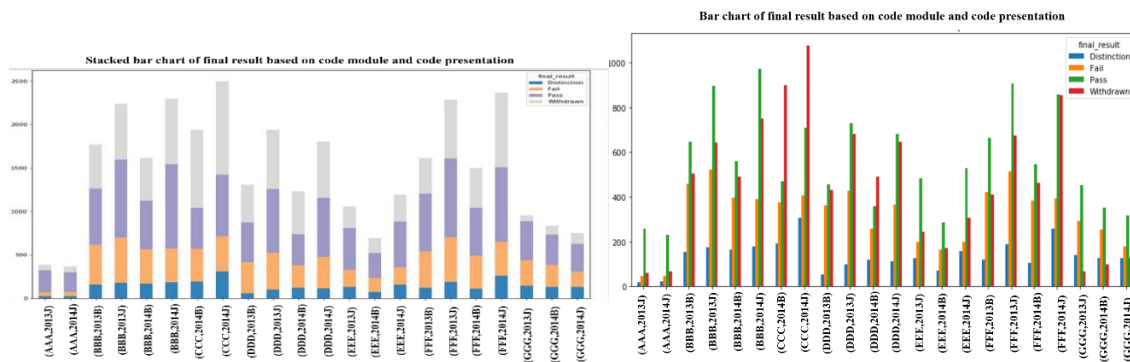


Fig. 5 Stacked bar and bar chart for results grouped by course module and course presentation

Fig. 5 Stacked bar and bar chart for results grouped by course module and course presentations shows the bar charts of final results grouped by code module and code presentation. The plot shows that course AAA has the least number of students registered compared to other courses with similar numbers in 2013 and 2014. Course BBB is offered in all the terms for both years and a greater number of students are registered in the spring term compared to autumn term with a similar percentage of students withdrawing and failing. For course CCC, a greater number of people are withdrawing the course and special focus must be given for changing the course structure. The course DDD and FFF have seen an increase in enrollments in spring compared to autumn. The total number of enrollments for course EEE decreased in 2014B and increased in the following terms. For course FFF, there is continuous decreases in enrollment of students, and a smaller percentage of students are failing.

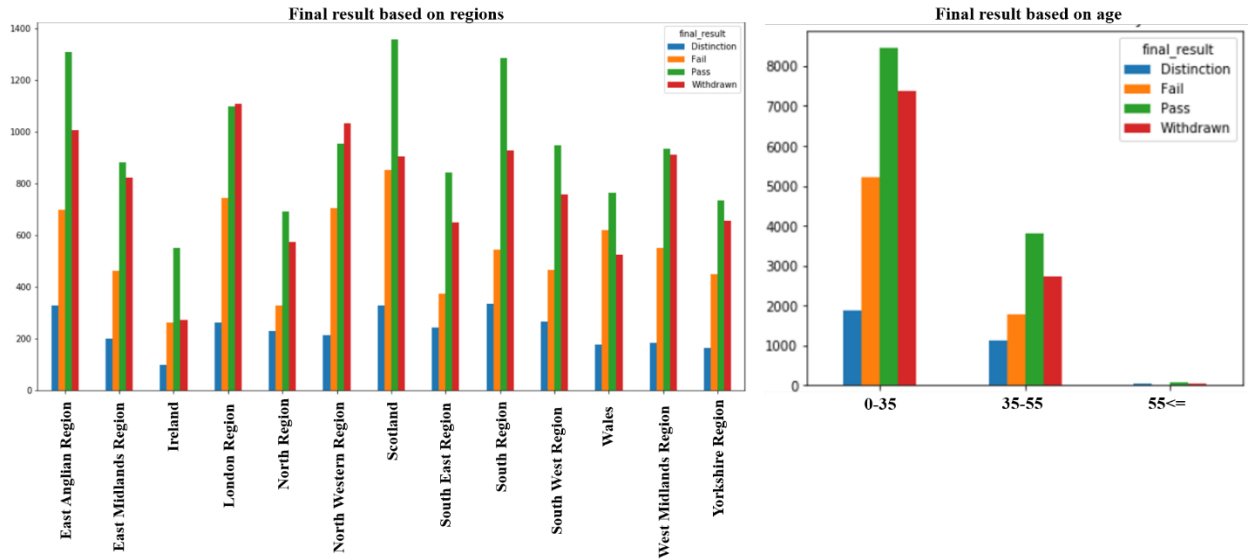


Fig. 6 Bar chart based on the region and age

Fig. 6 shows the bar chart of students in different regions and different age bands. It shows that students from Scotland region are passing more compared to other regions. The age plot shows that there are a greater number of people in the 0-35 age band compared to others and the percentage of classes in each band is similar.

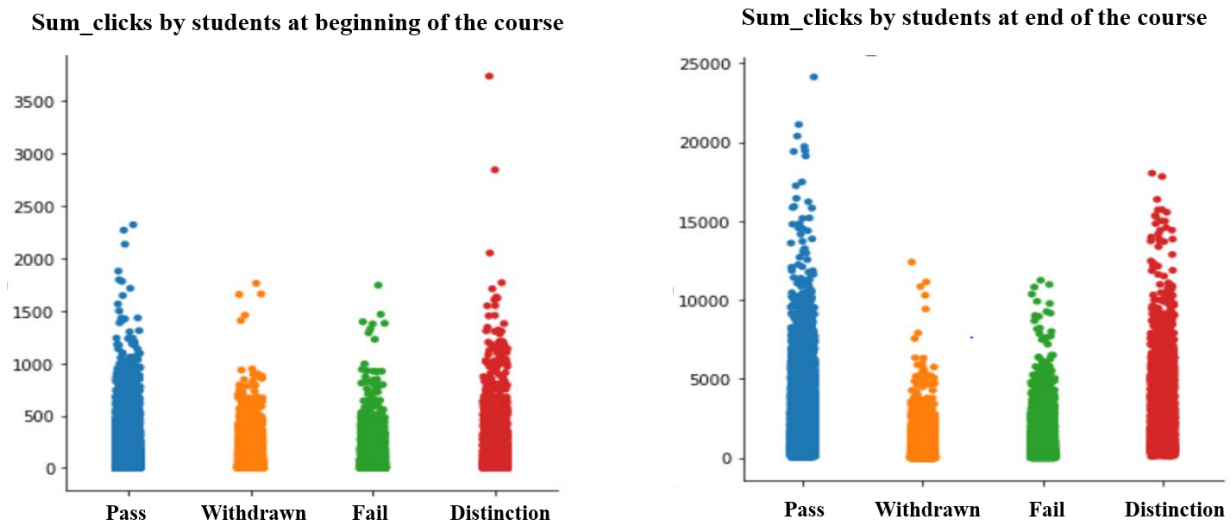


Fig. 7 sum of clicks by each student before the beginning of the course and at the end

Fig. 7 shows the total number of daily clicks (sum_clicks) by students categorized into different classes. Each marker indicates one instance. At the beginning of the course, the students in each

class access the VLE at a similar frequency, and people who access at a greater number of times, have a more chance of getting a distinction. The sum clicks plot towards the end of the course indicates that students who access the VLE resources more have higher chances of passing and distinctions. However, the sum_clicks of withdrawn may skew the machine learning results as there may not be enough information because of early withdrawal from the courses. The insights from the EDA will be used in the machine learning algorithm.

6. Results

To predict the result of students, predictive modeling is built with different classifiers available in Scikit-learn packages in python. Four types of classifying algorithms are used to predict the result of the students. Modeling is conducted systematically starting from using only biographic data and then using more information.

6.1. Modeling using only Biographic data

Fig. 8 shows the correlation heatmap and performance table with only biographic information. Correlation map shows that the final_result has no strong correlation coefficients with any of the attributes and hence all the attributes are considered for modeling. The Gradient Boosting algorithm is providing the best performance among the used classifiers with an accuracy of 44.9%. However, precision, recall, and F-score are poor for Fail class. The goal here is to provide an accurate prediction for the students who may fail so that they will be informed ahead.



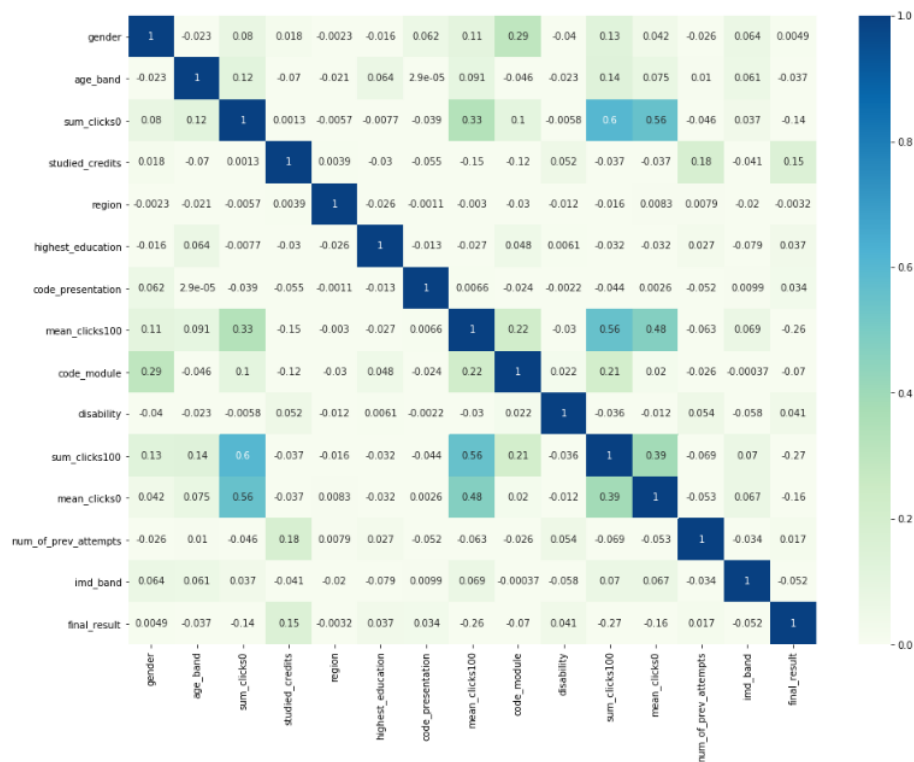
Precision	DecisionTree	RandomForest	GaussianNB	GradientBoosting
Distinction	0.216482	0.154512	0.010566	0.008902
Fail	0.283248	0.245053	0.107505	0.110155
Pass	0.351502	0.448739	0.785669	0.725746
Withdrawn	0.369449	0.427429	0.289556	0.479894
Averaged	0.322062	0.376393	0.636528	0.607627
F-score	DecisionTree	RandomForest	GaussianNB	GradientBoosting
Distinction	0.173862	0.159733	0.020156	0.017334
Fail	0.267757	0.256793	0.165291	0.171913
Pass	0.383415	0.443024	0.539528	0.553718
Withdrawn	0.373558	0.416318	0.348422	0.471670
Averaged	0.323620	0.373116	0.473893	0.502852

Recall	DecisionTree	RandomForest	GaussianNB	GradientBoosting
Distinction	0.145658	0.166120	0.235141	0.342262
Fail	0.254065	0.269870	0.358473	0.391796
Pass	0.421873	0.437744	0.410905	0.447745
Withdrawn	0.378102	0.406089	0.438216	0.464161
Averaged	0.329825	0.370662	0.412389	0.449329
Accuracy	DecisionTree	RandomForest	GaussianNB	GradientBoosting
Distinction	0.145584	0.165488	0.226950	0.341772
Fail	0.253974	0.269621	0.359035	0.391633
Pass	0.421942	0.437648	0.410908	0.447716
Withdrawn	0.377973	0.406139	0.438330	0.464126
Averaged	0.329825	0.370662	0.412389	0.449330

Fig. 8 Performance table for 10-fold cross-validation using only the biographic information

6.2. Modeling using Biographic and VLE data

To improve the performance of prediction, the information of VLE is considered along with biographic information. Fig. 9 shows the performance of prediction. It is observed that the performance has been improved significantly with the highest accuracy of 64.5%. But the performance of Fail class is still poor with a precision of 0.28 and an F-score of 0.35.

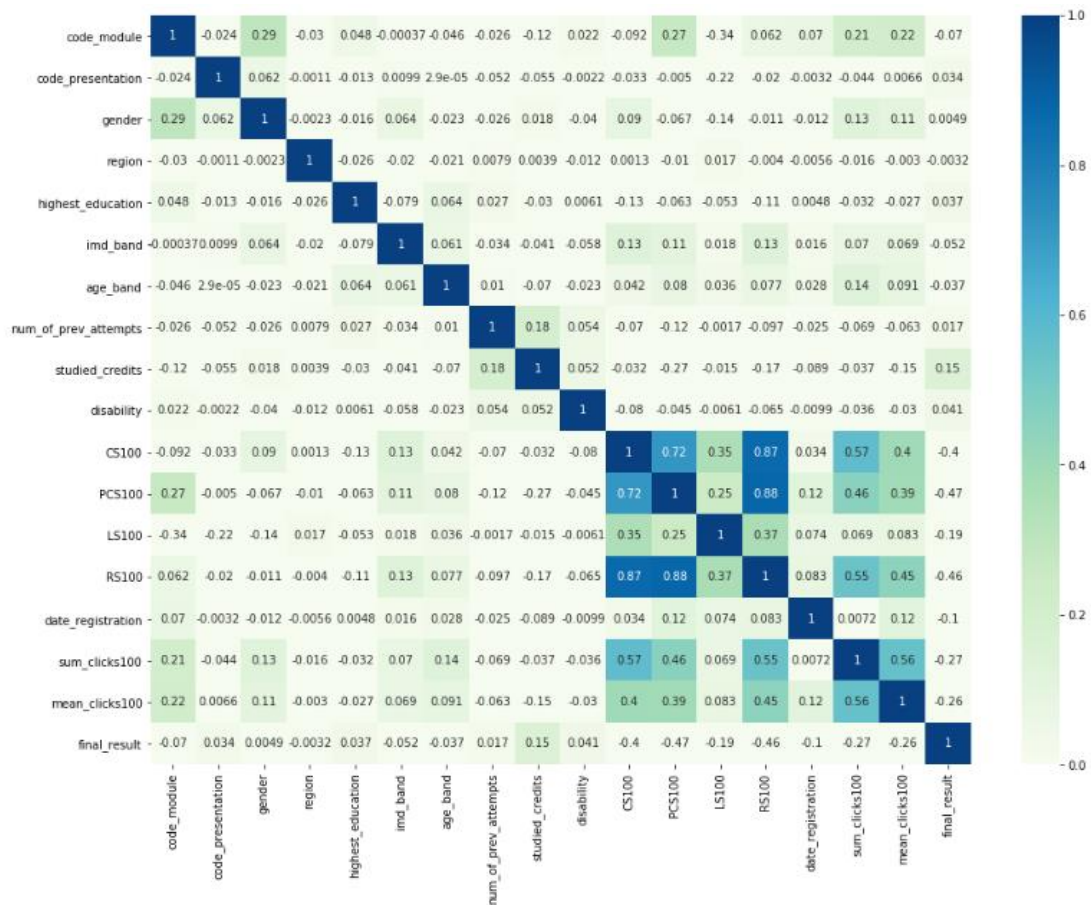


Precision	DecisionTree	RandomForest	GaussianNB	GradientBoosting	Recall	DecisionTree	RandomForest	GaussianNB	GradientBoosting
Distinction	0.242373	0.059332	0.158500	0.059647	Distinction	0.244210	0.476670	0.282134	0.490501
Fail	0.341842	0.276324	0.158956	0.286284	Fail	0.335931	0.484482	0.319242	0.521508
Pass	0.614164	0.908927	0.537959	0.914124	Pass	0.616812	0.626701	0.584325	0.631671
Withdrawn	0.627302	0.721674	0.784633	0.741946	Withdrawn	0.630733	0.713850	0.498441	0.719743
Averaged	0.524279	0.762145	0.598519	0.774814	Averaged	0.524897	0.634799	0.497622	0.645384
F-score	DecisionTree	RandomForest	GaussianNB	GradientBoosting	Accuracy	DecisionTree	RandomForest	GaussianNB	GradientBoosting
Distinction	0.243105	0.105272	0.202347	0.106082	Distinction	0.244097	0.471053	0.281030	0.500000
Fail	0.338674	0.351778	0.211935	0.369530	Fail	0.336027	0.484826	0.319203	0.521436
Pass	0.615393	0.741800	0.560017	0.747031	Pass	0.616734	0.626687	0.584307	0.631688
Withdrawn	0.628912	0.717669	0.609491	0.730624	Withdrawn	0.630818	0.713743	0.498280	0.719675
Averaged	0.524455	0.678757	0.528138	0.689856	Averaged	0.524898	0.634799	0.497622	0.645384

Fig. 9 Performance table for 10-fold cross-validation using biographic and VLE information

6.3. Modeling using only Biographic, VLE and assessment data

In the next step, the student assessment information is included, and Fig. 10 shows the results. The overall performance as improved slightly with an accuracy of 73% but the performance for Fail class is still bad with 60% accuracy and poor F-score and precision .



Precision	DecisionTree	RandomForest	GaussianNB	GradientBoosting	Recall	DecisionTree	RandomForest	GaussianNB	GradientBoosting
Distinction	0.475588	0.458262	0.593252	0.510194	Distinction	0.488491	0.654077	0.445188	0.666348
Fail	0.416368	0.367643	0.337920	0.372374	Fail	0.411988	0.577381	0.460277	0.591410
Pass	0.746477	0.902445	0.726233	0.895247	Pass	0.744654	0.758472	0.763810	0.764278
Withdrawn	0.705432	0.826792	0.830529	0.839706	Withdrawn	0.707425	0.745188	0.724878	0.750773
Averaged	0.637209	0.773855	0.685650	0.777496	Averaged	0.637100	0.721903	0.662443	0.729052
F-score	DecisionTree	RandomForest	GaussianNB	GradientBoosting	Accuracy	DecisionTree	RandomForest	GaussianNB	GradientBoosting
Distinction	0.481528	0.537998	0.508196	0.577345	Distinction	0.488459	0.651929	0.445299	0.665947
Fail	0.413997	0.448905	0.389671	0.456853	Fail	0.411781	0.576992	0.460959	0.591800
Pass	0.745461	0.824147	0.744500	0.824518	Pass	0.744633	0.758466	0.763870	0.764263
Withdrawn	0.706368	0.783780	0.774095	0.792693	Withdrawn	0.707416	0.745185	0.724820	0.750682
Averaged	0.637010	0.739896	0.669700	0.745897	Averaged	0.637100	0.721903	0.662443	0.729052

Fig. 10 Performance table for 10-fold cross-validation using biographic, VLE and assessment information

Combining distinction and pass classes

Different combinations of trials were conducted to improve the performance of Fail class. There are two different classes, distinction, and pass, which can be combined into one class. Fig. 11

shows the performance after combining pass and distinction. The overall accuracy improved slightly but the performance of Fail class did not improve significantly.

Precision	DecisionTree	RandomForest	GaussianNB	GradientBoosting	Recall	DecisionTree	RandomForest	GaussianNB	GradientBoosting
Fail	0.406044	0.360720	0.258389	0.361684	Fail	0.398647	0.566542	0.421735	0.584361
Pass	0.867564	0.964441	0.916249	0.960663	Pass	0.876304	0.876337	0.882584	0.875852
Withdrawn	0.694097	0.816741	0.842964	0.832312	Withdrawn	0.692628	0.742577	0.695805	0.744525
Averaged	0.711810	0.830759	0.801588	0.835908	Averaged	0.713650	0.787807	0.751081	0.791090
Recall	DecisionTree	RandomForest	GaussianNB	GradientBoosting	Accuracy	DecisionTree	RandomForest	GaussianNB	GradientBoosting
F-score	DecisionTree	RandomForest	GaussianNB	GradientBoosting	Fail	0.398608	0.566563	0.422614	0.584651
Fail	0.402228	0.440631	0.320270	0.446595	Pass	0.876313	0.876329	0.882544	0.875852
Pass	0.871880	0.918271	0.899079	0.916285	Withdrawn	0.692610	0.742591	0.695652	0.744473
Withdrawn	0.693286	0.777856	0.762329	0.785925	Averaged	0.713650	0.787807	0.751082	0.791090
Averaged	0.712654	0.804418	0.770899	0.808097					

Fig. 11 Performance table for 10-fold cross-validation using all student information and combining pass and distinction classes.

Removing withdrawn class instances

It was observed that the data of the students, who have withdrawn, is skewing the results. This is because the students have dropped from the class at different periods of the course length. Fig. 12 shows the results after dropping the data of students who have withdrawn. It shows that the performance of all classification algorithms improved and Fail class as accuracy of 90%. The Remaining performance metrics also improved significantly.

Precision	DecisionTree	RandomForest	GaussianNB	GradientBoosting	Recall	DecisionTree	RandomForest	GaussianNB	GradientBoosting
Fail	0.761266	0.756519	0.779373	0.753529	Fail	0.760586	0.906184	0.810935	0.907438
Pass	0.890316	0.964175	0.916823	0.964826	Pass	0.890360	0.896053	0.900535	0.894979
Averaged	0.849848	0.909723	0.875428	0.909715	Averaged	0.849535	0.898739	0.873468	0.898249
F-score	DecisionTree	RandomForest	GaussianNB	GradientBoosting	Accuracy	DecisionTree	RandomForest	GaussianNB	GradientBoosting
Fail	0.760557	0.824402	0.794526	0.823133	Fail	0.760561	0.906186	0.810923	0.907398
Pass	0.890247	0.928813	0.908531	0.928537	Pass	0.890333	0.896091	0.900524	0.895019
Averaged	0.849512	0.901449	0.874156	0.901056	Averaged	0.849534	0.898739	0.873468	0.898248

Fig. 12 Performance table for 10-fold cross-validation using all student information and by removing withdrawn classes

Prediction at different length of the course using the Random Forest algorithm

Further, a study is conducted with only biographic information and then by adding the information of VLE and assessments at different lengths (20%-100%) of course. Random Forest algorithm is selected randomly as the classification algorithm in this study. Fig. 13 shows that the performance has improved by adding the VLE and assessment information at different times of the courses.

Precision	No VLE,Ass	20%	40%	60%	80%	100%	Recall	No VLE,Ass	20%	40%	60%	80%	100%
Fail	0.249811	0.489840	0.591256	0.677610	0.731437	0.751485	Fail	0.457341	0.766627	0.816833	0.860314	0.879377	0.907058
Pass	0.864201	0.931592	0.939283	0.949647	0.954138	0.964815	Pass	0.715274	0.799248	0.833556	0.865157	0.885495	0.894215
Averaged	0.758908	0.843032	0.860197	0.882410	0.895993	0.909283	Averaged	0.670990	0.792485	0.829745	0.863930	0.883897	0.897625

F-score	No VLE,Ass	20%	40%	60%	80%	100%	Accuracy	No VLE,Ass	20%	40%	60%	80%	100%
Fail	0.322878	0.597092	0.685690	0.757752	0.798284	0.821826	Fail	0.457121	0.765868	0.816739	0.860205	0.879372	0.907176
Pass	0.782626	0.860224	0.883194	0.905352	0.918456	0.928133	Pass	0.715262	0.799175	0.833574	0.865155	0.885497	0.894264
Averaged	0.703809	0.807397	0.838314	0.868861	0.887082	0.900491	Averaged	0.670990	0.792486	0.829746	0.863930	0.883897	0.897624

Fig. 13 Performance table for 10-fold cross-validation using all student information at different stages of the course with the Random Forest algorithm

Results using Weka

Weka is an open-source machine learning software that can be accessed through a graphical user interface, standard terminal applications, or a Java API. It is widely used for teaching, research, and industrial applications, contains a plethora of built-in tools for standard machine learning tasks, and additionally gives transparent access to well-known toolboxes such as scikit-learn, and R. A study is conducted in Weka to compare the results from above study at the end of the course (100%).

Fig. 14 shows the results from Weka using Random Forest algorithm at the end of the course (100%). It is observed that accuracy of Weka is close to the python, but python is giving a better result. Moreover, flexibility of the python makes it to be used for automation.

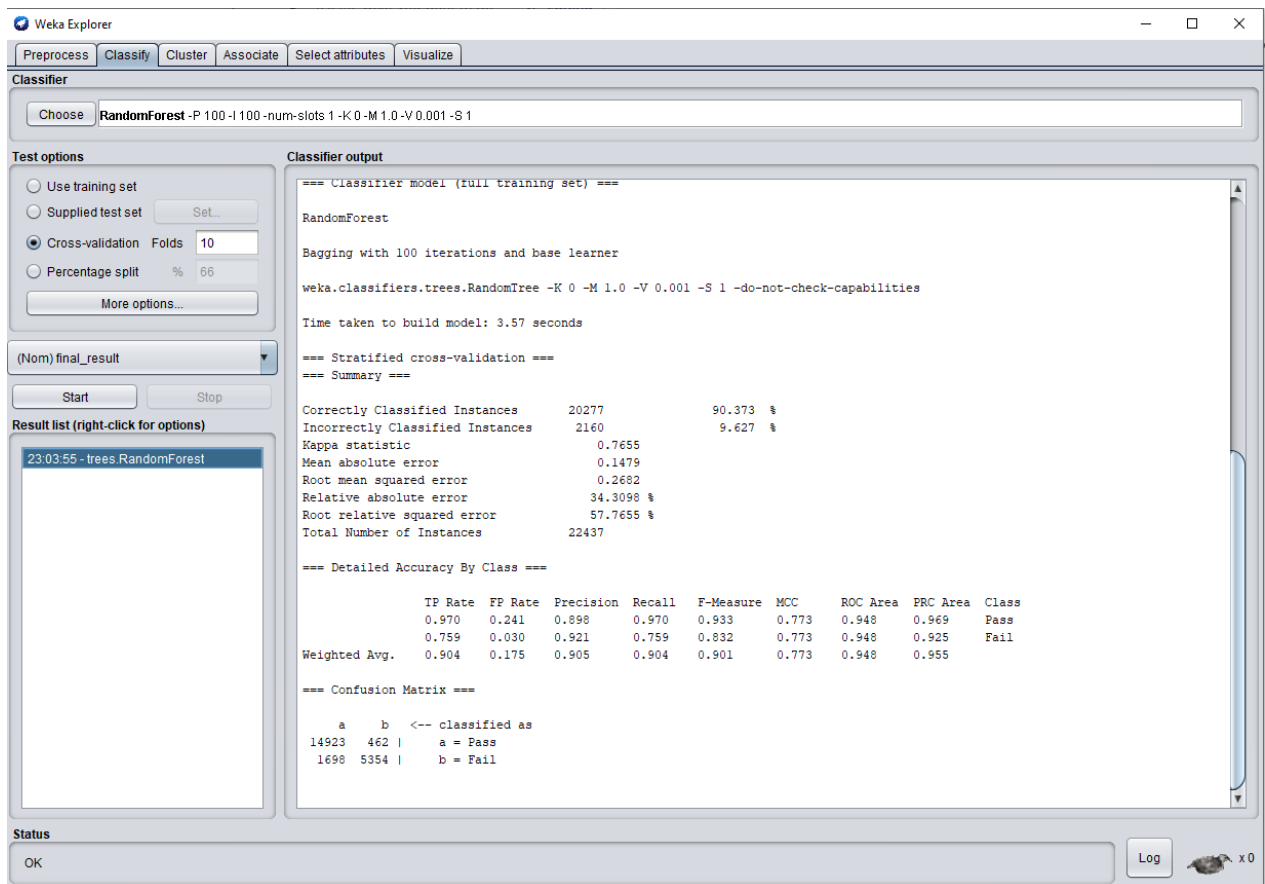


Fig. 14 Weka results for 10-fold cross-validation using all student information with Random Forest algorithm at the end of the course (100%).

7. Conclusions

The student data available in various csv files is cleaned and combined by extracting the important attributes. First EDA is performed to get insights of the data. Next, predictive model is developed using Decision Tree, Random Forest, Gaussian Naïve Bayes, and Gradient boosting algorithms. The problem is approached systematically, first by using only biographic data and then by including VLE, assessment data. Random forest algorithm has best accuracy with a value of 73% with all the information. However, the accuracy of the fail class is poor. Different trials were conducted to improve the performance of students who are failing, so that it could well inform the students at risk. Pass and distinction classes are combined as they represent a single

It was observed that the withdrawn class is skewing the results as there were not much information due to students dropping the course at different times of the course. The pass and distinction class are combined, as they represent same. The final data set only have two classes: Pass and Fail and

the accuracy of predictive model was improved to 90%, the best provided by Random Forest with precision of 0.90 and recall of 0.89. Next, Random forest algorithm is used to train and test the data at different lengths of the course starting from not using any VLE or assessments data and then predicting at 20%, 40%, 60%, 80% , 100% length of the course. It was observed that the accuracy of prediction is 67% without any VLE or assessments information, 80% at 20% course and the accuracy improved to 90% towards the end.

References

1. <https://core.ac.uk/download/pdf/83955564.pdf>
2. <https://www.kaggle.com/rocki37/open-university-learning-analytics-dataset>
3. <https://scikit-learn.org/stable/index.html>

Code GitHub Link

<https://github.com/Triveniedla/Analyzing-Open-University-Learning-Analytics-Dataset>