

Module Name: Data Analytics Major Project and Placement MOD007894

Assessment: Element 011 – Dissertation

Project Title: Developing a recommender system for student admission based on their student's academic performance stated in their application using Machine Learning Algorithms

Student ID: 2054437

Date: 16 September 2022

Declaration page

I, Yan Bogdatov, declare that the work in this dissertation titled “Developing a recommender system for the student admission based on their student’s academic performance stated in their application using Machine Learning Algorithms” is carried out by me. This work has not been submitted to Anglia Ruskin University or any other educational institution for the award of a degree or educational qualification. I also declare that the information published in this dissertation has been obtained and presented in accordance with academic rules and ethical conduct. Any information obtained from other sources has been appropriately referenced.

Acknowledgement page

I would like to express my deepest appreciation to Dr Vanessa Ng and Dr Mahdi Maktabdar for their invaluable help in providing me with essential guidance through the research. This research would never have been done without your effort and patience.

I am also thankful to my placement provider Alison Hartshorn for giving me the initial idea of the presented research.

Lastly, I would like to thank my family for their support and belief in me and my goal to pursue a master's degree in the United Kingdom.

Table of Contents

Declaration page	2
Acknowledgement page	2
List of Figures	4
List of Tables	4
Abstract.....	6
Chapter 1. Introduction	7
1.2 Research Overview	7
1.2 Problem Background.....	7
1.3 Research Aim	8
1.4 Research Objectives	8
1.5 Research Scope	9
1.6 Methodology.....	9
1.7 Contribution.....	10
Chapter 2. Literature Review	10
2.1 Important Factors for Successful Admission.....	10
2.2 Definition of Recommender Systems	11
2.3. Problem Approaches.....	11
2.3.1 Graduate recommender system.	11
2.3.2 Student performance predictor in a Master of Data Science Program	12
2.3.3 Categorizing students with neural networks	13
2.4 Machine learning algorithm evaluation.....	14
2.4.1 Accuracy	14
2.4.2 Precision/Recall.....	15
2.4.3 Mean squared error(MSE)	15
Chapter 3. Research Methodology	15
3.1 Research framework.	15
3.2 Data origin.....	17
3.3 Data content	18
3.4 Exploratory Data Analysis	19
3.4.1 Boxplots for the dataset parameters.....	19
3.4.2 Relation between the parameters.....	26
3.4.3 Correlation Heatmap	30
3.5 Machine learning model	31
3.5.2 Classification Machine Learning Models.....	32

3.6 Evaluation	33
3.6.1 Evaluation of Regression Algorithms	33
3.6.2 Evaluation of Regression Algorithms	33
3.7 User interface.....	33
3.7.1 Prediction page	34
3.7.2 Exploratory data analysis	34
Chapter 4. Results	35
4.1 Machine Learning algorithms Results.....	35
4.1.1 Regression machine learning algorithms.....	35
4.1.1 Classification machine learning algorithms	37
4.2 Web Application Results	40
4.2.1 Predictive System.....	41
Chapter 5. Discussion.....	42
Chapter 6. Conclusion	43
Bibliography	45

List of Figures

Figure 1. Research Framework	16
Figure 2. GRE boxplot.....	20
Figure 3. TOEFL boxplot	21
Figure 4. University Rating boxplot.....	22
Figure 5. SOP boxplot.....	23
Figure 6. LOR boxplot.....	24
Figure 7. Chance of Admit boxplot	25
Figure 8. Histograms for each of the features	26
Figure 9. GRE and TOEFL relation	27
Figure 10. GRE and CGPA relation	28
Figure 11. TOEFL and CGPA relation	28
Figure 12. CGPA and LOR relation.....	29
Figure 13. CGPA and SOP relation.....	29
Figure 14. Correlation Heatmap	30
Figure 15. Training and evaluating the regression models.....	35
Figure 16. Regression script output	36
Figure 17. Regression efficiency results.....	37
Figure 18. Converting output values for classification algorithms	37
Figure 19. Training and evaluating the classification models.....	38
Figure 20. Classification script output.....	38
Figure 21. Classification models accuracy score	40
Figure 22. Prediction page design.....	41
Figure 23. Predictions in Graphic User Interface	42

List of Tables

Table 1. Dataset features table.....	19
Table 2. Regression Algorithms Efficiency Results.....	36
Table 3. Classification Algorithms Accuracy Scores	39
Table 4. User Input Table	41

Abstract

Thousands of student entry applications are sent to the universities' admission offices, but there are not as many employees in these departments. So, the human factor still presents in the student admission process. This research will explore the machine learning algorithms in the student admission sphere, conduct an exploratory data analysis on the graduate admission dataset, train the classification and regression machine learning models on the example of Anglia Ruskin University, and evaluate each of the machine learning models to identify the most efficient algorithm for admission purposes and propose the web application prototype for the university admission office. The experimental results showed that the most efficient classification model was trained on the Logistic Regression algorithm. In contrast, the most efficient regression model was trained on the Extra Trees Regression algorithm.

Keywords: machine learning, classification, regression, students' admission

Chapter 1. Introduction

1.2 Research Overview

The research's primary purpose is to design and develop a solution for the International Admission Office to make early decisions on graduate admission applications of prospective students. The developed system should be a recommender system using machine learning algorithms. Given recommender system would increase the efficiency of the Admission Office of Anglia Ruskin University. The research will contain the literature review, which will discuss the problem overview, approaches made by other researchers in a similar sphere, and evaluation metrics of the proposed algorithms for the given problem. The research also will discuss the proposed methodology and the comparison of used machine learning algorithms for the following solution. The subsequent analysis should be supported with the artefact, which will contain Python Jupyter Notebook, and build a graphical user interface for the International Admission Department of Anglia Ruskin University. It should be mentioned that the research will be based on public data. Still, Anglia Ruskin University can reuse it in future by using the university's private data for retraining machine learning models.

1.2 Problem Background

Nowadays, most of the admission offices in universities consist of not more than five employees. At the same time, there are thousands of student applications per academic semester that the student admission office should correctly analyse. All the analysis is made manually, and the human factor risk is relatively high. Considering the abovementioned facts, there should be a solution to the given problem. Implementing machine learning and artificial intelligence is one of the most efficient and modern solutions. But it should be stated that the final decision should be strictly made by the staff of the admission office only, so the implemented system

should be used for early decisions only. So, it is supposed that involving machine learning systems and artificial intelligence in the work of the student admission office will significantly increase the department's effectiveness.

While conducting the research through existing solutions, either in academic sources or on the internet itself, it was discovered that for each university, the predictive system should be developed from scratch because each university has a different case and different admission requirements. That was the main reason for conducting the given research.

The research question that should be answered at the end of the given research is: what are the most efficient machine learning algorithms for predicting student admission to the university?

1.3 Research Aim

The primary research aim is to create a recommender system for the early decision on students' university admission applications using Machine Learning algorithms.

This aim should be reached by conducting a literature review to find out current approaches made by other machine learning researchers for university student admission field, using Jupyter Notebook for making an exploratory data analysis on the used dataset, training machine learning algorithms with datasets and comparing evaluation metrics of the given algorithms between each other to identify the most efficient one.

1.4 Research Objectives

The research objective for the given research are:

1. To conduct a literature review on the given solutions used for graduate predictions and compare them with each other

2. To create a script to train a machine learning model with admission data using different machine learning algorithms and compare them between each other
3. To create a ready web application prototype using the best machine learning models for early predictions in student admission

1.5 Research Scope

Some scholarly articles on using machine learning and artificial intelligence in student admission. The primary research scope is creating a recommender system for Anglia Ruskin University's Admission Office. So, given research should fit the requirements of Anglia Ruskin University admission, and the machine learning models could be easily retrained for further Anglia Ruskin University usage.

All the functions and dataset features that do not fit Anglia Ruskin University's requirements are considered out of scope. For example, a dataset with a German Language Proficiency Examination results feature is considered out of scope, as Anglia Ruskin University courses are taught in English.

1.6 Methodology

Given research will be conducted using the following methodology:

- the public dataset with students' academic results will be used
- the dataset will be cleaned and pre-processed for analysing and training of the machine learning algorithms
- the exploratory data analysis will be conducted on the given dataset to identify the data patterns and insights
- multiple classification and regression machine learning algorithms will be trained on the dataset

- trained machine learning models will be evaluated through machine learning metrics to identify the most efficient algorithm for the admission problem
- the most efficient machine learning model will be saved, and a web application will be created using this model

1.7 Contribution

The results of the given research will fill the gap in the topic of prediction systems in university admission offices. And built artefact, which this research will be followed with, will increase the Admission Office efficiency of Anglia Ruskin University. If given, the artefact will be retrained with university data and implemented in the department in future, it will minimise the human error in making early decisions on students' applications.

Chapter 2. Literature Review

2.1 Important Factors for Successful Admission

Using real students' data for five years (between 2007 and 2012), Gupta and Turek have researched the predictors for success in the MBA program. Given research includes successful admission into the university and a successful master's in business administration in the business school of Virginia, US. The methodologies used by Gupta and Turek were based on regression techniques. The data consisted of 14 columns: age, participation in Bootcamp, gender, graduate GPA, GMAT test, Major Field Test and undergraduate GPA. After getting the regression results, they were placed in the regression results table. The most impactful factors of being admitted and successfully graduating with an MBA in the business school of Virginia were GPA and GMAT. (Gupta and Turek, 2015)

2.2 Definition of Recommender Systems

Recommender systems use artificial intelligence and machine learning methods to provide user-item recommendations. For example, a recommender system for the book shop would be a system that classifies a book by genre and then recommends a book based on the genres of users' previously purchased books. (Portugal, Alencar and Cowan, 2018) Recommender systems can be divided into three main groups: collaborative, content-based and hybrid filtering.

- a) Content-based – given recommender systems can be described as the systems that recommend the product to the user based on users' previous ratings—only the products highly similar to the users' preferences.
- b) Collaborative – unlike content-based recommendation systems, collaborative ones are making their suggestions based not on the given users' preferences but the other users' preferences, so-called popular products,
- c) Hybrid filtering –given recommendation systems contain both collaborative and content-based approaches so that users will get mixed recommendations based either on their preferences or the preferences of other users.

(Adomavicius and Tuzhilin, 2005)

2.3. Problem Approaches.

2.3.1 Graduate recommender system.

El Guabassi, Bousalem, Marah and Qazdar were developing a recommendation system to assess the submitted students' applications. Supervised machine learning algorithms were used to build a predictive model. At first, El Guabassi, Bousalem, Marah and Qazdar calculated the correlation between the parameters in their dataset. Their results showed a high correlation between the examination scores, such as GRE and TOEFL and students' GPA. Researchers used four machine

learning regression algorithms while building a recommender system for early admission prediction. Linear, Decision Tree, Support Vector and Random Forest regressors were used. Three metrics were used to evaluate each machine learning model: Mean Square Error, Root Mean Square Error and R-squared. It was found that Random Forest Regressor provided the best efficiency in comparison with other algorithms, as it had the lowest Mean Square Error and Root Mean Square Error and the highest R-Squared coefficient. Moreover, researchers have built an interface where users can input the values and get an early admission prediction (El Guabassi, Bousalem, Marah and Qazdar, 2021).

2.3.2 Student performance predictor in a Master of Data Science Program

Yijun Zhao, Qiangwen Xu, Ming Chen and Gary Weiss have researched student performance prediction using machine learning algorithms. They have stated that given research was conducted not just to identify if the student would perform well or poorly but to explain the most relevant factors to educate admission offices on which factors are more relevant to success in a specific program. The dataset of 826 applicants was used, where 60% of students were accepted, while the other 40% got a rejection decision. The dataset contained the following sixteen features: GRE Verbal %, GRE Quantitative %, GRE Writing %, TOEFL Score, Gap time, Age, Marital Status, Gender, Citizenship, School GPA, Major, Degree, Country, Language, Rank, and GPA.

Key observations made by the researchers were:

- a) Rejected applicants most likely had a GPA lower than 3.0 (41%)
- b) Rejected applicants most likely had a degree from the universities under 2000th rank (31%)

- c) International students were 74% of the accepted students, 80% of the students that have not enrolled and 40% of the rejected students
- d) The student with more than two years for their gap year were rejected (33%)

After exploratory data analysis, the researchers applied machine learning classification algorithms. Logistic regression, Support Vector Machines, Decision Trees, Random Forest, K-Nearest Neighbor, Ensemble Learner L and Naïve Bayes, were used. As evaluation metrics, basic accuracy scores were used by comparing machine learning predictions on the test data and actual values. The Random Forest and the Ensemble Learner L classification machine learning algorithms gained the best accuracy scores, 83% and 86%, respectively. Given classification, machine learning prediction systems were built to identify the students into two groups: top and bottom-performing students (Zhao, Xu, Chen, and Weiss, 2020).

2.3.3 Categorizing students with neural networks

Based on the research done by Steven Walczak in 1994, experienced admission officers can categorise students without the help of a neural network. Still, the given process will be automated and will increase the overall performance of the admission office. He used a small US private university as an example for the presented research. Given university has five admission officers and over 1500 applicants per academic year, which make it impossible to identify each student manually, in case of creating a successful neural network for accessing applicants will significantly improve the efficiency of the admission process. Walczak determined that neural networks should be making their prediction based on personal or academic profiles and student demographic information. Demographic data should include the address and gender, the personal profile should be found on the student's motivation, such as how students' first enquiry was made (telephone call, email or sending

SAT/TOEFL scores), and finally, academic measures should include cumulative GPA and TOEFL scores. The neural network decided to rely on 26 input nodes and provided the output between 0 and 1 to determine the likelihood of further student enrolment.

Walczak's system had following pipeline: student database -> data pre-processing (converting text data to numeric values) -> neural network -> Enrolment Expert System (graphical interface for admission office) -> Admission Counsellor(end-user) -> Prediction Database.

Steven calculated the fundamental error counts percentage to evaluate his neural network. For the students who had enrolled in the university, the neural network predicted positive results in 92%, while the rate of correct predictions for the not enrolled student was 72%. Walczak explained that the university data he had used contained noisy data, which were not cleared before predicting it in the neural network (Walczak, 1994).

2.4 Machine learning algorithm evaluation

Sebastian Raschka stated three main reasons for predictive performance evaluation:

- Estimation of the general performance of the machine learning model
- Increase the predictive performance of machine learning algorithm for a particular dataset
- Comparison of machine learning algorithms to identify the best model for the problem (Raschka, 2018)

2.4.1 Accuracy

The accuracy score is the most straightforward evaluation measure. It measures the ratio of correctly predicted values over the total predictions.

The accuracy score formula is: $(tp+tn)/(tp+fp+tn+fn)$

2.4.2 Precision/Recall

Precision is the percentage of correctly predicted values from the total predicted patterns in the positive class.

Precision score formula is: $tp/(tp+fp)$

The recall is a fraction of correctly predicted positive patterns

Recall score formula is: $tp/(tp+tn)$

2.4.3 Mean squared error(MSE)

Mean squared error measures the difference between predicted and desired solutions.

The mean squared error formula is $\frac{1}{n} \sum_{j=1}^n [(p_j - A_j)]^2$, where p_j is the prediction value of instance j and A_j is a real value of model j (M and M.N, 2015).

Chapter 3. Research Methodology

3.1 Research framework.

Illustrated block diagram below will outline processes that will be used to achieve the research aim and create the artefact for the given research.

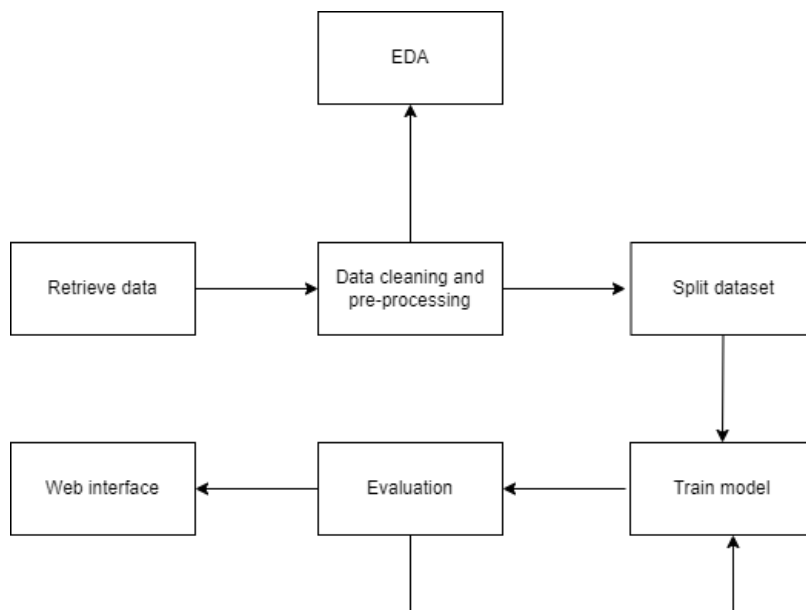


Figure 1. Research Framework

The first process that should be accomplished is to retrieve the data. For developing an efficient web application based on a machine learning model, correct data should be retrieved from the target's DMS(data management system); in our case, it is the Anglia Ruskin University dataset.

The next step could be cleaning the data from null values and cleaning the data outliers. After cleaning the data, all string values should be converted to integers before training the machine learning model.

OneHotEncoder, a function in the SKLearn Python library, is the best practice to encode categorical data into numerical features for further use in machine learning model training.

After finishing the cleaning and pre-processing stage, the exploratory analysis should be applied. Exploratory data analysis is critical for discovering data patterns and spotting anomalies using graphical representations.

The dataset should be split into “test”, and “train” splits to 80% to 20% of the main dataset, respectively. SKLearn library contains a particular function, “train_split_test”, for simplifying the process.

The next step after splitting the dataset should be training the machine learning model with the “train” dataset. Identifying the correct machine learning model is essential for developing an effective predictive system. Machine learning models could be divided into two groups: classification and regression.

After training the machine learning model, its effectiveness should be evaluated. The best way to measure the effectiveness of the machine learning model is to use such evaluation metrics as accuracy score, F1 score, Mean Squared Error, and AUC, which is the area under the ROC curve. Metrics choices will be discussed later.

A web interface should be developed when the model’s metrics results fit the requirements. Flask or Python Django library can be used to create simple web applications based on machine learning models.

3.2 Data origin

Due to data privacy reasons, it was impossible to use the data of Anglia Ruskin University, which was the primary target audience for the given research and application. It was decided to use the public data, and the features of the data should be compatible with Anglia Ruskin University’s needs. “Graduate Admission Dataset v2” was used for descriptive analysis and training of the machine learning model. So, the built machine learning model and data exploration visualisations based on the given public data can be easily retrained shortly in case Anglia Ruskin University uses the given application.

The given dataset was downloaded from Kaggle.com

(<https://www.kaggle.com/datasets/mohansacharya/graduate-admissions>).

3.3 Data content

As it was discovered in the literature research section, the main factors used for university admission are the TOEFL exam, GRE results, GPA, and the recommendation letter/ statement of purpose. Graduate Admission Dataset v2 contains the following parameters:

- GRE scores – student results in GRE examination. The maximum GRE score is 340.
- TOEFL scores – student results in the TOEFL examination. The maximum TOEFL score is 120.
- University rating – rating for the university between 1 and 5. The given parameter can be changed to the rating of Anglia Ruskin University in the future.
- Undergraduate GPA – Students' GPA during their bachelor's degree. The range for the given parameter is between 0 and 10.
- Research experience – Any research conducted by the student previously. 0 – the student had not conducted research before, 1 – the student had conducted research.
- Chance of admission: the chance of successful student admission to the university, between 0 and 1.

Based on the information above following feature table could be created:

Feature name	Description	Data Type
GRE	Examination results	Quantitative
TOEFL	Examination results	Quantitative
University Rating	University Rank	Quantitative
Undergraduate GPA	Previous academic results	Quantitative
Research Experience	The student has conducted the research/ Student has not completed the research.	Quantitative
Chance of Admit	Probability of the successful admission	Quantitative

Table 1. Dataset features table

3.4 Exploratory Data Analysis

3.4.1 Boxplots for the dataset parameters

At the first stage of exploratory data analysis, it was decided to draw the boxplots for each parameter in the dataset using the python library Matplotlib. Boxplots can be used to identify the dataset's quality for machine learning training. General trends for each of the dataset features can be identified.

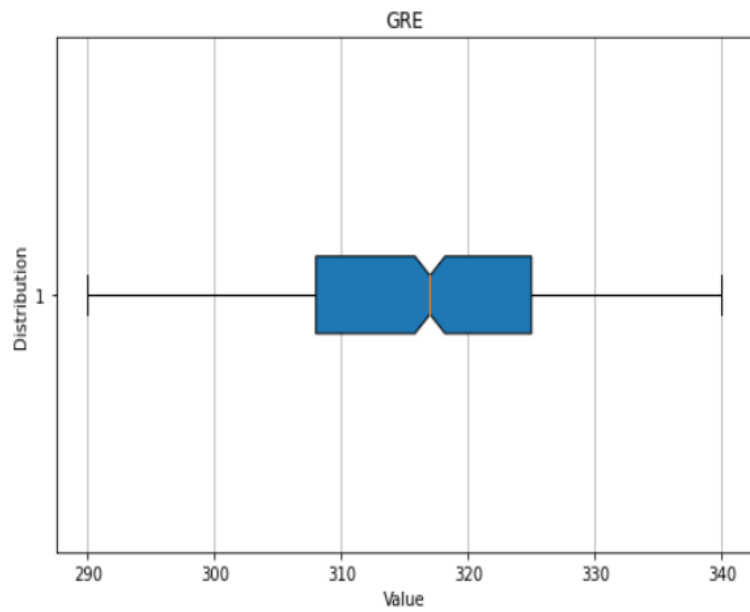


Figure 2. GRE boxplot

Descriptive analysis of GRE feature:

Minimal value: 290

Maximum value: 340

Most values are between 308 and 325

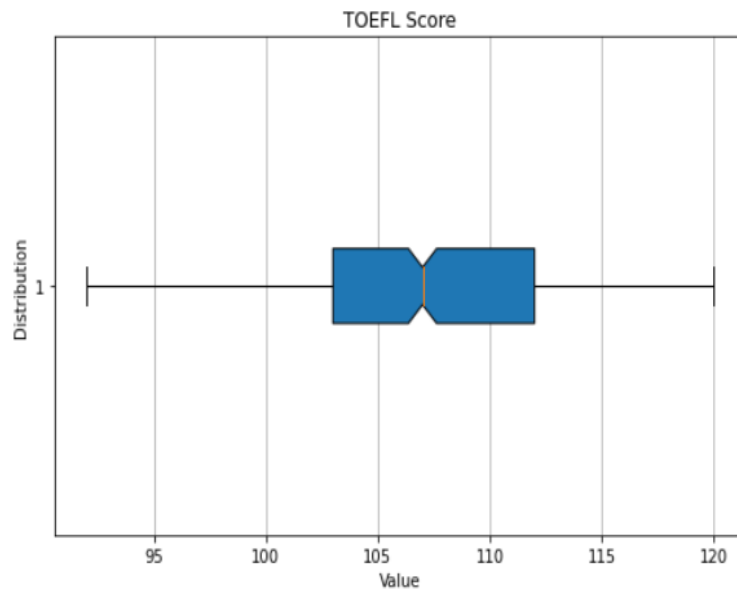


Figure 3. TOEFL boxplot

Descriptive analysis of TOEFL feature:

Minimal value: 92

Maximum value: 120

Most values are between 103 and 112

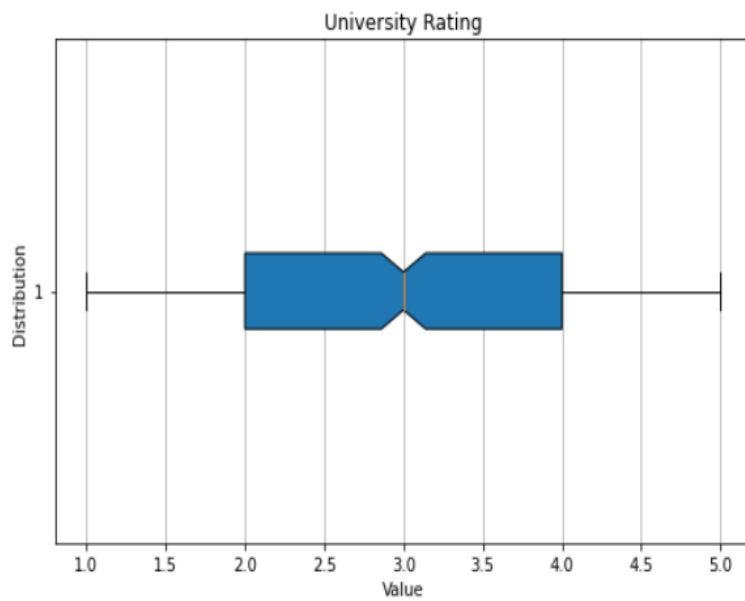


Figure 4. University Rating boxplot

Descriptive analysis of University Rating feature:

Minimal value: 1

Maximum value: 5

Most values are between 103 and 112

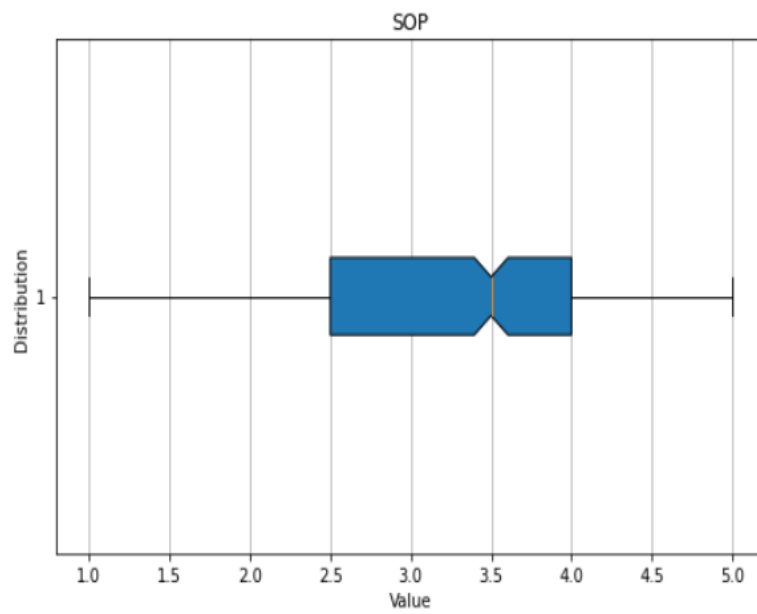


Figure 5. SOP boxplot

Descriptive analysis of SOP feature:

Minimal value: 1

Maximum value: 5

Most values are between 2.5 and 4

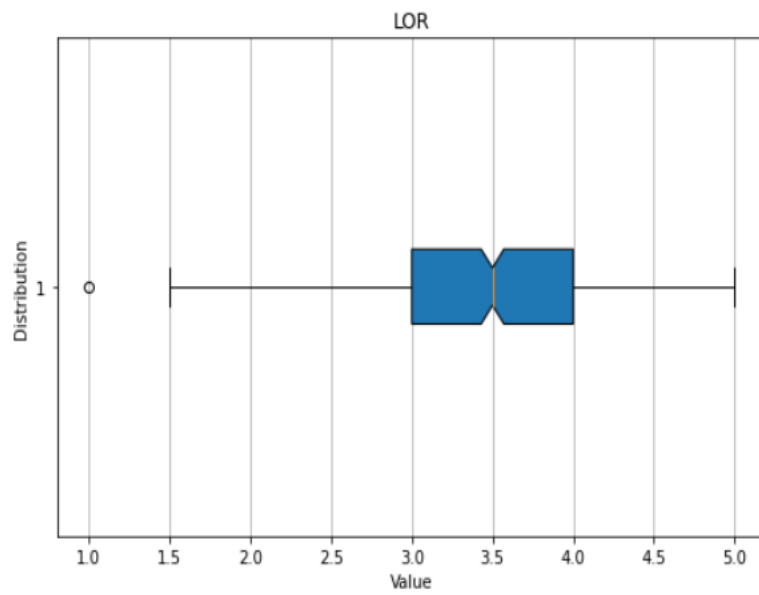


Figure 6. LOR boxplot

Descriptive analysis of LOR feature:

Minimal value: 1

Maximum value: 5

Most values are between 3 and 4

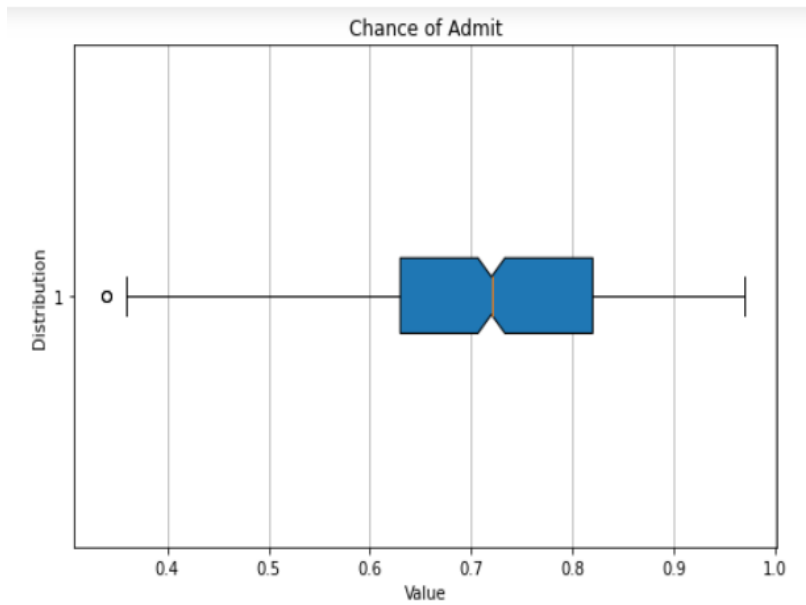


Figure 7. Chance of Admit boxplot

Descriptive analysis of Chance of Admit feature:

Minimal value: 0.34

Maximum value: 0.97

Most values are between 0.63 and 0.97

Based on the following insights, the given dataset is populated fairly and can be used for creating an efficient machine learning model. Moreover, this statement can be supported by the histograms pictured below.

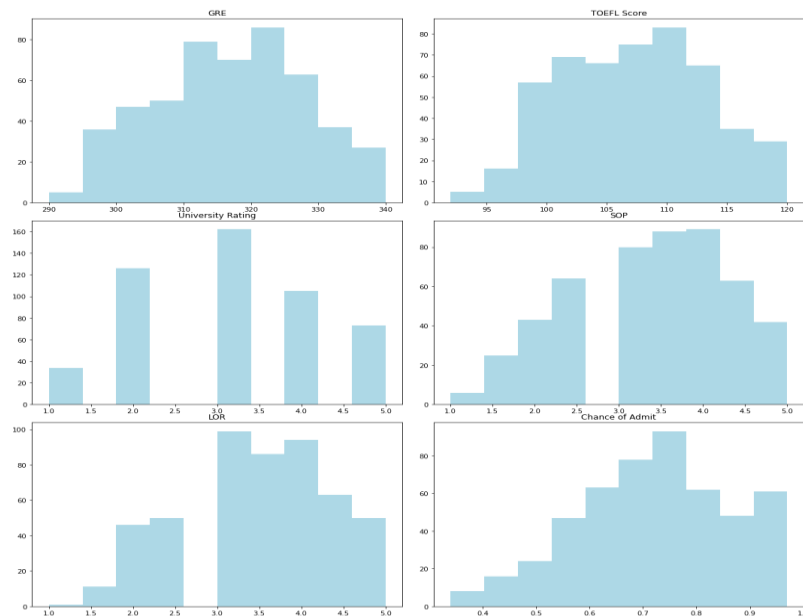


Figure 8. Histograms for each of the features

3.4.2 Relation between the parameters

The next important step for Exploratory Data Analysis is identifying the relationships between features of the dataset. “Seaborn” is a Python library used to plot linear regression graphics. A regression graphics will be used to find the relationship between the following dataset parameters:

- Relationship between GRE results and TOEFL results
- Relationship between GRE results and CGPA
- Relationship between TOEFL results and CGPA
- Relationship between CGPA and Letter of Recommendation(LOR)
- Relationship between CGPA and Statement of Purpose(SOP)

High relationship

GRE results and TOEFL results relationship:

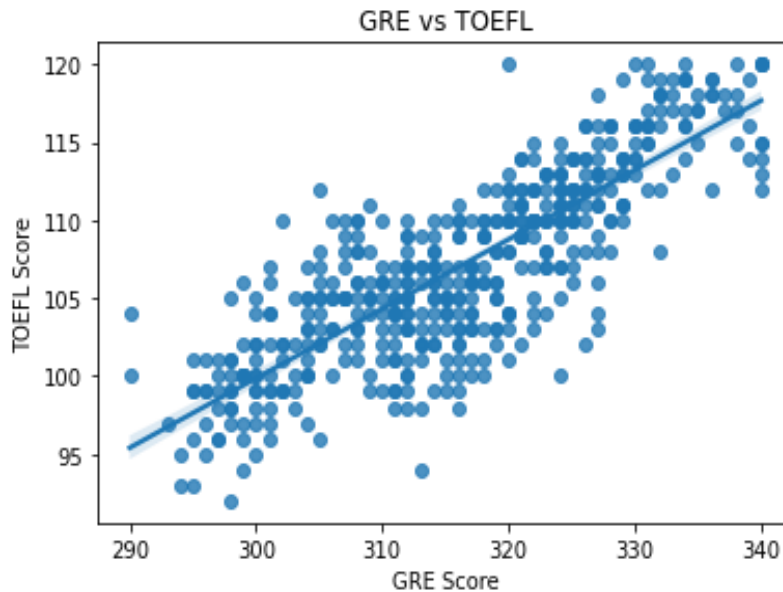


Figure 9. GRE and TOEFL relation

The figure above shows that GRE and TOEFL parameters are highly related. It means that the student who got higher results in the GRE examination will probably have higher marks in the TOEFL examination, and the student who did not perform well in one of the examinations will get lower results in the other.

GRE results and CGPA performance relationship

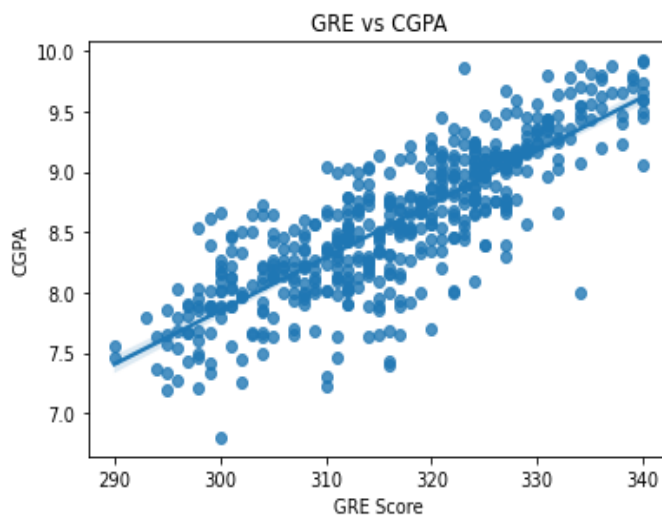


Figure 10. GRE and CGPA relation

As can be seen from the figure above, the GRE score is highly correlated with the cumulative GPA score. The students who achieve high academic results during their bachelor's degree would probably get higher marks on the GRE.

Slight relationship

TOEFL results and CGPA performance relationship

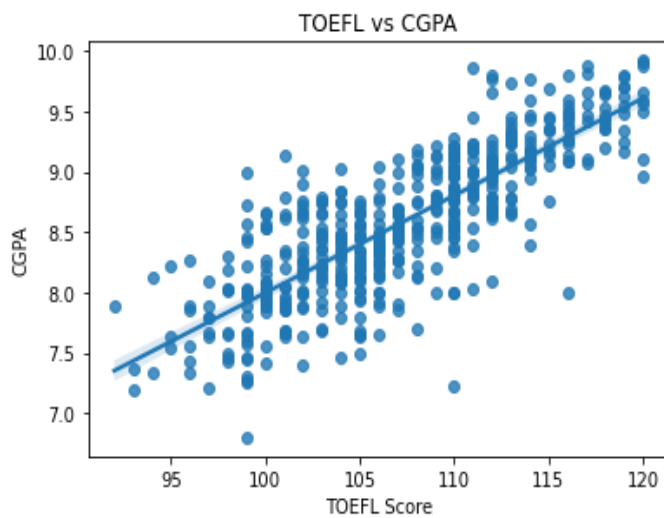


Figure 11. TOEFL and CGPA relation

At the same time, only slight relation was detected between the TOEFL results and cumulative GPA. This means that not all the students with high academic performance during their bachelor's degree would score highly on the TOEFL examination.

No relationship

CGPA performance and Letter of Recommendation (LOR) relationship

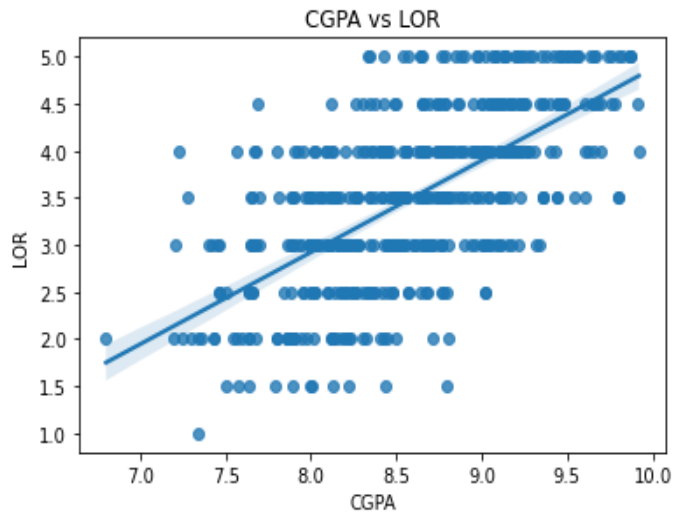


Figure 12. CGPA and LOR relation

Figure 12 shows no relation between students' performance during their bachelor's degree and the score of their recommendation letter.

CGPA performance and Statement of Purpose (SOP) relationship

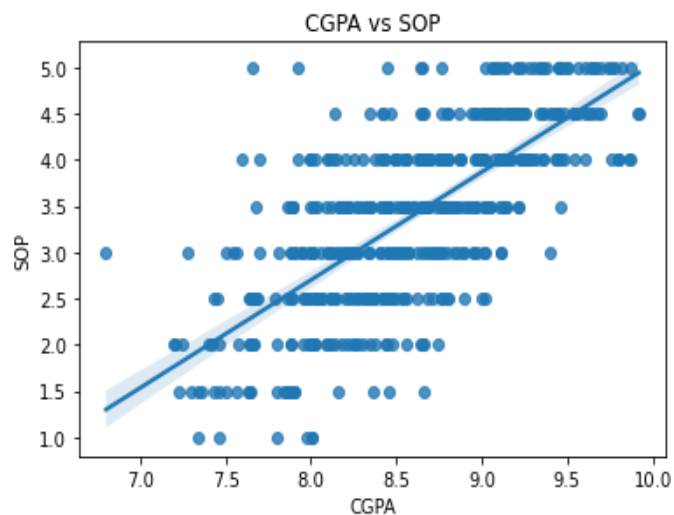


Figure 13. CGPA and SOP relation

Moreover, the Statement of Purpose is not related to a student's academic performance during their bachelor's degree.

3.4.3 Correlation Heatmap

To conclude the correlation between features in the Graduate Dataset v2, it was decided to draw a heatmap based on a correlation table for all the features.

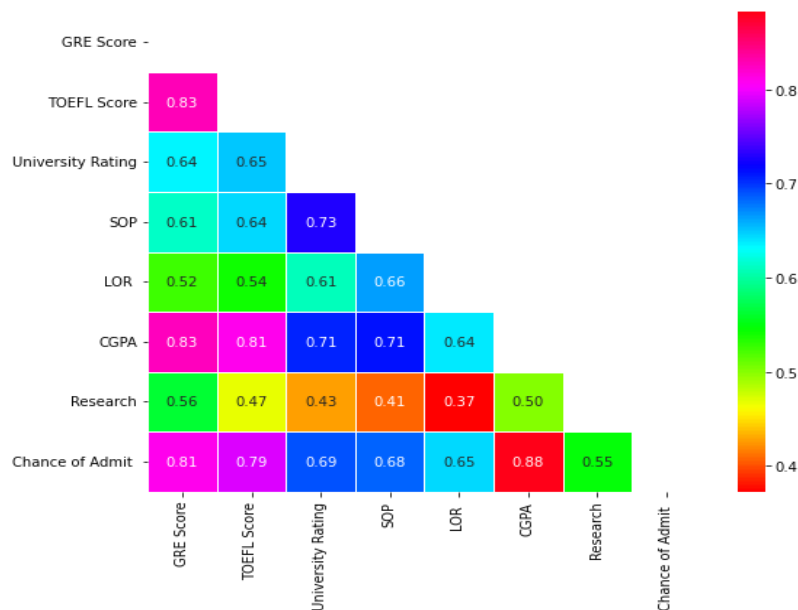


Figure 14. Correlation Heatmap

As it can be seen from the created heatmap above, the most correlated features between each other are:

- Cumulative GPA and GRE Score – 0,83
- Cumulative GPA and TOEFL Score – 0,81

And the least correlated features between each other are:

- Letter of Recommendation and GRE Score - 0,52
- Letter of Recommendation and TOEFL Score – 0,54
- Research and Cumulative GPA – 0,50

3.5 Machine learning model

The two most common ways of solving predictive problems are building a classification machine learning model or a regression machine learning model. The classification machine learning model can predict the label or categorical output variables, while the regression machine learning model can predict continuous variables, such as integer or float values.

As Graduate Admission Dataset is used, where the output value is the probability of student admission from 0 to 1, it is possible to use either classification or regression to solve the given predictive problem.

3.5.1 Regression Machine Learning Models

For predicting the output for the given problem, it was decided to use the following regression models:

- Ada Boost Regression
- Extra Trees Regression
- Linear Regression
- Decision Tree Regressor
- Random Forest Regressor
- KNeighbours Regressor
- Support Vector Regressor

All the mentioned regressors are part of the SKLearn Python library. Before training the models with Admission Graduate Dataset, it was divided into train and test splits of 80 and 20 per cent of the whole dataset, respectively. Moreover, the 'Normalize' function, which is the SKLearn Python library, was used for better accuracy.

The Regression Machine Learning model can predict the probability between 0 and 1 for the prospective student's admission based on their application.

3.5.2 Classification Machine Learning Models

For predicting the output for the given problem, it was decided to use the following regression models:

- Logistic Regression
- Random Forest Classification
- Gradient Boosting Classification
- KNeighbors Classification
- Support Vector Classification
- Gaussian Naïve Bayes

All the mentioned regressors are part of the SKLearn Python library. Before training the models with Admission Graduate Dataset, it was divided into train and test splits of 80 and 20 per cent of the whole dataset, respectively. For the 'Admit Chance' feature, our prediction target, it was decided to replace all values more than 0.75 with 1 and other values with 0 to convert the given problem into a classification problem.

Classification machine learning could divide applications of the students into two groups:

- Successful application when previous academic results of the student fit the university requirements
- Unsuccessful application when previous academic results of the student do not fit the university requirements

3.6 Evaluation

As there were two solutions for the problem (Classification and Regression), each solution will be evaluated differently.

3.6.1 Evaluation of Regression Algorithms

All the regression models will be evaluated for the Regression solution with Mean Squared Error. The bar chart will present all the results for choosing the most efficient algorithm. Predictions will be made based on the data divided for testing to assess each of the used regression machine learning algorithms. The next step will be calculating the mean squared error by comparing the actual outputs and outputs predicted by each of the algorithms used. The smaller the mean squared error, the better the machine learning algorithm predicts values for the given dataset.

3.6.2 Evaluation of Regression Algorithms

All the trained classification models will be evaluated with an Accuracy score for the classification solution. The bar graph will present accuracy scores for each machine learning algorithm. Predictions will be made based on the data divided for testing purposes to assess each of the used regression machine learning algorithms. The next step will be calculating the accuracy score by comparing the actual outputs and outputs predicted by each algorithm. The higher the accuracy score, the more efficient the machine learning algorithm is. The target accuracy score for the given machine learning classification problem is more than 90%.

3.7 User interface

The proposed user graphical interface should consist of two web pages:

- Prediction page
- Exploratory data analysis

3.7.1 Prediction page

The prediction page should consist of text boxes or sliders to collect the user input for the following parameters:

- GRE Score
- TOEFL Score
- University Rank
- Statement of Purpose rating
- Letter of Recommendation rating
- Cumulative GPA
- Research made in the past

Moreover, the validation function should be included. For the GRE Score user should be able to input the value between 130 and 340; for the TOEFL score value is between 0 and 120; for SOP, LOR and University Rank, the value should be between 1 and 5, and CGPA must be between 0 and 10, while Research parameter should be 1 or 0 only.

3.7.2 Exploratory data analysis

Moreover, mentioned web application should contain a separate page with the insights of the used dataset. This will make the application clearer, and the user can see which dataset the current machine learning algorithm was trained with. Displayed insights should contain graphs and charts which were discussed previously.

Chapter 4. Results

4.1 Machine Learning algorithms Results

As was discussed before, two different solutions for the given dataset were applied, as our output value (the value we should predict) can be a binary label: is the application successful or not, or the output value can be a probability of the successful application from zero to one.

4.1.1 Regression machine learning algorithms

Seven machine learning algorithms were trained with the dataset to find the most efficient one for the problem. The following seven algorithms were tested with the data:

- Ada Boost Regression
- Extra Trees Regression
- Linear Regression
- Random Forest Regressor
- Kneighbors Regressor
- Support Vector Regressor

```
reg_models = [  
    ['Ada Boost Regression:', AdaBoostRegressor()],  
    ['Extra Trees Regression:', ExtraTreesRegressor()],  
    ['Linear Regression:', LinearRegression()],  
    ['Decision Tree Regressor:', DecisionTreeRegressor()],  
    ['Random Forest Regressor:', RandomForestRegressor()],  
    ['KNeighbors Regressor:', KNeighborsRegressor()],  
    ['SVR:', SVR()]  
]  
  
mse_dict = []  
for name, model in reg_models:  
    model = model  
    model.fit(X_train, y_train)  
    predict = model.predict(X_test)  
    mse = np.sqrt(mean_squared_error(y_test, predict))  
    mse_dict.append(mse)  
    print(name, mse)
```



Figure 15. Training and evaluating the regression models

Algorithms were put in the Python dictionary and trained with the train split data of the Graduate Admission Dataset using a “for loop”. After training, each model predicted the chance of admission using test data of the same dataset. The next

step was calculating the mean squared error of predicted values compared to actual dataset values.

```
Ada Boost Regression: 0.08458109738286676
Extra Trees Regression: 0.07415827937594023
Linear Regression: 0.07765759656302856
Decision Tree Regressor: 0.10768008172359454
Random Forest Regressor: 0.07823075801754702
KNeighbors Regressor: 0.08882567196480981
SVR: 0.11746039395819052
```

Figure 16. Regression script output

Figure 16 shows that Extra Trees Regressor, Linear Regressor and Random Forest Regressor had quite a similar mean squared error, about 0.07.

The table with results can be seen below:

Algorithm	Mean Squared Error
Ada Boost Regression	0.084
Extra Trees Regression	0.074
Linear Regression	0.076
Decision Tree Regression	0.107
Random Forest Regression	0.078
KNeighbors Regression	0.088
Support Vector Regression	0.117

Table 2. Regression Algorithms Efficiency Results

It was decided to draw a bar graph using the Python Seaborn library to display used evaluation metrics.

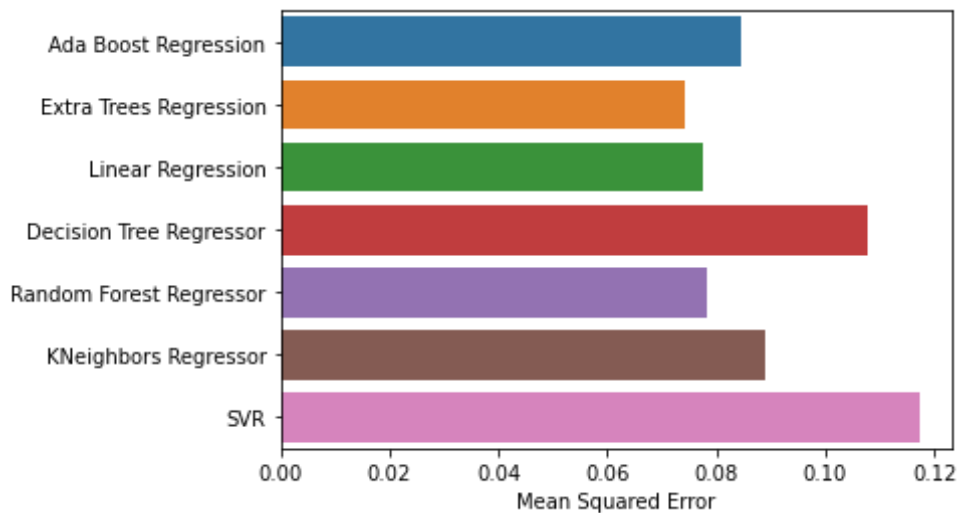


Figure 17. Regression efficiency results

The difference between the three best models is that linear regressions perform well when there are fewer features in the dataset. In contrast, Random Forest and Extra Trees Regression will perform better in the datasets with more columns. So, it was decided to choose the Extra Trees Regression model as the most efficient way to predict values for graduate applications.

4.1.1 Classification machine learning algorithms

For using the classification machine learning algorithms, output was transformed into binary labels by the line of code in the figure below.

```
[25] df.loc[df['Chance of Admit ' ] > 0.75, 'Chance of Admit ' ] = 1
[26] df.loc[df['Chance of Admit ' ] < 0.75, 'Chance of Admit ' ] = 0
```

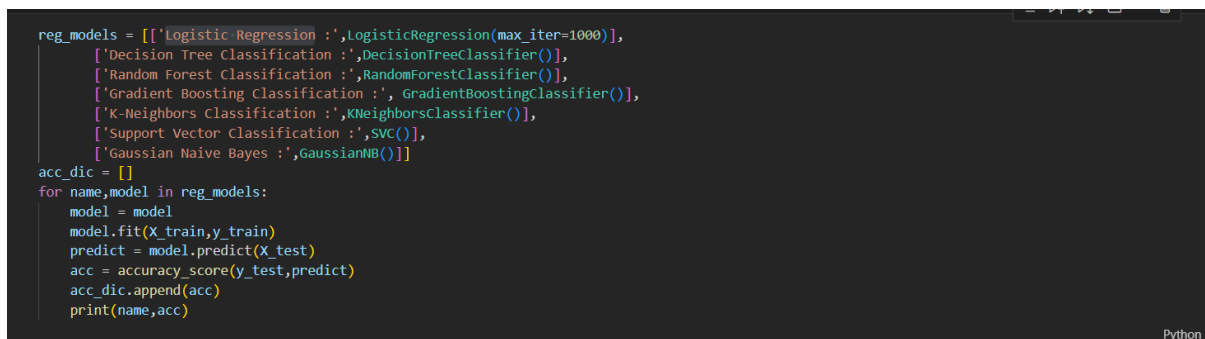
Figure 18. Converting output values for classification algorithms

All the values more than 0.75 (75% chance of being accepted to the university) were converted to 1, while other values below 0.75 were converted to 0.

To identify the best machine learning algorithm for a given classification problem, seven algorithms were trained with the train split data:

- Logistic Regression
- Decision Tree Classification
- Random Forest Classification
- Gradient Boosting Classification
- KNeighbors Classification
- Support Vector Classification
- Gaussian Naive Bayes

The next step was evaluating the models; for these purposes, trained algorithms made predictions on test split data and given predictions were compared to the actual results using the accuracy score, a function of the SKlearn Python library.

A screenshot of a Python script in a dark-themed editor. The script defines a list of models, trains them on training data, and evaluates them on test data using the accuracy score. The models listed are Logistic Regression, Decision Tree Classification, Random Forest Classification, Gradient Boosting Classification, K-Neighbors Classification, Support Vector Classification, and Gaussian Naive Bayes. The script uses the sklearn library for all operations.

```
reg_models = [['Logistic Regression :', LogisticRegression(max_iter=1000)],
               ['Decision Tree Classification :', DecisionTreeClassifier()],
               ['Random Forest Classification :', RandomForestClassifier()],
               ['Gradient Boosting Classification :', GradientBoostingClassifier()],
               ['K-Neighbors Classification :', KNeighborsClassifier()],
               ['Support Vector Classification :', SVC()],
               ['Gaussian Naive Bayes :', GaussianNB()]]
acc_dic = {}
for name, model in reg_models:
    model = model
    model.fit(X_train, y_train)
    predict = model.predict(X_test)
    acc = accuracy_score(y_test, predict)
    acc_dic.append(acc)
    print(name, acc)
```

Figure 19. Training and evaluating the classification models

Script for the training and evaluation used machine learning models gave the following output:

A screenshot of the output of the Python script shown in Figure 19. It displays the accuracy score for each of the seven classification models.

```
Logistic Regression : 0.92
Decision Tree Classification : 0.89
Random Forest Classification : 0.88
Gradient Boosting Classification : 0.87
K-Neighbors Classification : 0.88
Support Vector Classification : 0.84
Gaussian Naive Bayes : 0.91
```

Figure 20. Classification script output

As can see from the screen results that three models had close accuracy scores. Logistic Regression, Decision Tree Classification and Gaussian Naïve Bayes had 0.92, 0.89 and 0.91 accuracy scores, respectively. Given scores for the given machine learning problem were considered efficient.

The table with all the results of classification machine learning models is presented below.

Algorithm	Accuracy score
Logistic Regression	0.92
Decision Tree Classification	0.89
Random Forest Classification	0.88
Gradient Boosting Classification	0.87
KNeighbors Classification	0.84
Gaussian Naïve Bayes	0.91
Support Vector Classification	0.84

Table 3. Classification Algorithms Accuracy Scores

Moreover, it was decided to draw a bar graph with the results of the classification machine learning algorithms using the Seaborn – Python library.

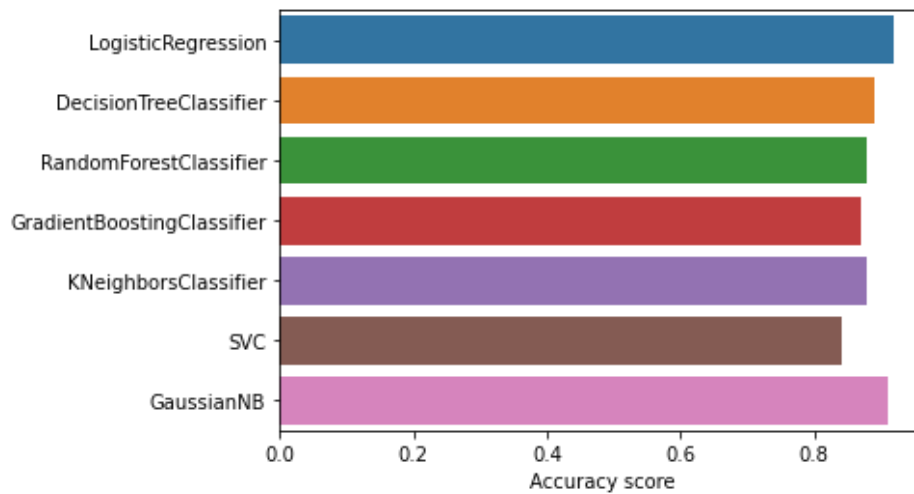


Figure 21. Classification models accuracy score

The model based on the logistic regression algorithm was the most efficient classification machine learning model for predicting successful student applications.

4.2 Web Application Results

The graphical User Interface for the most efficient classification and regression machine learning models was developed using the Python library Streamlit. Before creating the graphical interface for the given admission problem, the Extra Trees Regression and Logistic Regression models were saved using Pickle, Python Library.

The web application consists of two pages: Predictive System and Exploratory Data Analysis.

4.2.1 Predictive System

The predictive system page asks the user to input student values. The table below shows the dataset feature, the input type, and accepted values on the predictive system web page:

Dataset feature	Input type	Accepted values
GRE Score	Streamlit Slider	260 to 340
TOEFL Score	Streamlit Slider	0 to 120
Course Ranking	Select Box	1,2,3,4,5
SOP	Streamlit Slider	1 to 5
LOR	Streamlit Slider	1 to 5
CGPA	Number Input	0 to 10, with step = 0.01
Research	Select Box	"Yes", "No"

Table 4. User Input Table

The user input prediction page design is shown below:

The screenshot displays the 'Graduate Admission Predictor' web application. On the left, a sidebar contains a 'Choose the page' dropdown menu with 'Predictive system' selected. The main content area features a title 'Graduate Admission Predictor' and a series of input controls: 'GRE Score' (slider from 260 to 340, set at 300), 'TOEFL Score' (slider from 0 to 120, set at 100), 'Course Ranking' (dropdown menu set to 5), 'SOP' (slider from 0 to 5, set at 4), 'LOR' (slider from 0 to 5, set at 5), 'Enter CGPA' (text input field with '8.00' and step arrows), and 'Have student conducted a research before?' (dropdown menu set to 'Yes'). A 'Predict' button is located at the bottom of the form.

Figure 22. Prediction page design

After our target user - the academic officer, inputs students' data, there is a 'Predict' button. After pressing the button user got prediction results, either made by regression or classification algorithms.

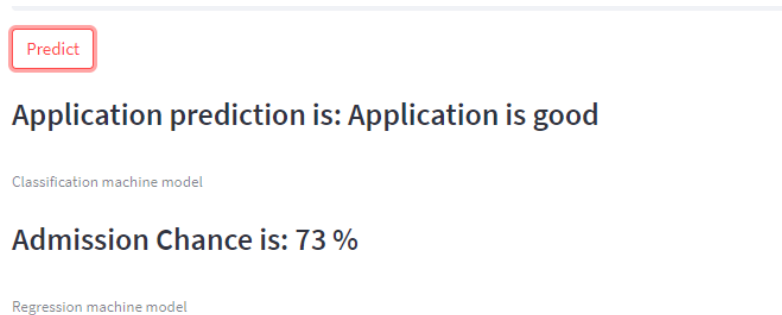


Figure 23. Predictions in Graphic User Interface

Chapter 5. Discussion

It should be stated that the proposed prototype of the recommender system for the admission office at Anglia Ruskin University is supposed to increase the department's efficiency, and the research is considered to make the used machine learning algorithms for the proposed recommender system more transparent. Implementing the recommender system at Anglia Ruskin University machine learning model should be retrained with the university's data; the script should not be changed much, as the used public dataset was assumed to be the closest one to the real Anglia Ruskin University dataset.

Two similar approaches were mentioned in the literature review section: the graduate recommender system and performance predictor approaches. The research about the graduate recommender system was working with a similar dataset. Still, the researcher only used a regression model to identify the success probability for student admission, while both classification and regression algorithms were used and compared in this research. The research about the performance

prediction system that used personal data, such as gender, sex, country, etc.; it is not suitable for the admission officer to look at the personal data for considering a student to be admitted, so the dataset for the given research was strictly stuck to student's academic performance.

Chapter 6. Conclusion

In summary, this research was conducted to fill the research gaps in creating unique recommender systems for universities. It was supposed to answer the main research question: what are the most efficient machine learning algorithms for predicting student admission to the university?

Among seven trained classification machine learning models – the logistic regression algorithm was the most efficient. The dataset logistic regression algorithm for graduate admission showed a 0.92 accuracy score. That means that a prediction system based on the logistic regression algorithm for predicting if the student will be successfully admitted or not will predict correct values in 92% of the cases.

Among seven trained regression machine learning models – The Extra Trees Regression algorithm is considered the most efficient algorithm being trained on the graduate admission dataset. The machine learning model trained on the Extra Trees Regression algorithm will predict the probability of successful student application. The predictions of the given regression algorithm will be between 0% and 100%. The model trained on the Extra Trees Regression algorithm showed the least Mean Squared Error - 0.074.

Both classification and regression machine learning algorithms were included in the web prototype, showing the results on the prediction page. That means that Anglia Ruskin University has a choice on which kind of prediction they need. If the

university needs to classify the student's applications into two groups, successful and not successful, they can use a classification algorithm. If the university needs to predict the student's acceptance probability, it can use the regression algorithm. It should be stated that both models are suitable for making early predictions.

In addition to the machine web application using machine learning models, exploratory data analysis was conducted. It was discovered that.

Investigation of how to implement the database into the application so that the new data can train the model simultaneously should be considered for future work. The web application can also be changed; the functionality of uploading a dataset with a high correlation between CGPA, TOEFL and GRE results. This observation shows that students getting higher marks in their bachelor's degree should get higher results in GRE and TOEFL examinations and vice versa.

No correlation was explored between CGPA and Letter of Recommendation/ Statement of Purpose. That means that students who got lower marks in their bachelor's degree can write a good statement of purpose or receive a good recommendation letter from their university tutors.

Applications of prospective students can be added. This web application improvement could make the department even more efficient. Moreover, more interactive dashboards with exploratory data analysis could be implemented, but it is still not the priority for the given project.

Bibliography

1. Adomavicius, G. and Tuzhilin, A., 2005. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), pp.734-749.
2. El Guabassi, I., Bousalem, Z., Marah, R. and Qazdar, A., 2021. A Recommender System for Predicting Students' Admission to a Graduate Program using Machine Learning Algorithms. *International Journal of Online and Biomedical Engineering (iJOE)*, 17(02), p.135.
3. Gupta, A. and Turek, J., 2015. An empirical investigation of predictors of success in an MBA programme. *Education + Training*, 57(3), pp.279-289.
4. M, H. and M.N, S., 2015. A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2), pp.01-11.
5. Portugal, I., Alencar, P. and Cowan, D., 2018. The use of machine learning algorithms in recommender systems: A systematic review. *Expert Systems with Applications*, 97, pp.205-227.
6. Raschka, S., 2018. Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning. [online] Available at: <<https://doi.org/10.48550/arXiv.1811.12808>>.
7. Walczak, S., 1994. Categorising university student applicants with neural networks. *International Conference on Neural Networks (ICNN'94)* (pp. 3680-3685). IEEE, Orlando, FL, USA)
8. Zhao, Y., Xu, Q., Chen, M. and Weiss, G., 2020. Predicting Student Performance in a Master of Data Science Program using Admissions Data. *Proceedings of the 13th International Conference on Educational Data Mining*, pp.325 - 333.