Prediction of College Admission

1st Tanvi Venkatesh
EE 695 Machine Learning
Stevens Institute of Technology
Hoboken, USA
tvenkat2@stevens.edu

2nd Jessica Kamman
EE 695 Machine Learning
Stevens Institute of Technology
Hoboken, USA
jkamman@stevens.edu

3rd Priyank Sanghvi
EE 695 Machine Learning
Stevens Institute of Technology
Hoboken, USA
psanghv1@stevens.edu

***Abstract*—The issue of student admittance is critical in educational institutions. This study discusses machine learning methods for predicting a student's chances of admission to a master's degree. This will allow students to know ahead of time if they have a possibility of being admitted. Machine learning models include linear regression,Logistic Regression, Decision Tree and random forest.**

I. Introduction

Machine Learning is a branch of computer science that involves programming machines (computers, for example) to learn from data. It may be roughly described as computational approaches that use experience to enhance performance or generate accurate predictions. Machine learning is a sub-field of Artificial Intelligence that analyses large data-sets using algorithms. In a word, Machine Learning is the process of creating models that accurately anticipate the outcome based on the input data. It allows machines to increase their accuracy as more data is fed into the system using statistical approaches. This model was developed to forecast the progress of prospective students by comparing the score of students currently studying at university. The model thus predicted whether the aspiring student should be admitted to university on the basis of various scores of students. Inspecting feature values that help identify what needs to be done to clean or pre-process until you see the range or distribution of values typical of each attribute. Then, machine learning techniques like logistic regression, decision tree and random forest algorithm are used.

The aim of this project is to develop the best machine learning model to help a student predict their chances of admissions to a specific master's program. This prediction could helps students in the future to make a more effective decision of applying to a certain school and Master's Program based on their chance of admission. Various factors will be considered related to the student such as academic achievements, work experiences personal background and chance of scholarship, GRE-Scores and TOEFL will all be taken as factors to predict the chance of admission.

The data set that will be used for this project was influenced by the UCLA Graduate Data set and aimed at assisting students in narrowing their university choices based on their profiles. The data set contains several parameters that are considered important during the application process for the Master program. By predicting the likely hood of admissions to a particular universities master program, the student gains insights to their chances of acceptance and a student can use this to tailor their application process based on their prediction of likely hood for admittance to that particular college. The data set includes the Chance of Admit Variable which based on this research done by UCLA Graduate , was conducted by asking how likely they would think of getting admitted to the particular Master Program.

The machine learning algorithms that will be used for the prediction of Admission for a Master's Programs are, Logistic Regression, Random Forest and Decision Tree.

As stated previously, currently there is just limited contributions out on the prediction of Admissions. Others have tried to answer this question solely with relying on the Logistic Regression, however with the data that is given, Logistic Regression alone is not the best algorithm to use. In our solution, we are using multiple algorithms to answer this problem of chance of admission.

II. Related Work

There are only a few similar existing problems that try to predict college admission, based on GRE-Scores and TOEFL-Scores. In those other existing solutions, the author uses similar data, but only tries to answer the problem with a Linear Regression model, no other models are implemented. Aside from that, we have added a few more data points which include the chance of scholarship as another independent variable that will be used in the prediction of the chance of admission. The existing work can be found here: https://www.kaggle.com/code/gireeshs/which-college-are-you-likely-to-get-into.

The related work that is currently available is very limited, and leaves a lot of gaps by not considering other Machine Learning Algorithms. Without implementing more than a Regression model, it does not necessarily tell us how accurate the prediction really is. The data-set only has one university that it tries to predict this chance on rather that numerous college, the Chance of Admit, instead of it being a true value, it is was a student prediction if they were going to be admitted. Based on the data , we are predicting on their own predictions, and due to that our version is much more accurate by incorporated multiple algorithms instead of just staying with Linear Regressions. The related work found on Kaggle also doesn't try to implement any type of accuracy scores, and without comparing it to another algorithm or accuracy scores one cannot be confident if the prediction is accurate or not. Our solution has a clear absolute error score for Decision Tree, and accuracy comparisons of Logistic regression and Random forest, which tells us what the error is of prediction.

III. Our Solution

The Solution of the problem has been implemented via Random Forest, Logistic regression and Decision tree. The following subsections will talk about in details everything pertaining to our Solution.

A. Description of Dataset

The data set contains several parameters which are considered important during the application for Masters Programs. The creation of this data set was influenced by the UCLA Graduate Data set and aimed at assisting students in narrowing down their university choices based on their profiles. By predicting their likelihood of admission to a particular university, students can gain valuable insights into their chances of being accepted. The data set has the following parameters that will be used:

GRE Scores (out of 340)

TOEFL Scores (out of 120)

University Rating (out of 5)

Statement of Purpose and Letter of Recommendation strength (out of 5)

Undergraduate GPA (out of 10)

Research Experience (either 0 or 1)

Chance of Admit (ranging from 0 to 1)

Chance of Scholarship (0 or 1)

Pre-Processing

For Pre-Processing step parameters were verified in the dataset to ensure we had enough data. As a part of the pre-processing steps we added the column for chance of Scholarship. We then theorized how we wanted to assess the data and how to tackle our problem statement. Part of the pre-processing Step the following steps where done:

1.) Import all necessary packages that were needed for all algorithms, and graphs.

2.) Check for any null data values in the data set. If there are any update the null values accordingly.

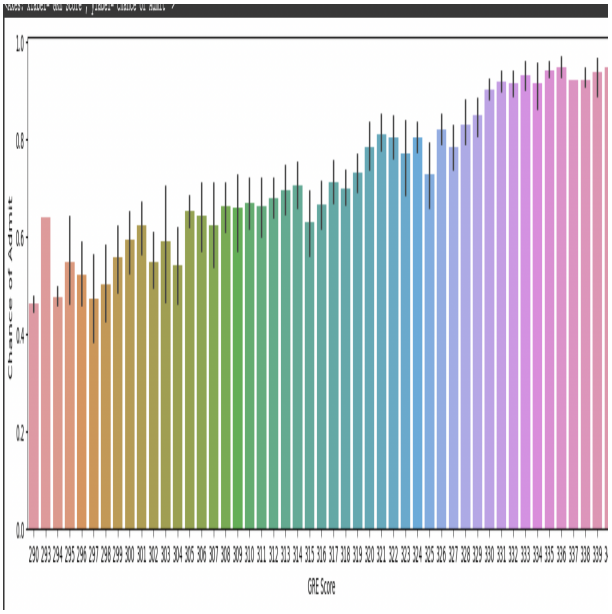
3.)Encode all categorical parameters.

4.) Check covariance of the parameters.

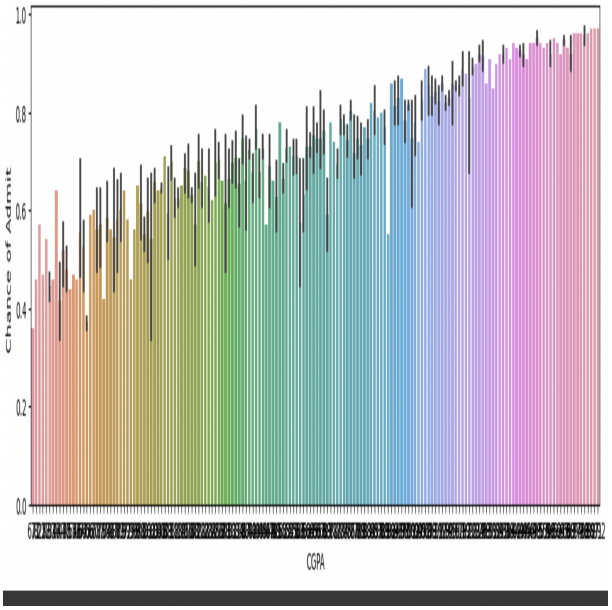
5.) Categorize the importance of all parameters, this will help learn more about the relationship of the parameters between each other.

6.) Visualization of the Data.

The below diagram shows the Chance of Admit with GRE Scores, the black lines indicate when a student is admitted to a University.



The below bar graph sows the Chance of Admit for CGPA, the black lines indicated when a student is admitted to a University



B. Machine Learning Algorithms

This subsection describes machine learning algorithms that you plan to use. For each ML algorithm, briefly 1) explain why it might be appropriate for the problem and 2) describe your main design. For example, if it is neural network, provide the network structure and your initial choice of some key parameters (e.g., activation function to use, number of layers, number of hidden nodes of each layer). You may change the parameters during the training process.

In this project we will be utilizing three different Machine Learning Algorithms in order to answer our problem of prediction of admission.

Decision Tree Models are a non-parametrics supervised learning method that is often used for classification and

regression. The main goal of this algorithm is to create a model that predicts the value of a target variable by inferring a decision from the data features. A tree can be modeled to see an approximation. The Decision Tree algorithm requires a little bit of data preparation such as analyzing how many data points will be used to train the tree. Decision Trees can be used for both numerical and categorical data. This algorithm performs well even if we have made assumptions that somewhat violated the true model, from which the data was generated. Decision Tree can have some disadvantages such as creating and overly complex tree that does not generalize the data well, this phenomena can also be called over fitting. One way to minimize this, one can prune the tree, setting the minimum numbers of samples required at a leaf node, or setting the maximum depth of the tree.

Logistic Regression, this is a suitable algorithm for our problem because it is used for binary outcomes, so in this case we are looking for a yes or no answer if a certain student would be accepted in the university master program. In our problem statement our relationship we want to solve for is the chance of admission, and that is being predicted by numerous different independent variables, such as the students GRE-Scores, TOEFL Scores, Grades and many more. We have a few different variables that are based on binary values and due to that Logistic Regression would be another suitable algorithms that could be compared to Linear Regression to show us prediction for admission. The provided code snippet uses the scikit-learn Logistic Regression class to train a logistic regression model. The provided code snippet uses the scikit-learn Logistic Regression class to train a logistic regression model. Fit the model to the training data. After training, the model is used to make predictions on the test data by calling predict function.

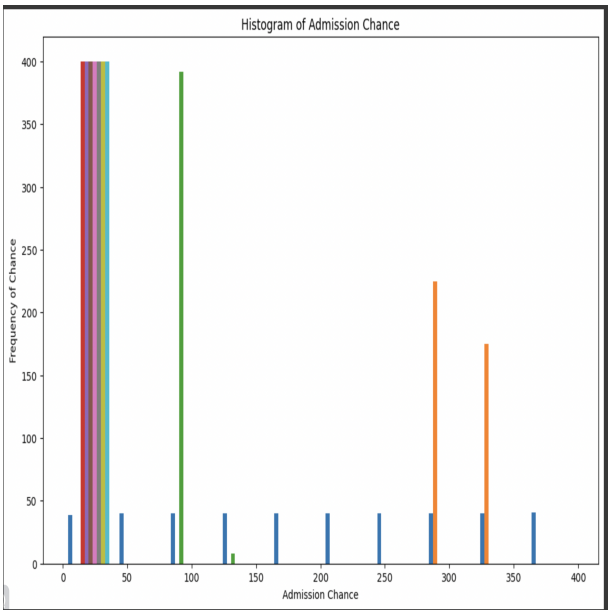
Random Forest, which is a Supervised Learning Algorithm which is used in most Classification and Regression problems. Random Forest often is used with the conjunction of continuous variables, and categorical variables. Random Forest often performs better for classification and regression problems. Since we have a few different categorical parameters this would be a perfect fit for Random Forest. Some of the parameters that we will be using during this algorithms are max features, max leaf nodes, and min leaf nodes to insure our predictive power is more accurate. We will also use random state with controls the randomness of the sample, this will allow us always to get the same result.

C. Implementation Details

This subsection describes details of your implementation. Please focus on how you test and validate the performance, tune the hyperparameters, and select the best-performing models. Elaborate on techniques that you apply to improve the performance and explain why you use these techniques. You include few most important results/figures to illustrate your idea but do not let figures/tables dominate the content of the report. You can include few lines of critical code if needed. But please avoid paste lengthy code in your report. Please make sure the figures/tables/code snapshots are of appropriate size including the font size.

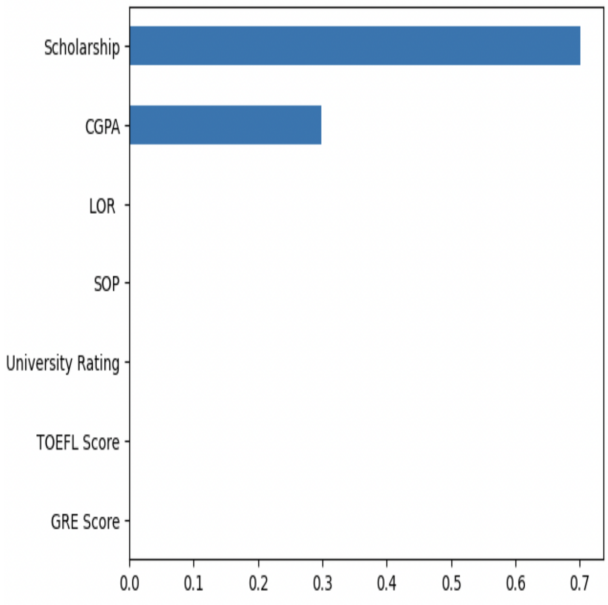
It is necessary for us to understand the data in order to do so. First uploading data set into Jupyter notebook and then creating a folder for the data set and workspace. Moving into the workspace use the commands to read the data The data must be cleaned and sorted, to get better understanding of the data and also to make the prediction efficient. To, clean the data we first need to see if any null values are present, find the data type, draw irrelevant columns. If null values are present they can be removed. It is a crucial step as it makes the data ready for prediction and working.

In order to get into depth of the data to get the best of the prediction, graphs and plots can be drawn. Here, the chance of admit is compared to every aspect of students profile and presented graphically. We also, have drawn a histogram to plot the frequency of chance. This data set consists of 400 rows, and we split the data dependent and independent variables. And further split into train and test sets 80 percent of data for training and 20 percent for testing. Hence we have trained and split the data. To verify is the splitting is performed correct the shape can be used. The data is split into X train, X test, y train and y test.

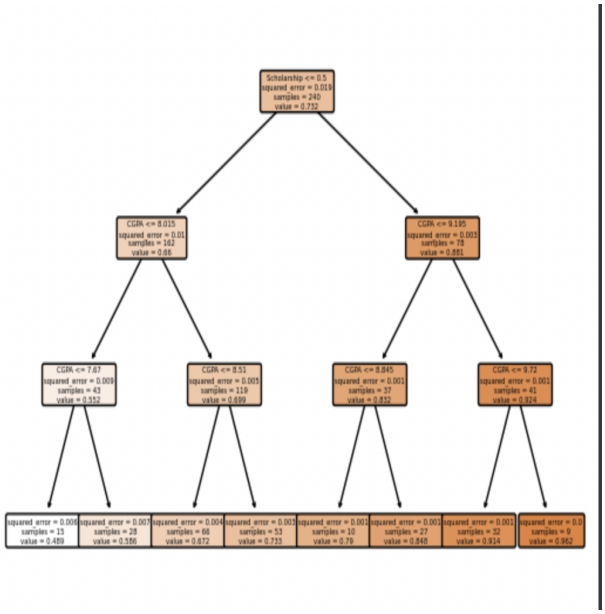


Decision tree Implementation the data is split to a specific percentage within training data and testing data. Various Parameters are being used in the Decision Tree regressor such as Max depth, mini sample split, splitter, min sample leaf, and criterion- squared error, friedman mse, and poission. Once this is done GridSearchCV is used on the params grid, to tell us which parameters are best fit. Once this tells us what the best fit is the DecisionTreeRegressor function is used to give us the sample. At that point, the score for X train, Ytrain, and X test and Ytest can be retrieved, and to further elaborate on our Decision Tree model the mean absolute percentage of error has been added to tell us how accurate this model actually is. When working on the Decision Tree implementation it was started with a train split set of 80/20 but it came apparent very fast that when doing it with 80/20 our model score for train and test had a very large spread in scores, which can indicate our model is overfitting. In order to minimize the potential overfitting of the model, the split was updated to 60/20 due to the fact that our dataset is fairly small at 400 entries. However, event though the training split was updated

the model was still overfitting. Additonal parameters were added to the param grid in order to improve the performance and to try to narrow the spread of the train and test score. In the beginning of the Decision Tree implementation the param grid only included max depth, min samples splitter and min samples. However, when this was all the spread was even larger, criterion squared error , friedman mse and poisson was added but this did not have a big impact on the scores. Last, max depth was updated to adding 3,4,5,6 and min samples was increase to 2,3,4,5,6,7,8 which improved the scores a tiny bit. When working on the Decision Tree, feature importance was also ran, which showed that the only variables that are of importance is Scholarship and CGPA. CGPA is the college Undergraduate GPA, this correlation makes sense, that when someone receives a Scholarship we expect them to have a higher Chance of Admit. In addition the GPA has a high feature important to be chance of admit. It was fairly eye opening, finding out that the TOEFL and GRE Scores do not hold any importance.



The final Result of the implementation of Decision Tree has the following outcome:

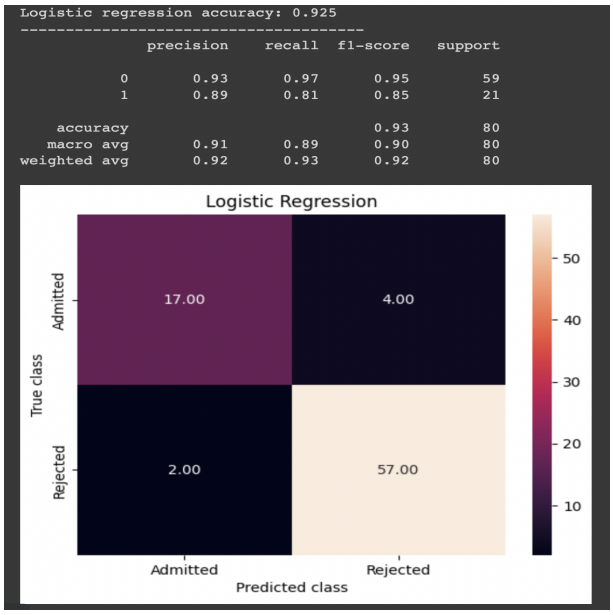


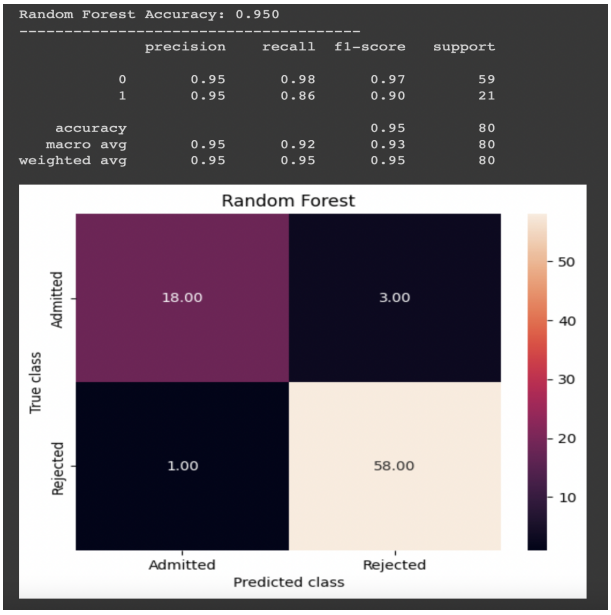
Logistic Regression Logistic regression, which is well-suited for binary outcomes. In our case, we are seeking a yes or no answer regarding the acceptance of a student into the university master's program. Our objective is to predict the likelihood of admission, which is influenced by various independent variables such as the student's GRE scores, TOEFL scores, grades, and others. Considering the presence of binary variables in our dataset, logistic regression serves as an appropriate alternative to linear regression, enabling us to obtain admission predictions.

Regarding the logistic regression results, the model achieved an accuracy of 0.925. The precision for class 0 (not admitted) was 0.93, indicating that 93 percent of the predicted not admitted instances were indeed correct. The recall for class 0 was 0.97, indicating that the model correctly identified 97 percent of the not admitted instances. The f1-score for class 0 was 0.95, representing the balanced harmonic mean of precision and recall for class 0. These performance metrics suggest that the logistic regression model performed well in predicting the not admitted class.

For class 1 (admitted), the precision was 0.89, indicating that 89 percent of the predicted admitted instances were correct. The recall for class 1 was 0.81, indicating that the model correctly identified 81 percent of the admitted instances. The f1-score for class 1 was 0.85, representing the balanced harmonic mean of precision and recall for class 1. While the model's performance for the admitted class is slightly lower than that of the not admitted class, it still demonstrates reasonably good predictive capability.

Overall, the logistic regression model achieved an accuracy of 0.93, with a macro average f1-score of 0.90, indicating the overall effectiveness of the model in capturing the relationship between the independent variables and the chance of admission. The weighted average f1-score of 0.92 further emphasizes the model's balanced performance across both classes.





Random Forest Data Splitting: The code first splits the dataset into training and test sets using the train test split function. The features (X) and target variable (y) are split into X train, X test, y train, and y test, with a test size of 20 percent of the data. The random state parameter is set to 1 to ensure reproductibility.

Model Initialization and Training: The random forest regression model is imported from scikit-learn's RandomForestRegressor class. An instance of the model is created with 1000 decision trees by setting the n estimators parameter to 1000. The random state parameter is set to 42 for reproductibility. The model is trained on the training data using the fit method .

Prediction and Error Calculation: The trained random forest regression model is used to predict the target variable for the test data by calling predict function. The predictions are stored in the predictions variable. The absolute errors between the predictions and the true values (y test) are calculated by subtracting y test from predictions and taking the absolute value.

Mean Absolute Error (MAE) Calculation: The mean absolute error (MAE) is calculated by taking the average of the absolute errors using np.mean(errors). The MAE represents the average magnitude of the errors made by the model.

Mean Absolute Percentage Error (MAPE) Calculation and Accuracy Calculation: The mean absolute percentage error (MAPE) is calculated by dividing the absolute errors (errors) by ytest and multiplying by 100 to get the percentage. The MAPE indicates the average percentage difference between the predictions and the true values. The accuracy is then calculated as 100 minus the mean MAPE, representing the average percentage of accuracy achieved by the model.

The MAE is printed using print('Mean Absolute Error:', round(np.mean(errors), 2), 'degrees.'). The accuracy is printed using print Accuracy, which is 91.88 percent.

IV. Comparison

This code demonstrates the use of scikit-learn's LogisticRegression class. This is commonly used in binary classification problems. This code demonstrates the use of scikit-learn's RandomForestRegressor class used for regression tasks. The logistic regression model used in the code has default parameters because no additional parameters were specified.

A random forest regression model is instantiated with 1000 decision trees and 42 random states. Training and prediction:

A logistic regression model is trained on the training data using the fit method. Next, use the trained model to predict labels for the test data.

A random forest regression model is trained on the training data using the fit method. The accuracy of the logistic regression model's predictions is evaluated using the accuracy evaluation function and printed as output.

The code provided does not evaluate or calculate the accuracy of the random forest regression model. In summary, the provided code demonstrates training and evaluation of a logistic regression model, but lacks the evaluation part of a random forest regression model. Comparisons between the two models based on the code provided are limited to using logistic regression for classification tasks and random forest regression for regression tasks.

V. Future Directions

If we were given an extra 3-6 months one of the future directions that would be taken is to gather more data. The data is very limited, and the way the data was conducted is not a great way to predict the admissions to an actual master program. Better data is needed that actual shows true data for various universities, and who was admitted versus who was not. The data at this point is gather from UCLA on a Chance of Admit. This Chance of Admit is the particular person theorizing what their chance is rather than a true yes or no admission. In the future once the data has been collected for various Universities, the models would be predicting on

actual data not just on a student theorizing on their chances. This could improve our accuracy of prediction and potentially lower our absolute error score.

VI. Conclusion

Last but not the least, don't forget to include references to any work you mentioned in the report.

The main goal of this work is to create a Machine Learning model which could be used by students who want to pursue their education. Many machine learning algorithms were utilized for this research. Logistic Regression model compared to other ones. Students can use the model to assess their chances of getting admission into a particular university with an average accuracy of 92 percent with random forest. The ultimate goal of research will be accomplished successfully, as the system allows students to save the lot of time and money that they would spend on educational mentors and application fees for colleges where they have less chances of getting admissions.

Finally, students can have an open-source machine Learning model which will help the students to know their chance of admission into a particular university with high accuracy.

Based on the data that was given by UCLA, I believe that our prediction score of 92 is a fairly well prediction score. Our data-set only has 400 entries, with a limited variable size, and the chance of admit is theorized instead of actual data. Based on the theoretical chance of admit, we do not know how pessimistic someone really is on the likely hood of them getting admitted, but their is a high correlation with Chance of Admit and Scholarship. For the most part when a student received a scholarship, the chance of admit was high. This could lead to a fairly biased prediction. For this type of problem the is one of the best algorithms, Random forest and Logistic Regression can answer or give insight in the problem, but not necessarily answer it well. If the data was an actual depiction of being admitted, Random forest and Logistic Regression would look much different.

References

- N. Gupta, A. Sawhney, and D. Roth, "Will i Get in? Modeling the Graduate Admission Process for American Universities," IEEE Int. Conf. Data Min. Work. ICDMW, vol. 0, pp. 631–638, 2016.
- A. Waters and R. Miikkulainen, "GRADE : Graduate Admissions," pp. 64–75, 2014.
- S. Sujay, "Supervised Machine Learning Modelling Analysis for Graduate Admission Prediction," vol. 7, no. 4, pp. 5–7, 2020.
- F. Salo, M. Injadat, A. Moubayed, A. B. Nassif, and A. Essex, "Clustering Enabled Classification using Ensemble Feature Selection for Intrusion Detection," in 2019 International Conference on Computing, Networking and Communications (ICNC), 2019, pp. 276–281
- A. Moubayed, M. Injadat, A. B. Nassif, H. Lutfiyya, and A. Shami, "E-Learning: Challenges and Research Opportunities Using Machine Learning Data Analytics," IEEE Access, 2018
- https://www.kaggle.com/datasets/mohansacharya/graduate-admissions