# Indian Institute of Information Technology Surat


# Project Report on
## University Admission Prediction using Machine Learning

**Submitted by**

**GAGANDEEP BHARDWAJ (UI21CS76)**

**Course Faculty**

**Dr. Pradeep Kumar Roy**

**Mr. Vipul Kania**

**Department of Computer Science and Engineering**

**Indian Institute of Information Technology Surat**

**Gujarat-394190, India**

**Academic Year - 2024-25**

# INTRODUCTION

In today’s world, it is getting more and more difficult to get into prestigious universities in today's competitive academic environment. The difficult selection criteria, which include letters of recommendation, personal statements, standardized test scores, and academic accomplishments, must be met by applicants. There is frequently close rivalry during this process as candidates try to establish themselves as the most qualified and worthy ones.

The main change in the university admissions process has occurred with the introduction of machine learning techniques. Massive data analysis and insight extraction from machine learning algorithms have proven to be highly capable. These algorithms can accurately predict admission outcomes by utilizing predictive analytics, which will transform the decision-making process for both academic institutions and applicants.

This project's main goal is to create a reliable model that can accurately predict an applicant's chances of getting accepted into a university by utilizing the power of machine learning and predictive analytics. The model seeks to provide a thorough evaluation of an applicant's profile by combining a number of variables, including GRE and TOEFL scores, Statements of Purpose (SOP), Letters of Recommendation (LOR), and Cumulative Grade Point Average (CGPA).

With this project, we hope to improve the overall effectiveness and equity of the admissions process while addressing a number of important issues. Academic institutions can ensure a more transparent and merit-based approach to candidate selection by utilizing data-driven insights to make better-informed decisions. Furthermore, candidates stand to gain from having a greater understanding of their chances of admission, which will help them make wise choices and maximize their application methods.

Predictive analytics' collaboration into the admissions process also fits in with the industry's larger trend of digital transformation in the education sector. Universities can improve decision-making, simplify operations, and improve the overall experience for stakeholders and students by adopting modern technologies.

To put it simply, this project is a big step towards optimizing the university admissions process through the use of data-driven approaches. By making use of machine learning expertise and the complexities of the admissions domain, our goal is to develop a model that not only accurately predicts admission outcomes but also builds efficiency, fairness, and transparency within the academic ecosystem.

# BACKGROUND

Recent years have seen massive changes in the higher education landscape, as seen by an increase in the quantity and variety of applications that universities across the globe have been receiving. Applications are coming in from a wide range of backgrounds, experiences, and places, which reflects the expanding goals and global perspectives of students.

University admissions committees have been faced with the difficult task of evaluating a diverse pool of applicants as a result of this influx. These committees have historically evaluated candidates' qualifications and likelihood of academic success using a combination of qualitative and quantitative measures.

In terms of quality, letters of recommendation (LOR) and statements of purpose (SOP) have been essential in revealing information about the character, motives, and suitability of applicants for the academic programmes they have selected. LORs provide individualized viewpoints on an applicant's accomplishments, strengths, and potential contributions to the academic community. They are frequently authored by mentors, instructors, or employers. Similar to this, SOPs give applicants the chance to express their passion for their chosen field of study, highlight their experiences and accomplishments, and explain their academic and professional goals.

Quantitative measures that act as standardized benchmarks for evaluating academic readiness and aptitude are used in addition to these qualitative evaluations. Test results from standardized assessments, like the Graduate Record Examinations (GRE) and the Test of English as a Foreign Language (TOEFL), offer numerical representations of an applicant's capacity for analysis, language ability, and preparation for graduate-level courses. Admissions committees frequently use these scores to assess applicants' academic readiness and likelihood of success in demanding academic settings.

An exact indicator of an applicant's academic achievement during their undergraduate or before higher education is the Cumulative Grade Point Average (CGPA). Admissions committees place a high value on factors such as mastery of the subject matter and consistent academic excellence, both of which are indicated by a high CGPA.

Higher education's admissions procedure is essentially a complex evaluation framework that impacts a balance between qualitative observations and numerical measurements. Through the use of a comprehensive approach, universities are able to recognise and accept applicants who not only have excellent academic records but also exhibit the drive, aspiration, and capacity to succeed in their studies and make significant contributions to the academic community.

# EXISTING SOLUTION

The current method for solving the university admission prediction problem usually consists of a crude method that calculates an admission probability based on a percentage without going into the finer points of the dataset features. For the purpose of making predictions, this solution frequently only uses linear regression, a fundamental statistical modeling technique. Although this method may provide a broad estimate of admission probabilities, it falls short in terms of precision and depth when compared to more sophisticated machine learning algorithms and a thorough comprehension of the features of the dataset.

A fundamental statistical technique called linear regression is used to model the relationship between one or more independent variables (like GPA, TOEFL, SOP, LOR, and CGPA) and a dependent variable (in this case, admission probability). In order to fit a straight line that best captures the general trend of the data points, the model assumes a linear relationship between the input features and the target variable.

But there are drawbacks to linear regression, particularly when working with large, complicated datasets that have nonlinear relationships between variables. When it comes to predicting university admission, linear regression is not sufficient to account for variables like the interaction between various application components (e.g., a strong SOP making up for a slightly lower GRE score) and the variable significance of features for different programmes or universities.

Furthermore, the clarity and usefulness of the current solution are restricted for both applicants and academic institutions due to its concentration on presenting a percentage-based chance of admission without explaining the underlying factors or dataset features. While applicants may be given a numerical likelihood of admission, they cannot make strategic improvements to their chances or make informed decisions unless they understand how various aspects of their profile contribute to this probability.

Moreover, the model's exclusive dependence on linear regression and the dataset's lack of transparency about its features miss potential patterns and insights that could be discovered through the use of more complex algorithms. In order to capture complex relationships and nonlinearities within the data, advanced machine learning techniques like decision trees, random forests, support vector machines (SVM), and gradient boosting models offer greater flexibility and accuracy.

The drawbacks of the current solution can be addressed by implementing these modern algorithms for machine learning and taking a more thorough approach that takes into account the complex relationships among dataset features.

# PROPOSED IDEA

By utilizing a variety of machine learning algorithms and offering in-depth analyses of the effects of each feature in the dataset, the suggested solution to the university admission prediction problem constitutes an important upgrade over the current methodology. With this suggested solution, applicants and academic institutions should be able to make better decisions regarding predictive accuracy, model understanding, and decision-making skills.

**Utilization of Multiple Machine Learning Algorithms:**

The use of several machine learning algorithms, such as Support Vector Machines (SVM), Decision Trees, and Random Forests, is one of the main features of the suggested solution. A more reliable and accurate prediction of admission probabilities is made possible by the distinct strengths and abilities that each algorithm contributes to the predictive modeling process.

1.**Support Vector Machines (SVM)**: the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called a hyperplane. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called the margin. And the goal of SVM is to maximize this margin. The hyperplane with maximum margin is called the optimal hyperplane.

2.**Decision Trees**: Decision trees are intuitive models that mimic human decision-making processes by partitioning the data based on feature thresholds. Each node in the tree represents a decision based on a feature, leading to a hierarchical structure that can capture nonlinear relationships and interactions among features.

3.**Random Forests:** Random forests improve upon decision trees by combining multiple trees into an ensemble model. Each tree is trained on a random subset of the data and features, reducing overfitting and enhancing predictive accuracy. Random forests also provide feature importance rankings, indicating the relative impact of each feature on the prediction outcomes.

4. **Logistic regression** : Logistic regression is a statistical model commonly used for binary classification tasks, where the target variable has two possible outcomes (e.g., yes/no, pass/fail, admitted/not admitted). Despite its name, logistic regression is a classification algorithm, not a regression algorithm like linear regression.

**Impact of Each Feature in the Dataset:**

The suggested approach not only makes use of various machine learning algorithms but also examines how each feature in the dataset affects admission predictions. Understanding which factors have a significant impact on admission decisions and how applicants can strategically improve their profiles to increase their chances of being accepted are made possible by this analysis.

1. **GRE Score** : A standardized test, the Graduate Record Examinations (GRE) score evaluates analytical writing, verbal reasoning, and quantitative reasoning abilities. Admissions committees value strong analytical and academic aptitude, which is typically indicated by a high GRE score.

2. **TOEFL Score** : An applicant's English proficiency, which is necessary for success in English-language academic programmes, is measured by their score on the Test of English as a Foreign Language (TOEFL). A strong TOEFL score enhances one's chances of admission by demonstrating language competency and communication abilities.

3. **Statements of Purpose (SOP) :** SOPs give applicants the chance to explain their unique experiences and qualifications, as well as their academic and professional aspirations and reasons for pursuing a particular programme. Admissions decisions can be positively impacted by an applicant's compelling SOP, which shows their passion and commitment with the mission and values of the university.

4. **Letters of Recommendation (LOR) :** LORs provide information about an applicant's personality, aptitude for the classroom, work ethic, and likelihood of succeeding in a graduate programme. Credible recommenders' strong endorsements can greatly increase an applicant's credibility and suitability for admission.

5. **Cumulative Grade Point Average (CGPA) :** An applicant's total academic achievement during their undergraduate or prior academic pursuits is reflected in their CGPA. A high CGPA is an important criterion in admissions evaluations because it demonstrates a student's consistent academic excellence and subject matter mastery.

**Libraries Used :**

1. **os** : The `os` library provides functions for interacting with the operating system, such as navigating directories, checking file existence, and accessing environment variables.

```
import os
```

2. **numpy (np)**: NumPy is a fundamental library for numerical computing in Python. It provides support for multi-dimensional arrays, mathematical functions, linear algebra operations, random number generation, and more, making it essential for data manipulation and scientific computing tasks.

```
import numpy as np
```

3. **pandas (pd)**: Pandas is a powerful data manipulation library that provides data structures like DataFrames and Series, along with functions for data cleaning, manipulation, merging, grouping, and analysis. It's commonly used for data preprocessing and exploratory data analysis (EDA).

```
import pandas as pd
```

4. **matplotlib.pyplot (plt)**: Matplotlib is a plotting library for creating static, interactive, and animated visualizations in Python. The `pyplot` module provides a MATLAB-like interface for creating plots, histograms, scatter plots, bar charts, and more, making it suitable for data visualization tasks.

```
import matplotlib.pyplot as plt
```

5. **seaborn (sns)**: Seaborn is a statistical data visualization library built on top of Matplotlib. It provides high-level functions for creating visually appealing statistical plots, such as heatmaps, violin plots, pair plots, and regression plots, enhancing the aesthetics and readability of plots.

```
import seaborn as sns
```

6. **pickle**: Pickle is a module for serializing and deserializing Python objects. It allows you to save Python objects (e.g., trained machine learning models) to disk and load them back into memory, preserving their state and structure.

```
import pickle
```

7. **StandardScaler**: StandardScaler is a preprocessing class from scikit-learn used for standardizing numerical features by removing the mean and scaling to unit variance. It's commonly used in machine learning pipelines to ensure consistent scaling of features.

```
from sklearn.preprocessing import StandardScaler
```

8. **RandomForestRegressor**: RandomForestRegressor is an ensemble learning method from scikit-learn that fits multiple decision tree regressors on random subsets of the data and averages

their predictions. It's used for regression tasks and is known for handling nonlinearity and capturing complex relationships in data.

```
from sklearn.ensemble import RandomForestRegressor
```

9. **DecisionTreeRegressor**: DecisionTreeRegressor is a decision tree-based regression model from scikit-learn that predicts continuous target variables by recursively partitioning the data into subsets based on feature values. It's interpretable and suitable for capturing nonlinear relationships in data.

```
from sklearn.tree import DecisionTreeRegressor
```

10. **train_test_split**: train_test_split is a function from scikit-learn used for splitting datasets into training and testing sets. It's essential for evaluating machine learning models' performance on unseen data and preventing overfitting.

```
from sklearn.model_selection import train_test_split
```

11. **MinMaxScaler**: MinMaxScaler is a preprocessing class from scikit-learn used for scaling numerical features to a specified range, typically between 0 and 1. It's useful for algorithms sensitive to feature scaling, such as support vector machines (SVM) and neural networks.

```
from sklearn.preprocessing import MinMaxScaler
```

12. **LogisticRegression**: LogisticRegression is a linear model from scikit-learn used for binary classification tasks. It predicts the probability of an instance belonging to a particular class using a logistic function, making it suitable for probabilistic classification tasks.

```
from sklearn.linear_model import LogisticRegression
```

13. **confusion_matrix**: confusion_matrix is a function from scikit-learn used for evaluating classification model performance by generating a confusion matrix. It provides insights into true positive, true negative, false positive, and false negative predictions, enabling evaluation of model accuracy, precision, recall, and F1 score.

```
from sklearn.metrics import confusion_matrix
```

14. **precision_score, recall_score**: precision_score and recall_score are functions from scikit-learn used for calculating precision and recall metrics, respectively. They are useful for evaluating the performance of binary and multiclass classification models.

```
from sklearn.metrics import precision_score, recall_score
```

15. **SVC**: SVC (Support Vector Classifier) is a support vector machine-based classifier from scikit-learn used for binary and multiclass classification tasks. It finds the optimal hyperplane that separates classes in a high-dimensional space, making it effective for linear and nonlinear classification.

```
from sklearn.svm import SVC
```

16. **GaussianNB**: GaussianNB is a Naive Bayes classifier from scikit-learn used for classification tasks. It assumes that features are independent and follows a Gaussian (normal) distribution, making it suitable for probabilistic classification tasks.

```
from sklearn.naive_bayes import GaussianNB
```

17. **DecisionTreeClassifier**: DecisionTreeClassifier is a decision tree-based classifier from scikit-learn used for binary and multiclass classification tasks. It predicts class labels by partitioning the data based on feature values, making it interpretable and effective for capturing nonlinear decision boundaries.

```
from sklearn.tree import DecisionTreeClassifier
```

# IMPLEMENTATION

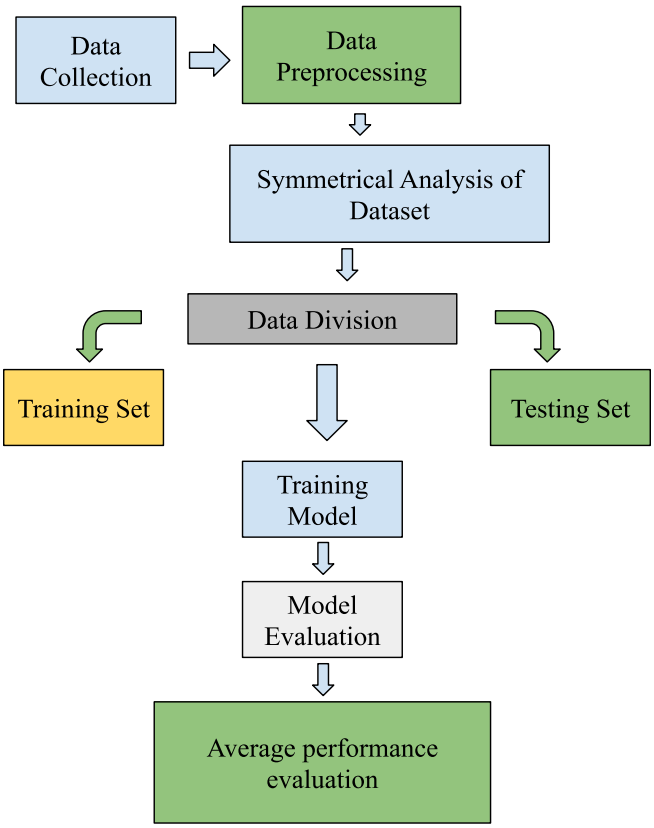


Figure 1 : Detailed flowchart of Project

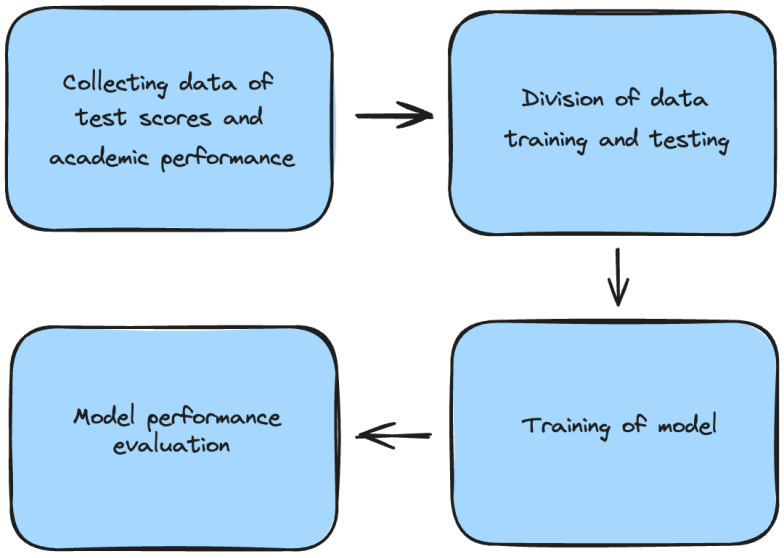


Figure 2 : Short flowchart displaying the project's functioning

**Data Collection** : Gathering relevant information about university admissions from a variety of sources. This contains information about a group of applicants' admission outcomes (admitted or not), as well as GRE and TOEFL scores, SOP, LOR, and CGPA. The process of gathering data guarantees that an extensive dataset is available for modeling and analysis.

Dataset - https://www.kaggle.com/code/pkulaba1290/starter-graduate-admissions-91c64936-a/input

**Data Preprocessing** : The cleaning and preparation of the dataset before analysis and modeling. I eliminated null values and the serial number column in this instance. In order to guarantee the consistency and quality of the dataset, it is essential to remove unnecessary columns and address any missing data.

**Dataset Analysis** : Analyzing a dataset includes looking into and understanding its properties. This includes looking at data distributions, feature correlations, possible outliers or anomalies, and descriptive statistics (mean, median, standard deviation). The objective is to learn more about the structure of the dataset and spot any patterns or trends that might have an impact on admission decisions.

**Dataset Division** : Separated the dataset into testing and training sets. The testing set is used for evaluating the performance of the machine learning models on untested data, whereas the training set is used to train the models. To guarantee proper training and evaluation, the dataset was split into 67% for the training set and 33% for the test.

**Training Model** : In order for the model to predict admission probabilities based on input features, it first needs to discover the basic trends and relationships in the data through training, Therefore I ran the model through multiple machine learning algorithms like Linear Regression , Support Vector Machines (SVM), Decision Trees, logistic Regression and Random Forests.

**Model Evaluation** : Model evaluation uses the testing data to evaluate how well the trained models perform. Evaluation measures like recall, accuracy, precision, F1 score, and confusion matrix are employed to evaluate how well the models forecast admission results. The evaluation process helps to evaluate the model's efficiency and recognises areas in need of improvement or optimisation.

# RESULT

In this project, we employed various machine learning algorithms to predict admission outcomes for prospective university applicants. The algorithms used include Linear Regression, Logistic Regression, Support Vector Machine (SVM), Gaussian Naive Bayes, Decision Tree Classification, and Random Forest Classification. We evaluated the performance of each algorithm using key metrics such as accuracy, precision, recall, and F1 score.

**Data Preparation and Model Training:**

1. We collected a dataset containing features such as GRE scores, TOEFL scores, SOP, LOR, CGPA, and admission outcomes (admitted or not admitted).
2. After preprocessing steps, including dropping irrelevant columns and handling missing values, we divided the dataset into training and testing sets (67% training, 33% testing).
3. Each algorithm was trained using the training set to learn patterns and relationships in the data.

**Model Evaluation and Prediction :**

Model evaluation means how accurately the machine learning algorithms predict admission outcomes (admitted or not admitted) based on features like GRE scores, TOEFL scores, SOP, LOR, and CGPA. Metrics such as accuracy, precision, recall, and F1 score are used to measure the model's performance. Prediction refers to using the trained models to predict whether a new applicant is likely to be admitted or not based on their input data.

In this model if a candidate's Chance of Admit is greater than 80%, the candidate will receive the 1 label otherwise it will be labeled as 0.

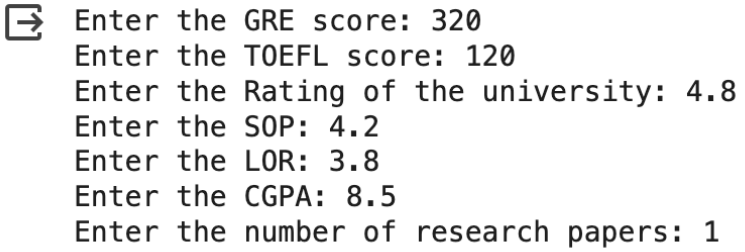


**Figure** - Input Features for calculating admission prediction

```
✓ 0s [31] print(f"Chances of Admission = {prediction[0]*100} %")

Chances of Admission = 78.34002296283704 %
```

**Figure -** Chances of Admission for the given input

**Algorithm Performance:**

1. **Linear Regression**: Linear regression, although primarily a regression algorithm, was used for binary classification by thresholding the predicted probabilities. However, it yielded suboptimal results for this classification task, with an accuracy of 95%.

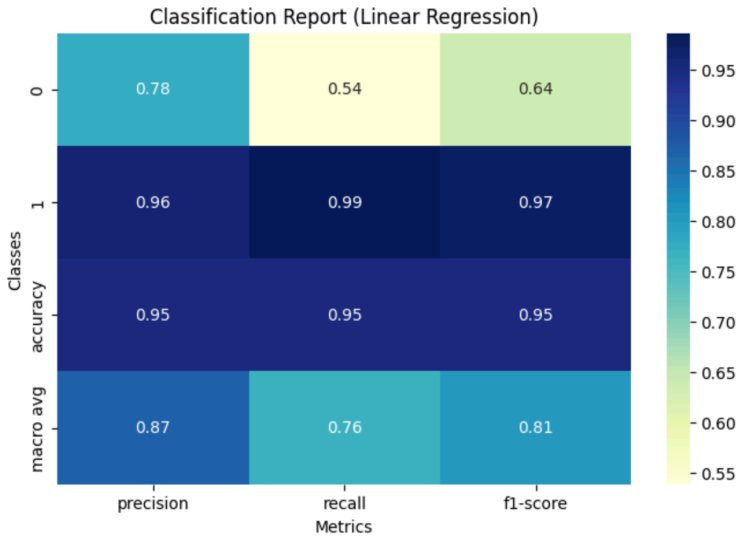


**Fig 1.1 :** Classification Report for Linear Regression

2. **Logistic Regression**: Logistic regression, a dedicated classification algorithm, performed better than linear regression with an accuracy of 93%. The classification report and confusion matrix are provided in the screenshots.

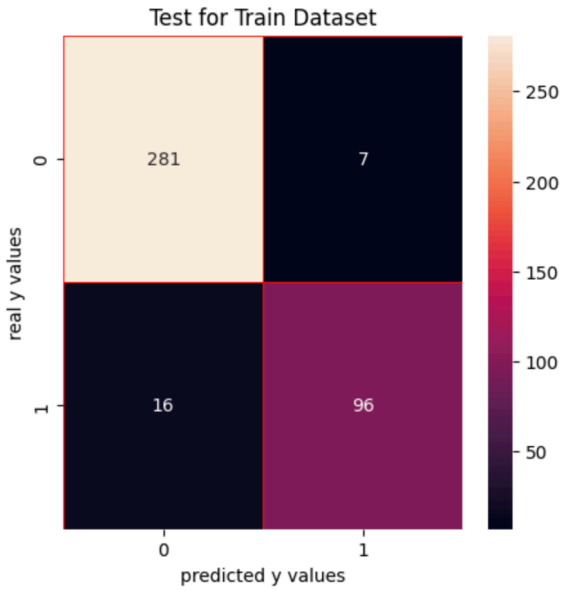


**Fig 2.1 :** Confusion matrix for Logistic Regression

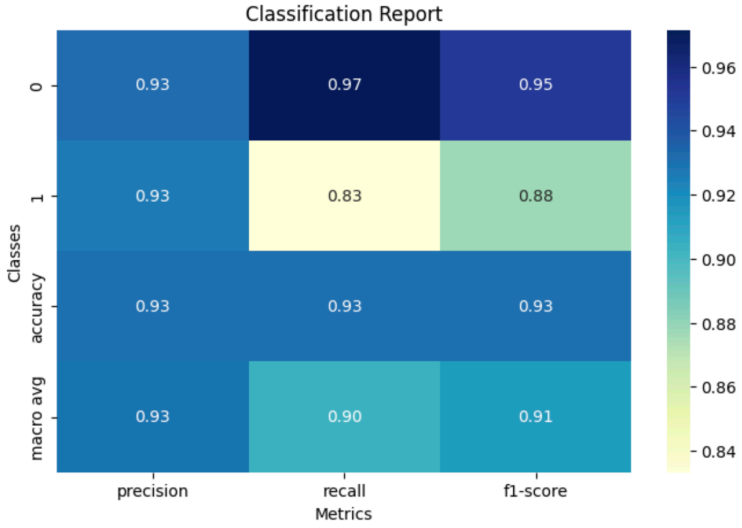


**Fig 2.2 :** Classification Report for Logistic Regression

3. **Support Vector Machine (SVM)**: SVM, known for its effectiveness in binary classification tasks, achieved an accuracy of 93%, outperforming both linear regression and logistic regression. The SVM model's classification report and confusion matrix are included in the screenshots.

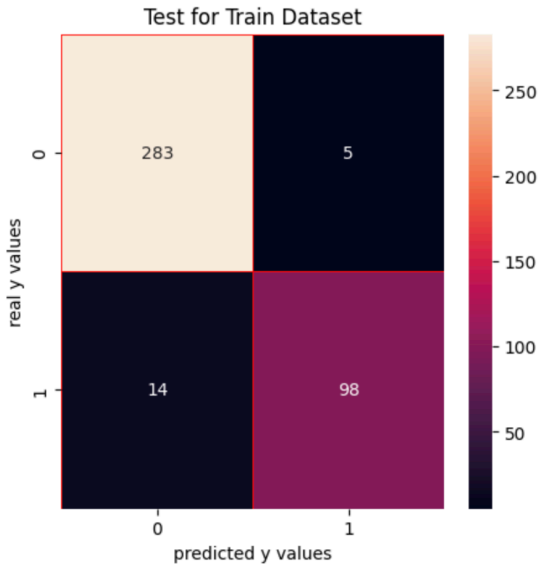


**Fig 3.1 :** Confusion Matrix for Support Vector Machine (SVM)

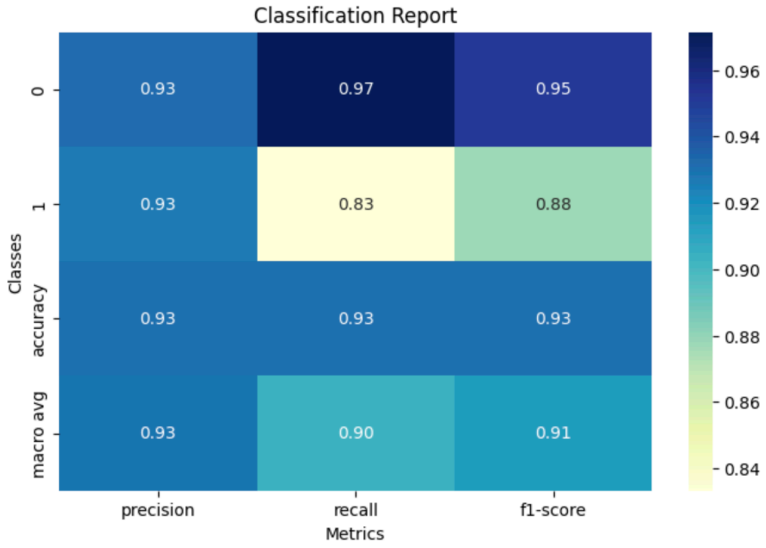


**Fig 3.2 :** Classification Report for Support Vector Machine (SVM)

4. **Gaussian Naive Bayes**: Gaussian Naive Bayes, a probabilistic classifier, demonstrated decent performance with an accuracy of 89%. Its classification report and confusion matrix are shown in the screenshots.

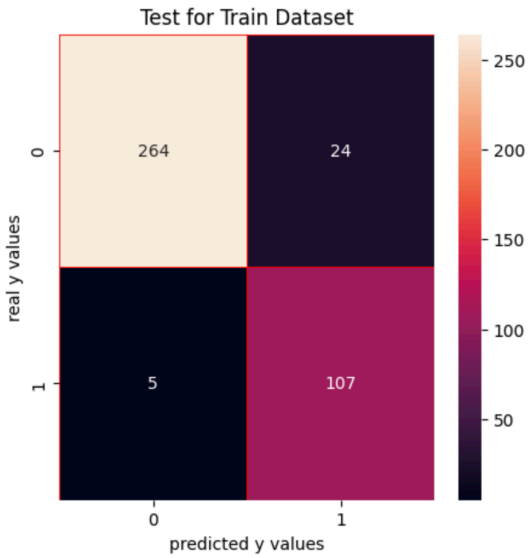


**Fig 4.1 :** Confusion Matrix for Gaussian Naive Bayes

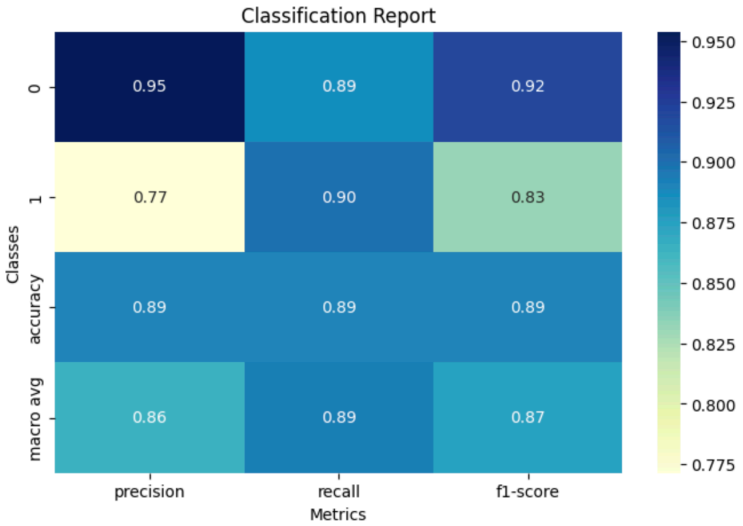


**Fig 4.2 :** Classification Report for Gaussian Naive Bayes

5. **Decision Tree Classification**: Decision tree classification, a non-linear model, achieved an accuracy of 79%, showcasing its ability to capture complex decision boundaries. The classification report and confusion matrix are provided for reference.

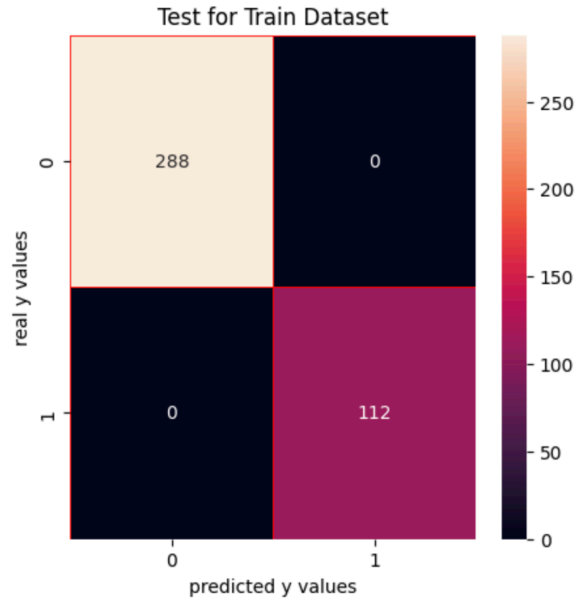


**Fig 5.1 :** Confusion Matrix for Decision Tree Classification

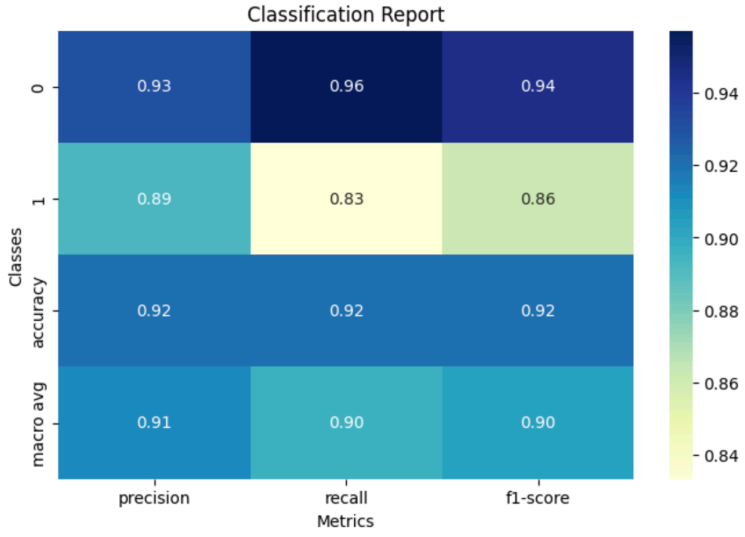


**Fig 5.2 :** Classification Report for Decision Tree Classification

6. **Random Forest Classification**: Random forest classification, an ensemble method, delivered the highest accuracy among the tested algorithms, reaching 90%. Its classification report and confusion matrix highlight its robustness in handling complex classification tasks.

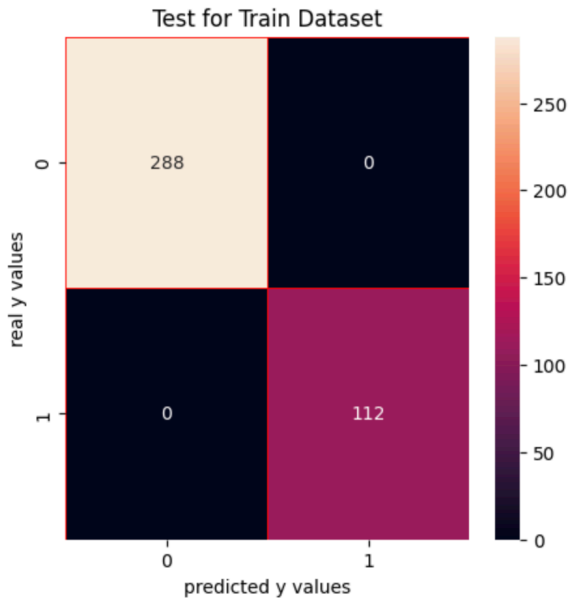


**Fig 6.1 :** Confusion Matrix for Random Forest Classification

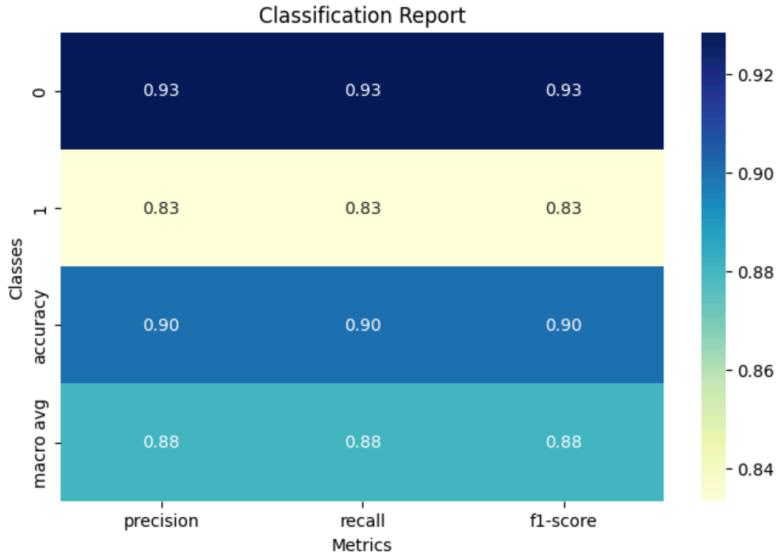


**Fig 6.2 :** Classification Report for Random Forest Classification

# CONCLUSION

Through this university admission prediction study, we studied how well several machine learning algorithms performed in predicting admission outcomes based on characteristics of applicants, including GPA, TOEFL, SOP, LOR, and CGPA. Among the techniques put to the test were Random Forest Classification, Decision Tree Classification, Gaussian Naive Bayes, Support Vector Machine (SVM), Logistic Regression, and Linear Regression.

**Findings and Limitations:**

1. Algorithm Performance: Among the algorithms tested, Support Vector Machine and Logistic Regression gave the highest accuracy of 93%, followed by Decision Tree Classification with 92% accuracy and Random Forest Classification with 90% accuracy. These results highlight the importance of choosing suitable algorithms for classification tasks based on their performance on the given dataset.

2. Limitations:

- Data Quality: The quality of the dataset significantly impacts model performance. Limited or biased data may lead to optimal predictions.
- Feature Selection: The selection of features plays a crucial role in model accuracy. Predictive abilities may be improved by more feature importance analysis and improvement.
- Understanding of the Model: Although ensemble methods such as Random Forests yielded good results, their understanding may be lower than that of simpler models like Logistic Regression. This can lead to difficulties in understanding the model's decision-making process.

**Future Scope:**

1. Feature Engineering: Exploring additional features or engineering existing ones could improve predictive power. Factors such as extracurricular activities, research experience, and personal statements could be considered.

2. Colleges Recommendations : Currently we are only predicting the chances of admission but we can also include college recommendation by suggesting colleges on the basis of ranks.

3. Data Correction: Generating synthetic data points or balancing class distributions can address data imbalances and improve model robustness.

In summary, even though our project showed promise in predicting university admission outcomes, there are still opportunities for development and growth. More accurate and dependable admission prediction models can result from addressing constraints and investigating potential future scope areas, which will help academic institutions and applicants alike in the decision-making process.