ISSN: 2321-2152

IJMECE

International Journal of modern
electronics and communication engineering

E-Mail
editor.ijmece@gmail.com
editor@ijmece.com

www.ijmece.com

ISSN2321-2152
www.ijmece .com
Vol 12, Issue 2, 2024

College Admission Prediction using Machine Learning

K.U. NIKITHA, MVSS. KRISHNA SANJAY, G. VAMSHI KRISHNA, M. KRISHNA SAI,
Ms. K. ANKITHA, Dr. K. VASANTH KUMAR, Mr. LALAM RAMU

Department of Computer Science and Engineering – Internet of Things
Malla Reddy Engineering College, Hyderabad, Telangana
0604.nikitha@gmail.com

***Abstract*—** The goal of this project is to create a machine learning model that will estimate a student's likelihood of admission to engineering and technology universities based on their 10th grade marks, 12th grade marks, 12th division, and AIEEE rank. Regression techniques like **Linear Regression, KNN Regressor, Decision Tree Regressor, or Random Forest Regressor** are used in the model to forecast the likelihood of admission. Data collection, data preprocessing, feature selection, model selection, model training, model evaluation, and model deployment are all aspects of the project. Based on their academic achievement and AIEEE rank, students can anticipate their chances of admission in Engineering & Technology Institutions by deploying the model in a web or mobile application.

Key Words: Linear Regression, KNN Regressor, Decision Tree Regressor, Random Forest Regressor, AIEEE rank

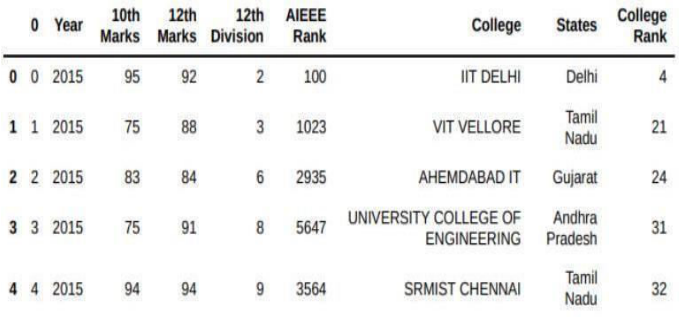
I. INTRODUCTION

One of the most sought-after educational institutions in the world, engineering and technology universities draw applicants from a variety of academic and cultural backgrounds. The selection process for admission to these colleges is rigorous, and candidates are chosen mostly based on their academic standing and AIEEE (All India Engineering Entrance Test) rank. It can be helpful for students to better plan their academic career to be able to predict their chances of admission depending on their academic performance and AIEEE rank. By using the model in a web or mobile application, students may quickly anticipate their chances of admission into engineering and technology universities based on their academic achievement and AIEEE rank. Students, parents, and educational institutions may find this project to be a helpful tool in making decisions and successfully planning students' academic careers.

1.DATA COLLECTION AND PREPROCESSING

In general, data for a machine learning project can be gathered from a variety of sources, including surveys, online scraping, public datasets, and collaborations with businesses. In the instance of Admission Prediction in Engineering & Technology Schools, we discovered the information we could utilize from Kaggle. The information may have been gathered through previous admission records of the institutions, student transcripts, or through surveys given to applicants. In order to prepare the data for analysis and modelling, it was pre-processed and cleaned to remove any discrepancies or inaccuracies. By eliminating missing values, scaling the numerical features, and transforming category characteristics into numerical ones, we cleaned and pre-processed the data.

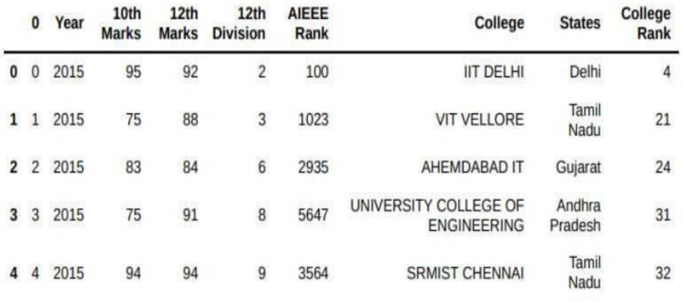
This is the dataset we found:



	0	Year	10th Marks	12th Marks	12th Division	AIEEE Rank	College	States	College Rank
0	0	2015	95	92	2	100	IIT DELHI	Delhi	4
1	1	2015	75	88	3	1023	VIT VELLORE	Tamil Nadu	21
2	2	2015	83	84	6	2935	AHEMDABAD IT	Gujarat	24
3	3	2015	75	91	8	5647	UNIVERSITY COLLEGE OF ENGINEERING	Andhra Pradesh	31
4	4	2015	94	94	9	3564	SRMIST CHENNAI	Tamil Nadu	32

Fig -1 Dataset

After data cleaning and pre-processing we were ready to use different algorithms:



	0	Year	10th Marks	12th Marks	12th Division	AIEEE Rank	College	States	College Rank
0	0	2015	95	92	2	100	IIT DELHI	Delhi	4
1	1	2015	75	88	3	1023	VIT VELLORE	Tamil Nadu	21
2	2	2015	83	84	6	2935	AHEMDABAD IT	Gujarat	24
3	3	2015	75	91	8	5647	UNIVERSITY COLLEGE OF ENGINEERING	Andhra Pradesh	31
4	4	2015	94	94	9	3564	SRMIST CHENNAI	Tamil Nadu	32

Fig -2 Cleaned Dataset

We assigned ‘College Rank’ for each College by referring to the National Ranking of these Engineering and Technology Colleges in India.

Here are some details about the range of data in our ‘CAP.csv’:

Total tuples: 1004

Total Attributes: 8

Total States: 20

Total Colleges: 38

AIEEE Rank: 45- 9878

2. FEATURE SELECTION

Selecting the relevant features using feature importance, we concluded that '10th Marks', '12th Marks', '12th Division', and 'AIEEE Rank' will be used to train our model. Using these we will be predicting 'College Rank'.

3. ALGORITHM SELECTION AND MODEL TRAINING

Here we split the data into training and testing sets and train the model on chosen algorithms on the training data. We used four test sizes: 0.1, 0.2, 0.3 and 0.4.

The selection of a regression algorithm depends on the nature of the data and the problem statement. In this project, we have multiple independent variables (10th Marks, 12th Marks, 12th Division, and AIEEE rank) and a single dependent variable (College Rank).

To choose the best regression algorithm, we can follow these steps:

1. Linear Regression: We can start by using a simple linear regression model to see how well the data fits the model. Linear regression assumes a linear relationship between the independent and dependent variables. If the data has a linear relationship, linear regression can be a good choice.

2. KNN Regressor: KNN Regressor is a machine learning algorithm used for regression tasks. It predicts the target value of a new data point based on the average of the k-nearest neighbors target values in the training set, where k is a user-defined hyperparameter.

3. Decision Tree Regressor: If the data has complex and non-linear relationships, decision tree regression can be a good choice. Decision tree regression can capture complex relationships between the independent and dependent variables.

4. Random Forest Regressor: Random forest regression is a popular regression algorithm that uses multiple decision

5. Trees to make predictions. It can handle non-linear relationships between the independent and dependent variables and can also prevent overfitting.

3. OVERVIEW OF ALGORITHMS USED

3.1 LINEAR REGRESSION

A form of regression technique called linear regression uses one or more input variables, usually referred to as independent variables or features, to predict a continuous target variable. A straight line can be used to show the connection between the input variables and the target variable in linear regression models.

The target variable's predicted and actual values are compared using the linear regression procedure to identify the line of best fit that minimizes the sum of the squared residuals. An equation in the form of: represents the line of best fit.

$$y = \beta 0 + \beta 1x1 + \beta 2x2 + \ldots + \beta nxn$$

where y is the predicted value of the target variable, β0 is the intercept or bias term, β1, β2, ..., βn are the coefficients of the input variables x1, x2, ..., xn, respectively.

The linear regression technique employs the Ordinary Least Squares (OLS) method, which minimizes the sum of the squared residuals, to get the ideal coefficient values. The disparities between the target variable's expected and actual values are known as the residuals. By entering the values of the input variables into the equation after the coefficients have been computed, we may utilize the linear regression model to predict new data.

A popular and straightforward regression approach that is simple to understand and use is linear regression. The assumption of linearity, vulnerability to outliers, and inability to manage non-linear connections between the input and target variables are a few of its drawbacks.

3.2 KNN REGRESSOR

The K-Nearest Neighbors (KNN) Regressor is a kind of regression method that makes predictions based on the separations between the input data points. As the KNN regressor is a non-parametric technique, it may model complicated connections without making any assumptions about the distribution of the data.

The KNN algorithm works as follows:

Distance calculation: The method determines the distances between each new data point and the input data points in the training set for each new data point. Any appropriate metric can be used as the distance measurement, including Manhattan and Euclidean.

1. Based on the estimated distances, the algorithm chooses the k nearest data points from the training set. K is a hyperparameter whose value may be selected based on the dataset.

2. In order to forecast the value of the new data point, the algorithm first calculates the average or weighted average of the target variable values of the k nearest neighbors.

A straightforward and efficient regression approach that can handle non-linear connections and adjust to changes in the data is the KNN regressor. With big datasets, it can be computationally costly and sensitive to the distance measure chosen. Moreover, the KNN regressor makes the assumption that the data distribution is uniform, which could not be the case for all datasets.

3. In order to forecast the value of the new data point, the algorithm first calculates the average or weighted average of the target variable values of the k nearest neighbors.

A straightforward and efficient regression approach that can handle non-linear connections and adjust to changes in the data is the KNN regressor. With big datasets, it can be computationally costly and sensitive to the distance measure chosen. Moreover, the KNN regressor makes the assumption that the data distribution is uniform, which could not be the case for all datasets.

3.3 DECISION TREE REGRESSOR

A decision tree is used in the Decision Tree Regressor, a form of regression technique, to forecast the target variable based on a number of input factors. Each node on the decision tree represents a feature or characteristic, and each branch on the decision tree represents a decision rule or condition.

The decision tree is constructed by recursively partitioning the dataset depending on the feature that leads to the largest information gain or decrease in the impurity of the target variable. The target variable's impurity is identified using the variance, mean squared error, or any other suitable measure.

By navigating the decision tree depending on the values of the input characteristics, we may use it to predict the target variable of a new data point after it has been generated. The decision tree evaluates the value of each node's input characteristic and determines which branch to take in accordance with the decision rule. The target variable's projected value is represented by the leaf node of the tree, where the prediction is made.

Over other regression methods, the decision tree regressor offers a number of benefits, including the capacity to handle non-linear connections and complicated decision boundaries, the capacity to manage missing values, and the simplicity of interpretation. Nevertheless, if the tree is too deep or the dataset is too little, the decision tree regressor might potentially experience overfitting. We can employ strategies like pruning, regularization, and ensemble learning to avoid overfitting.

3.4 RANDOM FOREST REGRESSOR

A regression technique known as the Random Forest Regressor makes predictions by using several decision trees. Because each decision tree in the random forest is trained using a random portion of the training data as well as a random subset of the input characteristics, there is less chance of overfitting and the predictions are more accurate.

The random forest algorithm works as follows:

1. Random subset selection: The algorithm chooses a random subset of the training data and the input characteristics..

2. Construction of the decision tree: A decision tree is built using a random subset of characteristics and data. Similar to the decision tree regressor, the decision tree is built by iteratively dividing the data depending on the feature that offers the greatest information gain or impurity reduction.

3. Ensemble learning: To generate many decision trees, the first two steps are repeated numerous times. The decision trees' forecasts are then combined using the random forest method to get a final prediction.

4. Prediction: The random forest method aggregates the predictions from all of the decision trees in the forest to get a final forecast for each new data point.

5. Strong regression algorithms that prevent overfitting, like the Random Forest Regressor, can handle non-linear correlations between the input and target variables. It is frequently used in machine learning applications such as financial forecasting, image analysis, and healthcare prediction. It could struggle to perform well on datasets with few features or training samples.

4. MODEL EVALUATION

After trying different regression algorithms, we compared the performance of each algorithm using metrics such as mean squared error, root mean squared error, R-squared score, etc., and choose the best-performing algorithm for the given problem statement. Here we used 'metrics' library from sklearn and used score() to test accuracy for every model.

Here's a summary table of model performance for these four algorithms and four different test sizes.

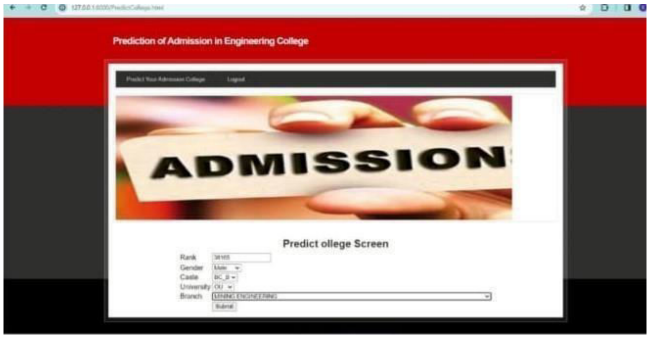
Table -1

Algorithm	Test Size= 0.1	Test Size= 0.2	Test Size= 0.3	Test Size = 0.4
Linear Regression	12.40	31.10	27.76	29.85
KNN Regressor (k=2)	90.85	90.83	91.26	77.42
Decision Tree Regressor	90.37	95.86	97.53	84.08
Random Forest Regressor	77.16	85.46	83.60	84.08

As we can see that the Decision Tree Regressor Algorithm with test size=0.3 (which is also standard train-test split size) has an accuracy score of 97.53, we selected this model with random state=3 as our final model.

5. MODEL DEPLOYMENT

Using Tkinter we deployed a 'College Predictor' application. It lets the user input: 10th Marks, 12th Marks, 12th Division, and AIEEE Rank along with 'Select State' option to filter colleges by demographic location. After entering the details the user has to click the 'Submit' button to display the list of colleges he is eligible to apply at.

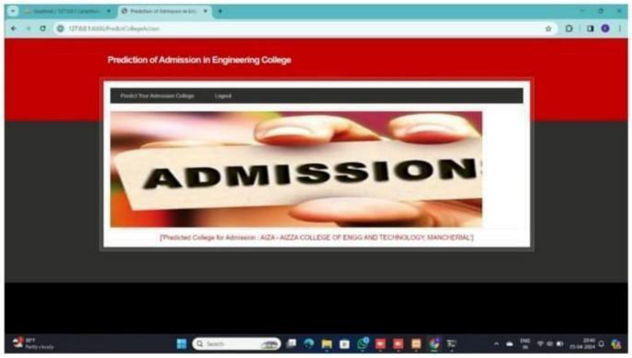


Prediction of Admission in Engineering College

Predict ollege Screen

Rank 38165
Gender Male
Caste BC_B
University OU
Branch MINING ENGINEERING
Submit

Fig-1



Prediction of Admission in Engineering College

['Predicted College for Admission : AIZA - AIZZA COLLEGE OF ENGG AND TECHNOLOGY, MANCHERIAL']

Fig-2

6. CONCLUSION

In conclusion, the goal of this study was to use machine learning regression methods to predict admission to engineering and technology colleges. The dataset was subjected to the Decision Tree Regressor, Linear Regressor, Random Forest Regressor, and KNN Regressor regression methods. The Decision Tree Regressor fared the best, according to the results, with an accuracy of 97.53%. The experiment showed how machine learning may be used to forecast college admittance based on student performance and rankings. These types of models can assist universities in streamlining their admissions procedure and assisting students in making wise choices. It is crucial to remember that the quality and amount of the dataset, as well as the selection of suitable algorithms and hyperparameters, have a significant impact on accuracy.

REFERENCES

[1] Pachauri, S., & Singh, S. P. (2019). Predictive modeling for student admission using machine learning algorithms. Journal of Education and Practice, 10(1), 78-89.

[2] Bhowmik, S. K., & Tiwari, R. K. (2020). Application of machine learning techniques for student admission prediction. Journal of Education and Practice, 11(4), 40- 49.

[3] Singh, D. (2018). Predicting student admission using machine learning techniques. International Journal of Advanced Research in Computer Science, 9(2), 62-67.

[4] Khare, R., & Sharma, V. (2021). Admission prediction in higher education using machine learning. In Proceedings of the International Conference on Advanced Computing and Intelligent Engineering (pp. 53-60). Springer.

[5] Gupta, A., & Gupta, R. (2018). Admission prediction in higher education institutes using machine learning algorithms. International Journal of Computer Science and Mobile Computing, 7(1), 143-151.

[6] "Introduction to Machine Learning with Python: A Guide for Data Scientists" by Andreas C. Müller and Sarah Guido (Covers Decision Trees, Random Forests, and KNN)

[7] "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems" by Aurélien Géron (Covers Linear Regression, Decision Trees, Random Forests, and KNN)

[8] "Machine Learning: A Probabilistic Perspective" by Kevin Murphy (Covers Linear Regression, Decision Trees, and KNN)

[9] "The Elements of Statistical Learning: Data Mining, Inference, and Prediction" by Trevor Hastie, Robert Tibshirani, and Jerome Friedman (Covers Linear Regression, Decision Trees, and Random Forests)

[10] "Pattern Recognition and Machine Learning" by Christopher Bishop (Covers Linear Regression, Decision Trees, and KNN)