

**INT 247**

**MAJOR PROJECT**

**REPORT ON:**

**“Graduate Admissions”**

**SUBMITTED BY: Tarun Singh Mahar**

**SECTION: K1623**

**REG. NO.: 11608343**

**ROLL NO.: A11**

**Graduate Admission**

The dataset contains several parameters which are considered important during the application for master’s Programs. The parameters/Attributes included are:

1. GRE Scores (out of 340)

2. TOEFL Scores (out of 120)

3. University Rating (out of 5)

4. Statement of Purpose and Letter of Recommendation Strength (out of 5)

5. Undergraduate GPA (out of 10)

6. Research Experience (either 0 or 1)

7. Chance of Admit (ranging from 0 to 1)

This dataset was built with the purpose of helping students in shortlisting universities with their profiles. The predicted output gives them a fair idea about their chances for a university.

**DATA SET**

The dataset used in this project includes quantitative and categorical features This dataset is created for prediction of Graduate Admissions from an Indian perspective.

The dataset contains two csv file

One file contains 400 records and 9 tuned features.

Second file contain 500 records and 9 features.

The output column is **Chance of Admission.**

|  |  |
| --- | --- |
| Number of instances | 400,500 |
| attribute | 9 |

1. **MODELS USED:**
2. **Logistic regression:**

[**Logistic regression**](http://www.statisticssolutions.com/academic-solutions/membership-resources/member-profile/data-analysis-plan-templates/data-analysis-plan-logistic-regression/) is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary).  Like all regression analyses, the logistic regression is a predictive analysis.  Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

1. **Random forest.**

Random forests is a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest is comprised of trees. It is said that the more trees it has, the more robust a forest is. Random forests create decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a pretty good indicator of the feature importance.

1. **Decision tree.**

Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

**Steps taken in project:**

1. Importing both data and combining them.

df1 = pd.read\_csv('Admission\_Predict\_Ver1.1.csv')

df2 = pd.read\_csv("Admission\_Predict.csv")

df = pd.concat([df1,df2],axis=0,sort=False)

df.shape

1. Cleaning dataset and removing NaN values if present.

df.rename(columns = {'Chance of Admit ':'Chance of Admit', 'LOR ':'LOR'}, inplace=True)

df.drop(labels='Serial No.', axis=1, inplace=True)

1. Feature selection using correlation matrix using heatmap.

fig, ax = plt.subplots(figsize=(10,10))

sns.heatmap(df.corr(), annot=True, cmap='Blues')

1. Preparing dataset

targets = df['Chance of Admit']

features = df.drop(columns = {'Chance of Admit'})

X\_train, X\_test, y\_train, y\_test = train\_test\_split(features, targets, test\_size=0.3, random\_state=42)

1. Implementing the models.

* Logistic regression model:

logreg = LogisticRegression()

logreg.fit(X\_train\_std, y\_train\_int)

y\_predict = logreg.predict(X\_test\_std)

logreg\_score = (logreg.score(X\_test\_std, y\_test\_int))\*100

logreg\_score

* decision tree:

dec\_tree = DecisionTreeClassifier(random\_state=0, max\_depth=6)

dec\_tree.fit(X\_train\_std, y\_train\_int)

y\_predict = dec\_tree.predict(X\_test\_std)

dec\_tree\_score = (dec\_tree.score(X\_test\_std, y\_test\_int))\*100

dec\_tree\_score

* random forest:

forest = RandomForestClassifier(n\_estimators=110,max\_depth=6,random\_state=0)

forest.fit(X\_train\_std, y\_train\_int)

y\_predict = forest.predict(X\_test\_std)

forest\_score = (forest.score(X\_test\_std, y\_test\_int))\*100

forest\_score

1. After calculating all the scores using all models we will identify which model is better for taken dataset.

maxi = max(logreg\_score, dec\_tree\_score, forest\_score)

if(maxi==logreg\_score):

print("Best model is Logistic regression for my given dataset")

print(logreg\_score)

elif(maxi==dec\_tree\_score):

print("Best model is Decission tree for my given dataset")

print(dec\_tree\_score)

else:

print("Best model is Random forest for my given dataset")

print(forest\_score)

References:

1. [https://towardsdatascience.com/feature-selection-techniques-in- machine-learning-with-python-f24e7da3f36e](https://towardsdatascience.com/feature-selection-techniques-in-%20%20%20%20machine-learning-with-python-f24e7da3f36e)

2. <https://towardsdatascience.com/hyperparameter-tuning-c5619e7e6624>

3. <https://www.geeksforgeeks.org/decision-tree/>

4. <https://www.statisticssolutions.com/what-is-logistic-regression/>

5. <https://seaborn.pydata.org/generated/seaborn.pairplot.html>