University Recommendation System

Background

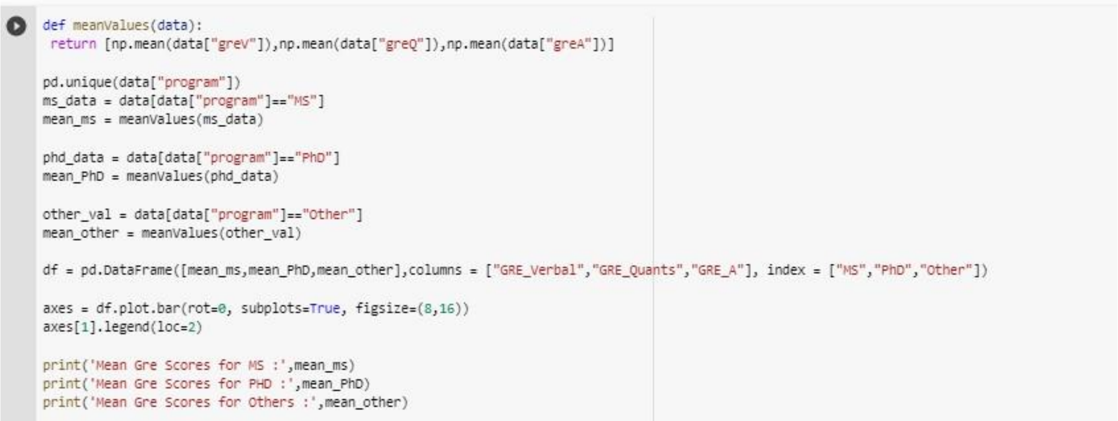
- Institutions take in the students who are suitable candidates based on their academic profile, standardized test scores. But in this entire process university selection is the most crucial & tedious step for applying to graduate studies.
- Some of them succeed and get admission into their desired programs in desired universities, but some are not because of the academic level of colleges which they have applied.
- To resolve this problem of not getting admission because of applications, even though students have good academic profile, I have developed this recommendation system.
- In this project, the knowledge acquired from the database of successful applicants is used to predict the schools with various data mining techniques. This data will be modeled into machine learning algorithms to predict the universities and their acceptance rate for the given user academic details

Problem Statement

- For an aspiring student who wants to apply for higher studies in other countries, university selection process is a challenging task.
- Lot of different criteria need to considered during application process based on individual's requirement.
- This problem can be addressed by modeling a recommender system based on various classification algorithms.

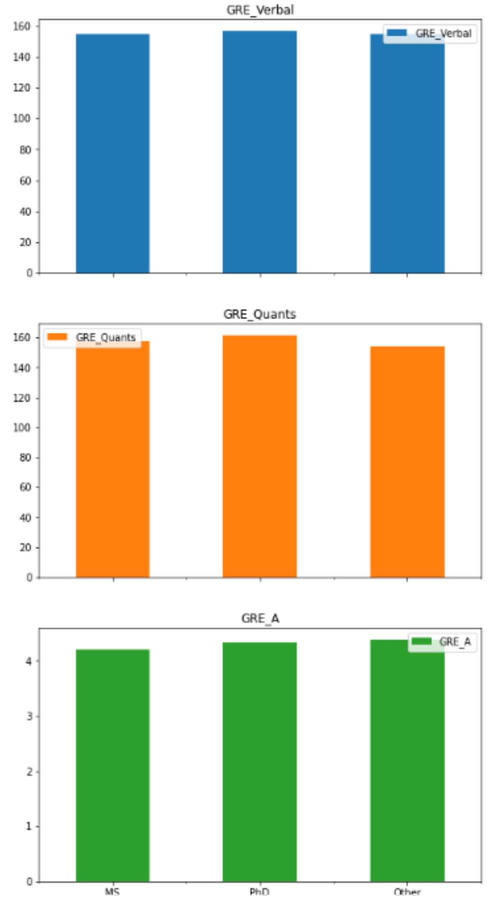
Exploratory Data Analysis

- **Mean GRE scores across different Degrees**

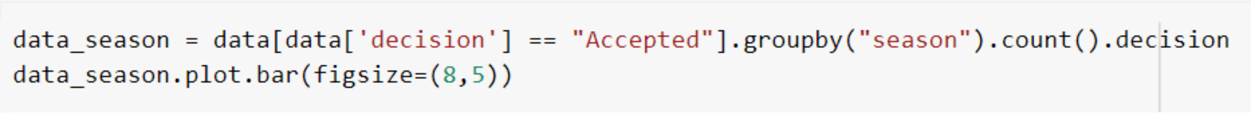


```
def meanValues(data):
 return [np.mean(data["greV"]),np.mean(data["greQ"]),np.mean(data["greA"])]

pd.unique(data["program"])
ms_data = data[data["program"]=="MS"]
mean_ms = meanValues(ms_data)

phd_data = data[data["program"]=="PhD"]
mean_PhD = meanValues(phd_data)

other_val = data[data["program"]=="Other"]
mean_other = meanValues(other_val)

df = pd.DataFrame([mean_ms,mean_PhD,mean_other],columns = ["GRE_Verbal","GRE_Quants","GRE_A"], index = ["MS","PhD","Other"])

axes = df.plot.bar(rot=0, subplots=True, figsize=(8,16))
axes[1].legend(loc=2)

print('Mean Gre Scores for MS :',mean_ms)
print('Mean Gre Scores for PHD :',mean_PhD)
print('Mean Gre Scores for Others :',mean_other)
```

```
Mean Gre Scores for MS : [154.62137768039108, 157.54543898593164, 4.213057092725064]
Mean Gre Scores for PHD : [156.63377895267496, 161.31669547568123, 4.328063764431734]
Mean Gre Scores for Others : [154.85909562673962, 153.97548002075757, 4.382815964523282]
```

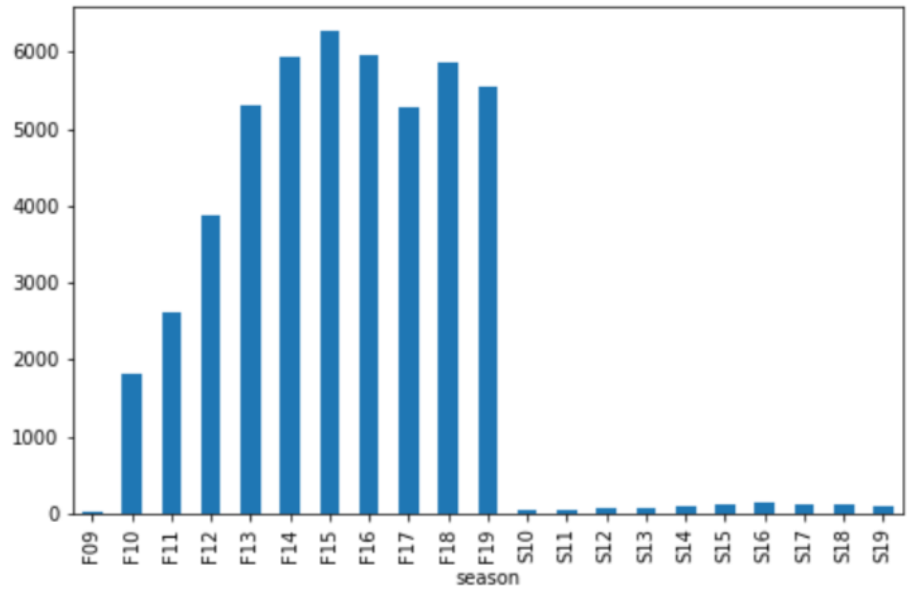


- **Incoming Students each Season**

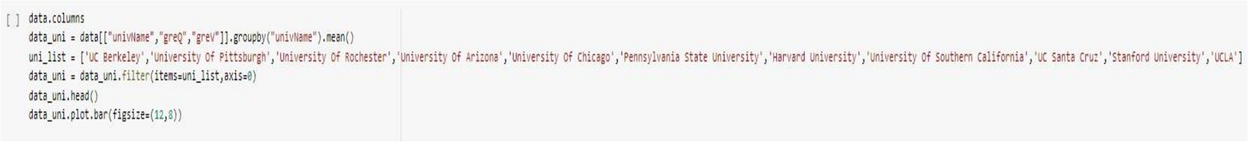


```
data_season = data[data['decision'] == "Accepted"].groupby("season").count().decision
data_season.plot.bar(figsize=(8,5))
```

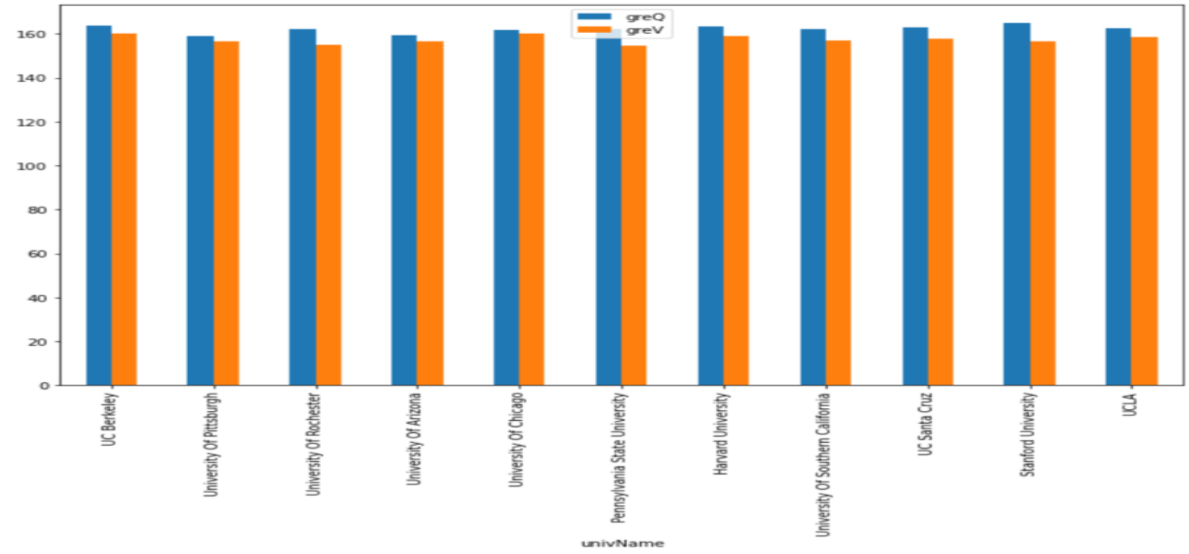
```
<AxesSubplot:xlabel='season'>
```



- Average Scores(GRE, CGPA) of Following top Universities



```
[ ] data.columns
    data_uni = data[["univName","greQ","greV"]].groupby("univName").mean()
    uni_list = ['UC Berkeley','University Of Pittsburgh','University Of Rochester','University Of Arizona','University Of Chicago','Pennsylvania State University','Harvard University','University Of Southern California','UC Santa Cruz','Stanford University','UCLA']
    data_uni = data_uni.filter(items=uni_list,axis=0)
    data_uni.head()
    data_uni.plot.bar(figsize=(12,8))
```



Algorithms Used

- KNN
- Random Forest
- Naïve B
- Logistic Regression
- Convolution Neural Network

KNN Algorithm

In KNN, the trained data is compared with test data and distances are calculated using Euclidean distance. It then classifies an instance by finding its nearest neighbors and recommend the top n nearest neighbor universities. Algorithm is stated as below.

- Initialize the value of k
- For getting recommendation, iterate from 1 to number of trained data.
- Calculate distance between test data and each row
- Sort the distances in ascending order
- Get top k rows and recommend to the user

Random Forest Algorithm

- Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.
- One of the most important features of this algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification
- In the recommendation system, it helps the student to predict which university best suits their Academic portfolio, with an accuracy of 92%

Naive Bayes Algorithm

- Gaussian Naive Bayes is a probabilistic classification algorithm based on applying Bayes' theorem with strong independence assumptions.
- In the context of classification, independence refers to the idea that the presence of one value of a feature does not influence the presence of another (unlike independence in probability theory)
- In our recommendation system, the Naïve Bayes Algorithm gives us the accuracy of 62.8%

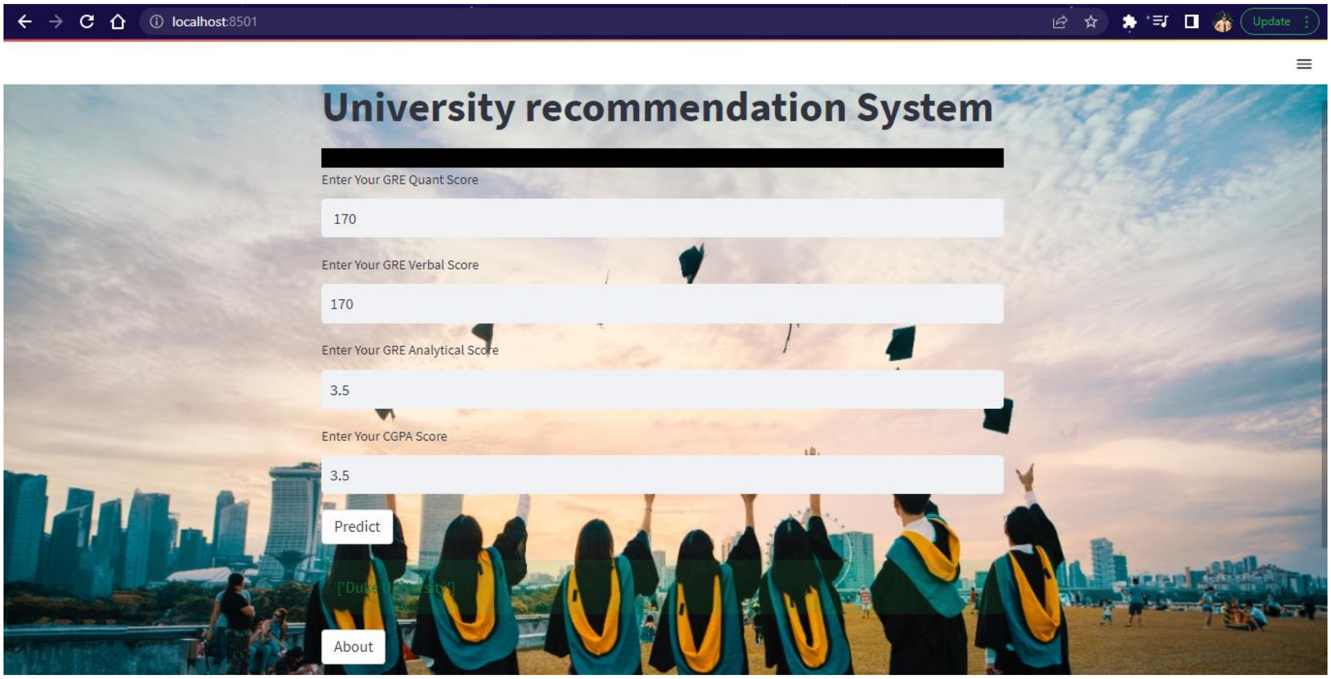
Logistic Regression Algorithm

- Logistic regression is a statistical method that is used for building machine learning models where the dependent variable is dichotomous: i.e., binary.
- Logistic regression is used to describe data and the relationship between one dependent variable and one or more independent variables. The independent variables can be nominal, ordinal, or of interval type.
- For the University Recommendation System, the Logistic Regression Algorithm gives an accuracy of 64%

Convolution Neural Network

- Convolutional neural networks are widely used in computer vision and have become the state of the art for many visual applications such as image classification, and have also found success in natural language processing for text classification.
- A convolutional neural network is a feed-forward neural network, often with up to 20 or 30 layers. The power of a convolutional neural network comes from a special kind of layer called the convolutional layer.
- The Convolution Neural Network algorithm provides an accuracy of 66.16% for the University Recommendation System

Created a web page using Streamlit



University recommendation System

[redacted]

Enter Your GRE Quant Score

170

Enter Your GRE Verbal Score

170

Enter Your GRE Analytical Score

3.5

Enter Your CGPA Score

3.5

```
['Duke University']
```

CONCLUSION :

- This project will help students in decision making of which University to choose for their higher education in other countries like USA.
  They can enter their Scores and can check the Universities to which the can apply

- Out of all the algorithms used we come to a conclusion that Random Forest Algorithm gives us the maximum accuracy of 92%

- Students can either check which universities to apply to (Through WEBPAGE - KNN trained Algorithm)
  or
  Check if they can get into their dream university (Through Random Forest Classification)