Individual project on

USA University Recommendation System based on Student profile for UG/graduate studies

Rama Tejaswini Thotapalli (SJSU Id: 013785681)

Under the guidance of Professor Shih Yu Chang, San Jose State University

Table of Contents

1. ABSTRACT:

In this paper, I have presented a recommender system for undergraduate & graduate admission seekers, which can help students to choose best graduate university matching their academic profile. Here I have used a different data mining techniques to transform database of students of relevant information into a universal database format using academic data of successful students who have already got opportunity to study abroad. After that I have developed a machine learning algorithm which can calculate similarity between training and test data based on weighted scores. I have used K-nearest Neighbor algorithm and feature-weighted algorithm for calculating top N similar users for the test users and recommend Top K universities to users from N similar users.

2. INTRODUCTION:

Many students who wants to pursue higher studies apply different universities with their academic profile as well as standardized test scores such as SAT, GRE, TOEFL, and IELTS. 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.

3. BACKGROUND & OBJECTIVE:

For an aspiring student who wants to apply for higher studies in other countries, university selection process is a challenging task as lot of different criteria need to consider during application process based on individual's requirement. 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.

This problem can be addressed by modeling a recommender system based on various classification algorithms. In this project based on the student data set and the student profile who is looking for the admit, various models will be trained and a list of 10 best universities will be suggested such that it maximizes the chances of a student getting admit from that university list..

4. DATASET:

The first step in building any recommendation system is the identification of the data set. In order to build the classification model for the recommender system, this data has to be organized with appropriate labels. This core data for the application process is not readily available on the internet for direct consumption. However, this whole approach is based on making maximum use of the available information. The graduate student data was scraped from the following websites

www.thegradcafe.com, and the Undergraduate university student data was scraped from https://collegescorecard.ed.gov/data/.

4.1. Graduate Student Dataset:

For Graduate Student data, we scraped www.thegradcafe.com website. About 271807 rows of raw student data was obtained as a result of web scraping. Each sample corresponds to the profile of a student. We have got 1949 html pages of the data and need to change it into CSV files.

```
: import matplotlib.pyplot as plt
  import requests
  import urllib.request
  from IPython.core.debugger import Tracer

  url_form = "http://thegradcafe.com/survey/index.php?q=u%2A&t=a&pp=250&o=d&p={0}"
  DATA_DIR = './WebScraped_data/html/'

  if __name__ == '__main__':
      for i in range(1691, 1948):
          url = url_form.format(i)
          handle = urllib.request.urlopen(url)
          html = handle.read()
          html = html.decode('utf8')
          #r = requests.get(url)
          fname = "{data_dir}/{page}.html".format(data_dir=DATA_DIR,page=str(i))
          with open(fname, 'wb') as f:
              f.write(html.encode('UTF-8'))
          print("getting {0}...".format(i))
  getting 1775...
  getting 1776...
  getting 1777...
  getting 1778...
  getting 1779...
  getting 1780...
  getting 1781...
```

After scraping the final data will look like

```
In [39]: df.head()
```

Out[39]:

	Unnamed: 0	Unnamed: 0.1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	0	University Of Waterloo	Systems Design Engineering	MS	NaN	Accepted	Website	(1, 7, 2019)	1.561964e+09	NaN	NaN	NaN	NaN	NaN	NaN	International
1	1	1	Northeastern University	Electrical Engineering	PhD	F19	Rejected	Website	(8, 7, 2019)	1.562569e+09	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	2	2	The University Of Auckland	Electrical And Electronic Engineering	MS	NaN	Accepted	Website	(19, 6, 2019)	1.560928e+09	NaN	NaN	NaN	NaN	NaN	NaN	International
3	3	3	Radford University	Counseling Psychology PsyD.	Other	F19	Accepted	Phone	(4, 3, 2019)	1.551686e+09	NaN	NaN	NaN	NaN	NaN	NaN	American
4	4	4	University Of Chittagong	Computer Science	MS	NaN	NaN	Other	(9, 7, 2019)	1.562656e+09	3.2	163.0	168.0	4.0	True	NaN	International

The list of attributes are made as dataset for pre-process cleansing. For graduate students the dataset consists of University Name, Major, Degree, Season, Decision, Decision Method, Decision Date, Undergraduate GPA, Is New GRE Verbal, GRE Quant, GRE Writing, Status, Postdate Comments, Research Experience, Recommendations and Undergraduate GPA. For Under graduate students dataset consists of Student profile and SAT scores.

4.2. Under Graduate student dataset:

Under Graduate student data is taken from the College rank score card website https://collegescorecard.ed.gov/data/. The data before cleaning looked like below.

ï»¿UNITID	OPEID	OPEID6	INSTNM	CITY	STABBR	ZIP	ACCREDAGEI	INSTURL	NPCURL	SCH_DEG	HCM2	MAIN	NUMBRANCI	PREDDEG	HIGHDEG	CONTROL	ST_FIPS	REGION
100654	100200	1002	Alabama A &	Normal	AL	35762	NULL	NULL	NULL	NULL	NULL	1	1	3	4	1	1	5
100663	105200	1052	University of	Birmingham	AL	35294-0110	NULL	NULL	NULL	NULL	NULL	1	1	3	4	1	1	5
100690	2503400	25034	Amridge Uni	Montgomery	AL	36117-3553	NULL	NULL	NULL	NULL	NULL	1	1	3	4	2	1	5
100706	105500	1055	University of	Huntsville	AL	35899	NULL	NULL	NULL	NULL	NULL	1	1	3	4	1	1	5
100724	100500	1005	Alabama Sta	Montgomery	AL	36104-0271	NULL	NULL	NULL	NULL	NULL	1	1	3	4	1	1	5
100751	105100	1051	The Universi	Tuscaloosa	AL	35487-0166	NULL	NULL	NULL	NULL	NULL	1	1	3	4	1	1	5
100760	100700	1007	Central Alab	Alexander Ci	AL	35010	NULL	NULL	NULL	NULL	NULL	1	1	2	2	1	1	5
100812	100800	1008	Athens State	Athens	AL	35611	NULL	NULL	NULL	NULL	NULL	1	1	3	3	1	1	5
100830	831000	8310	Auburn Univ	Montgomery	AL	36117-3596	NULL	NULL	NULL	NULL	NULL	1	1	3	4	1	1	5
100858	100900	1009	Auburn Univ	Auburn	AL	36849	NULL	NULL	NULL	NULL	NULL	1	1	3	4	1	1	5
100937	101200	1012	Birmingham	Birmingham	AL	35254	NULL	NULL	NULL	NULL	NULL	1	1	3	3	2	1	5

4.3. Data Preprocessing:

In order to use the obtained data for our analysis, we need to do the preprocessing and cleansing, as there are lots of anomalies in the dataset. For this we use pandas and numpy frameworks.

Cleansing the data was done by

- Removing the irrelevant columns by using the drop column feature
- Filling the null values with the appropriate value or deleting the row containing null values.
- Removing the spaces in the data and reducing the size of the dataset.

In our graduate dataset, The GRE scores were also cleansed since they contained scores of both old and new versions of the examination. Similarly the GPA scores available were based on different point systems, so all the GPA scores were uniformly scaled to 4 point scale by using normalize functions.

$$x_{\text{normalized}} = (x - x_{\text{minimum}}) / (x_{\text{maximum}} - x_{\text{minimum}})$$

Where x is the value of the GPA

4.4. Exploratory data Analysis:

Exploratory data Analysis is a technique, which employs a variety of techniques. It consists of various techniques like below.

1.Plotting raw data
2. Plotting simple statistics such as mean , standard deviation, etc.
3. Positioning such plots, so as to maximize our natural pattern recognition.

This is the description of the data.

```
data.describe()
```

	decdate_ts	cgpa	greV	greQ	greA	gre_subject	post_timestamp
count	6.145400e+04	55589.000000	61474.000000	61474.000000	61474.000000	7175.000000	6.147400e+04
mean	1.431551e+09	3.715970	231.556333	248.826447	4.144757	796.411150	1.431763e+09
std	9.540728e+07	0.506153	174.575147	208.551820	1.111126	122.305977	8.079993e+07
min	-1.000000e+00	0.400000	130.000000	130.000000	0.000000	310.000000	1.263283e+09
25%	1.363244e+09	3.520000	155.000000	157.000000	3.500000	710.000000	1.363417e+09
50%	1.426662e+09	3.750000	161.000000	164.000000	4.000000	800.000000	1.427094e+09
75%	1.490771e+09	3.900000	167.000000	170.000000	5.000000	890.000000	1.491030e+09
max	1.360120e+10	9.990000	800.000000	800.000000	6.000000	990.000000	1.562569e+09

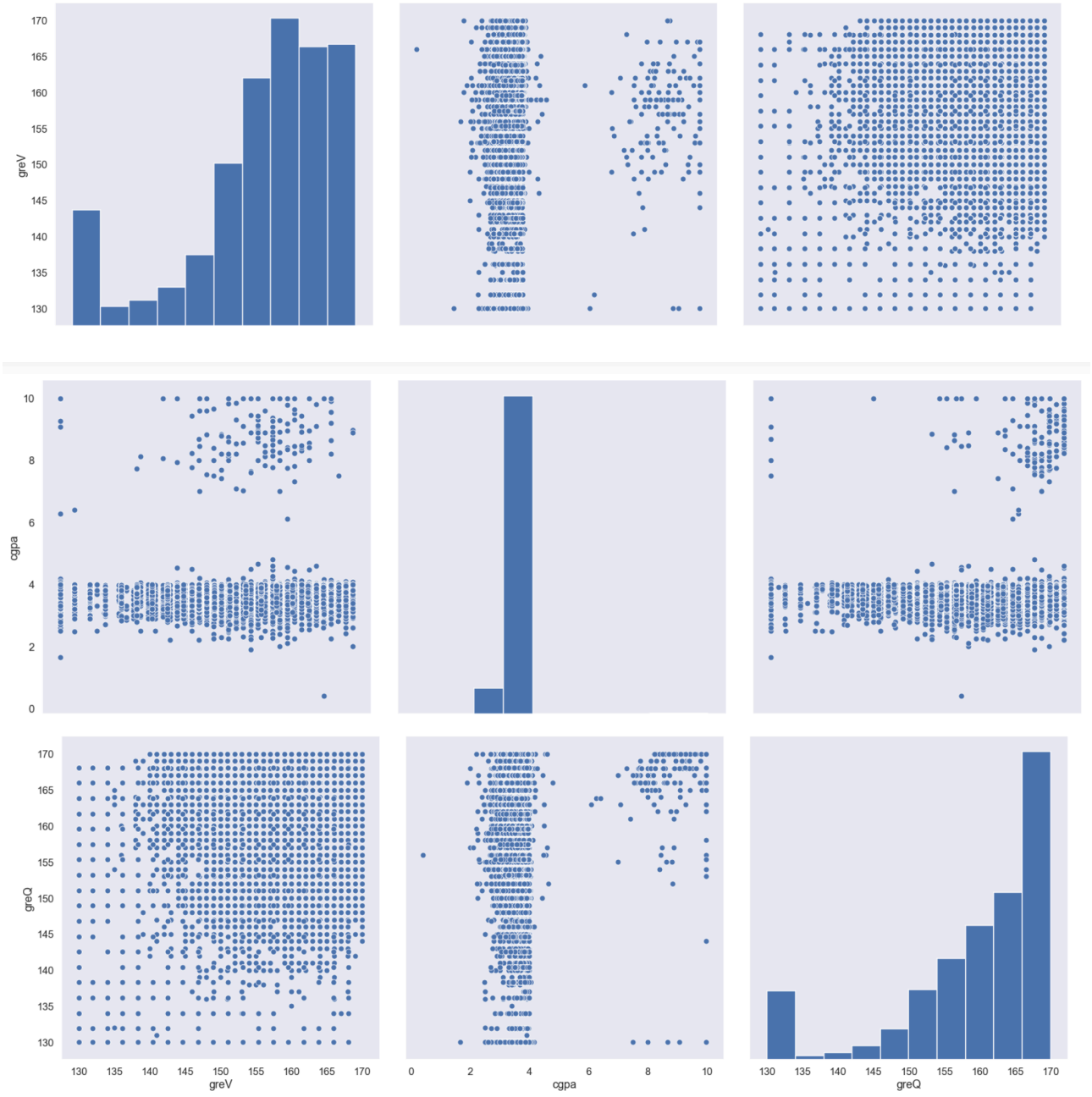
The Processed data will look like below.

```
data.columns = ['univName', 'major', 'program', 'season', 'decision', 'Method', 'decdate', 'decdate_ts', 'cgpa', 'greV'
          'greA', 'is_new_gre', 'gre_subject','status', 'post_data', 'post_timestamp', 'comments']
data.head()
```

	univName	major	program	season	decision	Method	decdate	decdate_ts	cgpa	greV	greQ	greA	is_new_gre	gre_subject	status	post
0	University Of Waterloo	Systems Design Engineering	MS	NaN	Accepted	Website	(1, 7, 2019)	1.561964e+09	NaN	NaN	NaN	NaN	NaN	NaN	International	
1	Northeastern University	Electrical Engineering	PhD	F19	Rejected	Website	(8, 7, 2019)	1.562569e+09	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2	The University Of Auckland	Electrical And Electronic Engineering	MS	NaN	Accepted	Website	(19, 6, 2019)	1.560928e+09	NaN	NaN	NaN	NaN	NaN	NaN	International	
3	Radford University	Counseling Psychology PsyD.	Other	F19	Accepted	Phone	(4, 3, 2019)	1.551686e+09	NaN	NaN	NaN	NaN	NaN	NaN	American	
4	University Of Chittagong	Computer Science	MS	NaN	NaN	Other	(9, 7, 2019)	1.562656e+09	3.2	163.0	168.0	4.0	True	NaN	International	

Exploratory data analysis: pair plots of GRE verbal, GRE Quantitative and GPA.

```
sns.pairplot(data, palette="husl", x_vars=["greV","cgpa","greQ"], y_vars=["greV","cgpa","greQ"], height=8)
plt.show()
```



Undergraduate Data EDA:

In Undergraduate data, we have taken below few rows of data like Institution name, city, tuition Fees, Sat Score, Admission rate, Debt and Men Ratio.

```
INSTNM', 'CITY', 'STABBR' ,'TUITIONFEE_OUT',  'SAT_AVG_ALL', 'ADM_RATE_ALL',
'DEBT_MDN_SUPP', 'UGDS_MEN'])
```

This data will be used for training the model and test data as SAT score and Maximum tuition fees.

5. ANALYSIS & METHODOLOGY

Here I have used Knowledge based recommendation System where User inputs are taken into account and compare with the training data.

For Graduate University Recommendation I have used Case based knowledge recommendation as it will take the User inputs and compare with trained data.

For Undergraduate Recommendation System, I have used Constraint based Knowledge recommendation system where user inputs taken into account as constraints and based on the constraints I compared with trained data.

I used two different models like K-Nearest Neighbors for Graduate data and Feature weighted algorithms for Undergraduate data.

K Nearest Neighbor:

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.

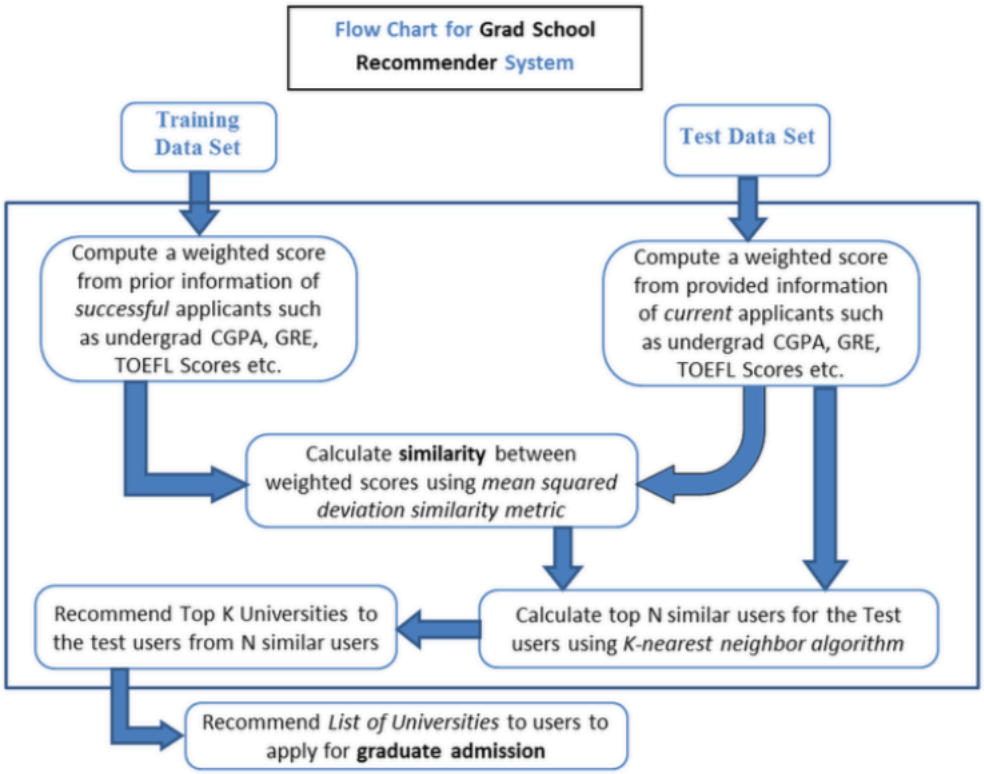
Algorithm is stated as below.

Input: undergraduate university, department, CGPA, GRE Scores of User

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

Output: highly recommended N Outgoing University analyzing Universal Table of Previous Successful Students

Flowchart of the graduate recommendation System:



6. IMPLEMENTATION

Training data for the KNN algorithm:

	univName	cgpa	greV	greQ	greA
14	Ohio State University	4.00	150.0	166.0	3.0
17	Texas A&M University	3.57	157.0	151.0	5.5
46	University Of California, Irvine	3.66	155.0	167.0	4.0
64	Boston University	3.10	161.0	157.0	4.0
203	Oregon State University	3.38	154.0	170.0	4.0

```
def knn(trainingSet, testInstance, k):
    print(k)
    distances = {}
    sort = {}
    length = testInstance.shape[1]

    for x in range(len(trainingSet)):

        dist = euclideanDistance(testInstance, trainingSet.iloc[x], length)

        distances[x] = dist[0]

    sorted_d = sorted(distances.items(), key=lambda x: x[1])


    neighbors = []

    for x in range(k):
        neighbors.append(sorted_d[x][0])

    classVotes = {}

    for x in range(len(neighbors)):
        response = trainingSet.iloc[neighbors[x]][-1]
        if response in classVotes:
            classVotes[response] += 1
        else:
            classVotes[response] = 1

    sortedVotes = sorted(classVotes.items(), key=lambda x: x[1], reverse=True)

    return(sortedVotes, neighbors)
```

Implementation of Feature weighted algorithm for Undergraduate universities:

The weightage of all the features are taken and find the similarity score. Based on the similarity score, the universities with highest similarities will be recommended to student. Suppose w1, w2 are weights and f1 and f2 are features the similarity is calculated by formula Similarity

$$score = w1* f1+w2*(1-f2)$$

Algorithm is stated as below.

Input: SAT Score and Maximum tuition fees of User

1. For getting recommendation, iterate from 1 to number of trained data
2. Find the rows in the training data similar to the user provided SAT score and max tuition fees.
3. Calculate the weightage of both the attributes and calculate the score as acceptance rate
4. Sort the distances in ascending order
5. Get top k rows and recommend to the user

Output: Top 5 Recommended Universities

6. RESULTS

For graduate University recommendation KNN algorithm**, for the input test = [145, 156, 4, 3.8] ,** The result is provided as

```
k = 7

result,neigh= knn(processed_data, test, k)


list1 = []
list2 = []
for i in result:
    list1.append(i[0])
    list2.append(i[1])
for i in list1:
    print(i)
```

```
7
University Of Colorado, Boulder
University Of Florida
University Of Arizona
University Of Pennsylvania (UPenn)
Syracuse University
University Of Texas At Austin
Emory University
```

```
from sklearn.neighbors import KNeighborsClassifier
neigh = KNeighborsClassifier(n_neighbors=5)
neigh.fit(processed_data.iloc[:,0:4], data['univName'])

print(neigh.predict(test))
```

```
['Syracuse University']
```

For Undergraduate Universities recommendation, weighted algorithm outputs:

Input of the data for SAT Score and Maximum Tuition Fees:

```
please input your sat score:1500
please input your maximum tuition you can accept:10000
{'Fort Valley State University': <__main__.CollageInfo object at 0x119e3a0f0>, 'Brigham Young University-Idaho':
__main__.CollageInfo object at 0x119e3df98>, 'Bemidji State University': <__main__.CollageInfo object at 0x119e76
18>, 'Southwest Minnesota State University': <__main__.CollageInfo object at 0x119e76a20>, 'Alcorn State Universi
y': <__main__.CollageInfo object at 0x119e76a90>, 'Delta State University': <__main__.CollageInfo object at 0x119
76b38>, 'Mississippi Valley State University': <__main__.CollageInfo object at 0x119e76d30>, 'Rust College': <__m
in__.CollageInfo object at 0x119e76f98>, 'United States Merchant Marine Academy': <__main__.CollageInfo object at
0x119ec19b0>, 'Dickinson State University': <__main__.CollageInfo object at 0x119ed3048>, 'Mayville State Univers
ty': <__main__.CollageInfo object at 0x119ed3ef0>, 'Minot State University': <__main__.CollageInfo object at 0x11
ed3fd0>, 'Central State University': <__main__.CollageInfo object at 0x119ece860>, 'Youngstown State University':
<__main__.CollageInfo object at 0x119eeed68>, 'Paul Quinn College': <__main__.CollageInfo object at 0x119f054e0>,
'The University of Texas of the Permian Basin': <__main__.CollageInfo object at 0x119f05fd0>, 'Brigham Young Univ
rsity-Hawaii': <__main__.CollageInfo object at 0x119f05668>, 'Piedmont International University': <__main__.Colla
eInfo object at 0x119efd668>}
```

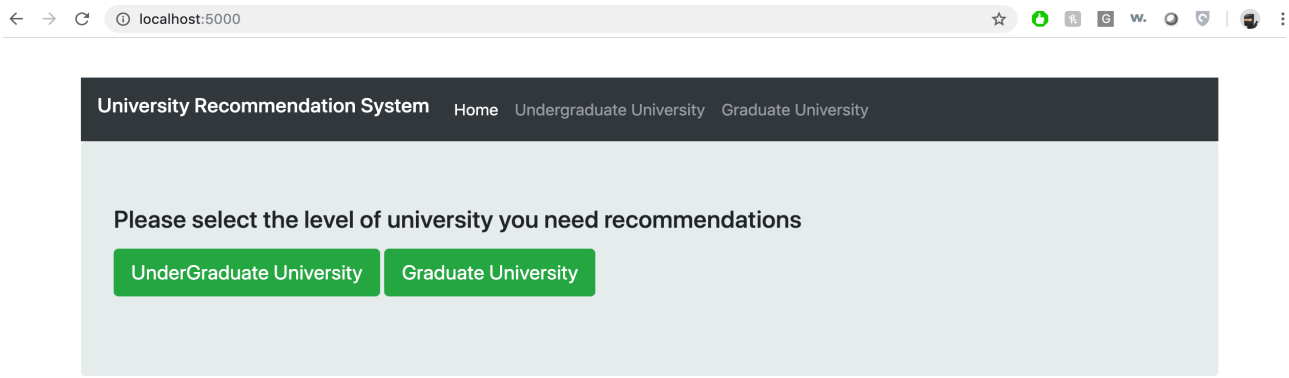
Output:

```
college : Brigham Young University-Idaho
score: 0.8108348810996646
city:Rexburg    state:ID        rank:170        tuition:3920.0  sat:1036.0      AC:0.95830221558376     debt:6606
0       Mal:0.4236


college : Mississippi Valley State University
score: 0.7441481637097037
city:Itta Bena  state:MS        rank:1243       tuition:6116.0  sat:825.0       AC:0.84452975047984     debt:1856
.0      Mal:0.4146


college : Alcorn State University
score: 0.6810460877865222
city:Alcorn State       state:MS        rank:1127       tuition:6546.0  sat:892.0       AC:0.78440808469682     d
bt:19000.0      Mal:0.362


college : The University of Texas of the Permian Basin
score: 0.6696032133488956
city:Odessa     state:TX        rank:900        tuition:6958.0  sat:953.0       AC:0.81484315225707     debt:8300
0       Mal:0.4428


college : Delta State University
score: 0.637365259808179
city:Cleveland  state:MS        rank:997        tuition:6418.0  sat:1028.0      AC:0.89407744874715     debt:1239
.5      Mal:0.4138
```

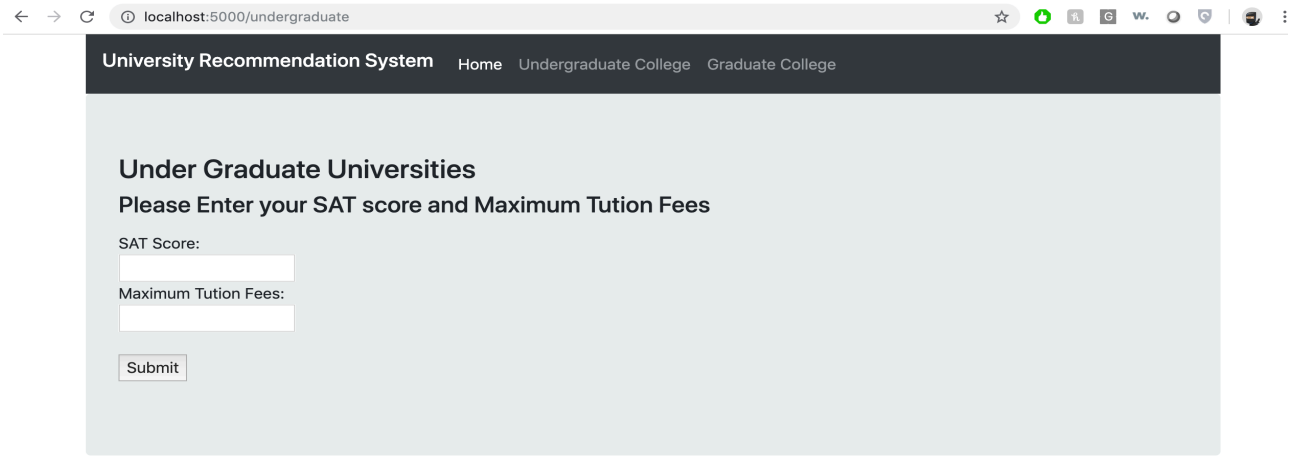
8. WEB APPLICATION:

I have developed a web application for University Recommendation System. Below are the Screenshots for this.

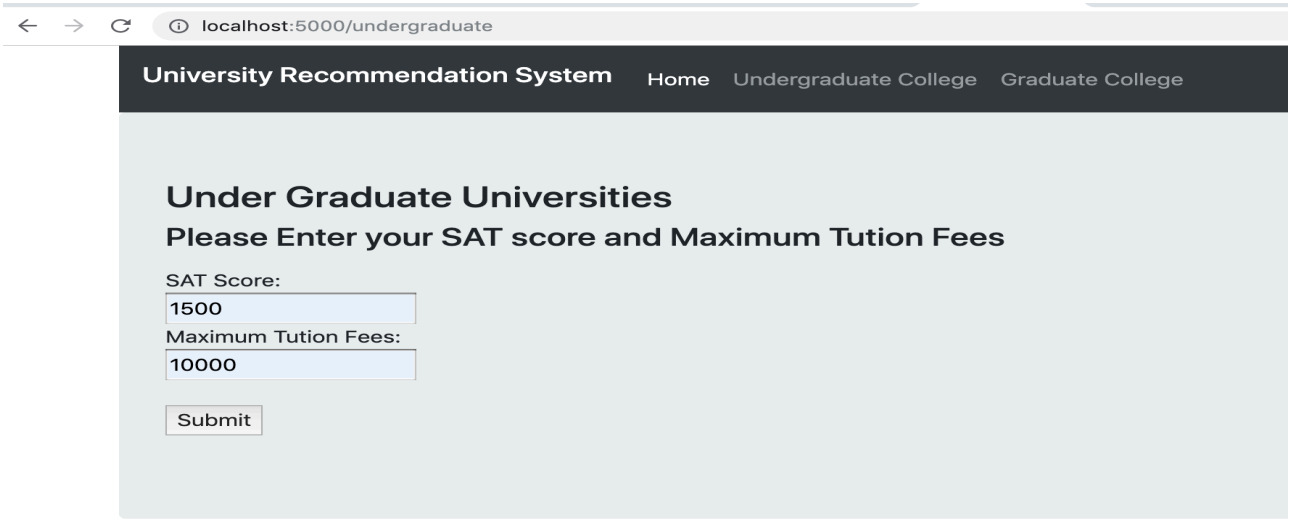
Home Page: This is the home page which has links to Under graduate and graduate Recommendation system



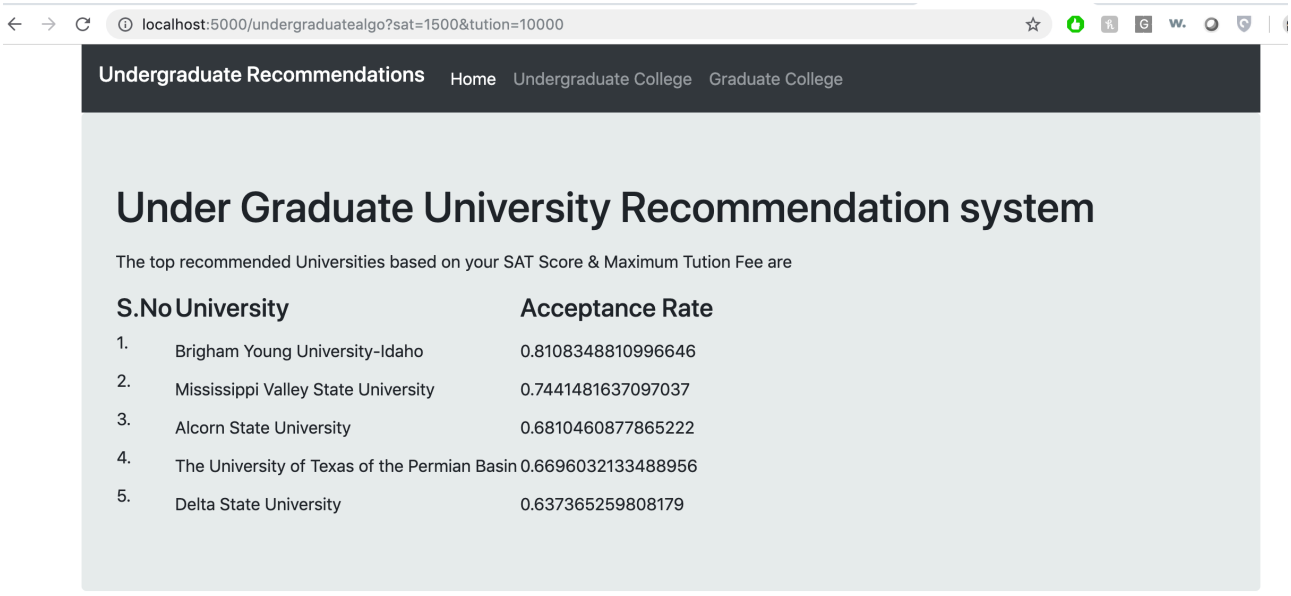
2. Once the user clicks on the Undergraduate College, he/she will land on this page where they can provide the SAT score and Maximum Tuition fees for their college recommendation.



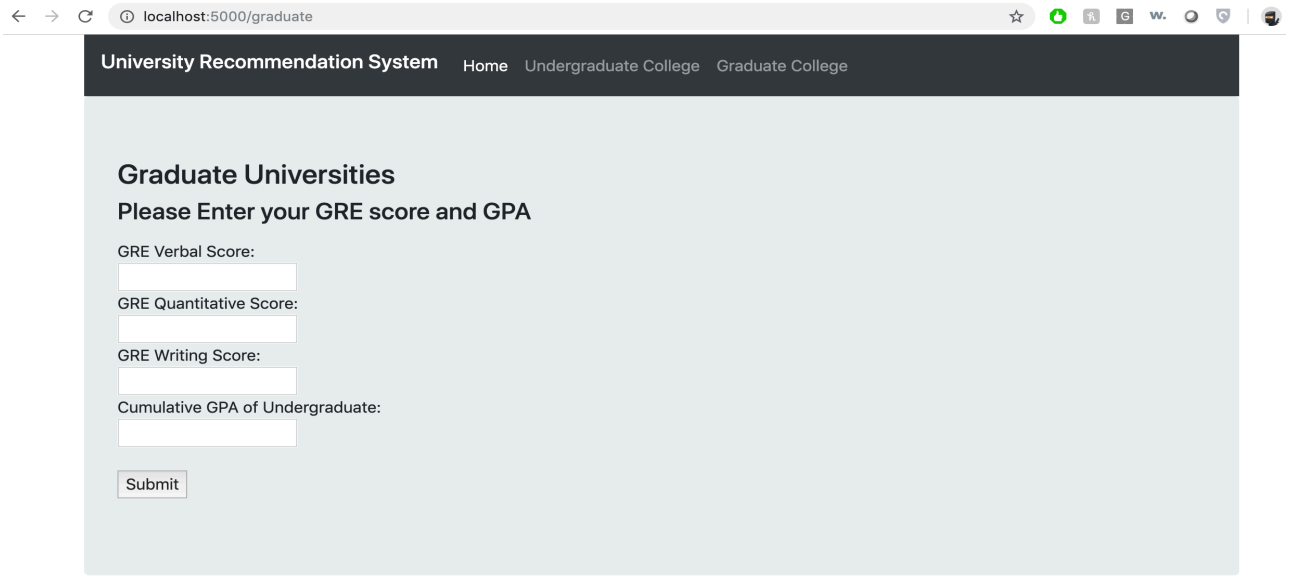
3. If user provides SAT Score as 1500 and Max tuition fees as 10000, Then the output of University names he gets is in the below screenshot.



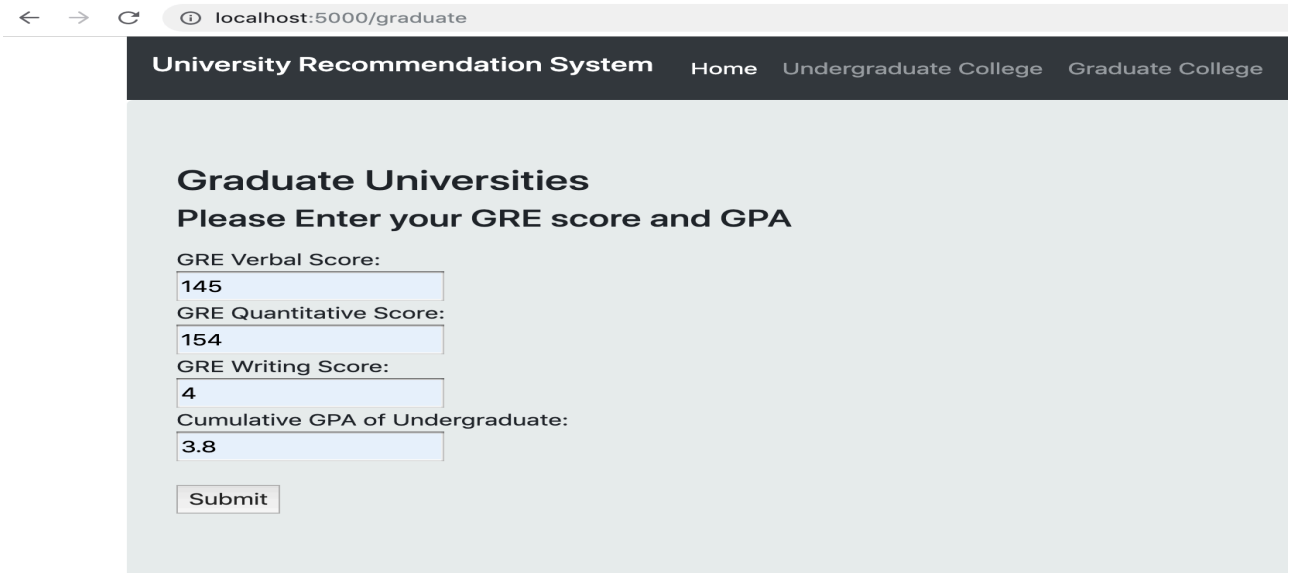
Once user submits he will get these recommendations as below.



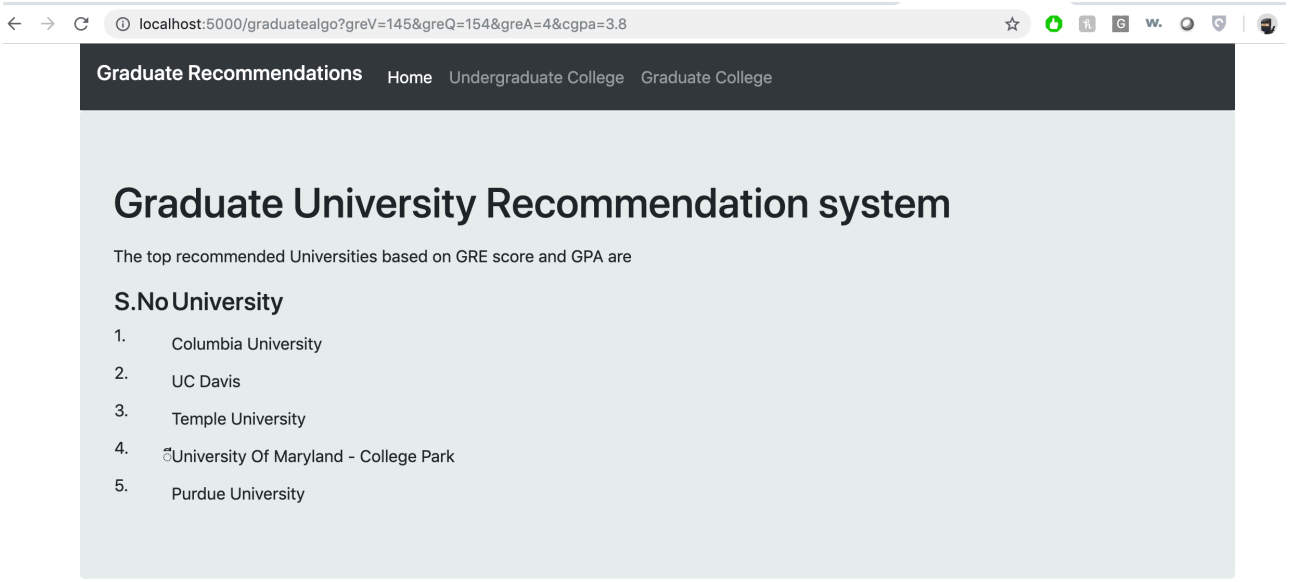
4. To Get the Graduate University recommendation list, User need to provide his/her GRE scores and GPA of the undergraduate University.



7. If user provides, GRE score and CGPA as below and click on submit, the recommendation system will provide output as below.



8. This is the output for the Graduate Recommendation System.



9. DEMO VIDEO:

10. CONCLUSION:

This project helps students in the decision making of the universities in which they apply. The data of the previous successful applicants can be taken into account. The data from the academic records of applicants is very important for the admission seekers in foreign. In this research, I have developed a technique of using those academic records of successful applicants for making school recommender system, which can help the current admission seekers. At first, I calculate similarity between training and test data set based on weighted scores. The weighted scores are calculated from prior information of successful applicants such as undergrad CGPA, GRE, TOEFL Scores and all other relevant records found in the universal database. I have used K-nearest Neighbor algorithm for graduate universities and feature weighted algorithm for Undergraduate Universities in order to calculate top N similar users and then recommend top K universities to the users. Our proposed recommender system will recommend list of universities to applicants trying to pursue higher study abroad and assist them to apply for graduate admission in appropriate universities.

11. REFERENCES:

1. https://ieeexplore.ieee.org/document/7760053
2. https://www.semanticscholar.org/paper/Recommender-System-for-Graduate-Studies-in-USA-Suresh/22924fda3f293f80a3f62f32799c08d0b81a9b20
3. https://www.researchgate.net/publication/311758642_Graduate_school_recommender_system_Assisting_admission_seekers_to_apply_for_graduate_studies_in_appropriate_graduate_schools
4. http://jmcauley.ucsd.edu/cse258/projects/fa15/026.pdf
5. https://devpost.com/software/graduate-school-recommendation-system
6. http://paper.ijcsns.org/07_book/201801/20180111.pdf