Recommending Graduate Universities

```
import sys
import os
import collections
from collections import defaultdict
import numpy as np
import pandas as pd
from scipy import stats
import re
```

Dataset Loading and Description

```
cs_file = "/content/Final.csv"
data = pd.read_csv(cs_file)
data.shape
```

```
(39494, 20)
```

Data Scraping

```
data.drop(data.columns[data.columns.str.contains('unnamed',case = False)],axis = 1, inplace =
data.head(14546)
```

	univName	major	program	season	decision	Method	decdate	decdate_t
0	University Of Waterloo	Systems Design Engineering	MS	NaN	Accepted	Website	(1, 7, 2019)	1.561964e+0
2	The University Of Auckland	Electrical And Electronic Engineering	MS	NaN	Accepted	Website	(19, 6, 2019)	1.560928e+0
3	Radford University	Counseling Psychology	Other	F19	Accepted	Phone	(4, 3, 2019)	1.551686e+0

```
data.columns = ['univName', 'major', 'program', 'season', 'decision', 'Method', 'decdate', 'd
           'greA', 'is_new_gre', 'gre_subject','status', 'post_data', 'post_timestamp', 'comm
data.head(14546)
```

	univName	major	program	season	decision	Method	decdate	decdate_t
0	University Of Waterloo	Systems Design Engineering	MS	NaN	Accepted	Website	(1, 7, 2019)	1.561964e+0
2	The University Of Auckland	Electrical And Electronic Engineering	MS	NaN	Accepted	Website	(19, 6, 2019)	1.560928e+0
3	Radford University	Counseling Psychology PsyD.	Other	F19	Accepted	Phone	(4, 3, 2019)	1.551686e+0
5	Georgia Southern University	Speech Language Pathology	MS	F19	Accepted	E-mail	(2, 7, 2019)	1.562051e+0
7	New York University (NYU) - Steinhardt	Communication Sciences And Disorders	MS	F19	Accepted	E-mail	(8, 7, 2019)	1.562569e+0
...	...	...	...	...	...	...	...	..
39475	Purdue University - West Lafayette	Computer Graphics Technology	MS	F19	Accepted	E-mail	(12, 2, 2019)	1.549958e+0
39476	University Of Georgia	English	PhD	F19	Accepted	Website	(12, 2, 2019)	1.549958e+0
39482	Clemson	Urban And Regional	PhD	F19	Accepted	E-mail	(10, 2,	1.549786e+0

Conditinal Data Scraping

```
data['major'] = 'Computer science'
data.head(201)
```

	univName	cgpa	greV	greQ	greA	major
393	Columbia University	3.65	165.0	162.0	5.5	Computer science
395	University Of Michigan	3.65	165.0	162.0	5.5	Computer science
582	Columbia University	2.90	143.0	142.0	0.0	Computer science
872	Columbia University	4.00	170.0	170.0	0.0	Computer science
873	Columbia University	4.00	170.0	170.0	0.0	Computer science
...	...	...	...	...	...	...
34636	University Of Michigan	3.90	168.0	168.0	5.5	Computer science
34685	Columbia University	3.72	168.0	164.0	4.0	Computer science
34867	University Of Michigan	3.88	164.0	168.0	4.5	Computer science
34902	University Of Michigan	3.79	160.0	170.0	4.0	Computer science
35332	Columbia University	3.90	170.0	167.0	4.0	Computer science

201 rows × 6 columns

Dataset Analysis

```
data = data[data['decision'] =='Accepted']

data.shape
```

```
(14546, 18)
```

```
data = data[pd.notnull(data['greQ'])]
data.shape
```

```
(4993, 18)
```

```
data['greQ'] = data['greQ'].fillna(130)
data['greV'] = data['greV'].fillna(130)
data['greA'] = data['greA'].fillna(0)
data.greA.head()
```

```
7     4.0
14    3.0
17    5.5
28    4.0
46    4.0
Name: greA, dtype: float64
```

Unique Universities Shortlisitng

```
uni_names = data['univName'].unique()

similar_univs = pd.DataFrame({'univName':uni_names})
similar_univs
```

	univName
0	New York University (NYU) - Steinhardt
1	Ohio State University
2	Texas A&M University
3	St. Johns University
4	University Of California, Irvine
...	...
755	University Of San Diego
756	Indiana University At Bloomington
757	University Of California Los Angeles
758	University Of Minnesota (UMN)
759	University Of Zurich

760 rows × 1 columns

Statistical Calculation (Description)

```
data.describe()
```

	decdate_ts	cgpa	greV	greQ	
count	4.993000e+03	4481.000000	4993.000000	4993.000000	49
mean	1.551716e+09	3.755642	158.714801	160.622271	
std	2.072409e+06	0.644852	6.505974	7.739675	
min	1.516694e+09	1.500000	135.000000	134.000000	
25%	1.550563e+09	3.540000	154.000000	155.000000	
50%	1.551427e+09	3.760000	159.000000	162.000000	

Data Preprocessing

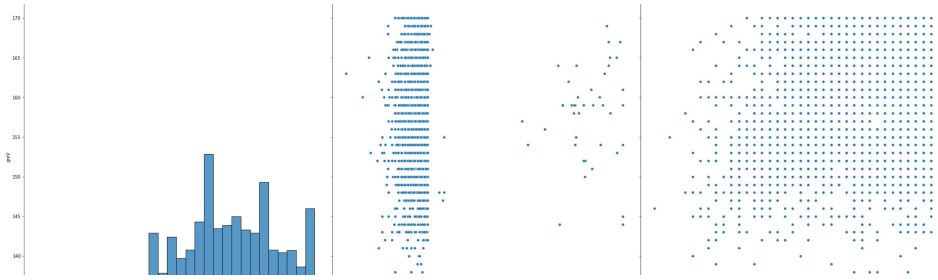
```
def convert_quant_score(quant_score):
    quant_list = []
    quant_score = quant_score.tolist()
    for old_quant in quant_score:
        if old_quant <= 170:
            quant_list.append(old_quant)
            continue
        else:
            old_quant = old_quant/4.7
            if old_quant <=130:
                quant_list.append(130)
            else:
                quant_list.append(old_quant)
    return quant_list

def convert_verbal_score(verbal_score):
    verbal_list = []
    verbal_score = verbal_score.tolist()
    for old_verbal in verbal_score:
        if old_verbal <= 170:
            verbal_list.append(old_verbal)
            continue
        else:
            old_verbal = old_verbal/4.7
            if old_verbal <=130:
                verbal_list.append(130)
            else:
                verbal_list.append(old_verbal)
    return verbal_list

data['greQ'] = convert_quant_score(data['greQ'])
data['greV'] = convert_verbal_score(data['greV'])
```

Exploratory Data Analysis

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.pairplot(data, palette="husl", x_vars=["greV","cgpa","greQ"], y_vars=["greV","cgpa","greQ
plt.show()
```



```
def normalize_gpa(data2,cgpa,totalcgpa):
    cgpa = data2[cgpa].tolist()
    totalcgpa = data2[totalcgpa].tolist()
    for i in range(len(cgpa)):
        if totalcgpa[i] != 0:
            cgpa[i] = cgpa[i] / totalcgpa[i]
        else:
            cgpa[i] = 0
    data2['cgpa'] = cgpa
    return data2
```

```
data = data.drop('major',1)
data = data.drop('program',1)
data = data.drop('season',1)
data = data.drop('decision',1)
data = data.drop('Method',1)
data = data.drop('decdate',1)
data = data.drop('decdate_ts',1)
data = data.drop('is_new_gre',1)
data = data.drop('gre_subject',1)
data = data.drop('status',1)
data = data.drop('post_data',1)
data = data.drop('post_timestamp',1)
data = data.drop('comments',1)

university_list = list(set(data['univName'].tolist()))
for i in range(len(university_list)):
    if(len(data[data['univName'] == university_list[i]]) < 100):
        data = data[data['univName'] != university_list[i]]
data = data.dropna()

data.head()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1
  """Entry point for launching an IPython kernel.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1
  This is separate from the ipykernel package so we can avoid
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4
  after removing the cwd from sys.path.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5
  """
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:7
  import sys
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:8

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:9
  if __name__ == '__main__':
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1
  # Remove the CWD from sys.path while we load stuff.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1
  # This is added back by InteractiveShellApp.init_path()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1
  if sys.path[0] == '':
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1
  del sys.path[0]
```

```
                univName  cgpa  greV  greQ  greA
```

```
processed_data = data[['greV', 'greQ', 'greA', 'cgpa', 'univName']]
processed_data.head()

processed_data.to_csv('/content/Final.csv')


import math
from sklearn import neighbors, datasets
from numpy.random import permutation
import matplotlib.pyplot as plt
from sklearn import svm
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import precision_recall_fscore_support


random_indices = permutation(data.index)
test_cutoff = math.floor(len(data)/5)
print(test_cutoff)
test = processed_data.loc[random_indices[1:test_cutoff]]
train = processed_data.loc[random_indices[test_cutoff:]]
train_output_data = train['univName']
print("train Output data", train_output_data)
train_input_data = train
train_input_data = train_input_data.drop('univName',1)
print("train input data", train_input_data)
test_output_data = test['univName']
```

```
print("test Output data", test_output_data)
test_input_data = test
test_input_data = test_input_data.drop('univName',1)
print("test input data", test_input_data)
```

```
45
train Output data 5249        Columbia University
35751    University Of Michigan
6853     University Of Michigan
13043       Columbia University
15258       Columbia University
                 ...
35412    University Of Michigan
25038       Columbia University
11773    University Of Michigan
4830     University Of Michigan
32915       Columbia University
Name: univName, Length: 183, dtype: object
train input data         greV   greQ  greA  cgpa
5249   154.0  170.0   0.0  3.94
35751  164.0  169.0   4.0  3.94
6853   158.0  170.0   4.0  4.00
13043  157.0  166.0   3.5  3.77
15258  155.0  158.0   5.0  4.00
...      ...    ...   ...   ...
35412  166.0  170.0   5.0  3.96
25038  162.0  167.0   5.0  3.98
11773  157.0  170.0   3.5  3.73
4830   163.0  170.0   4.0  4.00
32915  161.0  170.0   4.0  3.90

[183 rows x 4 columns]
test Output data 24930    University Of Michigan
11764    University Of Michigan
28350       Columbia University
18635       Columbia University
28158       Columbia University
873         Columbia University
24760       Columbia University
10403       Columbia University
14040    University Of Michigan
14477    University Of Michigan
4547        Columbia University
28469    University Of Michigan
30175    University Of Michigan
12939       Columbia University
16015    University Of Michigan
15790       Columbia University
25091    University Of Michigan
35856    University Of Michigan
24585    University Of Michigan
26632       Columbia University
20786    University Of Michigan
11001       Columbia University
36731    University Of Michigan
16897       Columbia University
```

```
1595        Columbia University
21678    University Of Michigan
6910        Columbia University
30476    University Of Michigan
15393    University Of Michigan
34685       Columbia University
```

```
def euclideanDistance(data1, data2, length):
    distance = 0
    for x in range(length):
        distance += np.square(data1[x] - data2[x])
    return np.sqrt(distance)

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)


testSet = [[142, 153, 5.0, 3.6]]
test = pd.DataFrame(testSet)
test.shape
```

```
(1, 4)
```

```
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
Columbia University
University Of Michigan
```

Final University Recommendation

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

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

```
['University Of Michigan']
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have
  "X does not have valid feature names, but"
```