University Admissions Recommendation System

**Problem Statement**

The main cause to create this system is to save time and choosing good options for students who are passing out of college and searching options which are suitable for their grades and get selected into those colleges and it also helps for those want to study in different places of the world.

**Table of Contents**

1. [**Introduction**](file:///C:\Users\home\Downloads\Heart_Disease_Prediction.html#Introduction)
2. [**Exploratory Data Analysis (EDA)**](file:///C:\Users\home\Downloads\Heart_Disease_Prediction.html#EDA)
3. [**Model Selection and Evaluation**](file:///C:\Users\home\Downloads\Heart_Disease_Prediction.html#ModelSelection)
4. [**Conclusion**](file:///C:\Users\home\Downloads\Heart_Disease_Prediction.html#Conclusion)
5. [**References**](file:///C:\Users\home\Downloads\Heart_Disease_Prediction.html#References)

**Introduction**

The major problem in University recommendation system is finding suitable university for a student with the marks he got. There are different factors that are taken into consideration while giving the result like GRE Score, Internship experience and research experience etc..., a student can easily find the university suitable for him easily and resolves his problem of deciding a college in very less time.

### **1.1 Dataset description**

The dataset was obtained from Kaggle and the dataset contains features like GRE Score, Internship experience and research etc. all the data required to recommend universities

### **1.2 Installing and importing dependencies**

**The following packages are essential to running this project successfully: numpy, pandas, matplotlib, sklearn,**

**1. Importing Required packages**

**import** pandas **as** pd

**import** seaborn **as** sns

**2. Load Dataset**

df=pd.read\_csv("/content/university\_admissions.csv")

df

|  | **userName** | **major** | **researchExp** | **industryExp** | **specialization** | **toeflScore** | **program** | **department** | **toeflEssay** | **internExp** | **...** | **termAndYear** | **confPubs** | **ugCollege** | **gmatA** | **cgpa** | **gmatQ** | **cgpaScale** | **gmatV** | **univName** | **admit** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 143saf | Systems and Control | 0 | 18 | Robotics | 112.0 | MS | Instrumentation & Control | 26 | 5.0 | ... | Fall - 2015 | 0 | Dharamsinh Desai University | NaN | 8.50 | NaN | 10.0 | NaN | Worcester Polytechnic Institute | 1.0 |
| **1** | 7790ashish | Manufacturing Engineering | 0 | 0 | NaN | NaN | MS | 0 | NaN | 0.0 | ... | Fall - 2013 | 0 | NaN | NaN | 0.00 | NaN | 0.0 | NaN | Worcester Polytechnic Institute | 1.0 |
| **2** | AB25 | (MIS / MSIM / MSIS / MSIT) | 0 | 66 | NaN | 94.0 | MS | Computer Engineering | 21 | 0.0 | ... | Fall - 2015 | 0 | IET DAVV | NaN | 78.28 | NaN | 100.0 | NaN | Worcester Polytechnic Institute | 1.0 |
| **3** | abhijitg | NaN | 0 | 0 | NaN | NaN | NaN | 0 | NaN | 0.0 | ... | NaN | NaN | NaN | NaN | 0.00 | NaN | 0.0 | NaN | Worcester Polytechnic Institute | 1.0 |
| **4** | abhijitgang | MIS | 0 | 0 | NaN | 81.0 | MS | computer | NaN | 0.0 | ... | Fall - 2011 | 0 | Pune University | NaN | 57.00 | NaN | 100.0 | NaN | Worcester Polytechnic Institute | 1.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **24160** | saikironbhatt | Electrical Engineering | 0 | 0 | Analog and Mixed Signal Design | 100.0 | MS | Electronics and Instrumentation Engineering | NaN | 0.0 | ... | Fall - 2013 | 0 | SRM | NaN | 8.58 | NaN | 10.0 | NaN | University of California Santa Barbara | 0.0 |
| **24161** | samish | Computer Science | 0 | 0 | NaN | 99.0 | MS | Computer Science | 27 | 0.0 | ... | Fall - 2012 | 0 | International Institute of Information Technol... | NaN | 8.70 | NaN | 10.0 | NaN | University of California Santa Barbara | 0.0 |
| **24162** | sandeep024 | Computer Science | 0 | 0 | General AI High performance computing | 107.0 | MS | computer science | NaN | 0.0 | ... | Fall - 2013 | 0 | Sir MVIT | NaN | 78.00 | NaN | 100.0 | NaN | University of California Santa Barbara | 0.0 |
| **24163** | sanksank | Electronic and Telecommunication Engineering | 0 | 0 | Signal Processing | 108.0 | MS | Electronics and Telecommunication | 27 | 0.0 | ... | Fall - 2012 | 0 | MU | NaN | 69.00 | NaN | 100.0 | NaN | University of California Santa Barbara | 0.0 |
| **24164** | SANStudent | Computer Science | 0 | 0 | NaN | 116.0 | MS | Computer Science | NaN | 0.0 | ... | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |

24165 rows × 26 columns

**3.Exploratory Data Analysis (EDA)**

df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 24165 entries, 0 to 24164

Data columns (total 26 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 userName 24165 non-null object

1 major 23970 non-null object

2 researchExp 24165 non-null int64

3 industryExp 24165 non-null int64

4 specialization 14590 non-null object

5 toeflScore 22100 non-null float64

6 program 24001 non-null object

7 department 24164 non-null object

8 toeflEssay 5340 non-null object

9 internExp 24159 non-null float64

10 greV 23570 non-null float64

11 greQ 23588 non-null float64

12 userProfileLink 24164 non-null object

13 journalPubs 24000 non-null object

14 greA 22843 non-null float64

15 topperCgpa 24164 non-null float64

16 termAndYear 24000 non-null object

17 confPubs 24000 non-null object

18 ugCollege 23088 non-null object

19 gmatA 68 non-null float64

20 cgpa 24164 non-null float64

21 gmatQ 70 non-null float64

22 cgpaScale 24164 non-null float64

23 gmatV 66 non-null float64

24 univName 24164 non-null object

25 admit 24164 non-null float64

dtypes: float64(12), int64(2), object(12)

memory usage: 4.8+ MB

df.researchExp.value\_counts()

0 23482

6 146

12 103

24 66

2 64

3 47

4 33

18 32

36 26

10 21

7 18

5 16

8 13

30 12

9 11

48 10

16 10

17 8

1 6

15 5

29 5

20 5

32 4

21 4

11 4

14 4

22 3

13 2

34 1

42 1

53 1

37 1

26 1

Name: researchExp, dtype: int64

df1 = df.drop(['gmatA','gmatQ','gmatV','specialization','department','program','userProfileLink','topperCgpa','termAndYear','userName','confPubs','journalPubs'],axis=1)

df1.columns

Index(['major', 'researchExp', 'industryExp', 'toeflScore', 'toeflEssay',

'internExp', 'greV', 'greQ', 'greA', 'ugCollege', 'cgpa', 'cgpaScale',

'univName', 'admit'],

dtype='object')

df1.dropna(inplace=True)

df1 = df1[df1["admit"] > 0]

df1

|  | **major** | **researchExp** | **industryExp** | **toeflScore** | **toeflEssay** | **internExp** | **greV** | **greQ** | **greA** | **ugCollege** | **cgpa** | **cgpaScale** | **univName** | **admit** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | Systems and Control | 0 | 18 | 112.0 | 26 | 5.0 | 160.0 | 167.0 | 4.5 | Dharamsinh Desai University | 8.50 | 10.0 | Worcester Polytechnic Institute | 1.0 |
| **2** | (MIS / MSIM / MSIS / MSIT) | 0 | 66 | 94.0 | 21 | 0.0 | 146.0 | 157.0 | 3.0 | IET DAVV | 78.28 | 100.0 | Worcester Polytechnic Institute | 1.0 |
| **5** | MIS | 0 | 0 | 273.0 | 6 | 0.0 | 410.0 | 1010.0 | 600.0 | Thadomal Shahani Engineering College | 52.00 | 100.0 | Worcester Polytechnic Institute | 1.0 |
| **6** | MIS | 0 | 0 | 104.0 | 27 | 0.0 | 150.0 | 161.0 | 4.5 | University of Mumbai | 62.20 | 100.0 | Worcester Polytechnic Institute | 1.0 |
| **7** | MIS-management related courses | 0 | 0 | 95.0 | 22 | 0.0 | 147.0 | 156.0 | 3.0 | MU | 52.00 | 100.0 | Worcester Polytechnic Institute | 1.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **23894** | Electrical Engineering | 0 | 0 | 273.0 | 5 | 0.0 | 570.0 | 800.0 | 4.0 | Delhi College Of Engineeing | 74.90 | 100.0 | University of California Santa Barbara | 1.0 |
| **23895** | Electrical Engineering | 0 | 0 | 113.0 | 29 | 0.0 | 700.0 | 800.0 | 4.5 | PESIT | 80.15 | 100.0 | University of California Santa Barbara | 1.0 |
| **23901** | Electrical Engineering | 0 | 0 | 114.0 | 30 | 0.0 | 157.0 | 166.0 | 4.0 | PSG College of Technology | 9.70 | 10.0 | University of California Santa Barbara | 1.0 |
| **23903** | Electrical & Computer Engineering | 0 | 0 | 113.0 | 30 | 0.0 | 164.0 | 170.0 | 4.0 | NIT Warangal | 8.64 | 10.0 | University of California Santa Barbara | 1.0 |
| **23904** | Electrical Engineering | 0 | 0 | 115.0 | 28 | 0.0 | 610.0 | 770.0 | 4.0 | SSN College of Engineering | 86.00 | 100.0 | University of California Santa Barbara | 1.0 |

2560 rows × 14 columns

df1=df1.drop('admit',1)

<ipython-input-35-8ca25ade8018>:1: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'labels' will be keyword-only

df1=df1.drop('admit',1)

university\_list = list(set(df1["univName"].tolist()))

**for** i **in** range(len(university\_list)):

**if** len(df1[df1["univName"] == university\_list[i]]) < 200:

df1 = df1[df1["univName"] != university\_list[i]]

df1

|  | **major** | **researchExp** | **industryExp** | **toeflScore** | **toeflEssay** | **internExp** | **greV** | **greQ** | **greA** | **ugCollege** | **cgpa** | **cgpaScale** | **univName** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **3303** | Computer Science | 0 | 0 | 90.0 | 22 | 0.0 | 152.0 | 160.0 | 3.0 | GJUS&T; Hisar Haryana(Campus) | 77.50 | 100.0 | University of Texas Dallas |
| **3304** | MIS | 0 | 0 | 107.0 | 30 | 0.0 | 30.0 | 48.0 | 5.5 | Pune University | 59.00 | 100.0 | University of Texas Dallas |
| **3305** | Electrical Engineering | 0 | 0 | 99.0 | 24 | 0.0 | 145.0 | 159.0 | 3.0 | JNTU | 84.70 | 100.0 | University of Texas Dallas |
| **3308** | Computer Science | 0 | 0 | 108.0 | 27 | 0.0 | 154.0 | 160.0 | 3.5 | Sri Bhagawan Mahaveer Jain College of Engineering | 73.00 | 100.0 | University of Texas Dallas |
| **3313** | Electrical Engineering | 0 | 0 | 104.0 | 27 | 0.0 | 390.0 | 800.0 | 4.0 | MU | 75.00 | 100.0 | University of Texas Dallas |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **20528** | Computer Engineering | 0 | 53 | 104.0 | 29 | 0.0 | 155.0 | 161.0 | 3.5 | Sardar Patel University | 7.47 | 10.0 | University of Florida |
| **20529** | Computer Engineering | 0 | 0 | 273.0 | 6 | 0.0 | 680.0 | 800.0 | 4.5 | PICT | 60.00 | 100.0 | University of Florida |
| **20531** | Information Science | 0 | 0 | 116.0 | 29 | 0.0 | 570.0 | 770.0 | 3.5 | Cummins College of Engineering Pune | 64.42 | 100.0 | University of Florida |
| **20533** | Civil Engineering | 0 | 0 | 93.0 | 25 | 0.0 | 145.0 | 158.0 | 3.5 | BVM Engineering College | 8.34 | 10.0 | University of Florida |
| **20534** | MIS | 0 | 0 | 106.0 | 25 | 0.0 | 38.0 | 50.0 | 4.0 | Shanxi University | 83.00 | 100.0 | University of Florida |

1264 rows × 13 columns

Utility function to normalize CGPA

**def** normalize\_gpa(df1, cgpa, totalcgpa):

cgpa = df1[cgpa].tolist()

totalcgpa = df1[totalcgpa].tolist()

**for** i **in** range(len(cgpa)):

**if** totalcgpa[i] != 0:

cgpa[i] = cgpa[i] / totalcgpa[i]

**else**:

cgpa[i] = 0

df1["cgpa"] = cgpa

**return** df1

Utility function to preprocess categorical features

**from** collections **import** defaultdict

**def** feature\_extraction\_categorical\_variable1(df1, feature):

feature\_list = list(df1[feature].astype(str))

student\_id\_for\_feature = defaultdict(list)

**for** i **in** range(len(feature\_list)):

feature\_list[i] = str(feature\_list[i])

feature\_list[i] = feature\_list[i].strip()

feature\_list[i] = feature\_list[i].replace("-", "")

feature\_list[i] = feature\_list[i].replace(".", "")

feature\_list[i] = feature\_list[i].partition("/")[0]

feature\_list[i] = feature\_list[i].partition("(")[0]

feature\_list[i] = feature\_list[i].replace(" ", "")

feature\_list[i] = feature\_list[i].lower()

df1[feature] = feature\_list

**return** df1

df1 = df1[df1['greV']<=170]

df1 = df1[df1['greQ']<=170]

df1['cgpaScale'].value\_counts()

100.0 430

10.0 412

4.0 13

0.0 8

5.0 1

Name: cgpaScale, dtype: int64

gpa = df['cgpa'].tolist()

gpa\_scale = df['cgpaScale'].tolist()

**for** i **in** range(len(gpa)):

**if** gpa\_scale[i]==10:

gpa[i] = gpa[i]\*0.4

**elif** gpa\_scale[i]==100:

gpa[i] = (gpa[i]/9.5)\*0.4

**elif** gpa\_scale[i]==5:

gpa[i] = gpa[i]\*0.8

**elif** gpa\_scale[i]==0:

gpa[i] = 0

**else**:

gpa[i] = gpa[i]

df['cgpa\_4'] = gpa

*# For Analytical Writing max. score is 6.0*

df1 = df1[df1['greA']<=6.0]

*# Max. score in TOEFL is 120*

*#df1 = df1[df1['toeflScore']<=120]*

df1 = df1[df1['toeflScore']>80]

df1=df1.drop('major',axis=1)

df1 = df1.drop('ugCollege',axis=1);

df1.hist()

array([[<matplotlib.axes.\_subplots.AxesSubplot object at 0x7f9f1dbd3e80>,

<matplotlib.axes.\_subplots.AxesSubplot object at 0x7f9f1d9f3250>,

<matplotlib.axes.\_subplots.AxesSubplot object at 0x7f9f1da1a2b0>],

[<matplotlib.axes.\_subplots.AxesSubplot object at 0x7f9f1d9ce670>,

<matplotlib.axes.\_subplots.AxesSubplot object at 0x7f9f1d973a60>,

<matplotlib.axes.\_subplots.AxesSubplot object at 0x7f9f1d99bd90>],

[<matplotlib.axes.\_subplots.AxesSubplot object at 0x7f9f1d99be80>,

<matplotlib.axes.\_subplots.AxesSubplot object at 0x7f9f1d954310>,

<matplotlib.axes.\_subplots.AxesSubplot object at 0x7f9f1d8a7a60>]],

dtype=object)

**def** find\_outliers\_IQR(df):

q1=df.quantile(0.25)

q3=df.quantile(0.75)

IQR=q3-q1

outliers = df[((df<(q1-1.5\*IQR)) | (df>(q3+1.5\*IQR)))]

**return** outliers

df2=df1.drop('univName',axis=1);

df2

|  | **researchExp** | **industryExp** | **toeflScore** | **toeflEssay** | **internExp** | **greV** | **greQ** | **greA** | **cgpa** | **cgpaScale** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **3303** | 0 | 0 | 90.0 | 22 | 0.0 | 152.0 | 160.0 | 3.0 | 77.50 | 100.0 |
| **3304** | 0 | 0 | 107.0 | 30 | 0.0 | 30.0 | 48.0 | 5.5 | 59.00 | 100.0 |
| **3305** | 0 | 0 | 99.0 | 24 | 0.0 | 145.0 | 159.0 | 3.0 | 84.70 | 100.0 |
| **3308** | 0 | 0 | 108.0 | 27 | 0.0 | 154.0 | 160.0 | 3.5 | 73.00 | 100.0 |
| **3315** | 0 | 0 | 105.0 | 29 | 0.0 | 154.0 | 162.0 | 4.0 | 8.10 | 10.0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **20513** | 0 | 0 | 116.0 | 29 | 0.0 | 161.0 | 168.0 | 5.0 | 8.48 | 10.0 |
| **20522** | 0 | 0 | 113.0 | 30 | 0.0 | 164.0 | 170.0 | 4.0 | 8.64 | 10.0 |
| **20528** | 0 | 53 | 104.0 | 29 | 0.0 | 155.0 | 161.0 | 3.5 | 7.47 | 10.0 |
| **20533** | 0 | 0 | 93.0 | 25 | 0.0 | 145.0 | 158.0 | 3.5 | 8.34 | 10.0 |
| **20534** | 0 | 0 | 106.0 | 25 | 0.0 | 38.0 | 50.0 | 4.0 | 83.00 | 100.0 |

861 rows × 10 columns

*'''outliers = find\_outliers\_IQR(df2)*

*outliers'''*

{"type":"string"}

*'''outliers = find\_outliers\_IQR(df1[['greV','greQ','greA','cgpa','cgpaScale']])*

*outliers'''*

{"type":"string"}

df1.dropna()

df1

|  | **researchExp** | **industryExp** | **toeflScore** | **toeflEssay** | **internExp** | **greV** | **greQ** | **greA** | **cgpa** | **cgpaScale** | **univName** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **3303** | 0 | 0 | 90.0 | 22 | 0.0 | 152.0 | 160.0 | 3.0 | 77.50 | 100.0 | University of Texas Dallas |
| **3304** | 0 | 0 | 107.0 | 30 | 0.0 | 30.0 | 48.0 | 5.5 | 59.00 | 100.0 | University of Texas Dallas |
| **3305** | 0 | 0 | 99.0 | 24 | 0.0 | 145.0 | 159.0 | 3.0 | 84.70 | 100.0 | University of Texas Dallas |
| **3308** | 0 | 0 | 108.0 | 27 | 0.0 | 154.0 | 160.0 | 3.5 | 73.00 | 100.0 | University of Texas Dallas |
| **3315** | 0 | 0 | 105.0 | 29 | 0.0 | 154.0 | 162.0 | 4.0 | 8.10 | 10.0 | University of Texas Dallas |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **20513** | 0 | 0 | 116.0 | 29 | 0.0 | 161.0 | 168.0 | 5.0 | 8.48 | 10.0 | University of Florida |
| **20522** | 0 | 0 | 113.0 | 30 | 0.0 | 164.0 | 170.0 | 4.0 | 8.64 | 10.0 | University of Florida |
| **20528** | 0 | 53 | 104.0 | 29 | 0.0 | 155.0 | 161.0 | 3.5 | 7.47 | 10.0 | University of Florida |
| **20533** | 0 | 0 | 93.0 | 25 | 0.0 | 145.0 | 158.0 | 3.5 | 8.34 | 10.0 | University of Florida |
| **20534** | 0 | 0 | 106.0 | 25 | 0.0 | 38.0 | 50.0 | 4.0 | 83.00 | 100.0 | University of Florida |

861 rows × 11 columns

df1.columns

Index(['researchExp', 'industryExp', 'toeflScore', 'toeflEssay', 'internExp',

'greV', 'greQ', 'greA', 'cgpa', 'cgpaScale', 'univName'],

dtype='object')

**4.Feature Engineering - Label Encoding**

*# import the LabelEncoder classes*

**from** sklearn.preprocessing **import** LabelEncoder

*#create instance of label encoder*

lab = LabelEncoder()

*#perform label encoding on 'Heart Disease' column*

df1['univName'] = lab.fit\_transform(df1['univName'])

**5.Feature Selection**

y = df1.univName

*#x = df1.drop('univName',axis=1);*

x=df2

**6.Model Selection and Evaluation**

**from** sklearn.model\_selection **import** train\_test\_split

x\_train, x\_test, y\_train, y\_test= train\_test\_split(x, y, test\_size= 0.20, random\_state=0)

*#feature Scaling*

**from** sklearn.preprocessing **import** StandardScaler

st\_x= StandardScaler()

x\_train= st\_x.fit\_transform(x\_train)

x\_test= st\_x.transform(x\_test)

**from** sklearn.svm **import** SVC *# "Support vector classifier"*

classifier = SVC(kernel='linear', random\_state=0)

classifier.fit(x\_train, y\_train)

SVC(kernel='linear', random\_state=0)

y\_pred= classifier.predict(x\_test)

**from** sklearn.metrics **import** confusion\_matrix,accuracy\_score

cm= confusion\_matrix(y\_test, y\_pred)

cm

array([[ 0, 1, 15, 11],

[ 0, 1, 5, 31],

[ 0, 1, 11, 24],

[ 2, 1, 7, 63]])

ac=accuracy\_score(y\_test,y\_pred)

ac

0.43352601156069365

**7. Create Pickle file**

**import** pickle

filename="SVC"

pickle.dump(classifier,open(filename,'wb'))

loaded\_file=pickle.load(open(filename,'rb'))

*#loaded\_file.predict(x\_test)*