INTELLIGENT ADMISSIONS: THE FUTURE OF UNIVERSITY DECISION MAKING WITH MACHINE LEARNING

Done by:

SWETHA V

SRIDEVI V

SWATHI S

SWETHA K

THE FUTURE OF UNIVERSITY DECISION MAKING WITH MACHINE LEARNING

1.INTRODUCTION

1.1 Overview

University admission is the process by which students are selected to attend a college or university. Theprocess typically involves several steps, including submitting an application, taking entrance exams, andparticipating in interviews or other evaluations.Students are often worried about their chances of admission in University. the university admission process for students can be demanding, but by being well-informed, prepared, and organized, students can increase their chances of being admitted to the university of their choice. The aim of this project is to help students in short listing universities with their profiles. Machine learning algorithms are then used to train a model on this data, which can be used to predict the chances of future applicants being admitted. With this project, students can make more informed decisions about which universities to apply to, and universities can make more efficient use of their resources by focusing on the most promising applicants.The predicted output gives them a fair idea about their admission chances in a particular university. This analysis should also help students who are currently preparing or will be preparing to get a better idea.

1.2 Purpose

Students are often worried about their chances of admission in University. the university admission process for students can be demanding, but by being well-informed, prepared, and organized, students can increase their chances of being admitted to the university of their choice.
The aim of this project is to help students in short listing universities with their profiles. Machine learning

algorithms are then used to train a model on this data, which can be used to predict the chances of future applicants being admitted. With this project, students can make more informed decisions about which universities to apply to, and universities can make more efficient use of their resources by focusing on the most promising applicants.The predicted output gives them a fair idea about their admission chances in a particular university. This analysis should also help students who are currently preparing or will be preparing to get a better idea.

Requirements

The business requirements for a machine learning model to predict chances of student admission in the university. A project aims to predict the chances of a student getting admitted to a particular university based on certain factors The business value of this project is that it will help students make more informed decisions about which universities to apply to, and help university counselors to better advise students on the universities they are most likely to be admitted to the university.

Literature Survey

The University Chances of Admission project is a well-researched topic in the field of education and machine learning. Many studies have been conducted to predict university admission using different machine learning techniques.One study by (Hsu and Chen, 2019) used decision tree, random forest, and logistic regression algorithms to predict the chance of university admission based on students' GPA, test scores, and personal information. The study found that the random forest algorithm performed the best with an accuracy of 85.5%.Another study by (Al-Shammari et al., 2018) used the k-nearest neighbor (KNN) algorithm to predict the chance of university admission based on students' GPA, test scores, and family income. The study found that the KNN algorithm performed well with an accuracy of 81.2%.A study by (Najafabadi et al., 2015) used a neural network to predict the chance of university admission based on students' GPA, test scores, and personal information. The study found that the neural network performed well with an accuracy of 94.3%..Overall, these studies suggest that various machine learning algorithms can be used to predict the chance of university admission with high accuracy

Impact

Social Impact:- The ability to accurately predict the chances of university admission can help students make more informed decisions about which universities to apply to, increasing their chances of being admitted and ultimately gaining access to higher education.
Business Model/Impact:- 1. using machine learning models to predict university admission, the service can help universities more efficiently process and evaluate applications, potentially increasing the number of successful admissions.
2.An increase in the number of successful admissions can lead to an increase in revenue for universities, as well as for the company providing the prediction service.

2. Problem Definition & Design Thinking

Collect the dataset

In this project we have used .csv data. This data is downloaded from kaggle.com. Please refer to the link given below to download the dataset.

Link : https://www.kaggle.com/rishal005/admission-predict

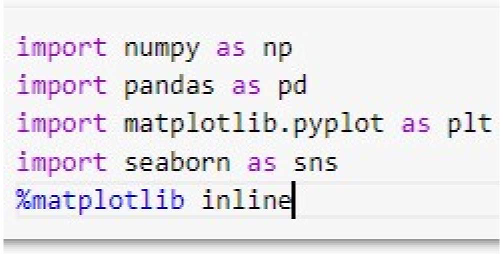
There are a number of techniques for understanding the data. But here we have used some of it. In an additional way, you can use multiple techniques.

Importing the libraries

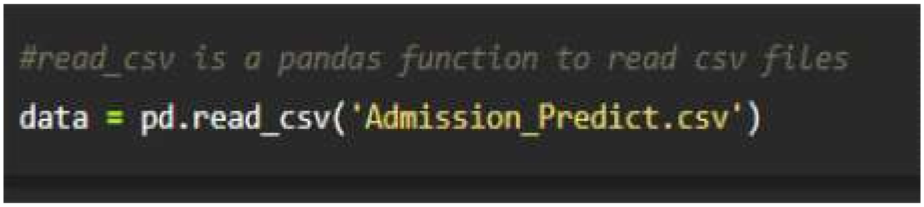
Import the necessary libraries as shown in the image. (optional) Here we have used visualisation style as fivethirtyeight.

Read the Dataset

Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas.
In pandas we have a function called read_csv() to read the dataset. As a parameter we have to give the directory of the csv file



```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```



```
#read_csv is a pandas function to read csv files
data = pd.read_csv('Admission_Predict.csv')
```

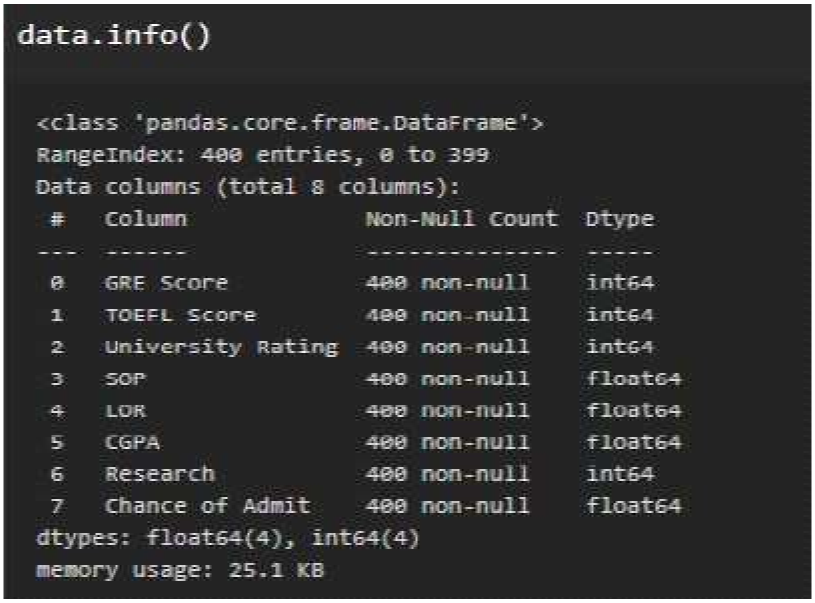
Data Preparation

The download data set is not suitable for training the machine learning model as it might have so much randomness so we need to clean the dataset properly in order to fetch good results. This activity includes the following steps.

- Handling missing values
- Handling categorical data
- Handling Imbalance Data

Handling missing values

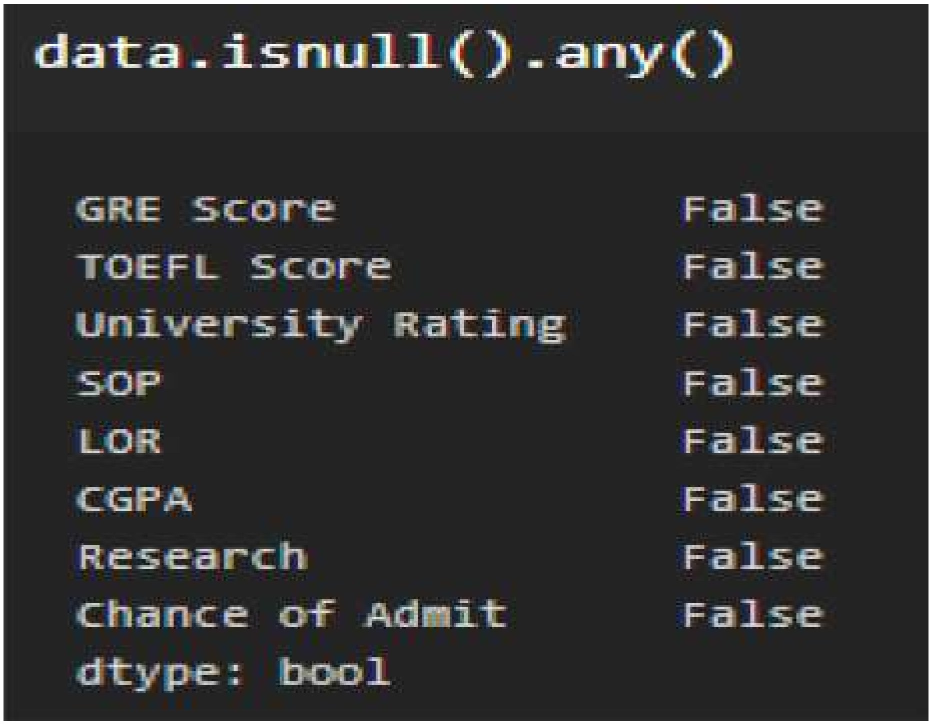
Let's find the shape of our dataset first. To find the shape of our data, the df.shape () method is used. To find the data type, df.info() function is used.



```
data.info()
```

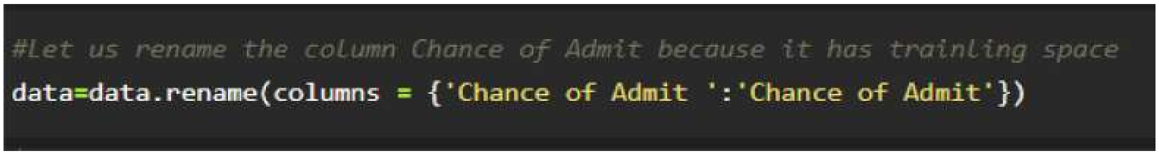
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   GRE Score          400 non-null    int64
 1   TOEFL Score        400 non-null    int64
 2   University Rating  400 non-null    int64
 3   SOP                400 non-null    float64
 4   LOR                400 non-null    float64
 5   CGPA               400 non-null    float64
 6   Research           400 non-null    int64
 7   Chance of Admit    400 non-null    float64
dtypes: float64(4), int64(4)
memory usage: 25.1 KB
```

For checking the null values, df.isnull() function is used. To sum those null values we use .sum() function. From the below image we found that there are no null values present in our dataset.



```
data.isnull().any()

GRE Score            False
TOEFL Score          False
University Rating    False
SOP                  False
LOR                  False
CGPA                 False
Research             False
Chance of Admit      False
dtype: bool
```

Let us rename the column, in python have a inbuilt function rename( ).
We can easily rename the column names.

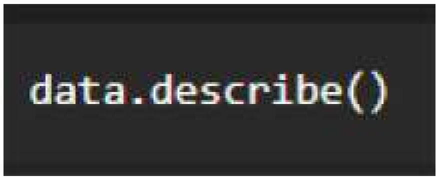


```
#Let us rename the column Chance of Admit because it has trainling space
data=data.rename(columns = {'Chance of Admit ':'Chance of Admit'})
```

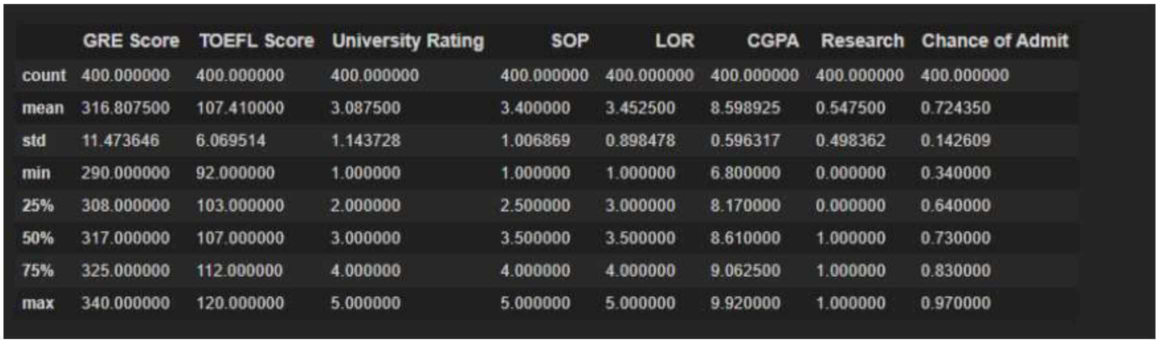
Exploratory Data Analysis

Descriptive statistical

Descriptive analysis is to study the basic features of data with the statistical process. Here pandas has a worthy function called describe. With this describe function we can understand the unique, top and frequent values of categorical features. And we can find mean, std, min, max and percentile values of continuous features.



```
data.describe()
```



	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
count	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000
mean	316.807500	107.410000	3.087500	3.400000	3.452500	8.598925	0.547500	0.724350
std	11.473646	6.069514	1.143728	1.006869	0.898478	0.596317	0.498362	0.142609
min	290.000000	92.000000	1.000000	1.000000	1.000000	6.800000	0.000000	0.340000
25%	308.000000	103.000000	2.000000	2.500000	3.000000	8.170000	0.000000	0.640000
50%	317.000000	107.000000	3.000000	3.500000	3.500000	8.610000	1.000000	0.730000
75%	325.000000	112.000000	4.000000	4.000000	4.000000	9.062500	1.000000	0.830000
max	340.000000	120.000000	5.000000	5.000000	5.000000	9.920000	1.000000	0.970000

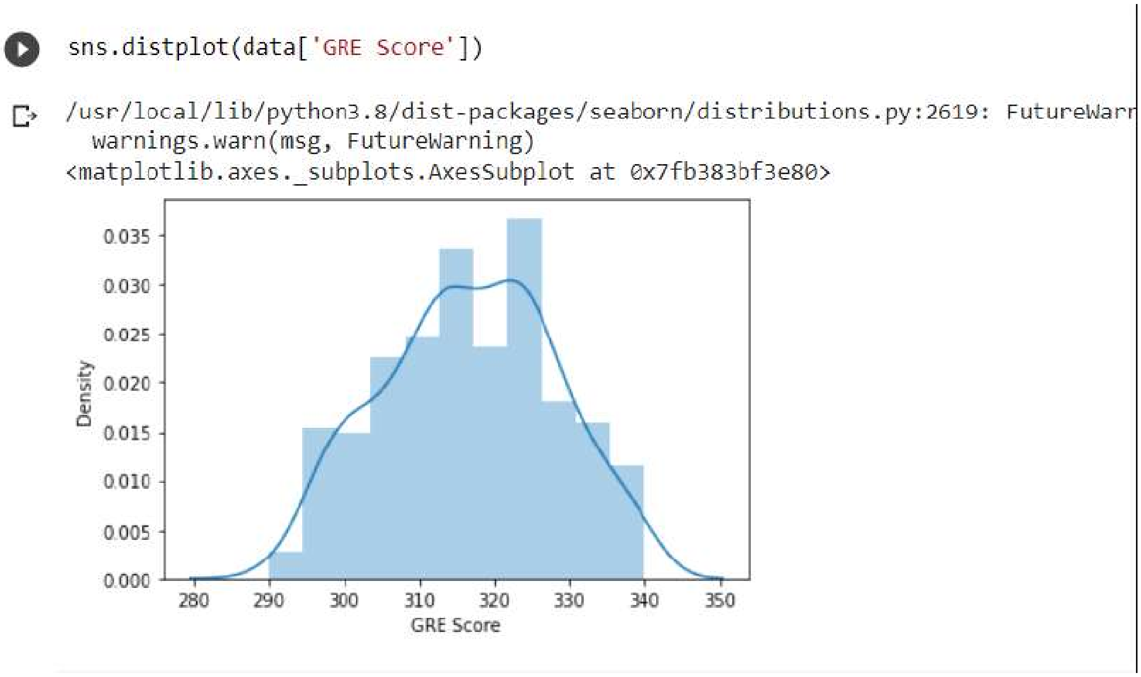
Visual analysis

Visual analysis is the process of using visual representations, such as charts, plots, and graphs, to explore and understand data. It is a way to quickly identify patterns, trends, and outliers in the data, which can help to gain insights and make informed decisions.

Univariate analysis

In simple words, univariate analysis is understanding the data with a single feature. Here we have displayed two different graphs such as distplot and countplot.

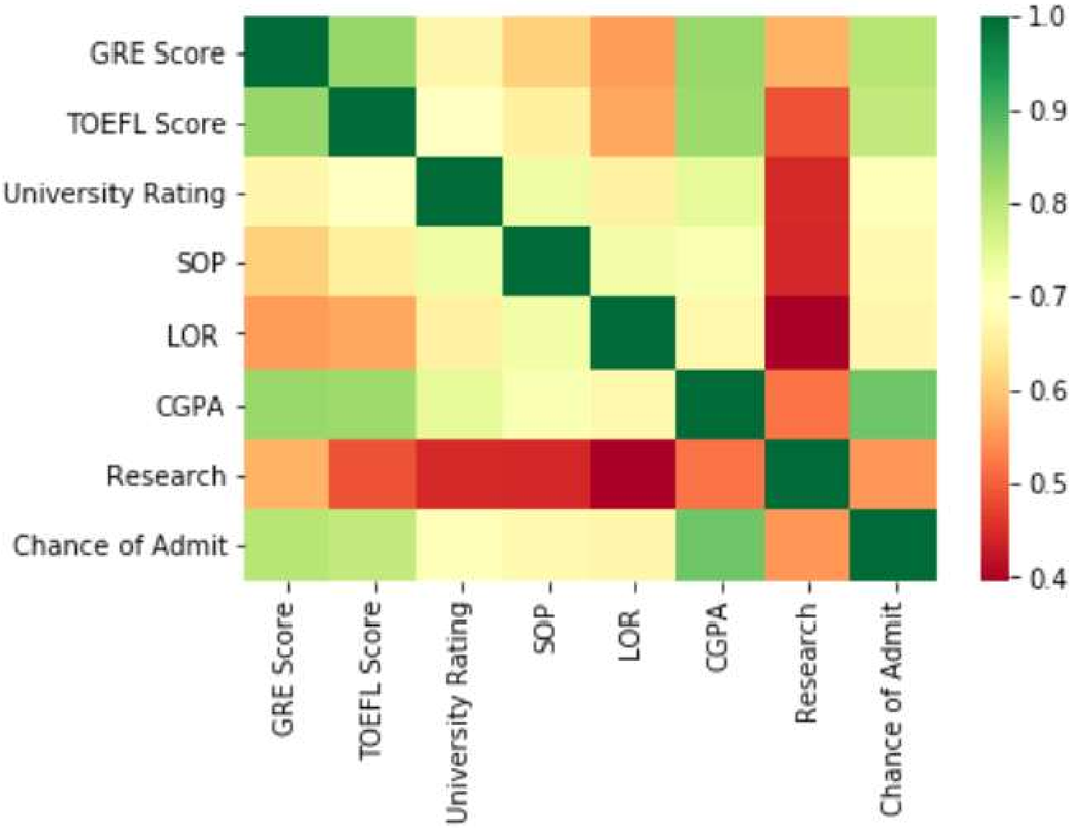
● The Seaborn package provides a wonderful function distplot. With the help of distplot, we can find the distribution of the feature. To make multiple graphs in a single plot, we use subplot.



```
sns.distplot(data['GRE Score'])
```

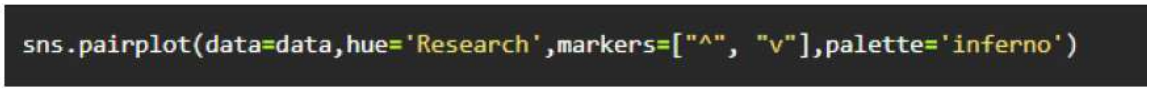
```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarn
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7fb383bf3e80>
```

Bivariate analysis

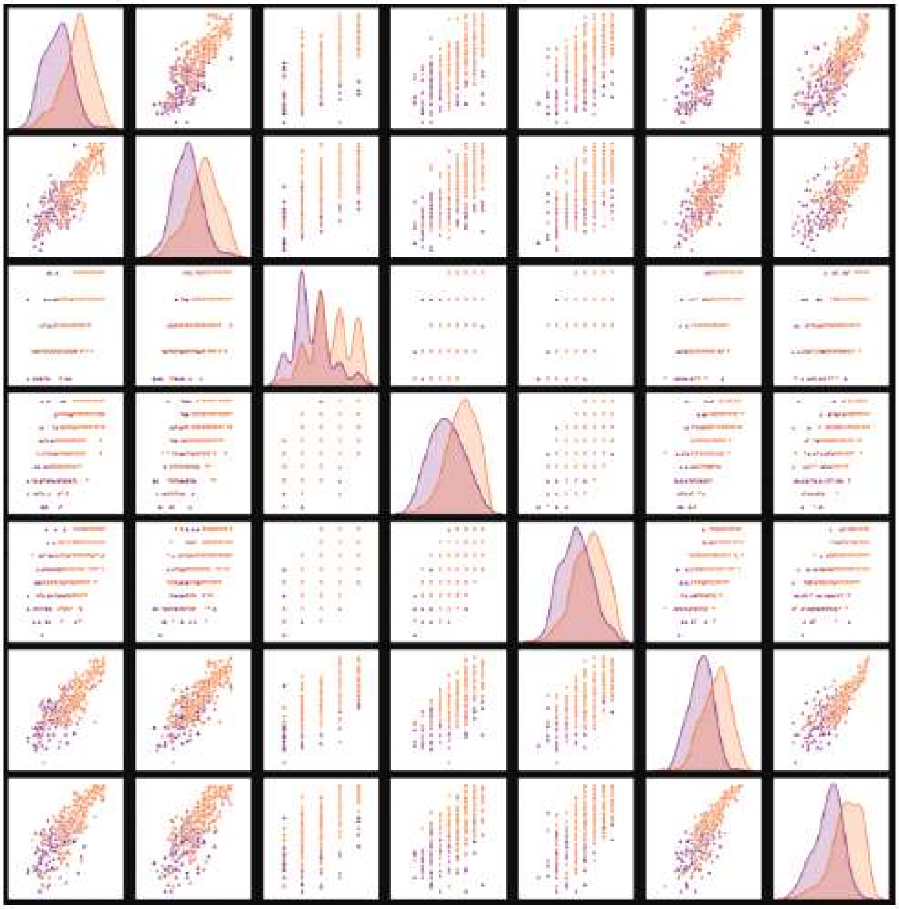


We see that the output variable "Chance of Admit" depends on CGPA, GRE, TOEFEL. The columns SOP, LOR and Research have less impact on university admission

Pair Plot: Plot pairwise relationships in a dataset

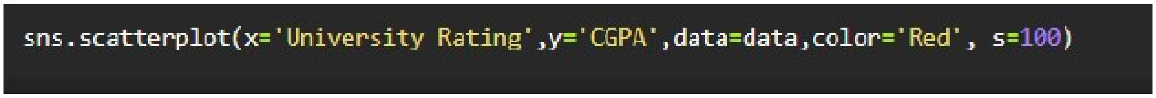


```
sns.pairplot(data=data,hue='Research',markers=["^", "v"],palette='inferno')
```



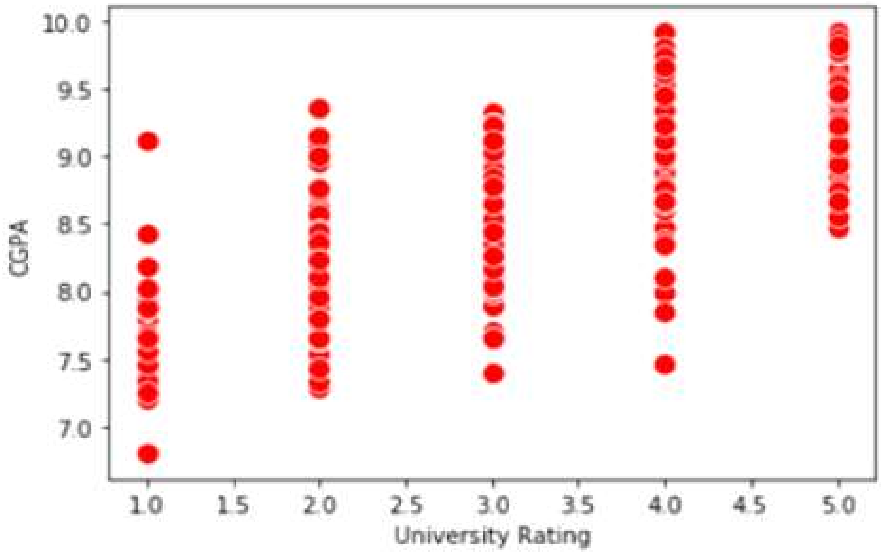
Pair plot usually gives pair wise relationships of the columns in the dataset
1.GRE score TOEFL score and CGPA all are linearly related to each other
2. Students in research score high in TOEFL and GRE compared to non research candidates

Scatter Plot: Matplot has a built-in function to create scatterplots called scatter(). A scatter plot is a type of plot that shows the data as a collection of points

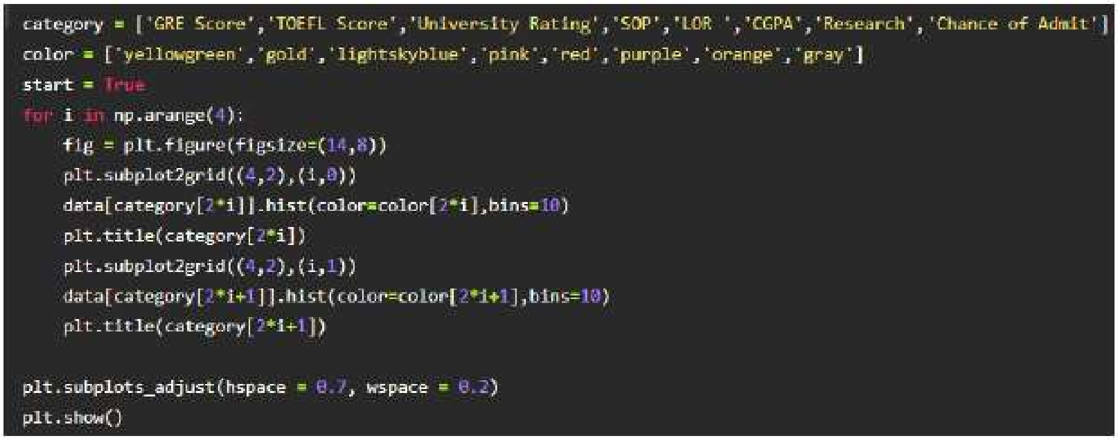


```
sns.scatterplot(x='University Rating',y='CGPA',data=data,color='Red', s=100)
```

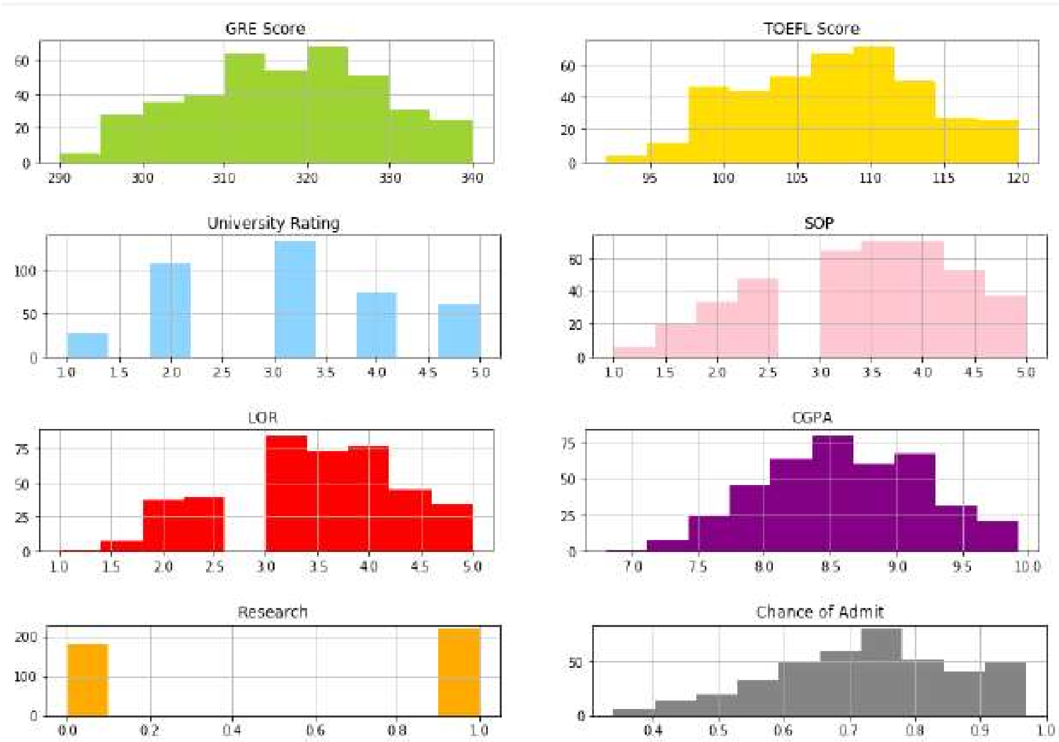
```
<matplotlib.axes._subplots.AxesSubplot at 0x2b6e49feec8>
```



Visualizing the Each column in a dataset using subplot( ).



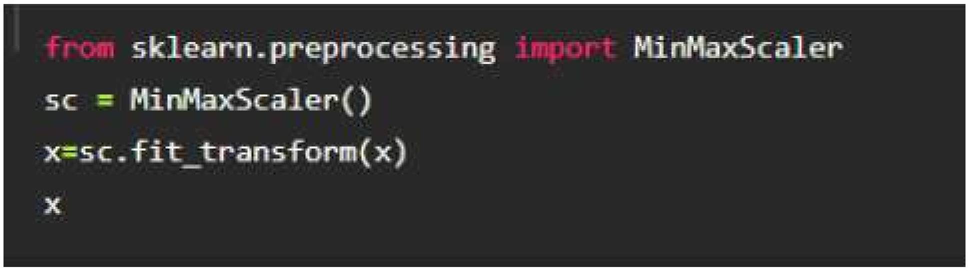
```
category = ['GRE Score','TOEFL Score','University Rating','SOP','LOR ','CGPA','Research','Chance of Admit']
color = ['yellowgreen','gold','lightskyblue','pink','red','purple','orange','gray']
start = True
for i in np.arange(4):
    fig = plt.figure(figsize=(14,8))
    plt.subplot2grid((4,2),(i,0))
    data[category[2*i]].hist(color=color[2*i],bins=10)
    plt.title(category[2*i])
    plt.subplot2grid((4,2),(i,1))
    data[category[2*i+1]].hist(color=color[2*i+1],bins=10)
    plt.title(category[2*i+1])

plt.subplots_adjust(hspace = 0.7, wspace = 0.2)
plt.show()
```



Scaling the Data

Scaling is one the important process, we have to perform on the dataset, because of data measures in different ranges can leads to mislead in prediction.

Models such as KNN, Logistic regression need scaled data, as they follow distance based method and Gradient Descent concept.

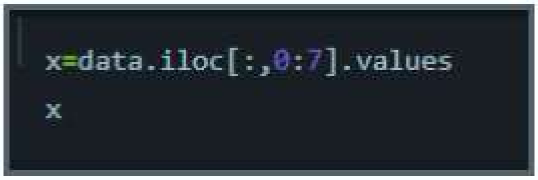


```
from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler()
x=sc.fit_transform(x)
x
```

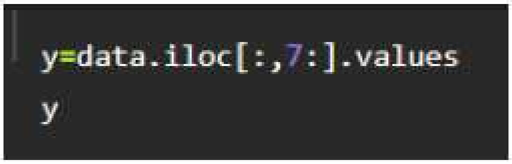
We will perform scaling only on the input values.Once the dataset is scaled, it will be converted into an array and we need to convert it back to a dataframe.

Splitting data into x and y

Now let’s split the Dataset into x and y

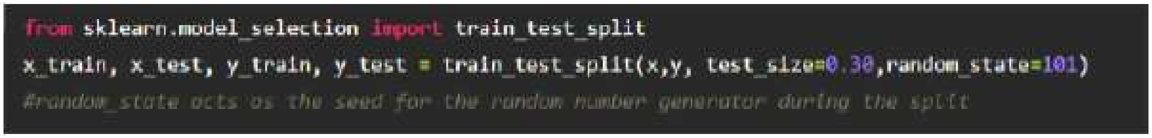


```
x=data.iloc[:,0:7].values
x
```



```
y=data.iloc[:,7:].values
y
```

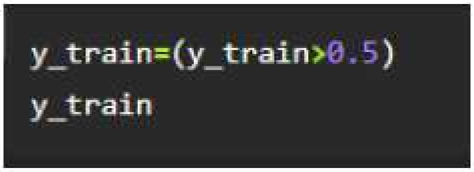
Changes: first split the dataset into x and y and then split the data set
Here x and y variables are created. On x variable, df is passed with dropping the target variable. And on y target variable is passed. For splitting training and testing data we are using the train_test_split() function from sklearn. As parameters, we are passing x, y, test_size, random_state



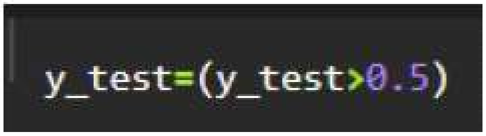
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.30,random_state=101)
#random_state acts as the seed for the random number generator during the split
```

Let us convert it into classification problem

chance of admit>0.5 as true chance of admit<0.5 as false



```
y_train=(y_train>0.5)
y_train
```



```
y_test=(y_test>0.5)
```

Model Building

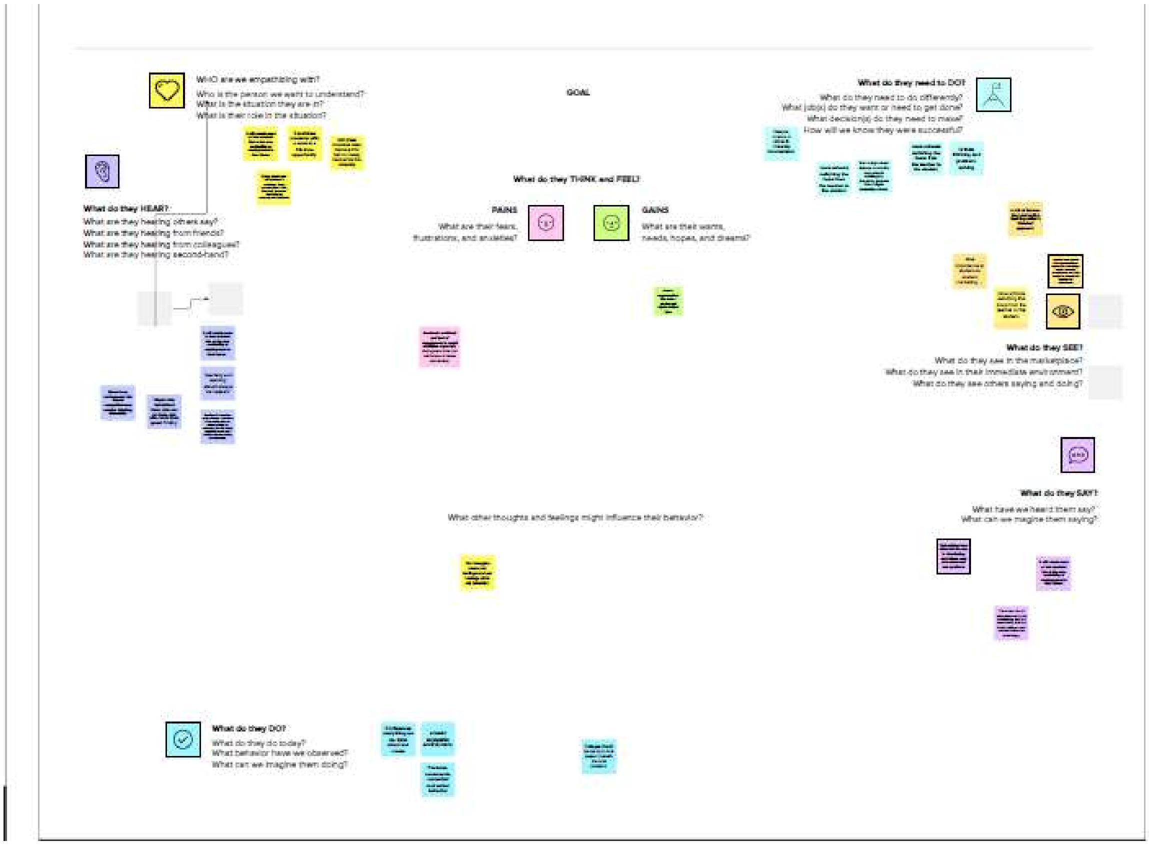
Training the model in multiple algorithms

Now our data is cleaned and it's time to build the model. We can train our data on different algorithms. For this project we are applying four classification algorithms. The best model is saved based on its performance.

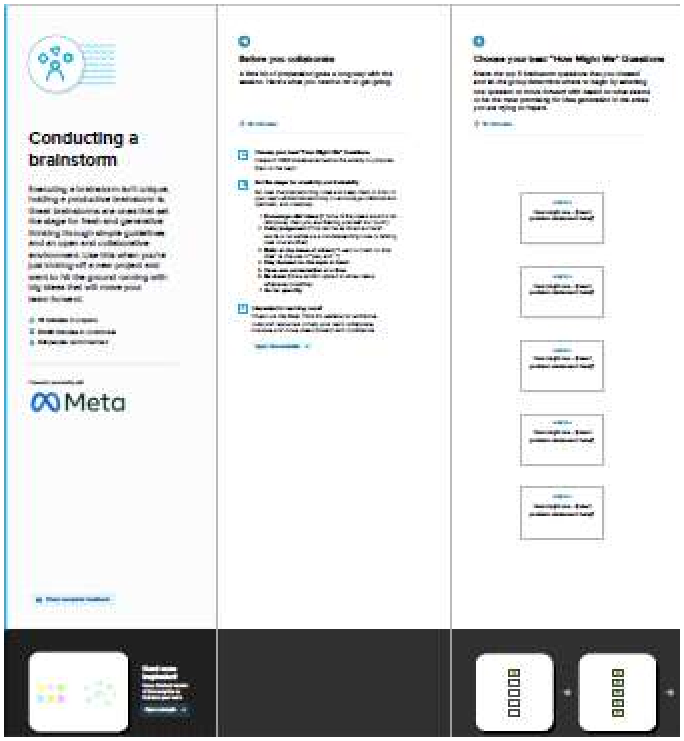
logistic Regression Model

A LogisticRegression algorithm is initialised and training data is passed to the model with the .fit() function. Test data is predicted with .predict() function and saved in a new variable. For evaluating the model, a confusion matrix and classification report is done

2.1 Empathy


WHO are we empathizing with?
Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?
GOAL
What do they need to DO?
What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) do they need to make?
How will we know they were successful?
What do they HEAR?
What are they hearing others say?
What are they hearing from friends?
What are they hearing from colleagues?
What are they hearing second-hand?
What do they THINK and FEEL?
PAINS
What are their fears, frustrations, and anxieties?
GAINS
What are their wants, needs, hopes, and dreams?
What do they SEE?
What do they see in the marketplace?
What do they see in their immediate environment?
What do they see others saying and doing?
What do they SAY?
What have we heard them say?
What can we imagine them saying?
What other thoughts and feelings might influence their behavior?
What do they DO?
What do they do today?
What behavior have we observed?
What can we imagine them doing?

2.2 Ideation & Brainstroming Map

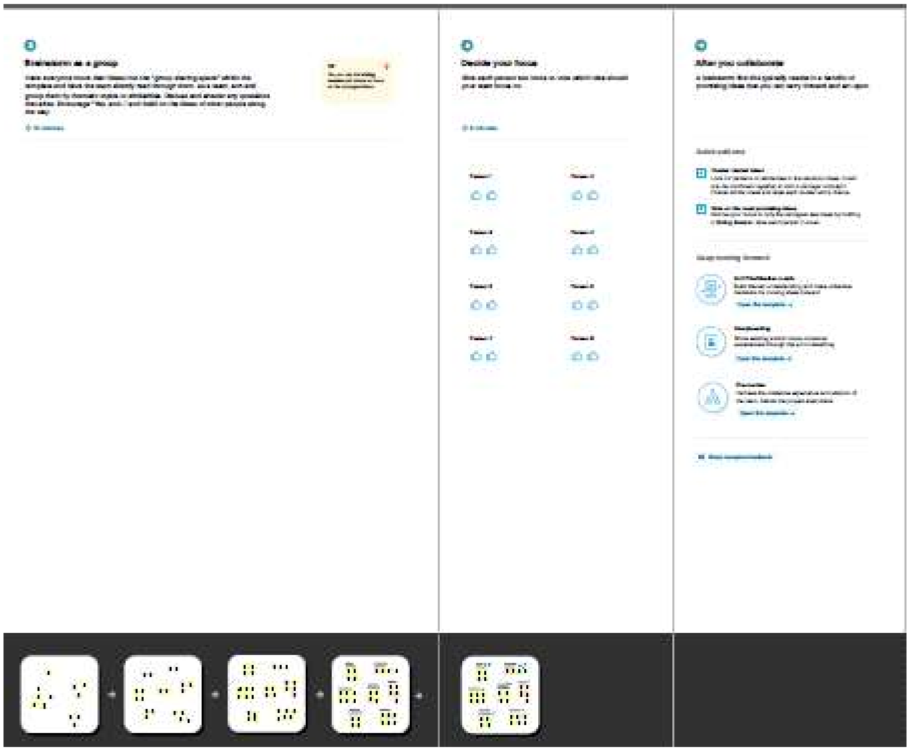


Conducting a brainstorm

Meta

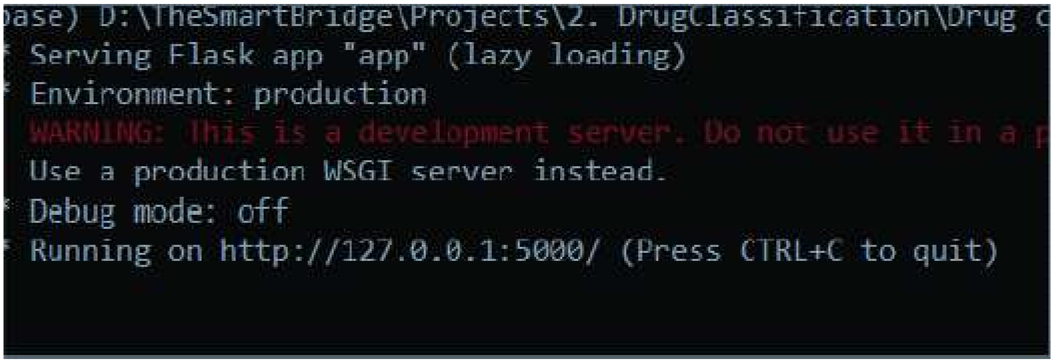
Before you collaborate

Choose your best "How Might We" Questions



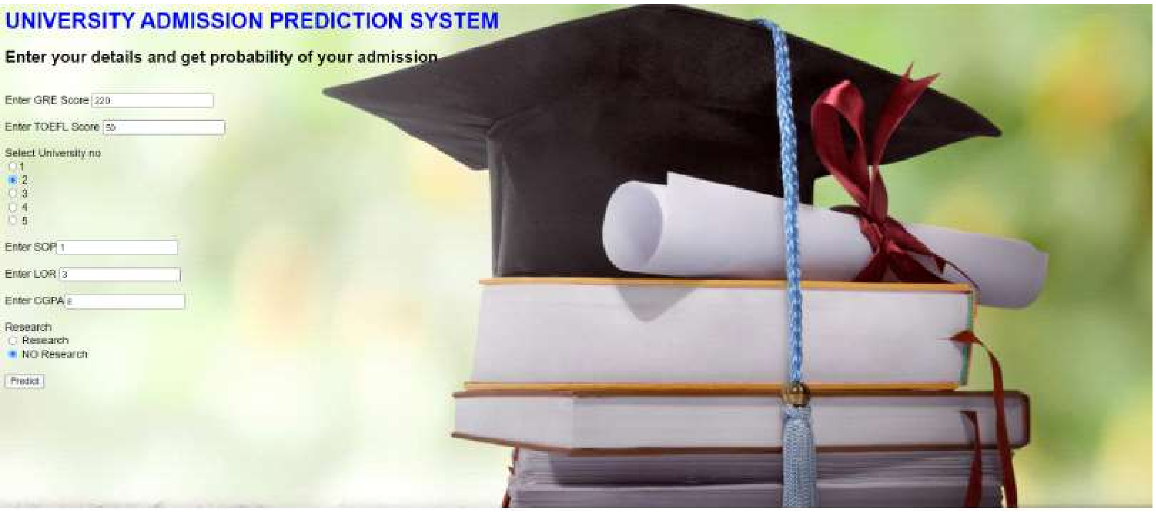
3.RESULT

● Open anaconda prompt from the start menu
● Navigate to the folder where your python script is.
● Now type “python app.py” command
● Navigate to the localhost where you can view your web page.
● Click on the predict button from the top left corner, enter the inputs, click on the submit button, and see the result/prediction on the web.



```
base) D:\TheSmartBridge\Projects\2. DrugClassification\Drug c
 Serving Flask app "app" (lazy loading)
 Environment: production
 WARNING: This is a development server. Do not use it in a p
 Use a production WSGI server instead.
 Debug mode: off
 Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Now,Go the web browser and write the localhost url (http://127.0.0.1:5000) to get the below result



UNIVERSITY ADMISSION PREDICTION SYSTEM

Enter your details and get probability of your admission

Enter GRE Score 220

Enter TOEFL Score 50

Select University no
○ 1
● 2
○ 3
○ 4
○ 5

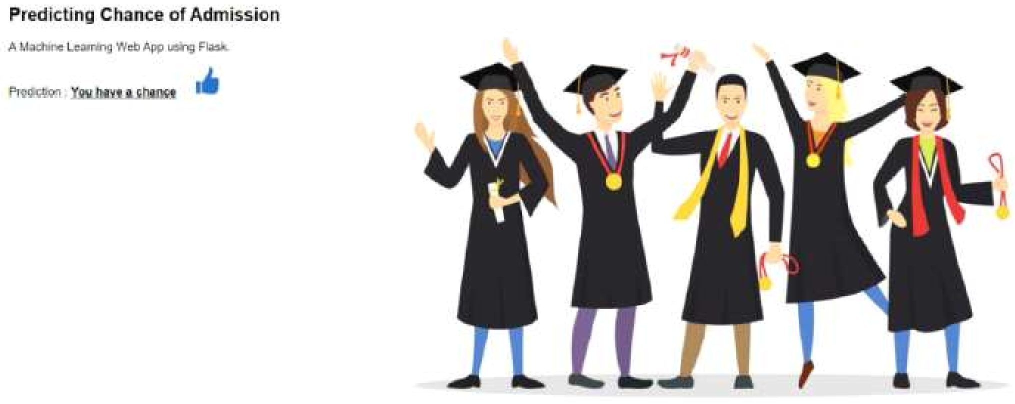
Enter SOP 1

Enter LOR 3

Enter CGPA 8

Research
○ Research
● NO Research

Now,when you click on click me to predict the button from the banner you will get redirected to the prediction page.



Predicting Chance of Admission

A Machine Learning Web App using Flask.

Prediction : You have a chance 👍

Input 1- Now, the user will give inputs to get the predicted result after clicking onto the predict button

4. ADVANTAGES & DISADVANTAGES

Advantages

His plan has the advantage of being less expensive than other options. He lacked the advantages of an advanced education. Speed is an advantage in most sports. The company's only advantage over the competition is its location.

The purpose of having a competitive advantage is to distinguish a company from its competitors by offering something different and of superior value to its customers. Competitive advantage also means the business can outperform its competition in the market and make a higher profit.One of the advantages of myelinating cultures is the ability to visualize axons that are only partially myelinated.

In contrast, a direct implementation benefits from all the advantages of developing in a highlevel programming environment.Permutation-free calculi have the advantages of a sequent calculus system, whilst reflecting the structure of normal natural deductions

verb (used with object), ad·van·taged, ad·van·taging. to be of service to; yield profit or gain to; benefit. to cause to advance; further; promote: Such action will advantage our cause. to prove beneficial to; profit: It would advantage him to work harder.

Disadvantages

Absence or deprivation of advantage or equality. the state or an instance of being in an unfavorable circumstance or condition: to be at a disadvantage. something that puts one in an unfavorable position or condition: His bad temper is a disadvantage

An AMCAS statement of disadvantage is an optional short essay where students can explain any circumstances where they experience a disadvantaged situation or circumstance in applying to medical school.

Childhood disadvantage can affect cognitive functioning, cause neurological impairments, disrupt executive functioning (e.g., one's ability to plan and problem solve), and cause alterations in attention (inattention) and consciousness (Stikkelbroek et al., 2016), all of which are in turn associated with intellectual .

If you are at a disadvantage, you have a problem or difficulty that many other people do not have, which makes it harder for you to be successful. The children from poor families were at a distinct disadvantage.

Socially disadvantaged individuals are those who have been subjected to racial or ethnic prejudice or cultural bias within American society because of their identities as members of groups and without regard to their individual qualities

Disadvantages can help propel you to see your situation from different perspectives and find approaches to succeed that you might not otherwise have found.

5. APPLICATIONS

An application in a software program which enables you to perform a range of useful tasks. Examples of applications are word processing programs, spreadsheet software, databases or graphics packages.

Application areas are deployment-oriented categories that focus on commonly deployed ITS services or systems. Application areas provide a starting point for identifying the ITS standards and other resources (e.g., case studies, lessons learned) that may be relevant to a specific type of deployment.

Application software is usually distinguished into two main classes: closed source vs open source software applications, and free or proprietary software applications.

Software is a set of instructions, data or programs used to operate computers and execute specific tasks. It is the opposite of hardware, which describes the physical aspects of a computer.

Application-level analysis is about analyzing the data transmitted by an application as the application would have interpreted it. This is a resource-intensive type of analysis in several regards.

Application Unit means, in respect of a Listed Class, such number of Units of a Listed Class or whole multiples thereof as specified in this Prospectus or such other number of Units of a Listed Class determined by the Manager, approved by the Trustee and notified to the Participating Dealers.

6. CONCLUCTION

The conclution of this project is to help students in short listing universities with their profiles. Machine learning algorithms are then used to train a model on this data, which can be used to predict the chances of future applicants being admitted. With this project, students canmake more informed decisions about which universities to apply to, and universities can make more efficient use of their resources by focusing on the most promising applicants.The predicted output gives them a fair idea about their admission chances in a particular university. This analysis should also help students who are currently preparing or will be preparing to get a better idea.

7. FUTURE SCOPE

The future scope of Machine Learning will take this digital world to a new era of automation. In this blog, we will see the Machine Learning future scope.

The scope of Artificial Intelligence is limited to domestic and commercial purposes as the medical and aviation sectors are also using AI to improve their services. If AI is outperforming human efforts, then opting for AI automation will reduce costs in the long run for a business.

Various benefits of AI lead to various use cases and job roles in the market, which are beneficial for deep tech enthusiasts or freshers looking to build their careers in the AI industry. The scope of AI is bright in India as firms need expert employees who can extract meaningful information from large chunks of data

Positive Impacts of Artificial Intelligence on Human Society In the workplace, artificial intelligence can boost efficiency and increase humans' capacity to perform certain tasks. AI frees humans to do work they are better equipped for, such as creative and empathic tasks, by taking over repetitive or dangerous tasks.

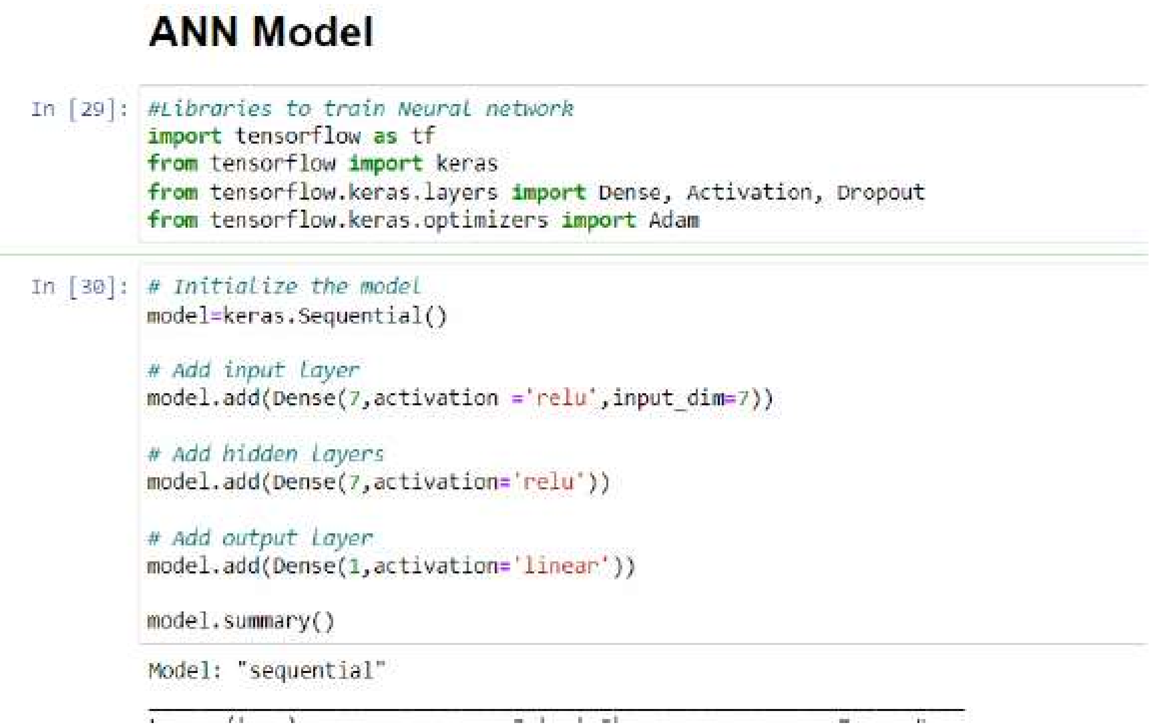
Posted on July 24, 2022 · The scope of a study explains the extent to which the research area will be explored in the work and specifies the parameters within the study will be operating. Basically, this means that you

8. APPENDIX

A.Source code

ANN model

Building and training an Artificial Neural Network (ANN) using the Keras library with TensorFlow as the backend. The ANN is initialised as an instance of the Sequential class, which is a linear stack of layers. Then, the input layer and two hidden layers are added to the model using the Dense class, where the number of units and activation function are specified. The output layer is also added using the Dense class with a sigmoid activation function. The model is then compiled with the Adam optimizer, binary cross-entropy loss function, and accuracy metric. Finally, the model is fit to the training data with a batch size of 100, 20% validation split, and 100



ANN Model

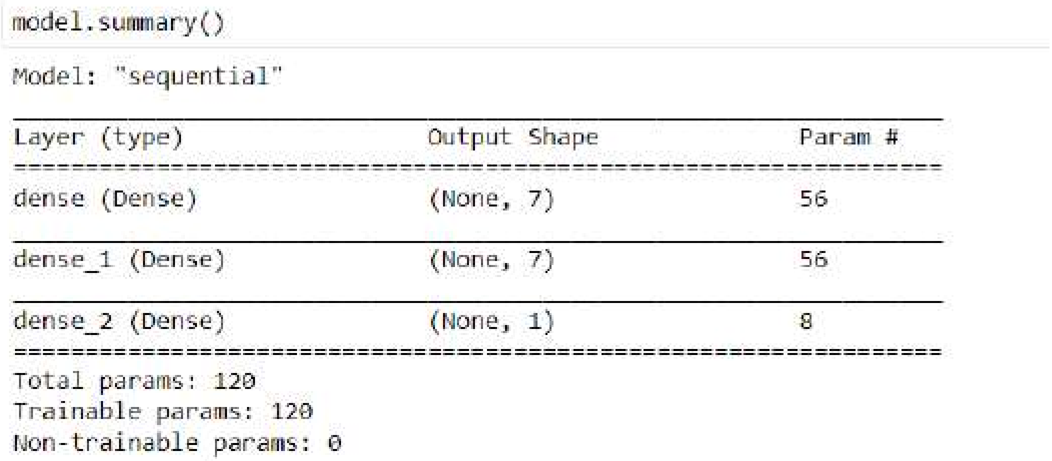
In [29]:

```
#Libraries to train Neural network
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.layers import Dense, Activation, Dropout
from tensorflow.keras.optimizers import Adam
```

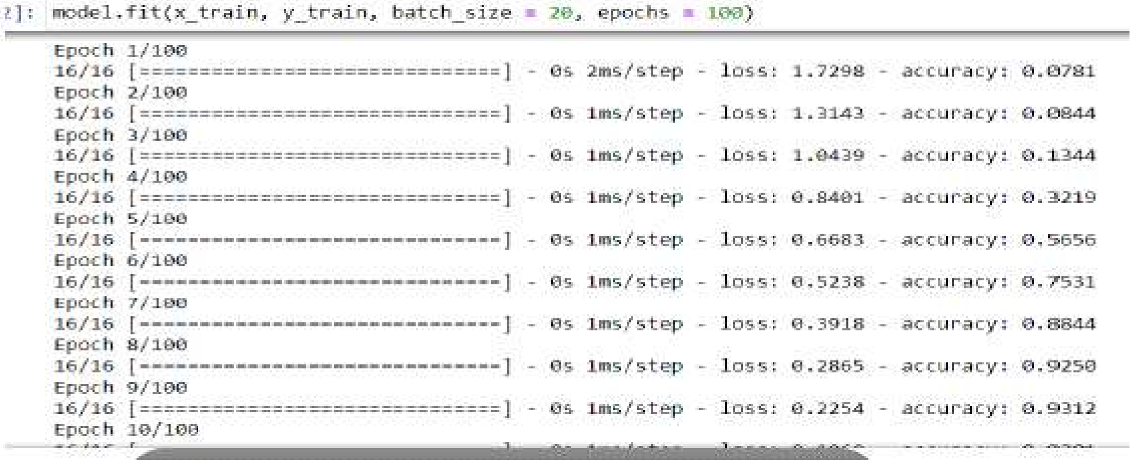
In [30]:

```
# Initialize the model
model=keras.Sequential()

# Add input layer
model.add(Dense(7,activation ='relu',input_dim=7))

# Add hidden layers
model.add(Dense(7,activation='relu'))

# Add output layer
model.add(Dense(1,activation='linear'))

model.summary()
```

```
Model: "sequential"
```



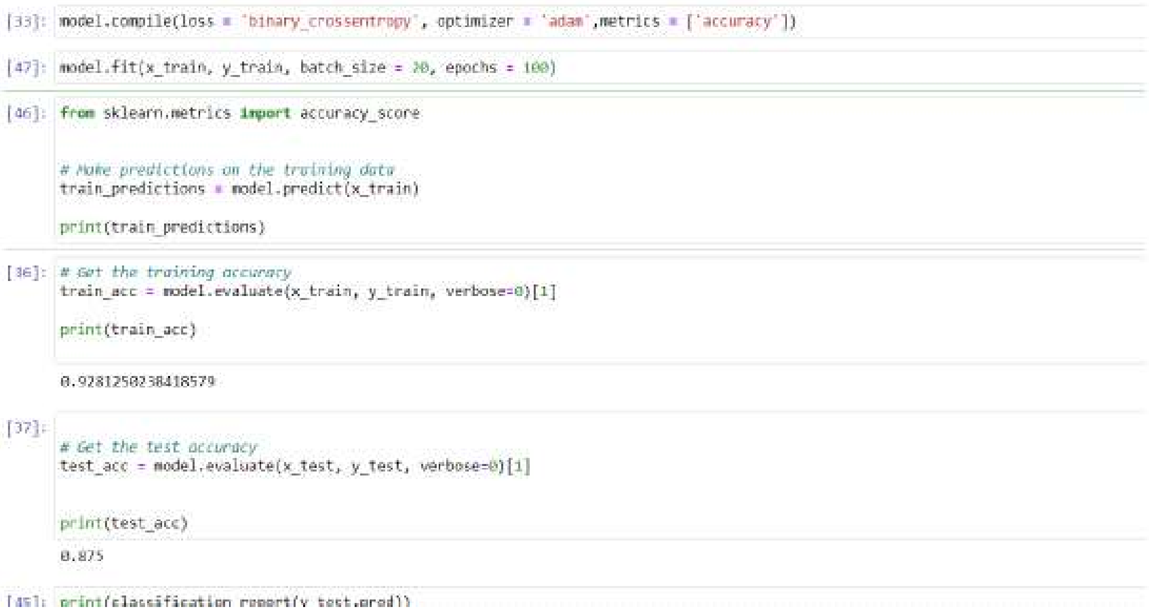
```
model.summary()

Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 7)                 56
_________________________________________________________________
dense_1 (Dense)              (None, 7)                 56
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 8
=================================================================
Total params: 120
Trainable params: 120
Non-trainable params: 0
```



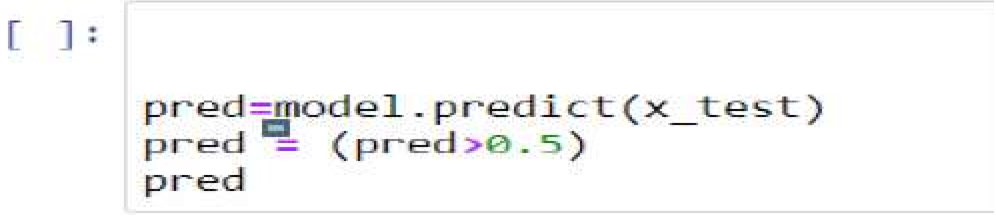
```
model.fit(x_train, y_train, batch_size = 20, epochs = 100)

Epoch 1/100
16/16 [==============================] - 0s 2ms/step - loss: 1.7298 - accuracy: 0.0781
Epoch 2/100
16/16 [==============================] - 0s 1ms/step - loss: 1.3143 - accuracy: 0.0844
Epoch 3/100
16/16 [==============================] - 0s 1ms/step - loss: 1.0439 - accuracy: 0.1344
Epoch 4/100
16/16 [==============================] - 0s 1ms/step - loss: 0.8401 - accuracy: 0.3219
Epoch 5/100
16/16 [==============================] - 0s 1ms/step - loss: 0.6683 - accuracy: 0.5656
Epoch 6/100
16/16 [==============================] - 0s 1ms/step - loss: 0.5238 - accuracy: 0.7531
Epoch 7/100
16/16 [==============================] - 0s 1ms/step - loss: 0.3918 - accuracy: 0.8844
Epoch 8/100
16/16 [==============================] - 0s 1ms/step - loss: 0.2865 - accuracy: 0.9250
Epoch 9/100
16/16 [==============================] - 0s 1ms/step - loss: 0.2254 - accuracy: 0.9312
Epoch 10/100
```

Testing the model

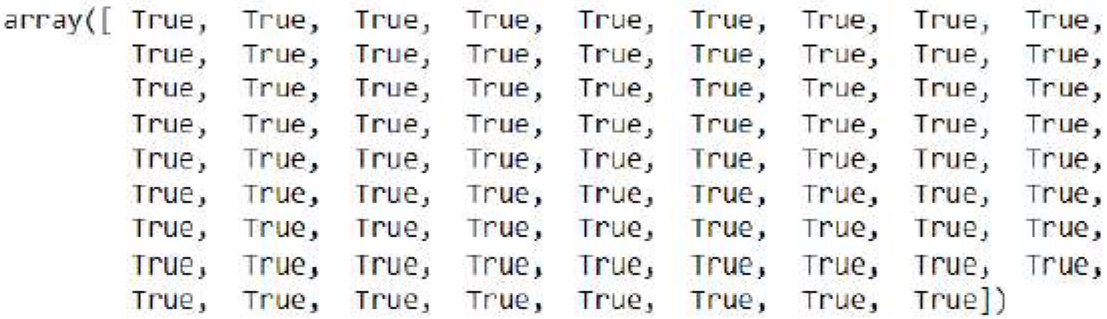
In ANN we first have to save the model to the test the inputs



```
[33]: model.compile(loss = 'binary_crossentropy', optimizer = 'adam',metrics = ['accuracy'])

[47]: model.fit(x_train, y_train, batch_size = 20, epochs = 100)

[46]: from sklearn.metrics import accuracy_score


      # Make predictions on the training data
      train_predictions = model.predict(x_train)

      print(train_predictions)

[36]: # Get the training accuracy
      train_acc = model.evaluate(x_train, y_train, verbose=0)[1]

      print(train_acc)

      0.9281250238418579

[37]:
      # Get the test accuracy
      test_acc = model.evaluate(x_test, y_test, verbose=0)[1]


      print(test_acc)
      0.875

[45]: print(classification_report(y_test,pred))
```



```
[ ]:

      pred=model.predict(x_test)
      pred = (pred>0.5)
      pred
```



```
array([ True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True])
```