|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **img2** | **IMG1** | **img5** | **img3** | **img6** |

**Government of Tamil Nadu**

**Naan Muthalvan - Project-Based Experiential Learning**

## A Review of Liver Patient Analysis Methods Using Machine Learning

Submitted by

Team ID: NM2023TMID22881

**D.Ramya - (20326ER060)**

**J.Ramya - (20326ER061)**

**S.Sandhiya - (20326ER062)**

**S.Sangeetha - (20326ER063)**

**Under the guidance of**

**Mrs. P.SANGEETHA M.SC., M.Phil.,**

**Guest Lecturer**

## PG and Research Department of Computer Science



## M.V.MUTHIAH GOVERNMENT ART SCOLLEGE FOR WOMEN

(Affiliated To Mother Teresa Women’s University, Kodaikanal) Reaccredited with ”A” Grade by NAAC

**DINDIGUL-624001.**

**APRIL-2023**

## M.V.MUTHIAH GOVERNMENT ARTS COLLEG FOR WOMEN

## (Affiliated to Mother Teresa Women’s University, Kodaikanal)

## Reaccredited with ”A” Grade by NAAC

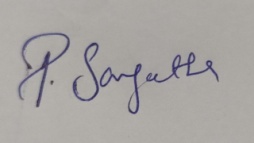
## Dindigul-624001



## PG & RESEARCH DEPARTMENT OF COMPUTER SCIENCE

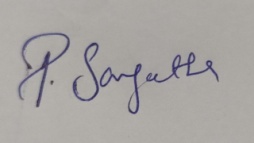
## BONAFIDE CERTIFICATE

This is to certify that this is a bonafide record of the project entitled, done by **Ms.D.RAMYA- (20326ER060), Ms. J.RAMYA (20326ER061)**, **Ms.S.SANDHIYA (20326ER062),** AND **Ms.S.SANGEETHA (20326ER063)**. This is submitted in partial fulfillment for the award of the degree of  **Bachelor of Science in Computer Science in M.V.MUTHIAH GOVERNMENT ARTS COLLEGE FOR WOMEN,DINDIGUL** during the period of December 2022 to April 2023.

** **

**Project Mentor(s) Head of the Department**

**Submitted for viva-voce Examination held on** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

****

**INTERNAL EXAMINER EXTERNAL EXAMINER**

TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **S.No** | **CONTENTS** | **PAGE**  **NO.** |
| 1 | **INTRODUCTION** | 1 |
|  | **1.1Over view** | 1 |
|  | **1.2 Purpose** | 2 |
| 2 | **PROBLEM DEFINITION** | 3 |
|  | **2.1 Empathy Map** | 3 |
|  | **2.2 Ideation & Brainstorming Map** | 4 |
| **3** | **RESULT** | 8 |
| **4** | **ADVANTAGES & DISADVANTAGE** | 10 |
| **5** | **APPLICATION** | 12 |
| **6** | **CONCLUSION** | 13 |
| **7** | **FUTURE SCOPE** | 14 |
| **8** | **APPENDIX** | 15 |
|  | **8.1 Source Code** | 15 |

**1. INTRODUCTION**

### 1.1Overview

Admission predict is a common task in the field of education, where the goal is to predict whether a student will be admitted to a particular college or university based on a set of input features such as their grades, test scores, extracurricular activities, and personal characteristics. This task can be framed as a binary classification problem, where the model must learn to distinguish between students who are likely to be admitted and those who are not.

To solve this task, various machine learning algorithms such as logistic regression, decision trees, and neural networks have been applied. These models are trained on a labeled dataset that contains information about past applicants and their admission status. The trained models are then used to predict the admission status of new applicants based on their input features.

Admission prediction models have many potential applications, such as assisting universities with their admissions process, helping students identify which schools they are most likely to be accepted to, and providing insights into the factors that contribute to admission decisions. However, it is important to note that admission prediction models must be used ethically and responsibly, and should not be used to unfairly discriminate against certain groups of applicants based on factors such as race, gender, or socioeconomic status.

### 1

### 1.2 Purpose

The purpose of admission prediction is to develop models that can accurately predict whether a student will be admitted to a particular college or university based on a set of input features. This task is important because it can help both students and universities make better-informed decisions about the college admissions process.

For students, admission prediction models can help them identify which schools they are most likely to be accepted to based on their academic and personal characteristics. This can help students make more informed decisions about where to apply, and can also help them focus their efforts on schools where they are more likely to be admitted.

For universities, admission prediction models can assist with the admissions process by providing a more efficient and objective way of evaluating applicants. By using data-driven models to predict admission decisions, universities can potentially reduce bias and ensure that all applicants are evaluated fairly and consistently.

Overall, the purpose of admission prediction is to improve the efficiency and fairness of the college admissions process, and to help students and universities make better-informed decisions about which schools to apply to and admit.

**2**

**Top of Form**

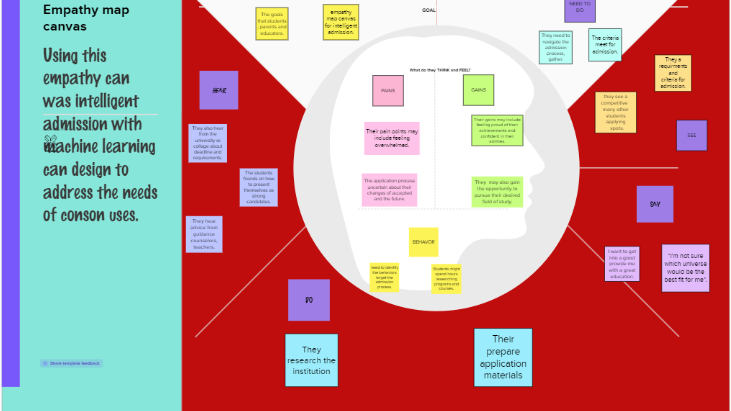
**Bottom of Form**

**Top of Form**

**2. Problem Definition**

### 2.1 Empathy map

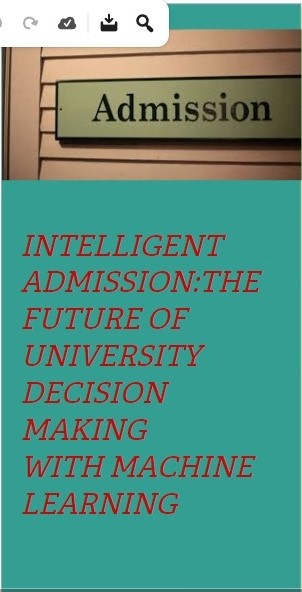
An empathy m& Design Thinking map is a collaborative tool teams can use to gain a deeper insight into their customers.

****

**3**

### 2.2 Ideation & Brainstrom Map

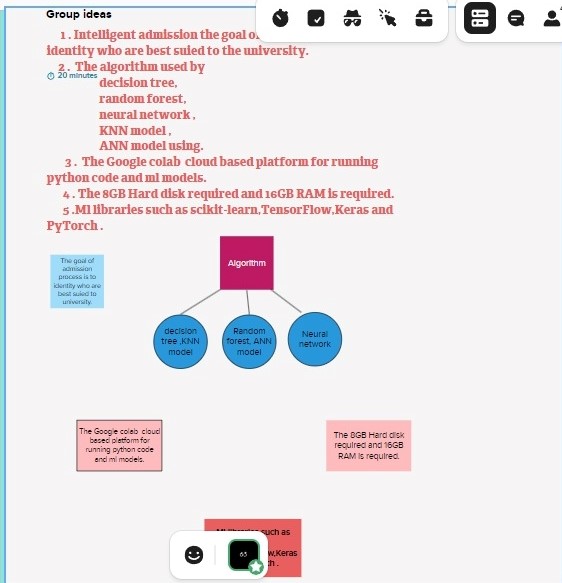
### A mind map is a highly effective tool used by creatives, marketers, and project manager to inspire their teams.

****

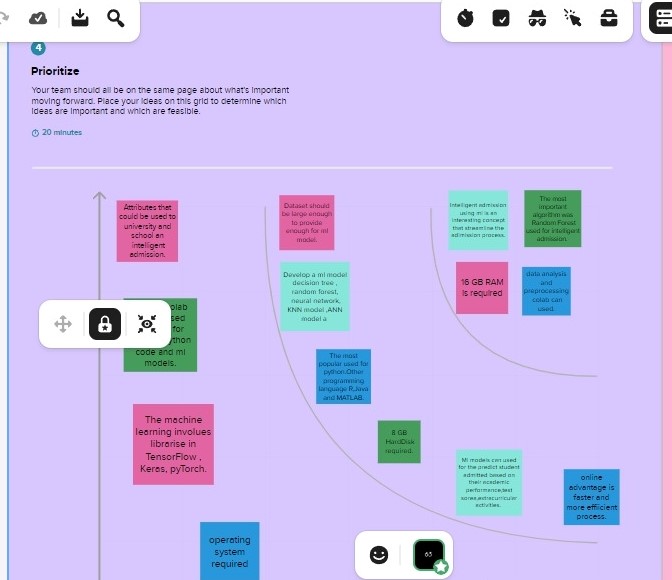
**4**

****

**5**

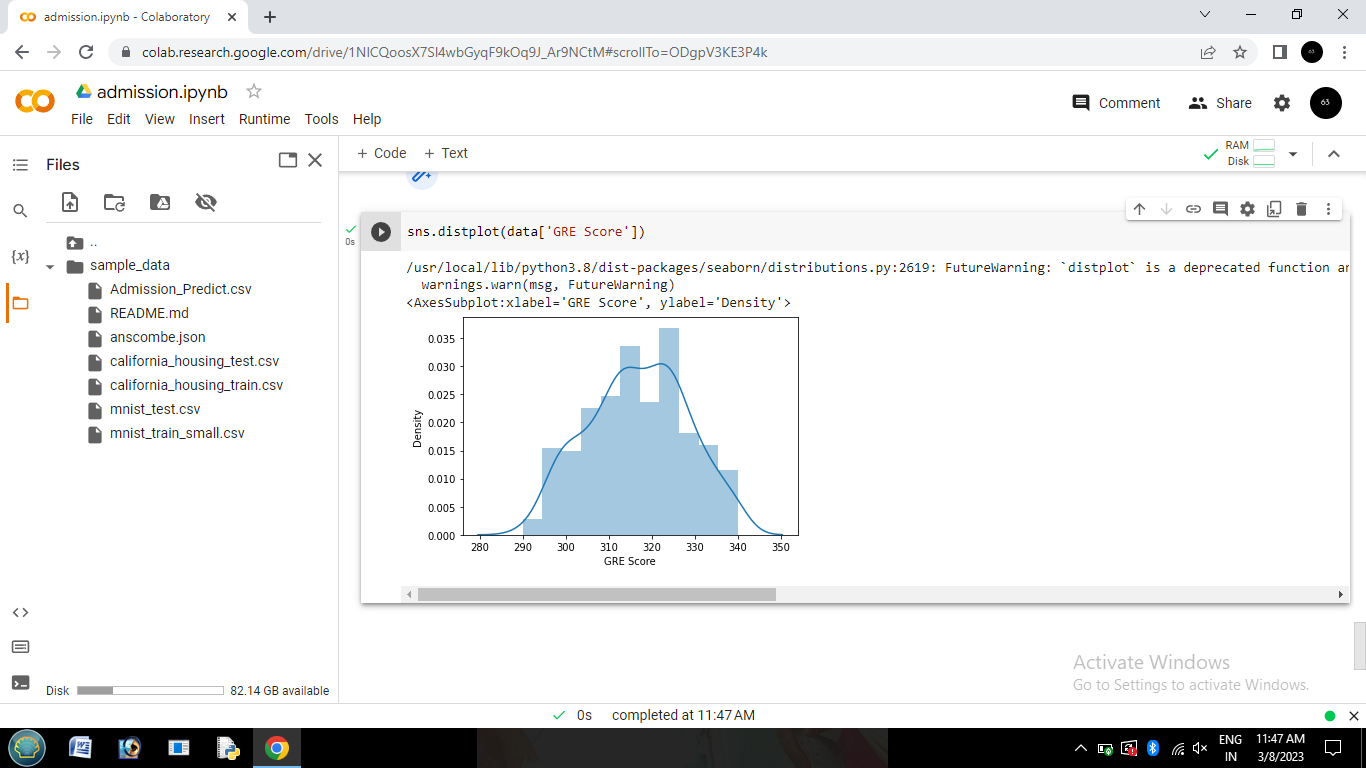
****

**6**

****

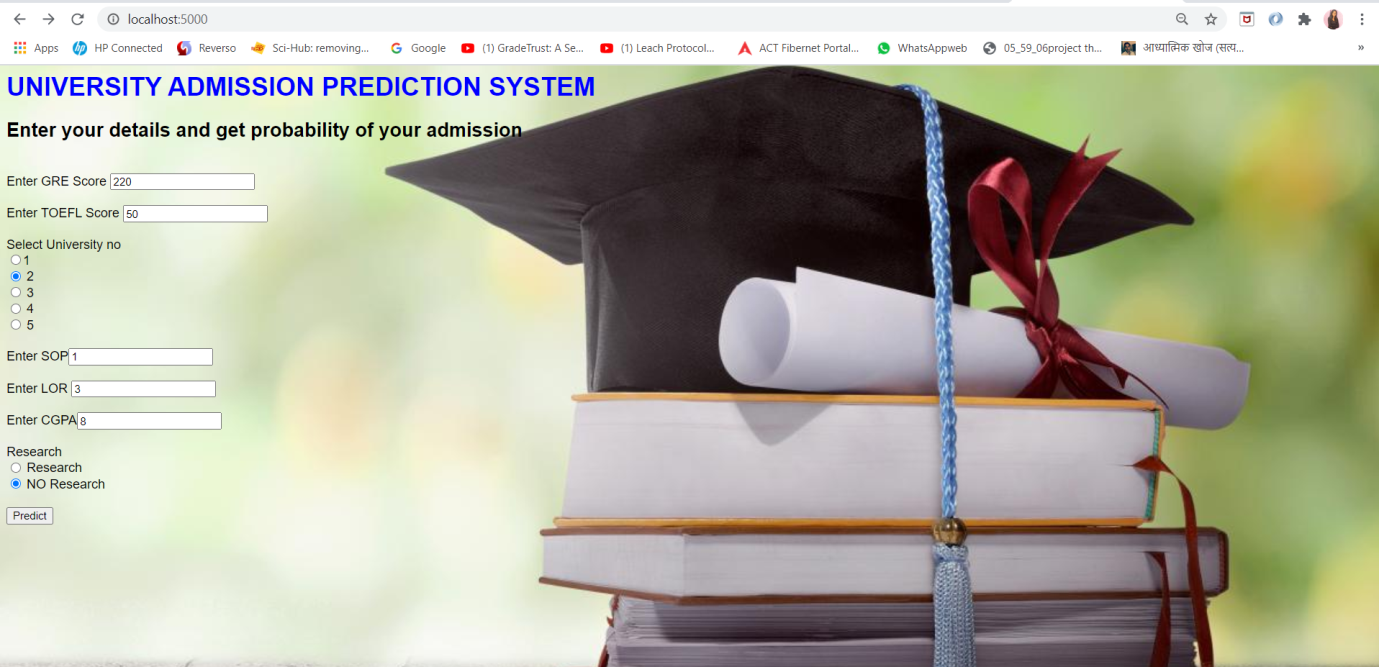
**7**

**3. RESULT**

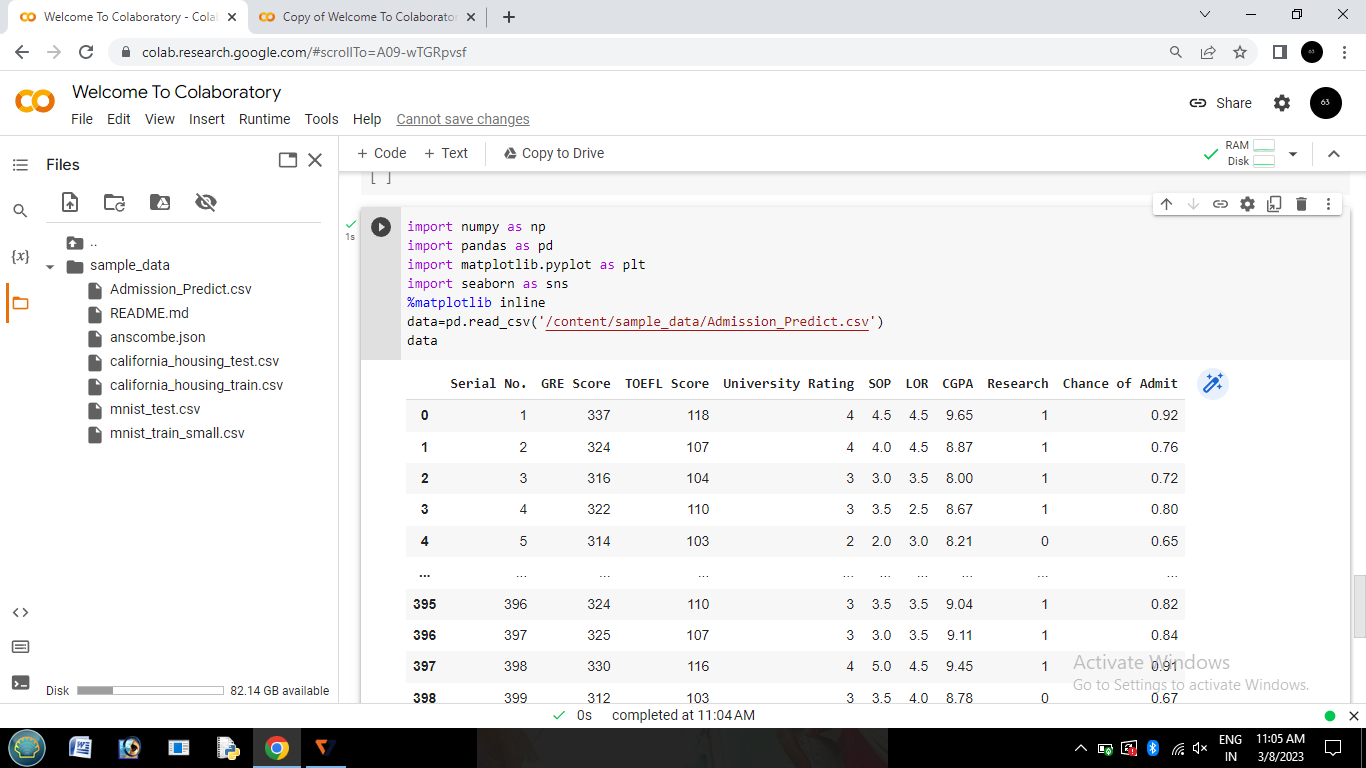
****

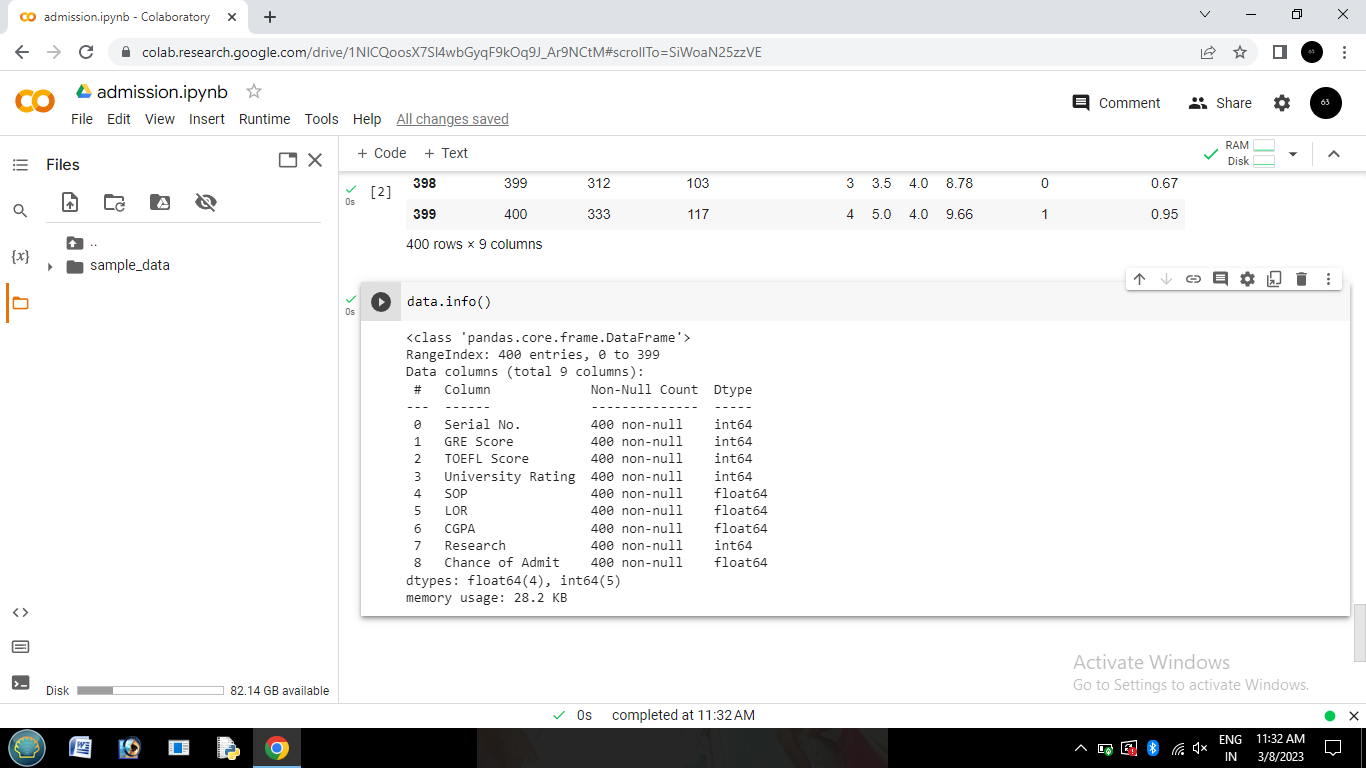
### 

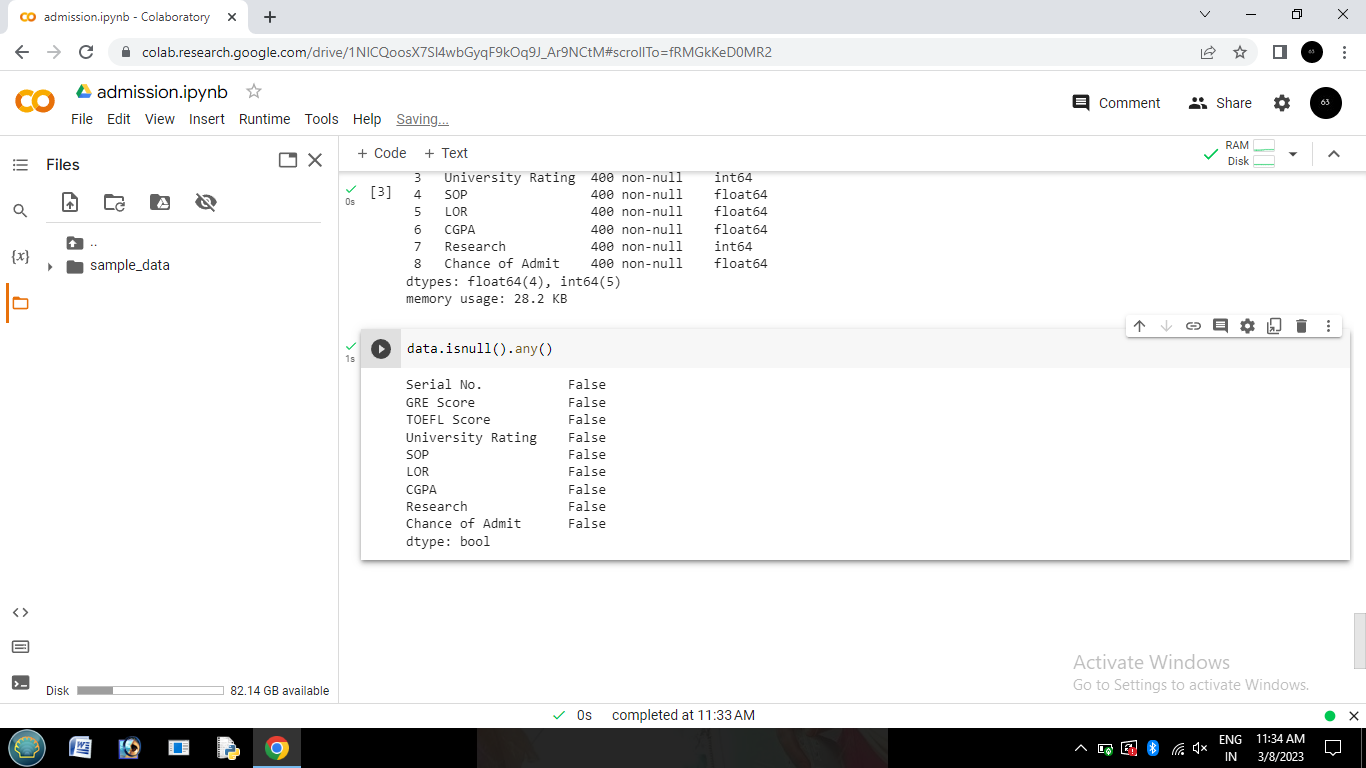
**8**

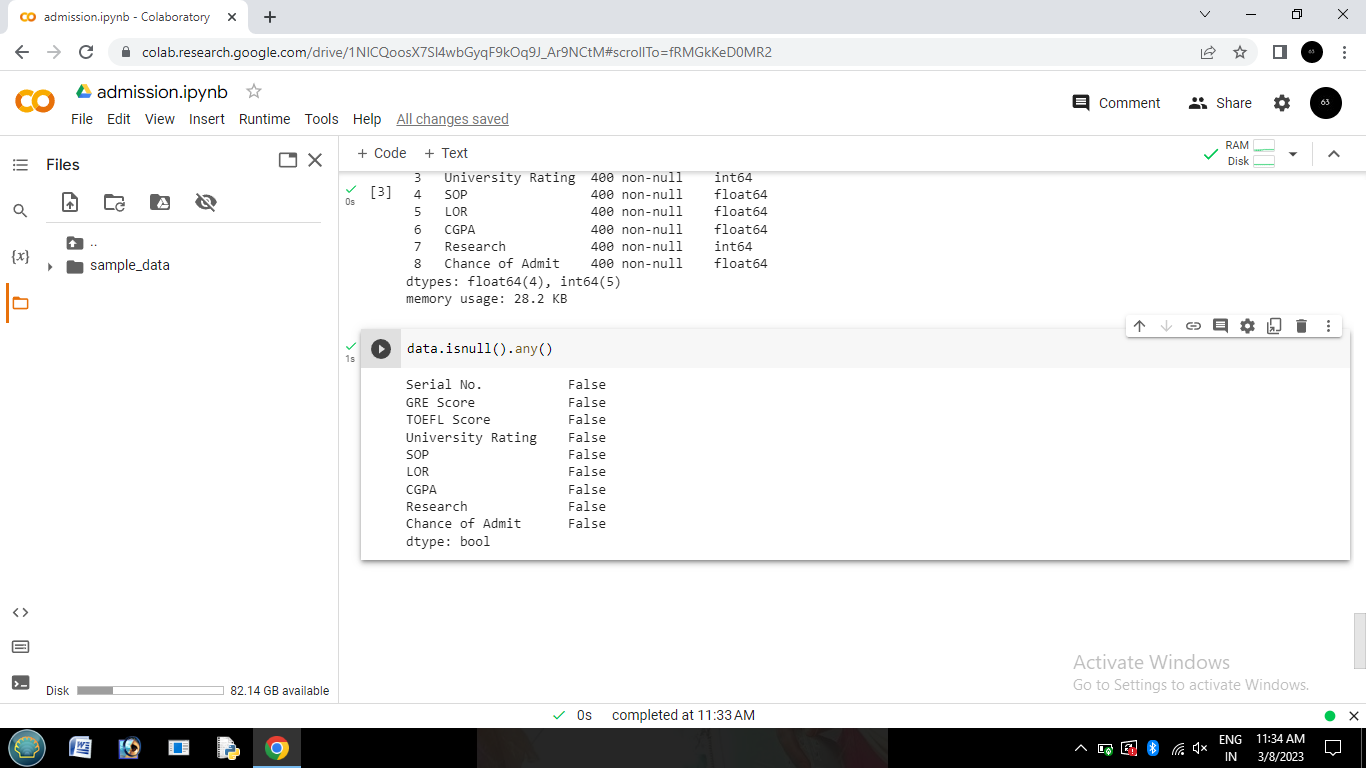
****

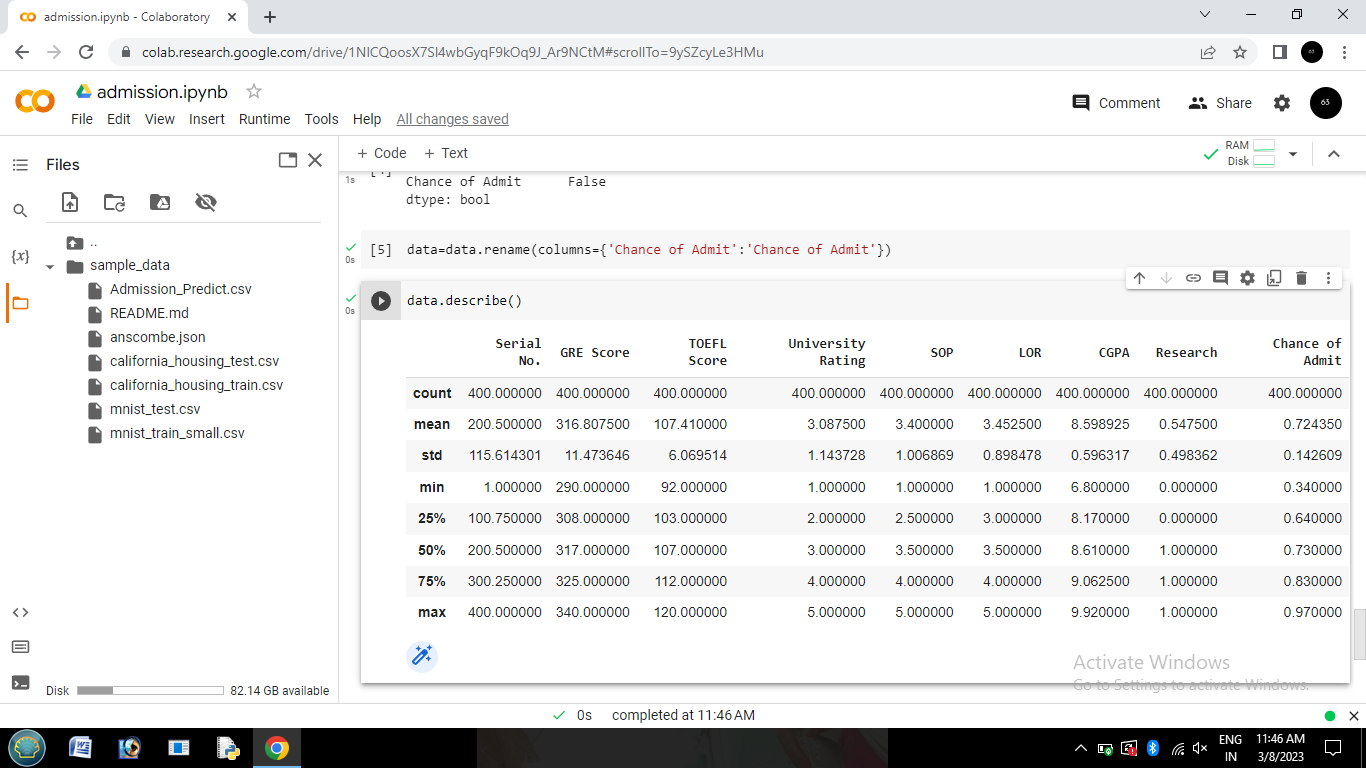
**SCREEN LAYOUT**

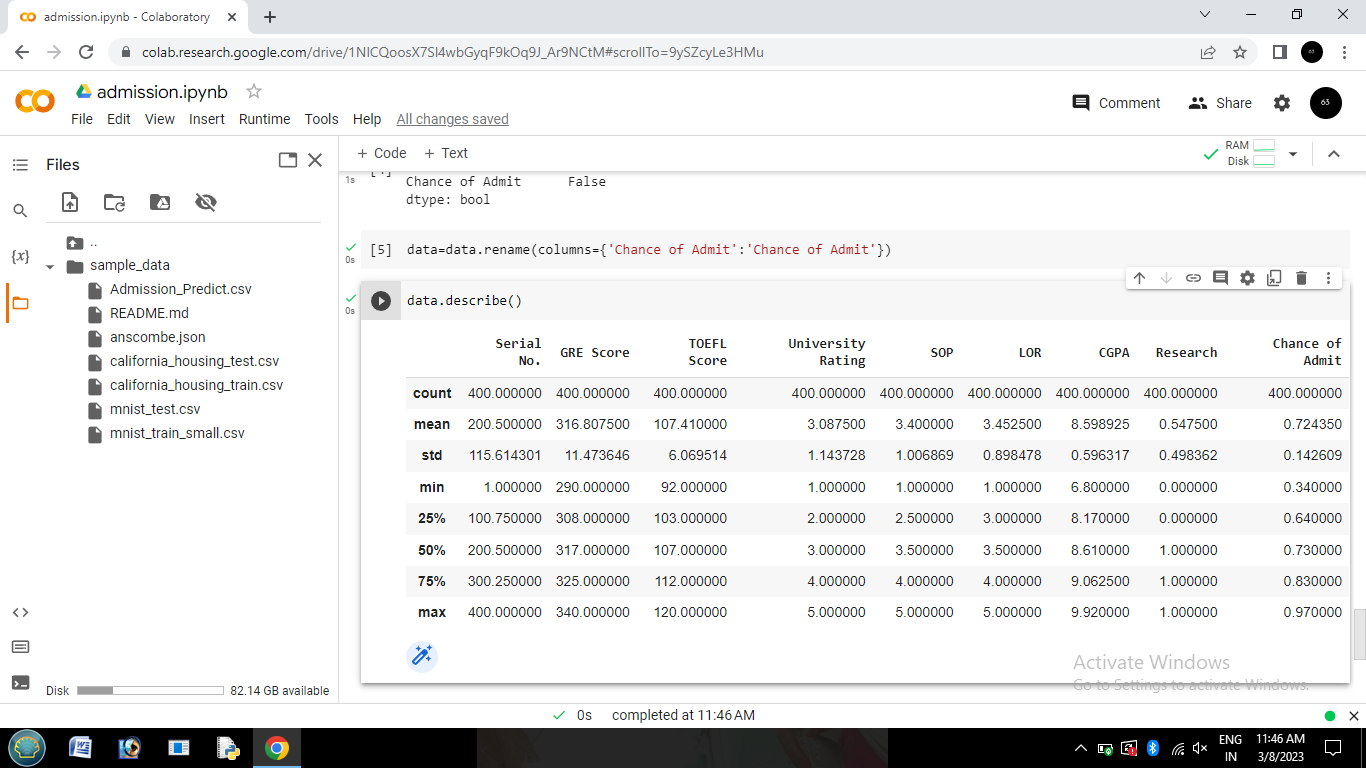
****

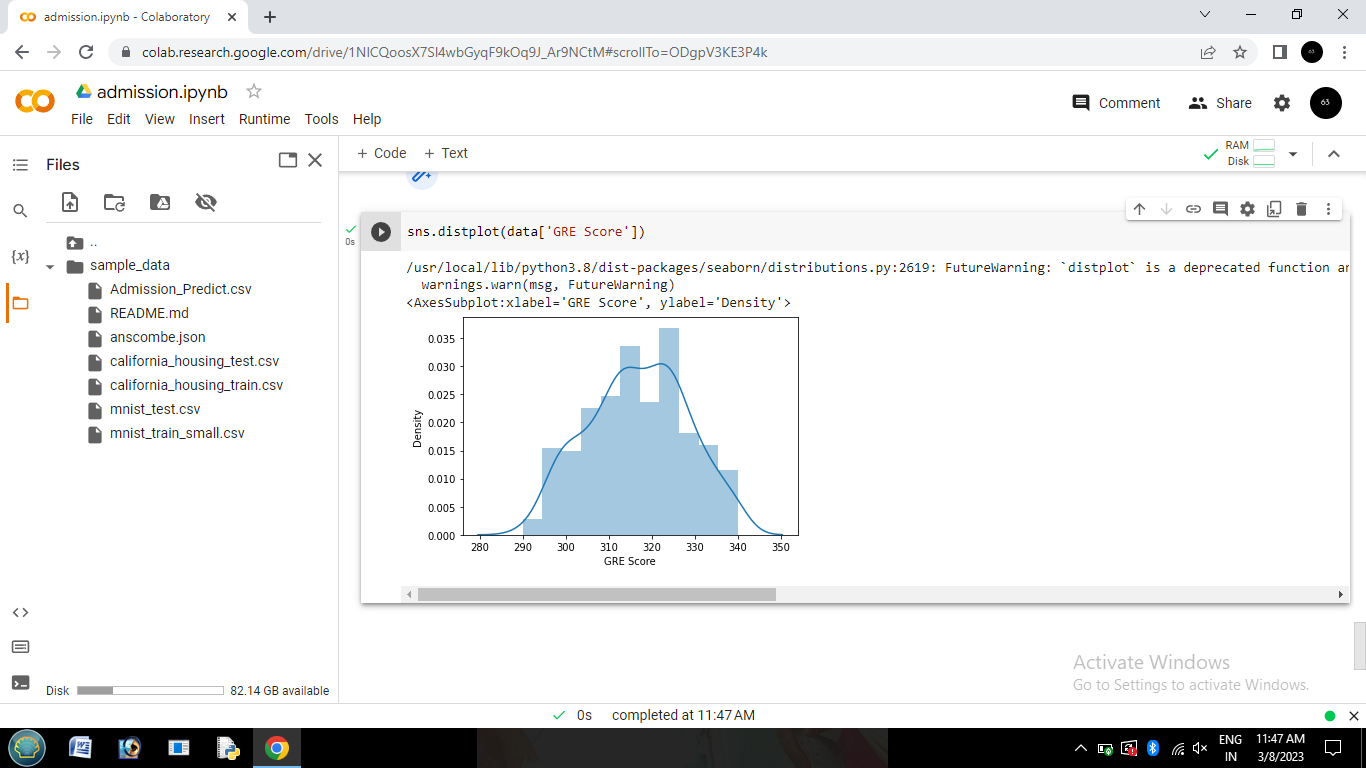
****

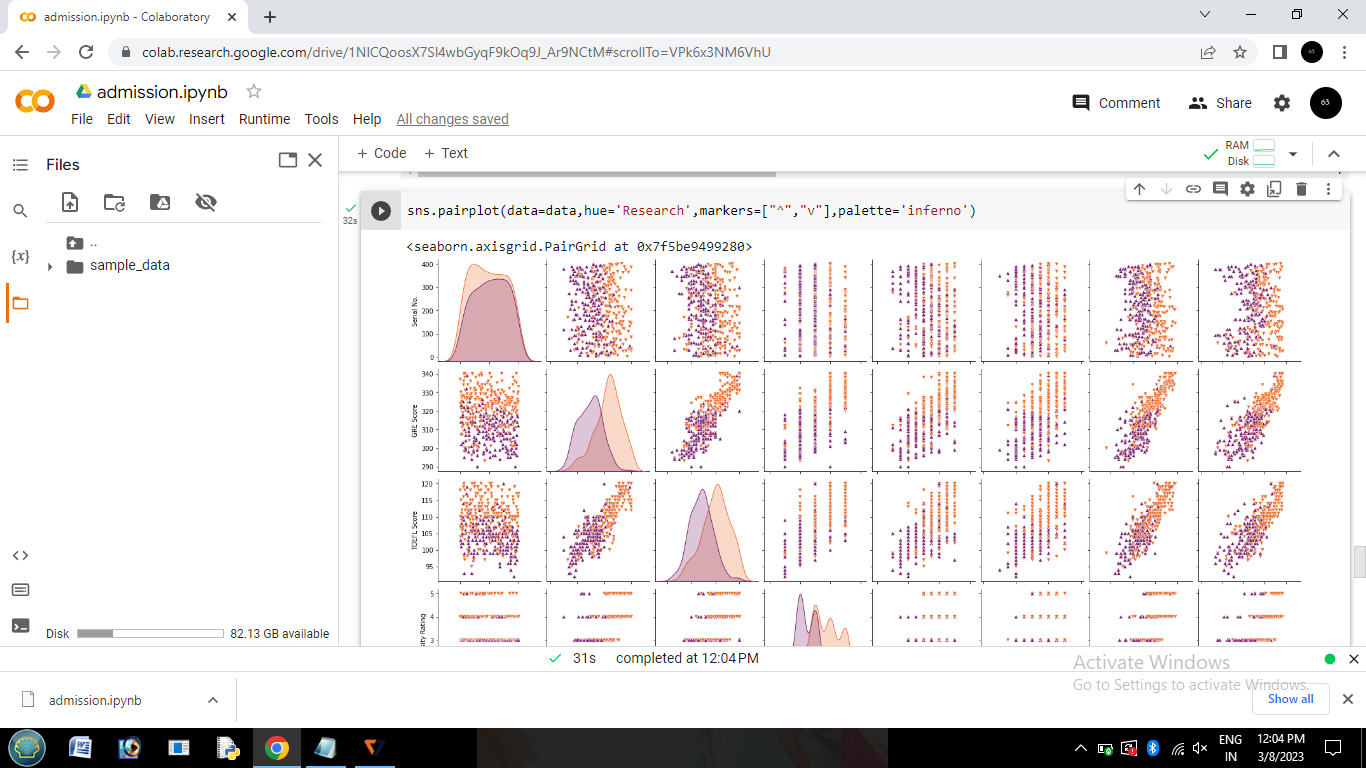
****

****

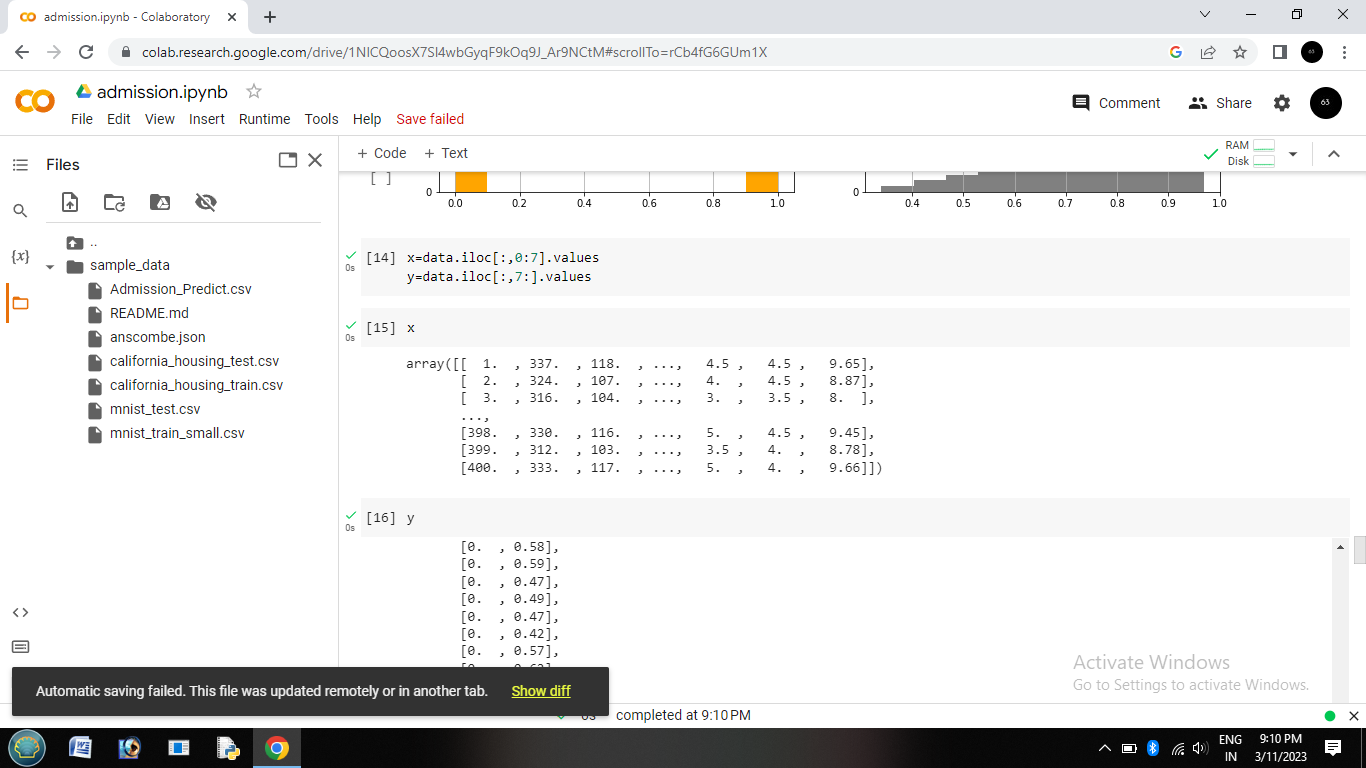
****

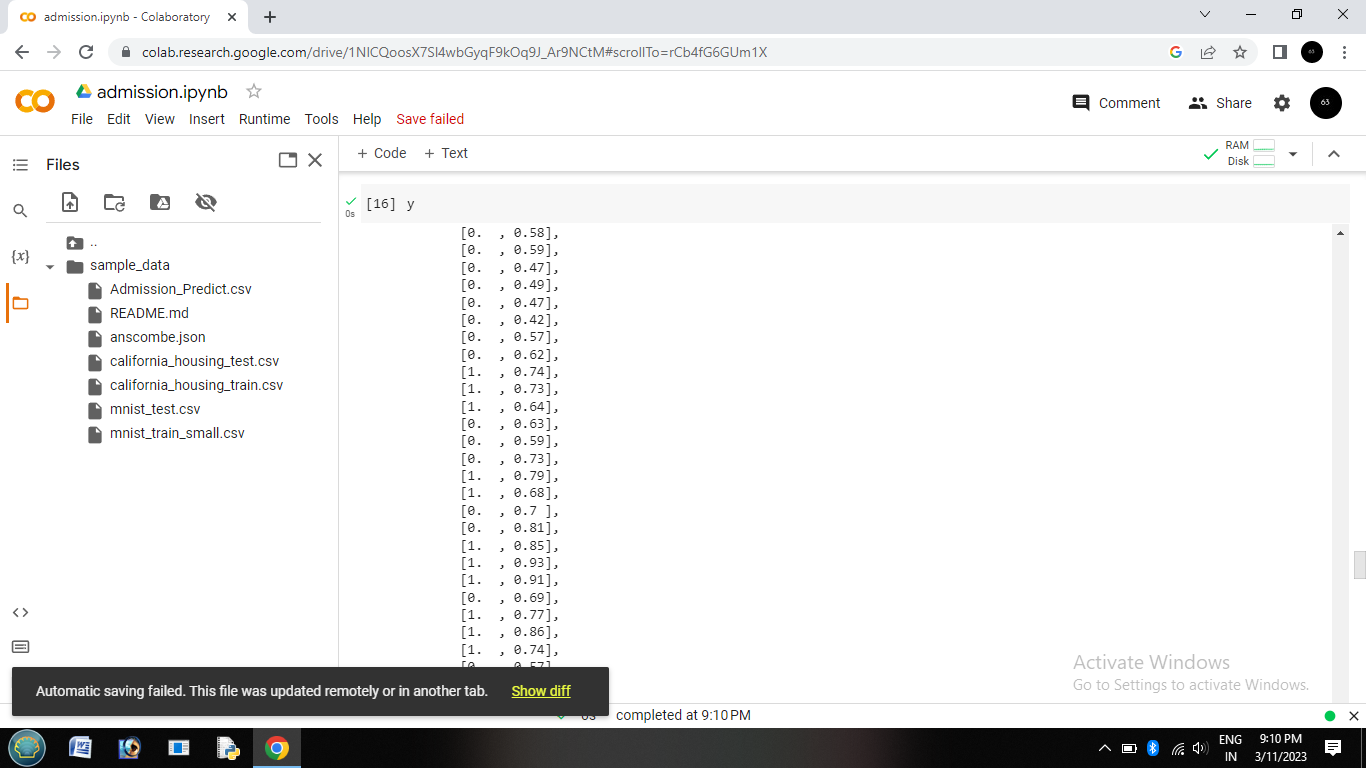
****

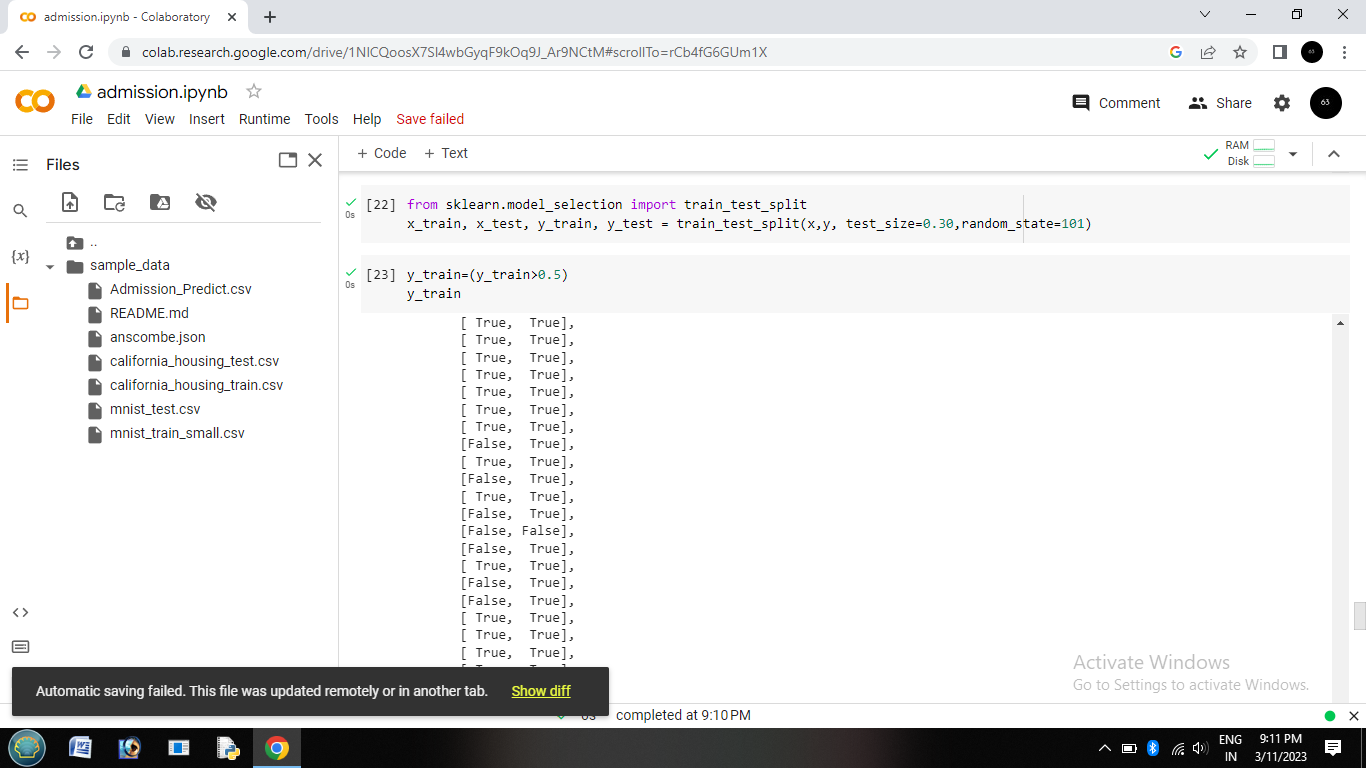
****

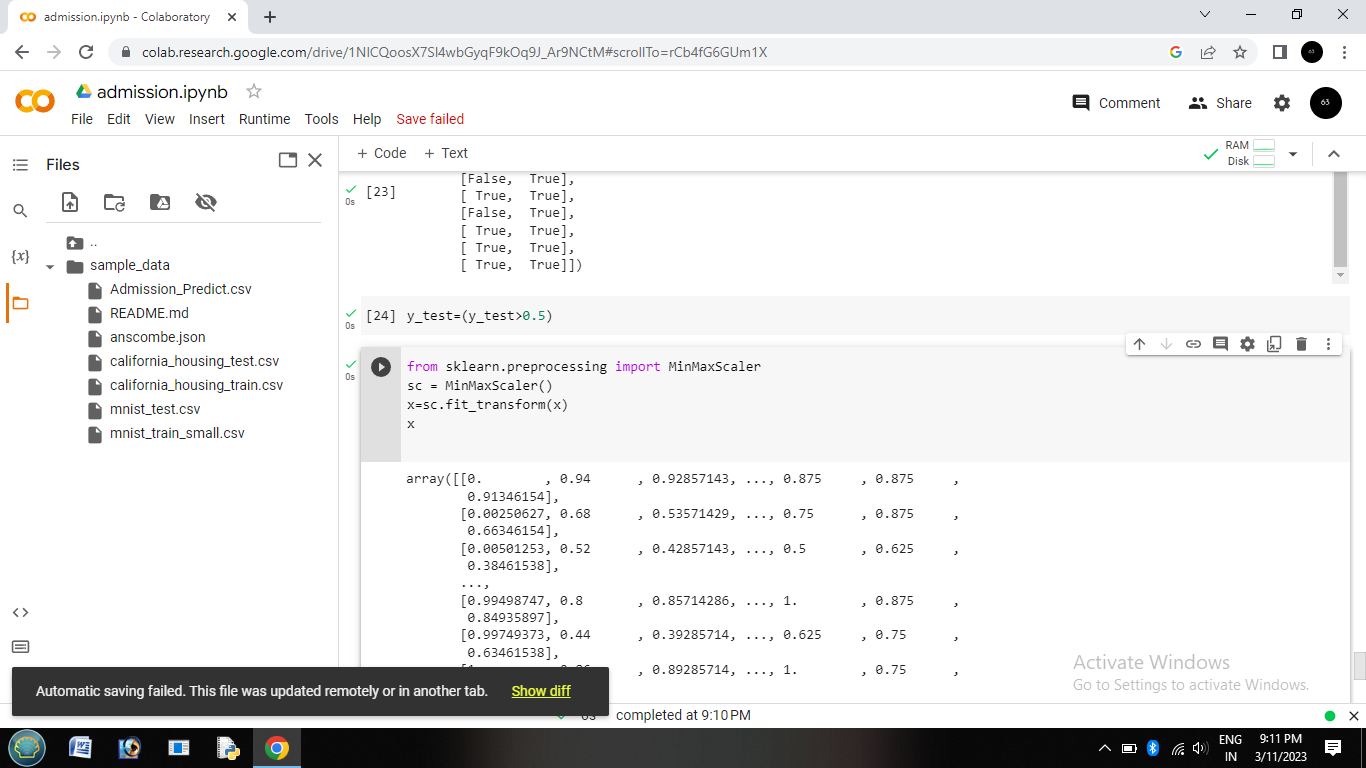
****

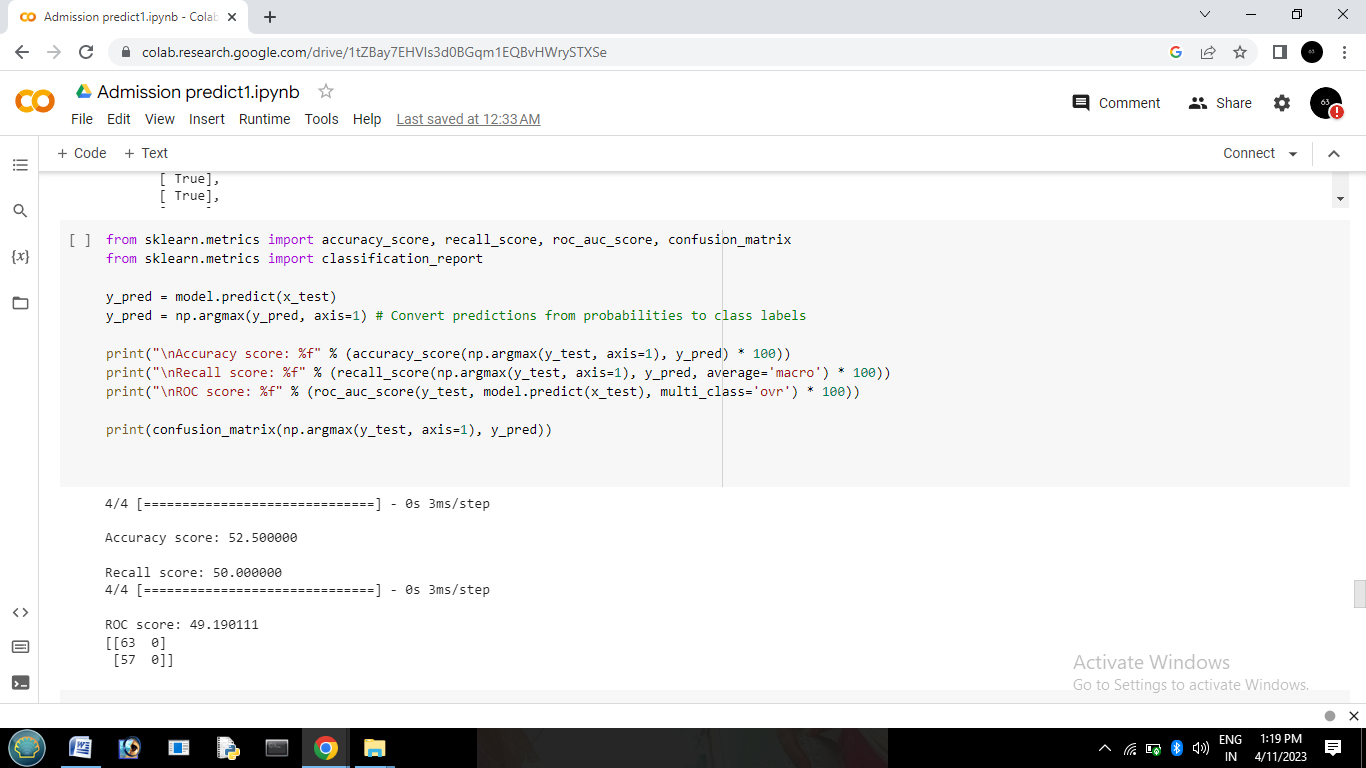
****

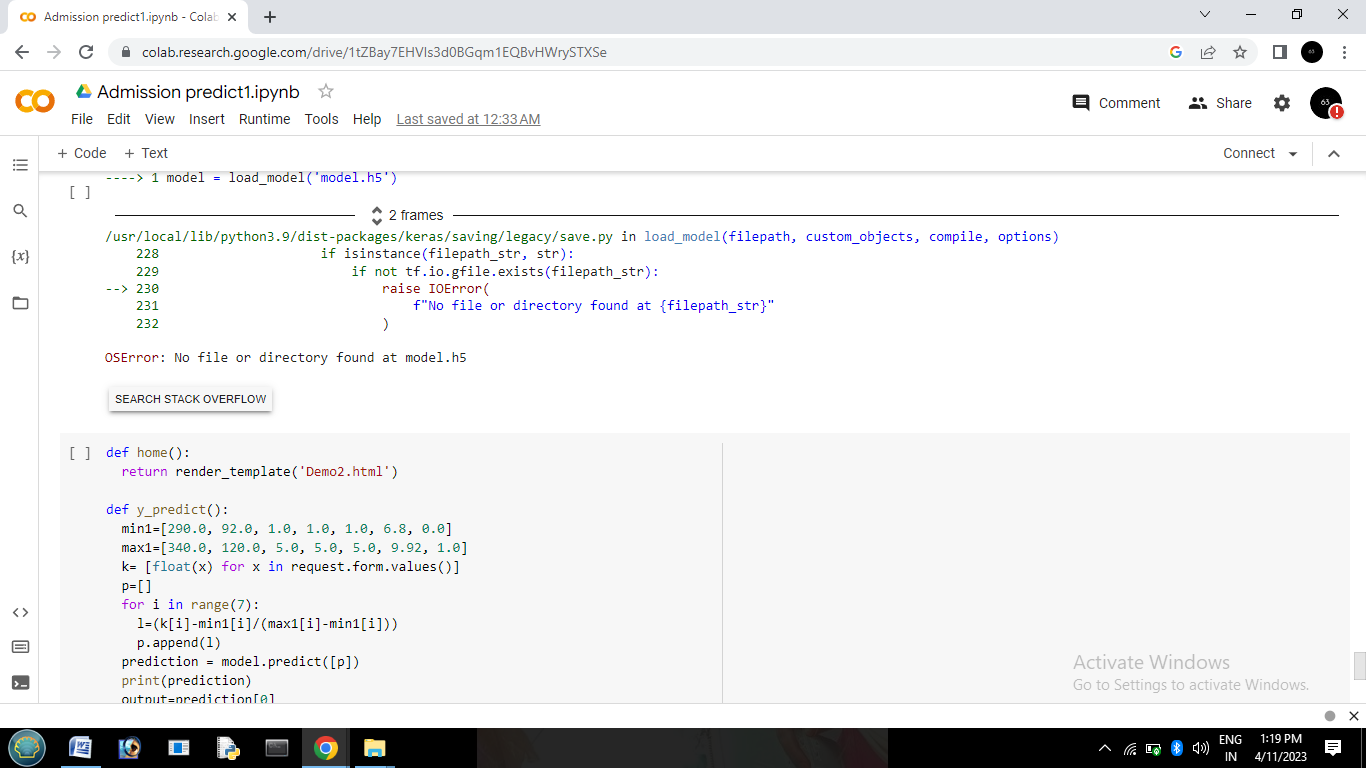
****

****

****

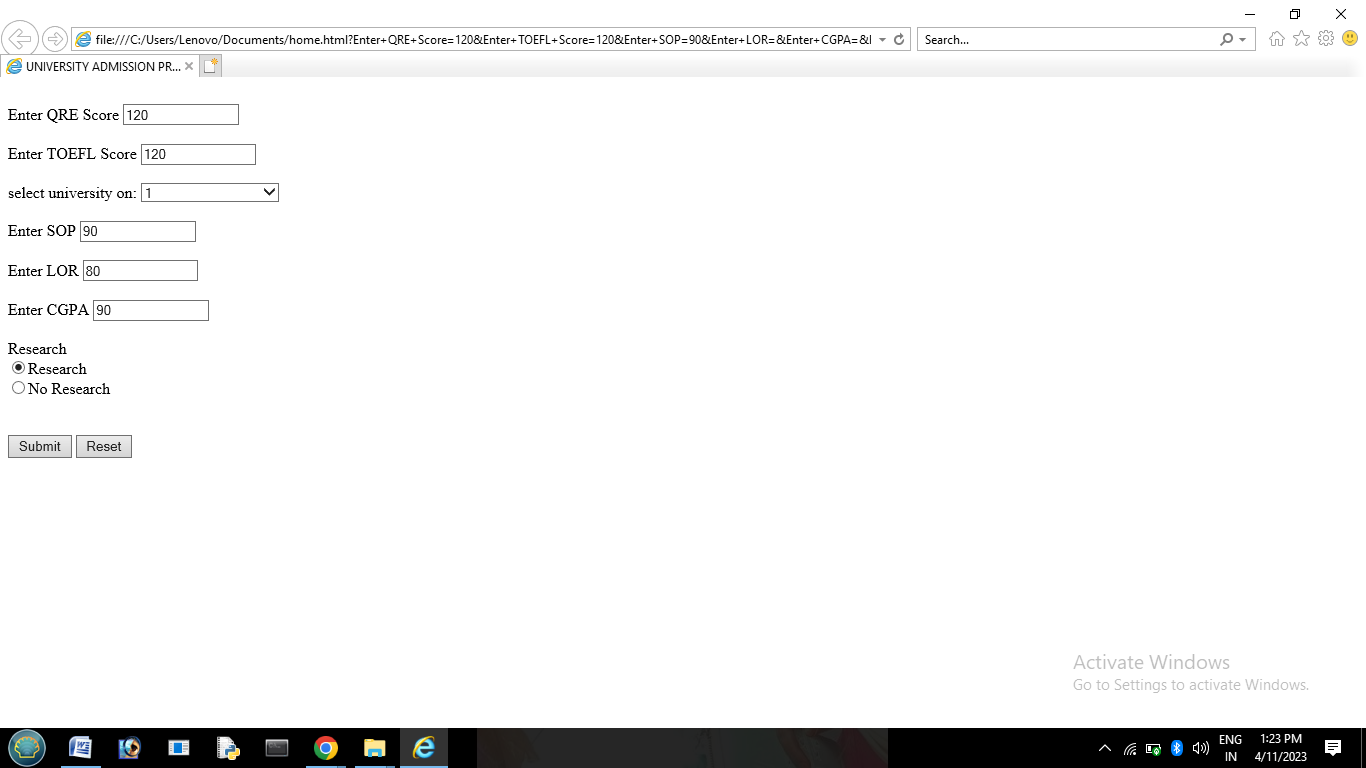
****



****

**PYCHARM**

**for html**

****

**4. ADVANTAGES & DISADVANTAGES**

Advantages:

* Increased Efficiency: Intelligent systems can quickly process large amounts of data, reducing the time and resources required for the admission process.
* Objective and Unbiased: An intelligent system can eliminate human bias from the admission process, making it more objective and fair.
* Improved Accuracy: Intelligent systems can analyze data and provide more accurate predictions and recommendations for admission.
* Consistency: With an intelligent system, all applicants are evaluated using the same criteria, ensuring consistency in the admission process.
* Enhanced Security: Intelligent systems can help prevent fraud and ensure the authenticity of application materials, improving the security of the admission process.

**Disadvantages:**

* Lack of Human Touch: An intelligent system may not be able to assess a candidate's non-academic qualities such as leadership potential, interpersonal skills, and emotional intelligence.
* Cost: Developing and implementing an intelligent admission system can be expensive, and may require ongoing maintenance and updates.
* Potential for Technical Issues: Any system can have technical glitches or errors, and an intelligent admission system is no exception. This could result in inaccurate assessments or even exclusion of certain applicants.
* Limited Scope: An intelligent system can only evaluate data that is available, and may not be able to take into account factors such as personal circumstances, potential, or extenuating circumstances that could affect an applicant's academic performance.
* Ethical Concerns: The use of an intelligent admission system raises ethical concerns about privacy, data protection, and discrimination, particularly if the system is not designed or implemented appropriately.
* It's important to note that any decision about whether or not to use an intelligent admission system should be carefully considered and weighed against these potential advantages and disadvantages.

Top of Form

Bottom of Form

Top of Form

**6**

**5. APPLICATION**

Once you’ve weighed up all the factors and carefully made your decision, it’s time for the really fun part: applying.

Though this might seem obvious, ensure you take care over this. You don’t want to miss out simply because you forgot to submit the required evidence or applied too late.

“Students should check entry requirements and deadlines before applying to make sure that they have the best possible chance of gaining a place on their chosen course,” Berry confirms.

“If they are unsure whether their qualifications are acceptable, they might like to contact the admissions office or international office in their chosen institution to check before submitting a full application.”

She emphasizes the importance of applying in good time: “Students should try to make an application as early as possible as this will give them plenty of time to make all the necessary arrangements for a move abroad, including organizing their finances, applying for scholarships and obtaining a student visa.”

**6. CONCLUSION**

* In conclusion, admission prediction is an important task in the field of education that can help both students and universities make better-informed decisions about the college admissions process. By developing models that can accurately predict whether a student will be admitted to a particular school based on their input features, admission prediction can help students identify which schools they are most likely to be accepted to, and can assist universities in evaluating applicants more efficiently and objectively.
* However, it is important to use admission prediction models ethically and responsibly, and to ensure that they are not used to unfairly discriminate against certain groups of applicants. Additionally, admission prediction models should be constantly monitored and updated to ensure that they remain accurate and unbiased over time.
* Overall, admission prediction has the potential to improve the fairness and efficiency of the college admissions process, and can provide valuable insights into the factors that contribute to admission decisions. By continuing to develop and refine admission prediction models, we can help ensure that the college admissions process is as transparent and equitable as possible.

Top of Form

Bottom of Form

**7. FUTURE ENHANCEMENT**

* There are several future enhancements that can be made to admission prediction models to improve their accuracy and usefulness:
* Incorporating more data: Admission prediction models can be improved by incorporating additional data sources, such as social media activity, personal essays, and letters of recommendation. This can provide a more comprehensive view of the applicant and improve the accuracy of the model.
* Using more advanced machine learning algorithms: While logistic regression and decision trees are commonly used for admission prediction, more advanced machine learning algorithms such as random forests, gradient boosting, and deep learning can be applied to improve model accuracy.

**8 . APPENDIX**

**step 1**

**import numpy as np**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**%matplotlib inline**

**data=pd.read\_csv('/content/sample\_data/Admission\_Predict.csv')**

**data**

**step 2**

**data.info()**

**step 3**

**data.isnull().any()**

**step 4**

**data=data.rename(columns={'Chance of Admit ':'Chance of Admit'})**

**step 5**

**data.describe()**

**step 6**

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

**step 7**

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

**step 8**

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

**step 9**

**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()**

**step 10**

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

**y=data.iloc[:,0:7].values**

**step 11**

**x**

**step12**

**y**

**step13**

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

**x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y, text\_size=0.30,random\_state=101)**

**step 14**

**y\_train=(y\_train>0.5)**

**y\_train**

**step 15**

**y\_text=(y\_text>0.5)**

**step 16**

**from sklearn.preprocessing import MinMaxScaler**

**sc=MinMaxscaler()**

**x=sc.fit\_transform(x)**

**x**

**step 17**

**from sklearn.linear\_model.logistic import LogisticRegression**

**cls =LogisticRegression(random\_state =0)**

**lr=cls.fit(x\_train, y\_train)**

**y\_pred =lr.predict(x\_test)**

**y\_pred**

**step 18**

**import tensorflow as tf**

**from tensorflow import keras**

**from keras import Sequential**

**from keras.layers import Dense**

**step 19**

**model=keras.Sequential()**

**model.add(Dense(7,activation = 'relu',input\_dim=7))**

**model.add(Dense(7,activation='relu'))**

**model.add(Dense(1,activation='linear'))**

**model.summary()**

**step 20**

**model.summary()**

**model.compile(loss = 'binary\_crossentropy', optimizer = 'adam',metrics = ['accuracy'])**

**model.fit(x\_train, y\_train, batch\_size = 20, epochs = 100)**

**from sklearn.metrics import accuracy\_score**

**train\_predictions = model.predict(x\_train)**

**print(train\_predictions)**

**train\_acc = model.evaluate(x\_train, y\_train, verbose=0)[1]**

**print(train\_acc)**

**test\_acc = model.evaluate(x\_test, y\_test, verbose=0)[1]**

**print(test\_acc)**

**print(classification\_report(y\_text,pred))**

**pred=model.predict(x\_test)**

**pred = (pred>0.5)**

**pred**

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

**print("\nAccuracy score: %f" %(accuracy\_score(y\_test,y\_pred) \* 100))**

**print("Recall score : %f" %(recall\_score(y\_test,y\_pred) \* 100))**

**print("ROC score : %f\n" %(roc\_auc\_score(y\_test,y\_pred) \* 100))**

**print(confusion\_matrix(y\_test,y\_pred))**

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

**print(classification\_report(y\_train,pred))**

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

**print(classification\_report(y\_test,pred))**

**model.save('model.h5')**

**import numpy as np**

**from flask import Flask, request, jsonify, render\_template**

**import pickle**

**app = Flask(\_\_name\_\_)**

**from tensorflow.keras.models import load\_model**

**#model = pickle.load(open('University.pkl', 'rb'))**

**model = load\_model('model.h5')**

**@app.route('/')**

**def home():**

**return render\_template('Demo2.html')**

**@app.route('/y\_predict',methods=['POST'])**

**def y\_predict():**

**min1=[290.0, 92.0, 1.0, 1.0, 6.8,0.0]**

**max1=[340.0, 120.0, 5.0, 5.0, 5.0, 9,92, 1.0]**

**k= [float(x) for x in request.form.values()**

**p[]**

**for i in range(7):**

**l=(k[i]-min1[i])/(max1[i]-min1[i])**

**p.append(1)**

**prediction = model.predict([p])**

**print(prediction)**

**output=prediction[0]**

**if(output==False):**

**return render\_template('nochance.html', prediction\_text='You Dont have a chance of getting')**

**else:**

**return render\_template('chance.html', prediction\_text='You have a chance of getting admission')**

**if \_\_name\_\_ == "\_\_main\_\_":**

**app.run(debug=False)**