# Intelligent Admissions: The Future of University Decision Making with Machine Learning

The project submitted to Smart Internz
By

DHARUN .S (2022K1995)
DHIVYA BHARATHI .R (2022K1996)
ELANGOVAN .G (2022K1997)
ELAVARASAN .S (2022K1998)
VARUN .C (2022K2026)

Department of Computer Science

GOVERNMENT ARTS AND SCIENCE

COLLEGE MODAKURICHI

(Affiliated to Bharathiar University)

ERODE - 638104


# 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

## 1.1 Overview

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.

## 1.2 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.

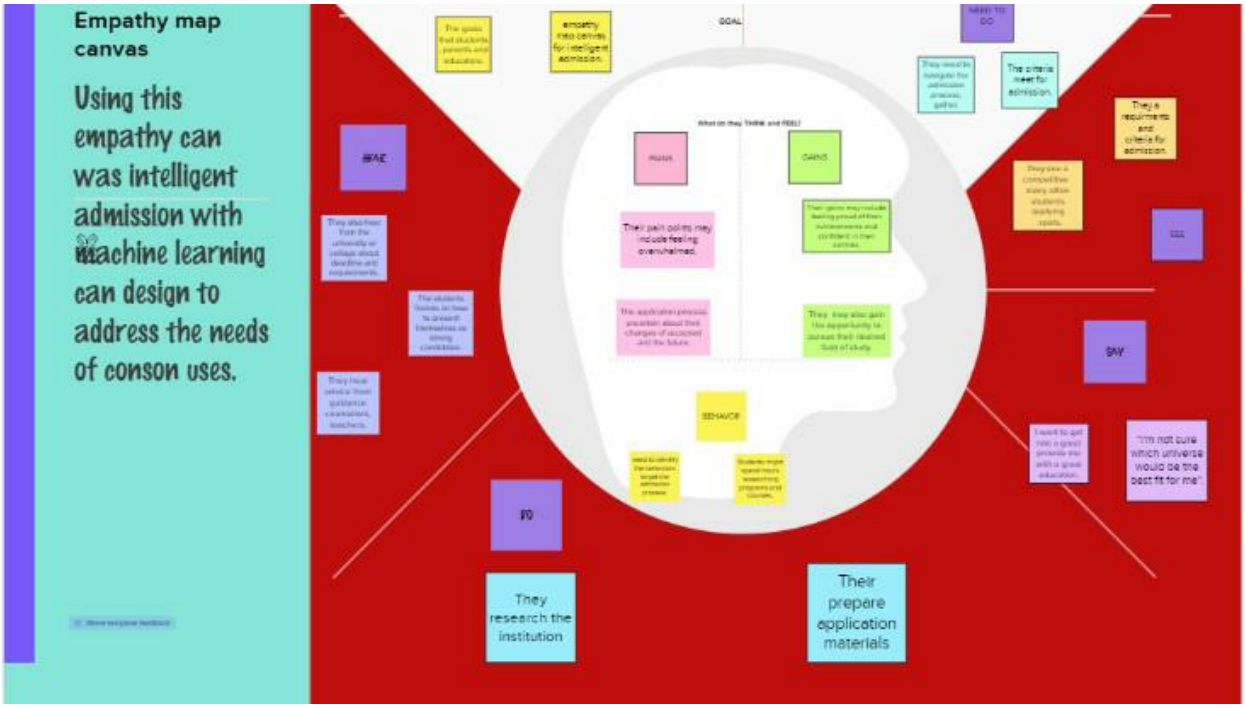
For universities, admission prediction models can assist with the admissions process by providing a more efficient and objective way of evaluating applicants. By using data-driven models to predict admission decisions, universities can potentially reduce bias and ensure that all applicants are evaluated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

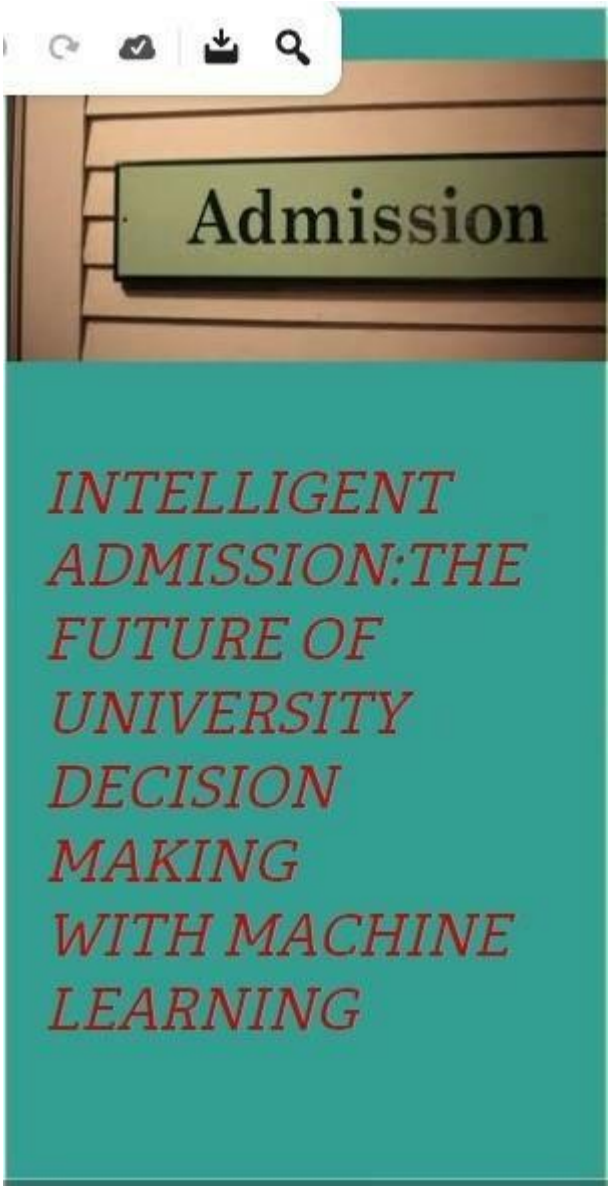
## 2.1 Empathy map

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

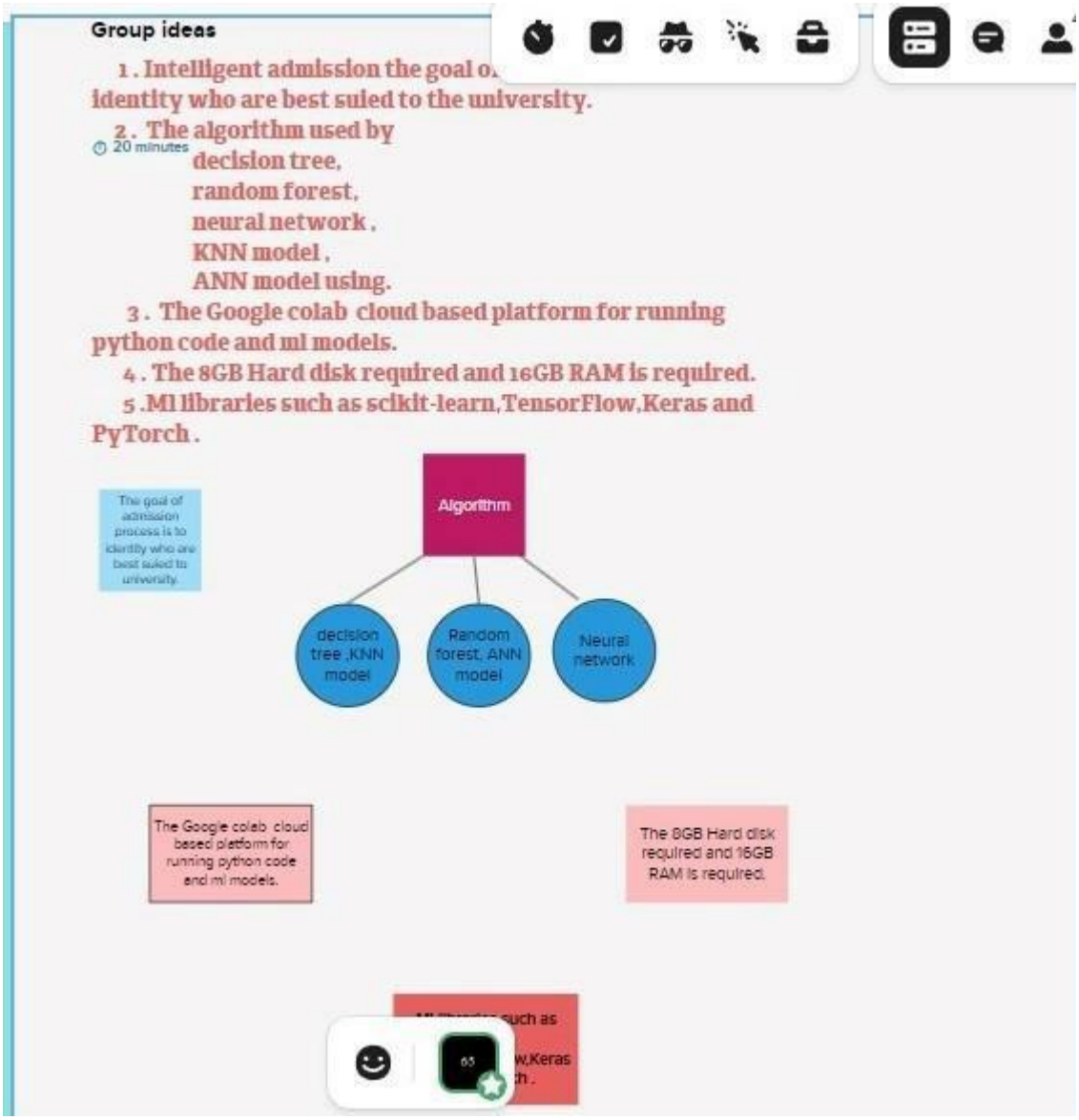


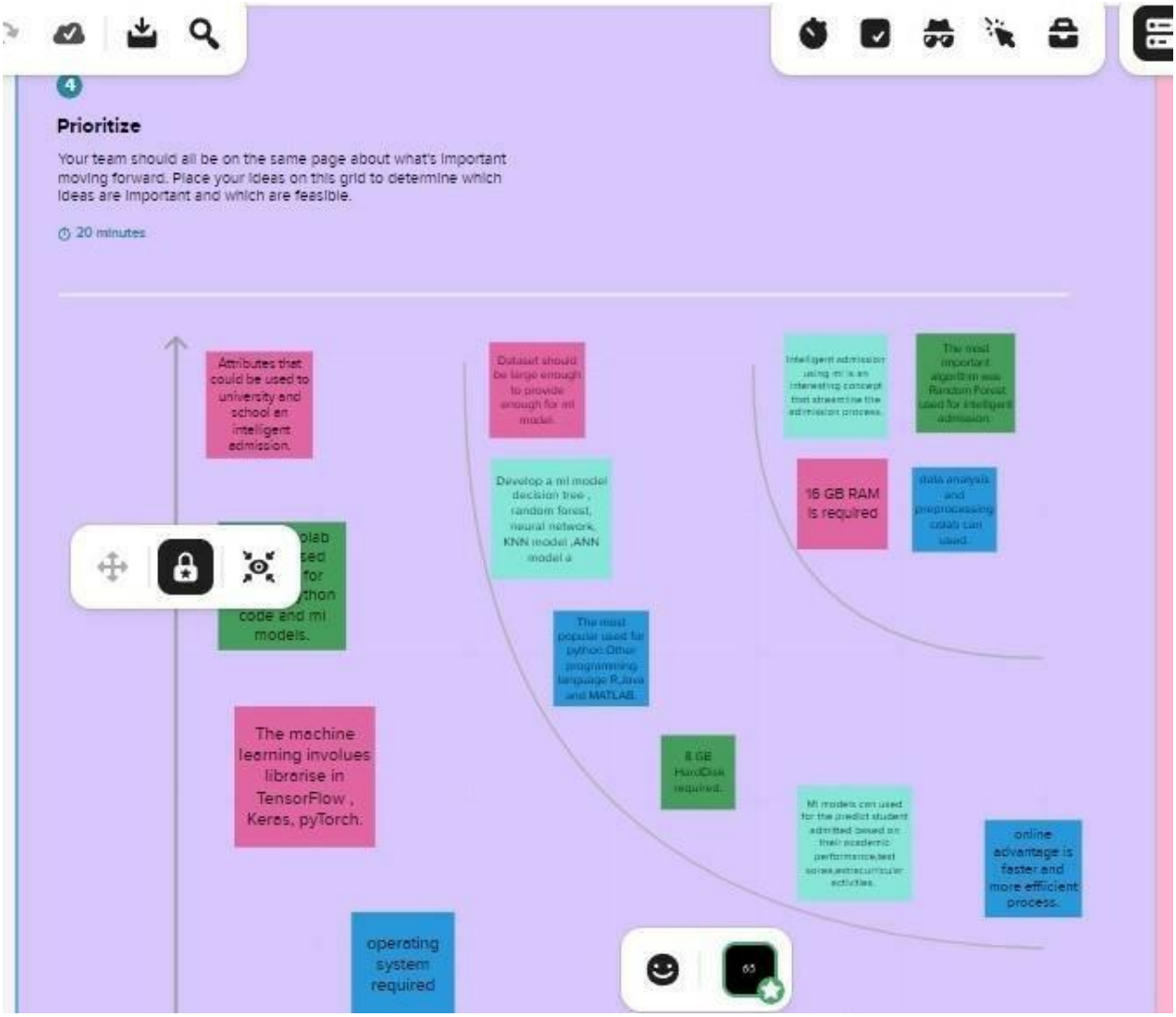
## 2.2 Ideation & Brainstrom Map

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

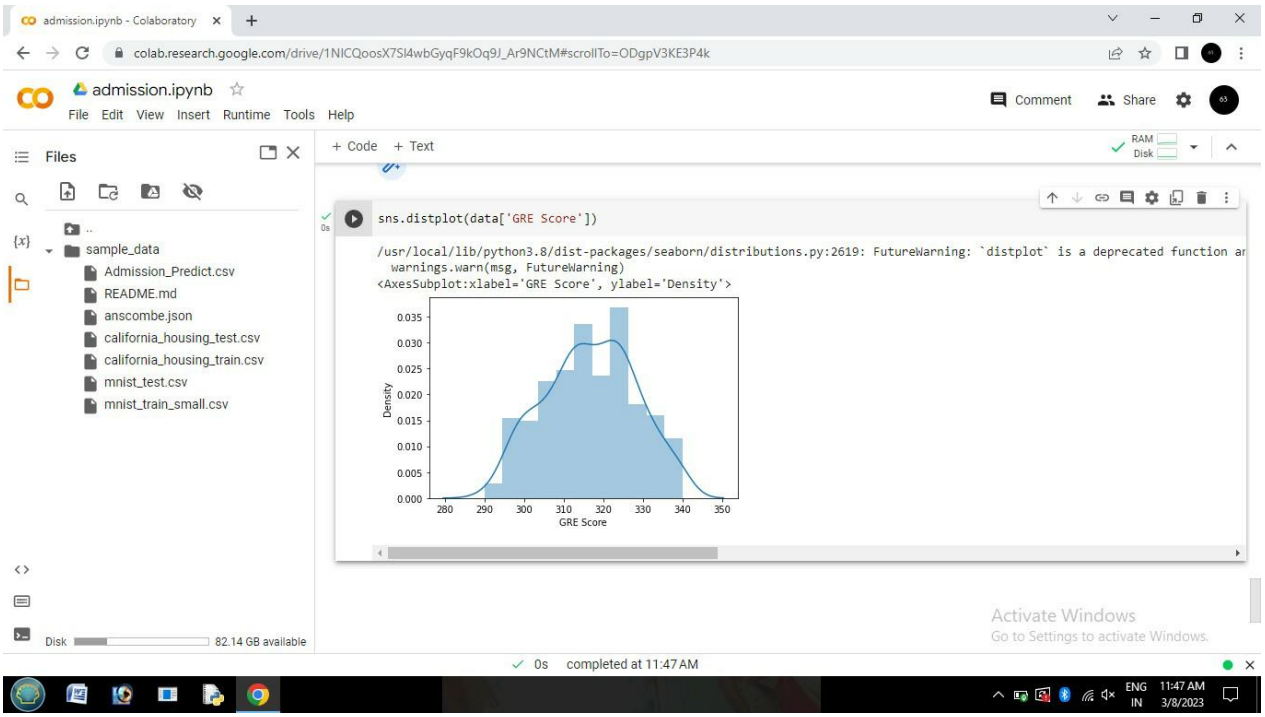


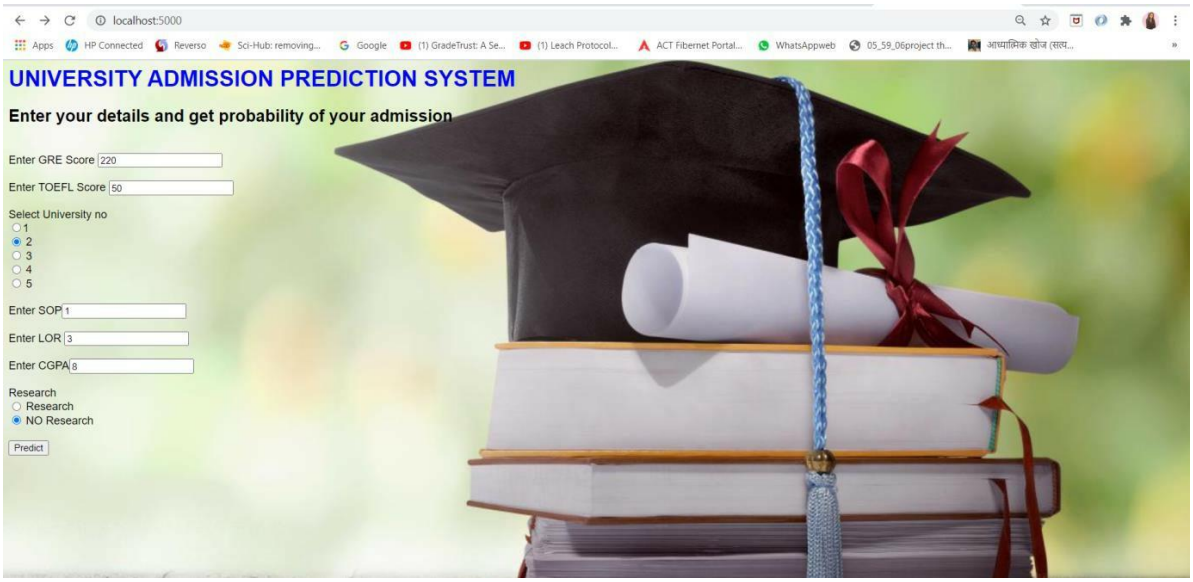

Define your problem statement
1. student admission are playing very important role in major activity of any university
2. The aim of project is to help student in short listing university with their profiles
3. This project is desigh to development intelligent admission
4. The goal intelligent admission is provide convience,savetime,bring more object , transperancy and speed transaction over the manual opertion.
5. This predicted output gives them fair idea about their admission.


Group ideas
1 . Intelligent admission the goal o.
identity who are best suied to the university.
2 . The algorithm used by
decision tree,
random forest,
neural network ,
KNN model ,
ANN model using.
3 . The Google colab cloud based platform for running python code and ml models.
4 . The 8GB Hard disk required and 16GB RAM is required.
5 .Ml libraries such as scikit-learn,TensorFlow,Keras and PyTorch .
20 minutes
The goal of admission process is to identify who are best suied to university.
Algorithm
decision tree ,KNN model
Random forest, ANN model
Neural network
The Google colab cloud based platform for running python code and ml models.
The 8GB Hard disk required and 16GB RAM is required.

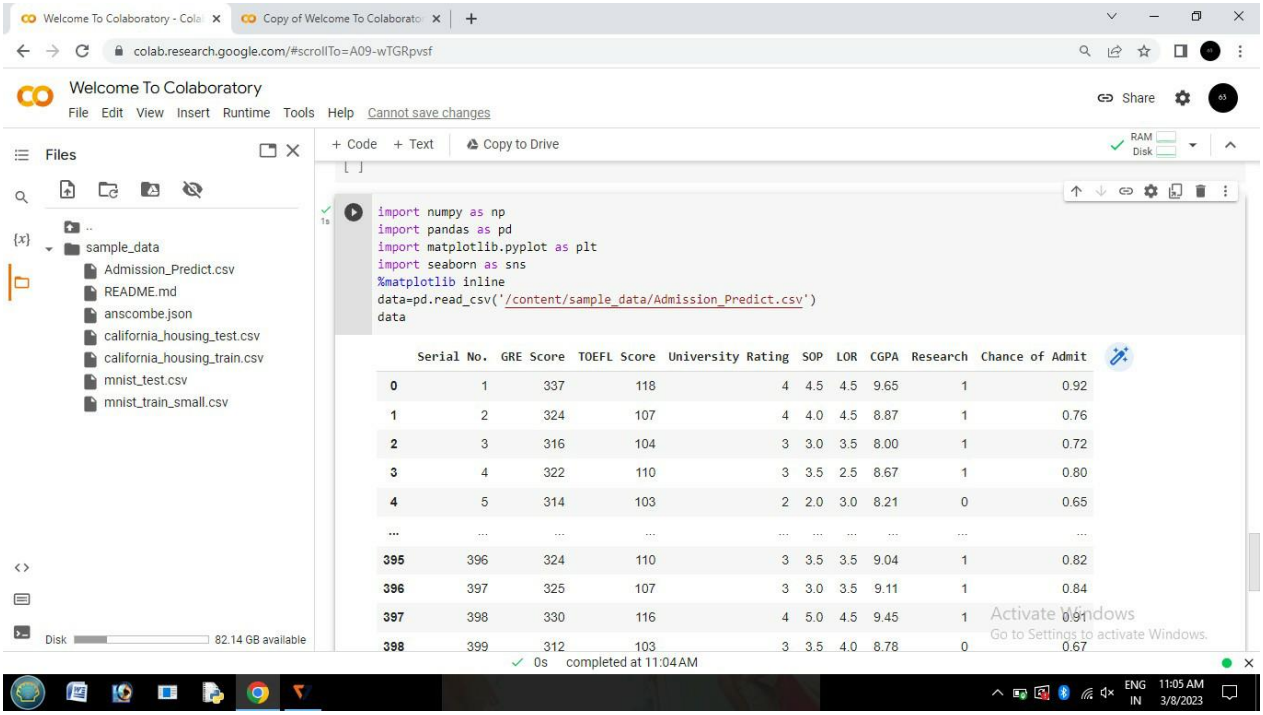

4
Prioritize
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.
20 minutes
Attributes that could be used to university and school an intelligent admission.
Dataset should be large enough to provide enough for ml model.
Intelligent admission using ml is an interesting concept that streamline the admission process.
The most important algorithm was Random Forest used for intelligent admission
Develop a ml model decision tree , random forest, neural network, KNN model ,ANN model a
16 GB RAM is required
data analysis and preprocessing could can used.
code and ml models.
The most popular used for python.Other programming language R,Java and MATLAB.
The machine learning involues librarise in TensorFlow , Keras, pyTorch.
8 GB HardDisk required.
ML models can used for the predict student admitted based on their academic performance,test scores,extracurricular activities.
online advantage is faster and more efficient process.
operating system required

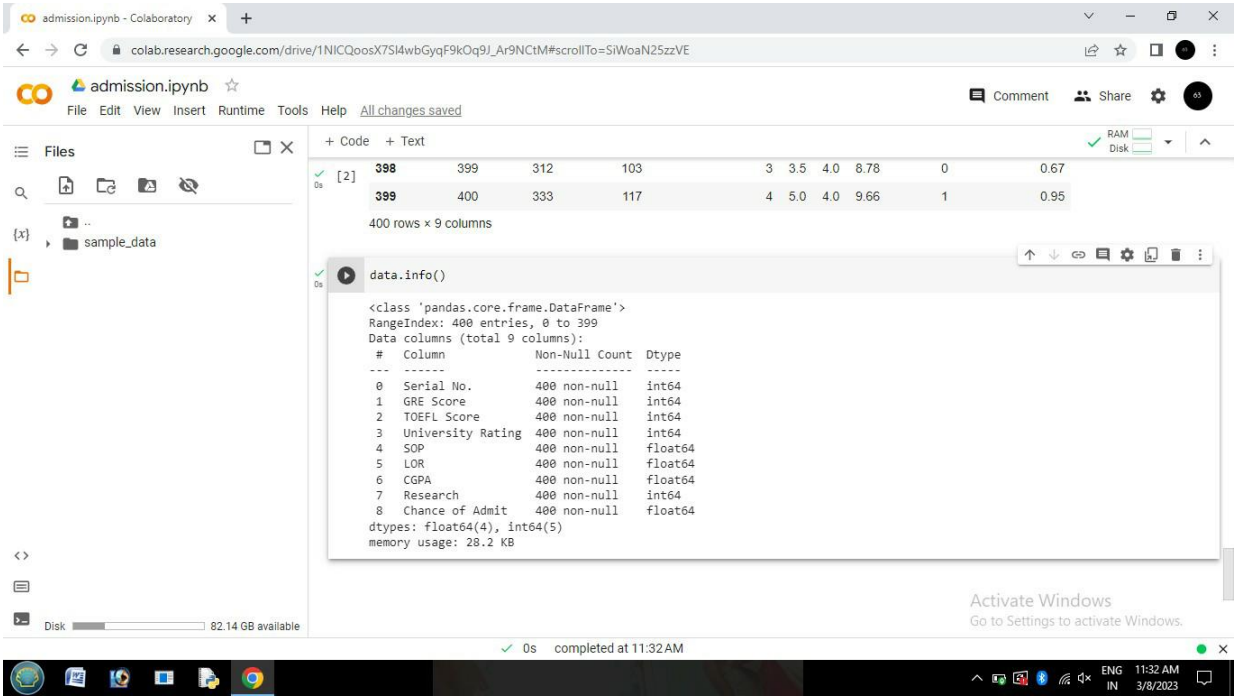
# 3. RESULT

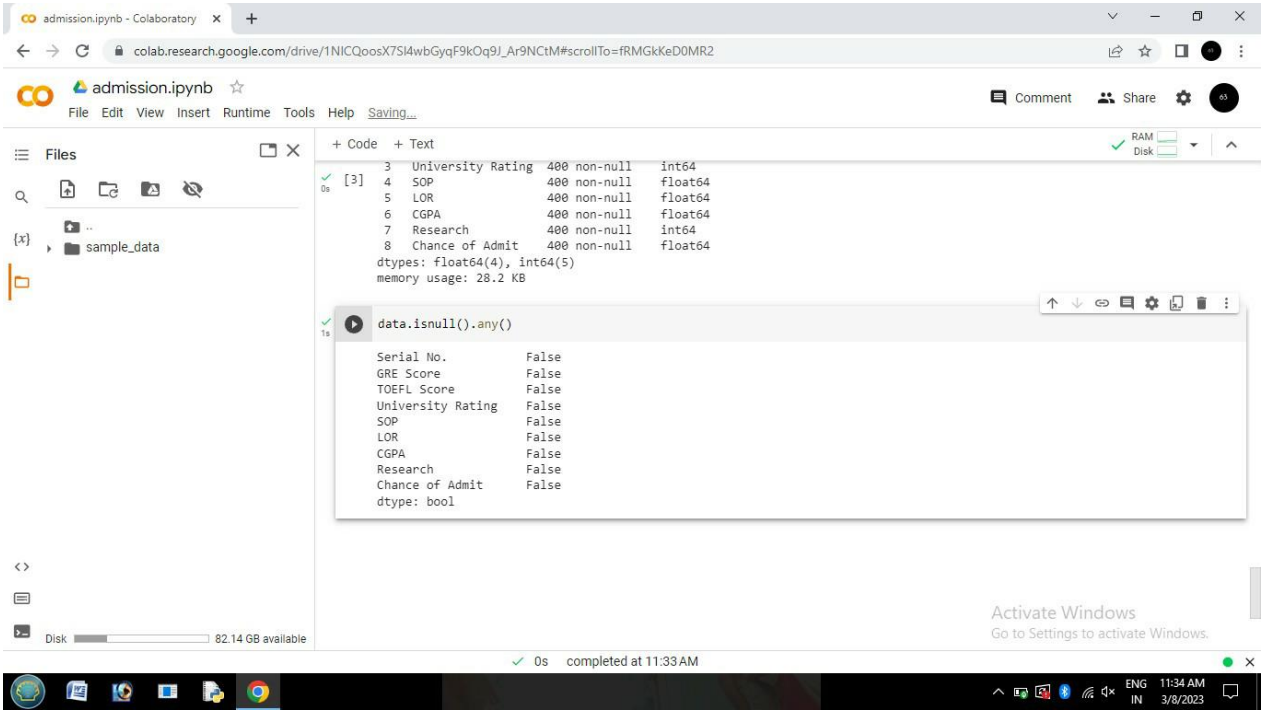

admission.ipynb - Colaboratory
colab.research.google.com/drive/1NICQoosX7Sl4wbGyqF9kOq9J_Ar9NCtM#scrollTo=ODgpV3KE3P4k
admission.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
Files
+ Code + Text
RAM
Disk
sample_data
Admission_Predict.csv
README.md
anscombe.json
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
sns.distplot(data['GRE Score'])
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function ar
warnings.warn(msg, FutureWarning)
<AxesSubplot:xlabel='GRE Score', ylabel='Density'>
Density
GRE Score
Disk 82.14 GB available
Activate Windows
Go to Settings to activate Windows.
0s completed at 11:47 AM
ENG
IN
11:47 AM
3/8/2023

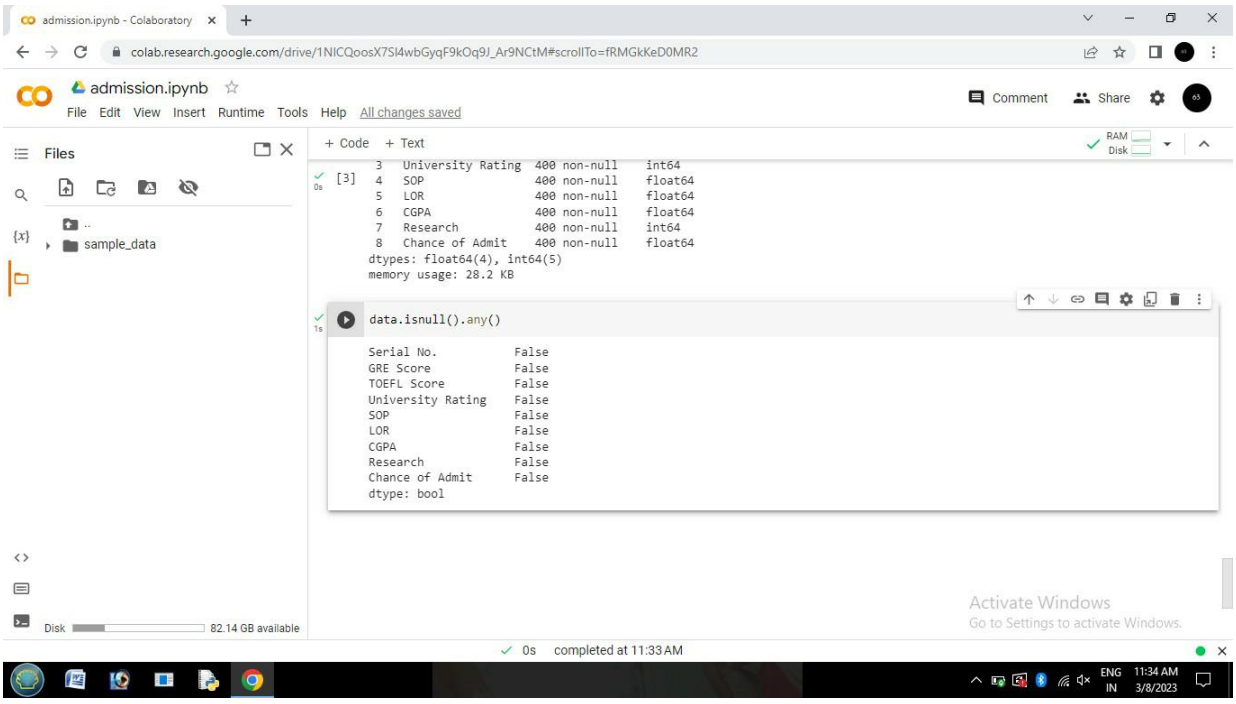

localhost:5000
Apps
HP Connected
Reverso
Sci-Hub: removing...
Google
(1) GradeTrust: A Se...
(1) Leach Protocol...
ACT Fibernet Portal...
WhatsAppweb
05_59_06project th...
आध्यामिक खोज (सत्य...
UNIVERSITY ADMISSION PREDICTION SYSTEM
Enter your details and get probability of your admission
Enter GRE Score 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
Predict

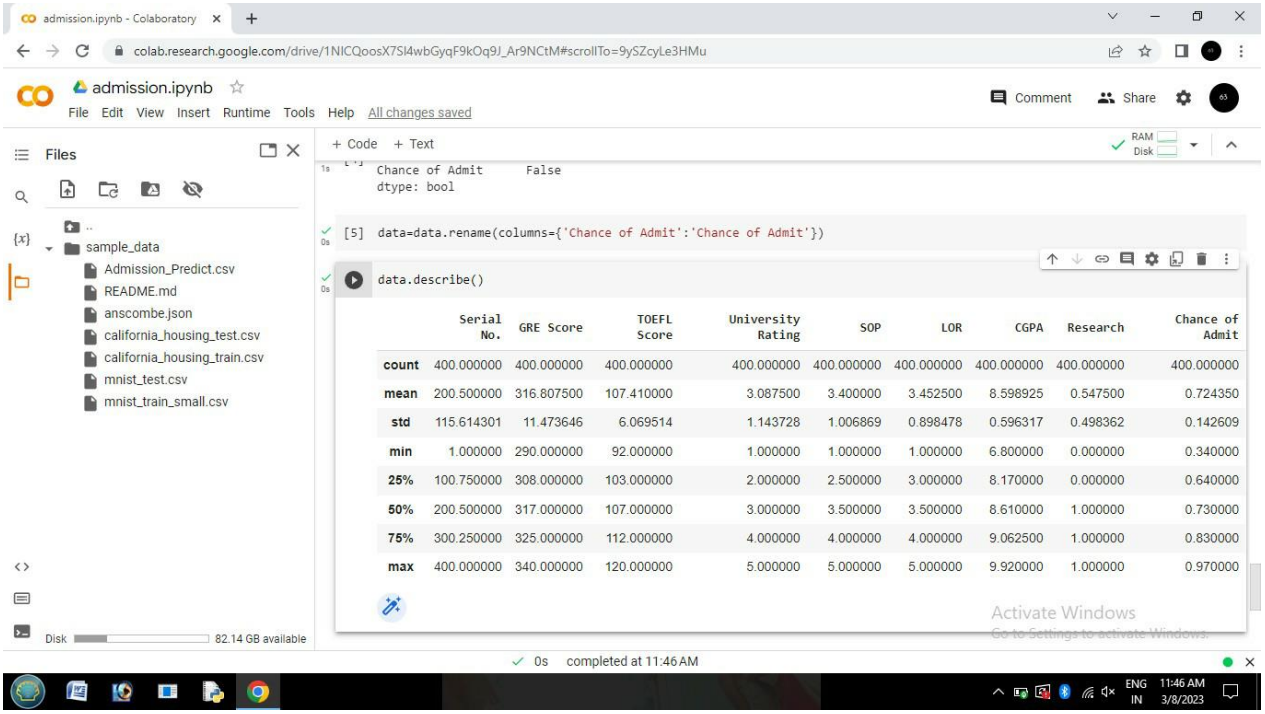
# SCREEN LAYOUT



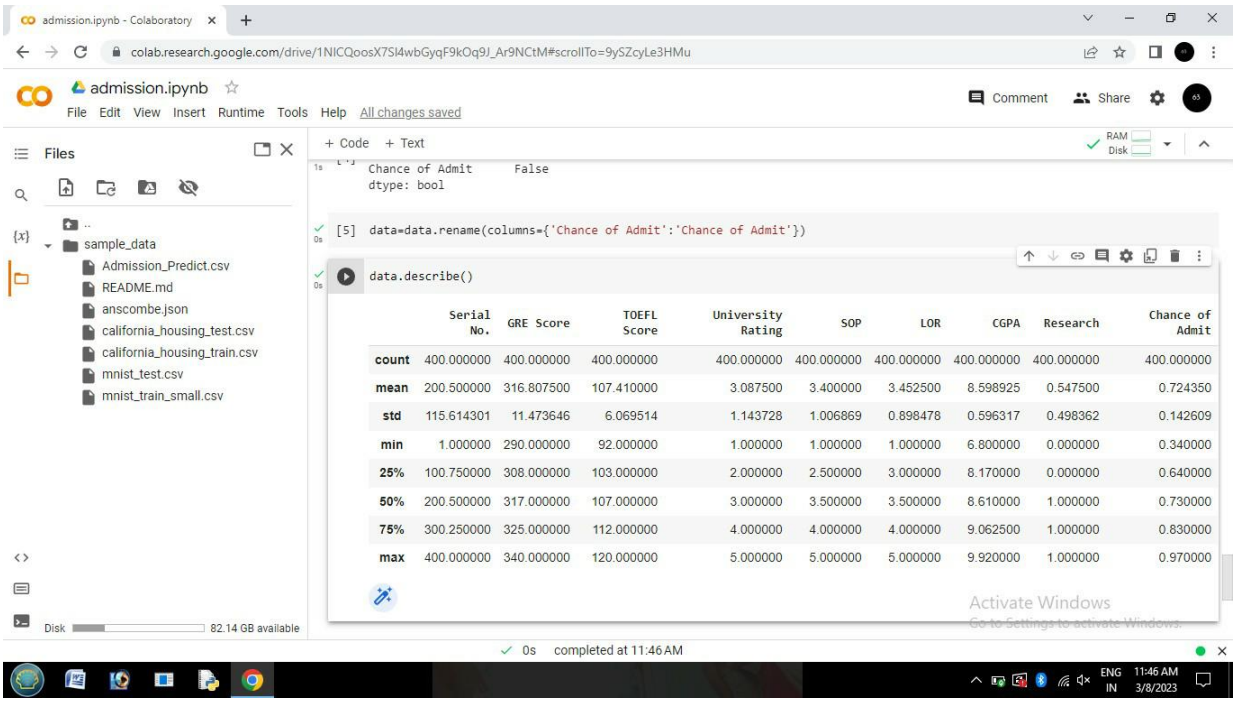




admission.ipynb - Colaboratory
colab.research.google.com/drive/1NICQoosX7SI4wbGyqF9kOq9J_Ar9NCtM#scrollTo=fRMGkKeD0MR2
admission.ipynb
File Edit View Insert Runtime Tools Help Saving...
Comment
Share
Files
sample_data
+ Code + Text
3 University Rating 400 non-null int64
4 SOP 400 non-null float64
5 LOR 400 non-null float64
6 CGPA 400 non-null float64
7 Research 400 non-null int64
8 Chance of Admit 400 non-null float64
dtypes: float64(4), int64(5)
memory usage: 28.2 KB
data.isnull().any()
Serial No. False
GRE Score False
TOEFL Score False
University Rating False
SOP False
LOR False
CGPA False
Research False
Chance of Admit False
dtype: bool
Disk 82.14 GB available
Activate Windows
Go to Settings to activate Windows.
0s completed at 11:33 AM
ENG IN 11:34 AM 3/8/2023

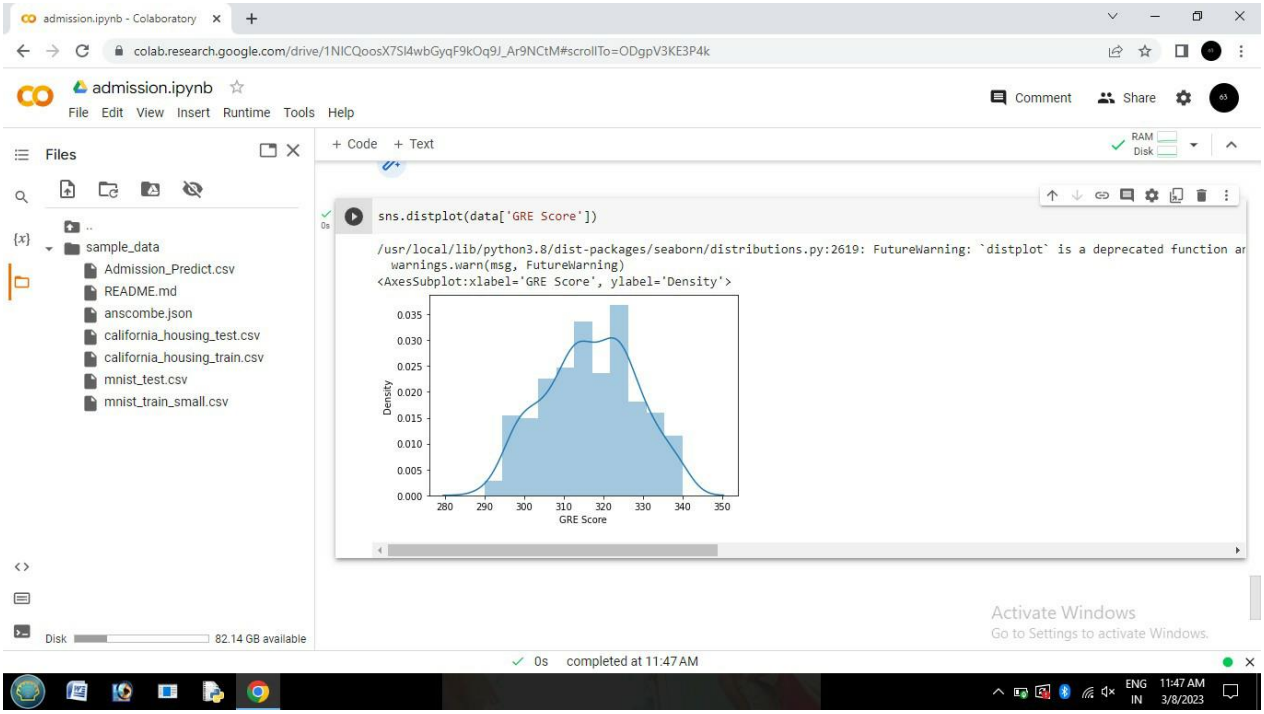

admission.ipynb - Colaboratory
colab.research.google.com/drive/1NICQoosX7SI4wbGyqF9kOq9J_Ar9NCtM#scrollTo=fRMGkKeD0MR2
admission.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
sample_data
+ Code + Text
3 University Rating 400 non-null int64
4 SOP 400 non-null float64
5 LOR 400 non-null float64
6 CGPA 400 non-null float64
7 Research 400 non-null int64
8 Chance of Admit 400 non-null float64
dtypes: float64(4), int64(5)
memory usage: 28.2 KB
data.isnull().any()
Serial No. False
GRE Score False
TOEFL Score False
University Rating False
SOP False
LOR False
CGPA False
Research False
Chance of Admit False
dtype: bool
Disk 82.14 GB available
Activate Windows
Go to Settings to activate Windows.
0s completed at 11:33 AM
ENG IN 11:34 AM 3/8/2023

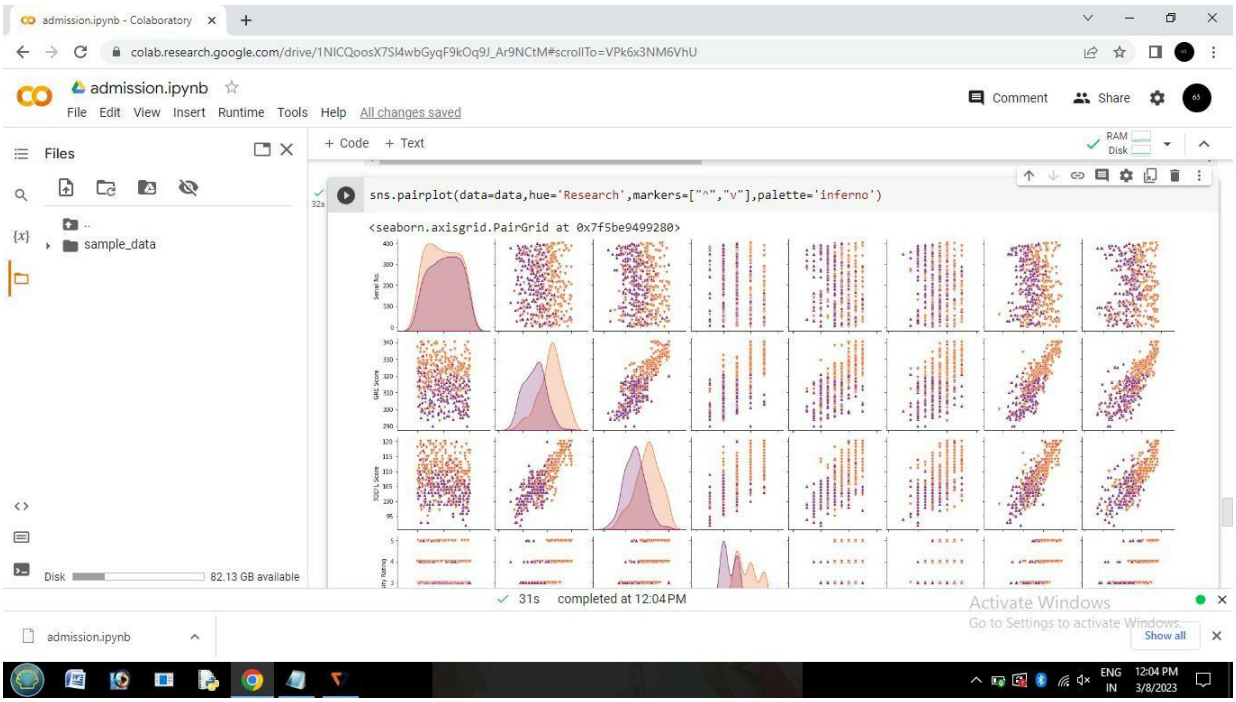


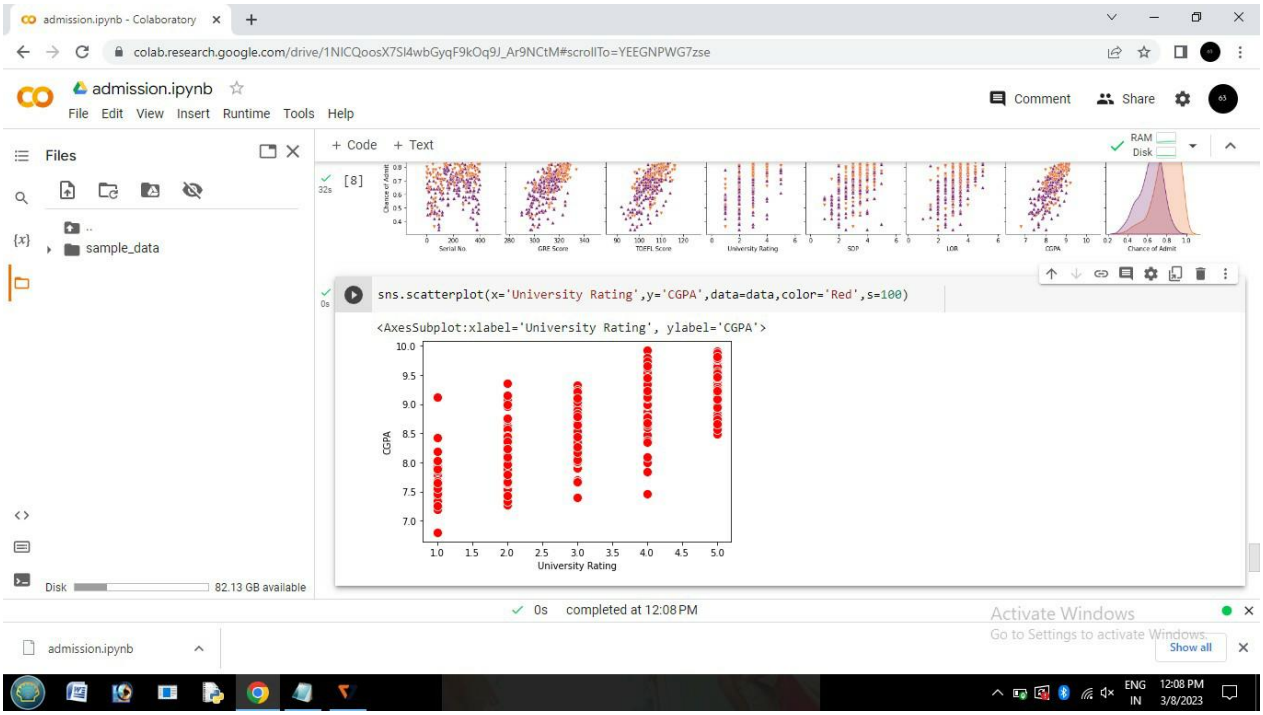

| | Serial No. | GRE Score | TOEFL Score | University Rating | SOP | LOR | CGPA | Research | Chance of Admit |
|---|---|---|---|---|---|---|---|---|---|
| count | 400.000000 | 400.000000 | 400.000000 | 400.000000 | 400.000000 | 400.000000 | 400.000000 | 400.000000 | 400.000000 |
| mean | 200.500000 | 316.807500 | 107.410000 | 3.087500 | 3.400000 | 3.452500 | 8.598925 | 0.547500 | 0.724350 |
| std | 115.614301 | 11.473646 | 6.069514 | 1.143728 | 1.006869 | 0.898478 | 0.596317 | 0.498362 | 0.142609 |
| min | 1.000000 | 290.000000 | 92.000000 | 1.000000 | 1.000000 | 1.000000 | 6.800000 | 0.000000 | 0.340000 |
| 25% | 100.750000 | 308.000000 | 103.000000 | 2.000000 | 2.500000 | 3.000000 | 8.170000 | 0.000000 | 0.640000 |
| 50% | 200.500000 | 317.000000 | 107.000000 | 3.000000 | 3.500000 | 3.500000 | 8.610000 | 1.000000 | 0.730000 |
| 75% | 300.250000 | 325.000000 | 112.000000 | 4.000000 | 4.000000 | 4.000000 | 9.062500 | 1.000000 | 0.830000 |
| max | 400.000000 | 340.000000 | 120.000000 | 5.000000 | 5.000000 | 5.000000 | 9.920000 | 1.000000 | 0.970000 |

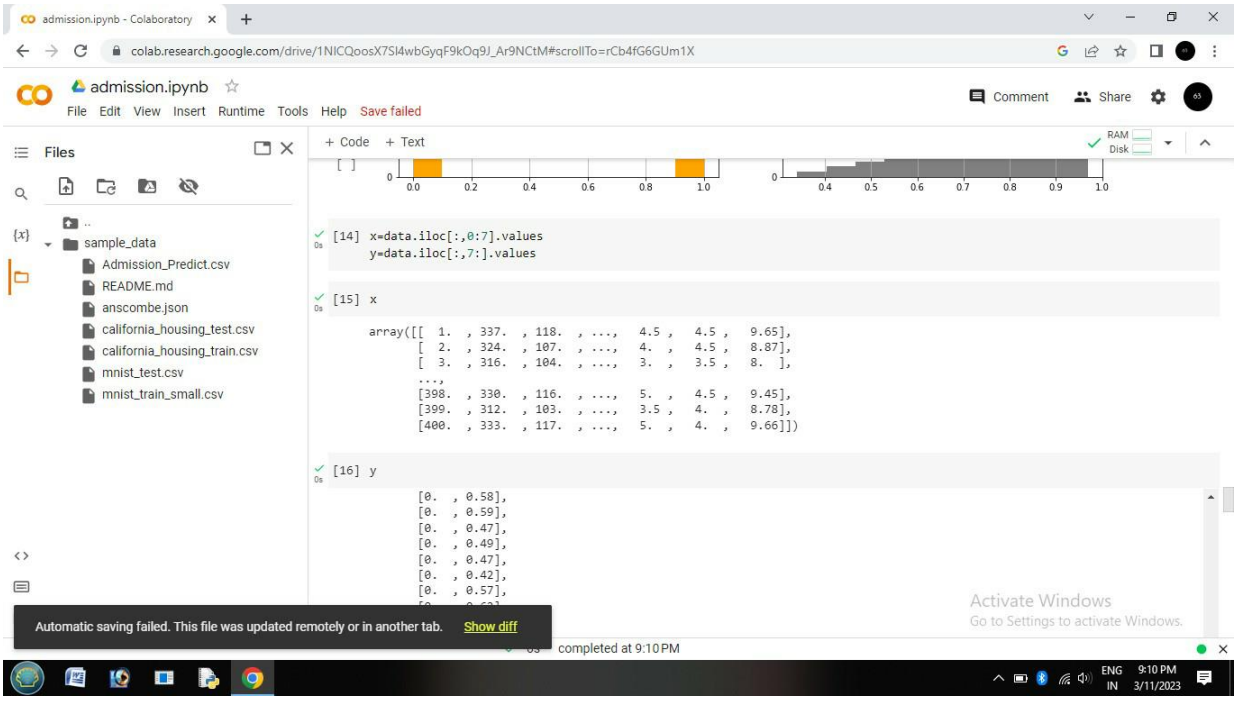


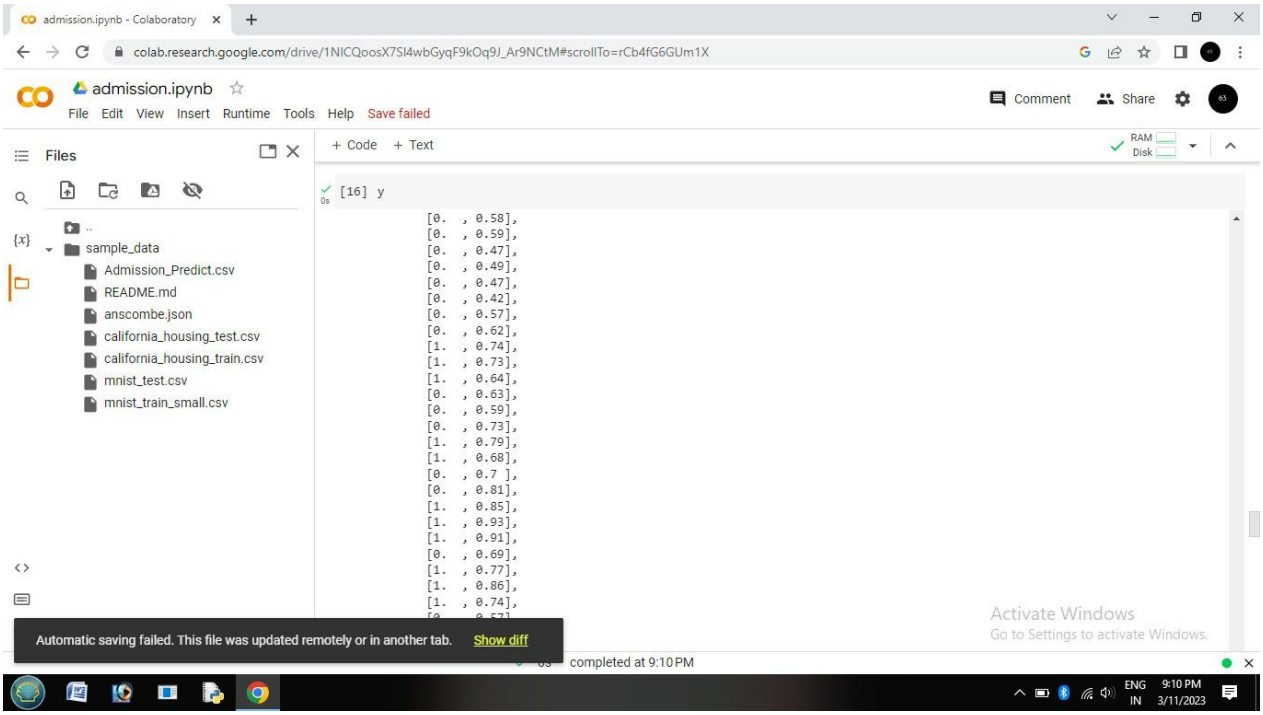

| | Serial No. | GRE Score | TOEFL Score | University Rating | SOP | LOR | CGPA | Research | Chance of Admit |
|---|---|---|---|---|---|---|---|---|---|
| count | 400.000000 | 400.000000 | 400.000000 | 400.000000 | 400.000000 | 400.000000 | 400.000000 | 400.000000 | 400.000000 |
| mean | 200.500000 | 316.807500 | 107.410000 | 3.087500 | 3.400000 | 3.452500 | 8.598925 | 0.547500 | 0.724350 |
| std | 115.614301 | 11.473646 | 6.069514 | 1.143728 | 1.006869 | 0.898478 | 0.596317 | 0.498362 | 0.142609 |
| min | 1.000000 | 290.000000 | 92.000000 | 1.000000 | 1.000000 | 1.000000 | 6.800000 | 0.000000 | 0.340000 |
| 25% | 100.750000 | 308.000000 | 103.000000 | 2.000000 | 2.500000 | 3.000000 | 8.170000 | 0.000000 | 0.640000 |
| 50% | 200.500000 | 317.000000 | 107.000000 | 3.000000 | 3.500000 | 3.500000 | 8.610000 | 1.000000 | 0.730000 |
| 75% | 300.250000 | 325.000000 | 112.000000 | 4.000000 | 4.000000 | 4.000000 | 9.062500 | 1.000000 | 0.830000 |
| max | 400.000000 | 340.000000 | 120.000000 | 5.000000 | 5.000000 | 5.000000 | 9.920000 | 1.000000 | 0.970000 |

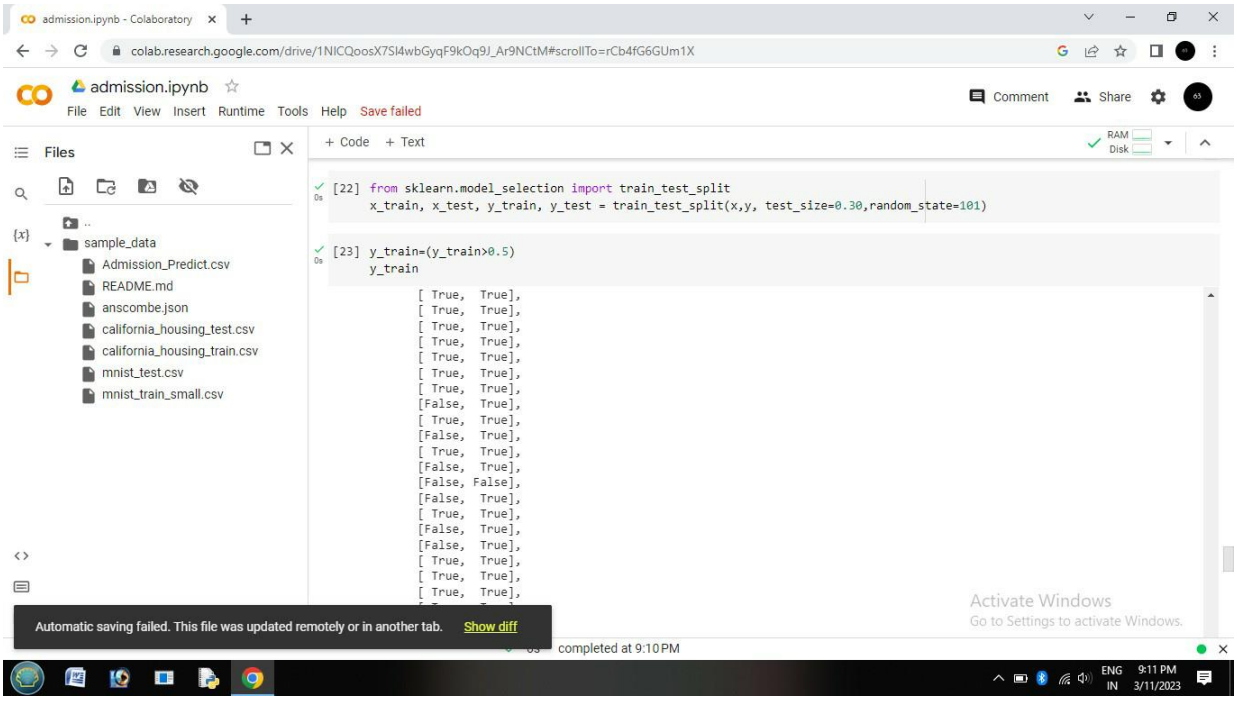

admission.ipynb - Colaboratory
colab.research.google.com/drive/1NICQoosX7SI4wbGyqF9kOq9J_Ar9NCtM#scrollTo=ODgpV3KE3P4k
admission.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
Files
+ Code + Text
sample_data
Admission_Predict.csv
README.md
anscombe.json
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
sns.distplot(data['GRE Score'])
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function ar
warnings.warn(msg, FutureWarning)
<AxesSubplot:xlabel='GRE Score', ylabel='Density'>
Density
GRE Score
Disk 82.14 GB available
Activate Windows
Go to Settings to activate Windows.
0s completed at 11:47 AM
ENG IN 11:47 AM 3/8/2023

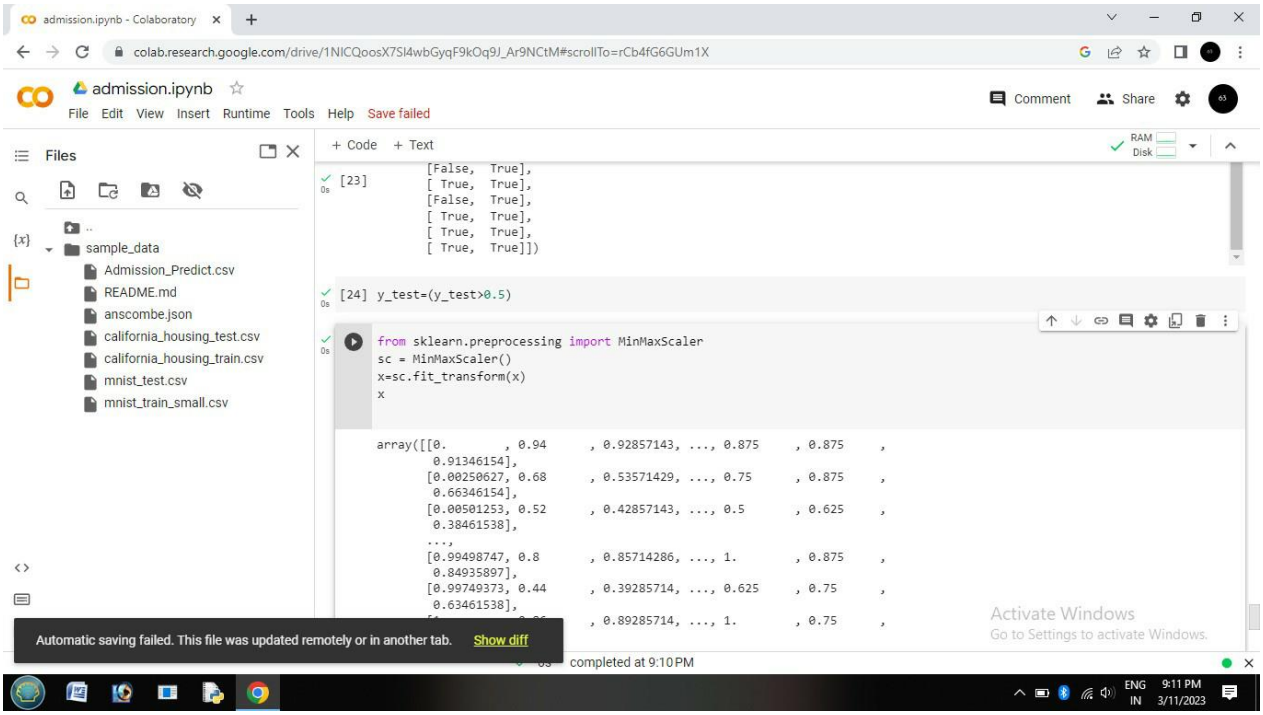

admission.ipynb - Colaboratory
colab.research.google.com/drive/1NICQoosX7SI4wbGyqF9kOq9J_Ar9NCtM#scrollTo=VPk6x3NM6VhU
admission.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code + Text
sample_data
sns.pairplot(data=data,hue='Research',markers=["^","v"],palette='inferno')
<seaborn.axisgrid.PairGrid at 0x7f5be9499280>
Disk 82.13 GB available
31s completed at 12:04 PM
Activate Windows
Go to Settings to activate Windows.
admission.ipynb
Show all
ENG IN 12:04 PM 3/8/2023

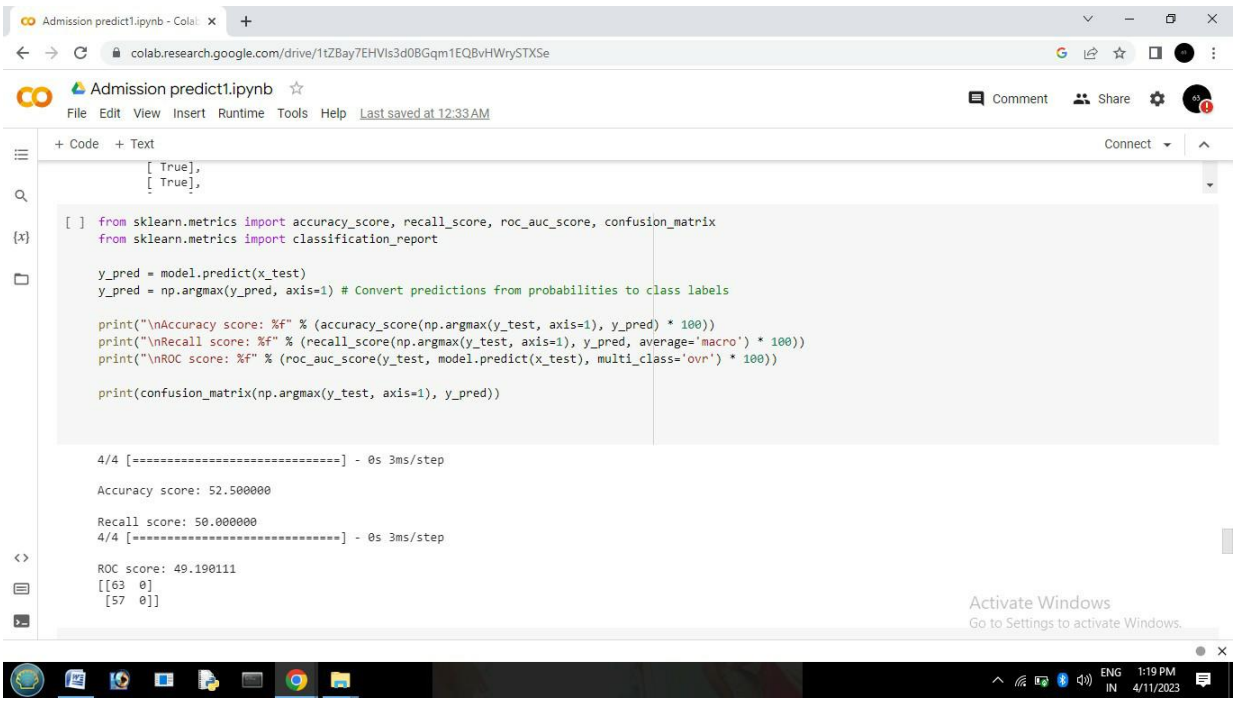

admission.ipynb - Colaboratory
colab.research.google.com/drive/1NICQoosX7Sl4wbGyqF9kOq9J_Ar9NCtM#scrollTo=YEEGNPWG7zse
admission.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
Files
sample_data
+ Code + Text
[8]
sns.scatterplot(x='University Rating',y='CGPA',data=data,color='Red',s=100)
<AxesSubplot:xlabel='University Rating', ylabel='CGPA'>
CGPA
University Rating
Disk 82.13 GB available
0s completed at 12:08 PM
Activate Windows
Go to Settings to activate Windows.
admission.ipynb
Show all
ENG IN 12:08 PM 3/8/2023

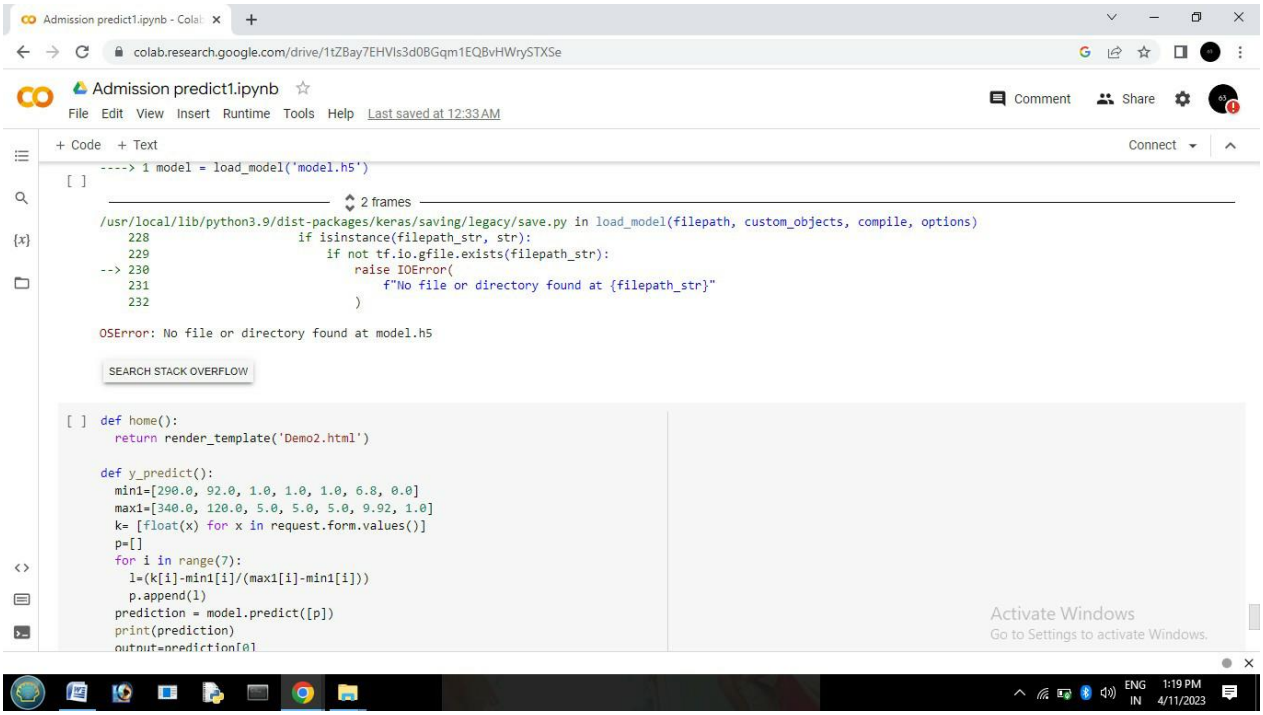

admission.ipynb - Colaboratory
colab.research.google.com/drive/1NICQoosX7Sl4wbGyqF9kOq9J_Ar9NCtM#scrollTo=rCb4fG6GUm1X
admission.ipynb
File Edit View Insert Runtime Tools Help Save failed
Comment
Share
Files
sample_data
Admission_Predict.csv
README.md
anscombe.json
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
+ Code + Text
[14] x=data.iloc[:,0:7].values
y=data.iloc[:,7:].values
[15] x
array([[ 1. , 337. , 118. , ..., 4.5 , 4.5 , 9.65],
[ 2. , 324. , 107. , ..., 4. , 4.5 , 8.87],
[ 3. , 316. , 104. , ..., 3. , 3.5 , 8. ],
...,
[398. , 330. , 116. , ..., 5. , 4.5 , 9.45],
[399. , 312. , 103. , ..., 3.5 , 4. , 8.78],
[400. , 333. , 117. , ..., 5. , 4. , 9.66]])
[16] y
[0. , 0.58],
[0. , 0.59],
[0. , 0.47],
[0. , 0.49],
[0. , 0.47],
[0. , 0.42],
[0. , 0.57],
Automatic saving failed. This file was updated remotely or in another tab. Show diff
completed at 9:10 PM
Activate Windows
Go to Settings to activate Windows.
ENG IN 9:10 PM 3/11/2023


admission.ipynb - Colaboratory
colab.research.google.com/drive/1NlCQoosX7Sl4wbGyqF9kOq9J_Ar9NCtM#scrollTo=rCb4fG6GUm1X
admission.ipynb
File Edit View Insert Runtime Tools Help Save failed
Comment
Share
Files
+ Code + Text
RAM
Disk
[16] y
[0. , 0.58],
[0. , 0.59],
[0. , 0.47],
[0. , 0.49],
[0. , 0.47],
[0. , 0.42],
[0. , 0.57],
[0. , 0.62],
[1. , 0.74],
[1. , 0.73],
[1. , 0.64],
[0. , 0.63],
[0. , 0.59],
[0. , 0.73],
[1. , 0.79],
[1. , 0.68],
[0. , 0.7 ],
[0. , 0.81],
[1. , 0.85],
[1. , 0.93],
[1. , 0.91],
[0. , 0.69],
[1. , 0.77],
[1. , 0.86],
[1. , 0.74],
..
sample_data
Admission_Predict.csv
README.md
anscombe.json
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
Automatic saving failed. This file was updated remotely or in another tab.
Show diff
Activate Windows
Go to Settings to activate Windows.
completed at 9:10 PM
ENG
IN
9:10 PM
3/11/2023


admission.ipynb - Colaboratory
colab.research.google.com/drive/1NlCQoosX7Sl4wbGyqF9kOq9J_Ar9NCtM#scrollTo=rCb4fG6GUm1X
admission.ipynb
File Edit View Insert Runtime Tools Help Save failed
Comment
Share
Files
+ Code + Text
RAM
Disk
[22] 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)
[23] y_train=(y_train>0.5)
y_train
[ True, True],
[ True, True],
[ True, True],
[ True, True],
[ True, True],
[ True, True],
[ True, True],
[False, True],
[ True, True],
[False, True],
[ True, True],
[False, True],
[False, False],
[False, True],
[ True, True],
[False, True],
[False, True],
[ True, True],
[ True, True],
[ True, True],
..
sample_data
Admission_Predict.csv
README.md
anscombe.json
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
Automatic saving failed. This file was updated remotely or in another tab.
Show diff
Activate Windows
Go to Settings to activate Windows.
completed at 9:10 PM
ENG
IN
9:11 PM
3/11/2023

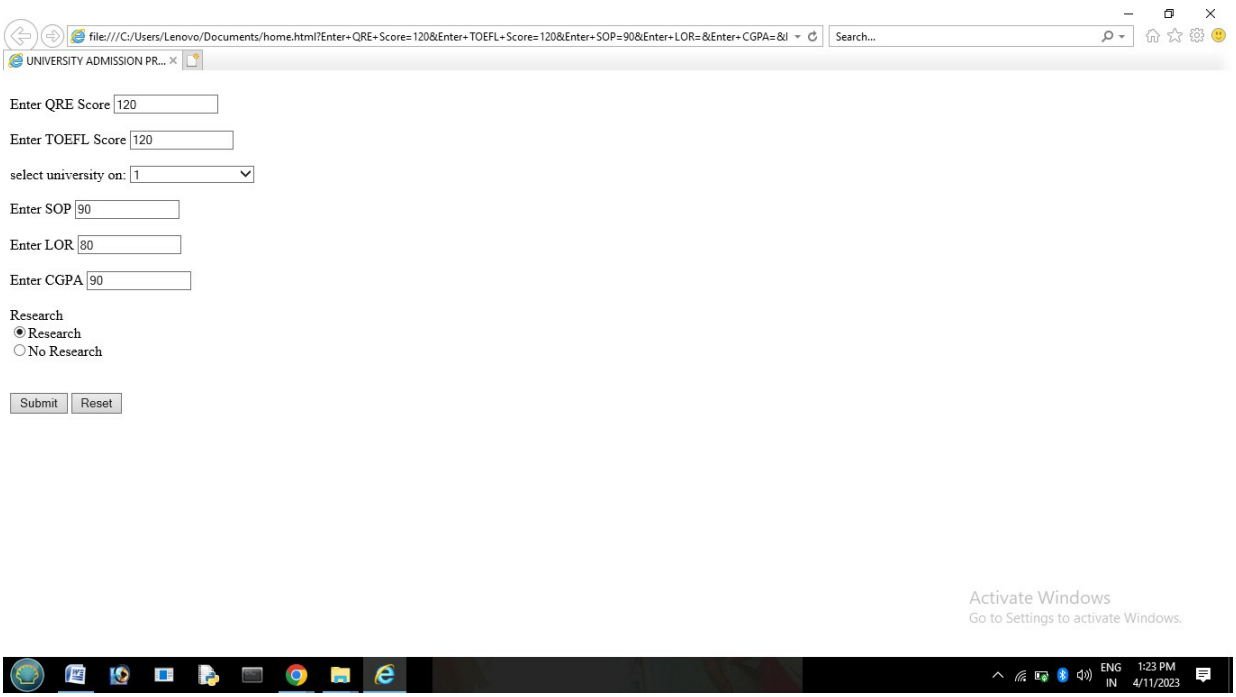

admission.ipynb - Colaboratory
colab.research.google.com/drive/1NICQoosX7Sl4wbGyqF9kOq9J_Ar9NCtM#scrollTo=rCb4fG6GUm1X
admission.ipynb
File Edit View Insert Runtime Tools Help Save failed
Comment
Share
Files
sample_data
Admission_Predict.csv
README.md
anscombe.json
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
+ Code + Text
[23] [False, True],
[ True, True],
[False, True],
[ True, True],
[ True, True],
[ True, True]])
[24] y_test=(y_test>0.5)
from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler()
x=sc.fit_transform(x)
x
array([[0. , 0.94 , 0.92857143, ..., 0.875 , 0.875 ,
0.91346154],
[0.00250627, 0.68 , 0.53571429, ..., 0.75 , 0.875 ,
0.66346154],
[0.00501253, 0.52 , 0.42857143, ..., 0.5 , 0.625 ,
0.38461538],
...,
[0.99498747, 0.8 , 0.85714286, ..., 1. , 0.875 ,
0.84935897],
[0.99749373, 0.44 , 0.39285714, ..., 0.625 , 0.75 ,
0.63461538],
, 0.89285714, ..., 1. , 0.75 ,
Automatic saving failed. This file was updated remotely or in another tab. Show diff
completed at 9:10 PM
Activate Windows
Go to Settings to activate Windows.
ENG IN 9:11 PM 3/11/2023


Admission predict1.ipynb - Colab
colab.research.google.com/drive/1tZBay7EHVIs3d0BGqm1EQBvHWrySTXSe
Admission predict1.ipynb
File Edit View Insert Runtime Tools Help Last saved at 12:33 AM
Comment
Share
+ Code + Text
Connect
[ True],
[ True],
from sklearn.metrics import accuracy_score, recall_score, roc_auc_score, confusion_matrix
from sklearn.metrics import classification_report
y_pred = model.predict(x_test)
y_pred = np.argmax(y_pred, axis=1) # Convert predictions from probabilities to class labels
print("\nAccuracy score: %f" % (accuracy_score(np.argmax(y_test, axis=1), y_pred) * 100))
print("\nRecall score: %f" % (recall_score(np.argmax(y_test, axis=1), y_pred, average='macro') * 100))
print("\nROC score: %f" % (roc_auc_score(y_test, model.predict(x_test), multi_class='ovr') * 100))
print(confusion_matrix(np.argmax(y_test, axis=1), y_pred))
4/4 [==============================] - 0s 3ms/step
Accuracy score: 52.500000
Recall score: 50.000000
4/4 [==============================] - 0s 3ms/step
ROC score: 49.190111
[[63 0]
[57 0]]
Activate Windows
Go to Settings to activate Windows.
ENG IN 1:19 PM 4/11/2023




# 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.

**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.

# 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_transfor
m(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.summar**

**y()**

**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_t
est) 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=['POS
T']) 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 o
if__name__== "__main__": app.run(debug=False)
```