Government of Tamil Nadu

Naan Muthalvan - Project-Based Experiential Learning

Identifying Patterns And Trends In Campus Placement Data Using Machine Learning

Submitted by

Team ID: NM2023TMID22797

V.MENAKA-(20326ER056)

J.NITHYASRI-(20326ER057)

K.NITHYASRI-(20326ER058)

S.PRIYA-(20326ER059)

Under the guidance of

Mrs. P. SANGEETHA M.Sc., M.Phil.,

Guest Lecturer

PG and Research Department of Computer Science

M.V.MUTHIAH GOVERNMENT ARTS COLLEGE FOR WOMEN

(Affiliated To Mother Teresa Women's University, Kodaikanal)

Reaccredited with "A" Grade by NAAC

DINDIGUL-624001.

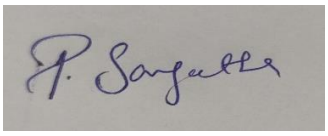
APRIL-2023

M.V.MUTHIAH GOVERNMENT ARTS COLLEGE 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 "Identifying Patterns And Trends In Campus Placement Data using Machine Learning"done by **Ms.V.MENAKA-(20326ER056),Ms.J.NITHYASRI-(20326ER057),Ms.K.NITHYASRI-(20326ER058),and Ms.S.PRIYA-(20326ER059).** 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 11.04.2023

IDENTIFYING PATTERNS AND TRENDS IN CAMPUS PLACEMENT DATA USING MACHINE LEARNING.

1.INTRODUCTION:

1.1 Overview:

Campus recruitment is a strategy for sourcing, engaging and hiring young talent for internship and entry-level positions. College recruiting is typically a tactic for medium- to large-sized companies with high-volume recruiting needs, but can range from small efforts (like working with university career centers to source potential candidates) to large-scale operations (like visiting a wide array of colleges and attending recruiting events throughout the spring and fall semester).

Campus recruitment often involves working with university career services centers and attending career fairs to meet in-person with college students and recent graduates.Our solution revolves around the placement season of a Business School in India.

Where it has various factors on candidates getting hired such as work experience,exam percentage etc., Finally it contains the status of recruitment and remuneration details. We will be using algorithms such as KNN, SVM and ANN.

We will train and test the data with these algorithms. From this the best model is selected and saved in .pkl format. We will be doing flask integration and IBM deployment.

1.2 Purpose:

Identifying patterns and trends in campus placement data using machine learning is a technique that involves the use of advanced analytical algorithms and statistical models to analyze and interpret the data related to campus placement.

This technique is useful for identifying the patterns and trends in the data, which can provide valuable insights into the factors that influence the placement of students in various companies.

Machine learning algorithm can be used to analyze the data related to students academic performance,their technical skills, and their performance in interviews.

These algorithms can also be used to identify the correlation between various factors and the success rate of students in securing placements.

The identification of patterns and trends in campus placement data can be highly beneficial for students, placement officers, and academic institutions.

2.PROBLEM DEFINITION & Design Thinking:

2.1 Empathy Map

Empathy for Identifying Patterns and Trends in Campus Placement Data using Machine Learning

Empathy in this case can refer to the ability of the machine learning algorithm to understand the context and nuances of the data it is analyzing. This includes understanding the factors that may impact the placement of students, such as their academic performance, background, and the current job market.

To develop empathy in a machine learning algorithm, it is important to train the model on a diverse set of data that includes a wide range of backgrounds and experiences. This will help the algorithm to identify patterns and trends that may not be immediately obvious, and to recognize the impact of different factors on placement outcomes.

In addition to training the algorithm on diverse data, it is also important to include feedback mechanisms that allow the algorithm to learn from its mistakes and adjust its approach over time. This can help the algorithm to continually refine its understanding of the data and improve its ability to identify meaningful patterns and trends.

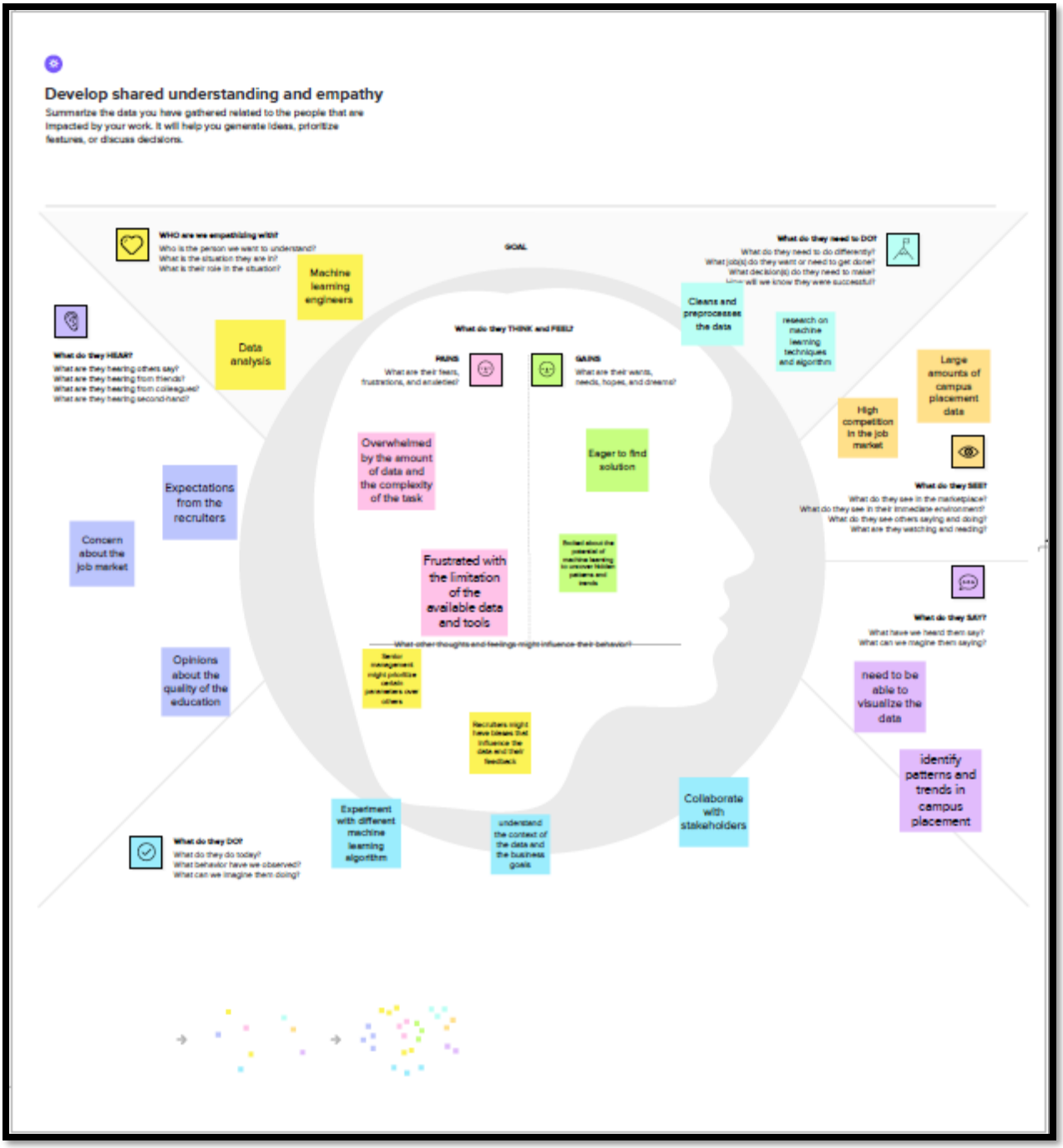
Template

Empathy map canvas

Empathy map canvas For Identifying patterns and trends in campus placement Data

Originally created by Dave Gray at

XPLANE®


Develop shared understanding and empathy
Summarize the data you have gathered related to the people that are impacted by your work. It will help you generate ideas, prioritize features, or discuss decisions.
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?
Machine learning engineers
Cleans and preprocesses the data
research on machine learning techniques and algorithm
Large amounts of campus placement data
Data analysis
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?
High competition in the job market
Overwhelmed by the amount of data and the complexity of the task
Eager to find solution
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 are they watching and reading?
Expectations from the recruiters
Concern about the job market
Frustrated with the limitation of the available data and tools
Excited about the potential of machine learning to uncover hidden patterns and trends
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?
Opinions about the quality of the education
Senior management might prioritize certain parameters over others
need to be able to visualize the data
Recruiters might have biases that influence the data and their feedback
identify patterns and trends in campus placement
Collaborate with stakeholders
Experiment with different machine learning algorithm
understand the context of the data and the business goals
What do they DO?
What do they do today?
What behavior have we observed?
What can we imagine them doing?

2.2 Ideation & Brainstorming Map

Brainstorm Map for Identifying Patterns and Trends in Campus Placement Data using Machine Learning

Here is a brainstorm map that outlines some possible strategies for identifying patterns and trends in campus placement data using machine learning:

Data preprocessing: Clean and normalize the data by removing duplicates, handling missing values, and transforming variables to be consistent across the dataset.

Feature selection: Identify the most important features that impact placement outcomes, such as academic performance, background, and job market trends.

Dimensionality reduction: Reduce the dimensionality of the data by using techniques such as principal component analysis (PCA) or t-distributed stochastic neighbor embedding (t-SNE) to visualize the data in a lower-dimensional space.

Algorithm selection: Select the most appropriate machine learning algorithm for the task at hand, such as decision trees, random forests, or neural networks.

Model training: Train the model on a diverse set of data that includes a wide range of backgrounds and experiences, and use techniques such as cross-validation to ensure that the model is generalizable to new data.

Model evaluation: Evaluate the performance of the model using metrics such as accuracy, precision, recall, and F1 score, and use techniques such as confusion matrices and ROC curves to visualize the results.

Interpretation: Interpret the results of the model by analyzing the most important features and identifying the key factors that impact placement outcomes.

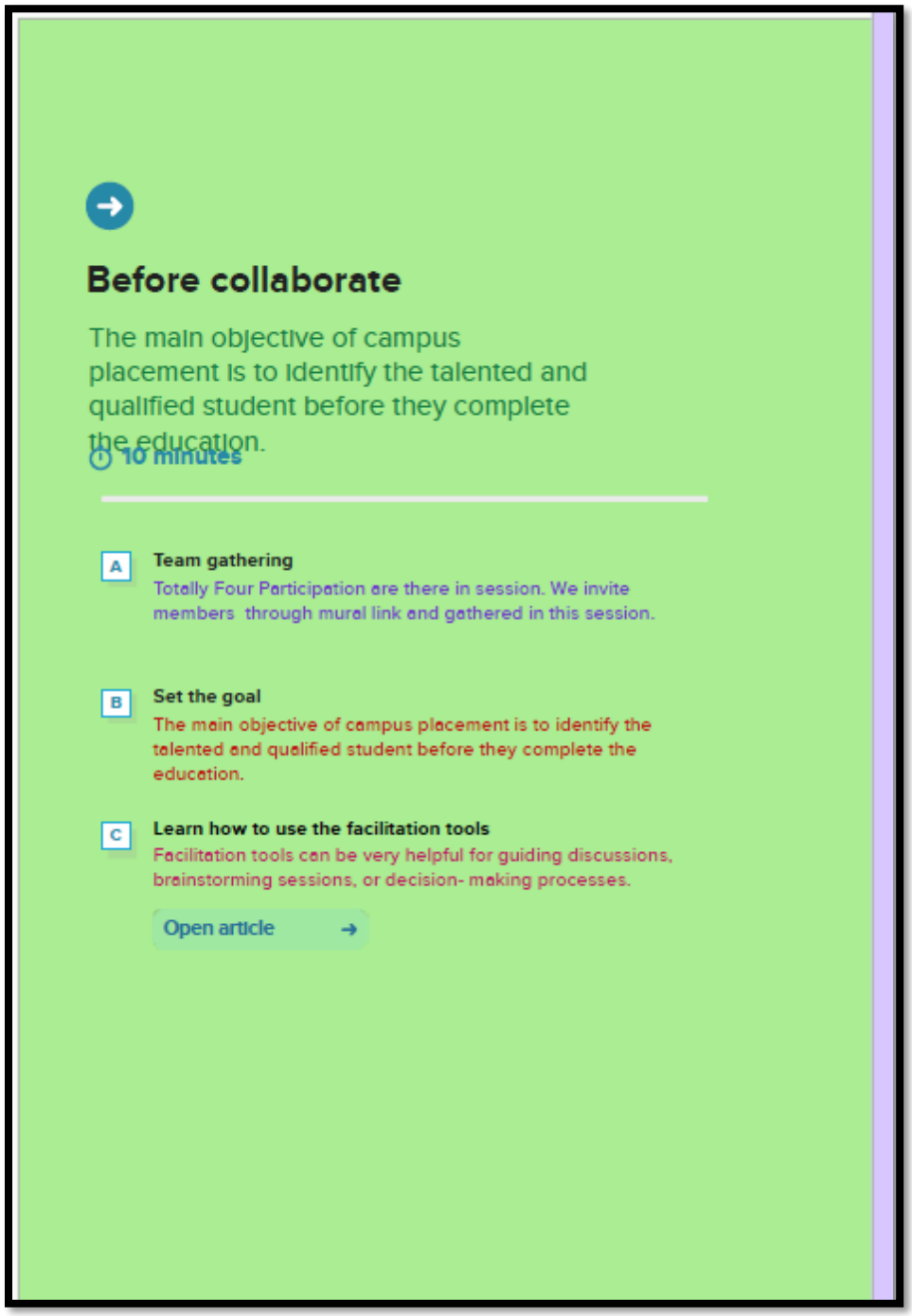
Visualization: Visualize the results of the model using techniques such as heatmaps, scatter plots, and bar charts to identify patterns and trends in the data.

Template

Brainstorm & idea prioritization

Identifying Patterns and Trends in Campus placement Data using Machine learning

10 minutes to prepare
1 hour to collaborate
2-8 people recommended


Before collaborate
The main objective of campus placement is to identify the talented and qualified student before they complete the education.
10 minutes
A
Team gathering
Totally Four Participation are there in session. We invite members through mural link and gathered in this session.
B
Set the goal
The main objective of campus placement is to identify the talented and qualified student before they complete the education.
C
Learn how to use the facilitation tools
Facilitation tools can be very helpful for guiding discussions, brainstorming sessions, or decision- making processes.
Open article

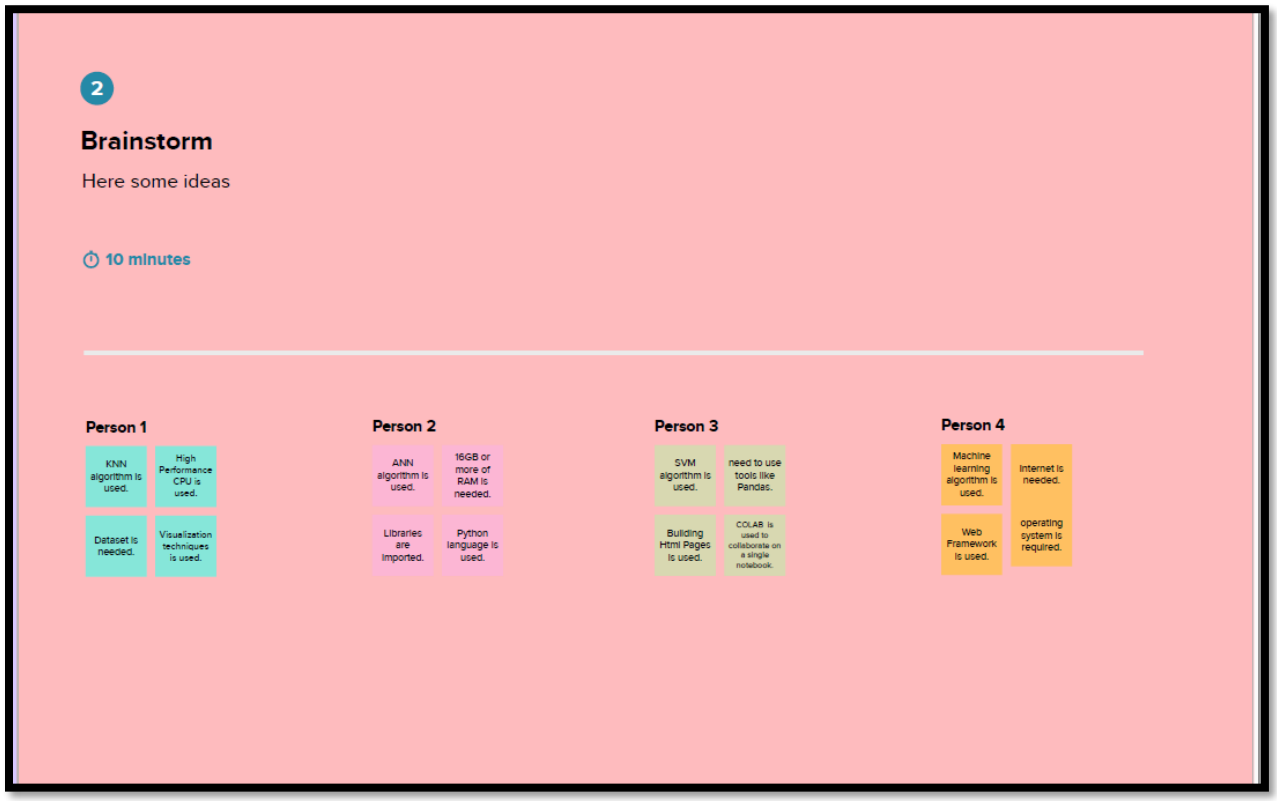
1

problem statement

5 minutes

1)The Main objective of Campus placement is to identify the talented and qualified student before they complete the education.
2)This Project is used to Career opportunities for student in reputed corporate companies.
3)Campus recruitment is a strategy for sourcing, engaging and hiring young talent for internship and entry level position.
4)Campus recruitment often involves working with university career services centers and attending career fairs to meet in person with college students and recent graduates.

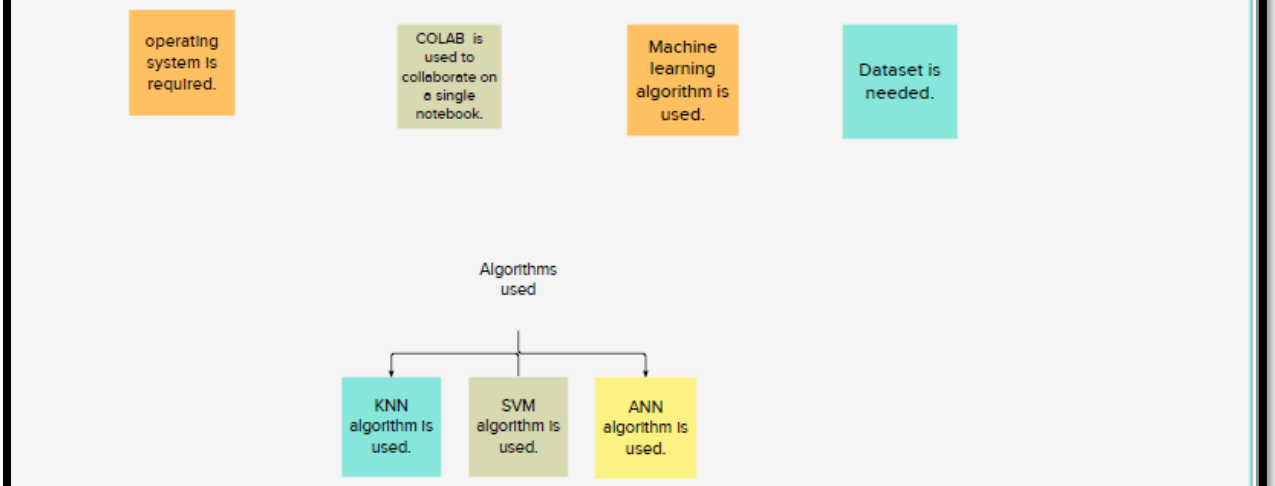
5)This Prediction uses a Machine learning algorithm to gives the result.

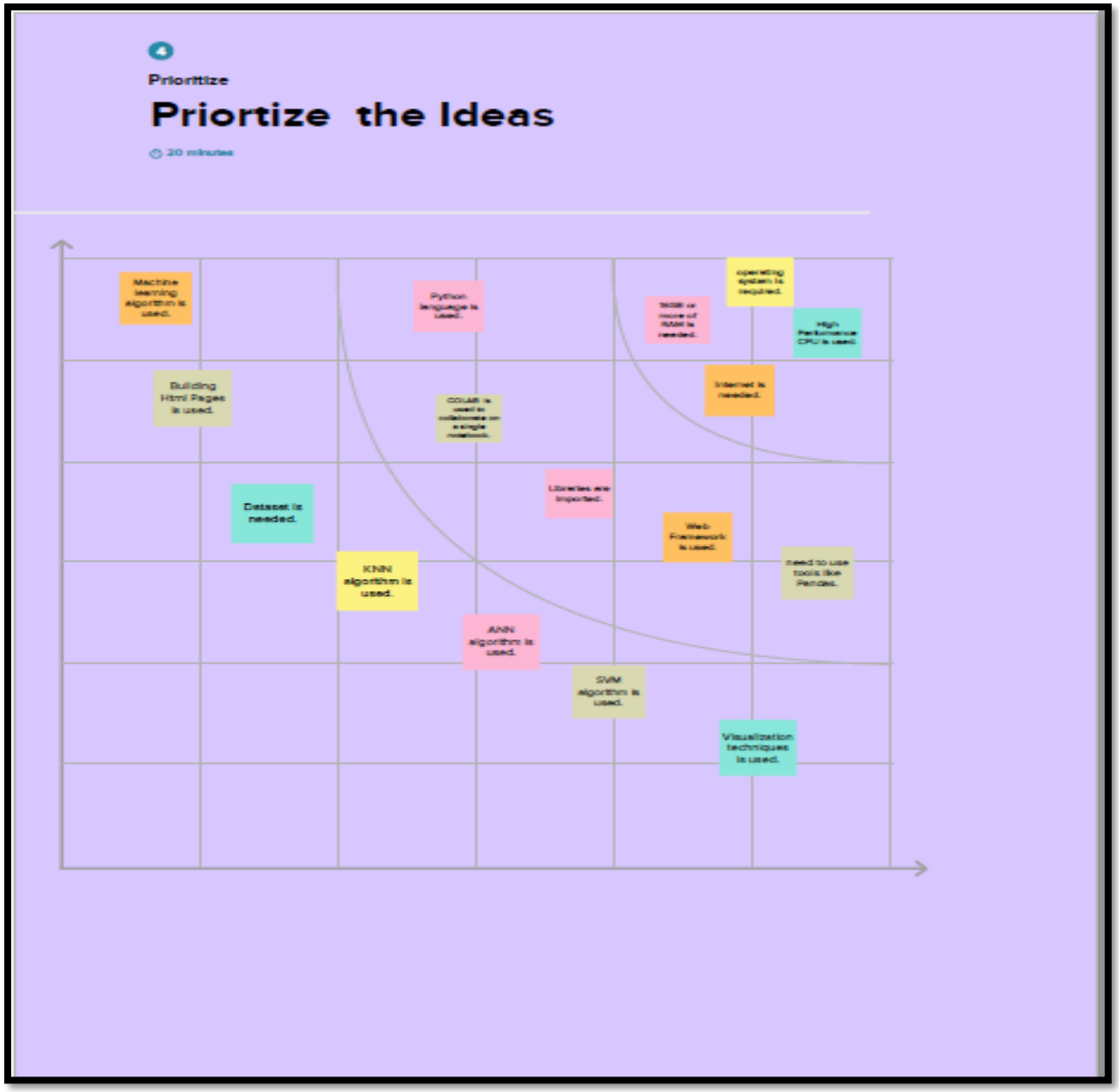

2
Brainstorm
Here some ideas
10 minutes
Person 1
KNN algorithm is used.
High Performance CPU is used.
Dataset is needed.
Visualization techniques is used.
Person 2
ANN algorithm is used.
16GB or more of RAM is needed.
Libraries are imported.
Python language is used.
Person 3
SVM algorithm is used.
need to use tools like Pandas.
Building Html Pages is used.
COLAB is used to collaborate on a single notebook.
Person 4
Machine learning algorithm is used.
Internet is needed.
Web Framework is used.
operating system is required.

3

Group ideas

1)Operating system is required.
2)COLAB is used to collaborate on a single notebook.
3)Machine learning algorithm is used.
4)Dataset is needed.
20 minutes
5)KNN,ANN,SVM algorithm is used.



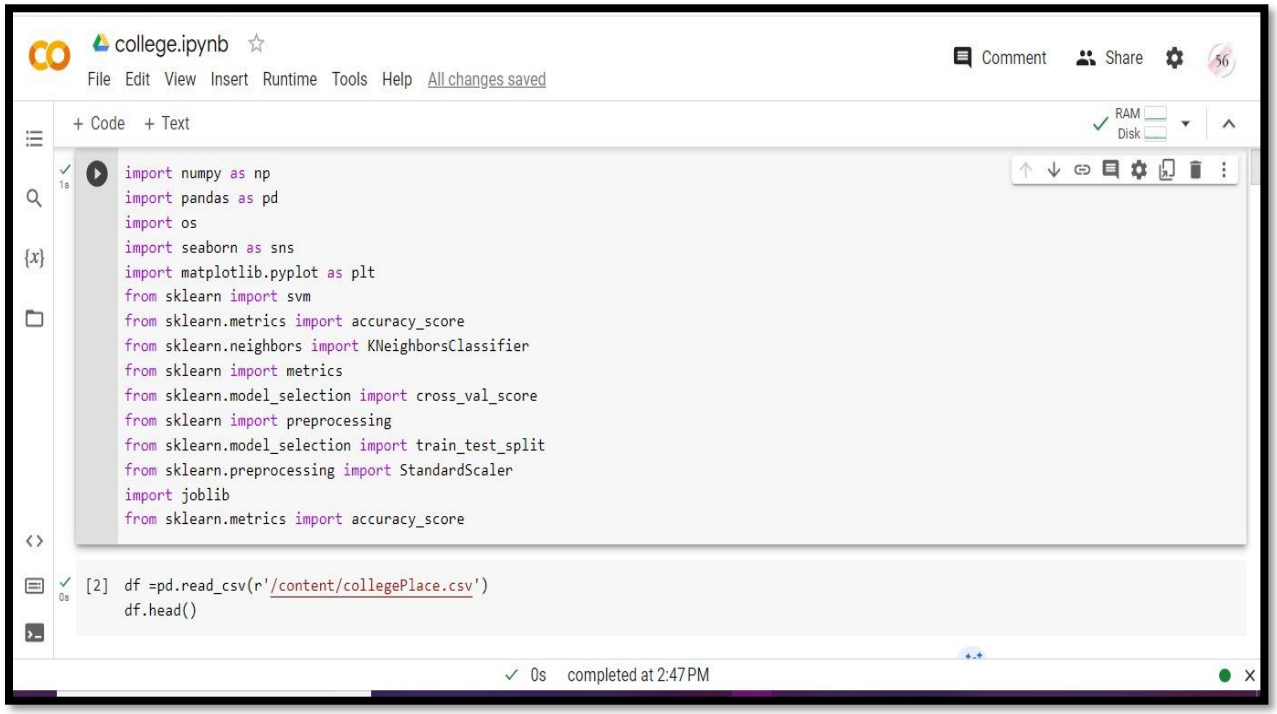

4
Prioritize
Priortize the Ideas
20 minutes
Machine learning algorithm is used.
Python language is used.
operating system is required.
High Performance CPU is used.
Building Html Pages is used.
Internet is needed.
Dataset is needed.
Libraries are imported.
Web Framework is used.
KNN algorithm is used.
ANN algorithm is used.
SVM algorithm is used.
Visualization techniques is used.


After collaborate
We can export the mural as pdf to share .It is helpful to getting information.

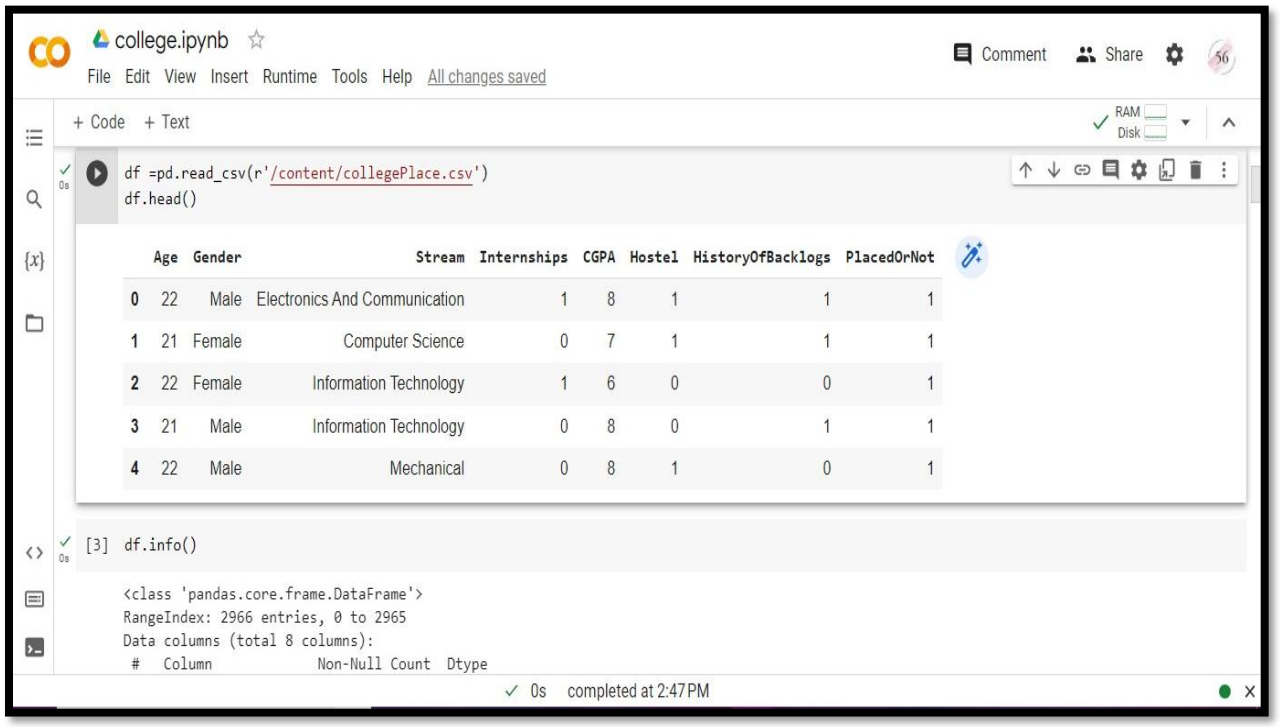
3.Screen Layout:

Data collection & Preparation:

Importing the libraries

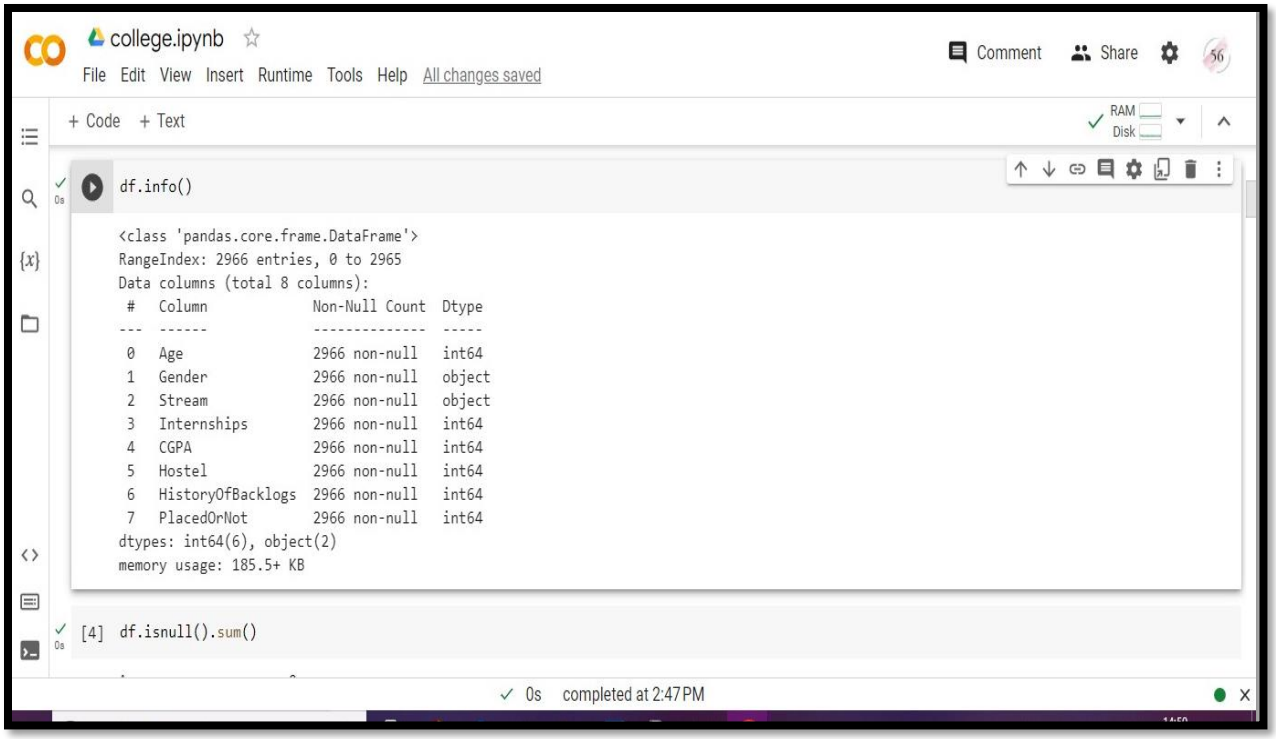


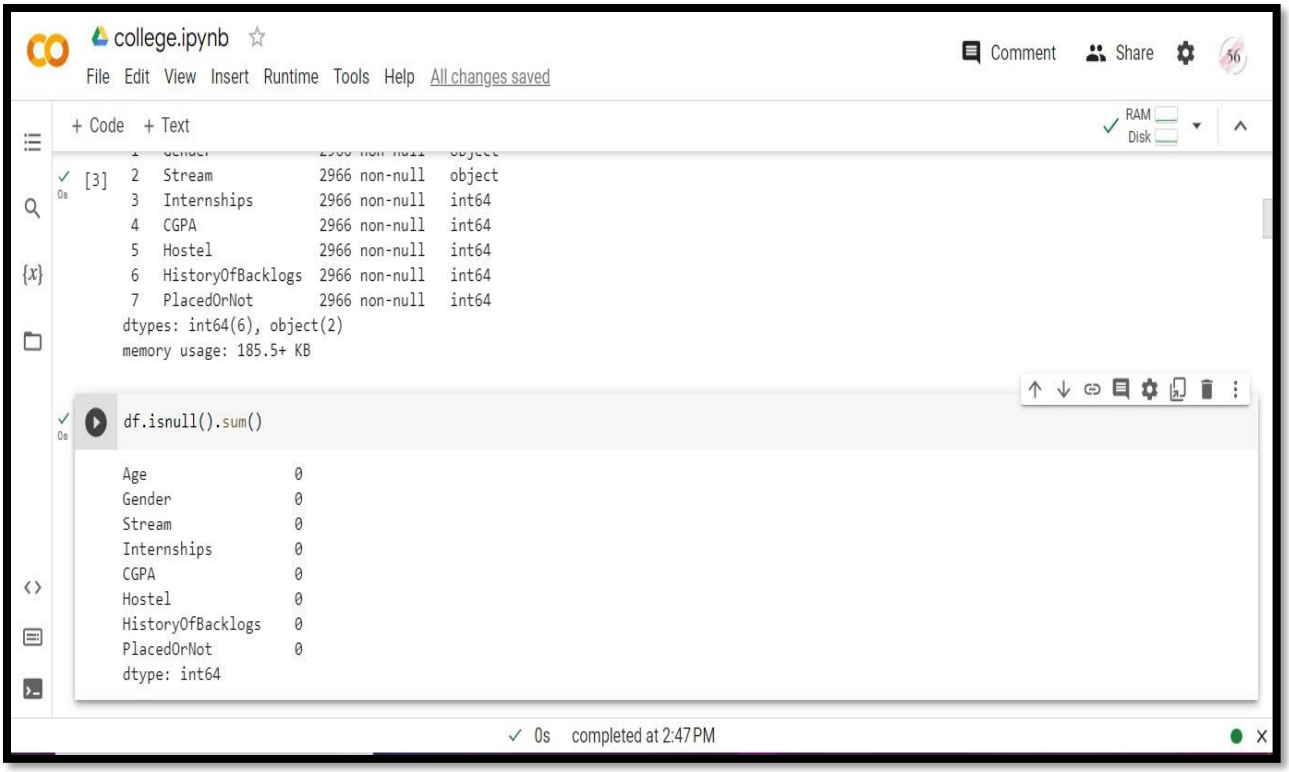
Read the Dataset:



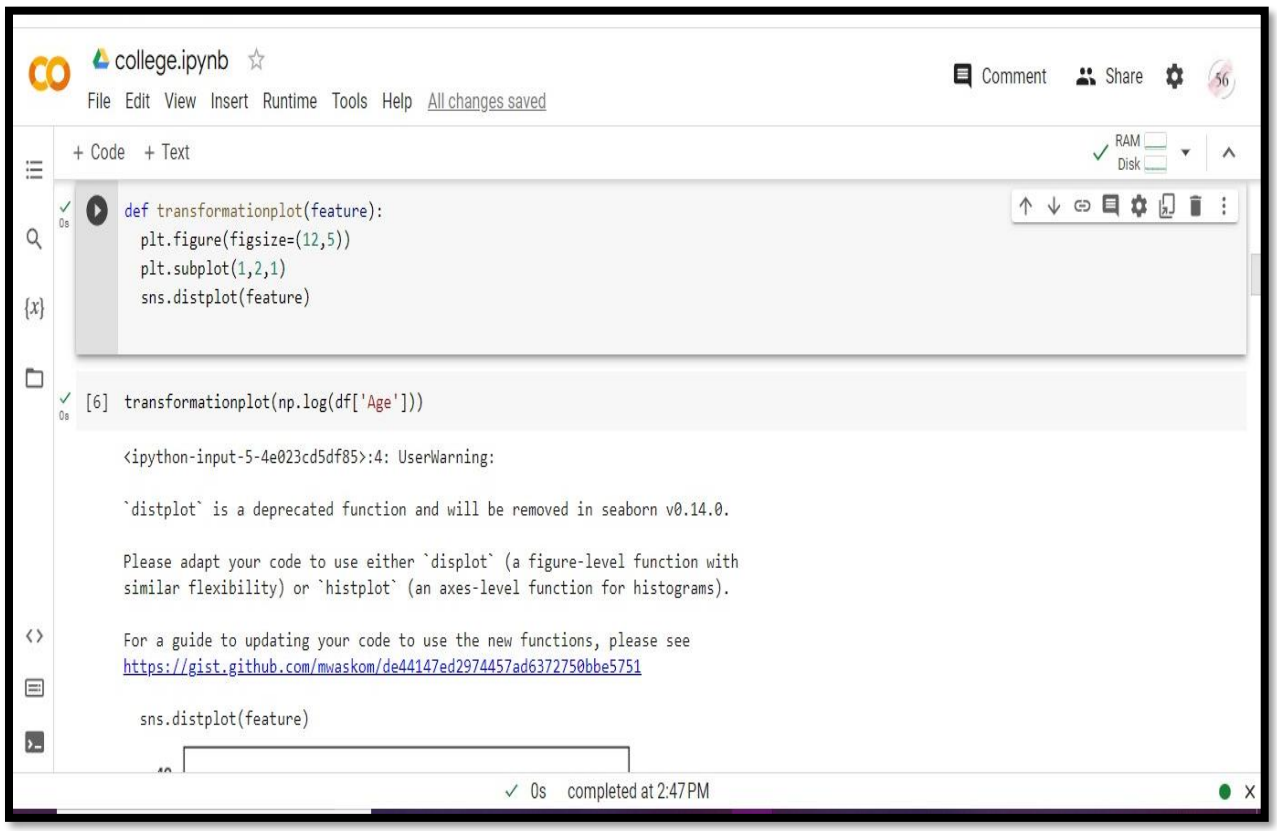
Data Preparation:

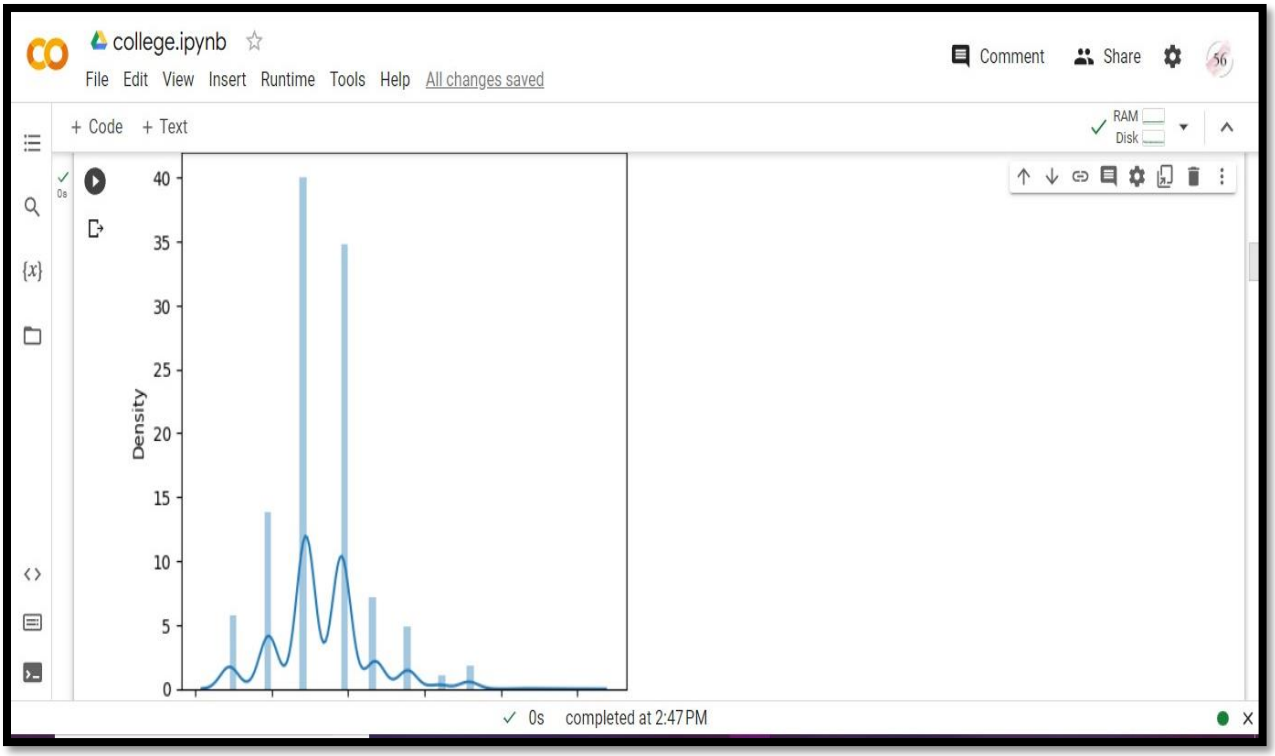
Handling missing value:

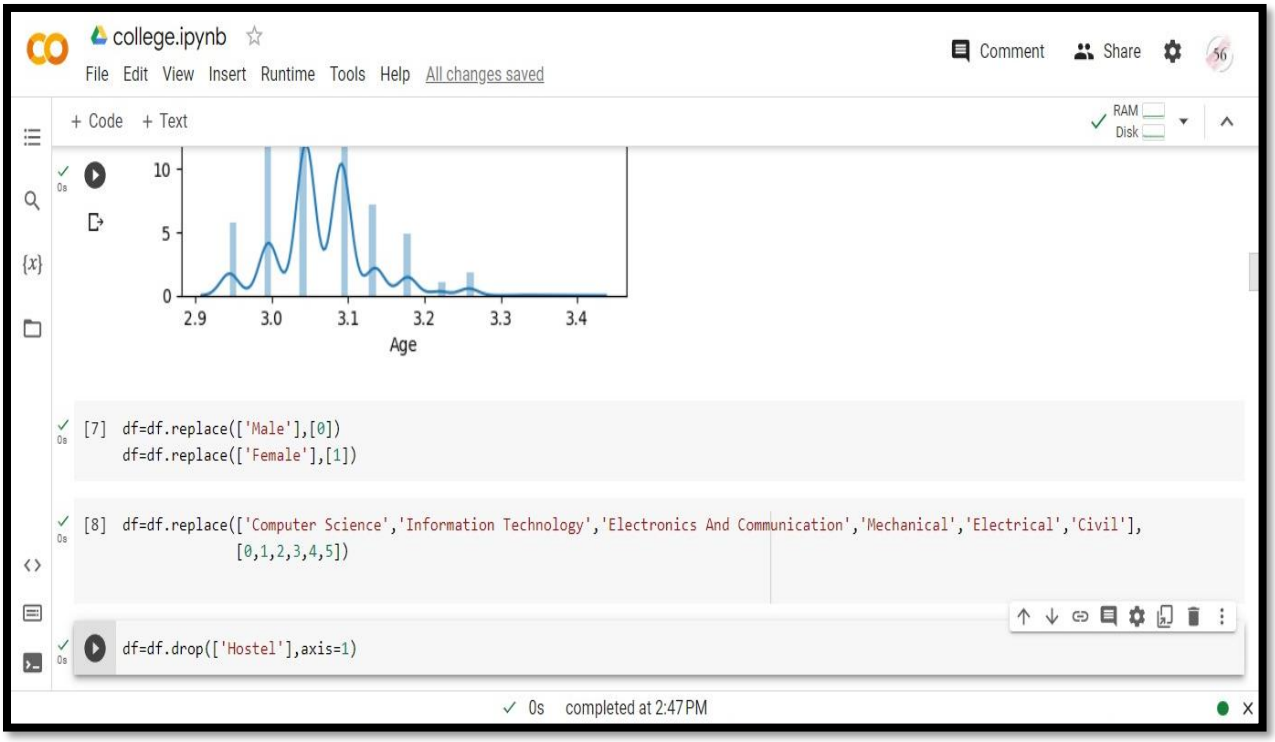



college.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[3]
2 Stream 2966 non-null object
3 Internships 2966 non-null int64
4 CGPA 2966 non-null int64
5 Hostel 2966 non-null int64
6 HistoryOfBacklogs 2966 non-null int64
7 PlacedOrNot 2966 non-null int64
dtypes: int64(6), object(2)
memory usage: 185.5+ KB
df.isnull().sum()
Age 0
Gender 0
Stream 0
Internships 0
CGPA 0
Hostel 0
HistoryOfBacklogs 0
PlacedOrNot 0
dtype: int64
0s completed at 2:47 PM

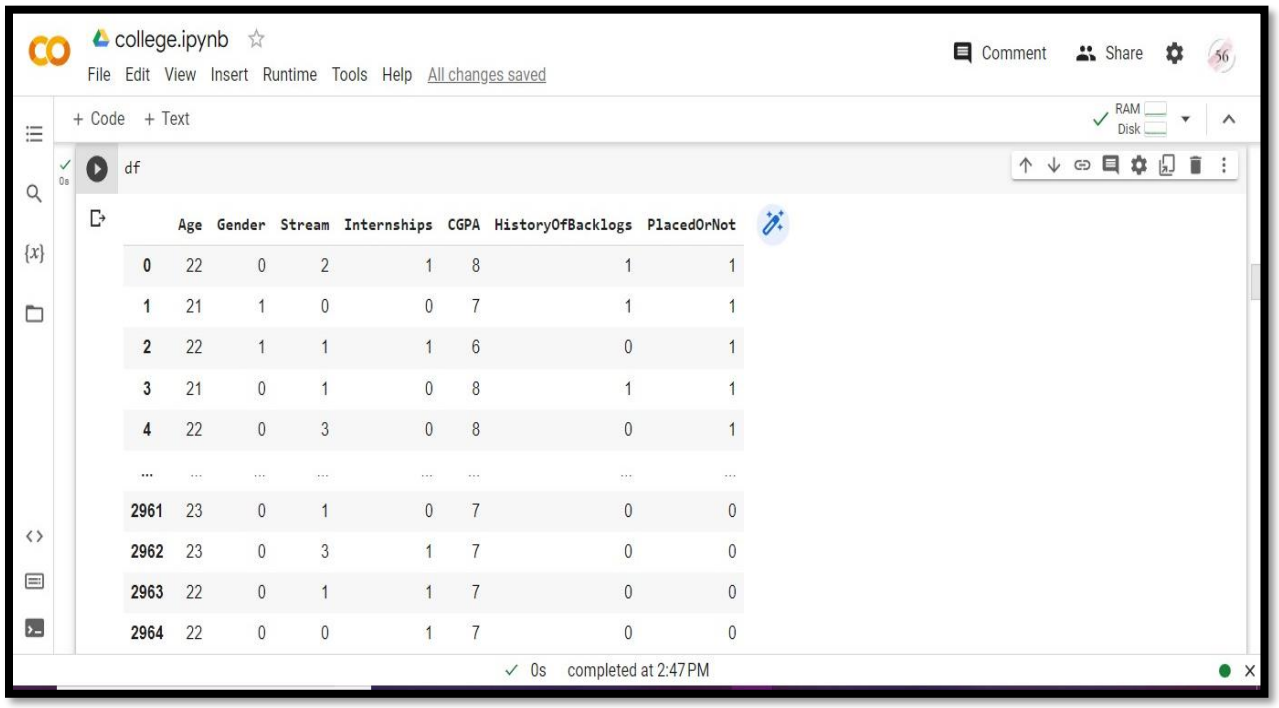
Handling Outliers:




college.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Density
40
35
30
25
20
15
10
5
0
0s completed at 2:47 PM


college.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
56
+ Code + Text
RAM
Disk
10
5
0
2.9
3.0
3.1
3.2
3.3
3.4
Age
[7] df=df.replace(['Male'],[0])
df=df.replace(['Female'],[1])
[8] df=df.replace(['Computer Science','Information Technology','Electronics And Communication','Mechanical','Electrical','Civil'],
[0,1,2,3,4,5])
df=df.drop(['Hostel'],axis=1)
0s completed at 2:47 PM

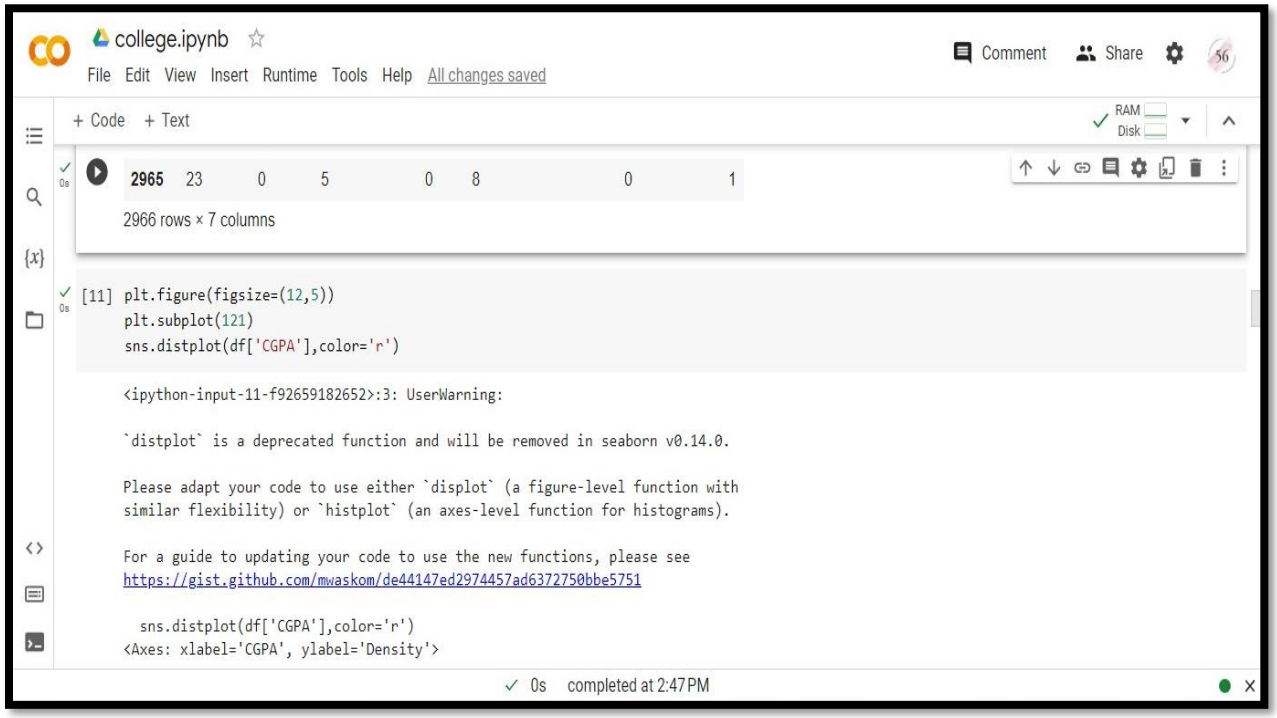
Handling Categorial Values:

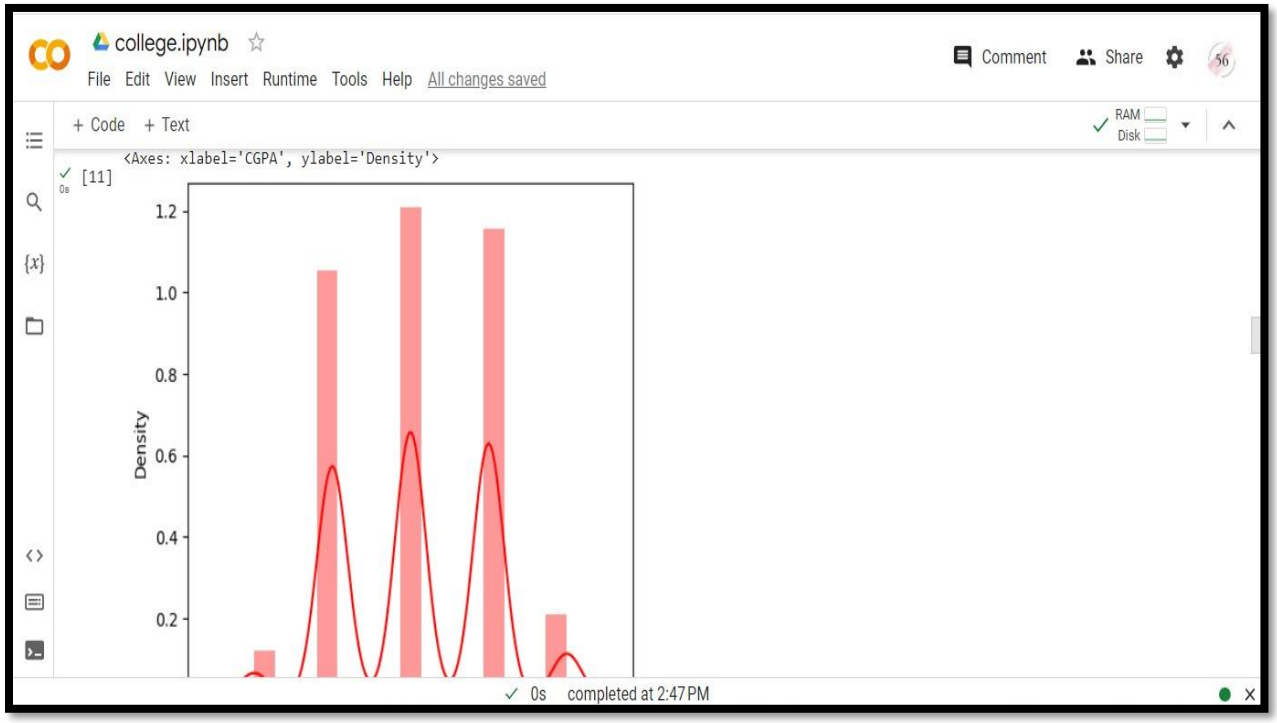


	Age	Gender	Stream	Internships	CGPA	HistoryOfBacklogs	PlacedOrNot
0	22	0	2	1	8	1	1
1	21	1	0	0	7	1	1
2	22	1	1	1	6	0	1
3	21	0	1	0	8	1	1
4	22	0	3	0	8	0	1
...	...	...	...	...	...	...	...
2961	23	0	1	0	7	0	0
2962	23	0	3	1	7	0	0
2963	22	0	1	1	7	0	0
2964	22	0	0	1	7	0	0

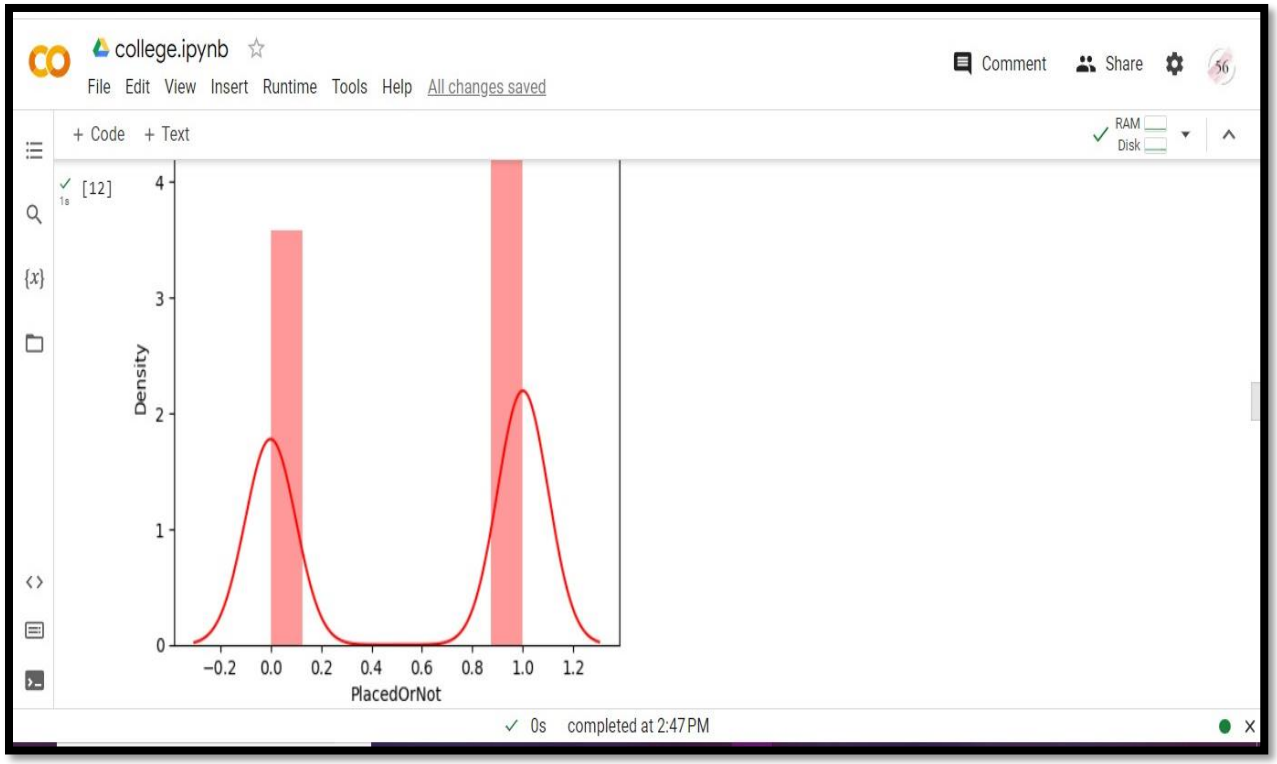
Exploratory Data Analysis:

Univariate analysis:

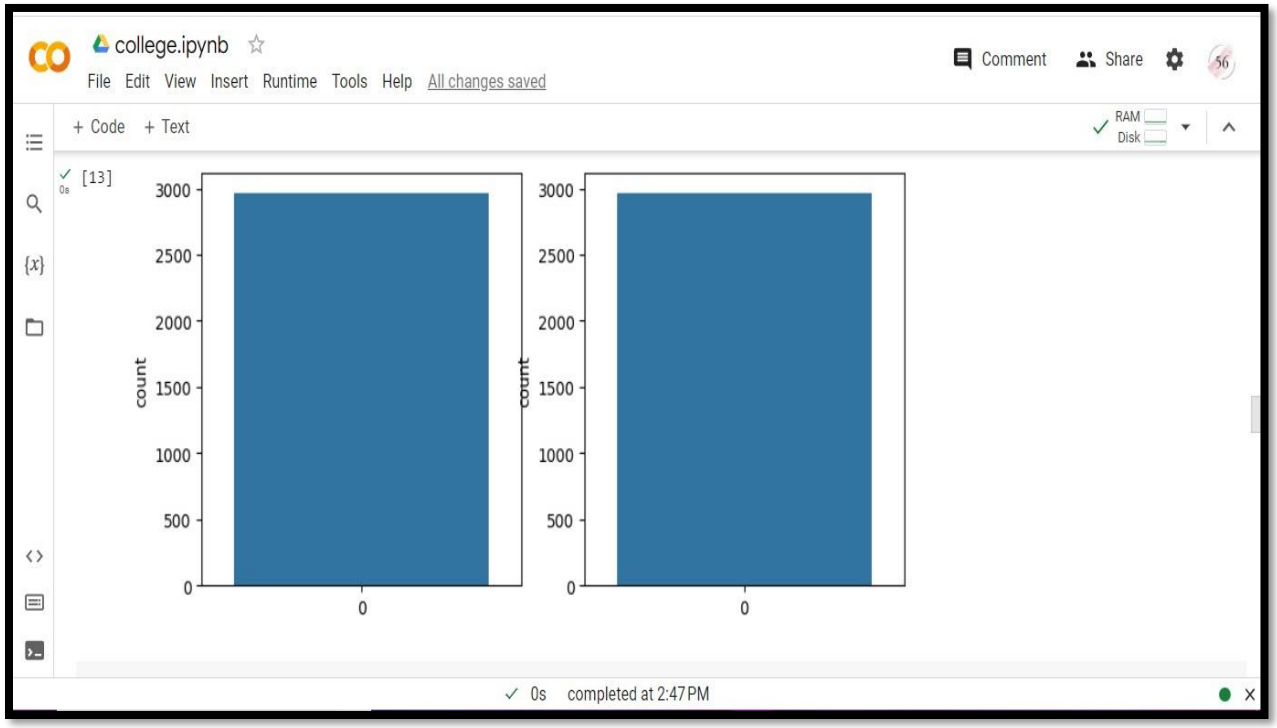



college.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[11]
<Axes: xlabel='CGPA', ylabel='Density'>
Density
1.2
1.0
0.8
0.6
0.4
0.2
0s completed at 2:47 PM

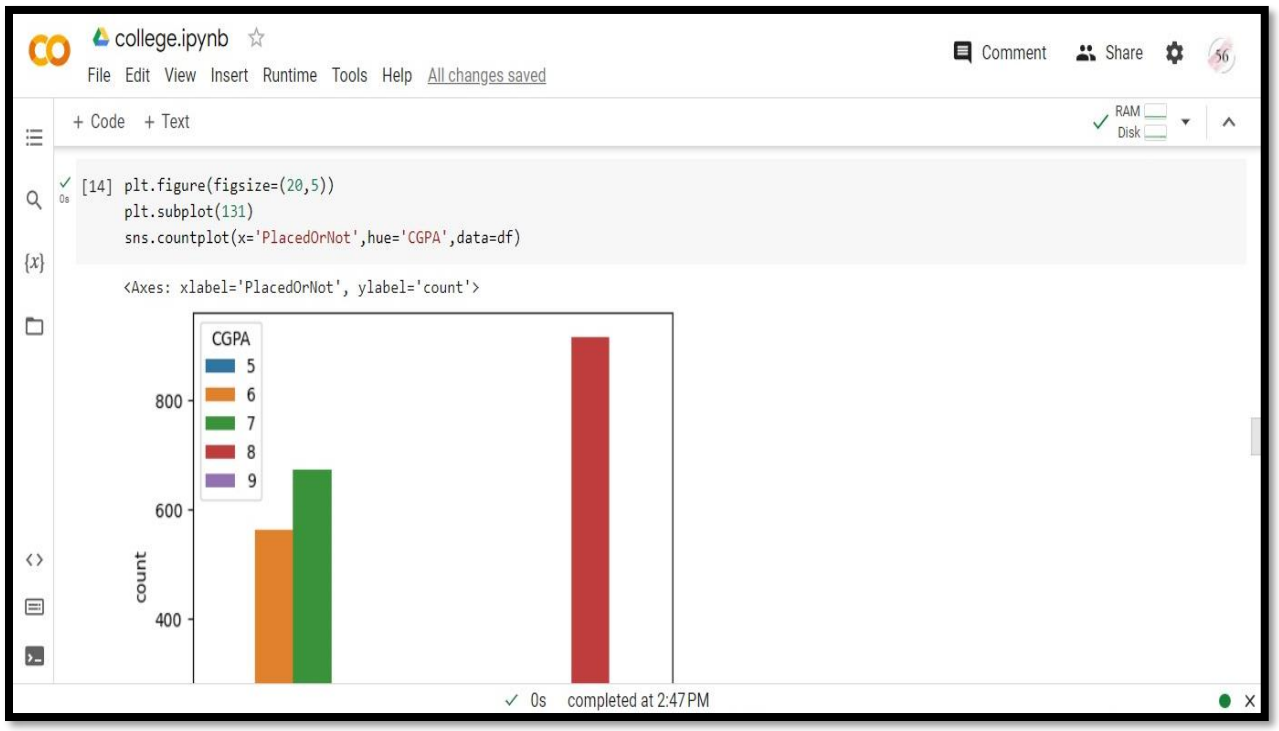

college.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
56
+ Code + Text
RAM
Disk
[12]
plt.figure(figsize=(12,5))
plt.subplot(121)
sns.distplot(df['PlacedOrNot'],color='r')
<ipython-input-12-5e468beb8a0d>:3: UserWarning:
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
sns.distplot(df['PlacedOrNot'],color='r')
<Axes: xlabel='PlacedOrNot', ylabel='Density'>
4
0s completed at 2:47 PM

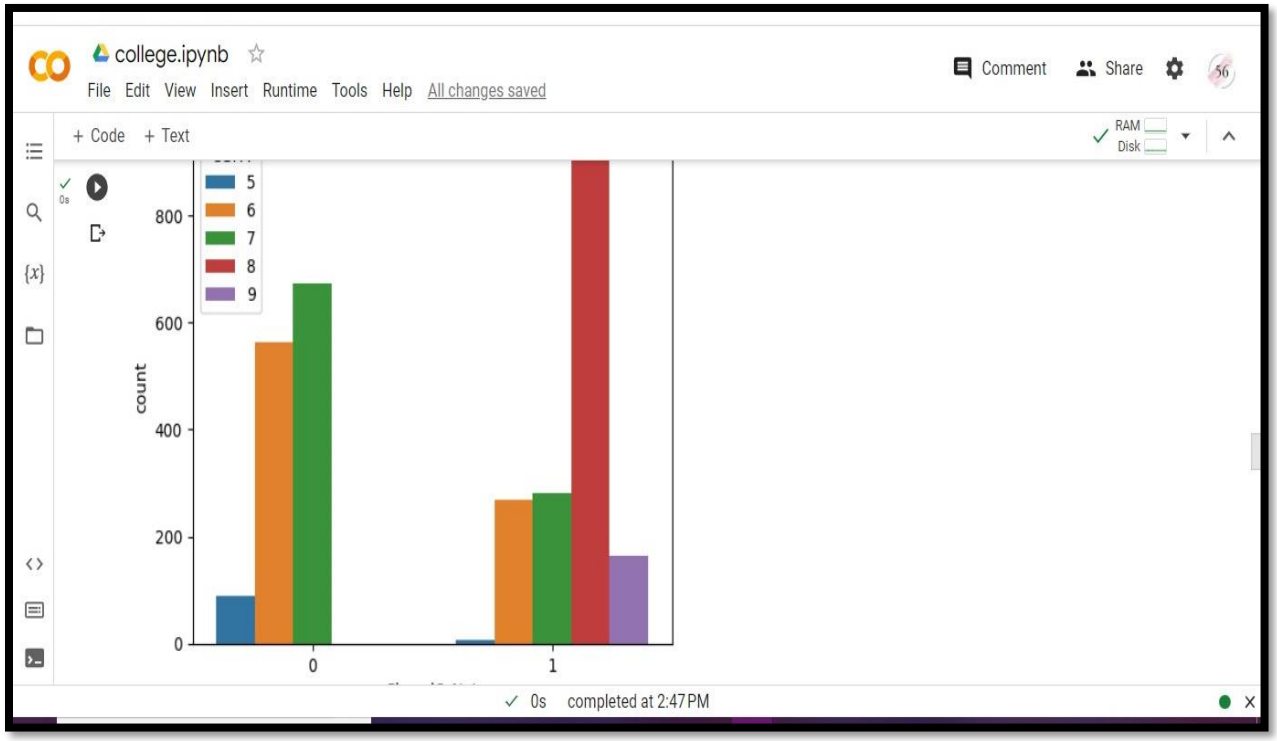

college.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[12]
Density
PlacedOrNot
0s completed at 2:47PM

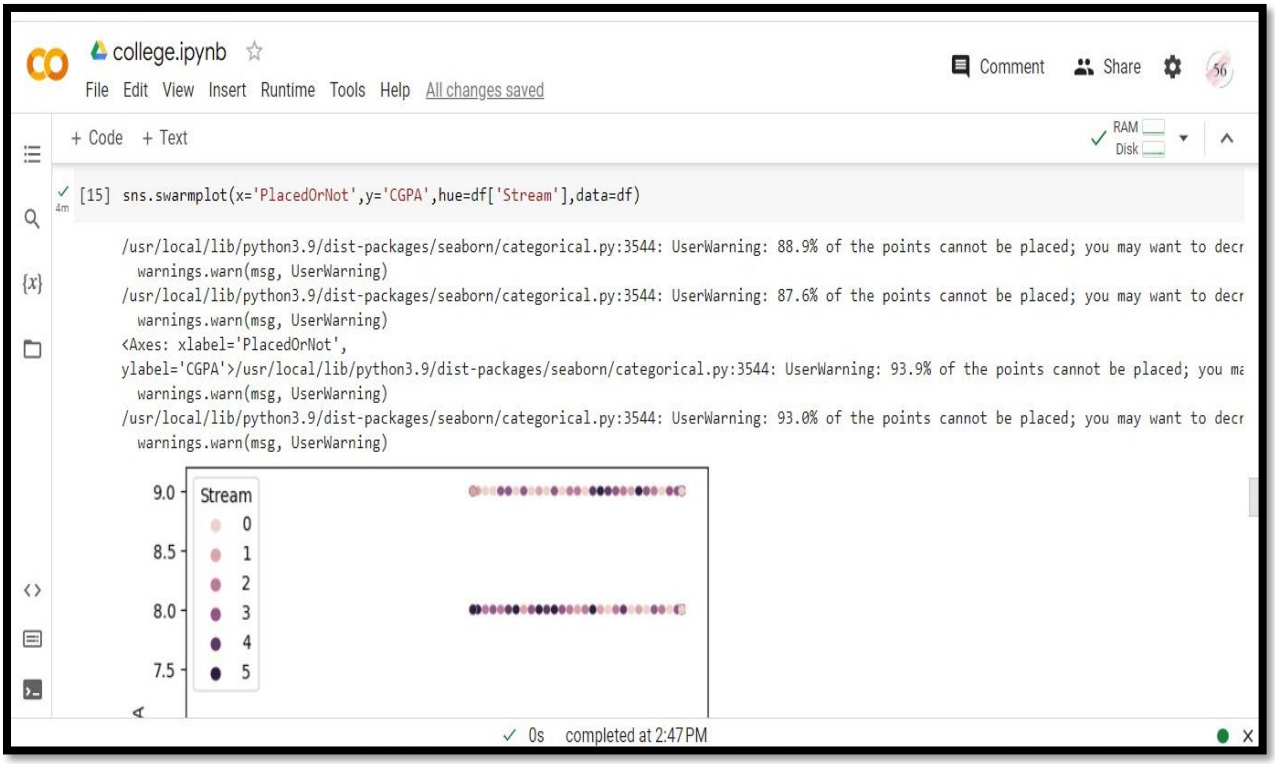
Bivariate analysis:

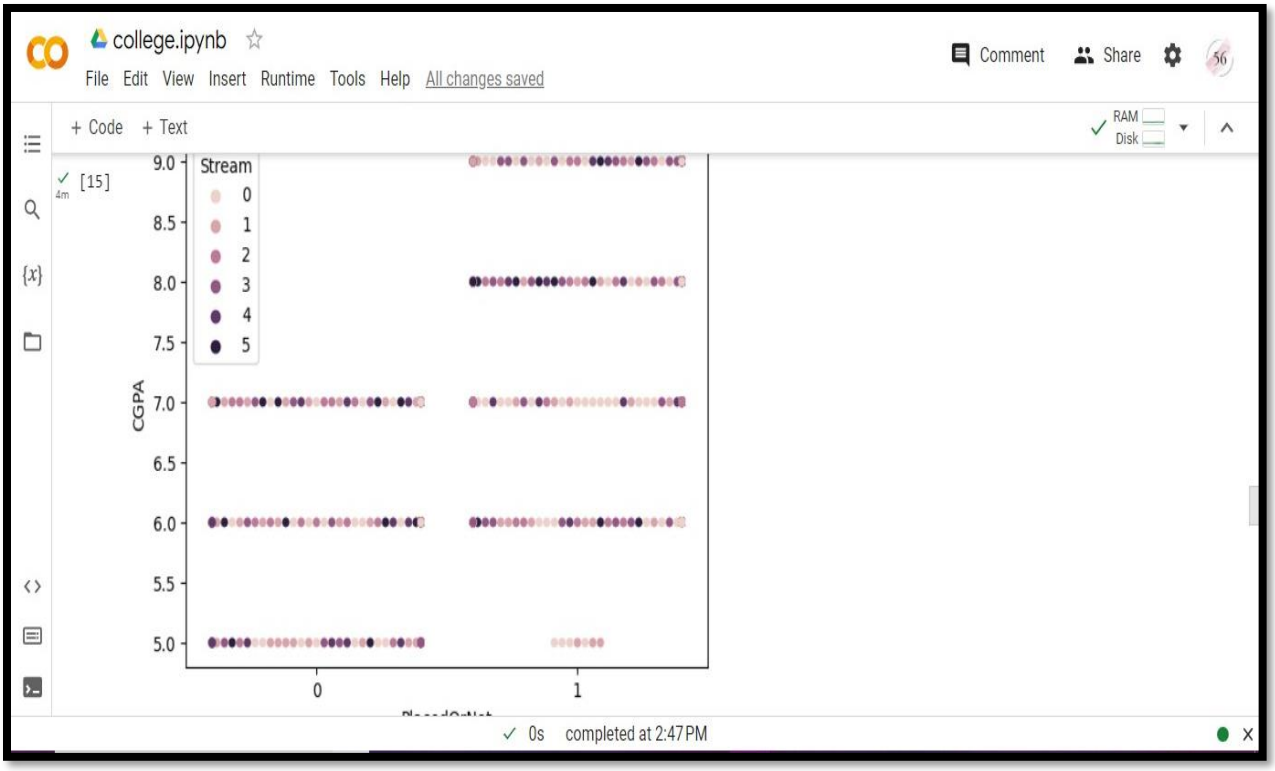

college.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[13]
0s
3000
2500
2000
1500
1000
500
0
count
0
3000
2500
2000
1500
1000
500
0
count
0
0s completed at 2:47 PM

Multivariate analysis:

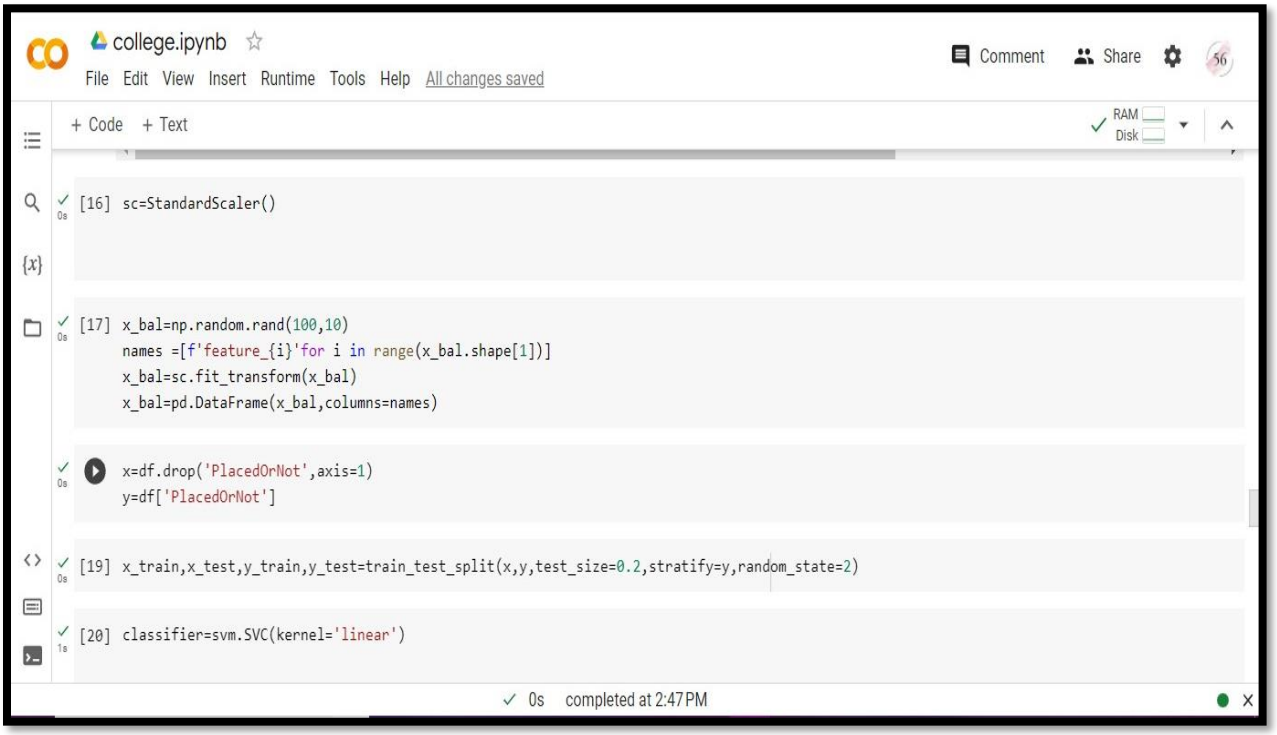



college.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
5
6
7
8
9
count
800
600
400
200
0
0
1
0s completed at 2:47 PM


college.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[15] sns.swarmplot(x='PlacedOrNot',y='CGPA',hue=df['Stream'],data=df)
/usr/local/lib/python3.9/dist-packages/seaborn/categorical.py:3544: UserWarning: 88.9% of the points cannot be placed; you may want to decr
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.9/dist-packages/seaborn/categorical.py:3544: UserWarning: 87.6% of the points cannot be placed; you may want to decr
warnings.warn(msg, UserWarning)
<Axes: xlabel='PlacedOrNot',
ylabel='CGPA'>/usr/local/lib/python3.9/dist-packages/seaborn/categorical.py:3544: UserWarning: 93.9% of the points cannot be placed; you ma
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.9/dist-packages/seaborn/categorical.py:3544: UserWarning: 93.0% of the points cannot be placed; you may want to decr
warnings.warn(msg, UserWarning)
Stream
0
1
2
3
4
5
9.0
8.5
8.0
7.5
0s completed at 2:47 PM


college.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[15]
4m
Stream
0
1
2
3
4
5
CGPA
9.0
8.5
8.0
7.5
7.0
6.5
6.0
5.5
5.0
0
1
0s completed at 2:47 PM

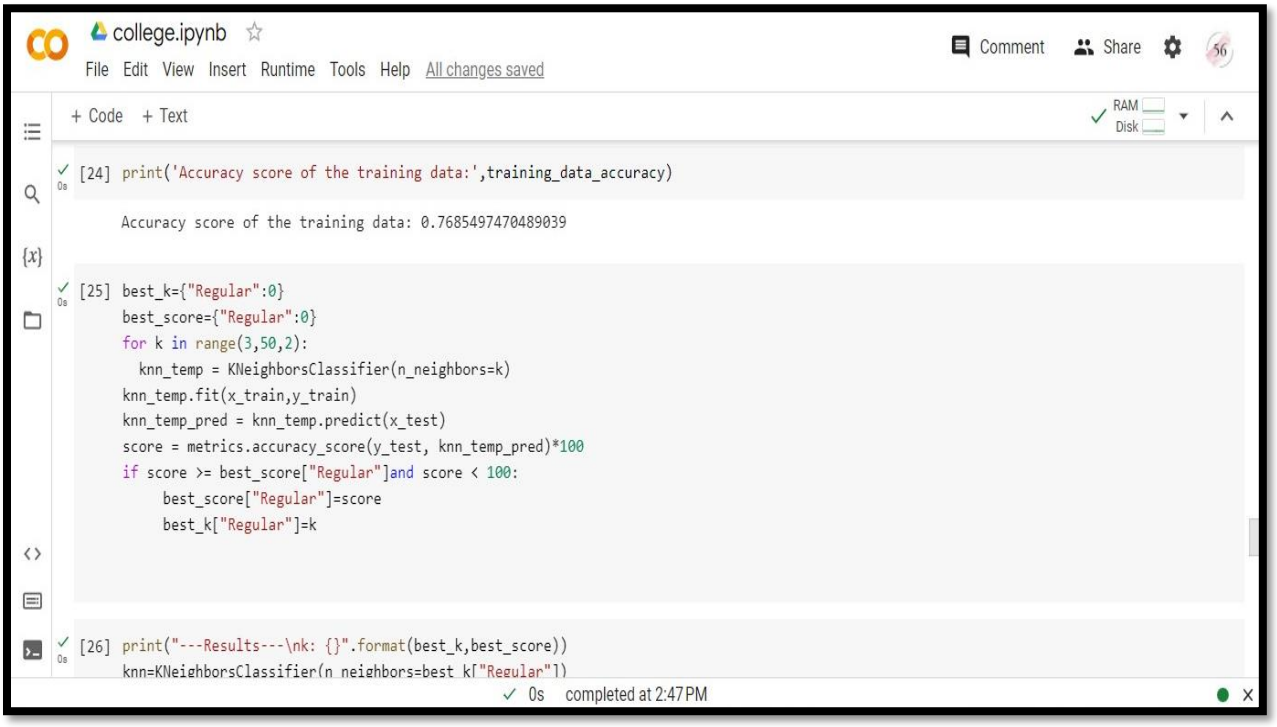
Scaling the data:



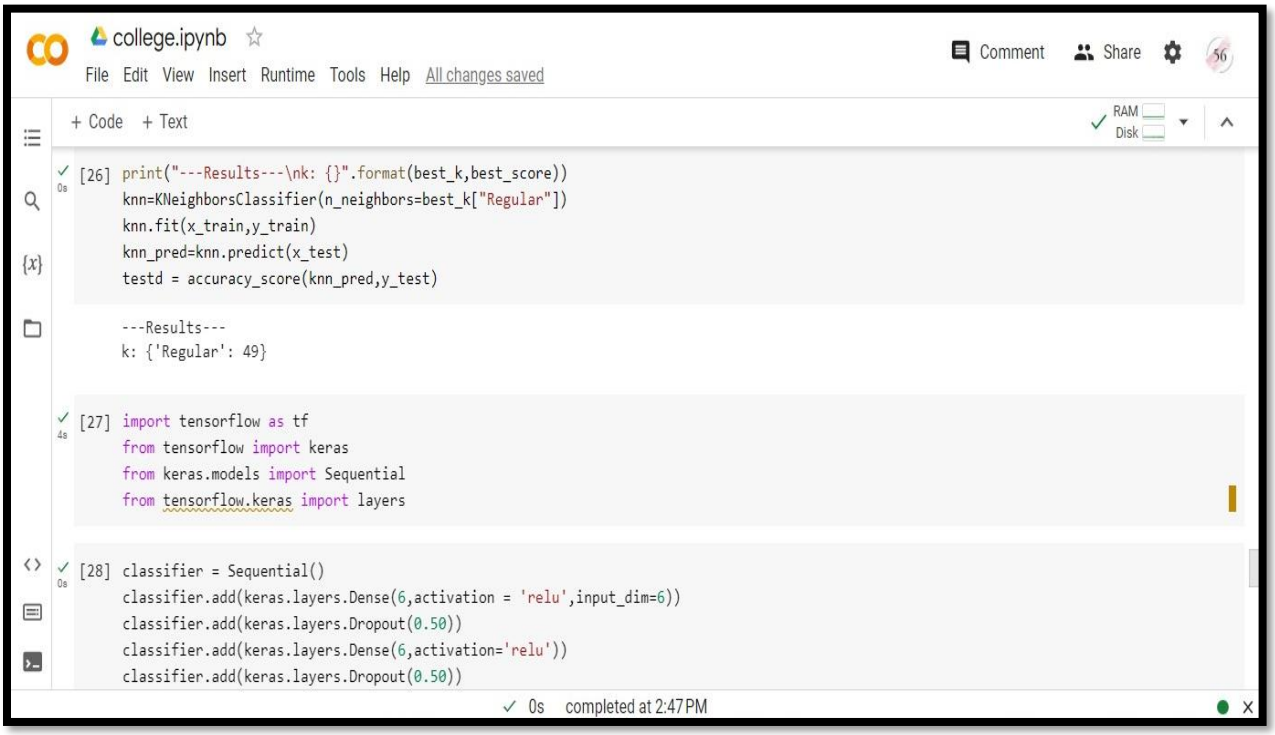
Model Building:

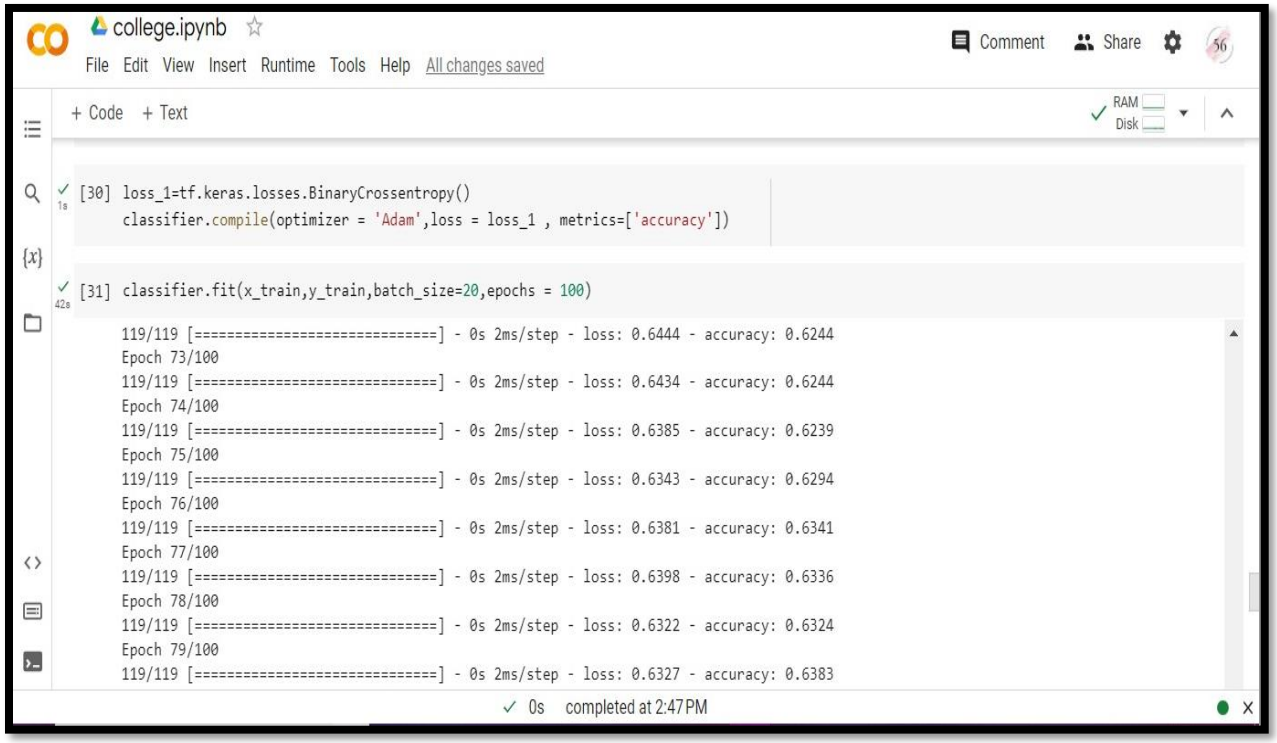
SVM model:

KNN model:


college.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[24] print('Accuracy score of the training data:',training_data_accuracy)
Accuracy score of the training data: 0.7685497470489039
[25] best_k={"Regular":0}
best_score={"Regular":0}
for k in range(3,50,2):
knn_temp = KNeighborsClassifier(n_neighbors=k)
knn_temp.fit(x_train,y_train)
knn_temp_pred = knn_temp.predict(x_test)
score = metrics.accuracy_score(y_test, knn_temp_pred)*100
if score >= best_score["Regular"]and score < 100:
best_score["Regular"]=score
best_k["Regular"]=k
[26] print("---Results---\nk: {}".format(best_k,best_score))
0s completed at 2:47 PM

Artificial neural network model:




college.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[30] loss_1=tf.keras.losses.BinaryCrossentropy()
classifier.compile(optimizer = 'Adam',loss = loss_1 , metrics=['accuracy'])
[31] classifier.fit(x_train,y_train,batch_size=20,epochs = 100)
119/119 [==============================] - 0s 2ms/step - loss: 0.6444 - accuracy: 0.6244
Epoch 73/100
119/119 [==============================] - 0s 2ms/step - loss: 0.6434 - accuracy: 0.6244
Epoch 74/100
119/119 [==============================] - 0s 2ms/step - loss: 0.6385 - accuracy: 0.6239
Epoch 75/100
119/119 [==============================] - 0s 2ms/step - loss: 0.6343 - accuracy: 0.6294
Epoch 76/100
119/119 [==============================] - 0s 2ms/step - loss: 0.6381 - accuracy: 0.6341
Epoch 77/100
119/119 [==============================] - 0s 2ms/step - loss: 0.6398 - accuracy: 0.6336
Epoch 78/100
119/119 [==============================] - 0s 2ms/step - loss: 0.6322 - accuracy: 0.6324
Epoch 79/100
119/119 [==============================] - 0s 2ms/step - loss: 0.6327 - accuracy: 0.6383
0s completed at 2:47 PM

Model Deployment:

Save the best model:

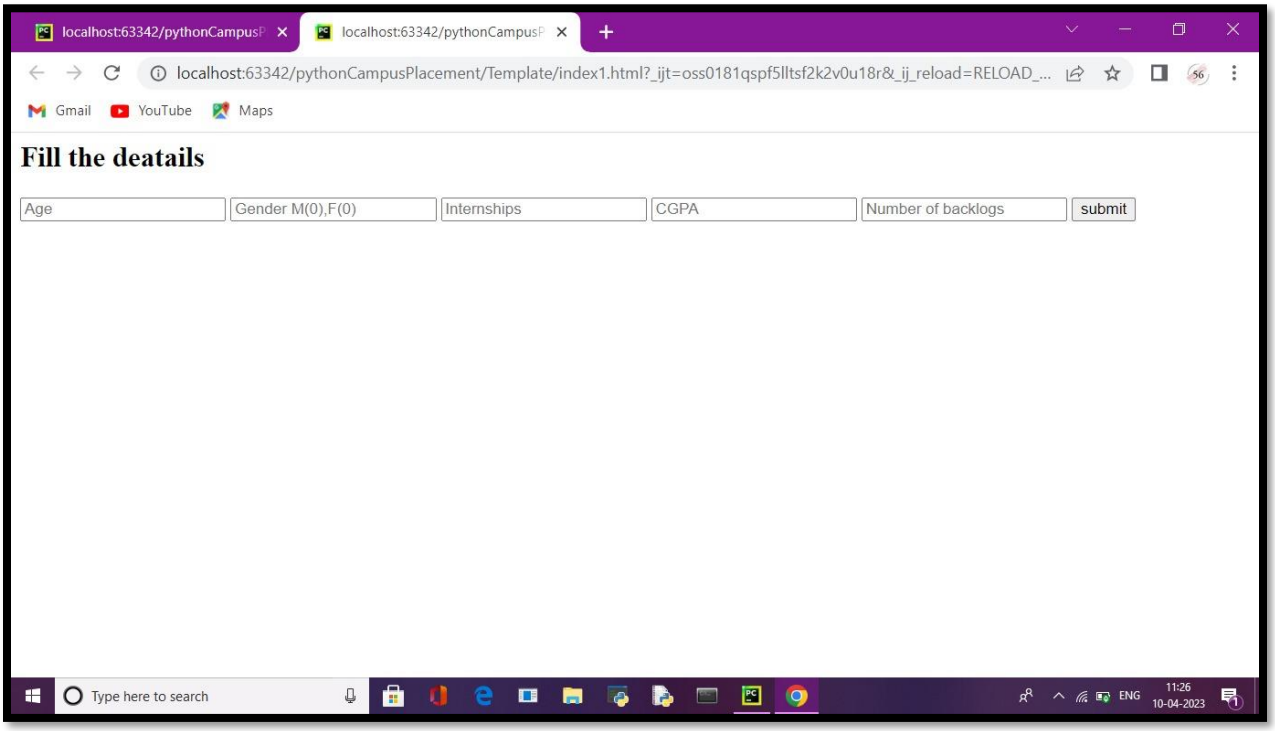
4.Results:

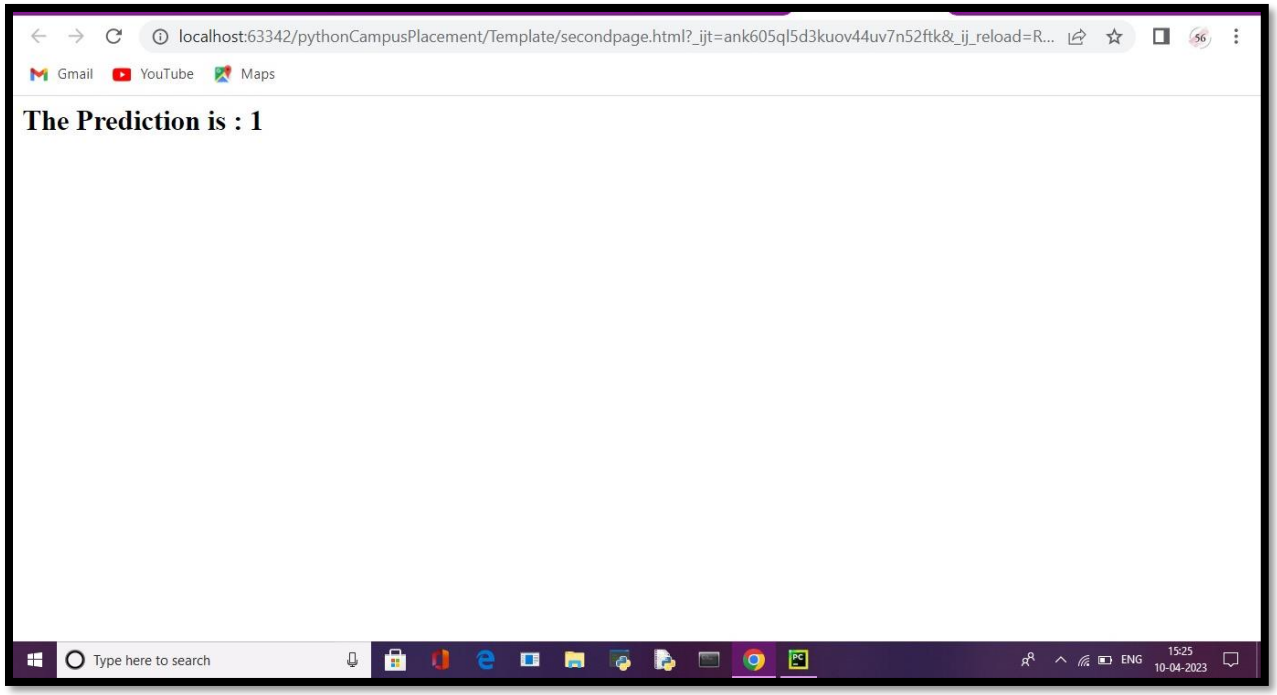
Integrate with Web Framework:

1)index.html:

2)index1.html:

3)secondpage.html:




localhost:63342/pythonCampusPlacement/Template/secondpage.html?_ijt=ank605ql5d3kuov44uv7n52ftk&_ij_reload=R...
Gmail
YouTube
Maps
The Prediction is : 1
Type here to search
ENG
15:25
10-04-2023

5. ADVANTAGE:

Advantages:

Increased accuracy: Using machine learning can lead to increased accuracy in identifying patterns and trends in campus placement data. The algorithm can process large amounts of data and identify subtle relationships and correlations that might not be apparent through manual analysis.

Improved efficiency: Machine learning can process large amounts of data quickly and accurately, making it a more efficient solution for identifying patterns and trends in campus placement data than manual analysis.

Ability to handle complex data: Machine learning can handle complex data structures, such as unstructured data, and identify patterns and trends in that data.

Scalability: Machine learning algorithms can be easily scaled to handle large volumes of data.

Automated feedback: Machine learning models can incorporate automated feedback mechanisms that allow the algorithm to continuously learn and improve over time.

6.APPLICATION:

There are several applications for identifying patterns and trends in campus placement data using machine learning. Some of the most significant applications include:

Improving student outcomes: Machine learning can be used to identify patterns and trends in campus placement data that can help educators and administrators identify areas where students may need additional support or resources. This can help improve student outcomes and ensure that students are well-prepared for the job market.

Enhancing career services: Machine learning can be used to identify trends in the job market and help career services offices provide more targeted and relevant advice to students. For example, career services offices can use machine learning to identify which industries are hiring and what skills are in demand, allowing them to provide more tailored advice to students.

Streamlining recruitment: Machine learning can be used by employers to streamline their recruitment processes and identify candidates who are most likely to be a good fit for their organizations. By analyzing campus placement data, employers can identify patterns and trends in the skills and qualifications of successful candidates, and use this information to inform their recruitment strategies.

Identifying skills gaps: Machine learning can be used to identify skills gaps in the job market, allowing educators and policymakers to develop more targeted training programs and ensure that students are well-prepared for the job market.

Improving diversity and inclusion: Machine learning can help identify biases in the campus placement process and provide insights into ways to improve diversity and inclusion. For example, machine learning can be used to identify biases in the hiring process or to identify areas where certain groups of students may be underrepresented.

7.CONCLUSION:

Conclusion for Identifying Patterns and Trends in Campus Placement Data using Machine Learning

Identifying patterns and trends in campus placement data using machine learning can provide valuable insights to educational institutions and employers to improve their recruitment processes and prepare students for the job market. By analyzing factors such as academic performance, skills, and demographics, machine learning algorithms can identify correlations and make predictions about which students are more likely to be successful in obtaining employment.

Furthermore, this data can help educational institutions tailor their curriculum to meet the demands of employers and the job market. By analyzing which skills and traits are in high demand, institutions can adjust their programs to better prepare students for their careers.

Overall, the use of machine learning in analyzing campus placement data can provide significant benefits for both educational institutions and employers, ultimately leading to improved job outcomes for students and a more efficient recruitment process for employers.

8.FUTURE SCOPE:

Future enhancement for Identifying Patterns and Trends in Campus Placement Data using Machine Learning

There are several potential enhancements that could be made to identify patterns and trends in campus placement data using machine learning. Here are a few ideas:

Incorporate natural language processing (NLP): Many campus placement reports include written feedback from both employers and students. By incorporating NLP techniques, machine learning algorithms could extract insights from this unstructured data to identify patterns and trends in what employers are looking for in candidates and how students are responding to their job offers.

Use graph analysis techniques: Campus placement data typically involves complex relationships between multiple variables such as colleges, companies, job roles, and students. Graph analysis techniques such as network analysis and graph clustering could be used to identify patterns and trends in these relationships.

Develop predictive models: Rather than just identifying patterns and trends in past data, machine learning algorithms could be used to develop predictive models that forecast future placement trends. These models could help colleges and students prepare for upcoming recruitment seasons and anticipate changes in the job market.

Incorporate external data sources: Campus placement data could be enriched by incorporating external data sources such as industry reports, economic indicators, and social media data. This would provide a more comprehensive view of the job market and enable more accurate predictions of future trends.

Develop an interactive dashboard: To make the insights generated from machine learning more accessible to stakeholders, an interactive dashboard could be developed that visualizes key trends and allows users to drill down into specific data points. This would enable colleges and students to make more informed decisions about their recruitment strategies.

9.APPENDIX:

9.1 Source code:

```
import numpy as np
import pandas as pd
import os
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import svm
from sklearn.metrics import accuracy_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
from sklearn.model_selection import cross_val_score
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import joblib
from sklearn.metrics import accuracy_score
df =pd.read_csv(r'/content/collegePlace.csv')
df.head()
df.info()
df.isnull().sum()
def transformationplot(feature):
  plt.figure(figsize=(12,5))
  plt.subplot(1,2,1)
  sns.distplot(feature)
transformationplot(np.log(df['Age']))
df=df.replace(['Male'],[0])
```

```
df=df.replace(['Female'],[1])
df=df.replace(['Computer Science','Information Technology','Electronics And Communication','Mechanical','Electrical','Civil'],
        [0,1,2,3,4,5])
df=df.drop(['Hostel'],axis=1)
df
plt.figure(figsize=(12,5))
plt.subplot(121)
sns.distplot(df['CGPA'],color='r')
plt.figure(figsize=(12,5))
plt.subplot(121)
sns.distplot(df['PlacedOrNot'],color='r')
plt.figure(figsize=(18,4))
plt.subplot(1,4,1)
sns.countplot(df['Gender'])
plt.subplot(1,4,2)
sns.countplot(df['Stream'])
plt.show()
plt.figure(figsize=(20,5))
plt.subplot(131)
sns.countplot(x='PlacedOrNot',hue='CGPA',data=df)
sns.swarmplot(x='PlacedOrNot',y='CGPA',hue=df['Stream'],data=df)

sc=StandardScaler()
x_bal=np.random.rand(100,10)
names =[f'feature_{i}'for i in range(x_bal.shape[1])]
x_bal=sc.fit_transform(x_bal)
x_bal=pd.DataFrame(x_bal,columns=names)
x=df.drop('PlacedOrNot',axis=1)
```

```
y=df['PlacedOrNot']
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,stratify=y,random_state=2)
classifier=svm.SVC(kernel='linear')
classifier.fit(x_train,y_train)
x_train_prediction=classifier.predict(x_train)
training_data_accuracy=accuracy_score(x_train_prediction,y_train)
print('Accuracy score of the training data:',training_data_accuracy)
best_k={"Regular":0}
best_score={"Regular":0}
for k in range(3,50,2):
  knn_temp = KNeighborsClassifier(n_neighbors=k)
knn_temp.fit(x_train,y_train)
knn_temp_pred = knn_temp.predict(x_test)
score = metrics.accuracy_score(y_test, knn_temp_pred)*100
if score >= best_score["Regular"]and score < 100:
    best_score["Regular"]=score
    best_k["Regular"]=k

print("---Results---\nk: {}".format(best_k,best_score))
knn=KNeighborsClassifier(n_neighbors=best_k["Regular"])
knn.fit(x_train,y_train)
knn_pred=knn.predict(x_test)
testd = accuracy_score(knn_pred,y_test)
import tensorflow as tf
from tensorflow import keras
from keras.models import Sequential
from tensorflow.keras import layers
classifier = Sequential()
```

```
classifier.add(keras.layers.Dense(6,activation = 'relu',input_dim=6))
classifier.add(keras.layers.Dropout(0.50))
classifier.add(keras.layers.Dense(6,activation='relu'))
classifier.add(keras.layers.Dropout(0.50))
classifier.add(keras.layers.Dense(1,activation='sigmoid'))
loss_1=tf.keras.losses.BinaryCrossentropy()
classifier.compile(optimizer = 'Adam',loss = loss_1 , metrics=['accuracy'])
classifier.fit(x_train,y_train,batch_size=20,epochs = 100)
import pickle
pickle.dump(knn,open(''placement.pkl'','wb'))
model=pickle.load(open('placement.pkl','rb'))
```

1)index.html

```
<section id="hero" class="d-flex-column justify-content-center">
<div class="container">
<div class="row justify-content-center">
<div class="col-xl-8">
<h1>Identifying Patterns and Trends</h1>
<h1>in Campus placement Data</h1>
<h1> using Machine Learning</h1>
</div>
</div>
</div>
</section>
```

2.index1.html:

```
<section id="about"class="about">
<div class="container">
<div class="section-title">
<h2>Fill the deatails</h2>
</div>
<div.class="row content">
<div class="first">
<form action="{{ url_for('y_predict')}}"method="POST">
<input type="number"id="sen1"name="sen1"placeholder="Age">
<input type="number"id="sen2"name="sen2"placeholder="Gender M(0),F(0)"
<input type="number"id="sen3"name="sen3"placeholder="Stream
CS(0),IT(1),ECE(2),Mech(3),EEE(4),Civil(5)">
<input type="number"id="sen4"name="sen4"placeholder="Internships">
```

```
 <input type="number"id="sen5"name="sen5"placeholder="CGPA">
<input type="number"id="sen6"name="sen6"placeholder="Number of backlogs">
            <input type="submit"value="submit">

</form>
</div>
</div>
</div>
</section>
```

3.Secondpage.html:

```
    <section id="hero" class="d-flex flex-column justify-content-center">
    <div class="container">
    <div class="row justify-content-center">
   <div class="col-xl-8">
    <h1>The Prediction is : {{y}}</h1>
    <h3> 0 represents Not-placed </h3>
    <h3> 1 represents Placed<h2>

    </div>
    </div>
   </div>
   </section>
```

4.Project.py:

```
from flask import Flask, render_template , request
app=Flask(name)
import pickle
import joblib
model=pickle.load(open("placement123.pkl",'rb'))
ct=joblib.load('placement')
@app.route('/')
def hello():
 return render_template("index.html")
@app.route('/guest' , methods =["post"])
def y_predict():
    x_test=[[(yo) for yo in request.form.values()]]
prediction =model.predict(x_test)
    prediction = prediction[0]
return render_template("secondpage.html",y=prediction)
app.run(debug=True)
```