

IDENTIFYING PATTERNS AND TRENDS IN CAMPUS PLACEMENT DATA USING MACHINE LERNING

BY,

R.VAITHEGI

S.SATHYA

M.SNEKA

P.KRISHNA JEYANTHI



RANI ANNA COLLEGE FOR WOMEN

CHAPTER

TABLE OF CONTENT

1.	INTRODUCTION 1.1 Overview 1.2 Purpose
2.	PROBLEM DEFINITION & DESIGN THINKING 2.1 Empathy Map 2.2 Ideation & Brainstorming Map
3.	RESULT
4.	ADVANTAGES & DISADVANTAGES
5.	APPLICATIONS
6.	CONCLUSION
7.	FUTURE SCOPE
8.	APPENDIX

PROJECT REPORT

CHAPTER 1

INTRODUCTION :

1.1 Overview

Campus recruitment is a strategy for sourcing, engaging and hiring young talent for internship and entry-level positions. Campus placement is a program conducted within universities or other educational institution to provide jobs to students nearing completion of their studies. Campus recruitment often involves working with university career fairs to meet in-person with college students and recent graduates. Campus recruiting efforts and finding ways to improve could be a full-time job in and of itself. Technology and recruitment approaches to find, attract and hire college talent are evolving every day. perhaps the best place to start is with the big picture.

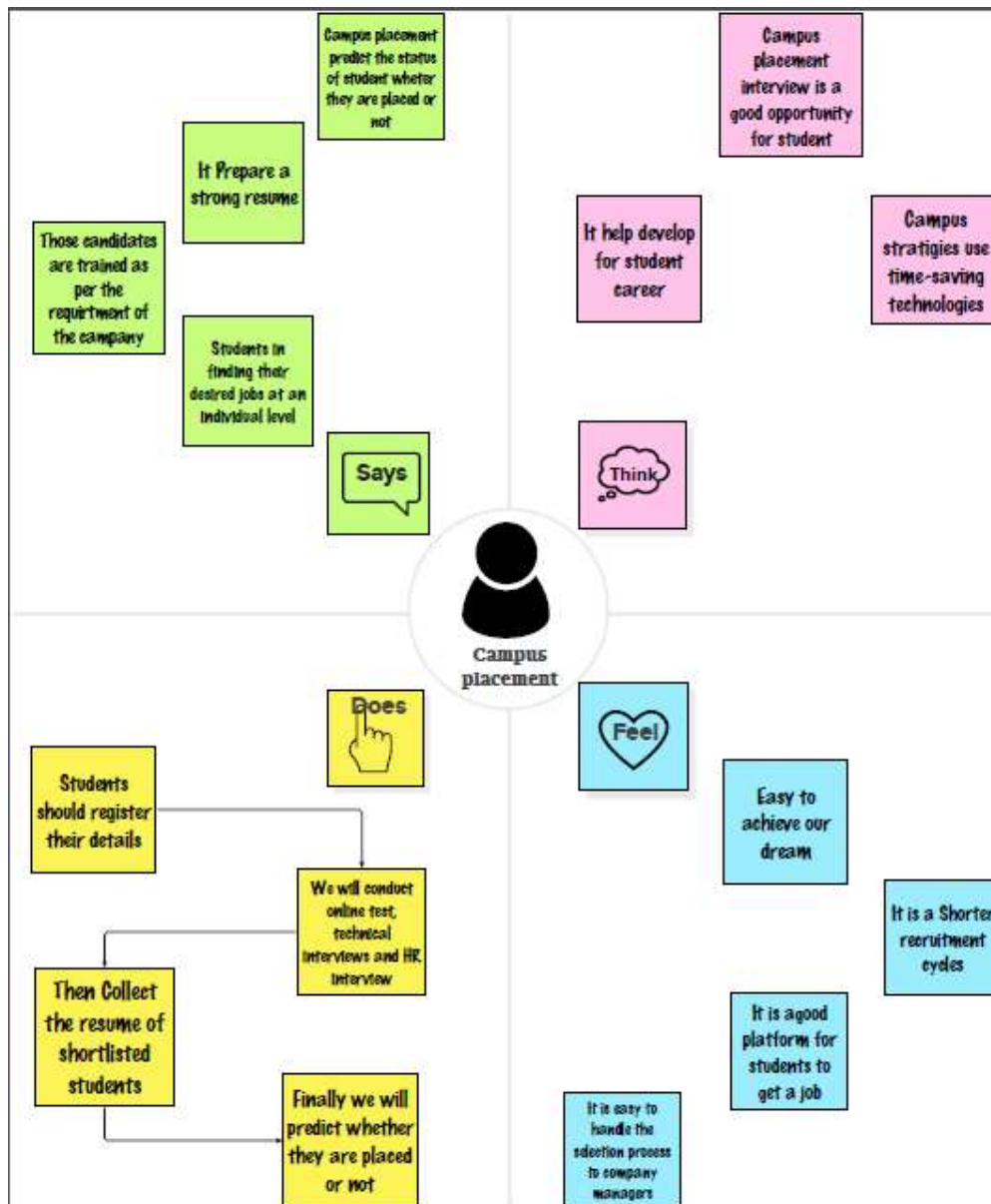
1.2 Purpose:

The purpose of campus placement is so simple to provide you a start of your career and a safe future. There are two types of placement one is on campus placement and the other one is off campus placement. On campus placement, the recruiter companies which come to your college know that they are going to recruit fresher students and they don't expect so much from that students. Off campus placement, then you can feel that there is an increment in the expectations of the recruiting company. And it will be a little it difficult to put some extra efforts in getting yourself placed in a reputed organization.

CHAPTER 2

PROBLEM DEFINITION & DESIGN THINKING

2.1 Empathy Map

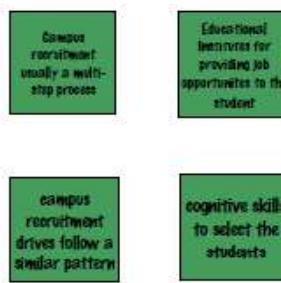


2.2 Ideation & Brainstorming Map

Krishna jeyanthi



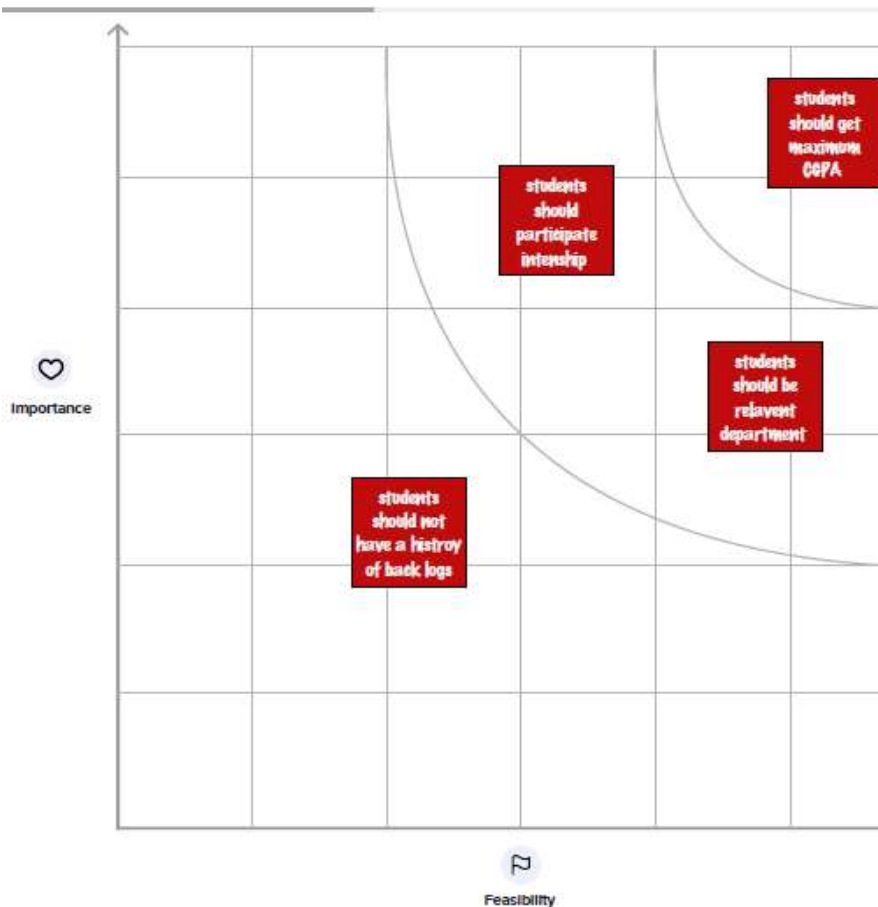
sathya



sneka



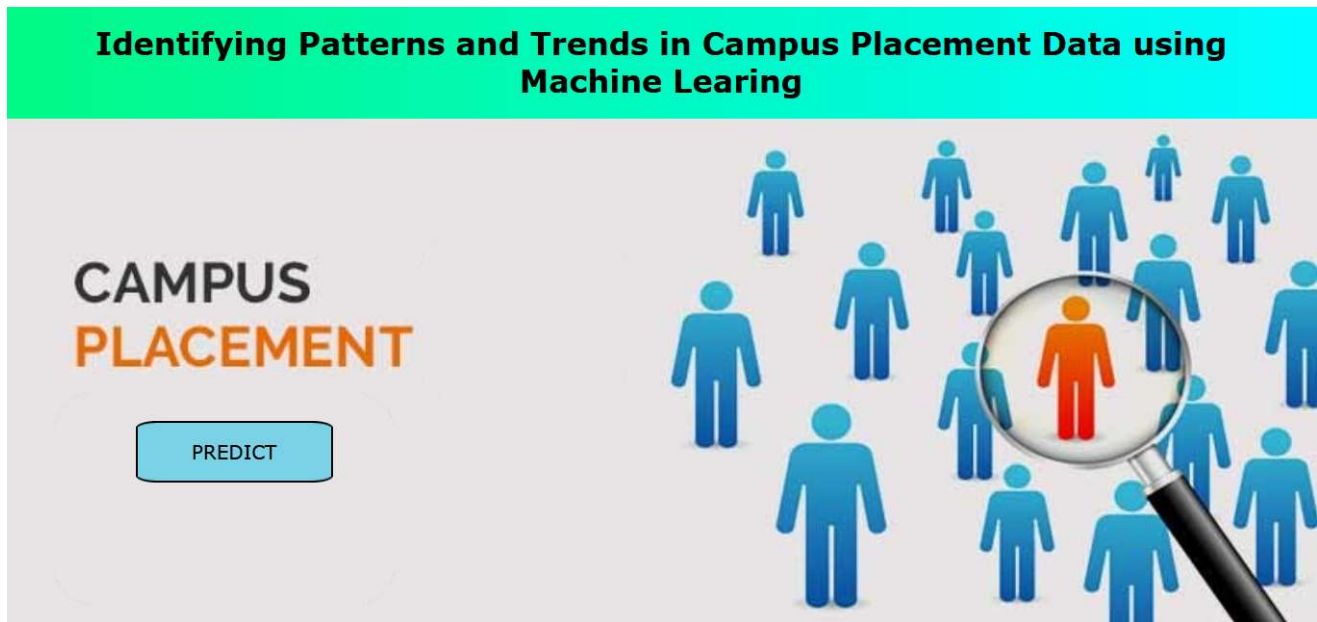
vaithegi



CHAPTER 3

RESULT:

Home Page:



Form Page:

The image shows the form page of the web application, titled "Identifying Patterns and Trends in Campus Placement Data using Machine Learning". The page features a light blue header with the title. Below the header, the text "Fill the details" is displayed in bold black letters. The form contains several input fields with labels and values:

Field	Value
AGE	21
GENDER	1
DEPARTMENT	CS
INTERNSHIP	2
CGPA	7
BACKLOGS	0

A blue button labeled "Submit" is located at the bottom of the form. The background of the page is a dark, textured image of a building interior with large windows and arched doorways.

Result Page:

😊 you are placed 😊



CHAPTER 4

ADVANTAGES :

- ❖ Save Time & Efforts
- ❖ Large Opportunities For Students
- ❖ Campus and Company Bond / Strong Relationship
- ❖ Improved Retention Rates
- ❖ Getting New Knowledge & Skills

DISADVANTAGES :

- ❖ Limited Staff & Time
- ❖ Lack Of Proper Branding
- ❖ High Candidates Number
- ❖ Importance Of a Resume

CHAPTER 5

APPLICATIONS:

- ★ To hire best student of university or college
- ★ Government use for many recruitments process such as TNPSC, SSC and more.
- ★ Education institutions use for admitting higher studies
- ★ Job hiring for private companies

CHAPTER 6

Conclusion :

Maximum work goes manually in the present placement system which make it take time to changes. This includes main problem like searching for the data of students and sorting them along with it. Also, updating student data is a cumbersome job does not have a method to notify the student in time which makes the management of the placements very difficult. The registration of the student for an upcoming placement, the addition of a new user, notifying students, sharing information etc is all met. The student list based on the criteria required which otherwise would have been very difficult to manage.

CHAPTER 7

FUTURE SCOPE:

★ We will arrange a course for the students to increase their skills

★ We will estimate the model for each company individually
As per their requirement.

★ We will implement this project for whole student database instead of feeding details of each student

CHAPTER 8

APPENDIX

Source Code:

▼ Identifying Patterns and Trends in Campus

| ▼ DataSet

| ▼ Flask

| | ▼ static

| | | image

| | | CSS

| | ▼ Templates

| | | index.html

| | | index1.html

| | | secondpage.html

| | | secondpage2.html

| | app.py

| | placement.pkl

| ▼ Training

| | CAMPUSPLACEMENT.ipynb

HOME PAGE:

Index.html

<html>

 <head>

 <title>HOME</title>

 <link rel="stylesheet" type="text/css" href="{{url_for('static',
filename='index.css')}}">

 </head>

 <body>

```
<div>
```

```
    <h1>Identifying Patterns and Trends in Campus Placement  
Data using Machine Learning</h1>
```

```
</div>
```

```
<div>
```

```
    <form action="/guest" method="GET">
```

```
        <input type="submit" name="predict" onclick="action"  
value="PREDICT">
```

```
    </form>
```

```
</div>
```

```
</body>
```

```
</html>
```

```
index.css
```

```
* {
```

```
    margin: 0;
```

```
    box-sizing: border-box;
```

```
    font-family: Verdana;
```

```
}
```

```
body {
```

```
    background-image: url("/static/placement.jpg");
```

```
    background-repeat: no-repeat;
```

```
    background-size: cover;
```

```
}
```

```
h1 {
```

```
    padding: 20px;
```

```
    text-align:center;
```

```
    background      : linear-gradient(to right,#01F985,#00FBFF);
```

```
}
```

```
input[type=submit] {  
    border: 2px solid black;  
    height: 10%;  
    width: 15%;  
    font-size: 20px;  
    text-align: center;  
    padding: 14px 28px;  
    margin: 23% 10%;  
    margin-bottom: 0px;  
    background-color: #79d2e6;  
    border-radius: 10%  
}
```

```
input[type=submit]:hover {  
    background-color: red;  
}
```

```
#myImg {  
    position: absolute;  
    top: 100px;  
    width: 100%;  
    height: 500px;  
    object-fit: cover;  
    object-position: 0% 0%;  
    animation: mymove 5s infinite;  
}
```

```
@keyframes mymove {  
    0% {  
        object-position: 0% 0%;
```

```

}
25% {
    object-position: 20% 0%;
}
50%{
    object-position:30% 50%;
}
100% {
    object-position: 100% 100%;
}
}

```

FORM PAGE:

Index1.html

```

<html>

    <head>

        <title>HOME</title>

        <link rel="stylesheet" href="{{url_for('static',filename =
'index1.css')}}}" type="text/css">

    </head>

    <body>

        <div>

            <h1>Identifying Patterns and Trends in Campus Placement
Data using Machine Learning</h1>

        </div>

        <div id="formId">

            <form action="/y_predict" method="POST">

                <h3>Fill the details</h3>

                <table>

```

```

        <tr>

            <td>AGE</td>

            <td>:</td>

            <td><input type="number" id="sen1"
name="sen1" min="21" placeholder="21 to 45"></td>

        </tr>

        <tr>

            <td>GENDER</td>

            <td>:</td>

            <td><input type="number" id="sen2"
name="sen2" min="0" max="1" placeholder="Gender M(0),F(1)"></td>

        </tr>

        <tr>

            <td>DEPARTMENT</td>

            <td>:</td>

            <td><select id="sen3" name="sen3">
<option value="-1"selected>[choose yours]</option><option
value=0>Civil</option><option value=1>CS</option><option value=2>Electronics
</option><option value=3>Electronics & Communication</option><option value=4>
Information Technology</option><option value=5> Mechanical</option> </select>
</td>

        </tr>

        <tr>

            <td>INTERNSHIP</td>

            <td>:</td><td><input type="number"
id="sen4" name="sen4" min="0" max="3" placeholder="Internship"></td>

        </tr>

        <tr>

            <td>CGPA</td>

            <td>:</td>

```

```

                                <td><input type="number" id="sen5"
name="sen5" min="4" max="10" placeholder="CGPA"></td>

                                </tr>

                                <tr>

                                <td>BACKLOGS</td>

                                <td>:</td>

                                <td><input type="number" id="sen6"
name="sen6" min="0" max="5" placeholder="Number of backlogs"></td>

                                </tr>

                                <tr>

                                <td colspan="3"><input type="submit"
value="Submit" onclick="action"></td>

                                </tr>

                                </table>

                                <!--<input type="number" id="sen3" name="sen3"
placeholder="Stream civil(0),CS(1),Electrical(2),Electronics And
Communication(3),Information Technology(4),Mechanical(5)">-->

                                </form>

                                </div>

                                </body>

                                </html>

```

index1.css

```

* {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

    font-family: Verdana;

}

h1 {

```

```
padding: 20px;
text-align:center;
background: linear-gradient(to right,#01F985,#00FBFF);
}
#formId {
background: url('office.jpg');
background-repeat: no-repeat;
background-size: cover;
}
form {
margin: auto;
padding:10px;
height: 82%;
width: 45%;
background-color: #ffffff;
box-shadow: 0 0 10px 5px #fff;
opacity: 0.7;
text-align: center;
border-radius: 10px;
font-size:25px;
}
table {
font-size: 15px;
font-weight: bold;
}
input {
margin: 5%;
```



```
padding: 10px;
height: 10%;
width: 100%;
border-radius: 5px;
border: none;
border: 1px solid grey;
background-color: #ffffff;
font-size: 15px;
box-sizing: border-box;
}
select {
padding: 10px;
margin: 5%;
border-radius: 5px;
border: 1px solid grey;
background-color: #ffffff;
font-size: 15px;
box-sizing: border-box;
}
input[type=submit] {
border: 2px solid black;
height: 10%;
width: 25%;
text-align: center;
margin: 0% 40%;
background-color: #79d2e6;
```

```

        border-radius: 10%
    }
    input[type=submit]:hover {
        background-color: red;
    }

```

RESULT PAGE:

1)secondpage.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>HOME</title>
```

```
    <link rel="stylesheet"
```

```
href="{{url_for('static',filename='secondpage.css')}}" type="text/css">
```

```
    <!--style type="text/css">
```

```
        body{ background-image: url('image.jpg');
```

```
background-repeat: no-repeat;
```

```
background-attachment: fixed;
```

```
background-size: 100% 100%;}
```

```
    </style-->
```

```
</head>
```

```
<body>
```

```
    <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 id="hero">😊 you are placed 😊</h1>
```

```

</div></div></div>

</section>

</body>

</html>

```

secondpage.css

```

* {
    box-sizing: border-box;
    font-family: Verdana;
}

body {
    margin: 0;
    background-image: url("image4.png");
    background-repeat: no-repeat;
    background-size: cover;
}

h1 {
    margin: 0;
    padding: 20px;
    text-align:center;
    background : linear-gradient(to right,#01F985,#00FBFF);
}

#formId {
    //height: 100%;
    //width: 100%;
    background-color: black;
}

```

```
}  
  
form {  
    border:none;  
    background-color: hotpink;  
    text-align: center;  
    //opacity:0.3;  
    background : linear-gradient(to right,#01F985,#00FBFF);  
    margin:100px;  
    padding:100px;  
    border-radius: 10px;  
    font-size:25px;  
}  
  
table {  
    font-size: 30px;  
}  
  
input {  
    font-size:25px;  
    margin: 5%;  
}  
  
select {  
    font-size: 25px;  
    margin:5%;  
}  
  
input[type=submit] {  
    border: 2px solid black;  
    height: 10%;  
    width: 25%;
```

```

font-size: 20px;
text-align: center;
padding: 14px 28px;
margin: 10%;
background-color: #79d2e6;
border-radius: 10%
}
input[type=submit]:hover {
background-color: red;
}

```

2)secondpage2.html

```

<!DOCTYPE html>
<html>
<head>
    <title>HOME</title>
    <link rel="stylesheet" href="{{url_for('static',filename
='secondpage2.css')}}" type="text/css">
</head>
    <body>
        <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>😞 sorry,you are not placed 😞 </h1>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```
</div>
```

```
</section>
```

```
</body>
```

```
</html>
```

secondpage2.css

```
* {
```

```
    box-sizing: border-box;
```

```
    font-family: Verdana;
```

```
}
```

```
body {
```

```
    margin: 0;
```

```
    background-image: url("1.png");
```

```
    background-repeat: no-repeat;
```

```
    background-attachment: fixed;
```

```
    background-size: cover;
```

```
}
```

```
h1{
```

```
    margin: 25% 55%;
```

```
    margin-right: 0px;}
```

app.py

```
from flask import Flask,render_template,request

app=Flask(__name__, static_folder='static')

import pickle

model=pickle.load(open("placement.pkl",'rb'))

@app.route('/')

def hello():

    return render_template("index.html")

@app.route('/guest')

def guest():

    return render_template("index1.html")

@app.route('/y_predict',methods=["POST"])

def y_predict():

    sen1=request.form["sen1"]

    sen2=request.form["sen2"]

    sen3=request.form["sen3"]

    sen4=request.form["sen4"]

    sen5=request.form["sen5"]

    sen6=request.form["sen6"]

    value=[[int(sen1),int(sen2),int(sen3),int(sen4),int(sen5),int(sen6)]]

    #x_test=[[yo) for yo in request.form.values()]]

    prediction=model.predict(value)

    prediction=prediction[0]

    if prediction==1:

        return render_template("secondpage.html")

    else:

        return render_template("secondpage2.html")

if __name__=='__main__':

    app.run(debug=True)
```

CAMPUSPLACEMENT.ipynb

CAMPUSPLACEMENT.ipynb

File Edit View Insert Runtime Tools Help Saving...

+ Code + Text

RAM
Disk

✓

⬆

⬇

from logging import warning
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
import joblib
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report, confusion_matrix
import warnings
warnings.filterwarnings('ignore')

Read the dataset

[2] df=pd.read_csv('/content/collegePlace.xls')
df.head()

1	21	Female	Computer Science	0	7	1	1	1
2	22	Female	Information Technology	1	6	0	0	1
3	21	Male	Information Technology	0	8	0	1	1
4	22	Male	Mechanical	0	8	1	0	1

CAMPUSPLACEMENT.ipynb

File Edit View Insert Runtime Tools Help Saving...

+ Code + Text

RAM
Disk

✓

⬆

⬇

[2]

1	21	Female	Computer Science	0	7	1	1	1
2	22	Female	Information Technology	1	6	0	0	1
3	21	Male	Information Technology	0	8	0	1	1
4	22	Male	Mechanical	0	8	1	0	1

Data Preparation

1)Handling Missing Values

df.info()

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 2966 entries, 0 to 2965  
Data columns (total 8 columns):  
#   Column          Non-Null Count  Dtype  
---  ---  
0   Age             2966 non-null   int64  
1   Gender          2966 non-null   object  
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
```


CAMPUSPLACEMENT.ipynb

File Edit View Insert Runtime Tools Help All changes saved

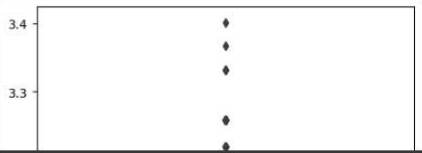
+ Code + Text

df.isnull().sum()

Age 0
Gender 0
Stream 0
Internships 0
CGPA 0
Hostel 0
HistoryOfBacklogs 0
PlacedOrNot 0
dtype: int64

2)Handling Outliers

def transformation(feature):
plt.figure(figsize=(12,5))
plt.subplot(1,2,1)
sns.boxplot(feature)
transformation(np.log(df['Age']))



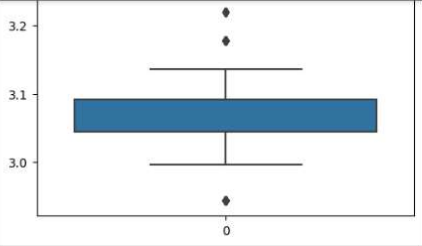
A box plot showing the distribution of Age. The y-axis ranges from 3.3 to 3.4. The plot shows a median around 3.35, with several outliers above the upper whisker.

CAMPUSPLACEMENT.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

[5] plt.figure(figsize=(12,5))
plt.subplot(1,2,1)
sns.boxplot(feature)
transformation(np.log(df['Age']))



A box plot showing the distribution of Age. The y-axis ranges from 3.0 to 3.2. The plot shows a median around 3.08, with several outliers above the upper whisker.

3)Handling Categorical Values

[6] df=df.replace(['Male'],[0])
df=df.replace(['Female'],[1])
df=df.replace(['Civil','Computer Science','Electrical','Electronics And Communication','Information Technology','Mechanical'],[0,1,2,3,4,5])

df=df.drop(['Hostel'],axis=1)
df

	Age	Gender	Stream	Internships	CGPA	HistoryOfBacklogs	PlacedOrNot
0	22	0	3	1	8	1	1

CAMPUSPLACEMENT.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[7] print(df['Age'].value_counts(),'\n-----\n',
df['Stream'].value_counts(),'\n-----\n',
df['Internships'].value_counts(),'\n-----\n',
df['CGPA'].value_counts(),'\n-----\n',
df['HistoryOfBacklogs'].value_counts(),'\n-----\n',
df['PlacedOrNot'].value_counts())

21	1084
22	941
20	375
23	195
19	156

Completed at 7:51 AM

```
CAMPUSPLACEMENT.ipynb
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text
24 131
26 50
25 29
28 3
30 1
29 1
Name: Age, dtype: int64
-----
1 776
4 691
3 424
5 424
2 334
0 317
Name: Stream, dtype: int64
-----
0 1331
1 1234
2 350
3 51
Name: Internships, dtype: int64
-----
7 956
8 915
6 834
9 165
5 96
Name: CGPA, dtype: int64
-----
0 2396
1 570
Name: HistoryOfBacklogs, dtype: int64
-----
1 1639
0 1327
```

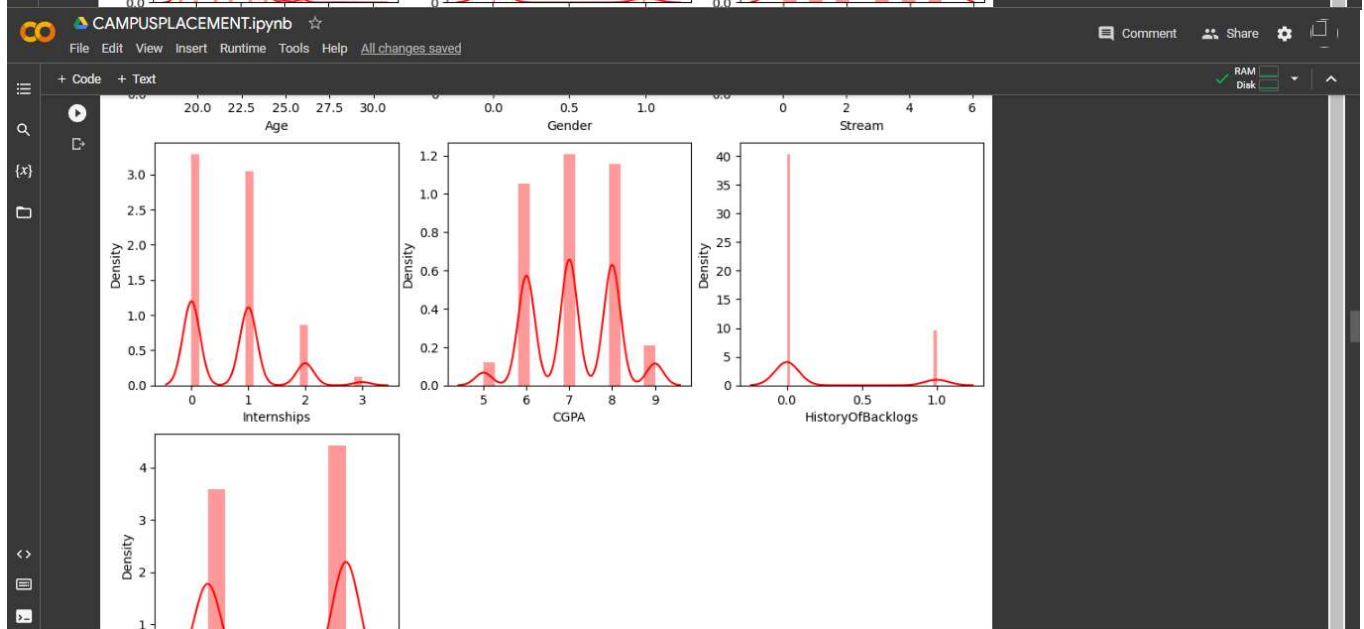
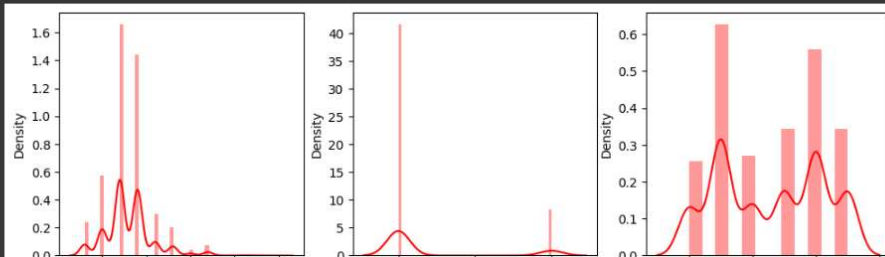
```
CAMPUSPLACEMENT.ipynb
File Edit View Insert Runtime Tools Help All changes saved

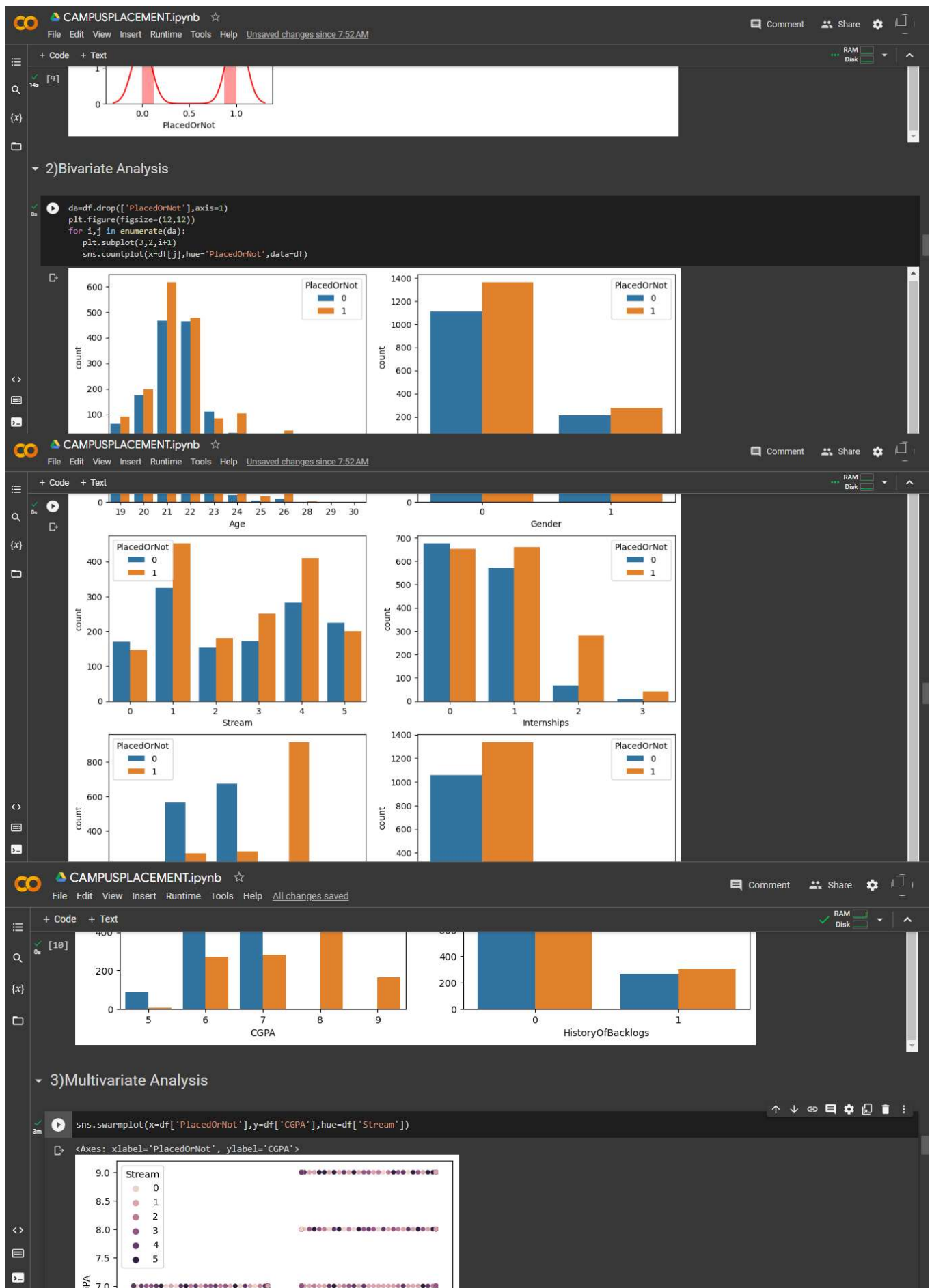
+ Code + Text
1 1639
0 1327
Name: PlacedOrNot, dtype: int64
```

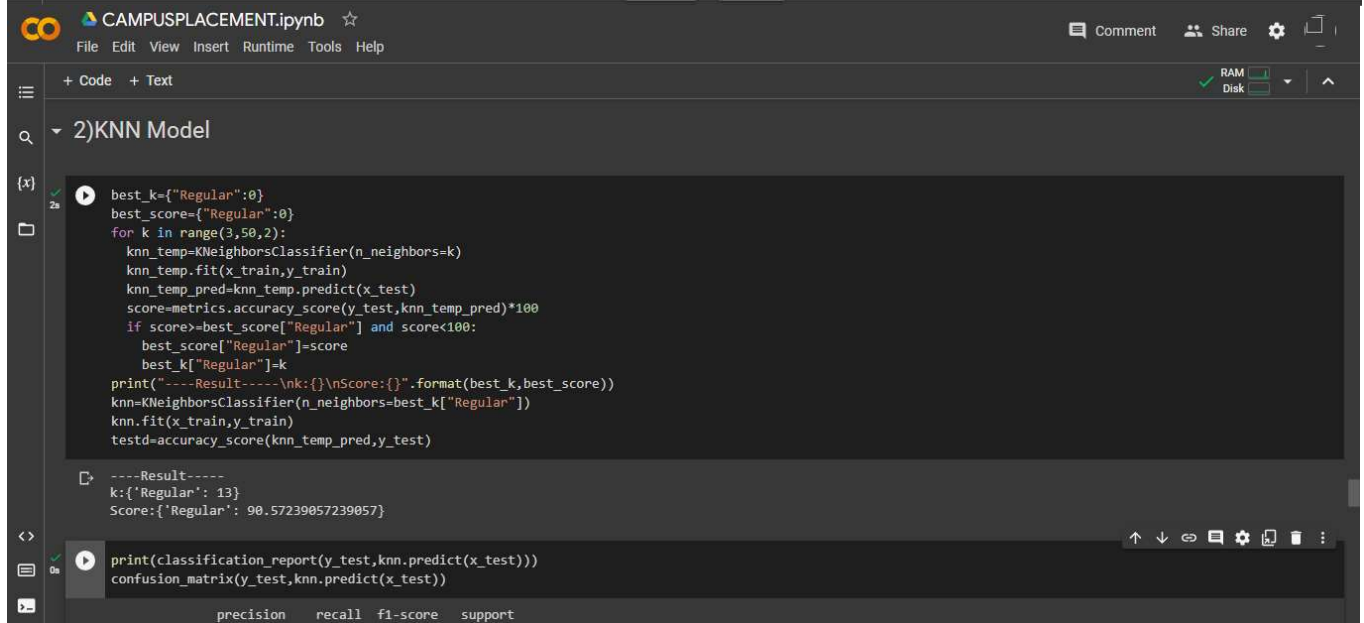
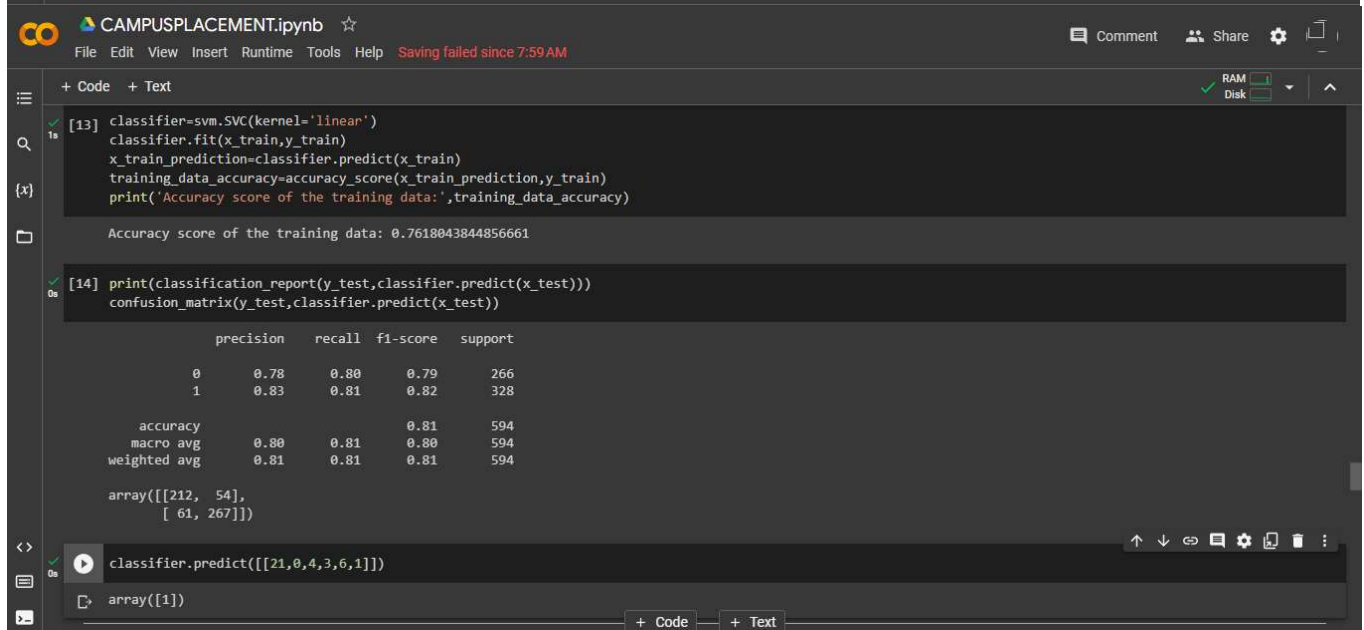
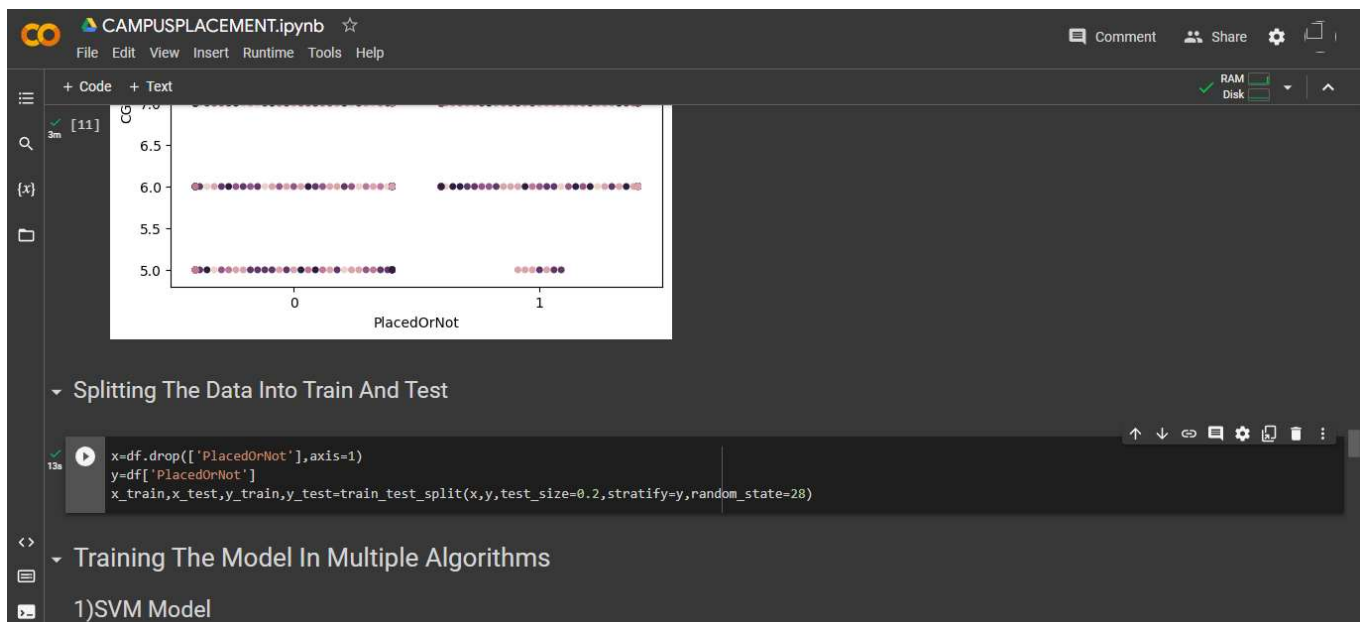
Visual Analysis

1)Univariate Analysis

```
plt.figure(figsize=(12,12))
for i,j in enumerate(df):
    plt.subplot(3,3,i+1)
    sns.distplot(df[j],color='r')
```







CAMPUSPLACEMENT.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM Disk

0s

[17]

0	0.86	0.94	0.90	266
1	0.95	0.88	0.91	328
accuracy			0.91	594
macro avg	0.90	0.91	0.91	594
weighted avg	0.91	0.91	0.91	594

array([[251, 15],
[41, 287]])

2s

[18]

knn.predict([[21,0,4,3,6,1]])

array([1])

3)Artificial Neural Network Model(ANN)

44s

```
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))
```

CAMPUSPLACEMENT.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM Disk

44s

```
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)
```

119/119 [=====] - 0s 2ms/step - loss: 0.6388 - accuracy: 0.6674

Epoch 73/100

119/119 [=====] - 0s 2ms/step - loss: 0.6427 - accuracy: 0.6564

Epoch 74/100

119/119 [=====] - 0s 2ms/step - loss: 0.6578 - accuracy: 0.6400

Epoch 75/100

119/119 [=====] - 0s 2ms/step - loss: 0.6441 - accuracy: 0.6573

Epoch 76/100

119/119 [=====] - 0s 2ms/step - loss: 0.6450 - accuracy: 0.6526

Epoch 77/100

119/119 [=====] - 0s 2ms/step - loss: 0.6398 - accuracy: 0.6728

Epoch 78/100

119/119 [=====] - 0s 2ms/step - loss: 0.6444 - accuracy: 0.6619

Epoch 79/100

119/119 [=====] - 0s 2ms/step - loss: 0.6403 - accuracy: 0.6632

Epoch 80/100

119/119 [=====] - 0s 2ms/step - loss: 0.6413 - accuracy: 0.6619

Epoch 81/100

119/119 [=====] - 0s 2ms/step - loss: 0.6461 - accuracy: 0.6403

CAMPUSPLACEMENT.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

RAM Disk

+ Code + Text

Epoch 92/100
119/119 [-----] - 0s 3ms/step - loss: 0.6443 - accuracy: 0.6556
Epoch 93/100
119/119 [-----] - 0s 3ms/step - loss: 0.6439 - accuracy: 0.6644
Epoch 94/100
119/119 [-----] - 0s 3ms/step - loss: 0.6431 - accuracy: 0.6589
Epoch 95/100
119/119 [-----] - 0s 3ms/step - loss: 0.6497 - accuracy: 0.6522
Epoch 96/100
119/119 [-----] - 0s 2ms/step - loss: 0.6384 - accuracy: 0.6686
Epoch 97/100
119/119 [-----] - 0s 2ms/step - loss: 0.6488 - accuracy: 0.6560
Epoch 98/100
119/119 [-----] - 0s 2ms/step - loss: 0.6438 - accuracy: 0.6627
Epoch 99/100
119/119 [-----] - 0s 2ms/step - loss: 0.6469 - accuracy: 0.6636
Epoch 100/100
119/119 [-----] - 0s 2ms/step - loss: 0.6421 - accuracy: 0.6560
<keras.callbacks.History at 0x7f3ec017cf70>

classifier.predict([[21,0,4,3,6,1]])

1/1 [-----] - 0s 119ms/step
array([[0.63559717]], dtype=float32)

CAMPUSPLACEMENT.ipynb

File Edit View Insert Runtime Tools Help Saving...

Comment Share

RAM Disk

+ Code + Text

Model Deployment

Model Deployment

import pickle
pickle.dump(knn,open("placement.pkl","wb"))

Files

sample_data
collegePlace.xls
placement.pkl