

THIRUVALLUVAR UNIVERSITY
PERIYAR ARTS COLLEGE
CUDDALORE – 607001.



DEPARTMENT OF COMPUTER APPLICATIONS

MACHINE LEARNING WITH PYTHON

Project Title : Identifying Patterns and Trends in Campus Placement Data using Machine Learning

Team Id : NM2023TMID22863

Team Leader : AJITHKUMAR V(DAA22430EAEF00786B142C928B2E5642)

Team member: DHANALAKSHMI N(D07AF99278721CD9C364C0E15DAA6F8C)

Team member: MANONMANI P(453D00CE5432D228A70488876DA2555D)

Team member: JEGANNATHAN M(C9D82D74B8BA69C8C984E1DFB2A2A1BE)

INTRODUCTION:

OVERVIEW:

The placement both for final jobs and summer internships is an integral part of any institute's annual calendar of activities.

Campus placement is hiring young talent for internships and entry level positions.

PURPOSE:

The companies will be benefited from getting wide choice of candidates to select for different job posts. Companies can select the right and talented candidate from a vast pool of young applicants within a limited

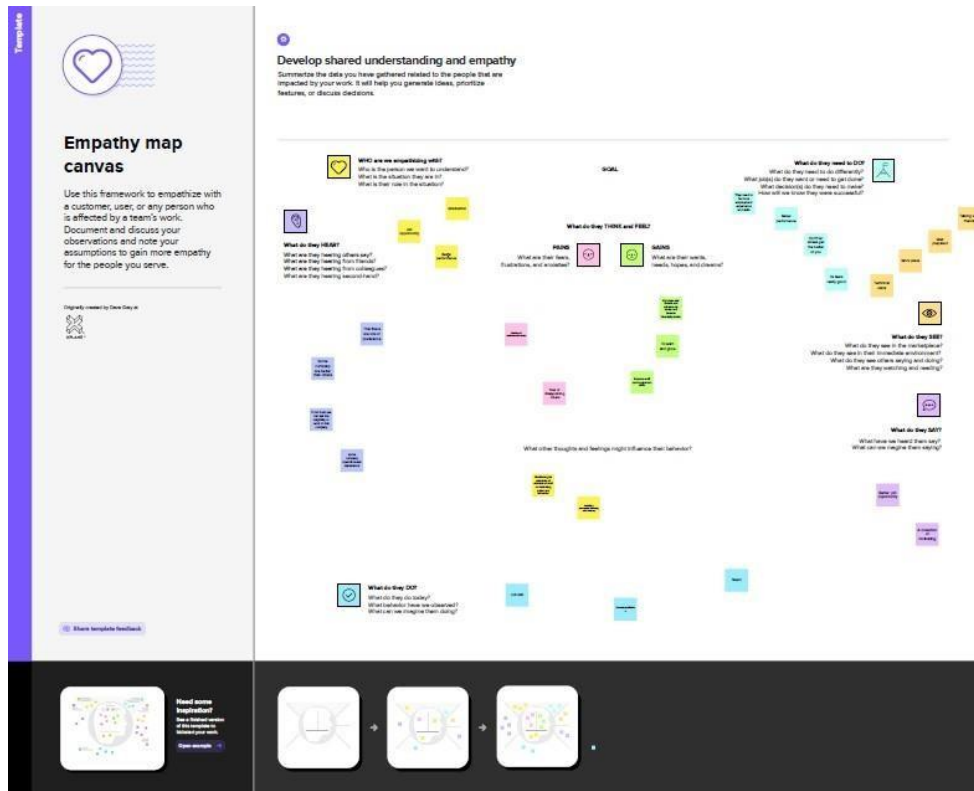
time. On the other hand, students have the advantage of getting a good job according to their qualification level even before the completion of their academic course in college.

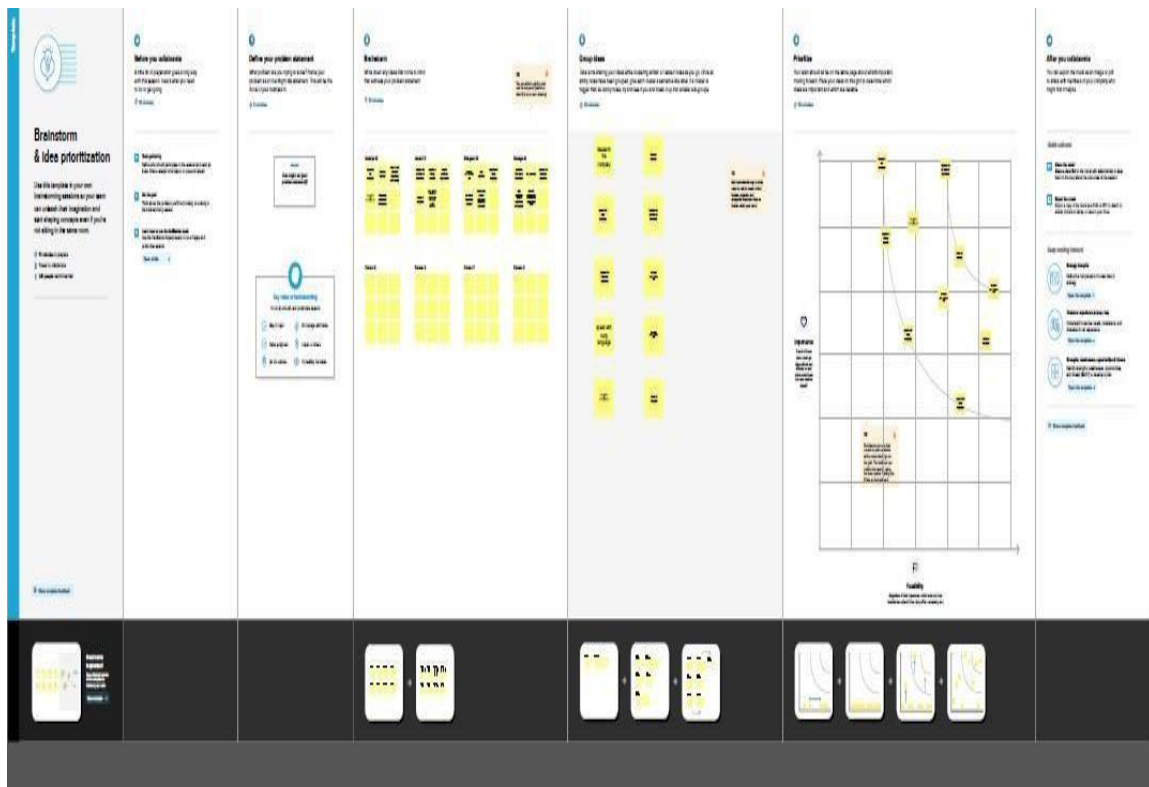
Campus placement or campus recruiting is a program conducted within universities or other educational institutions to provide jobs to students nearing completion of their studies.

In this type of program the educational institutions partner with corporations who wish to recruit from the student population.

PROBLEM DEFINITION AND DESIGN

THINKING EMPATHY MAP



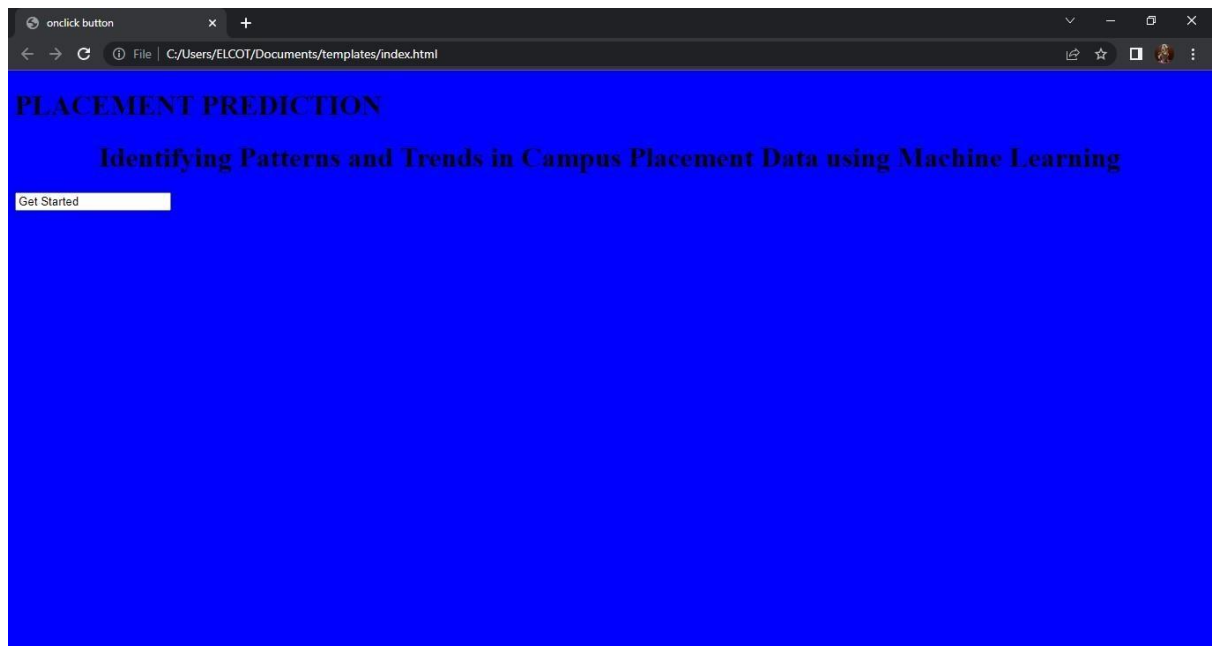


IDEATION & BRAINSTROMING MAP

RESULT

Final finding output of the project

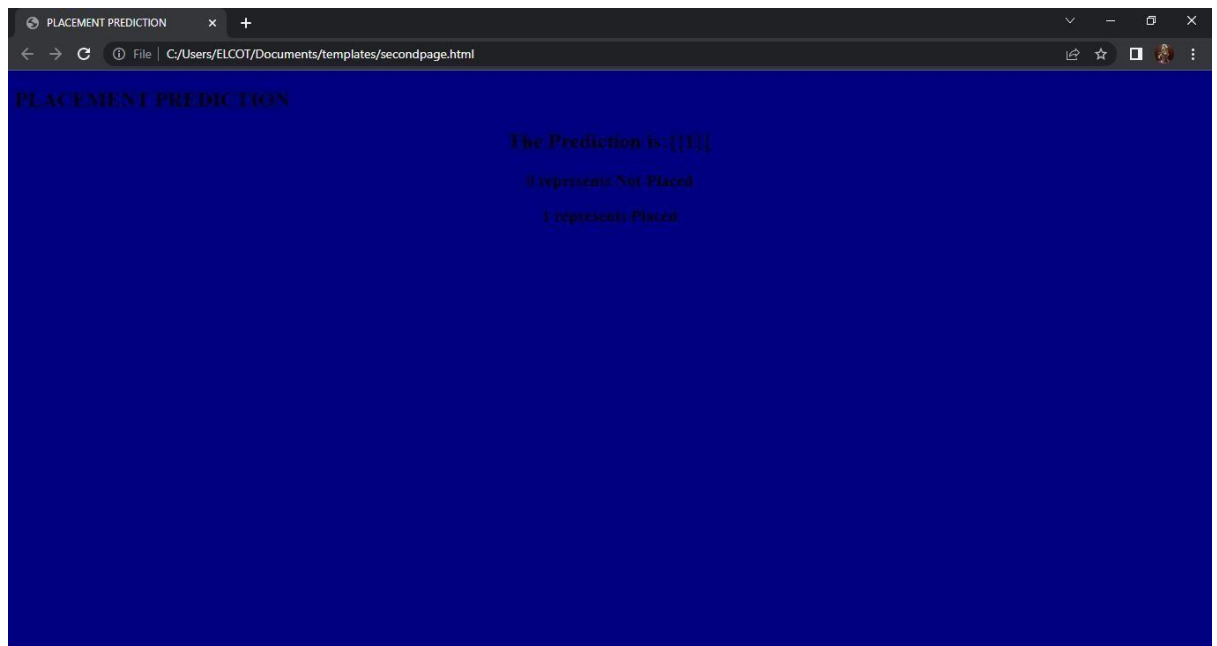
○ Index.html



○ Index1.html

A screenshot of a web browser window. The address bar shows the file path 'C:/Users/ELCOT/Documents/templates/index1.html'. The page has a white background. At the top, the text 'FILL THE DETAILS' is displayed in a bold, black serif font. Below this text, there are seven input fields arranged vertically. The first five are text boxes containing the numbers '22', '0', '2', '1', and '8' respectively. The sixth is a dropdown menu with '1' selected. The seventh is a button labeled 'Submit'.

○ Second page.html



ADVANTAGE OF CAMPUS PLACEMENT:

- The companies will be benefited from getting wide choice of candidates to select for different jobs.
- The companies will be benefited from getting wide choice of candidates to select for different job posts.

- Companies can select the right and talented candidate from a vast pool of young applicants within a limited time.
- On the other hand, students have the advantage of getting a good job according to their qualification level even before the completion of their academic course in college.

DISADVANTAGES OF CAMPUS PLACEMENT:

- Campus recruitment is an expensive affair for majority of the companies as it adds up costs to the bottom line.
- Companies incur different expenses related to travel, boarding, training

etc while conducting campus selection process.

- The experienced and skilled candidates having practical job exposures cannot be recruited through campus placements.
- Fresh candidates selected through campus placements require adequate training for work.
- This is an additional expense for the company. Also, students can't work with their dream company and will have to remain satisfied with the company that recruits them during campus selection. **APPLICATION**
- Companies hold on campus recruitment drives for students in their final year, and sever large. ○ You can expect questions related to

coding, algorithm and machine learning.

CONCLUSION

- Goal for future placement
- Performance of student
 - At the completion of placement, student and supervisors should complete the end of placement evaluation form.
 - To determine what merits satisfactory or unsatisfactory performance on placement.

FUTURE SCOPE

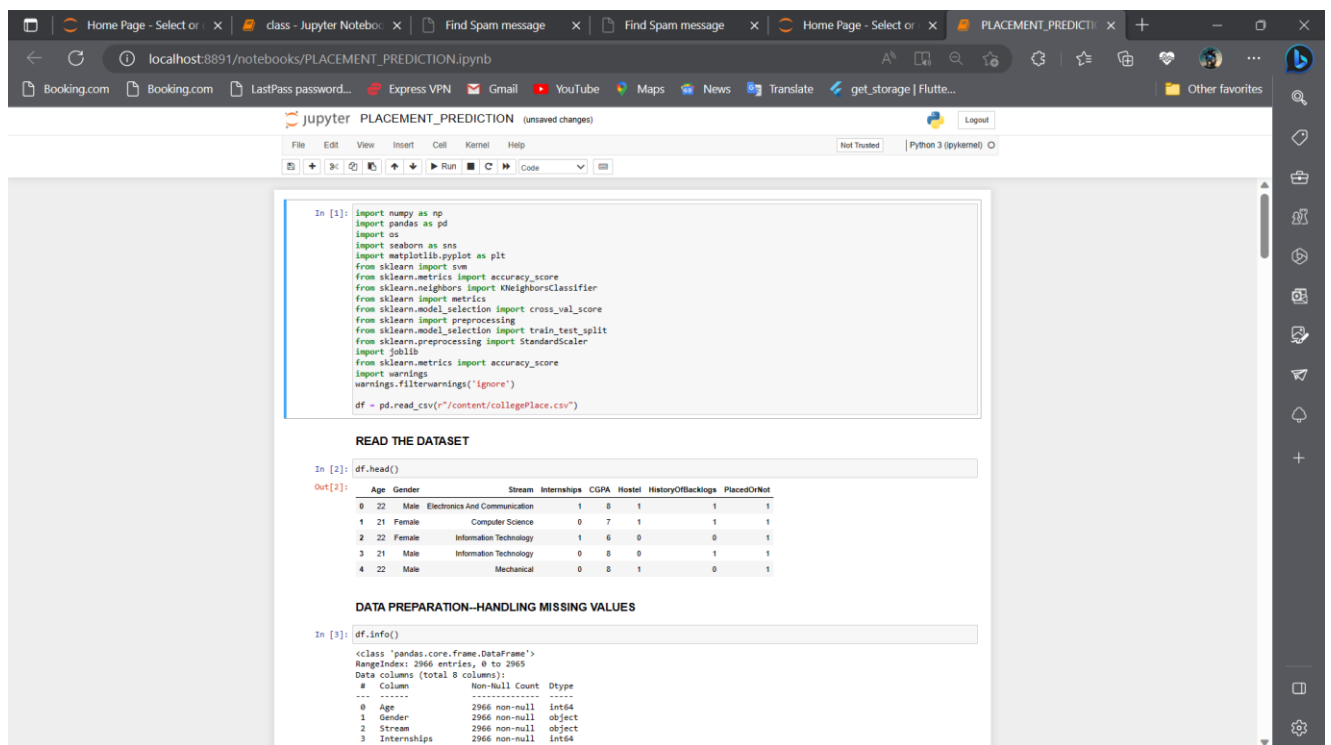
- Given the boom in the market, college grads or undergrads have access to a wide pool of employers promising attractive roles and benefits.

- We need to use technology and creative communication strategies to stay top of mind in our target demographic.

APPENDIX

Source code

PLACEMENT_PREDICTION.ipynb



```
In [1]: 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
import warnings
warnings.filterwarnings('ignore')
df = pd.read_csv(r"/content/collegePlace.csv")
```

READ THE DATASET

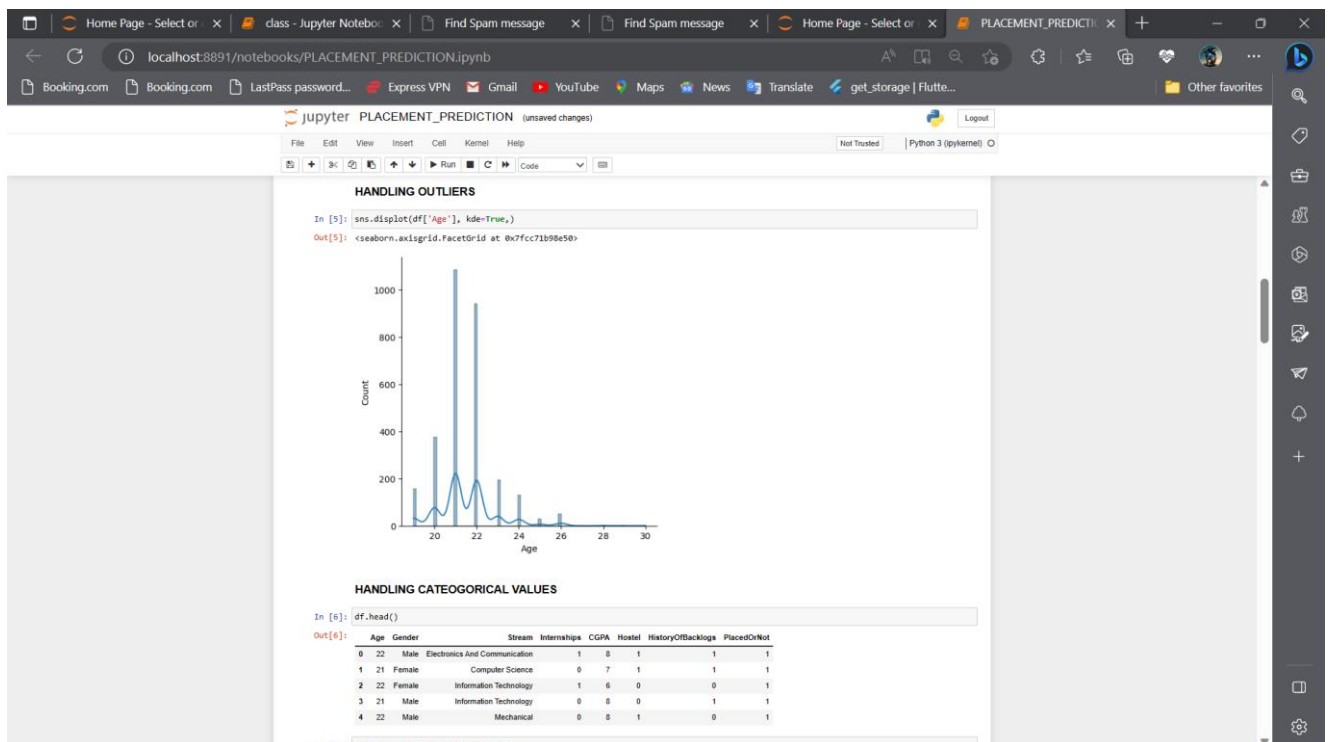
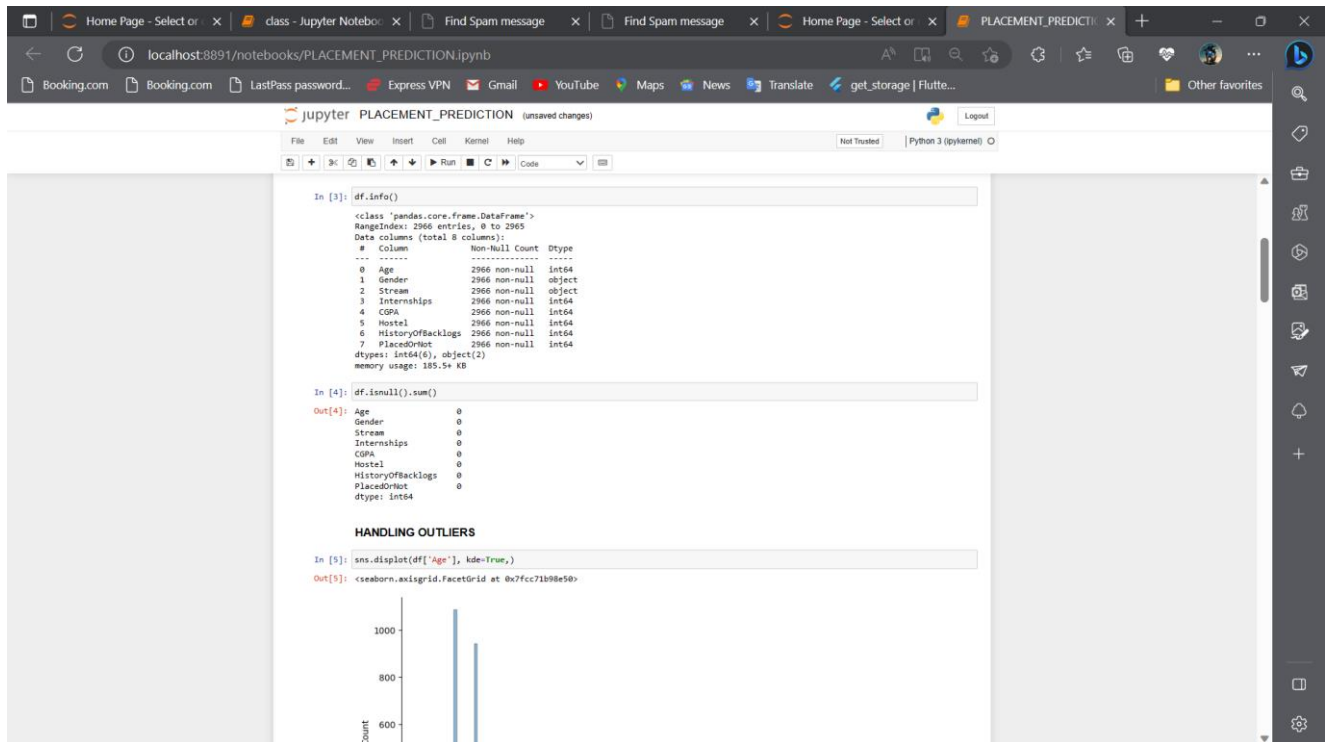
```
In [2]: df.head()
```

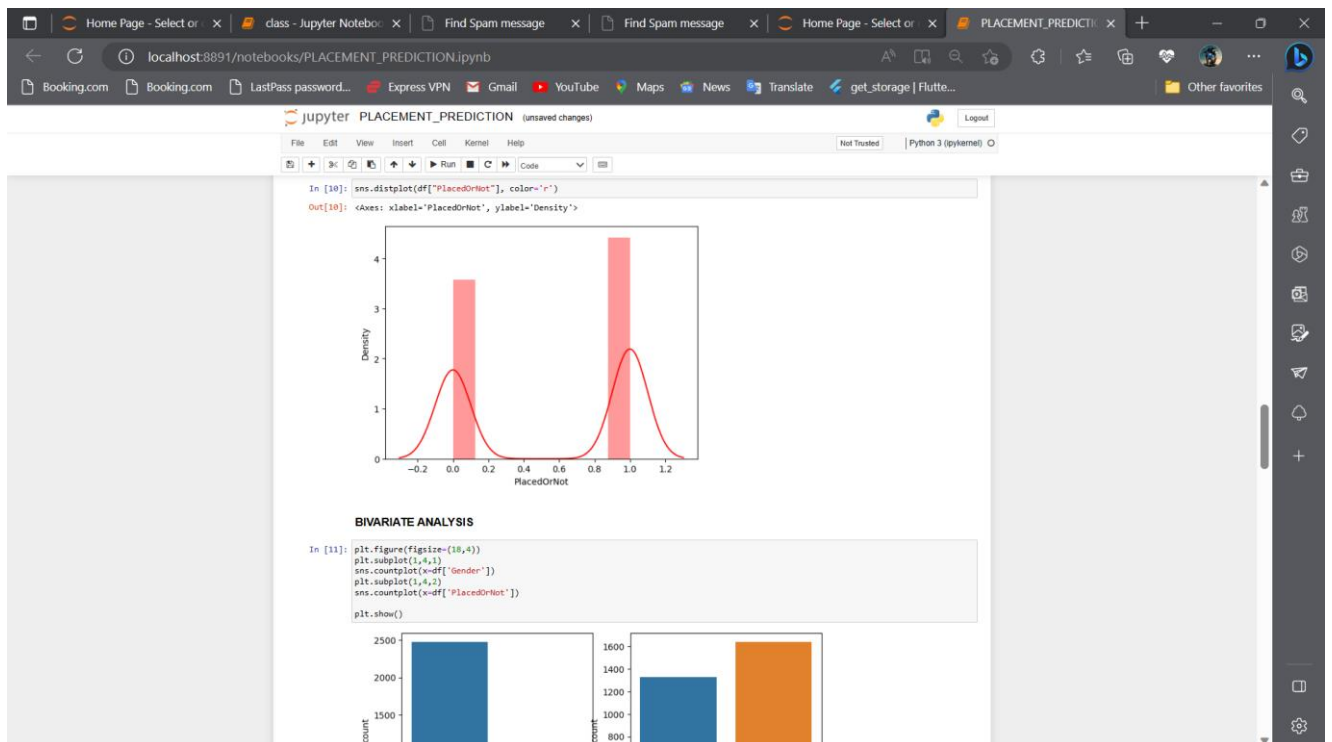
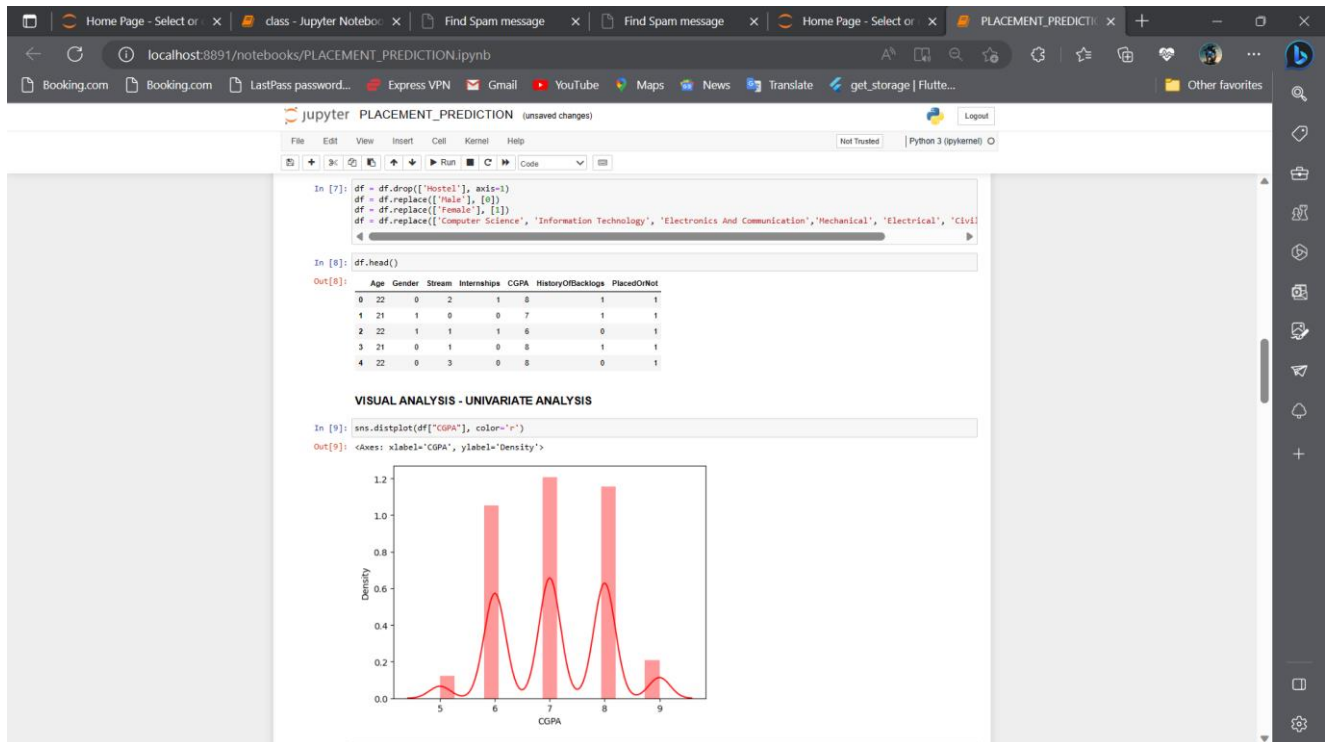
	Age	Gender	Stream	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot
0	22	Male	Electronics And Communication	1	8	1	1	1
1	21	Female	Computer Science	0	7	1	1	1
2	22	Female	Information Technology	1	6	0	0	1
3	21	Male	Information Technology	0	8	0	1	1
4	22	Male	Mechanical	0	8	1	0	1

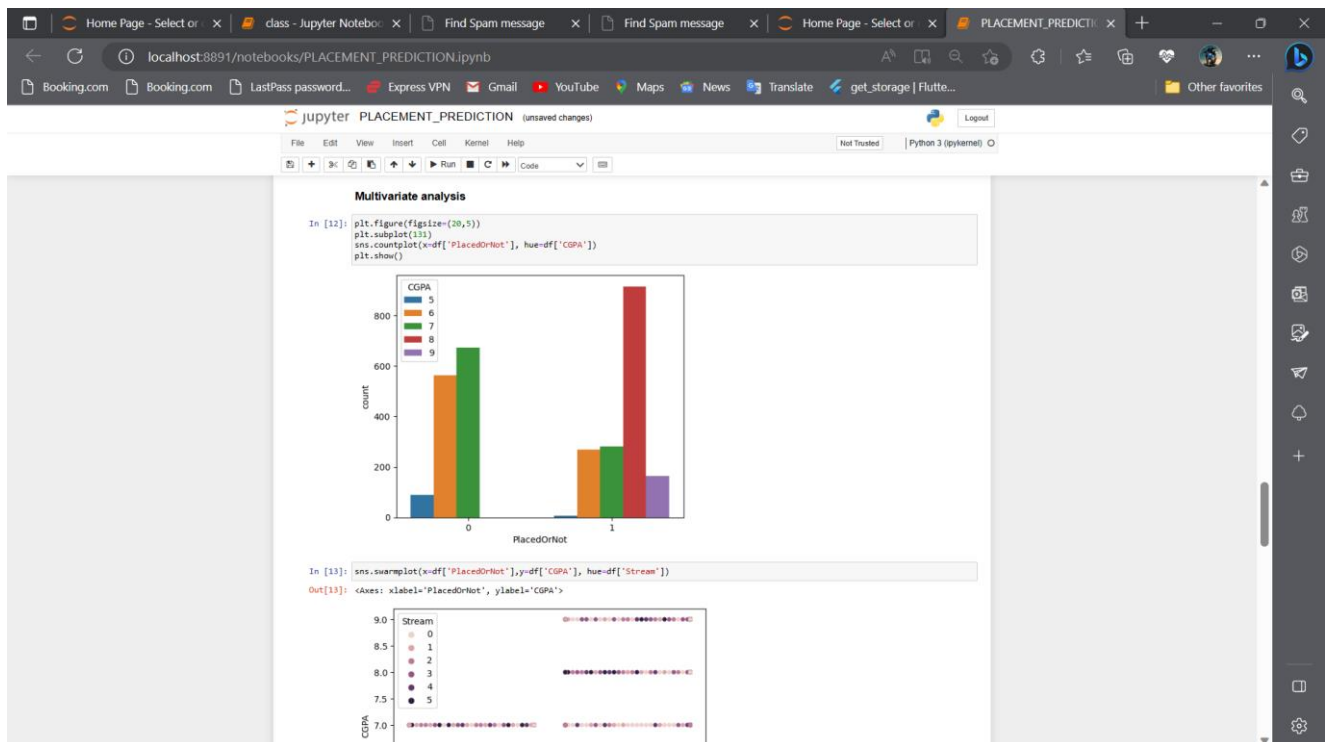
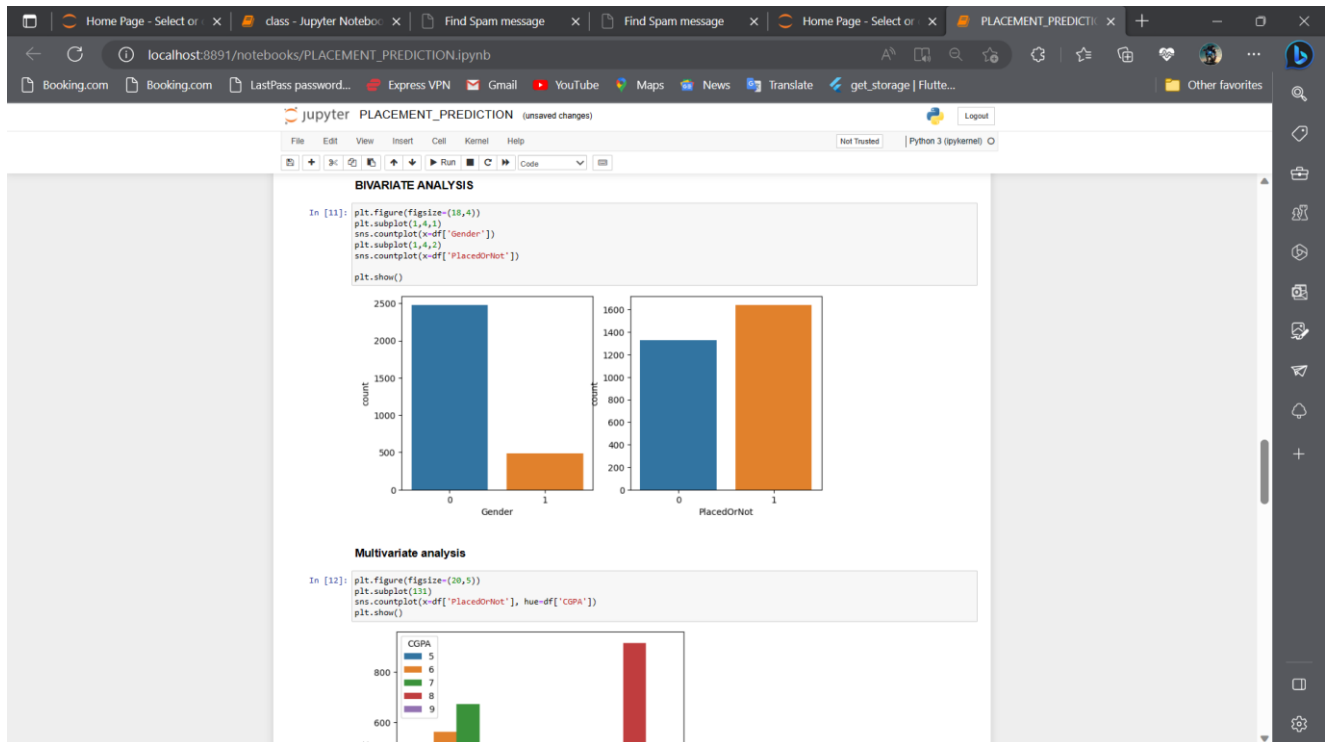
DATA PREPARATION--HANDLING MISSING VALUES

```
In [3]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2966 entries, 0 to 2965
Data columns (total 9 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   float64
```







Home Page - Select or x class - Jupyter Notebo x Find Spam message x Find Spam message x Home Page - Select or x PLACEMENT_PREDICTI x + -

localhost:8891/notebooks/PLACEMENT_PREDICTION.ipynb

Booking.com Booking.com LastPass password... Express VPN Gmail YouTube Maps News Translate get_storage | Flutter... Other favorites

jupyter PLACEMENT_PREDICTION (unsaved changes) Logout

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (pykernel)

SCALING THE DATA

```
In [14]: Y=df[['PlacedOnHot']]
X=df.drop(['PlacedOnHot'],axis=1)

In [15]: X
```

	Age	Gender	Stream	Internships	CGPA	HistoryOfBacklogs
0	22	0	2	1	8	1
1	21	1	0	0	7	1
2	22	1	1	1	6	0
3	21	0	1	0	8	1
4	22	0	3	0	8	0
...
2961	23	0	1	0	7	0
2962	23	0	3	1	7	0
2963	22	0	1	1	7	0
2964	22	0	0	1	7	0
2965	23	0	5	0	8	0

2966 rows x 6 columns

SPLITTING THE DATA INTO TRAIN AND TEST

```
In [16]: X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size =0.2, stratify=Y, random_state=2 )
```

SVM MODEL

```
In [17]: classifier = svm.SVC(kernel='linear')

In [18]: classifier.fit(X_train, Y_train)

Out[18]: SVC(kernel='linear')
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [19]: X_train_prediction = classifier.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
```

Home Page - Select or x class - Jupyter Notebo x Find Spam message x Find Spam message x Home Page - Select or x PLACEMENT_PREDICTI x + -

localhost:8891/notebooks/PLACEMENT_PREDICTION.ipynb

Booking.com Booking.com LastPass password... Express VPN Gmail YouTube Maps News Translate get_storage | Flutter... Other favorites

jupyter PLACEMENT_PREDICTION (unsaved changes) Logout

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (pykernel)

SVM MODEL

```
In [17]: classifier = svm.SVC(kernel='linear')

In [18]: classifier.fit(X_train, Y_train)

Out[18]: SVC(kernel='linear')
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [19]: X_train_prediction = classifier.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)

In [20]: print('Accuracy score of the training data:',training_data_accuracy)
Accuracy score of the training data: 0.7685497470489039
```

KNN MODEL

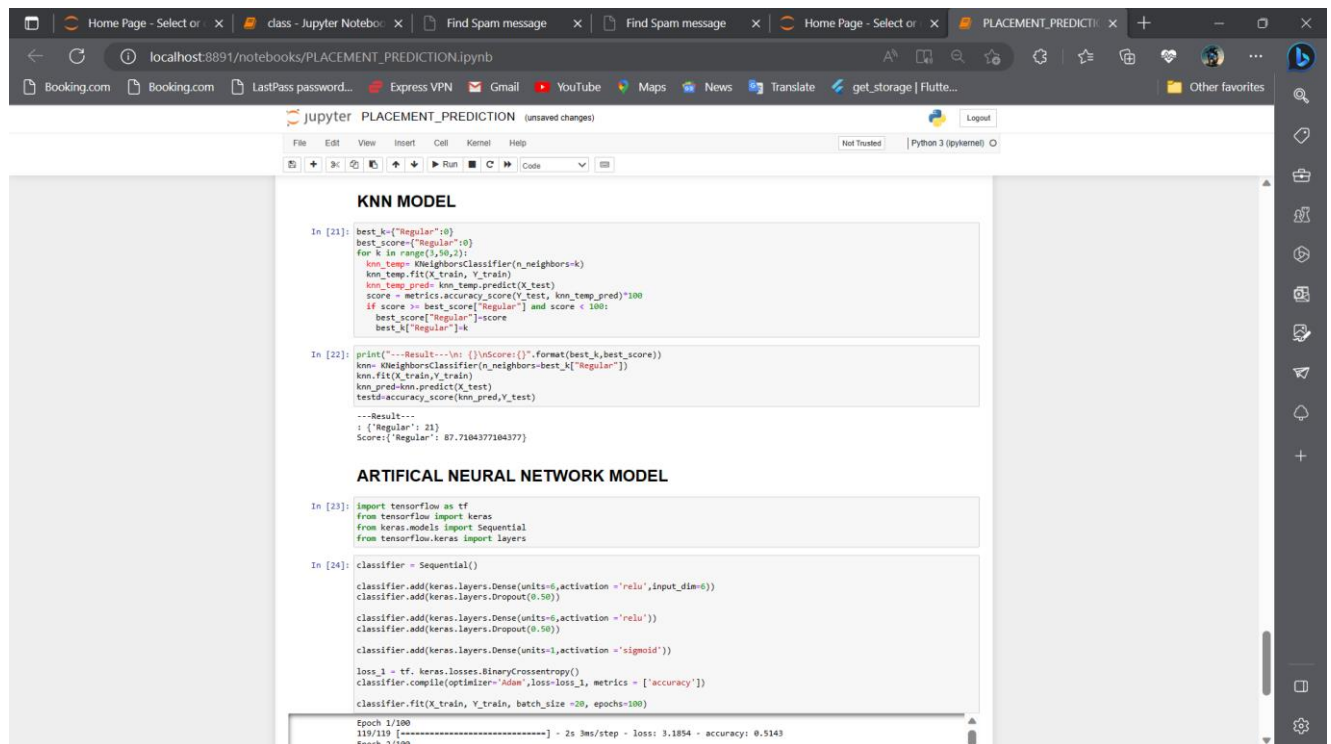
```
In [21]: best_k={"Regular":0}
best_score={"Regular":0}
for k in range(1,50,2):
    knn_temp= KNeighborsClassifier(n_neighbors=k)
    knn_temp.fit(X_train, Y_train)
    knn_temp_pred= knn_temp.predict(X_test)
    score = metrics.accuracy_score(Y_test, knn_temp_pred)*100
    if score >= best_score["Regular"] and score < 100:
        best_score["Regular"]=score
        best_k["Regular"]=k

In [22]: print("====Result====:")
knn= KNeighborsClassifier(n_neighbors=best_k["Regular"])
knn.fit(X_train,Y_train)
knn_pred=knn.predict(X_test)
test_acc=accuracy_score(knn_pred,Y_test)

---Result---
: ('Regular': 21)
Score: ('Regular': 87.7104377104377)
```

ARTIFICIAL NEURAL NETWORK MODEL

```
In [23]: import tensorflow as tf
from tensorflow import keras
```



The screenshot shows a Jupyter Notebook interface with the title 'PLACEMENT_PREDICTION'. The notebook contains two sections: 'KNN MODEL' and 'ARTIFICIAL NEURAL NETWORK MODEL'.

KNN MODEL

```
In [21]: best_k={"Regular":0}
best_score={"Regular":0}
for k in range(1,50,1):
    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

In [22]: print("---Result---\n: {}".format(best_k,best_score))
knn = KNeighborsClassifier(n_neighbors=best_k["Regular"])
knn.fit(X_train,Y_train)
knn_pred=knn.predict(X_test)
testid=accuracy_score(knn_pred,Y_test)

---Result---
: {'Regular': 21}
Score: {'Regular': 87.71843771043777}
```

ARTIFICIAL NEURAL NETWORK MODEL

```
In [23]: import tensorflow as tf
from tensorflow import keras
from keras.models import Sequential
from tensorflow.keras import layers

In [24]: classifier = Sequential()

classifier.add(keras.layers.Dense(units=6,activation='relu',input_dim=6))
classifier.add(keras.layers.Dropout(0.50))

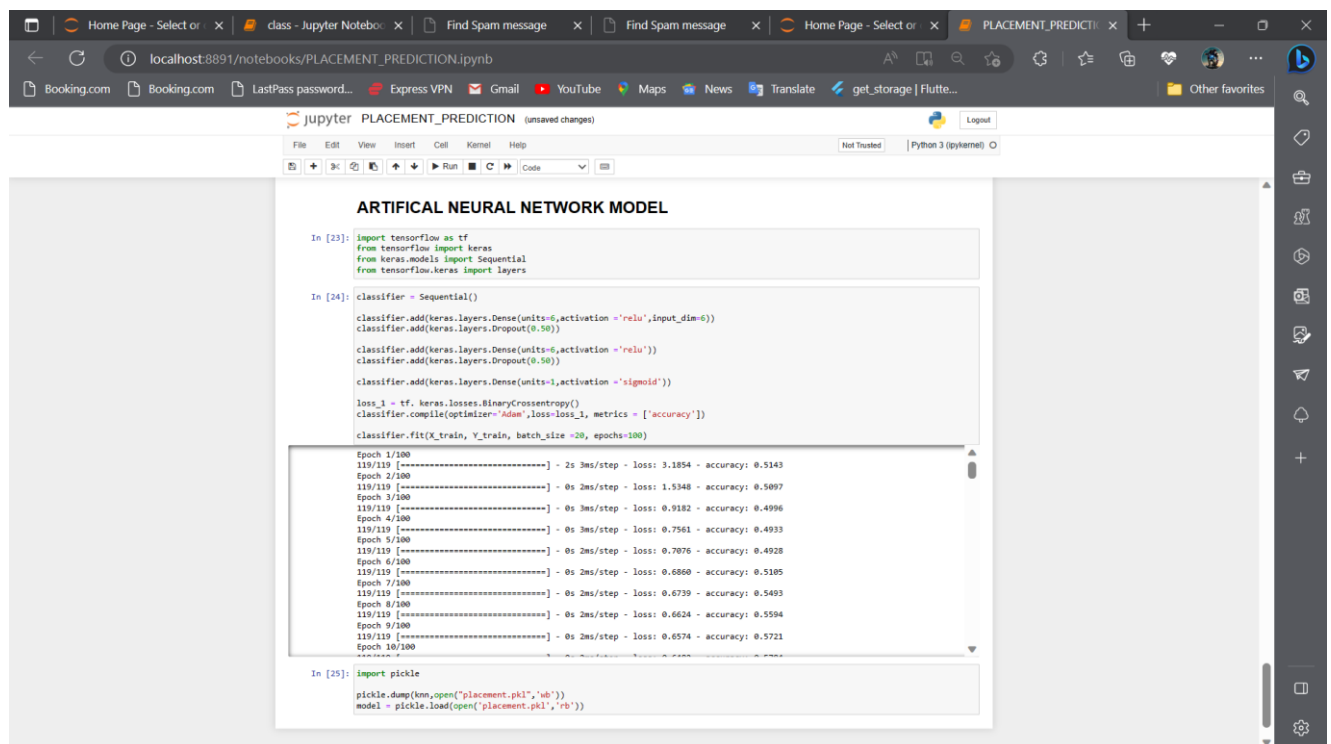
classifier.add(keras.layers.Dense(units=6,activation='relu'))
classifier.add(keras.layers.Dropout(0.50))

classifier.add(keras.layers.Dense(units=1,activation='sigmoid'))

loss_l = tf.keras.losses.BinaryCrossentropy()
classifier.compile(optimizer='Adam',loss=loss_l,metrics=['accuracy'])

classifier.fit(X_train, Y_train, batch_size=20, epochs=100)

Epoch 1/100
119/119 [=====] - 2s 3ms/step - loss: 3.1854 - accuracy: 0.5143
Epoch 2/100
```



The screenshot continues the Jupyter Notebook from the previous one, showing the training progress of the Artificial Neural Network model and saving it as a pickle file.

ARTIFICIAL NEURAL NETWORK MODEL

```
In [23]: import tensorflow as tf
from tensorflow import keras
from keras.models import Sequential
from tensorflow.keras import layers

In [24]: classifier = Sequential()

classifier.add(keras.layers.Dense(units=6,activation='relu',input_dim=6))
classifier.add(keras.layers.Dropout(0.50))

classifier.add(keras.layers.Dense(units=6,activation='relu'))
classifier.add(keras.layers.Dropout(0.50))

classifier.add(keras.layers.Dense(units=1,activation='sigmoid'))

loss_l = tf.keras.losses.BinaryCrossentropy()
classifier.compile(optimizer='Adam',loss=loss_l,metrics=['accuracy'])

classifier.fit(X_train, Y_train, batch_size=20, epochs=100)

Epoch 1/100
119/119 [=====] - 2s 3ms/step - loss: 3.1854 - accuracy: 0.5143
Epoch 2/100
119/119 [=====] - 0s 2ms/step - loss: 1.5348 - accuracy: 0.5097
Epoch 3/100
119/119 [=====] - 0s 3ms/step - loss: 0.9182 - accuracy: 0.4996
Epoch 4/100
119/119 [=====] - 0s 3ms/step - loss: 0.7561 - accuracy: 0.4933
Epoch 5/100
119/119 [=====] - 0s 2ms/step - loss: 0.7076 - accuracy: 0.4928
Epoch 6/100
119/119 [=====] - 0s 2ms/step - loss: 0.6860 - accuracy: 0.5105
Epoch 7/100
119/119 [=====] - 0s 2ms/step - loss: 0.6739 - accuracy: 0.5403
Epoch 8/100
119/119 [=====] - 0s 2ms/step - loss: 0.6624 - accuracy: 0.5594
Epoch 9/100
119/119 [=====] - 0s 2ms/step - loss: 0.6574 - accuracy: 0.5721
Epoch 10/100

In [25]: import pickle

pickle.dump(knn,open("placement.pkl",'wb'))
model = pickle.load(open("placement.pkl",'rb'))
```

About.html


```
*about.html - Notepad
File Edit View

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Prediction</title>
    <meta name="description" content="">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="static/styles.css">
    <link href="https://fonts.googleapis.com/css?family=Montserrat:500&display=swap" rel="stylesheet">
  </head>
  <body>
    <header>
      <nav>
        <ul class="nav_links">
          <li><a href="/">Home</a></li>
          <li><a href="/about">About</a></li>
        </ul>
      </nav>
      <a class="cta" href="/predict">Predict</a>
    </header>

    <div class="idp-p">
      <p>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.
    </p>
    <p>
      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.
    </p>
    </div>
    <div class="idp-c">
      <a class="cta" href="/predict">Predict</a>
    </div>
  </body>
</html>

Ln 37, Col 99 100% Windows (CRLF) UTF-8
```

Home.html

```
home.html - Notepad
File Edit View

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Navbar</title>
    <meta name="description" content="">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="static/styles.css">
    <link href="https://fonts.googleapis.com/css?family=Montserrat:500&display=swap" rel="stylesheet">
  </head>
  <body>
    <header>
      <nav>
        <ul class="nav_links">
          <li><a href="/">Home</a></li>
          <li><a href="/about">About</a></li>
        </ul>
      </nav>
      <a class="cta" href="/predict">Predict</a>
    </header>

    <div class="idp-text">
      <h1>IDENTIFYING PATTERNNS AND TRENDS IN CAMPUS PLACEMENT DATA USING MACHINE LEARNING</h1>
    </div>
    <div class="idp-b">
      <a class="cta" href="/predict">Predict</a>
    </div>
  </body>
</html>

Ln 28, Col 45 150% Windows (CRLF) UTF-8
```

Predict.html

```
predict.html - Notepad
File Edit View

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Prediction </title>
    <meta name="description" content="">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="static/styles.css">
    <link href="https://fonts.googleapis.com/css?family=Montserrat:500&display=swap" rel="stylesheet">
  </head>
  <body>
    <div class="header">
      <nav>
        <ul class="nav_links">
          <li><a href="/">Home</a></li>
          <li><a href="/about">About</a></li>
        </ul>
      </nav>
      <a class="cta" href="#">Predict</a>
    </div>
    <div class="login-box">
      <h2>Fill The Details</h2>
      <form action="/pred" method="POST">
        <div class="custom">
          <label class="label-input">Age</label>
          <select class="user-box-op" id="sen1" name="sen1">
            <option value="" selected disabled hidden>Choose your Age</option>
            <option value="22">22</option>
            <option value="23">23</option>
            <option value="24">24</option>
            <option value="25">25</option>
            <option value="26">26</option>
            <option value="27">27</option>
            <option value="28">28</option>
            <option value="29">29</option>
            <option value="30">30</option>
            <option value="31">31</option>
            <option value="32">32</option>
            <option value="33">33</option>
            <option value="34">34</option>
            <option value="35">35</option>
            <option value="36">36</option>
            <option value="37">37</option>
            <option value="38">38</option>
            <option value="39">39</option>
            <option value="40">40</option>
            <option value="41">41</option>
            <option value="42">42</option>
            <option value="43">43</option>
          </select>
        </div>
      </form>
    </div>
  </body>
</html>
```

```
predict.html - Notepad
File Edit View

    <div class="custom">
      <label class="label-input">Stream</label>
      <select class="user-box-op" id="sen3" name="sen3">
        <option value="" selected disabled hidden>Select Education</option>
        <option value="0">Computer Science</option>
        <option value="1">Information Technology</option>
        <option value="2">Electronics And Communication</option>
        <option value="3">Mechanical</option>
        <option value="5">Electrical</option>
        <option value="6">Civil</option>
      </select>
    </div>
    <div class="custom">
      <label class="label-input">Internships</label>
      <select class="user-box-op" id="sen4" name="sen4">
        <option value="" selected disabled hidden>Select Option</option>
        <option value="0">No</option>
      </select>
    </div>
    <div class="custom">
      <label class="label-input">CGPA</label>
      <select class="user-box-op" id="sen5" name="sen5">
        <option value="" selected disabled hidden>Select CGPA</option>
        <option value="5">5</option>
        <option value="6">6</option>
        <option value="7">7</option>
        <option value="8">8</option>
        <option value="9">9</option>
      </select>
    </div>
    <div class="custom">
      <label class="label-input">You have any arrears in Exams?</label>
      <select class="user-box-op" id="sen6" name="sen6">
        <option value="" selected disabled hidden>Select Option</option>
        <option value="1">Yes</option>
        <option value="0">No</option>
      </select>
    </div>
    <input type="Submit" value="Predict" class="cta">
  </form>
</div>
</div>
```

Submit.html

```
submit.html - Notepad
File Edit View

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<title>prediction</title>
<meta name="description" content="">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="static/styles.css">
<link href="https://fonts.googleapis.com/css?family=Montserrat:500&display=swap" rel="stylesheet">
</head>
<body>
<header>
<nav>
<ul class="nav_links">
<li><a href="/">Home</a></li>
<li><a href="/about">About</a></li>
</ul>
</nav>
<a class="cta" href="/predict">Predict</a>
</header>
<div class="idp-text">
<h1>The Prediction is : {{Y}} </h1>
<h3><span>0</span> represents <span>Not-placed</span></h3>
<h3>1 represents placed</h3>
</div>
</body>
</html>
```

App.py

```
app.py - Visual Studio Code
File Edit Selection View Go Run Terminal Help
C:\Users\HP\Desktop> All Projects > COMPLETED > Project 4 > Flask > app.py > pred
1 from flask import Flask, request, render_template
2 import pickle
3 import sklearn
4 app = Flask(__name__)
5 model = pickle.load(open('placement.pkl', 'rb'))
6 @app.route('/')
7 def home():
8     return render_template('home.html')
9 @app.route('/about')
10 def about():
11     return render_template('about.html')
12 @app.route('/predict')
13 def predict():
14     return render_template('predict.html')
15 @app.route('/pred', methods=['post'])
16 def pred():
17     sen1 = request.form['sen1']
18     sen2 = request.form['sen2']
19     sen3 = request.form['sen3']
20     sen4 = request.form['sen4']
21     sen5 = request.form['sen5']
22     sen6 = request.form['sen6']
23     variables = [[int(sen1), int(sen2), int(sen3), int(sen4), int(sen5), int(sen6)]]
24
25     model.predict(variables)
26     output = model.predict(variables)
27     return render_template('submit.html', Y=output[0])
28 if __name__ == "__main__":
29     app.run(debug=True)
```