MACHINE LEARNING PROJECT

Satisfaction of Airline Passengers

Institute Name: ITVedant Education Pvt Ltd
Name: Sanjana Joshi
Email Address: sanjanaj1097@gmail.com

1. Aim:

The aim of this project is to develop a classification based machine learning model to predict the satisfaction of airline passengers and to find the supervised machine learning algorithm that predicts the satisfaction most accurately. Also, the user can enter data of the model using Web App UI on portal.

2. Dataset Description:

This dataset includes attributes which caters to the satisfaction of airline passengers. It is a large scale dataset consisting of 129880 rows and 24 columns.

Column Summary

Gender: Gender of the passengers (Female, Male)

Customer Type: The customer type (Loyal customer, disloyal customer)

Age: The actual age of the passengers

Type of Travel: Purpose of the flight of the passengers (Personal Travel, Business Travel)

Class: Travel class in the plane of the passengers (Business, Eco, Eco Plus)

Flight distance: The flight distance of this journey

Inflight wifi service: Satisfaction level of the inflight wifi service (0:Not Applicable;1-5)

Departure/Arrival time convenient: Satisfaction level of Departure/Arrival time convenient

Ease of Online booking: Satisfaction level of online booking

Gate location: Satisfaction level of Gate location

Food and drink: Satisfaction level of Food and drink

Online boarding: Satisfaction level of online boarding

Seat comfort: Satisfaction level of Seat comfort

Inflight entertainment: Satisfaction level of inflight entertainment

On-board service: Satisfaction level of On-board service

Leg room service: Satisfaction level of Leg room service

Baggage handling: Satisfaction level of baggage handling

Check-in service: Satisfaction level of Check-in service

Inflight service: Satisfaction level of inflight service

Cleanliness: Satisfaction level of Cleanliness

Departure Delay in Minutes: Minutes delayed when departure

Arrival Delay in Minutes: Minutes delayed when Arrival

Satisfaction: Neutral/Dissatisfied or Satisfied

3. Objectives:

- **Collection of Data:** Gather detailed information on the dataset, which will help to train model more accurately.
- **Data pre-processing:** Data pre-processing is the process of cleaning our data set. There might be unnecessary information or missing values or outliers in the dataset. These can be handled by data cleaning. If there are many missing values in a variable we will drop those values or substitute it with the average/mean value. So data cleaning is crucial part of ML which will help to get accurate prediction.
- **Visualization:** Create visualizations to illustrate relationship between various features and target value(i.e satisfaction). This could include pie chart, bar chart, heatmap, etc.
- **Label encoding:** ML algorithm works only on numerical columns. Hence, in order to pass categorical columns in machine learning algorithm, it needs to be encoded into numerical values.
- **Scaling:** Scaling is done on numerical columns so that it can be brought under a small range.
- **Train and Test Model:** Split data set into training and testing sets.
- **ML Evaluation:** Train the models using supervised machine learning models to predict output.
- **ML Comparison:** Compare all the supervised machine learning models to find out which is the best machine learning model.
- **Web Deployment :** Deploy the machine learning model using Web App UI where user can enter the data and get the output.

4. Steps with Coding Outputs:

1. Load Libraries and dataset:

```
In [25]: import pandas as pd
         import seaborn as sns
         import matplotlib.pyplot as plt
```

```
In [26]: # Load the dataset

         df = pd.read_csv("C:\\Users\\DELL\\Desktop\\Sanjana\\itvedant\\airline_passenger_satisfaction.csv")
         df
```

Out[26]:

	Unnamed: 0	Gender	customer_type	age	type_of_travel	customer_class	flight_distance	inflight_wifi_service	departure_arrival_time_convenient	ease_of
0	0	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3	4	
1	1	Male	disloyal Customer	25	Business travel	Business	235	3	2	
2	2	Female	Loyal Customer	26	Business travel	Business	1142	2	2	
3	3	Female	Loyal Customer	25	Business travel	Business	562	2	5	
4	4	Male	Loyal Customer	61	Business travel	Business	214	3	3	
...	...	...	...	...	...	...	...	...	...	
129875	129875	Male	disloyal Customer	34	Business travel	Business	526	3	3	
129876	129876	Male	Loyal Customer	23	Business travel	Business	646	4	4	
129877	129877	Female	Loyal Customer	17	Personal Travel	Eco	828	2	5	
129878	129878	Male	Loyal Customer	14	Business travel	Business	1127	3	3	
129879	129879	Female	Loyal Customer	42	Personal Travel	Eco	264	2	5	

129880 rows × 24 columns

2. EDA:

Drop unncessary Columns

```
In [27]: #Column 'Unnamed: 0' is irrelevant. Therefore, dropping the column from the dataset.

         df = df.drop(['Unnamed: 0'],axis=1)
         df
```

Out[27]:

	Gender	customer_type	age	type_of_travel	customer_class	flight_distance	inflight_wifi_service	departure_arrival_time_convenient	ease_of_online_boo
0	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3	4	
1	Male	disloyal Customer	25	Business travel	Business	235	3	2	
2	Female	Loyal Customer	26	Business travel	Business	1142	2	2	
3	Female	Loyal Customer	25	Business travel	Business	562	2	5	
4	Male	Loyal Customer	61	Business travel	Business	214	3	3	
...	...	...	...	...	...	...	...	...	
129875	Male	disloyal Customer	34	Business travel	Business	526	3	3	
129876	Male	Loyal Customer	23	Business travel	Business	646	4	4	
129877	Female	Loyal Customer	17	Personal Travel	Eco	828	2	5	
129878	Male	Loyal Customer	14	Business travel	Business	1127	3	3	
129879	Female	Loyal Customer	42	Personal Travel	Eco	264	2	5	

129880 rows × 23 columns

In [28]:
```
df.head(10)
```

Out[28]:

	Gender	customer_type	age	type_of_travel	customer_class	flight_distance	inflight_wifi_service	departure_arrival_time_convenient	ease_of_online_booking
0	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3	4	3
1	Male	disloyal Customer	25	Business travel	Business	235	3	2	3
2	Female	Loyal Customer	26	Business travel	Business	1142	2	2	2
3	Female	Loyal Customer	25	Business travel	Business	562	2	5	5
4	Male	Loyal Customer	61	Business travel	Business	214	3	3	3
5	Female	Loyal Customer	26	Personal Travel	Eco	1180	3	4	2
6	Male	Loyal Customer	47	Personal Travel	Eco	1276	2	4	2
7	Female	Loyal Customer	52	Business travel	Business	2035	4	3	4
8	Female	Loyal Customer	41	Business travel	Business	853	1	2	2
9	Male	disloyal Customer	20	Business travel	Eco	1061	3	3	3

10 rows × 23 columns

In [29]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 129880 entries, 0 to 129879
Data columns (total 23 columns):
 #   Column                            Non-Null Count   Dtype
---  ------                            --------------   -----
 0   Gender                            129880 non-null  object
 1   customer_type                     129880 non-null  object
 2   age                               129880 non-null  int64
 3   type_of_travel                    129880 non-null  object
 4   customer_class                    129880 non-null  object
 5   flight_distance                   129880 non-null  int64
 6   inflight_wifi_service             129880 non-null  int64
 7   departure_arrival_time_convenient 129880 non-null  int64
 8   ease_of_online_booking            129880 non-null  int64
 9   gate_location                     129880 non-null  int64
 10  food_and_drink                    129880 non-null  int64
 11  online_boarding                   129880 non-null  int64
 12  seat_comfort                      129880 non-null  int64
 13  inflight_entertainment            129880 non-null  int64
 14  onboard_service                   129880 non-null  int64
 15  leg_room_service                  129880 non-null  int64
 16  baggage_handling                  129880 non-null  int64
 17  checkin_service                   129880 non-null  int64
 18  inflight_service                  129880 non-null  int64
 19  cleanliness                       129880 non-null  int64
 20  departure_delay_in_minutes        129880 non-null  int64
 21  arrival_delay_in_minutes          129487 non-null  float64
 22  satisfaction                      129880 non-null  object
dtypes: float64(1), int64(17), object(5)
memory usage: 22.8+ MB
```

In [30]:
```
df.describe()
```

Out[30]:

	age	flight_distance	inflight_wifi_service	departure_arrival_time_convenient	ease_of_online_booking	gate_location	food_and_drink	online_boa
count	129880.000000	129880.000000	129880.000000	129880.000000	129880.000000	129880.000000	129880.000000	129880.0
mean	39.427957	1190.316392	2.728696	3.057599	2.756876	2.976925	3.204774	3.2
std	15.119360	997.452477	1.329340	1.526741	1.401740	1.278520	1.329933	1.3
min	7.000000	31.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
25%	27.000000	414.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.0
50%	40.000000	844.000000	3.000000	3.000000	3.000000	3.000000	3.000000	3.0
75%	51.000000	1744.000000	4.000000	4.000000	4.000000	4.000000	4.000000	4.0
max	85.000000	4983.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.0

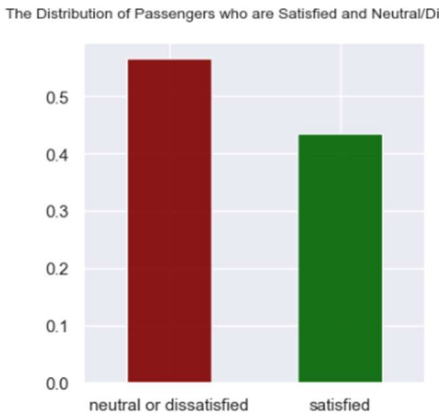
```
In [31]: df.isnull().sum()

Out[31]: Gender                              0
         customer_type                       0
         age                                 0
         type_of_travel                      0
         customer_class                      0
         flight_distance                     0
         inflight_wifi_service               0
         departure_arrival_time_convenient   0
         ease_of_online_booking              0
         gate_location                       0
         food_and_drink                      0
         online_boarding                     0
         seat_comfort                        0
         inflight_entertainment              0
         onboard_service                     0
         leg_room_service                    0
         baggage_handling                    0
         checkin_service                     0
         inflight_service                    0
         cleanliness                         0
         departure_delay_in_minutes          0
         arrival_delay_in_minutes          393
         satisfaction                        0
         dtype: int64
```

```
In [32]: #As we can see null values in "Arrival_delay_in_minutes" column, we need to fill na with mean

         arrival_delay_in_minutes_mean=df["arrival_delay_in_minutes"].mean()
         df["arrival_delay_in_minutes"]=df["arrival_delay_in_minutes"].fillna(arrival_delay_in_minutes_mean)

In [33]: #Now check again if null values are there after filling na with mean

         df.isnull().sum()

Out[33]: Gender                              0
         customer_type                       0
         age                                 0
         type_of_travel                      0
         customer_class                      0
         flight_distance                     0
         inflight_wifi_service               0
         departure_arrival_time_convenient   0
         ease_of_online_booking              0
         gate_location                       0
         food_and_drink                      0
         online_boarding                     0
         seat_comfort                        0
         inflight_entertainment              0
         onboard_service                     0
         leg_room_service                    0
         baggage_handling                    0
         checkin_service                     0
         inflight_service                    0
         cleanliness                         0
         departure_delay_in_minutes          0
         arrival_delay_in_minutes            0
         satisfaction                        0
         dtype: int64

In [34]: #Checking for duplicates

         df.duplicated().sum()

Out[34]: 0
```

3. Visualization:

The Distribution of Passengers who are Satisfied and Neutral/Dissatisfied

In [35]:
```
fig = plt.figure(figsize = (4,4))
df.satisfaction.value_counts(normalize=True).plot(kind='bar', color= ['maroon','darkgreen'], alpha = 0.9, rot=0)
plt.title('The Distribution of Passengers who are Satisfied and Neutral/Dissatisfied\n')
plt.show()
```



In [36]:
```
df.satisfaction.value_counts()
```

Out[36]:
```
neutral or dissatisfied    73452
satisfied                  56428
Name: satisfaction, dtype: int64
```

As we can see above, in general, based on all the parameters, more number of passengers are neutral/dissatisfied than satisfied

Now using bar plot of numerical features

In [*]:
```
numerical = [df['age'], df['flight_distance'], df['departure_delay_in_minutes'], df['arrival_delay_in_minutes'], df['departure_a
             df['ease_of_online_booking'], df['checkin_service'], df['online_boarding'], df['gate_location'], df['onboard_servic
             df['seat_comfort'], df['leg_room_service'], df['cleanliness'], df['food_and_drink'], df['inflight_service'],
              df['inflight_wifi_service'],df['inflight_entertainment'], df['baggage_handling']]

sns.set(rc={'figure.figsize':(4, 4),
            "font.size":8,
            "axes.titlesize":8,
            "axes.labelsize":8},
             style="darkgrid")


for i in numerical:
    sns.barplot(data=df, x=df['satisfaction'], y=i, palette=['#FFA200','#62D82B'])
    plt.show()
```

From the bar plot for numerical features

1. Passengers were more satisfied under features such as age, flight distance, ease of online booking, checkin service, online boarding, onboard service, seat comfort, leg room service, cleanliness, food and drink, inflight services, inflight wifi service, inflight entertainment and baggage handling
2. Passengers were more dissatisfied under features such as departure delay, arrival delay, departure and arrival time convinient

Also, from the above bar plot, it is known that the location of the Gate does not play a major role in passenger satisfaction. Therefore we can ignore Gate location column.

In [38]:
```
df = df.drop('gate_location',axis=1)
```

In [39]:
```
df
```

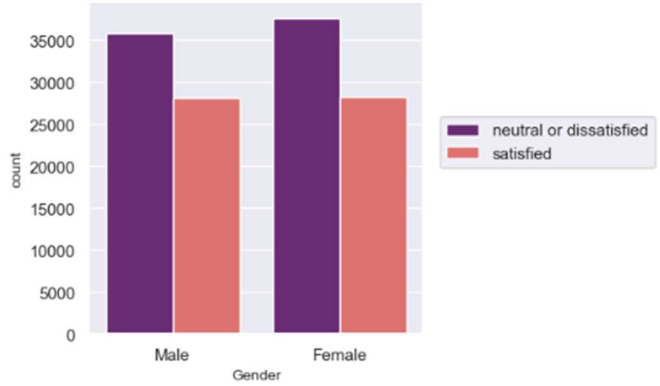
Out[39]:

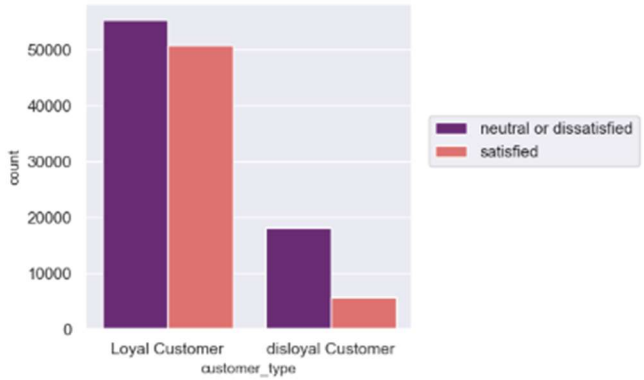
	Gender	customer_type	age	type_of_travel	customer_class	flight_distance	inflight_wifi_service	departure_arrival_time_convenient	ease_of_online_bool
0	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3	4	
1	Male	disloyal Customer	25	Business travel	Business	235	3	2	
2	Female	Loyal Customer	26	Business travel	Business	1142	2	2	
3	Female	Loyal Customer	25	Business travel	Business	562	2	5	
4	Male	Loyal Customer	61	Business travel	Business	214	3	3	
...	...	...	...	...	...	...	...	...	
129875	Male	disloyal Customer	34	Business travel	Business	526	3	3	
129876	Male	Loyal Customer	23	Business travel	Business	646	4	4	
129877	Female	Loyal Customer	17	Personal Travel	Eco	828	2	5	
129878	Male	Loyal Customer	14	Business travel	Business	1127	3	3	
129879	Female	Loyal Customer	42	Personal Travel	Eco	264	2	5	

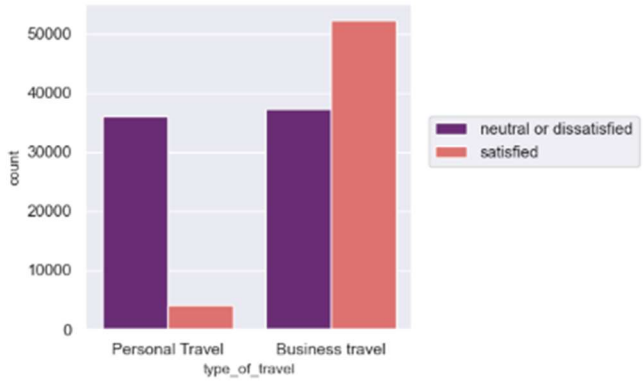
129880 rows × 22 columns

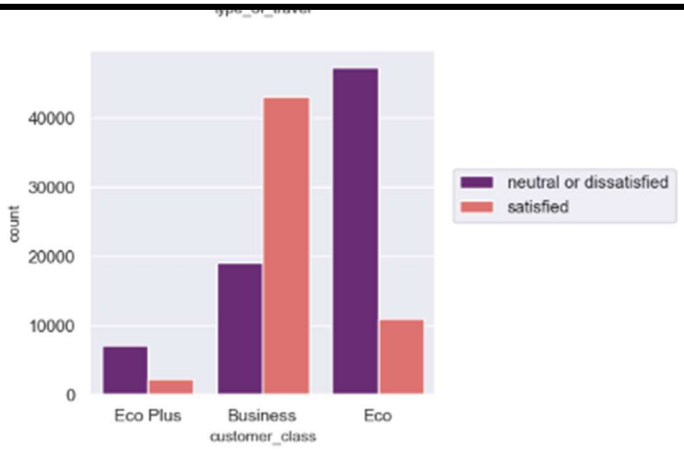
Now using count plot of Categorical features

In [40]:
```
categorical = [df['Gender'], df['customer_type'], df['type_of_travel'], df['customer_class'], df['satisfaction']]

sns.set(rc={'figure.figsize':(4, 4),
            "font.size":10,
            "axes.titlesize":10,
            "axes.labelsize":15},
            style="darkgrid",
           )

for col in categorical[:-1]:
    plt.figure(figsize=(4, 4))
    sns.countplot(data=df, x=col, hue ='satisfaction', palette='magma')
    plt.legend(loc=(1.05, 0.5))
```









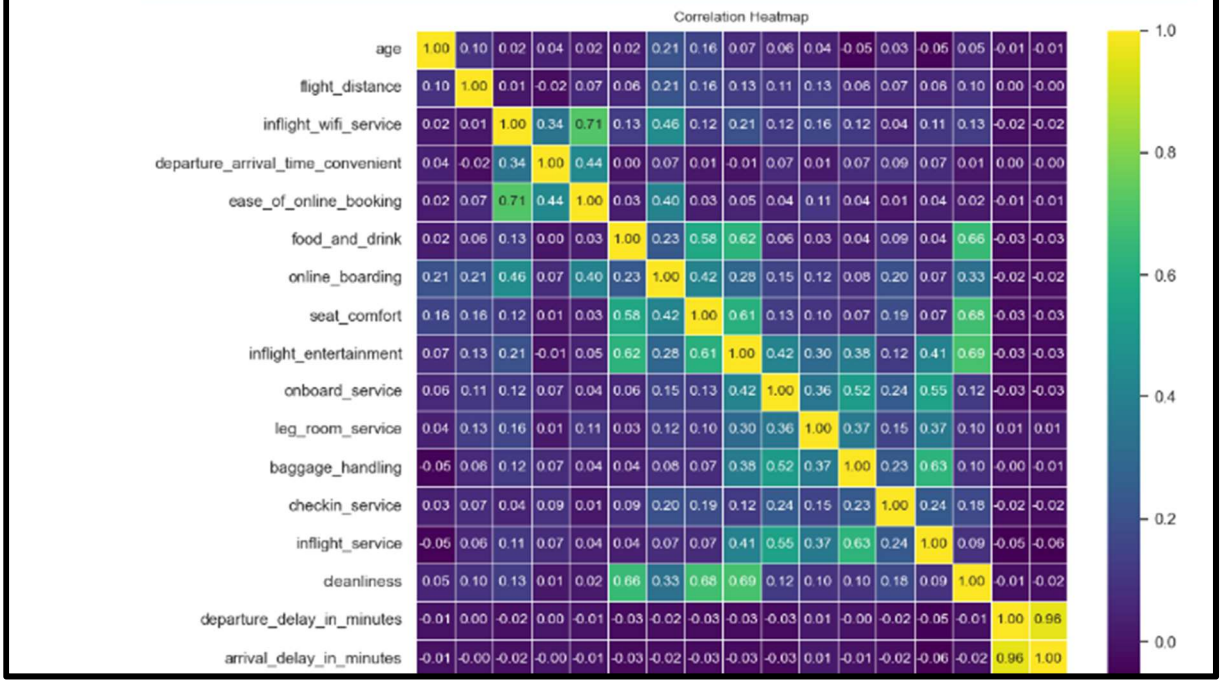
From the above count plot for categorical values:

1. Gender does not play a significant role in satisfaction, as men and women seem to be concerned about the same factors.
2. The number of loyal customers for the airline is high, but the dissatisfaction rate is high regardless of loyalty. Airlines must pay attention to retaining loyal customers.
3. Travelers with business travel intentions appear to be more satisfied with flights, than travelers with personal travel intentions.
4. Passengers in business class seem to be the most satisfied, while passengers in economy class are the most dissatisfied.

Using heatmap visualization

In [41]:
```
plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(), annot=True, cmap='viridis', fmt='.2f', linewidths=0.5)
plt.title('Correlation Heatmap')
plt.show()
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_21804\2618243022.py:2: FutureWarning: The default value of numeric_only in DataFram
e.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_
only to silence this warning.
  sns.heatmap(df.corr(), annot=True, cmap='viridis', fmt='.2f', linewidths=0.5)
```



4. Check if Dependent variable is balanced or not

Check if Dependent variable is balanced or not

In [42]:
```
target = df['satisfaction'].value_counts().reset_index()
target.rename(columns={'index':'satisfaction','satisfaction':'count'}, inplace=True)
target['percentage'] = round((target['count']/target['count'].sum())*100,2)
target
```

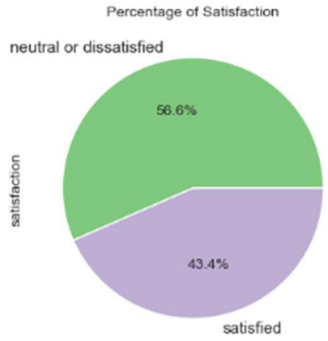
Out[42]:

	satisfaction	count	percentage
0	neutral or dissatisfied	73452	56.55
1	satisfied	56428	43.45

As we can see above, percentage of dissatisfied to satisfied 56.55: 43.45. Even though ratio is out of balance, but the numbers are still tolerable. Therefore it does not require Imbalanced Data handling to overcome this.

In [43]:
```
fig = plt.figure(figsize=(4,4))
ax1 = plt.subplot(111)
df['satisfaction'].value_counts().plot.pie(colors = sns.color_palette("Accent"),autopct='%1.1f%%',ax=ax1)
ax1.set_title('Percentage of Satisfaction')
```

Out[43]: Text(0.5, 1.0, 'Percentage of Satisfaction')



5. Label Encoding:

Using Label Encoder for converting categorical features into numerical features

In [44]:
```
df_categorical = df.select_dtypes(['object'])
df_categorical
```

Out[44]:

	Gender	customer_type	type_of_travel	customer_class	satisfaction
0	Male	Loyal Customer	Personal Travel	Eco Plus	neutral or dissatisfied
1	Male	disloyal Customer	Business travel	Business	neutral or dissatisfied
2	Female	Loyal Customer	Business travel	Business	satisfied
3	Female	Loyal Customer	Business travel	Business	neutral or dissatisfied
4	Male	Loyal Customer	Business travel	Business	satisfied
...	...	...	...	...	...
129875	Male	disloyal Customer	Business travel	Business	neutral or dissatisfied
129876	Male	Loyal Customer	Business travel	Business	satisfied
129877	Female	Loyal Customer	Personal Travel	Eco	neutral or dissatisfied
129878	Male	Loyal Customer	Business travel	Business	satisfied
129879	Female	Loyal Customer	Personal Travel	Eco	neutral or dissatisfied

129880 rows × 5 columns

In [45]:
```
import sklearn
from sklearn.preprocessing import LabelEncoder

le = sklearn.preprocessing.LabelEncoder()

for i in df_categorical:
    df[i] = le.fit_transform(df[i])
df
```

Out[45]:

	Gender	customer_type	age	type_of_travel	customer_class	flight_distance	inflight_wifi_service	departure_arrival_time_convenient	ease_of_online_boo
0	1	0	13	1	2	460	3	4	
1	1	1	25	0	0	235	3	2	
2	0	0	26	0	0	1142	2	2	
3	0	0	25	0	0	562	2	5	
4	1	0	61	0	0	214	3	3	
...	...	...	...	...	...	...	...	...	...
129875	1	1	34	0	0	526	3	3	
129876	1	0	23	0	0	646	4	4	
129877	0	0	17	1	1	828	2	5	
129878	1	0	14	0	0	1127	3	3	
129879	0	0	42	1	1	264	2	5	

129880 rows × 22 columns

In [46]:
```
df.dtypes
```

Out[46]:
```
Gender                                 int32
customer_type                          int32
age                                    int64
type_of_travel                         int32
customer_class                         int32
flight_distance                        int64
inflight_wifi_service                  int64
departure_arrival_time_convenient      int64
ease_of_online_booking                 int64
food_and_drink                         int64
online_boarding                        int64
seat_comfort                           int64
inflight_entertainment                 int64
onboard_service                        int64
leg_room_service                       int64
baggage_handling                       int64
checkin_service                        int64
inflight_service                       int64
cleanliness                            int64
departure_delay_in_minutes             int64
arrival_delay_in_minutes             float64
satisfaction                           int32
dtype: object
```

In [47]:
```
df.isnull().sum()
```

Out[47]:
```
Gender                                 0
customer_type                          0
age                                    0
type_of_travel                         0
customer_class                         0
flight_distance                        0
inflight_wifi_service                  0
departure_arrival_time_convenient      0
ease_of_online_booking                 0
food_and_drink                         0
online_boarding                        0
seat_comfort                           0
inflight_entertainment                 0
onboard_service                        0
leg_room_service                       0
baggage_handling                       0
checkin_service                        0
inflight_service                       0
cleanliness                            0
departure_delay_in_minutes             0
arrival_delay_in_minutes               0
satisfaction                           0
dtype: int64
```

Now, we can see all categorical features have been converted to numerical features

6. Split the datasets into training set and testing set:

Split the dataset into X and Y

```
In [48]: x=df.drop('satisfaction',axis=1)
         x
```

Out[48]:

	Gender	customer_type	age	type_of_travel	customer_class	flight_distance	inflight_wifi_service	departure_arrival_time_convenient	ease_of_online_boo
0	1	0	13	1	2	460	3	4	
1	1	1	25	0	0	235	3	2	
2	0	0	26	0	0	1142	2	2	
3	0	0	25	0	0	562	2	5	
4	1	0	61	0	0	214	3	3	
...	...	...	...	...	...	...	...	...	
129875	1	1	34	0	0	526	3	3	
129876	1	0	23	0	0	646	4	4	
129877	0	0	17	1	1	828	2	5	
129878	1	0	14	0	0	1127	3	3	
129879	0	0	42	1	1	264	2	5	

129880 rows × 21 columns

```
In [49]: y = df['satisfaction']
         y

Out[49]: 0         0
         1         0
         2         1
         3         0
         4         1
                  ..
         129875    0
         129876    1
         129877    0
         129878    1
         129879    0
         Name: satisfaction, Length: 129880, dtype: int32
```

```
In [51]: from sklearn.model_selection import train_test_split

         x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2,random_state=42)
```

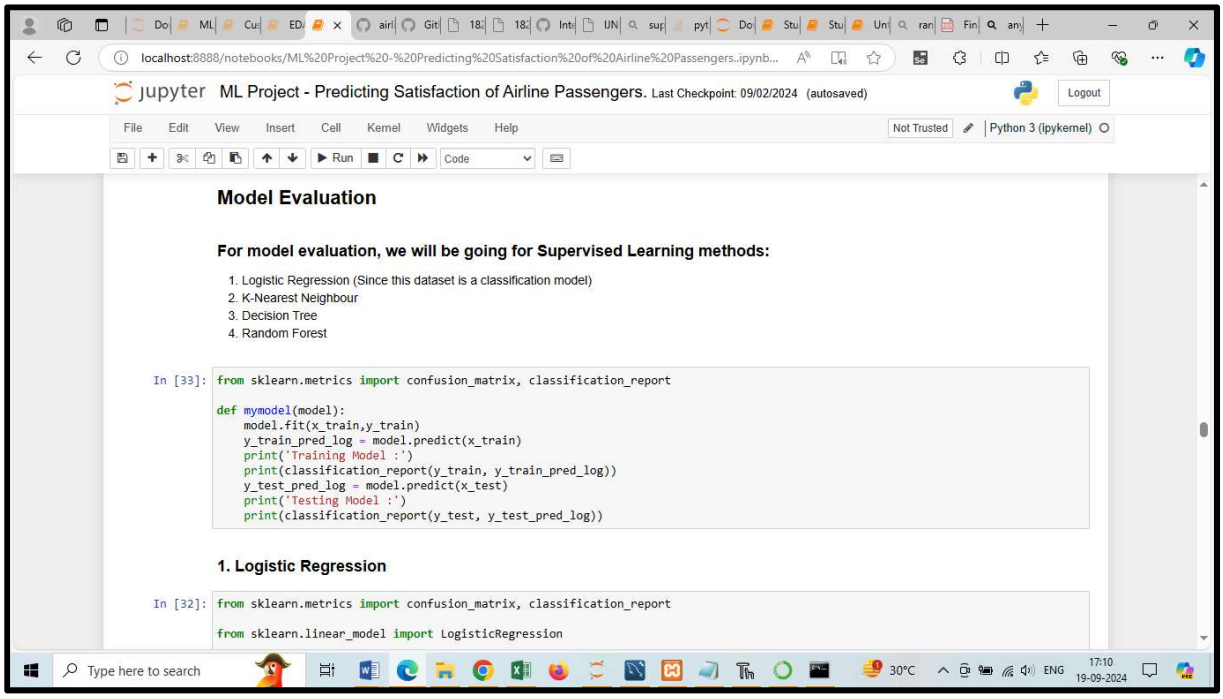
7. Feature Scaling:

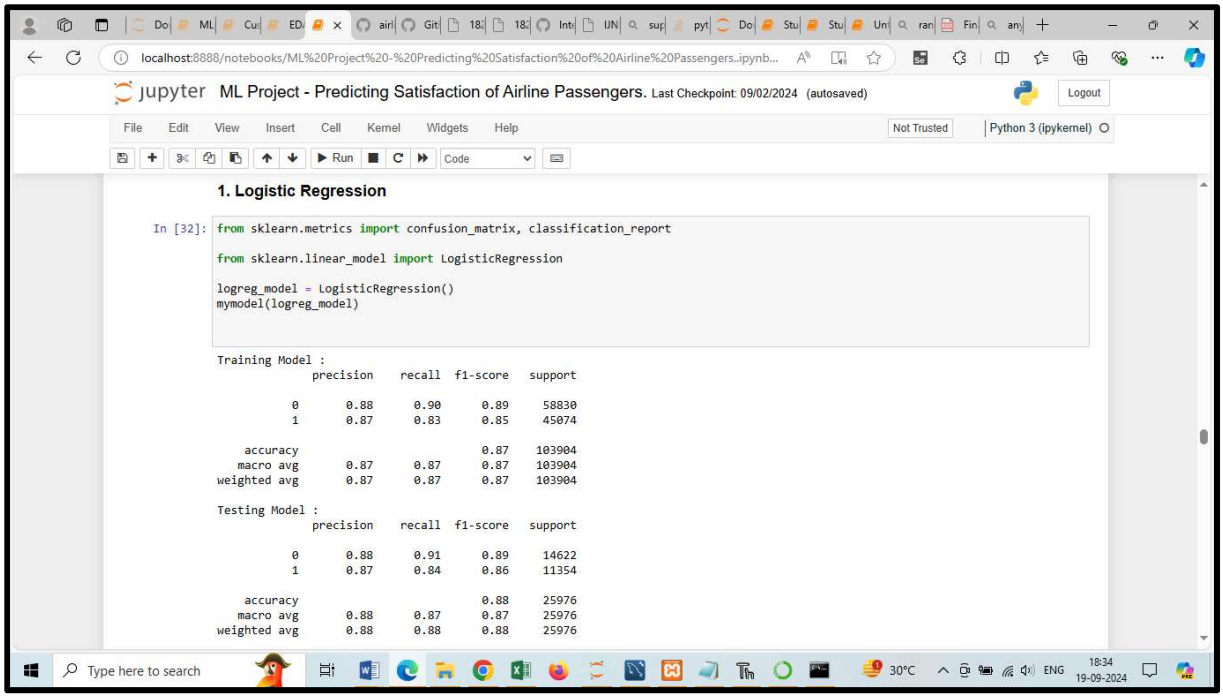
Feature Scaling

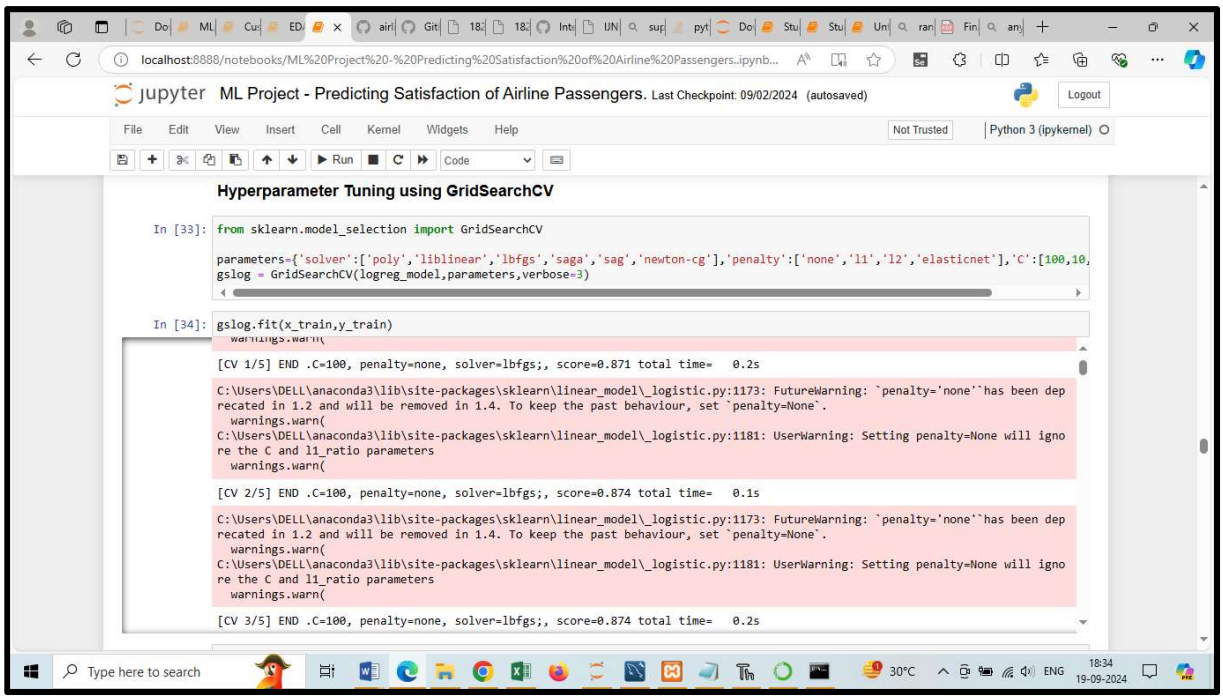
```
In [52]: from sklearn.preprocessing import StandardScaler

         scaler = StandardScaler()

         x_train = scaler.fit_transform(x_train)
         x_test = scaler.fit_transform(x_test)

         print('X_train: ',x_train)
         print('X_test: ',x_test)
```

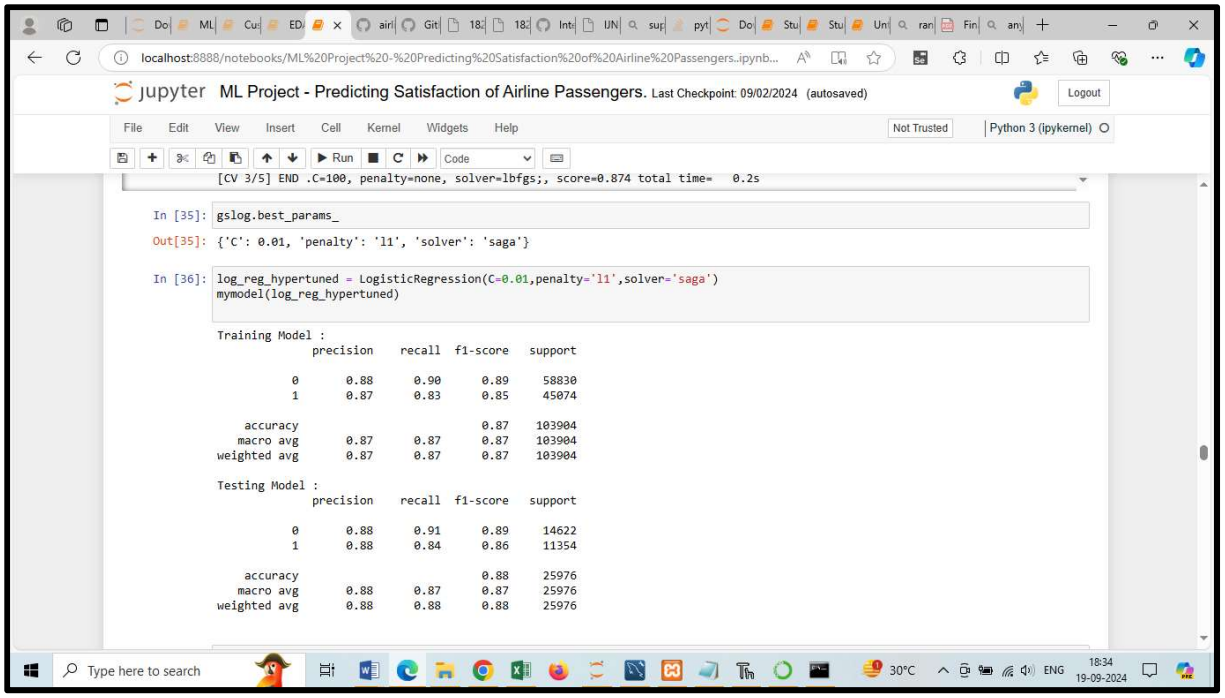
```
X_train:  [[-0.98623577 -0.47483775  1.62649954 ...  1.3056862  -0.09824202
  -0.28874091]
 [ 1.01395633  2.10598253 -0.88620297 ... -0.21881382  2.55208965
   2.8581419 ]
 [-0.98623577 -0.47483775 -2.07643047 ...  0.54343619 -0.38689201
  -0.39277009]
 ...
 [-0.98623577 -0.47483775 -0.15884172 ...  1.3056862  -0.38689201
  -0.39277009]
 [-0.98623577 -0.47483775 -1.67968797 ...  0.54343619 -0.38689201
  -0.39277009]
 [ 1.01395633 -0.47483775 -0.35721297 ... -0.98106383 -0.38689201
  -0.39277009]]
X_test:  [[ 1.01857407 -0.46774636  1.75251478 ...  0.54259374 -0.38483534
  -0.39354777]
 [-0.98176463 -0.46774636 -0.10136965 ...  0.54259374 -0.30572254
  -0.39354777]
 [-0.98176463 -0.46774636 -0.10136965 ... -0.97171354 -0.38483534
  -0.39354777]
 ...
 [ 1.01857407 -0.46774636 -0.76347123 ...  0.54259374 -0.33209347
  -0.39354777]
 [ 1.01857407 -0.46774636 -0.63105092 ...  1.29974739  1.98854875
   2.14354166]
 [-0.98176463 -0.46774636  1.28904368 ... -0.2145599  -0.14749693
  -0.39354777]]
```

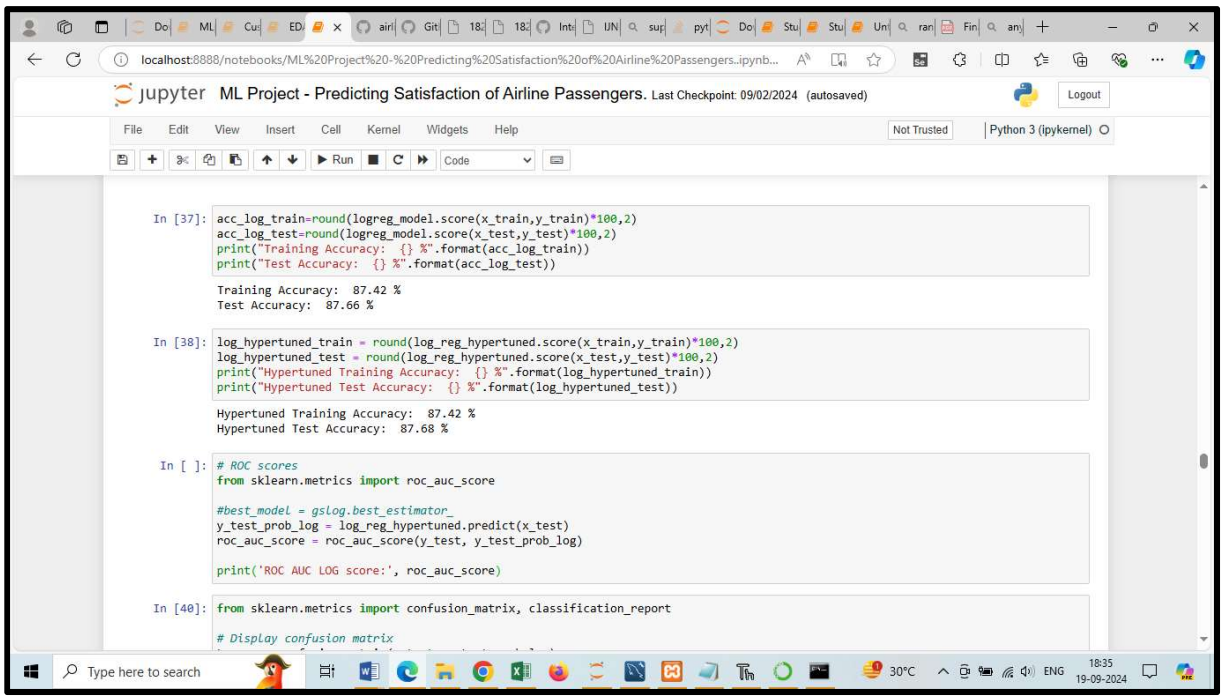
8. Model Evaluation:

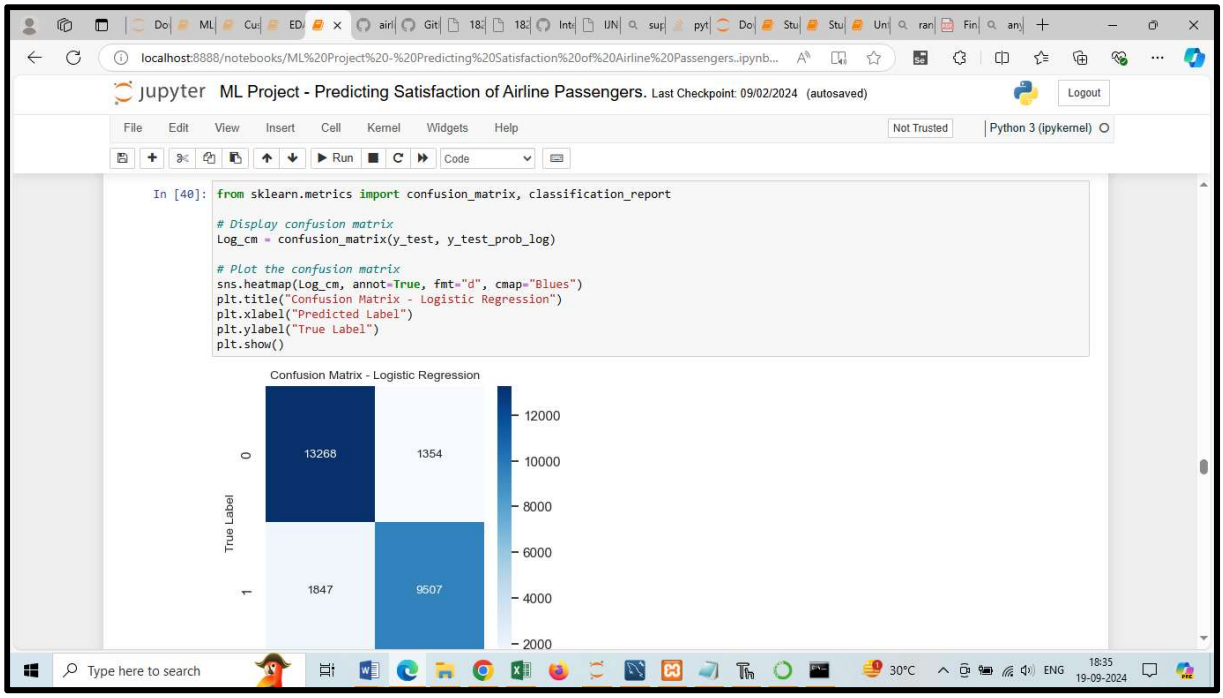


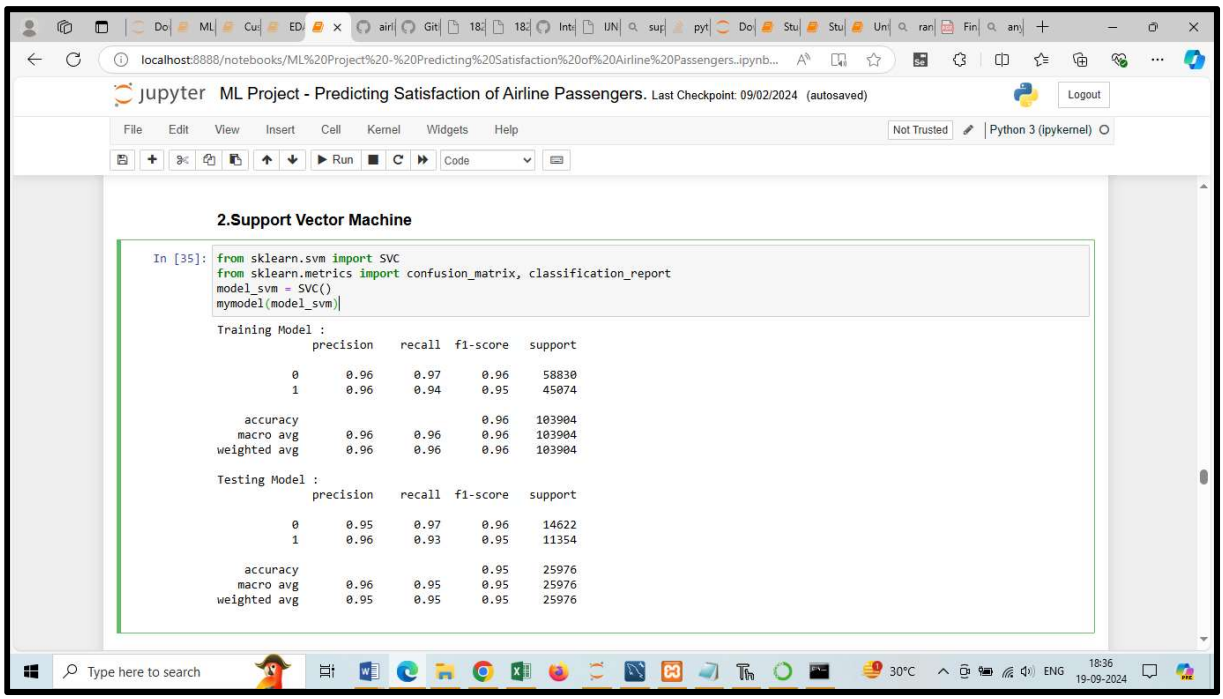

localhost:8888/notebooks/ML%20Project%20-%20Predicting%20Satisfaction%20of%20Airline%20Passengers..ipynb...
jupyter ML Project - Predicting Satisfaction of Airline Passengers. Last Checkpoint: 09/02/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
1. Logistic Regression
In [32]:
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.linear_model import LogisticRegression
logreg_model = LogisticRegression()
mymodel(logreg_model)
Training Model :
precision recall f1-score support
0 0.88 0.90 0.89 58830
1 0.87 0.83 0.85 45074
accuracy 0.87 103904
macro avg 0.87 0.87 0.87 103904
weighted avg 0.87 0.87 0.87 103904
Testing Model :
precision recall f1-score support
0 0.88 0.91 0.89 14622
1 0.87 0.84 0.86 11354
accuracy 0.88 25976
macro avg 0.88 0.87 0.87 25976
weighted avg 0.88 0.88 0.88 25976
Type here to search
30°C
ENG
18:34
19-09-2024

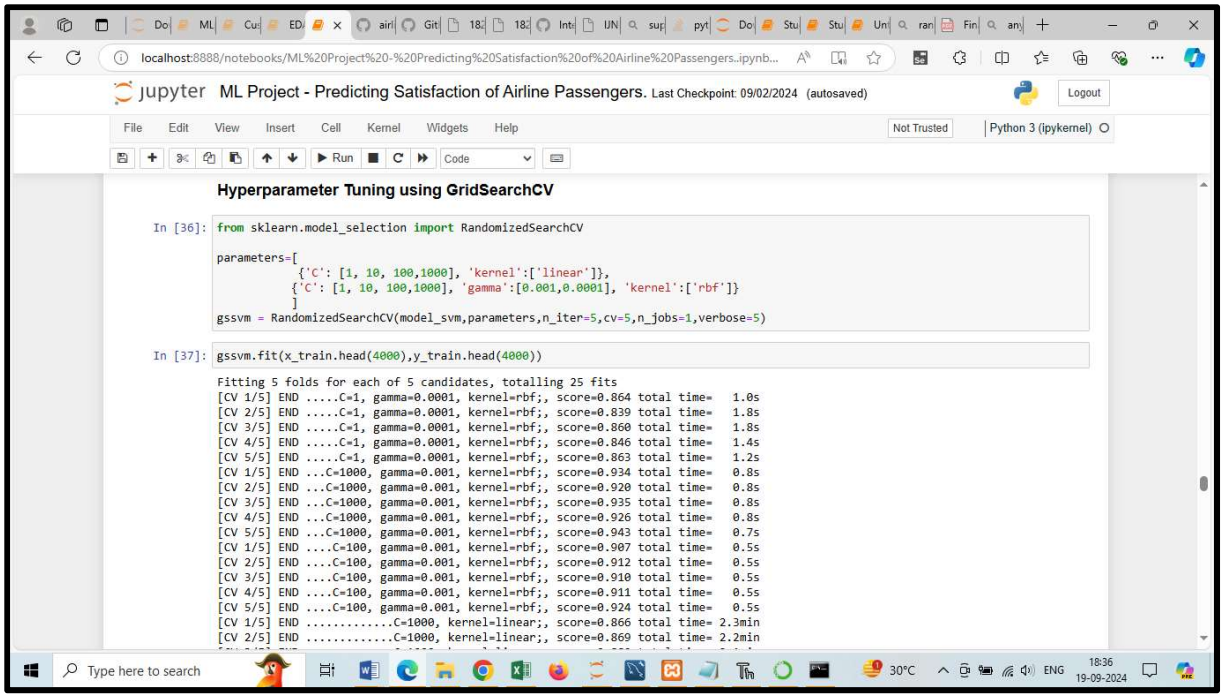

localhost:8888/notebooks/ML%20Project%20-%20Predicting%20Satisfaction%20of%20Airline%20Passengers..ipynb...
jupyter ML Project - Predicting Satisfaction of Airline Passengers. Last Checkpoint: 09/02/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Hyperparameter Tuning using GridSearchCV
In [33]:
from sklearn.model_selection import GridSearchCV
parameters={'solver':['poly','liblinear','lbfgs','saga','sag','newton-cg'],'penalty':['none','l1','l2','elasticnet'],'C':[100,10,
gslog = GridSearchCV(logreg_model,parameters,verbose=3)
In [34]:
gslog.fit(x_train,y_train)
[CV 1/5] END .C=100, penalty=none, solver=lbfgs;, score=0.871 total time= 0.2s
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\linear_model_logistic.py:1173: FutureWarning: `penalty='none'`has been dep
recated in 1.2 and will be removed in 1.4. To keep the past behaviour, set `penalty=None`.
warnings.warn(
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\linear_model_logistic.py:1181: UserWarning: Setting penalty=None will igno
re the C and l1_ratio parameters
warnings.warn(
[CV 2/5] END .C=100, penalty=none, solver=lbfgs;, score=0.874 total time= 0.1s
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\linear_model_logistic.py:1173: FutureWarning: `penalty='none'`has been dep
recated in 1.2 and will be removed in 1.4. To keep the past behaviour, set `penalty=None`.
warnings.warn(
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\linear_model_logistic.py:1181: UserWarning: Setting penalty=None will igno
re the C and l1_ratio parameters
warnings.warn(
[CV 3/5] END .C=100, penalty=none, solver=lbfgs;, score=0.874 total time= 0.2s
Type here to search
30°C
ENG
18:34
19-09-2024

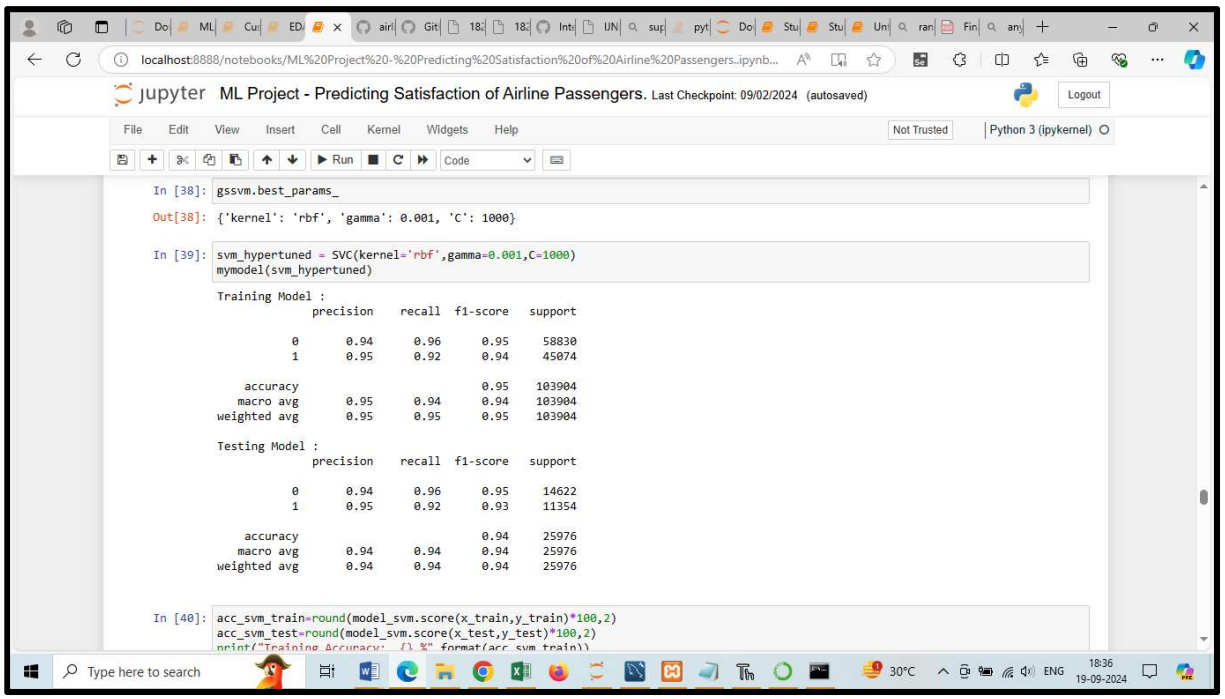

localhost:8888/notebooks/ML%20Project%20-%20Predicting%20Satisfaction%20of%20Airline%20Passengers..ipynb...
jupyter ML Project - Predicting Satisfaction of Airline Passengers. Last Checkpoint: 09/02/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
[CV 3/5] END .C=100, penalty=none, solver=lbfgs;, score=0.874 total time= 0.2s
In [35]:
gslog.best_params_
Out[35]:
{'C': 0.01, 'penalty': 'l1', 'solver': 'saga'}
In [36]:
log_reg_hypertuned = LogisticRegression(C=0.01,penalty='l1',solver='saga')
mymodel(log_reg_hypertuned)
Training Model :
precision recall f1-score support
0 0.88 0.90 0.89 58830
1 0.87 0.83 0.85 45074
accuracy 0.87 103904
macro avg 0.87 0.87 0.87 103904
weighted avg 0.87 0.87 0.87 103904
Testing Model :
precision recall f1-score support
0 0.88 0.91 0.89 14622
1 0.88 0.84 0.86 11354
accuracy 0.88 25976
macro avg 0.88 0.87 0.87 25976
weighted avg 0.88 0.88 0.88 25976
Type here to search
30°C
ENG
18:34
19-09-2024

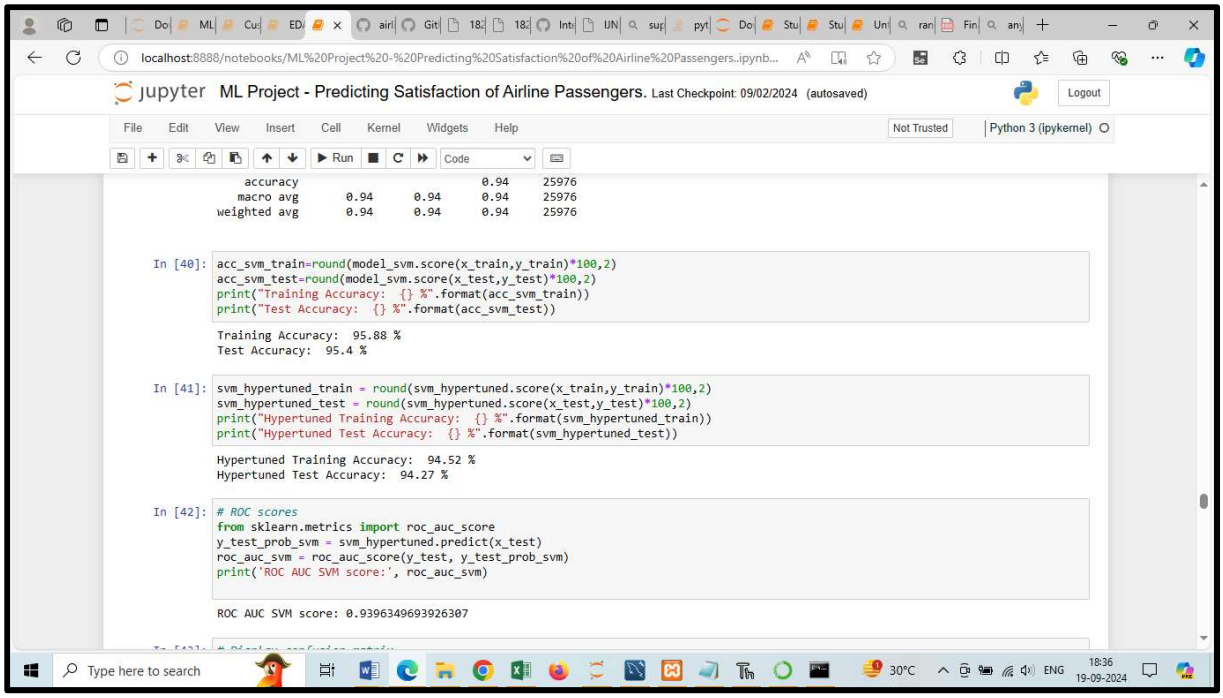

localhost:8888/notebooks/ML%20Project%20-%20Predicting%20Satisfaction%20of%20Airline%20Passengers..ipynb...
jupyter ML Project - Predicting Satisfaction of Airline Passengers. Last Checkpoint: 09/02/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [37]:
acc_log_train=round(logreg_model.score(x_train,y_train)*100,2)
acc_log_test=round(logreg_model.score(x_test,y_test)*100,2)
print("Training Accuracy: {} %".format(acc_log_train))
print("Test Accuracy: {} %".format(acc_log_test))
Training Accuracy: 87.42 %
Test Accuracy: 87.66 %
In [38]:
log_hypertuned_train = round(log_reg_hypertuned.score(x_train,y_train)*100,2)
log_hypertuned_test = round(log_reg_hypertuned.score(x_test,y_test)*100,2)
print("Hypertuned Training Accuracy: {} %".format(log_hypertuned_train))
print("Hypertuned Test Accuracy: {} %".format(log_hypertuned_test))
Hypertuned Training Accuracy: 87.42 %
Hypertuned Test Accuracy: 87.68 %
In [ ]:
# ROC scores
from sklearn.metrics import roc_auc_score
#best_model = gslog.best_estimator_
y_test_prob_log = log_reg_hypertuned.predict(x_test)
roc_auc_score = roc_auc_score(y_test, y_test_prob_log)
print('ROC AUC LOG score:', roc_auc_score)
In [40]:
from sklearn.metrics import confusion_matrix, classification_report
# Display confusion matrix
Type here to search
30°C
ENG
18:35
19-09-2024

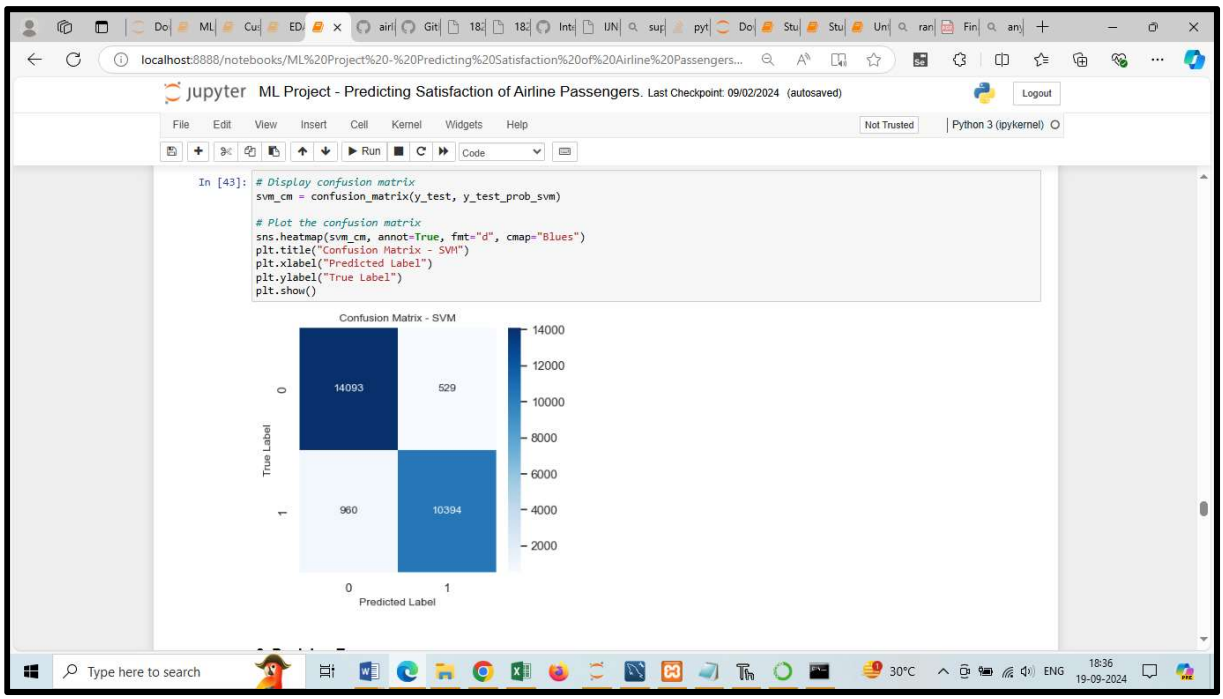

jupyter ML Project - Predicting Satisfaction of Airline Passengers. Last Checkpoint: 09/02/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [40]:
from sklearn.metrics import confusion_matrix, classification_report
# Display confusion matrix
Log_cm = confusion_matrix(y_test, y_test_prob_log)
# Plot the confusion matrix
sns.heatmap(Log_cm, annot=True, fmt="d", cmap="Blues")
plt.title("Confusion Matrix - Logistic Regression")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.show()
Confusion Matrix - Logistic Regression
True Label
0
1
13268
1354
1847
9507
12000
10000
8000
6000
4000
2000
Type here to search
30°C
ENG
18:35
19-09-2024

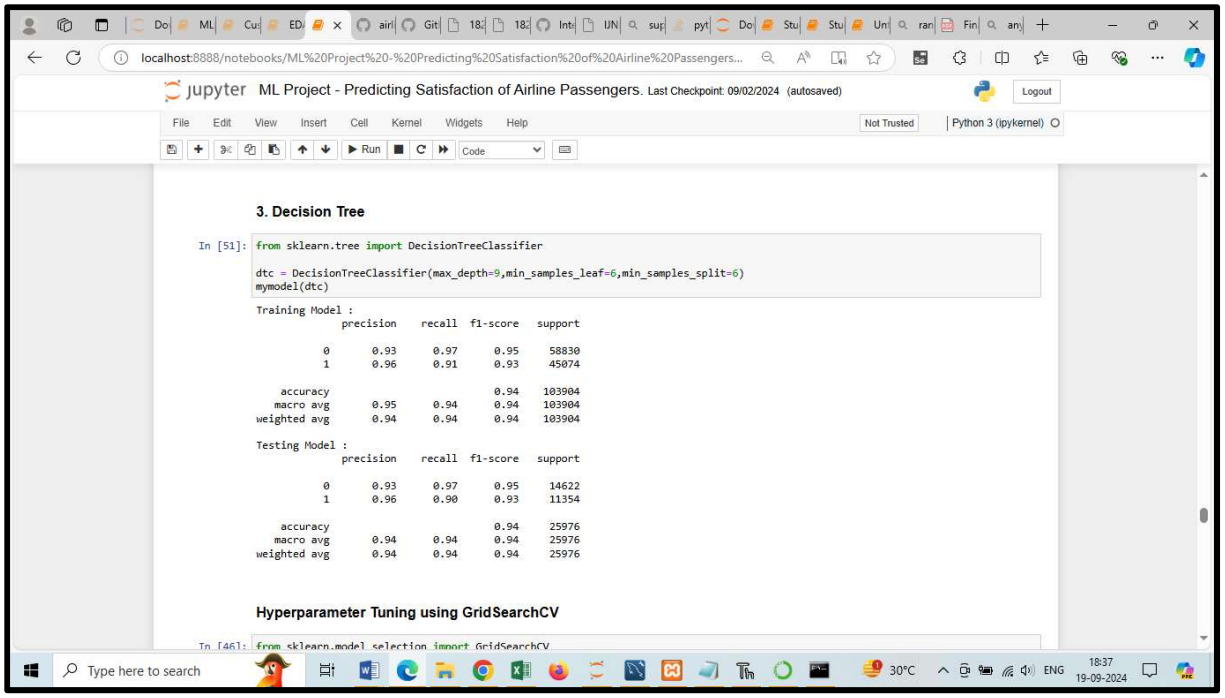

jupyter ML Project - Predicting Satisfaction of Airline Passengers. Last Checkpoint: 09/02/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
2.Support Vector Machine
In [35]:
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix, classification_report
model_svm = SVC()
mymodel(model_svm)
Training Model :
precision recall f1-score support
0 0.96 0.97 0.96 58830
1 0.96 0.94 0.95 45074
accuracy 0.96 103904
macro avg 0.96 0.96 0.96 103904
weighted avg 0.96 0.96 0.96 103904
Testing Model :
precision recall f1-score support
0 0.95 0.97 0.96 14622
1 0.96 0.93 0.95 11354
accuracy 0.95 25976
macro avg 0.96 0.95 0.95 25976
weighted avg 0.95 0.95 0.95 25976
Type here to search
30°C
ENG
18:36
19-09-2024

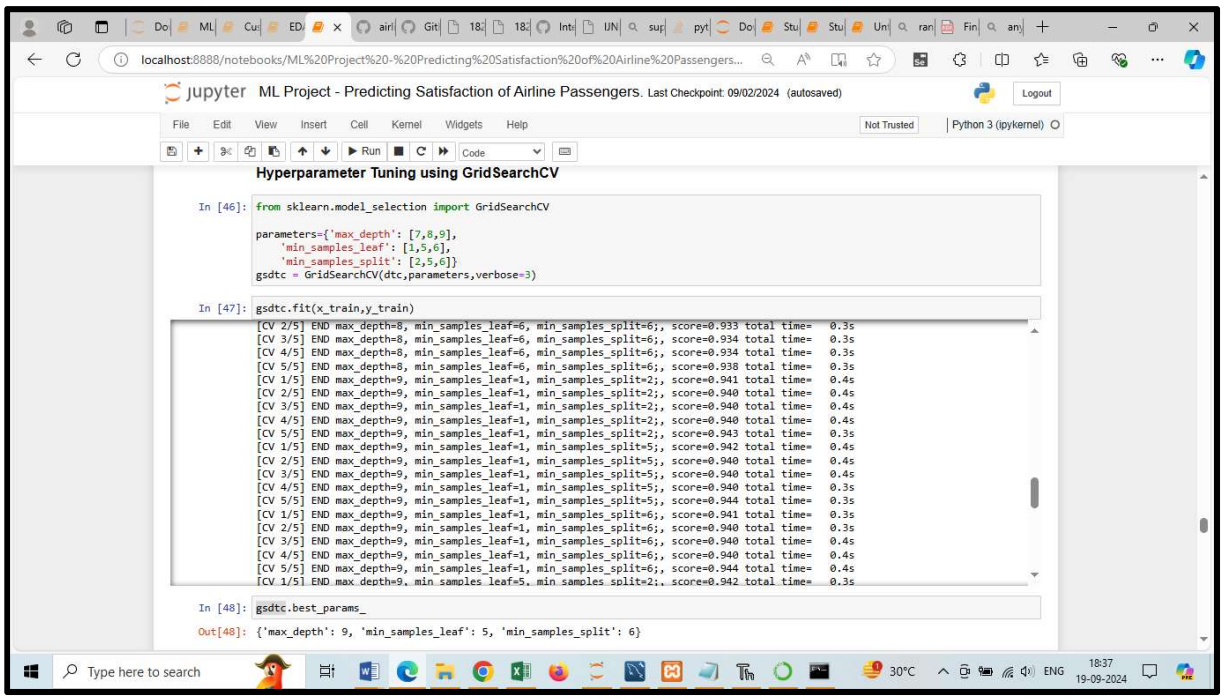

localhost:8888/notebooks/ML%20Project%20-%20Predicting%20Satisfaction%20of%20Airline%20Passengers..ipynb...
jupyter ML Project - Predicting Satisfaction of Airline Passengers. Last Checkpoint: 09/02/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Hyperparameter Tuning using GridSearchCV
In [36]:
from sklearn.model_selection import RandomizedSearchCV
parameters=[
{'C': [1, 10, 100,1000], 'kernel':['linear']},
{'C': [1, 10, 100,1000], 'gamma':[0.001,0.0001], 'kernel':['rbf']}
]
gssvm = RandomizedSearchCV(model_svm,parameters,n_iter=5,cv=5,n_jobs=1,verbose=5)
In [37]:
gssvm.fit(x_train.head(4000),y_train.head(4000))
Fitting 5 folds for each of 5 candidates, totalling 25 fits
[CV 1/5] END .....C=1, gamma=0.0001, kernel=rbf;, score=0.864 total time= 1.0s
[CV 2/5] END .....C=1, gamma=0.0001, kernel=rbf;, score=0.839 total time= 1.8s
[CV 3/5] END .....C=1, gamma=0.0001, kernel=rbf;, score=0.860 total time= 1.8s
[CV 4/5] END .....C=1, gamma=0.0001, kernel=rbf;, score=0.846 total time= 1.4s
[CV 5/5] END .....C=1, gamma=0.0001, kernel=rbf;, score=0.863 total time= 1.2s
[CV 1/5] END ...C=1000, gamma=0.001, kernel=rbf;, score=0.934 total time= 0.8s
[CV 2/5] END ...C=1000, gamma=0.001, kernel=rbf;, score=0.920 total time= 0.8s
[CV 3/5] END ...C=1000, gamma=0.001, kernel=rbf;, score=0.935 total time= 0.8s
[CV 4/5] END ...C=1000, gamma=0.001, kernel=rbf;, score=0.926 total time= 0.8s
[CV 5/5] END ...C=1000, gamma=0.001, kernel=rbf;, score=0.943 total time= 0.7s
[CV 1/5] END ....C=100, gamma=0.001, kernel=rbf;, score=0.907 total time= 0.5s
[CV 2/5] END ....C=100, gamma=0.001, kernel=rbf;, score=0.912 total time= 0.5s
[CV 3/5] END ....C=100, gamma=0.001, kernel=rbf;, score=0.910 total time= 0.5s
[CV 4/5] END ....C=100, gamma=0.001, kernel=rbf;, score=0.911 total time= 0.5s
[CV 5/5] END ....C=100, gamma=0.001, kernel=rbf;, score=0.924 total time= 0.5s
[CV 1/5] END ....C=1000, kernel=linear;, score=0.866 total time= 2.3min
[CV 2/5] END ....C=1000, kernel=linear;, score=0.869 total time= 2.2min
Type here to search
30°C
ENG
18:36
19-09-2024

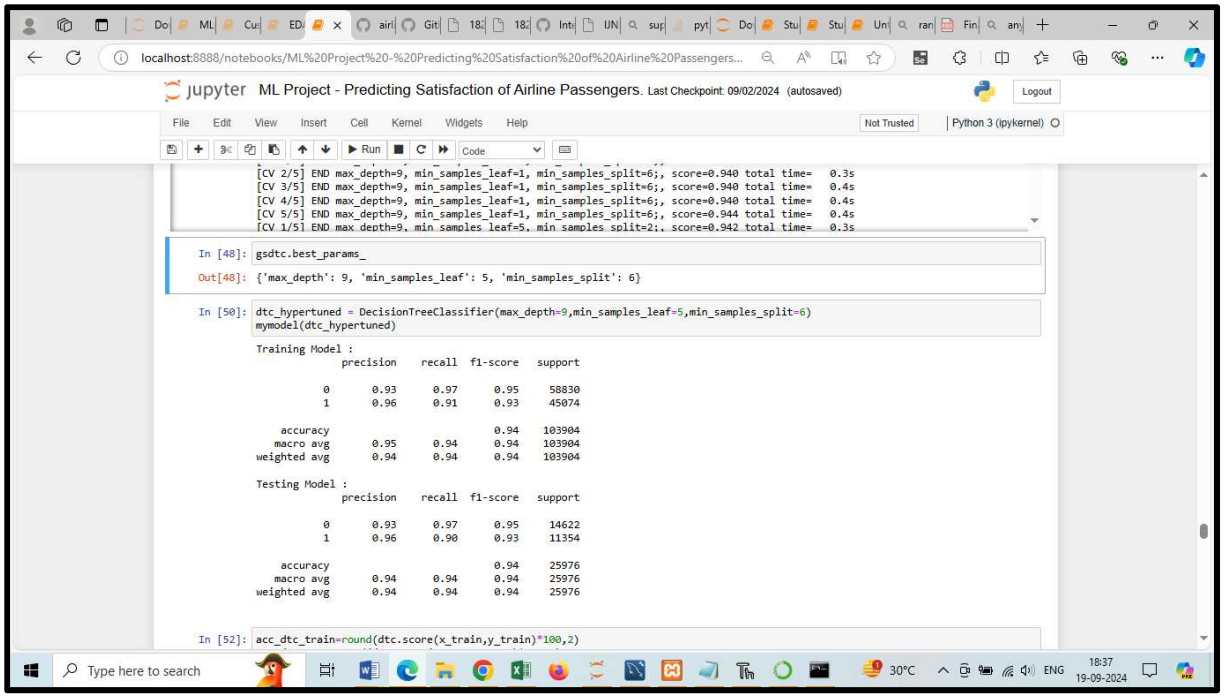

localhost:8888/notebooks/ML%20Project%20-%20Predicting%20Satisfaction%20of%20Airline%20Passengers..ipynb...
jupyter ML Project - Predicting Satisfaction of Airline Passengers. Last Checkpoint: 09/02/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [38]:
gssvm.best_params_
Out[38]:
{'kernel': 'rbf', 'gamma': 0.001, 'C': 1000}
In [39]:
svm_hypertuned = SVC(kernel='rbf',gamma=0.001,C=1000)
mymodel(svm_hypertuned)
Training Model :
precision recall f1-score support
0 0.94 0.96 0.95 58830
1 0.95 0.92 0.94 45074
accuracy 0.95 103904
macro avg 0.95 0.94 0.94 103904
weighted avg 0.95 0.95 0.95 103904
Testing Model :
precision recall f1-score support
0 0.94 0.96 0.95 14622
1 0.95 0.92 0.93 11354
accuracy 0.94 25976
macro avg 0.94 0.94 0.94 25976
weighted avg 0.94 0.94 0.94 25976
In [40]:
acc_svm_train=round(model_svm.score(x_train,y_train)*100,2)
acc_svm_test=round(model_svm.score(x_test,y_test)*100,2)
print("Training Accuracy: {} %".format(acc_svm_train))
Type here to search
30°C
ENG
18:36
19-09-2024

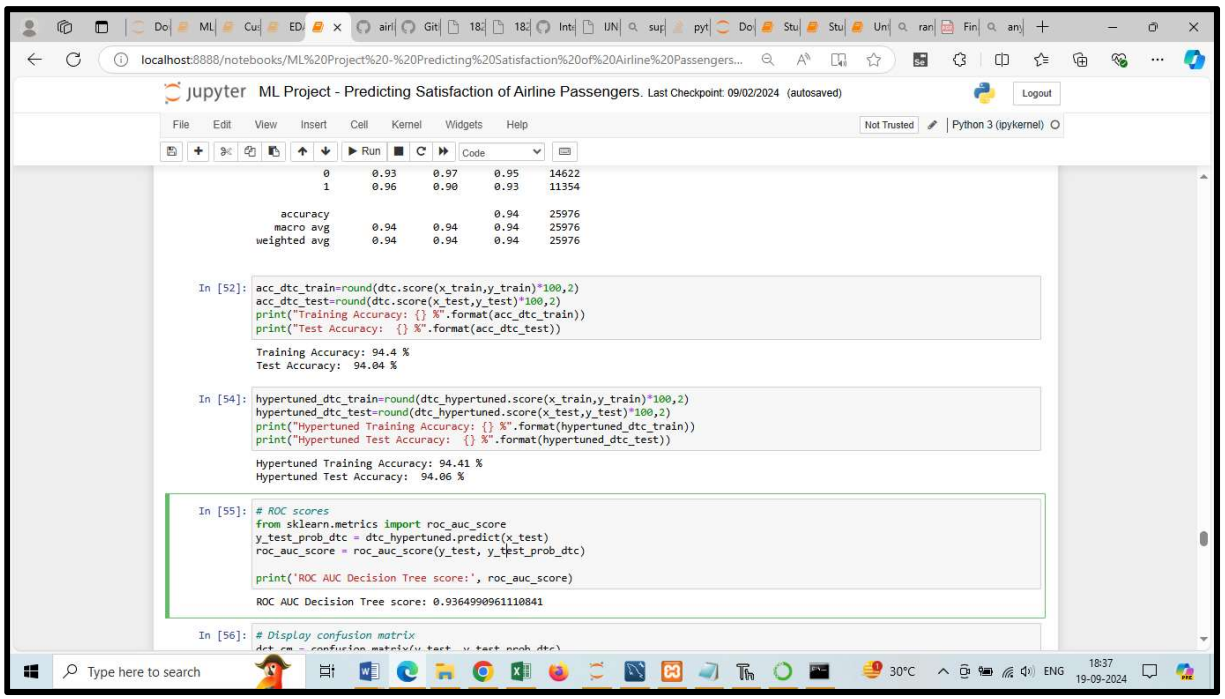

jupyter ML Project - Predicting Satisfaction of Airline Passengers. Last Checkpoint: 09/02/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
accuracy 0.94 25976
macro avg 0.94 0.94 0.94 25976
weighted avg 0.94 0.94 0.94 25976
In [40]:
acc_svm_train=round(model_svm.score(x_train,y_train)*100,2)
acc_svm_test=round(model_svm.score(x_test,y_test)*100,2)
print("Training Accuracy: {} %".format(acc_svm_train))
print("Test Accuracy: {} %".format(acc_svm_test))
Training Accuracy: 95.88 %
Test Accuracy: 95.4 %
In [41]:
svm_hypertuned_train = round(svm_hypertuned.score(x_train,y_train)*100,2)
svm_hypertuned_test = round(svm_hypertuned.score(x_test,y_test)*100,2)
print("Hypertuned Training Accuracy: {} %".format(svm_hypertuned_train))
print("Hypertuned Test Accuracy: {} %".format(svm_hypertuned_test))
Hypertuned Training Accuracy: 94.52 %
Hypertuned Test Accuracy: 94.27 %
In [42]:
# ROC scores
from sklearn.metrics import roc_auc_score
y_test_prob_svm = svm_hypertuned.predict(x_test)
roc_auc_svm = roc_auc_score(y_test, y_test_prob_svm)
print('ROC AUC SVM score:', roc_auc_svm)
ROC AUC SVM score: 0.9396349693926307
Type here to search
30°C
ENG
18:36
19-09-2024

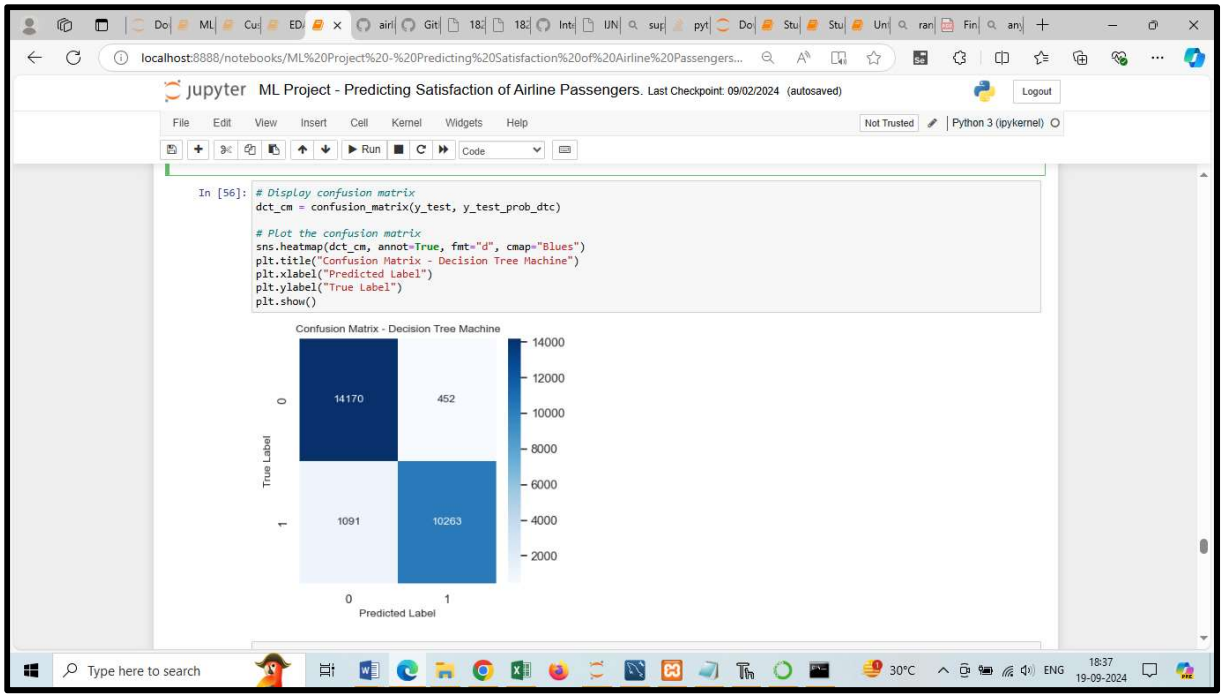

jupyter ML Project - Predicting Satisfaction of Airline Passengers. Last Checkpoint: 09/02/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [43]:
# Display confusion matrix
svm_cm = confusion_matrix(y_test, y_test_prob_svm)
# Plot the confusion matrix
sns.heatmap(svm_cm, annot=True, fmt="d", cmap="Blues")
plt.title("Confusion Matrix - SVM")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.show()
Confusion Matrix - SVM
14093
529
960
10394
True Label
Predicted Label
0
1
14000
12000
10000
8000
6000
4000
2000
Type here to search
30°C
ENG
18:36
19-09-2024

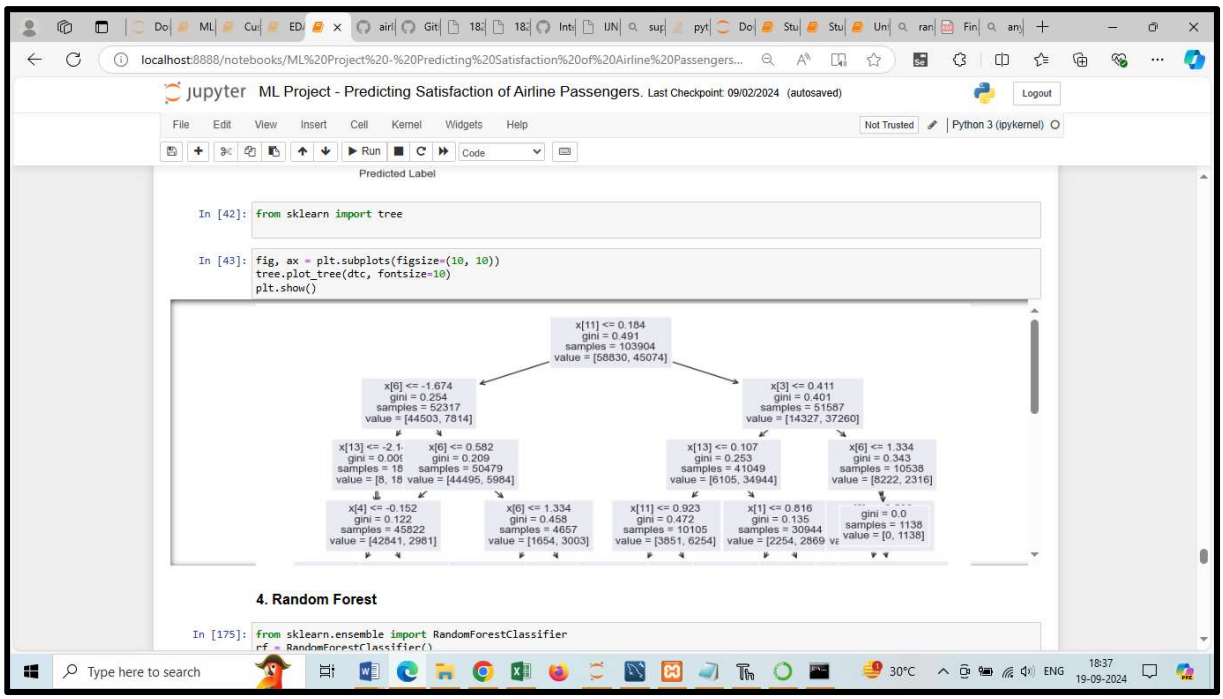

jupyter ML Project - Predicting Satisfaction of Airline Passengers. Last Checkpoint: 09/02/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
3. Decision Tree
In [51]:
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier(max_depth=9,min_samples_leaf=6,min_samples_split=6)
mymodel(dtc)
Training Model :
precision recall f1-score support
0 0.93 0.97 0.95 58830
1 0.96 0.91 0.93 45074
accuracy 0.94 103904
macro avg 0.95 0.94 0.94 103904
weighted avg 0.94 0.94 0.94 103904
Testing Model :
precision recall f1-score support
0 0.93 0.97 0.95 14622
1 0.96 0.90 0.93 11354
accuracy 0.94 25976
macro avg 0.94 0.94 0.94 25976
weighted avg 0.94 0.94 0.94 25976
Hyperparameter Tuning using GridSearchCV
In [46]:
from sklearn.model_selection import GridSearchCV
Type here to search
30°C
ENG
18:37
19-09-2024

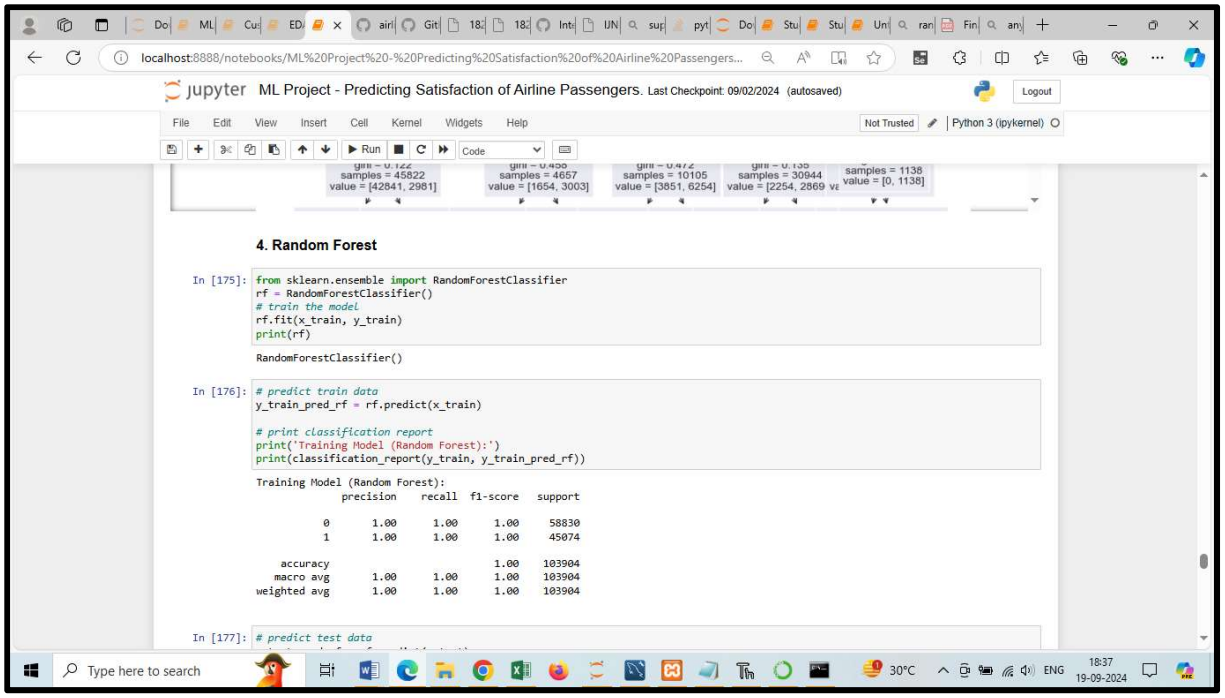

jupyter ML Project - Predicting Satisfaction of Airline Passengers. Last Checkpoint: 09/02/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Hyperparameter Tuning using GridSearchCV
In [46]:
from sklearn.model_selection import GridSearchCV
parameters={'max_depth': [7,8,9],
'min_samples_leaf': [1,5,6],
'min_samples_split': [2,5,6]}
gsdtc = GridSearchCV(dtc,parameters,verbose=3)
In [47]:
gsdtc.fit(x_train,y_train)
[CV 2/5] END max_depth=8, min_samples_leaf=6, min_samples_split=6;, score=0.933 total time= 0.3s
[CV 3/5] END max_depth=8, min_samples_leaf=6, min_samples_split=6;, score=0.934 total time= 0.3s
[CV 4/5] END max_depth=8, min_samples_leaf=6, min_samples_split=6;, score=0.934 total time= 0.3s
[CV 5/5] END max_depth=8, min_samples_leaf=6, min_samples_split=6;, score=0.938 total time= 0.3s
[CV 1/5] END max_depth=9, min_samples_leaf=1, min_samples_split=2;, score=0.941 total time= 0.4s
[CV 2/5] END max_depth=9, min_samples_leaf=1, min_samples_split=2;, score=0.940 total time= 0.4s
[CV 3/5] END max_depth=9, min_samples_leaf=1, min_samples_split=2;, score=0.940 total time= 0.4s
[CV 4/5] END max_depth=9, min_samples_leaf=1, min_samples_split=2;, score=0.940 total time= 0.4s
[CV 5/5] END max_depth=9, min_samples_leaf=1, min_samples_split=2;, score=0.943 total time= 0.3s
[CV 1/5] END max_depth=9, min_samples_leaf=1, min_samples_split=5;, score=0.942 total time= 0.4s
[CV 2/5] END max_depth=9, min_samples_leaf=1, min_samples_split=5;, score=0.940 total time= 0.4s
[CV 3/5] END max_depth=9, min_samples_leaf=1, min_samples_split=5;, score=0.940 total time= 0.4s
[CV 4/5] END max_depth=9, min_samples_leaf=1, min_samples_split=5;, score=0.940 total time= 0.3s
[CV 5/5] END max_depth=9, min_samples_leaf=1, min_samples_split=5;, score=0.944 total time= 0.3s
[CV 1/5] END max_depth=9, min_samples_leaf=1, min_samples_split=6;, score=0.941 total time= 0.3s
[CV 2/5] END max_depth=9, min_samples_leaf=1, min_samples_split=6;, score=0.940 total time= 0.3s
[CV 3/5] END max_depth=9, min_samples_leaf=1, min_samples_split=6;, score=0.940 total time= 0.4s
[CV 4/5] END max_depth=9, min_samples_leaf=1, min_samples_split=6;, score=0.940 total time= 0.4s
[CV 5/5] END max_depth=9, min_samples_leaf=1, min_samples_split=6;, score=0.944 total time= 0.4s
[CV 1/5] END max_depth=9, min_samples_leaf=5, min_samples_split=2;, score=0.942 total time= 0.3s
In [48]:
gsdtc.best_params_
Out[48]:
{'max_depth': 9, 'min_samples_leaf': 5, 'min_samples_split': 6}
Type here to search
30°C
ENG
18:37
19-09-2024

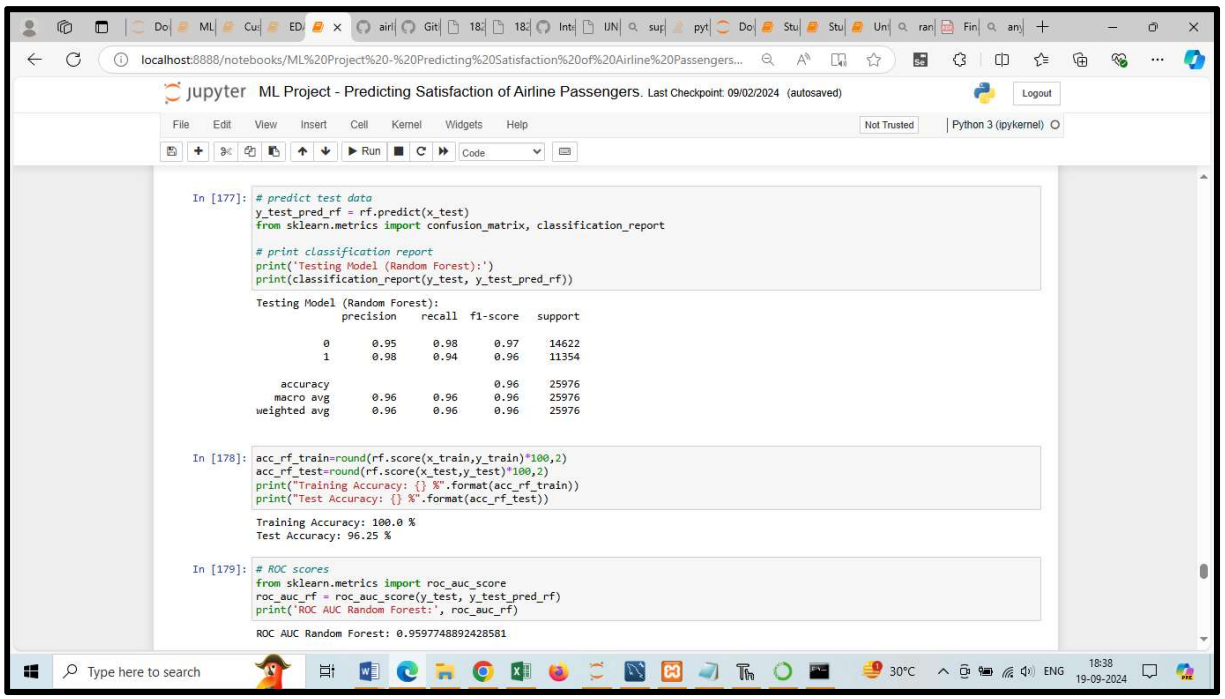

localhost:8888/notebooks/ML%20Project%20-%20Predicting%20Satisfaction%20of%20Airline%20Passengers...
jupyter ML Project - Predicting Satisfaction of Airline Passengers. Last Checkpoint: 09/02/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
[CV 2/5] END max_depth=9, min_samples_leaf=1, min_samples_split=6;, score=0.940 total time= 0.3s
[CV 3/5] END max_depth=9, min_samples_leaf=1, min_samples_split=6;, score=0.940 total time= 0.4s
[CV 4/5] END max_depth=9, min_samples_leaf=1, min_samples_split=6;, score=0.940 total time= 0.4s
[CV 5/5] END max_depth=9, min_samples_leaf=1, min_samples_split=6;, score=0.944 total time= 0.4s
[CV 1/5] END max_depth=9, min_samples_leaf=5, min_samples_split=2;, score=0.942 total time= 0.3s
In [48]: gsdtc.best_params_
Out[48]: {'max_depth': 9, 'min_samples_leaf': 5, 'min_samples_split': 6}
In [50]: dtc_hypertuned = DecisionTreeClassifier(max_depth=9,min_samples_leaf=5,min_samples_split=6)
mymodel(dtc_hypertuned)
Training Model :
precision recall f1-score support
0 0.93 0.97 0.95 58830
1 0.96 0.91 0.93 45074
accuracy 0.94 103904
macro avg 0.95 0.94 0.94 103904
weighted avg 0.94 0.94 0.94 103904
Testing Model :
precision recall f1-score support
0 0.93 0.97 0.95 14622
1 0.96 0.90 0.93 11354
accuracy 0.94 25976
macro avg 0.94 0.94 0.94 25976
weighted avg 0.94 0.94 0.94 25976
In [52]: acc_dtc_train=round(dtc.score(x_train,y_train)*100,2)
Type here to search
30°C
ENG
18:37
19-09-2024

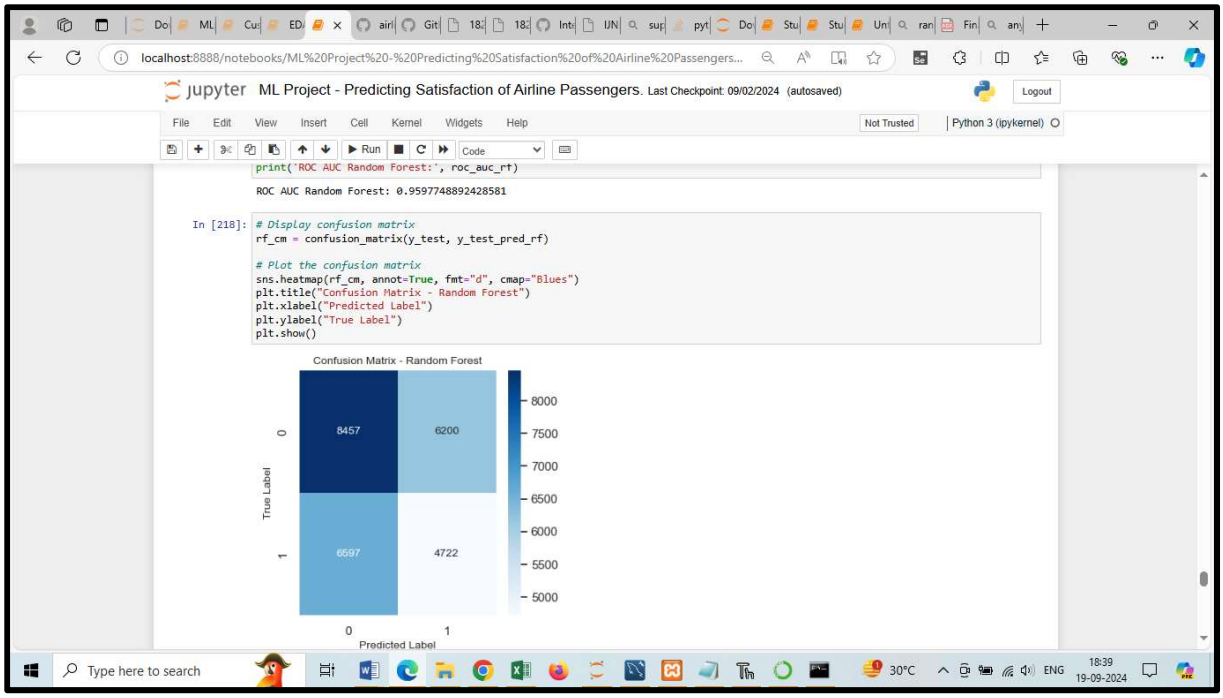

localhost:8888/notebooks/ML%20Project%20-%20Predicting%20Satisfaction%20of%20Airline%20Passengers...
jupyter ML Project - Predicting Satisfaction of Airline Passengers. Last Checkpoint: 09/02/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
0 0.93 0.97 0.95 14622
1 0.96 0.90 0.93 11354
accuracy 0.94 25976
macro avg 0.94 0.94 0.94 25976
weighted avg 0.94 0.94 0.94 25976
In [52]: acc_dtc_train=round(dtc.score(x_train,y_train)*100,2)
acc_dtc_test=round(dtc.score(x_test,y_test)*100,2)
print("Training Accuracy: {} %".format(acc_dtc_train))
print("Test Accuracy: {} %".format(acc_dtc_test))
Training Accuracy: 94.4 %
Test Accuracy: 94.04 %
In [54]: hypertuned_dtc_train=round(dtc_hypertuned.score(x_train,y_train)*100,2)
hypertuned_dtc_test=round(dtc_hypertuned.score(x_test,y_test)*100,2)
print("Hypertuned Training Accuracy: {} %".format(hypertuned_dtc_train))
print("Hypertuned Test Accuracy: {} %".format(hypertuned_dtc_test))
Hypertuned Training Accuracy: 94.41 %
Hypertuned Test Accuracy: 94.06 %
In [55]: # ROC scores
from sklearn.metrics import roc_auc_score
y_test_prob_dtc = dtc_hypertuned.predict(x_test)
roc_auc_score = roc_auc_score(y_test, y_test_prob_dtc)
print('ROC AUC Decision Tree score:', roc_auc_score)
ROC AUC Decision Tree score: 0.9364990961110841
In [56]: # Display confusion matrix
Type here to search
30°C
ENG
18:37
19-09-2024


localhost:8888/notebooks/ML%20Project%20-%20Predicting%20Satisfaction%20of%20Airline%20Passengers...
jupyter ML Project - Predicting Satisfaction of Airline Passengers. Last Checkpoint: 09/02/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [56]:
# Display confusion matrix
dct_cm = confusion_matrix(y_test, y_test_prob_dtc)
# Plot the confusion matrix
sns.heatmap(dct_cm, annot=True, fmt="d", cmap="Blues")
plt.title("Confusion Matrix - Decision Tree Machine")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.show()
Confusion Matrix - Decision Tree Machine
14170
452
1091
10263
True Label
Predicted Label
14000
12000
10000
8000
6000
4000
2000
Type here to search
30°C
ENG
18:37
19-09-2024

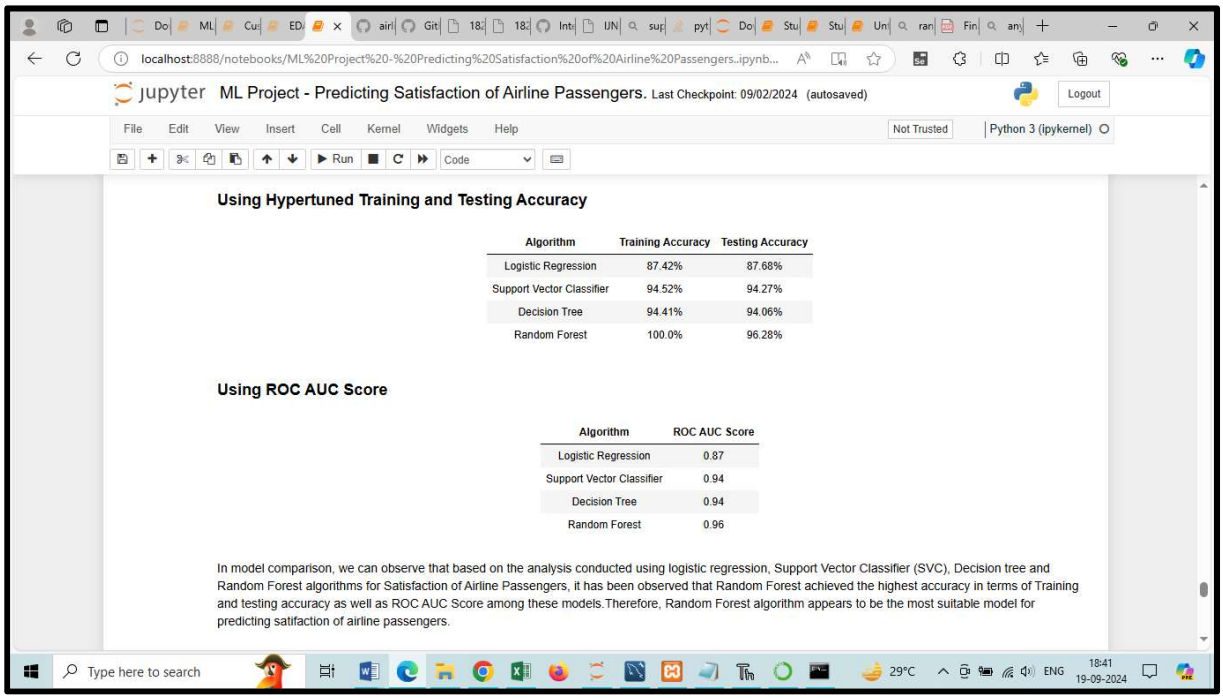

localhost:8888/notebooks/ML%20Project%20-%20Predicting%20Satisfaction%20of%20Airline%20Passengers...
jupyter ML Project - Predicting Satisfaction of Airline Passengers. Last Checkpoint: 09/02/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Predicted Label
In [42]:
from sklearn import tree
In [43]:
fig, ax = plt.subplots(figsize=(10, 10))
tree.plot_tree(dtc, fontsize=10)
plt.show()
x[11] <= 0.184
gini = 0.491
samples = 103904
value = [58830, 45074]
x[6] <= -1.674
gini = 0.254
samples = 52317
value = [44503, 7814]
x[3] <= 0.411
gini = 0.401
samples = 51587
value = [14327, 37260]
x[13] <= -2.1
gini = 0.00
samples = 18
value = [8, 18
x[6] <= 0.582
gini = 0.209
samples = 50479
value = [44495, 5984]
x[13] <= 0.107
gini = 0.253
samples = 41049
value = [6105, 34944]
x[6] <= 1.334
gini = 0.343
samples = 10538
value = [8222, 2316]
x[4] <= -0.152
gini = 0.122
samples = 45822
value = [42841, 2981]
x[6] <= 1.334
gini = 0.458
samples = 4657
value = [1654, 3003]
x[11] <= 0.923
gini = 0.472
samples = 10105
value = [3851, 6254]
x[1] <= 0.816
gini = 0.135
samples = 30944
value = [2254, 2869
gini = 0.0
samples = 1138
value = [0, 1138]
4. Random Forest
In [175]:
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()
Type here to search
30°C
ENG
18:37
19-09-2024


localhost:8888/notebooks/ML%20Project%20-%20Predicting%20Satisfaction%20of%20Airline%20Passengers...
jupyter ML Project - Predicting Satisfaction of Airline Passengers. Last Checkpoint: 09/02/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
samples = 45822
value = [42841, 2981]
samples = 4657
value = [1654, 3003]
samples = 10105
value = [3851, 6254]
samples = 30944
value = [2254, 2869
samples = 1138
value = [0, 1138]
4. Random Forest
In [175]:
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()
# train the model
rf.fit(x_train, y_train)
print(rf)
RandomForestClassifier()
In [176]:
# predict train data
y_train_pred_rf = rf.predict(x_train)
# print classification report
print('Training Model (Random Forest):')
print(classification_report(y_train, y_train_pred_rf))
Training Model (Random Forest):
precision recall f1-score support
0 1.00 1.00 1.00 58830
1 1.00 1.00 1.00 45074
accuracy 1.00 103904
macro avg 1.00 1.00 1.00 103904
weighted avg 1.00 1.00 1.00 103904
In [177]:
predict test data
Type here to search
30°C
ENG
18:37
19-09-2024

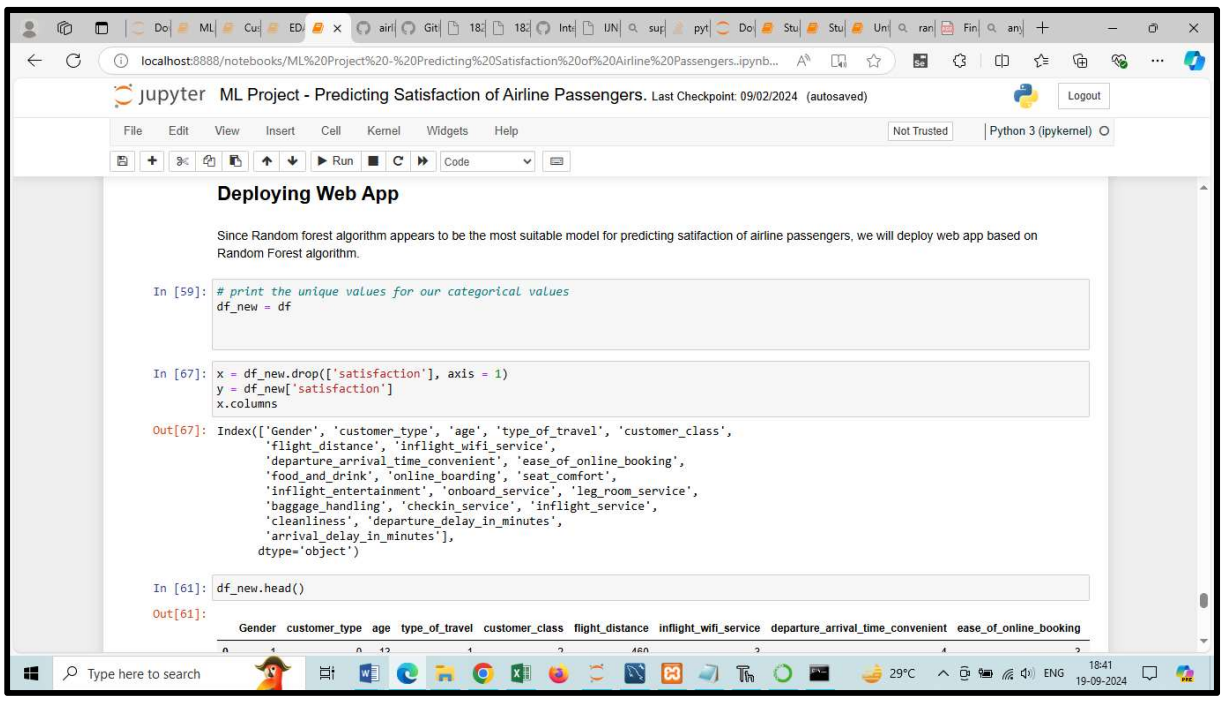

localhost:8888/notebooks/ML%20Project%20-%20Predicting%20Satisfaction%20of%20Airline%20Passengers...
jupyter ML Project - Predicting Satisfaction of Airline Passengers. Last Checkpoint: 09/02/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [177]:
# predict test data
y_test_pred_rf = rf.predict(x_test)
from sklearn.metrics import confusion_matrix, classification_report
# print classification report
print('Testing Model (Random Forest):')
print(classification_report(y_test, y_test_pred_rf))
Testing Model (Random Forest):
precision recall f1-score support
0 0.95 0.98 0.97 14622
1 0.98 0.94 0.96 11354
accuracy 0.96 25976
macro avg 0.96 0.96 0.96 25976
weighted avg 0.96 0.96 0.96 25976
In [178]:
acc_rf_train=round(rf.score(x_train,y_train)*100,2)
acc_rf_test=round(rf.score(x_test,y_test)*100,2)
print("Training Accuracy: {} %".format(acc_rf_train))
print("Test Accuracy: {} %".format(acc_rf_test))
Training Accuracy: 100.0 %
Test Accuracy: 96.25 %
In [179]:
# ROC scores
from sklearn.metrics import roc_auc_score
roc_auc_rf = roc_auc_score(y_test, y_test_pred_rf)
print('ROC AUC Random Forest:', roc_auc_rf)
ROC AUC Random Forest: 0.9597748892428581
Type here to search
30°C
ENG
18:38
19-09-2024

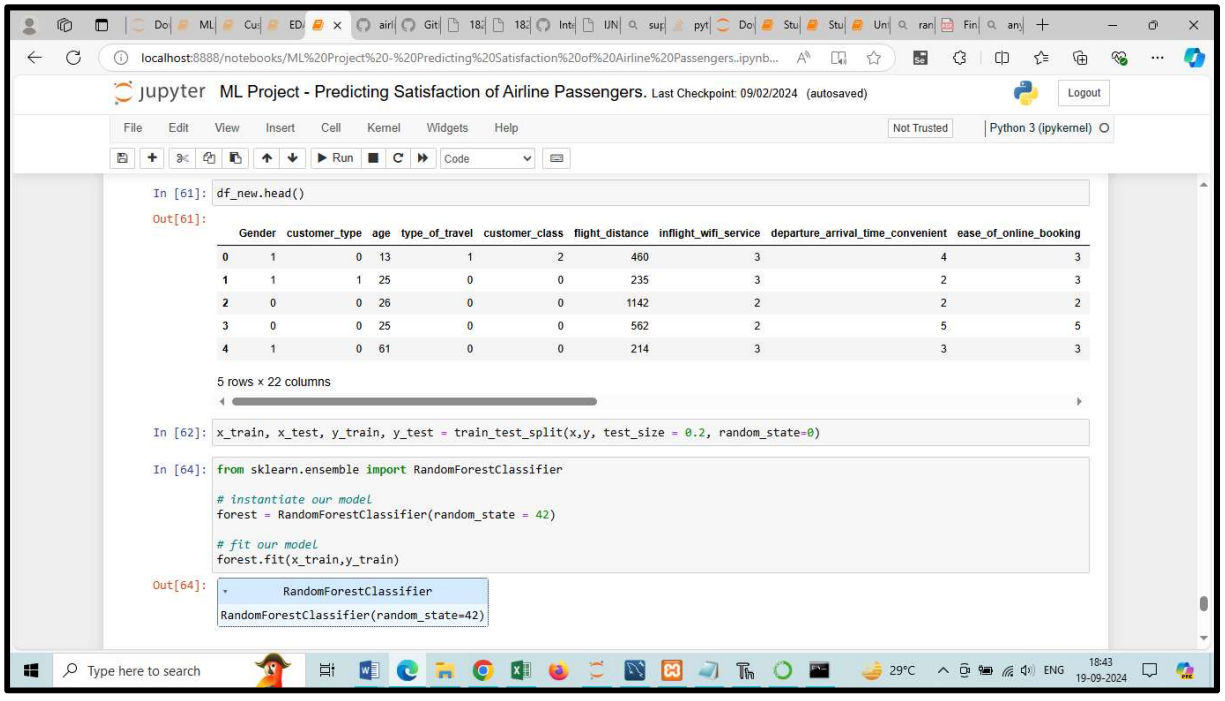


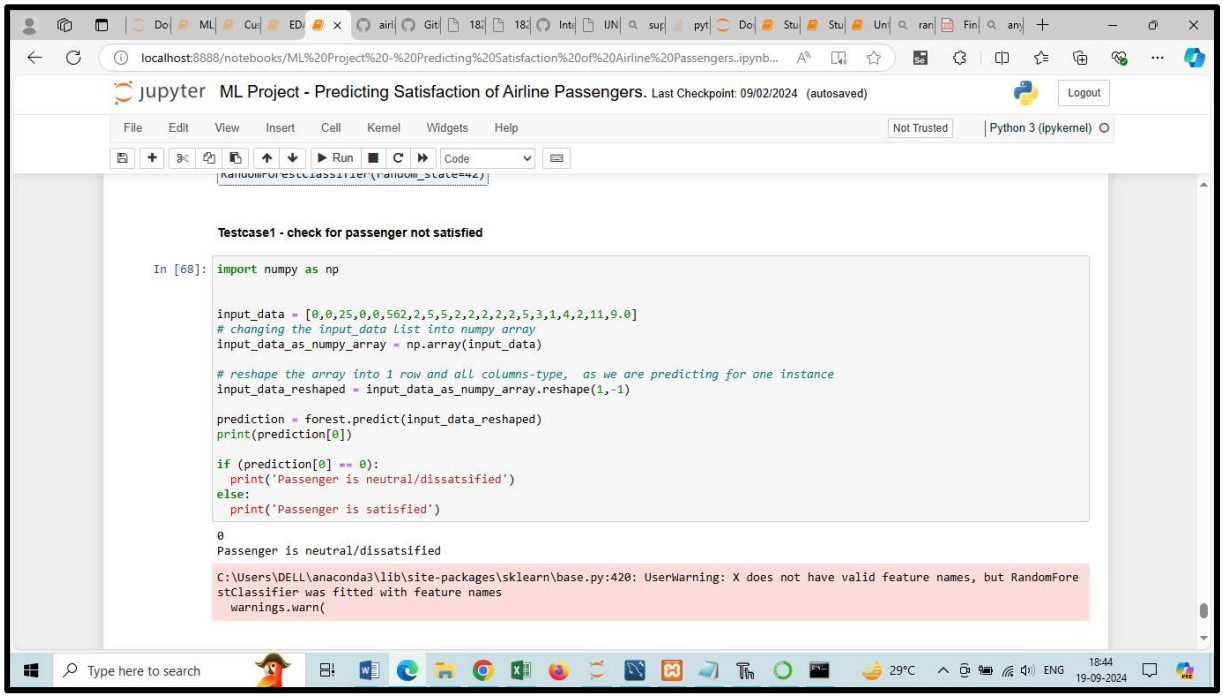
9. Model Comparison:

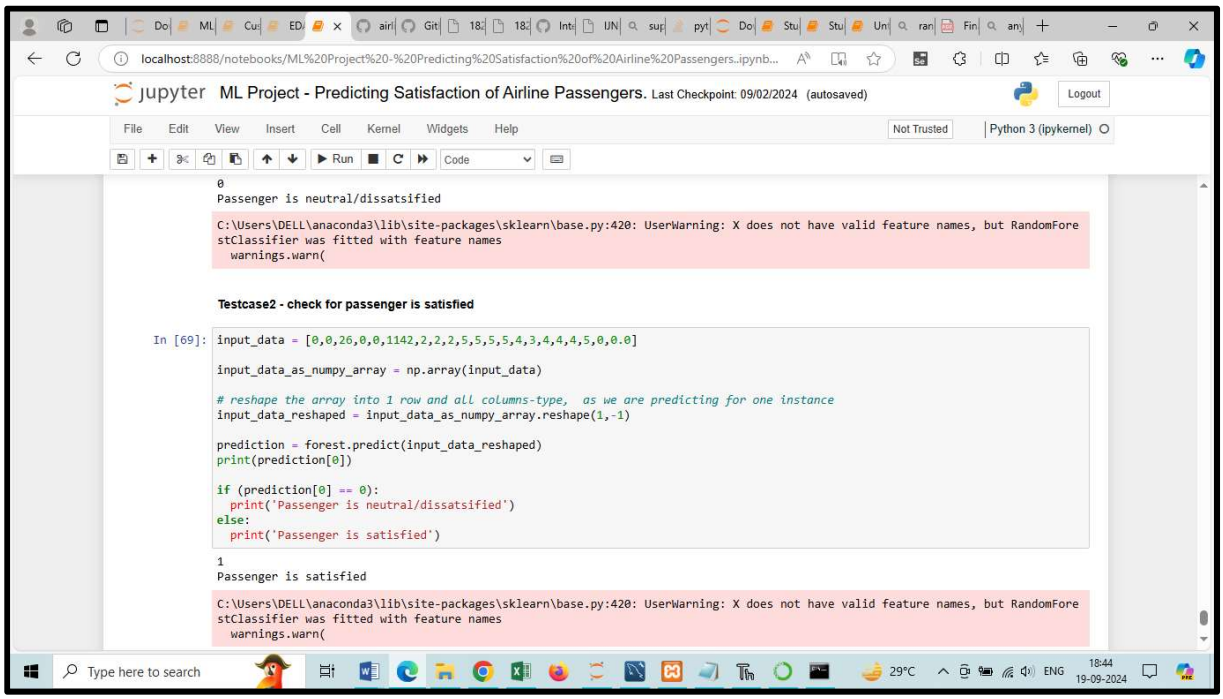


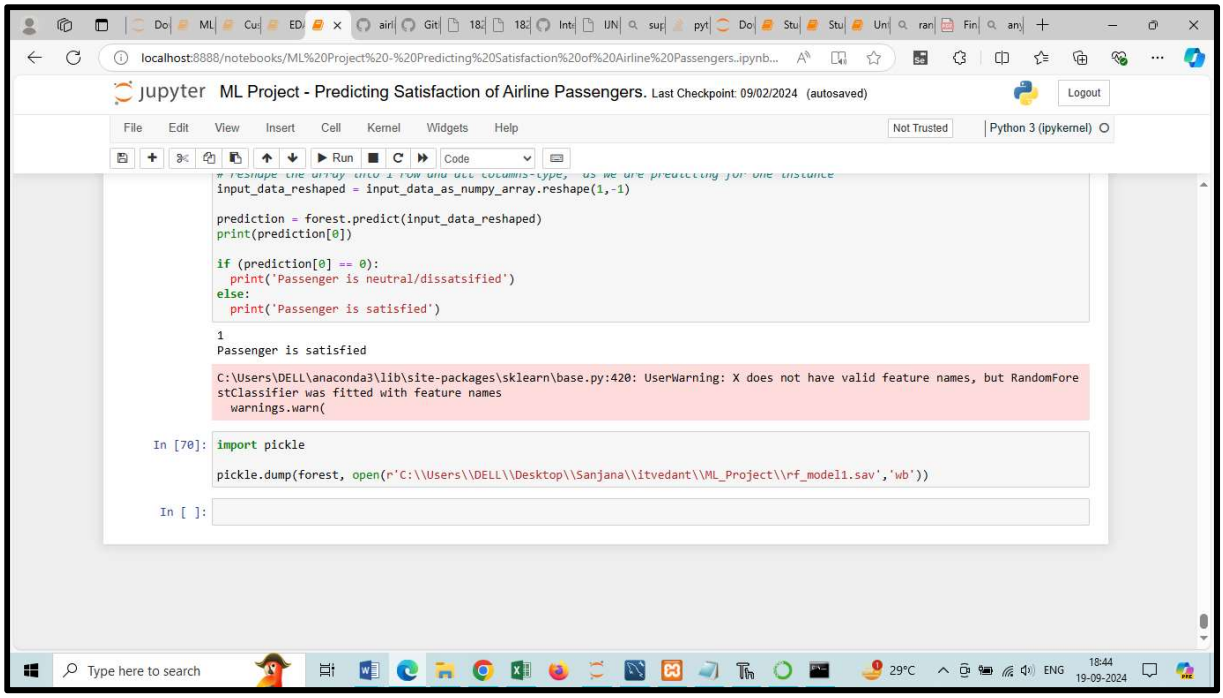
10.Web App Deployment

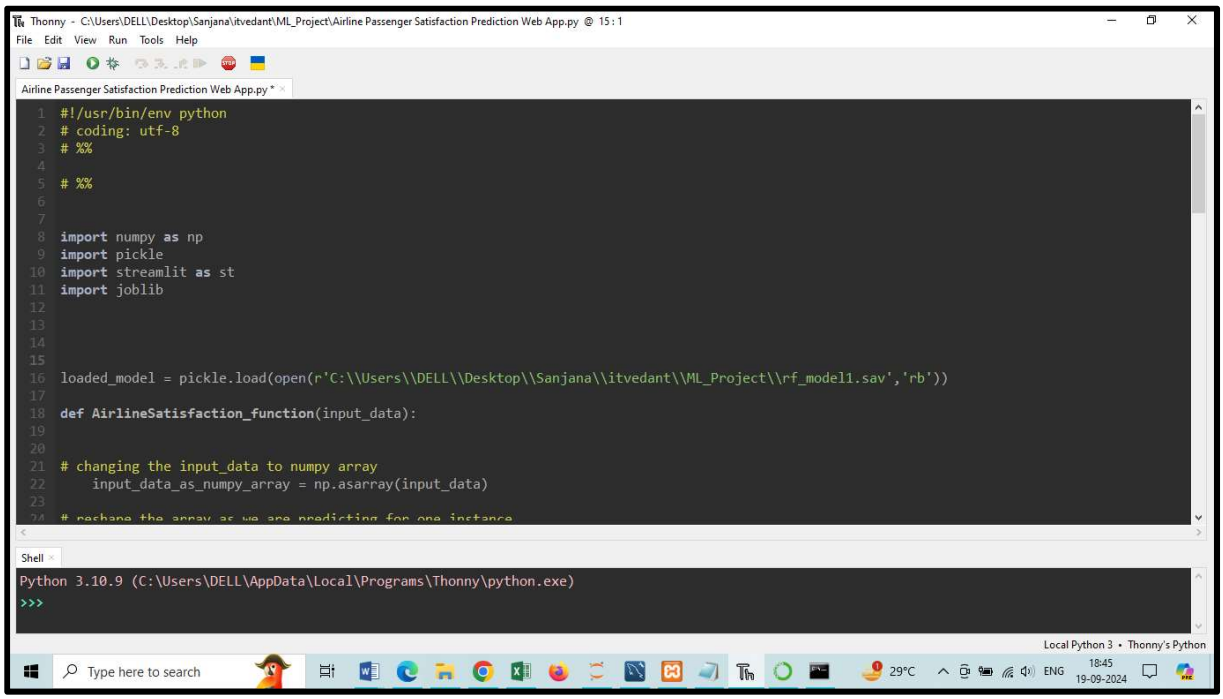


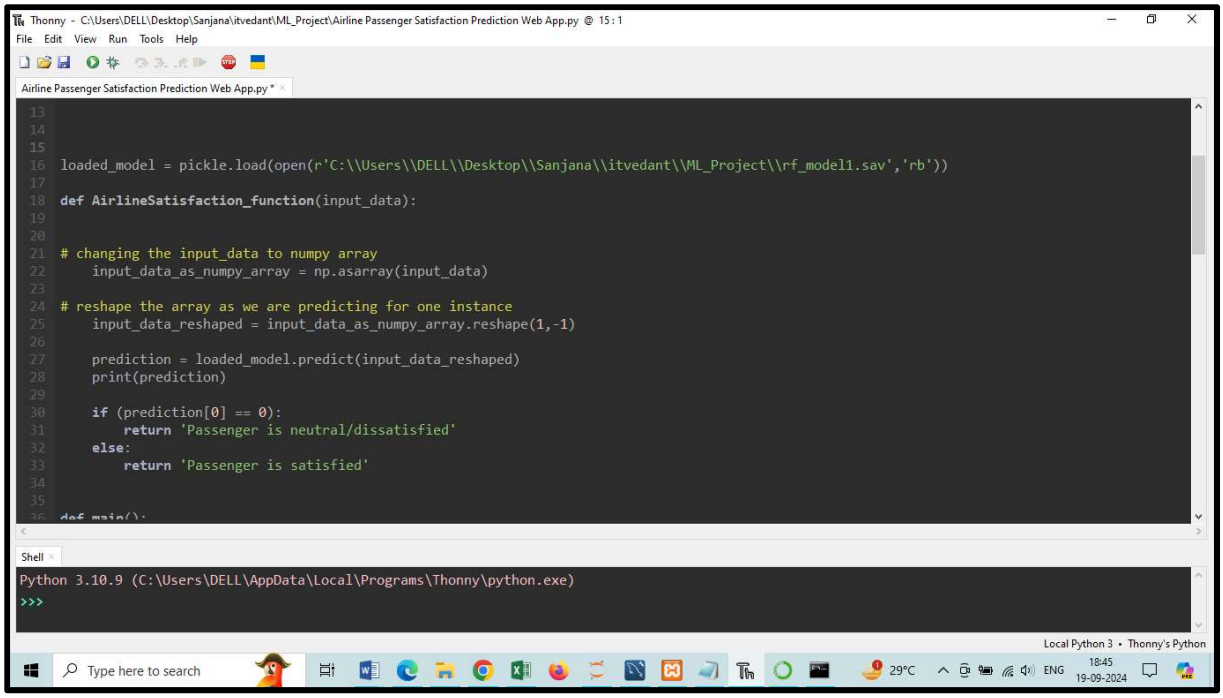


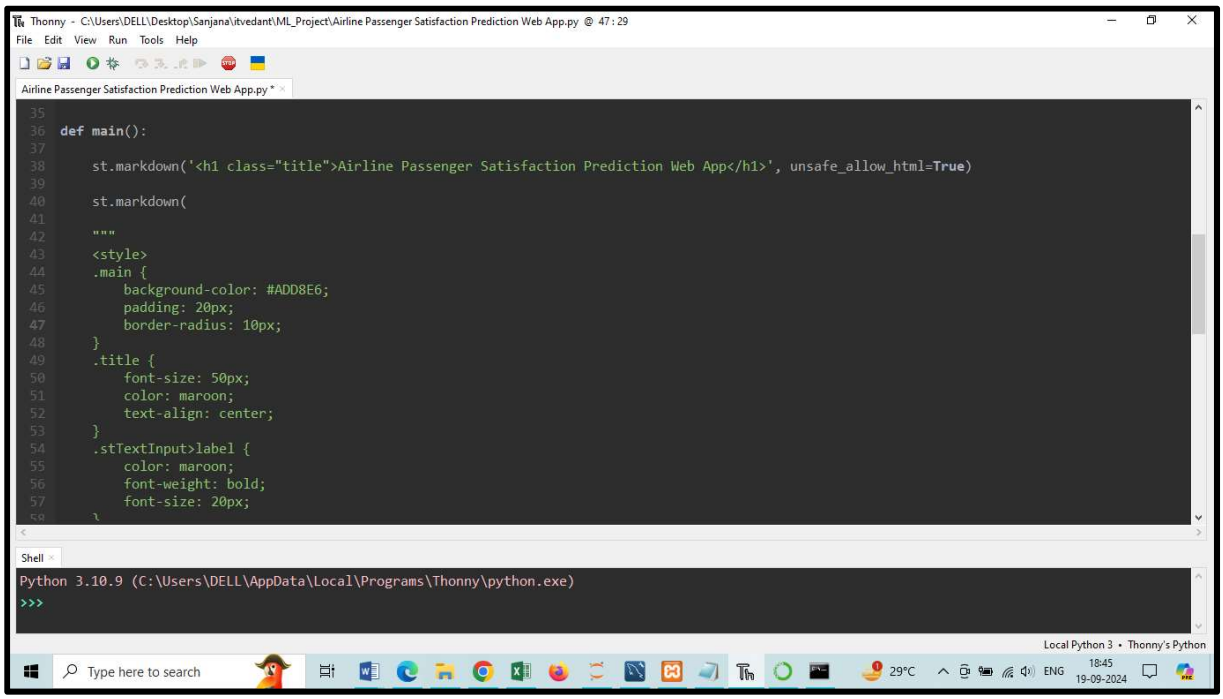

localhost:8888/notebooks/ML%20Project%20-%20Predicting%20Satisfaction%20of%20Airline%20Passengers..ipynb...
jupyter ML Project - Predicting Satisfaction of Airline Passengers. Last Checkpoint: 09/02/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Testcase1 - check for passenger not satisfied
In [68]:
import numpy as np
input_data = [0,0,25,0,0,562,2,5,5,2,2,2,2,2,5,3,1,4,2,11,9.0]
# changing the input_data list into numpy array
input_data_as_numpy_array = np.array(input_data)
# reshape the array into 1 row and all columns-type, as we are predicting for one instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
prediction = forest.predict(input_data_reshaped)
print(prediction[0])
if (prediction[0] == 0):
print('Passenger is neutral/dissatsified')
else:
print('Passenger is satisfied')
0
Passenger is neutral/dissatsified
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarning: X does not have valid feature names, but RandomFore
stClassifier was fitted with feature names
warnings.warn(
Type here to search
29°C
ENG
18:44
19-09-2024

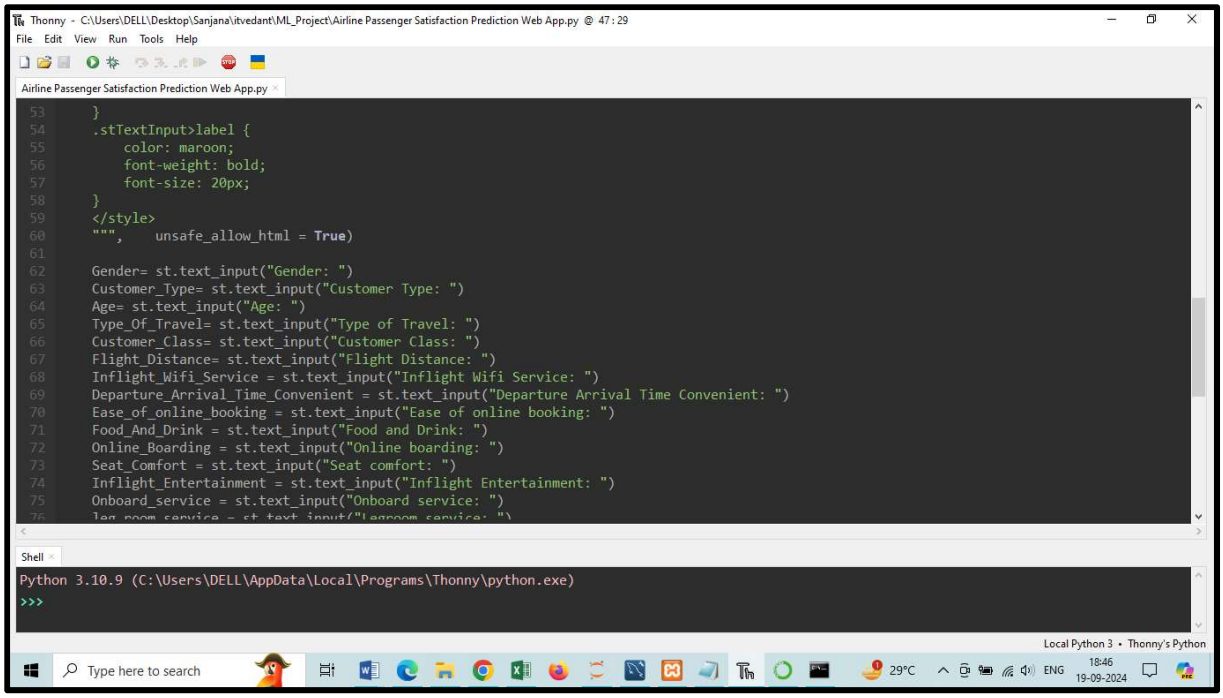

localhost:8888/notebooks/ML%20Project%20-%20Predicting%20Satisfaction%20of%20Airline%20Passengers..ipynb...
jupyter ML Project - Predicting Satisfaction of Airline Passengers. Last Checkpoint: 09/02/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
0
Passenger is neutral/dissatsified
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarning: X does not have valid feature names, but RandomFore
stClassifier was fitted with feature names
warnings.warn(
Testcase2 - check for passenger is satisfied
In [69]:
input_data = [0,0,26,0,0,1142,2,2,2,5,5,5,5,4,3,4,4,4,5,0,0.0]
input_data_as_numpy_array = np.array(input_data)
# reshape the array into 1 row and all columns-type, as we are predicting for one instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
prediction = forest.predict(input_data_reshaped)
print(prediction[0])
if (prediction[0] == 0):
print('Passenger is neutral/dissatsified')
else:
print('Passenger is satisfied')
1
Passenger is satisfied
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarning: X does not have valid feature names, but RandomFore
stClassifier was fitted with feature names
warnings.warn(
Type here to search
29°C
ENG
18:44
19-09-2024

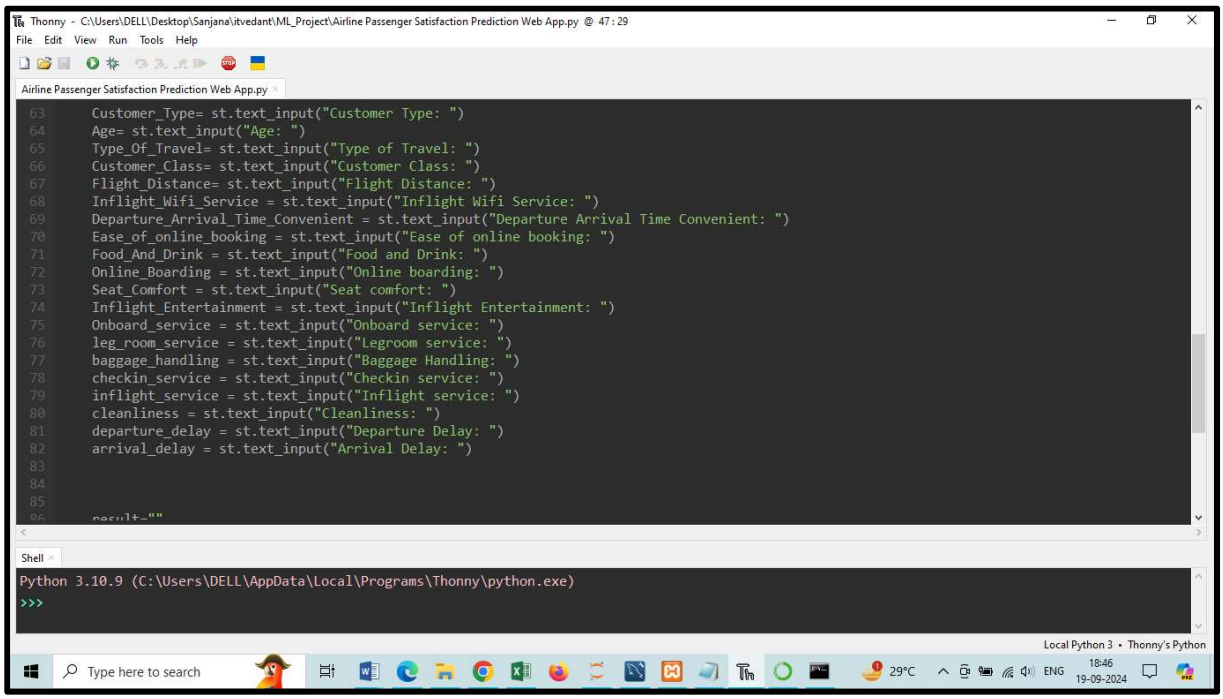

localhost:8888/notebooks/ML%20Project%20-%20Predicting%20Satisfaction%20of%20Airline%20Passengers..ipynb...
jupyter ML Project - Predicting Satisfaction of Airline Passengers. Last Checkpoint: 09/02/2024 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
prediction = forest.predict(input_data_reshaped)
print(prediction[0])
if (prediction[0] == 0):
print('Passenger is neutral/dissatsified')
else:
print('Passenger is satisfied')
1
Passenger is satisfied
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarning: X does not have valid feature names, but RandomFore
stClassifier was fitted with feature names
warnings.warn(
In [70]:
import pickle
pickle.dump(forest, open(r'C:\\Users\\DELL\\Desktop\\Sanjana\\itvedant\\ML_Project\\rf_model1.sav','wb'))
In [ ]:
Type here to search
29°C
ENG
18:44
19-09-2024

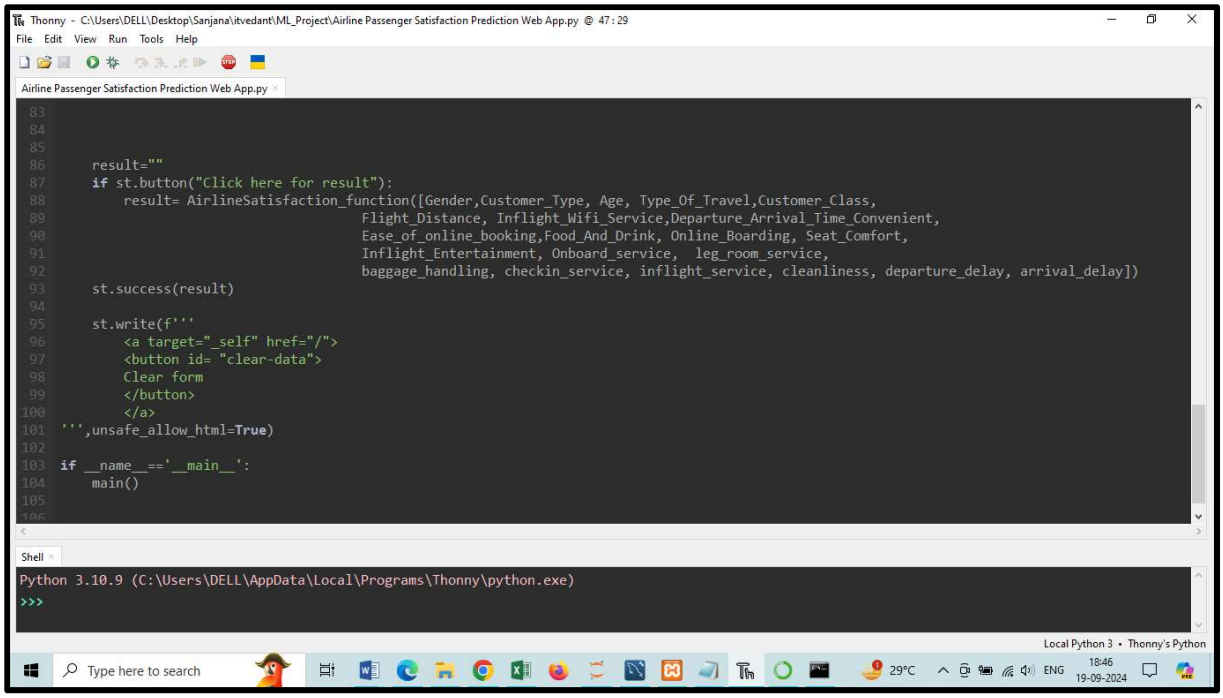

Thonny - C:\Users\DELL\Desktop\Sanjana\itvedant\ML_Project\Airline Passenger Satisfaction Prediction Web App.py @ 15 : 1
File Edit View Run Tools Help
Airline Passenger Satisfaction Prediction Web App.py *
#!/usr/bin/env python
# coding: utf-8
# %%
# %%
import numpy as np
import pickle
import streamlit as st
import joblib
loaded_model = pickle.load(open(r'C:\\Users\\DELL\\Desktop\\Sanjana\\itvedant\\ML_Project\\rf_model1.sav','rb'))
def AirlineSatisfaction_function(input_data):
# changing the input_data to numpy array
input_data_as_numpy_array = np.asarray(input_data)
Shell
Python 3.10.9 (C:\Users\DELL\AppData\Local\Programs\Thonny\python.exe)
>>>
Local Python 3 • Thonny's Python
Type here to search
29°C
ENG
18:45
19-09-2024


Thonny - C:\Users\DELL\Desktop\Sanjana\itvedant\ML_Project\Airline Passenger Satisfaction Prediction Web App.py @ 15 : 1
File Edit View Run Tools Help
Airline Passenger Satisfaction Prediction Web App.py *
loaded_model = pickle.load(open(r'C:\\Users\\DELL\\Desktop\\Sanjana\\itvedant\\ML_Project\\rf_model1.sav','rb'))
def AirlineSatisfaction_function(input_data):
# changing the input_data to numpy array
input_data_as_numpy_array = np.asarray(input_data)
# reshape the array as we are predicting for one instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
prediction = loaded_model.predict(input_data_reshaped)
print(prediction)
if (prediction[0] == 0):
return 'Passenger is neutral/dissatisfied'
else:
return 'Passenger is satisfied'
Shell
Python 3.10.9 (C:\Users\DELL\AppData\Local\Programs\Thonny\python.exe)
>>>
Local Python 3 • Thonny's Python
Type here to search
29°C
ENG
18:45
19-09-2024

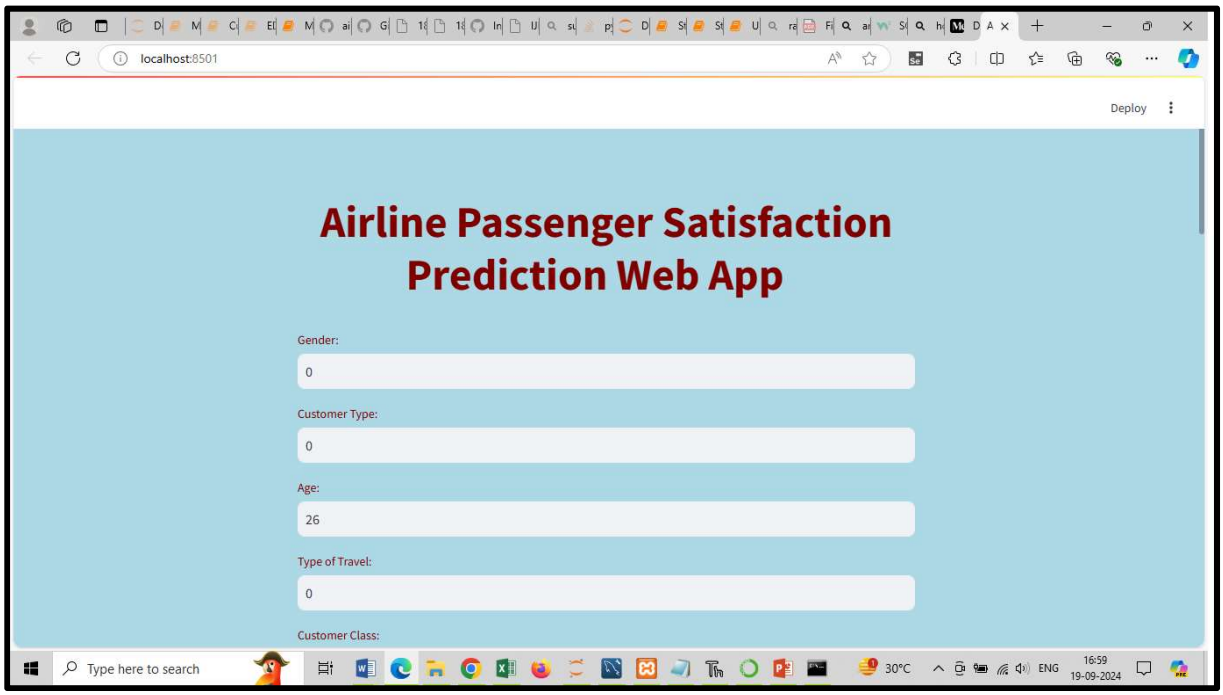

Thonny - C:\Users\DELL\Desktop\Sanjana\itvedant\ML_Project\Airline Passenger Satisfaction Prediction Web App.py @ 47 : 29
File Edit View Run Tools Help
Airline Passenger Satisfaction Prediction Web App.py *
def main():
st.markdown('<h1 class="title">Airline Passenger Satisfaction Prediction Web App</h1>', unsafe_allow_html=True)
st.markdown(
"""
<style>
.main {
background-color: #ADD8E6;
padding: 20px;
border-radius: 10px;
}
.title {
font-size: 50px;
color: maroon;
text-align: center;
}
.stTextInput>label {
color: maroon;
font-weight: bold;
font-size: 20px;
Shell
Python 3.10.9 (C:\Users\DELL\AppData\Local\Programs\Thonny\python.exe)
>>>
Local Python 3 • Thonny's Python
Type here to search
29°C
ENG
18:45
19-09-2024

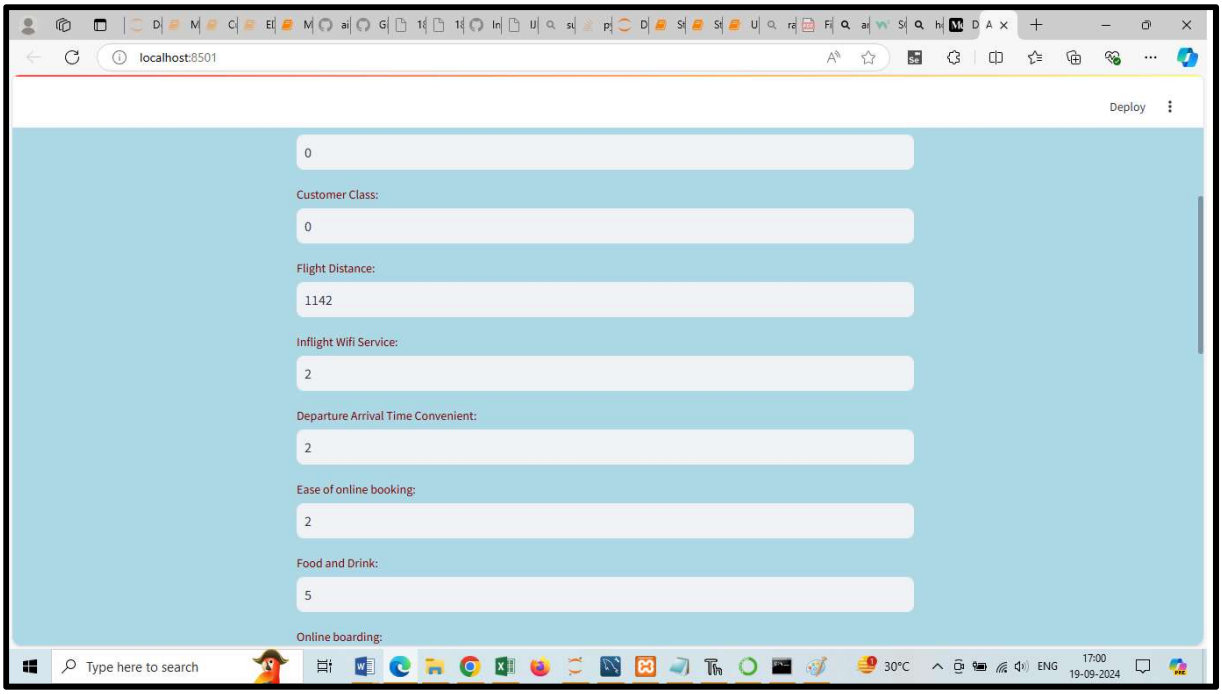

Thonny - C:\Users\DELL\Desktop\Sanjana\itvedant\ML_Project\Airline Passenger Satisfaction Prediction Web App.py @ 47 : 29
File Edit View Run Tools Help
Airline Passenger Satisfaction Prediction Web App.py
}
.stTextInput>label {
color: maroon;
font-weight: bold;
font-size: 20px;
}
</style>
""", unsafe_allow_html = True)
Gender= st.text_input("Gender: ")
Customer_Type= st.text_input("Customer Type: ")
Age= st.text_input("Age: ")
Type_Of_Travel= st.text_input("Type of Travel: ")
Customer_Class= st.text_input("Customer Class: ")
Flight_Distance= st.text_input("Flight Distance: ")
Inflight_Wifi_Service = st.text_input("Inflight Wifi Service: ")
Departure_Arrival_Time_Convenient = st.text_input("Departure Arrival Time Convenient: ")
Ease_of_online_booking = st.text_input("Ease of online booking: ")
Food_And_Drink = st.text_input("Food and Drink: ")
Online_Boarding = st.text_input("Online boarding: ")
Seat_Comfort = st.text_input("Seat comfort: ")
Inflight_Entertainment = st.text_input("Inflight Entertainment: ")
Onboard_service = st.text_input("Onboard service: ")
Shell
Python 3.10.9 (C:\Users\DELL\AppData\Local\Programs\Thonny\python.exe)
>>>
Local Python 3 • Thonny's Python
Type here to search
29°C
ENG
18:46
19-09-2024

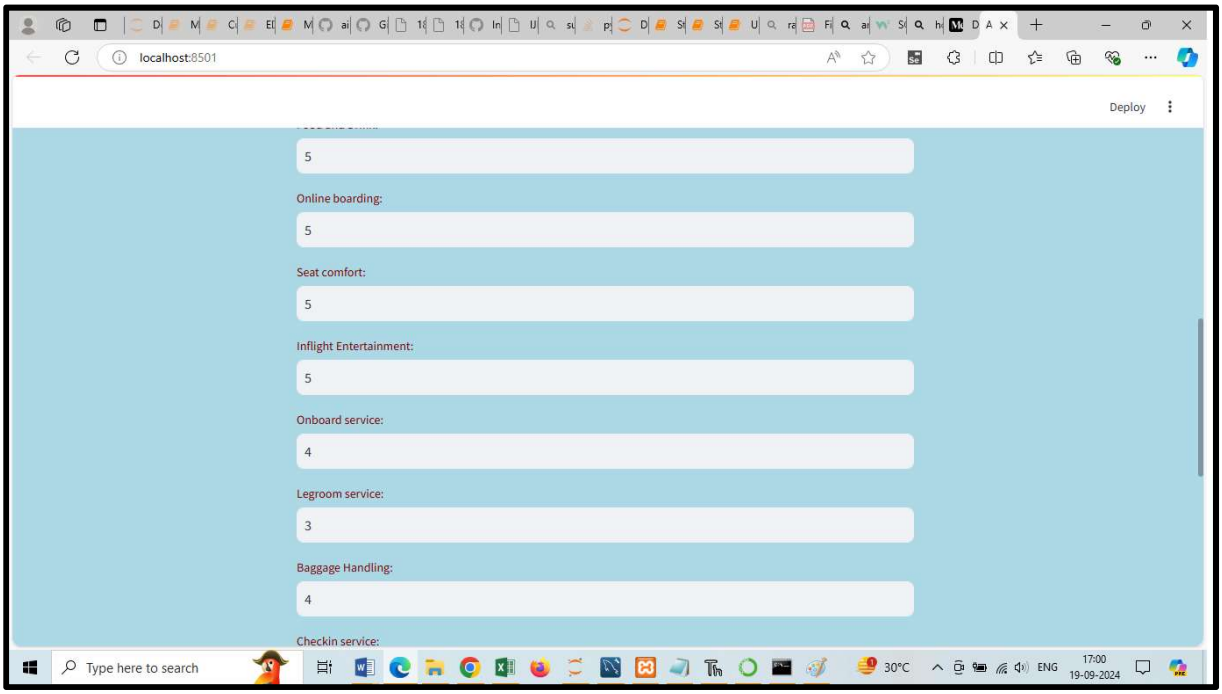

Thonny - C:\Users\DELL\Desktop\Sanjana\itvedant\ML_Project\Airline Passenger Satisfaction Prediction Web App.py @ 47 : 29
File Edit View Run Tools Help
Airline Passenger Satisfaction Prediction Web App.py
Customer_Type= st.text_input("Customer Type: ")
Age= st.text_input("Age: ")
Type_Of_Travel= st.text_input("Type of Travel: ")
Customer_Class= st.text_input("Customer Class: ")
Flight_Distance= st.text_input("Flight Distance: ")
Inflight_Wifi_Service = st.text_input("Inflight Wifi Service: ")
Departure_Arrival_Time_Convenient = st.text_input("Departure Arrival Time Convenient: ")
Ease_of_online_booking = st.text_input("Ease of online booking: ")
Food_And_Drink = st.text_input("Food and Drink: ")
Online_Boarding = st.text_input("Online boarding: ")
Seat_Comfort = st.text_input("Seat comfort: ")
Inflight_Entertainment = st.text_input("Inflight Entertainment: ")
Onboard_service = st.text_input("Onboard service: ")
leg_room_service = st.text_input("Legroom service: ")
baggage_handling = st.text_input("Baggage Handling: ")
checkin_service = st.text_input("Checkin service: ")
inflight_service = st.text_input("Inflight service: ")
cleanliness = st.text_input("Cleanliness: ")
departure_delay = st.text_input("Departure Delay: ")
arrival_delay = st.text_input("Arrival Delay: ")
result=""
Shell
Python 3.10.9 (C:\Users\DELL\AppData\Local\Programs\Thonny\python.exe)
>>>
Local Python 3 • Thonny's Python
Type here to search
29°C
ENG
18:46
19-09-2024

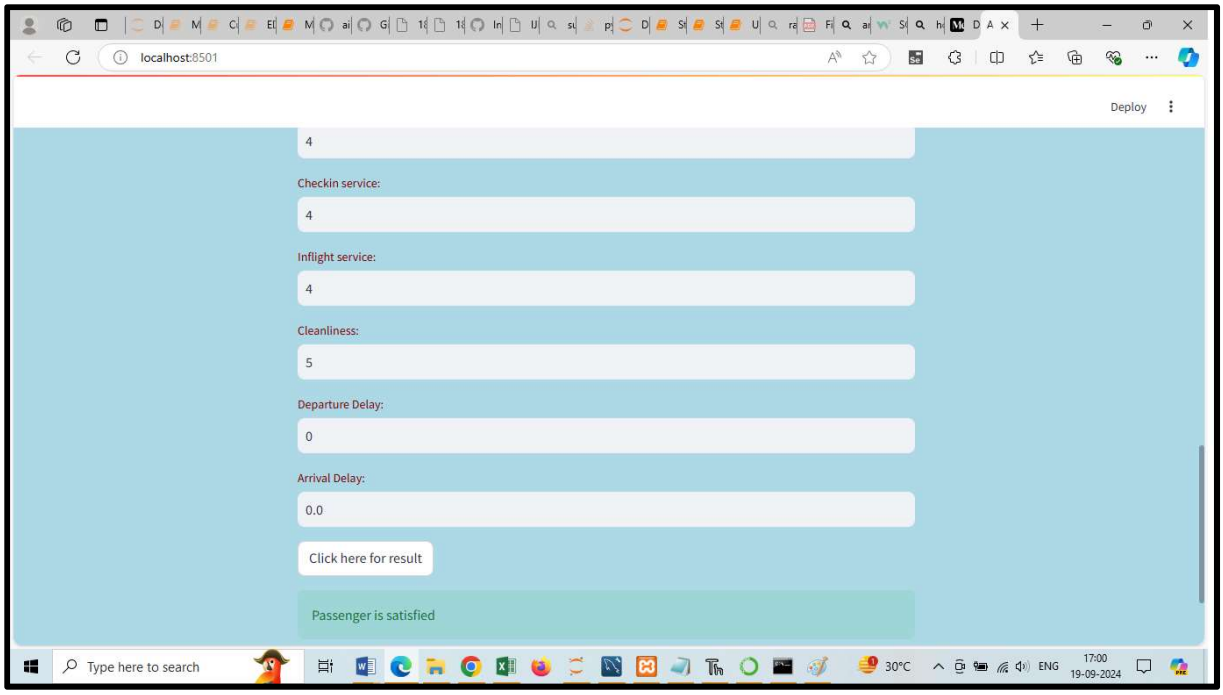


Output: When Passenger is Satisfied

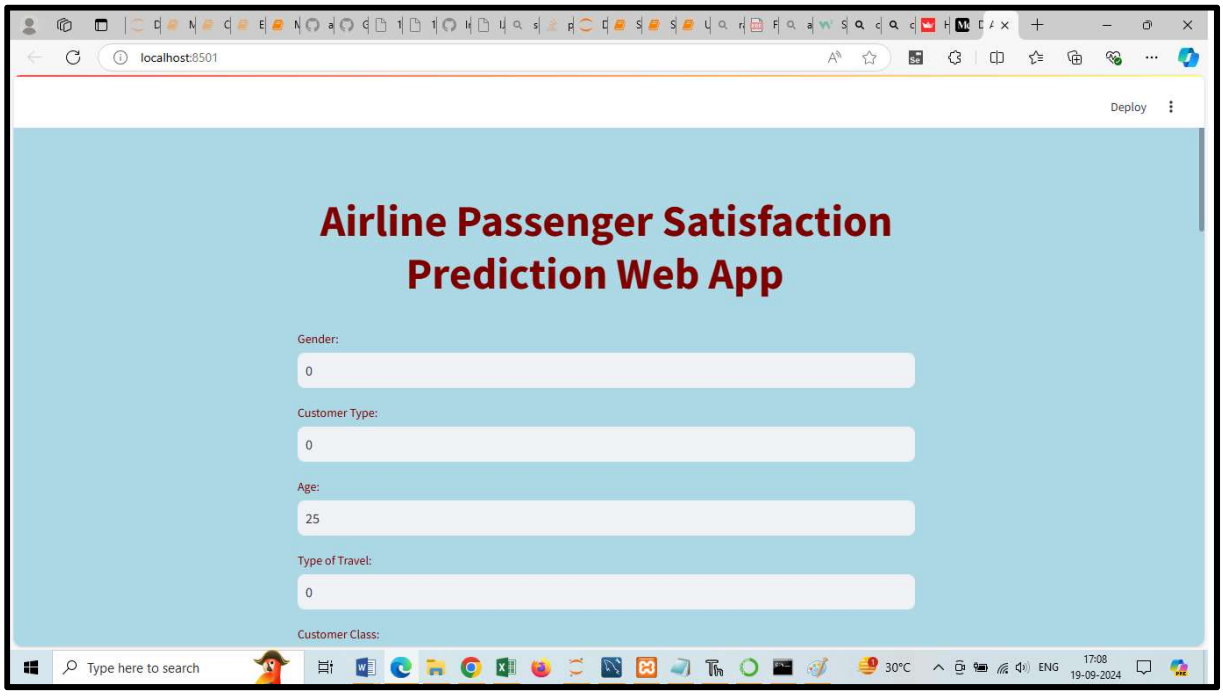


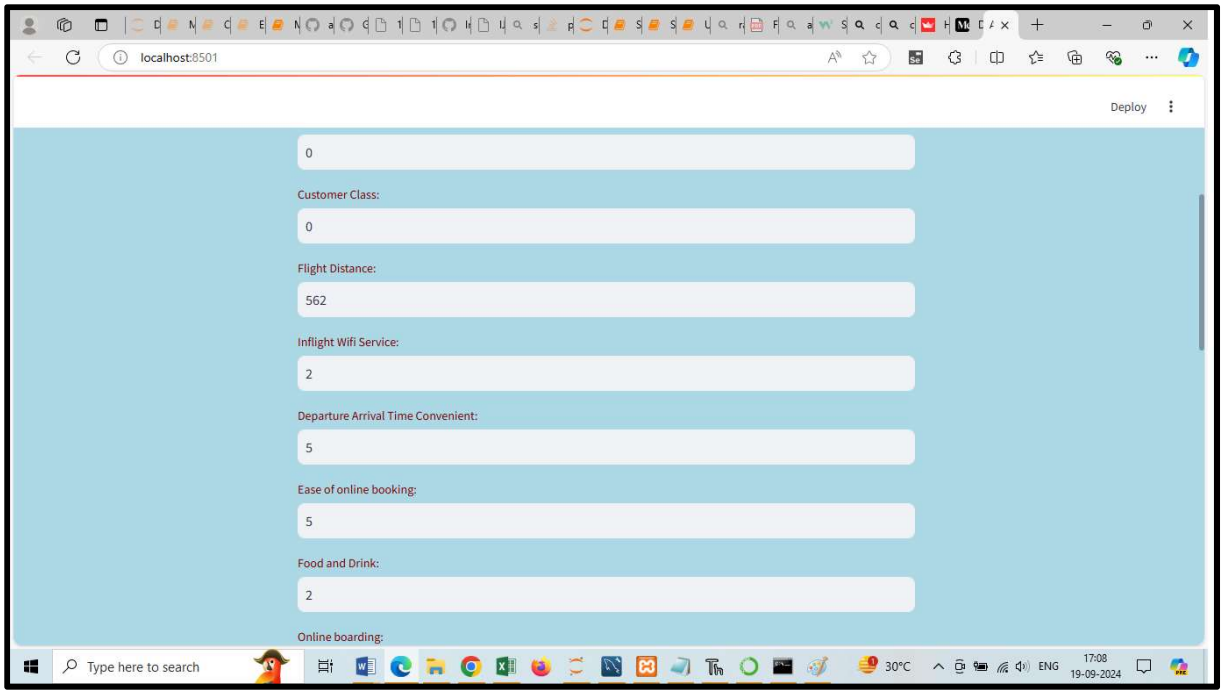

localhost:8501
Deploy
0
Customer Class:
0
Flight Distance:
1142
Inflight Wifi Service:
2
Departure Arrival Time Convenient:
2
Ease of online booking:
2
Food and Drink:
5
Online boarding:
Type here to search
30°C
ENG
17:00
19-09-2024

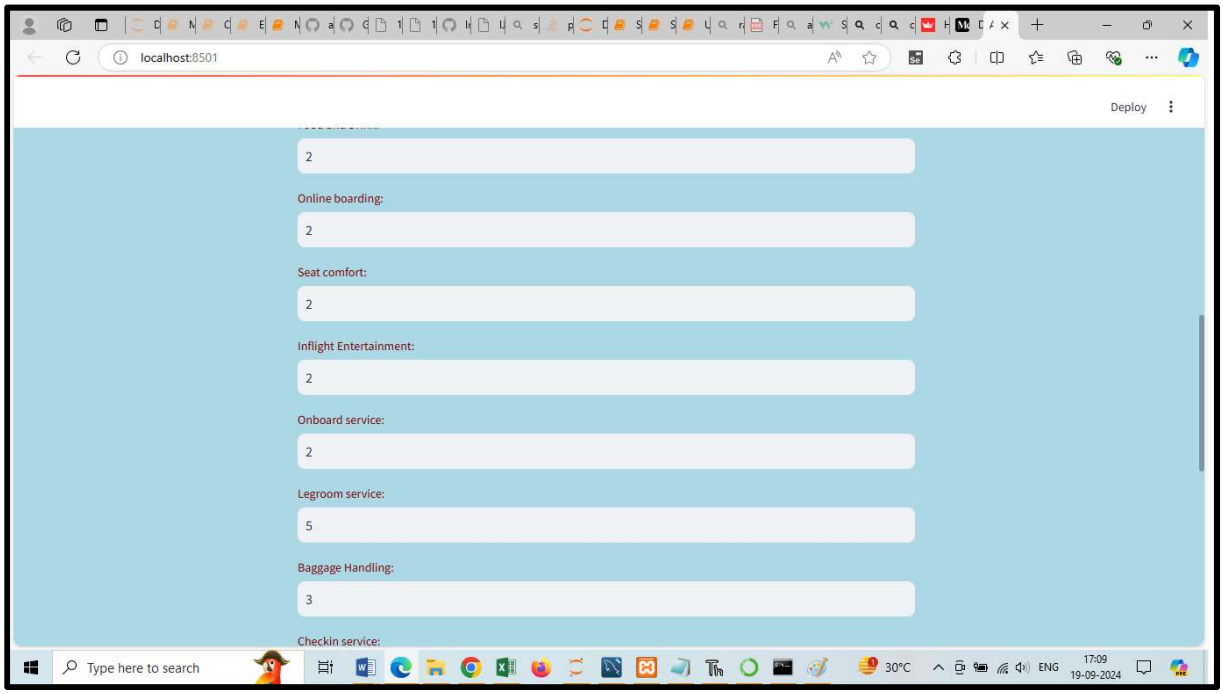

localhost:8501
Deploy
5
Online boarding:
5
Seat comfort:
5
Inflight Entertainment:
5
Onboard service:
4
Legroom service:
3
Baggage Handling:
4
Checkin service:
Type here to search
30°C
ENG
17:00
19-09-2024

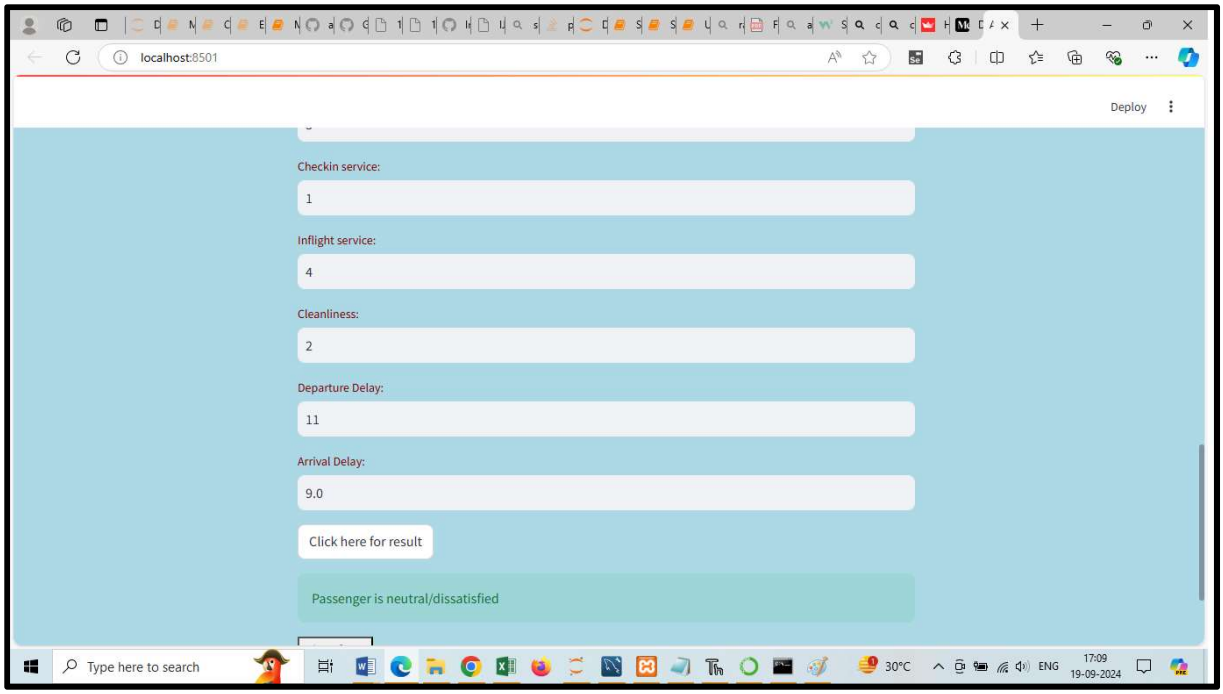


Output: When Passenger is dissatisfied/Neutral




localhost:8501
Deploy
0
Customer Class:
0
Flight Distance:
562
Inflight Wifi Service:
2
Departure Arrival Time Convenient:
5
Ease of online booking:
5
Food and Drink:
2
Online boarding:
Type here to search
30°C
ENG
17:08
19-09-2024


localhost:8501
Deploy
2
Online boarding:
2
Seat comfort:
2
Inflight Entertainment:
2
Onboard service:
2
Legroom service:
5
Baggage Handling:
3
Checkin service:
Type here to search
30°C
ENG
17:09
19-09-2024



5. Conclusion:

This project successfully demonstrated the application of supervised learning techniques. The use of the above machine learning algorithms helped predict the accuracy as well as outcomes of the data. Also building a web based application helps to keep user engaged with the system and make them use the trained model for prediction of their data.