

ASSINGNMENT-2

name:g.veda pranav,

roll:21bce8931

The screenshot displays a Jupyter Notebook interface with two visible code cells. The top cell shows the output of `df.DailyRate.value_counts()`, which is a Series of counts for the 'DailyRate' variable. The bottom cell shows the output of `df.info()`, which provides a summary of the DataFrame's structure, including the number of entries, data columns, and their respective data types.

```
[537] df.DailyRate.value_counts()
...
691 6
488 5
530 5
1329 5
1082 5
..
650 1
279 1
316 1
314 1
628 1
Name: DailyRate, Length: 886, dtype: int64

[538] df.info()
...
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
#   column              Non-Null Count  Dtype
---  -
0   Age                  1470 non-null    int64
1   Attrition            1470 non-null    object
2   BusinessTravel       1470 non-null    object
3   DailyRate            1470 non-null    int64
4   Department           1470 non-null    object
5   DistanceFromHome     1470 non-null    int64
6   Education            1470 non-null    int64
7   EducationField       1470 non-null    object
8   EmployeeCount        1470 non-null    int64
9   EmployeeNumber       1470 non-null    int64
10  EnvironmentSatisfaction 1470 non-null    int64
11  Gender               1470 non-null    object
```

```
[533] #Import the Libraries.
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import roc_curve, roc_auc_score
from sklearn.model_selection import train_test_split
from sklearn.datasets import make_classification
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import LabelBinarizer
from sklearn.multiclass import OneVsRestClassifier
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import RandomForestClassifier

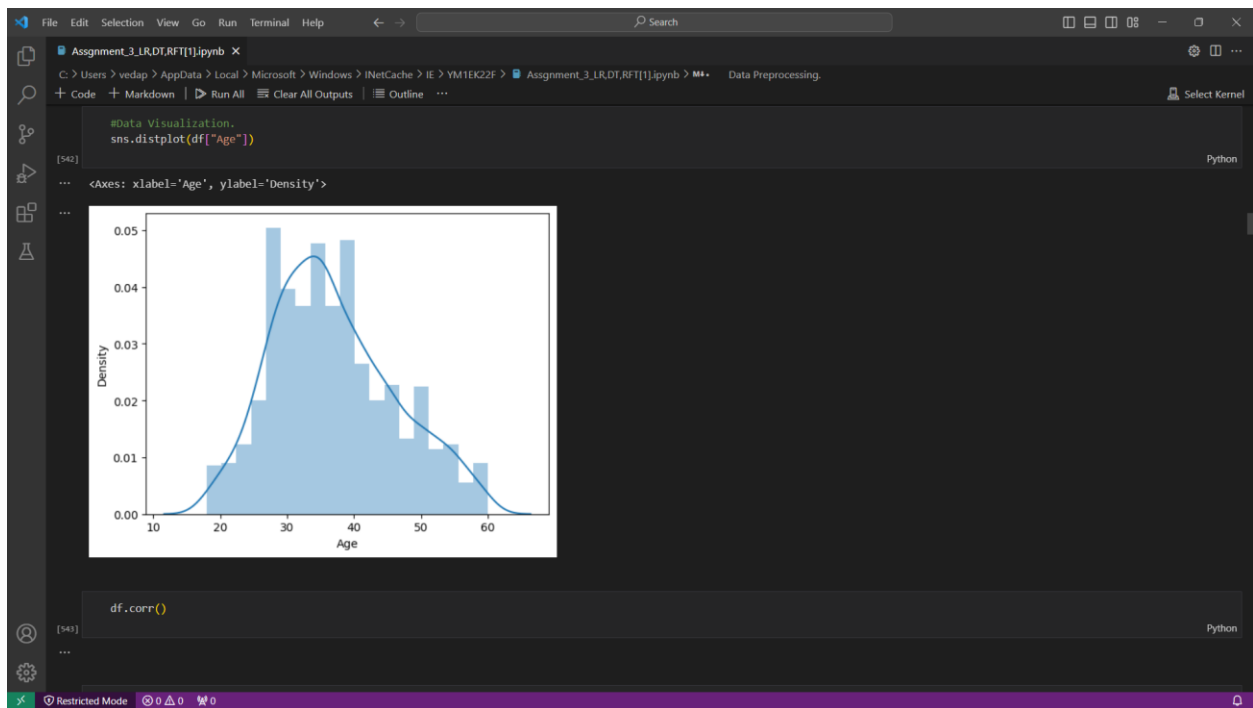
[534] #Importing the dataset.
df=pd.read_csv("/content/WA_Fn-UseC_HR-Employee-Attrition.csv")

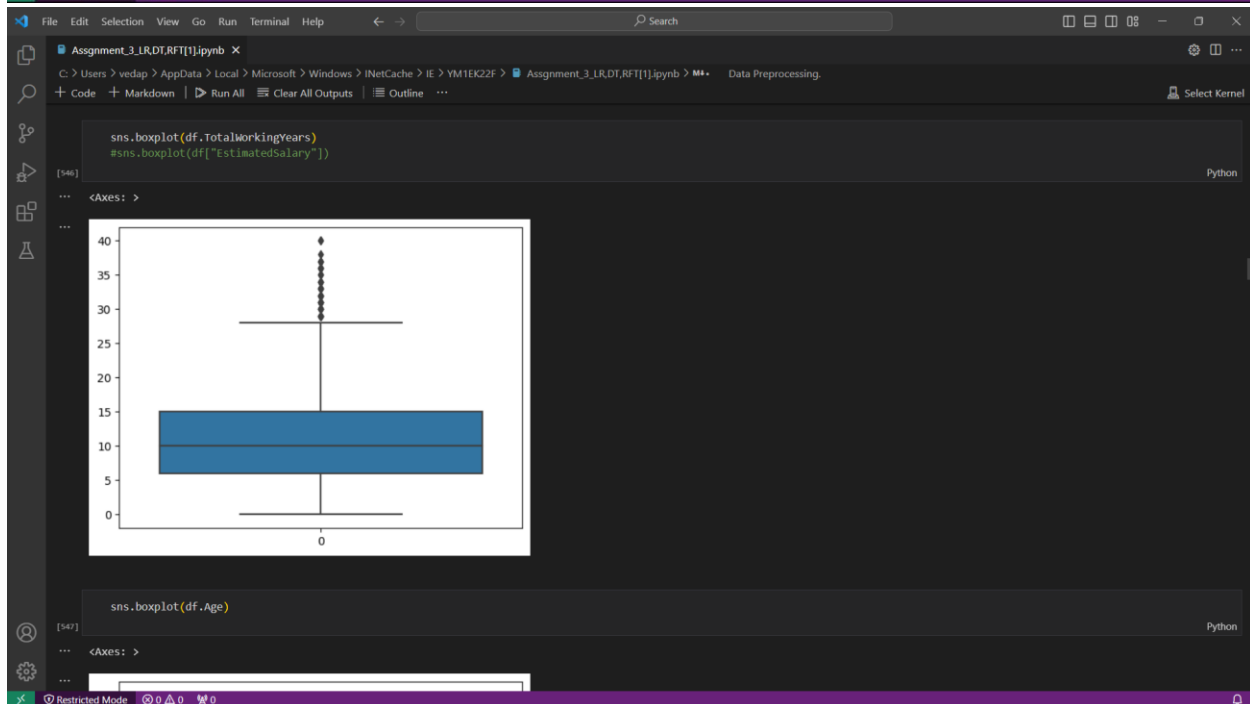
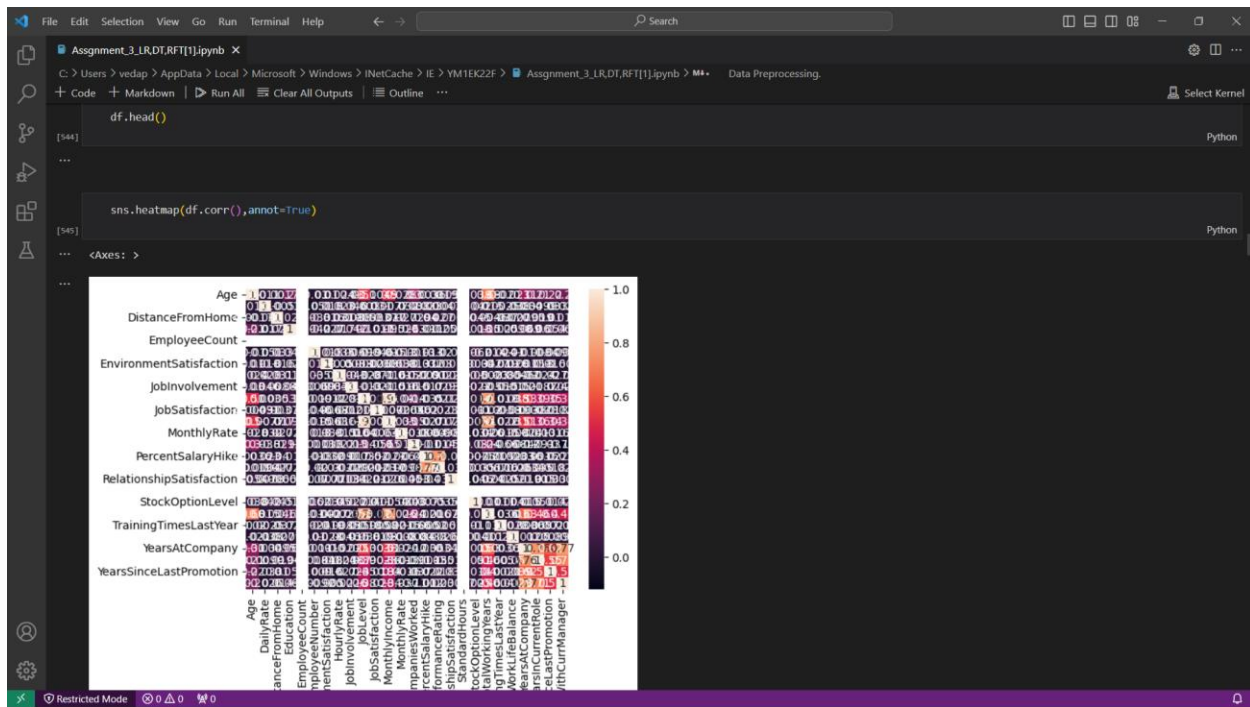
[535] df.head()
...

[536] df.shape
...
(1470, 35)

[537] df.DailyRate.value_counts()
...
691 6
488 5
530 5
1329 5
1082 5
..
650 1
279 1
316 1
314 1
628 1
Name: DailyRate, Length: 886, dtype: int64
```

```
File Edit Selection View Go Run Terminal Help
Assignment_3_LR,DT,RF[1].ipynb
C:\Users\vedap> AppData> Local> Microsoft> Windows> iNetCache> IE> YM1EK22F> Assignment_3_LR,DT,RF[1].ipynb> M... Data Preprocessing.
+ Code + Markdown | Run All | Clear All Outputs | Outline ...
df.describe()
[539] Python
...
#Checking for Null Values.
df.isnull().any()
[540] Python
...
Age                False
Attrition           False
BusinessTravel      False
DailyRate           False
Department          False
DistanceFromHome    False
Education            False
EducationField       False
EmployeeCount        False
EmployeeNumber       False
EnvironmentSatisfaction  False
Gender              False
HourlyRate           False
JobInvolvement       False
JobLevel            False
JobRole             False
JobSatisfaction      False
MaritalStatus        False
MonthlyIncome        False
MonthlyRate          False
NumCompaniesWorked   False
Over18              False
Overtime             False
PercentSalaryHike    False
PerformanceRating    False
...
YearsAtCompany       False
```





File Edit Selection View Go Run Terminal Help

Search

Assignment_3_LR,DT,RF[1].ipynb

C:\Users\vedap> AppData> Local> Microsoft> Windows> iNetCache> IE> YM1EK2ZF> Assignment_3_LR,DT,RF[1].ipynb> Data Preprocessing.

+ Code + Markdown Run All Clear All Outputs Outline

Select Kernel

[548]

df.head()

Python

...

[549]

#splitting Dependent and Independent variables
x=df.iloc[:,4]
x.head()

Python

...

[550]

y=df.DailyRate
y.head()

Python

...

[551]

#label encoding
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
x.BusinessTravel=le.fit_transform(x.BusinessTravel)
x.head()

Python

...

Restricted Mode 0 0 0

File Edit Selection View Go Run Terminal Help

Search

Assignment_3_LR,DT,RF[1].ipynb

C:\Users\vedap> AppData> Local> Microsoft> Windows> iNetCache> IE> YM1EK2ZF> Assignment_3_LR,DT,RF[1].ipynb> Data Preprocessing.

+ Code + Markdown Run All Clear All Outputs Outline

Select Kernel

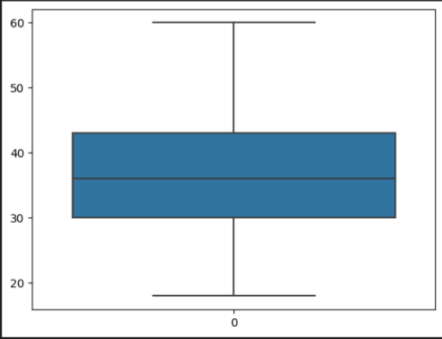
[547]

sns.boxplot(df.Age)

Python

...

<Axes: >



[548]

df.head()

Python

...

[549]

#splitting Dependent and Independent variables
x=df.iloc[:,4]

Python

...

Restricted Mode 0 0 0

```
File Edit Selection View Go Run Terminal Help
C:\Users\vedap > AppData > Local > Microsoft > Windows > iNetCache > IE > YM1EK22F > Assignment_3_LR,DT,RF[1].ipynb > M... Data Preprocessing.
+ Code + Markdown | Run All | Clear All Outputs | Outline ...
Select Kernel

#feature scaling
from sklearn.preprocessing import MinMaxScaler
ms=MinMaxScaler()
x_scaled=pd.DataFrame(ms.fit_transform(x),columns=x.columns)

[553] Python

x_scaled

[554] Python

...

#Splitting Data into Train and Test.
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x_scaled,y,test_size=0.2,random_state=0)

[555] Python

x_train.shape,x_test.shape,y_train.shape,y_test.shape

[556] Python

... ((1176, 4), (294, 4), (1176,), (294,))

x_train.head()

[557] Python

...

• Model Building
```

```
File Edit Selection View Go Run Terminal Help
C:\Users\vedap > AppData > Local > Microsoft > Windows > iNetCache > IE > YM1EK22F > Assignment_3_LR,DT,RF[1].ipynb > M... Data Preprocessing.
+ Code + Markdown | Run All | Clear All Outputs | Outline ...
Select Kernel

• Model Building

o Import the model building Libraries
o Initializing the model
o Training and testing the model
o Evaluation of Model
o Save the Model

from sklearn.tree import DecisionTreeClassifier
dtc=DecisionTreeClassifier()

[558] Python

dtc.fit(x_train,y_train)

[559] Python

...

pred=dtc.predict(x_test)

[560] Python

pred

[561] Python

... array([ 635, 575, 663, 1490, 458, 160, 791, 1283, 142, 439, 1376,
359, 995, 654, 1037, 1305, 618, 616, 326, 1107, 650, 824,
499, 781, 1442, 587, 720, 1282, 501, 818, 401, 1212, 1023,
598, 1457, 982, 720, 571, 669, 567, 194, 1239, 201, 119,
913, 238, 591, 771, 495, 813, 1214, 1361, 143, 1490, 592,
367, 688, 561, 1169, 381, 243, 1372, 408, 1476, 458, 939,
1084, 827, 933, 1102, 1179, 691, 310, 672, 1375, 883, 155,
```

```
File Edit Selection View Go Run Terminal Help
C:\Users\vedap> AppData> Local> Microsoft> Windows> iNetCache> IE> YM1EK22F> Assignment_3_LR,DT,RF[1].ipynb> Data Preprocessing.
+ Code + Markdown | Run All | Clear All Outputs | Outline ...
y_test
[562] Python
...
442 635
1091 575
981 662
785 1492
1332 459
...
1439 557
481 254
124 249
198 1261
1229 369
Name: DailyRate, Length: 294, dtype: int64

df
[563] Python
...

dtc.predict(ms.transform([[1,19,1900,1900]]))
[564] Python
...
array([1499])

Evaluation of classification model

#Accuracy score
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report,roc_auc_score,roc_curve
```

```
File Edit Selection View Go Run Terminal Help
C:\Users\vedap> AppData> Local> Microsoft> Windows> iNetCache> IE> YM1EK22F> Assignment_3_LR,DT,RF[1].ipynb> Data Preprocessing.
+ Code + Markdown | Run All | Clear All Outputs | Outline ...
Evaluation of classification model

#Accuracy score
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report,roc_auc_score,roc_curve
[565] Python

accuracy_score(y_test,pred)
[566] Python
...
0.3435374149659864

confusion_matrix(y_test,pred)
[567] Python
...
array([[0, 0, 1, ..., 0, 0, 0],
       [0, 0, 1, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 1, 0, 0],
       [0, 0, 0, ..., 0, 0, 1],
       [0, 0, 0, ..., 0, 0, 0]])

pd.crosstab(y_test,pred)
[568] Python
...

predicted no    predicted yes
Actual No 58=TN 0=FP Actual yes 6=FN 16=TP
```

```
File Edit Selection View Go Run Terminal Help
C: > Users > vedap > AppData > Local > Microsoft > Windows > iNetCache > IE > YM1EK22F > Assignment_3_LR,DT,RF[1].ipynb > Data Preprocessing.
+ Code + Markdown | Run All | Clear All Outputs | Outline ...
pd.crosstab(y_test,pred)

...

predicted no    predicted yes

Actual No 58=TN 0=FP Actual yes 6=FN 16=TP

(58+16)/80 #accuracy

[568]
... 0.925

print(classification_report(y_test,pred))

[570]
...
      precision    recall  f1-score   support

102      0.00      0.00      0.00         1
103      0.00      0.00      0.00         1
104      0.00      0.00      0.00         0
106      0.00      0.00      0.00         1
109      0.00      0.00      0.00         0
111      1.00      1.00      1.00         1
119      1.00      1.00      1.00         1
140      0.00      0.00      0.00         0
141      0.00      0.00      0.00         1
142      0.00      0.00      0.00         1
143      0.00      0.00      0.00         0
145      0.00      0.00      0.00         1
147      1.00      1.00      1.00         1
155      1.00      1.00      1.00         1
160      1.00      1.00      1.00         1

Restricted Mode
```

```
File Edit Selection View Go Run Terminal Help
C: > Users > vedap > AppData > Local > Microsoft > Windows > iNetCache > IE > YM1EK22F > Assignment_3_LR,DT,RF[1].ipynb > Data Preprocessing.
+ Code + Markdown | Run All | Clear All Outputs | Outline ...
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

# precision
# of all positive predictions how many are really positive
#when it predicts yes,how often it is correct

# precision = TP/(TP+FP)
#16/8+16

[571]

# Recall
# of all real positive cases how many are predicted positive
#when it is acutally is yes ,how often does it predict yes

# Recall = TP/(FN+TP)
#16/16+6

[572]
... 7.0

# F1 score

# 2*precision*Recall/(Precision+Recall)

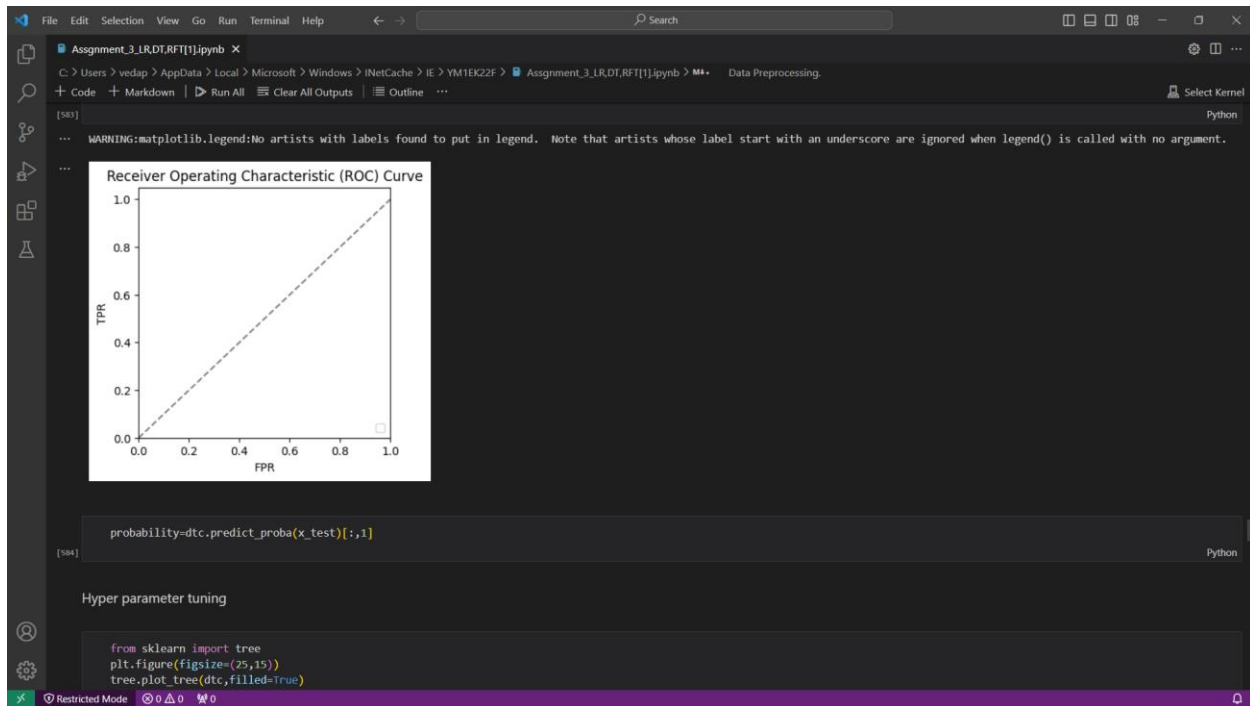
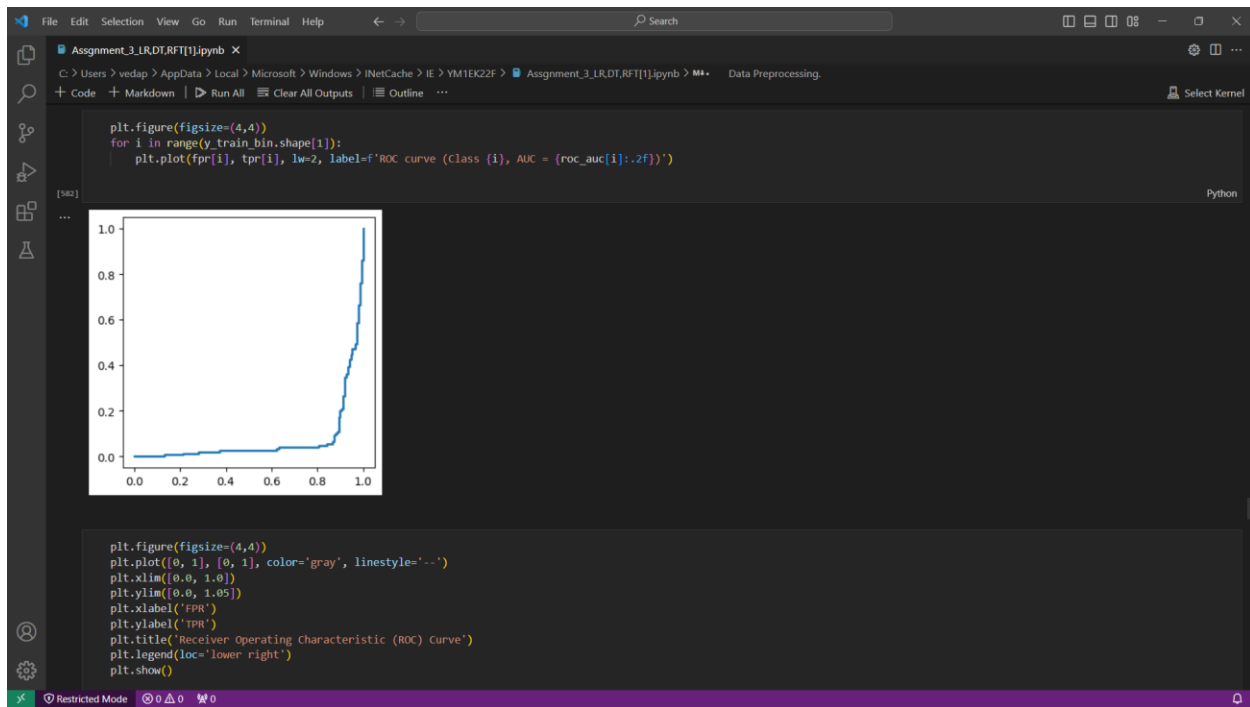
[573]

Roc-AUC curve

Restricted Mode
```

```
File Edit Selection View Go Run Terminal Help
C:\Users\vedap > AppData > Local > Microsoft > Windows > iNetCache > IE > YM1EK22F > Assignment_3_LR,DT,RF[1].ipynb > Data Preprocessing.
+ Code + Markdown | Run All | Clear All Outputs | Outline
Roc-AUC curve
image.png
probability=dtc.predict_proba(x_test)
probability
array([[0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]])
y_test
442 635
1091 575
981 662
785 1492
1332 459
...
1439 557
481 254
124 249
198 1261
1229 369
```

```
File Edit Selection View Go Run Terminal Help
C:\Users\vedap > AppData > Local > Microsoft > Windows > iNetCache > IE > YM1EK22F > Assignment_3_LR,DT,RF[1].ipynb > Data Preprocessing.
+ Code + Markdown | Run All | Clear All Outputs | Outline
x, y = make_classification(n_samples=1500, n_features=50, n_classes=2, random_state=42)
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
y_train_bin = label_binarize(y_train, classes=np.unique(y))
y_test_bin = label_binarize(y_test, classes=np.unique(y))
classifier = OneVsRestClassifier(LogisticRegression())
classifier.fit(X_train, y_train_bin)
y_probs = classifier.predict_proba(x_test)
fpr = dict()
tpr = dict()
roc_auc = dict()
for i in range(y_train_bin.shape[1]):
    fpr[i], tpr[i], _ = roc_curve(y_test_bin[:, i], y_probs[:, i])
    roc_auc[i] = roc_auc_score(y_test_bin[:, i], y_probs[:, i])
plt.figure(figsize=(4,4))
```


```
File Edit Selection View Go Run Terminal Help
Assignment_3_LR,DT,RF[1].ipynb X
C:\Users\vedap> AppData> Local> Microsoft> Windows> iNetCache> IE> YM1EK22F> Assignment_3_LR,DT,RF[1].ipynb> M... Data Preprocessing.
+ Code + Markdown | Run All | Clear All Outputs | Outline ...
Select Kernel

[586] from sklearn.model_selection import GridSearchCV
from sklearn.tree import DecisionTreeClassifier
Python

[587] # Define the decision tree classifier
dt_classifier = DecisionTreeClassifier(random_state=42)
Python

[588] # Define the parameter grid
parameter_grid = {
    'criterion': ['gini', 'entropy'],
    'splitter': ['best', 'random'],
    'max_depth': [None, 10, 20, 30, 40],
    'max_features': ['auto', 'sqrt', 'log2']
}
Python
+ Code + Markdown

[589] # Create GridSearchCV object
grid_search = GridSearchCV(dt_classifier, param_grid=parameter_grid, cv=10, scoring='accuracy')
Python

[590] # Fit the GridSearchCV to your training data (X_train, y_train)
grid_search.fit(X_train, y_train)
Python

...

[591] # Get the best estimator and its hyperparameters
```

```
File Edit Selection View Go Run Terminal Help
Assignment_3_LR,DT,RF[1].ipynb X
C:\Users\vedap> AppData> Local> Microsoft> Windows> iNetCache> IE> YM1EK22F> Assignment_3_LR,DT,RF[1].ipynb> M... Data Preprocessing.
+ Code + Markdown | Run All | Clear All Outputs | Outline ...
Select Kernel

[591] # Get the best estimator and its hyperparameters
best_dt_classifier = grid_search.best_estimator_
best_params = grid_search.best_params_
Python

[592] # Make predictions on the test set using the best estimator
pred = best_dt_classifier.predict(X_test)
Python

[593] # Evaluate the best classifier on the test data
accuracy = accuracy_score(y_test, pred)
Python

[594] best_params
Python
...
{'criterion': 'gini',
 'max_depth': 10,
 'max_features': 'log2',
 'splitter': 'best'}

[595] accuracy
Python
...
0.8966666666666666
```

```
File Edit Selection View Go Run Terminal Help
C:\Users\vedap > AppData > Local > Microsoft > Windows > iNetCache > IE > YM1EK22F > Assignment_3_LR,DT,RF[1].ipynb > M... Data Preprocessing.
+ Code + Markdown | Run All | Clear All Outputs | Outline ...
Random Forest

from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split

[596] Python

# Define the RandomForestClassifier and parameter grid
rfc = RandomForestClassifier(random_state=42)
param_grid = {
    'n_estimators': [10, 50, 100],
    'max_depth': [None, 10, 20],
}

[597] Python

# Create GridSearchCV object
rfc_cv = GridSearchCV(rfc, param_grid, cv=10, scoring="accuracy")

[598] Python

# Fit the GridSearchCV to the training data
rfc_cv.fit(X_train, y_train)

[599] Python
...
Restricted Mode 0 0 0 0
```

```
File Edit Selection View Go Run Terminal Help
C:\Users\vedap > AppData > Local > Microsoft > Windows > iNetCache > IE > YM1EK22F > Assignment_3_LR,DT,RF[1].ipynb > M... Data Preprocessing.
+ Code + Markdown | Run All | Clear All Outputs | Outline ...
# Get the best estimator and its hyperparameters
best_classifier = rfc_cv.best_estimator_
best_params = rfc_cv.best_params_

[600] Python

# Make predictions on the test set using the best estimator
pred = best_classifier.predict(X_test)

[601] Python

# Evaluate the best classifier on the test data
accuracy = accuracy_score(y_test, pred)

[602] Python

best_params

[603] Python
... {'max_depth': 10, 'n_estimators': 100}

accuracy

[604] Python
... 0.9233333333333333
Restricted Mode 0 0 0 0
```