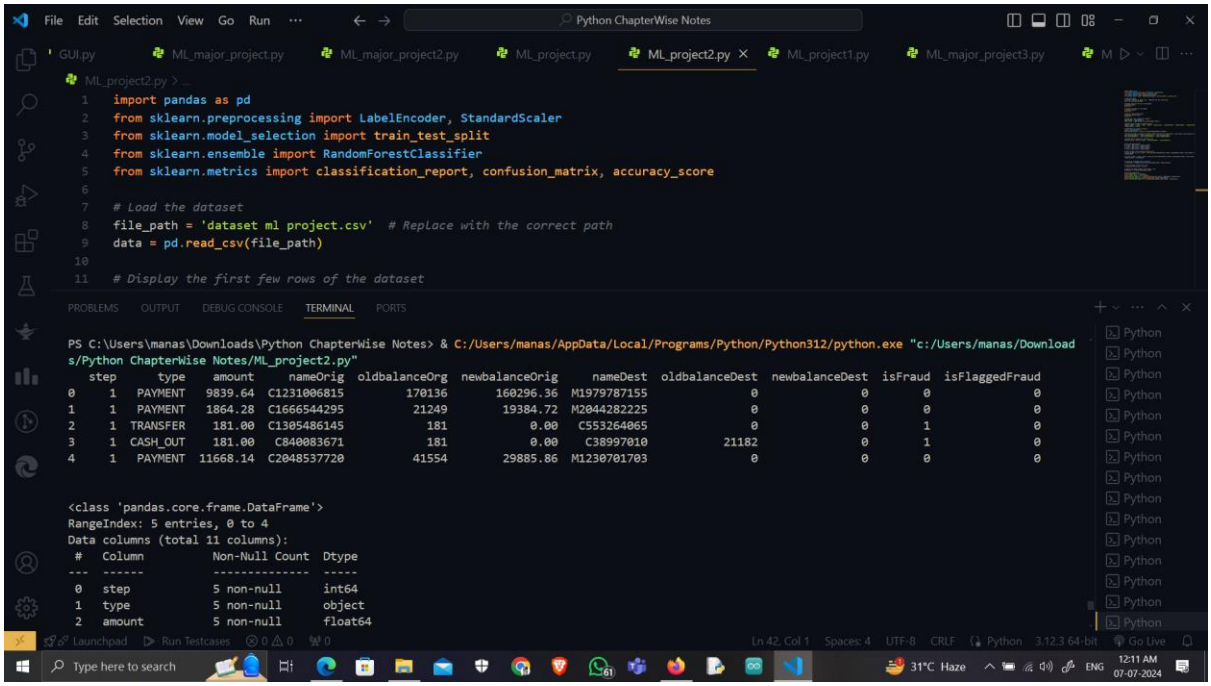
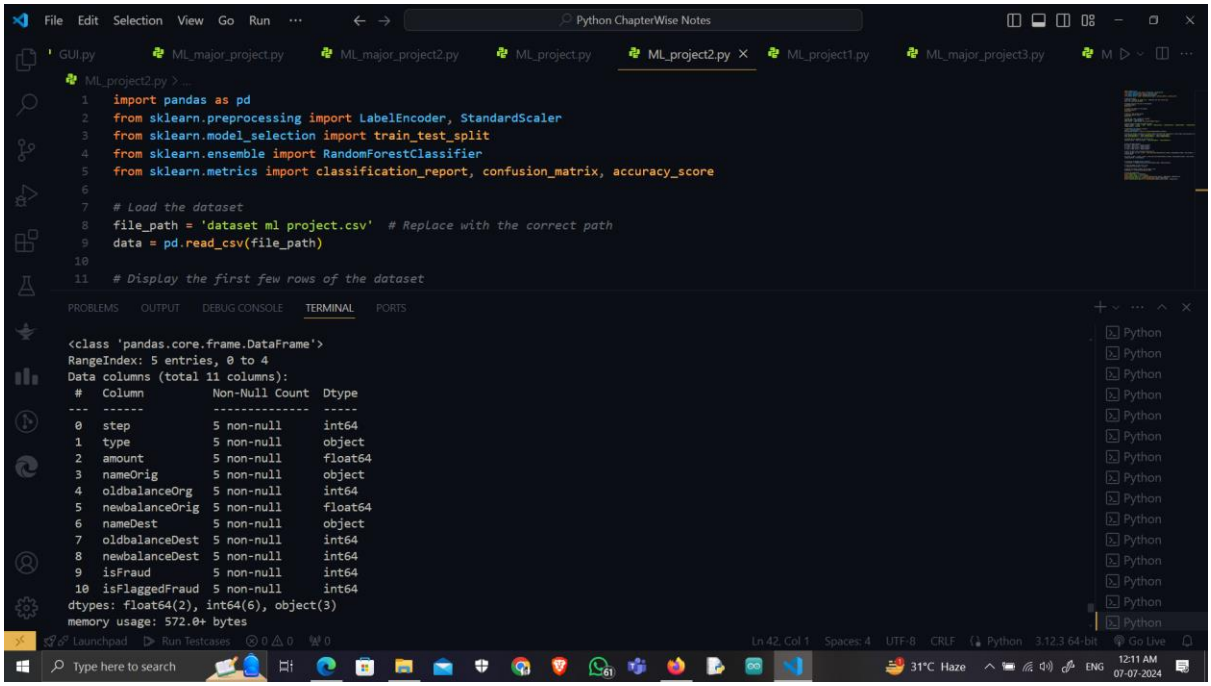
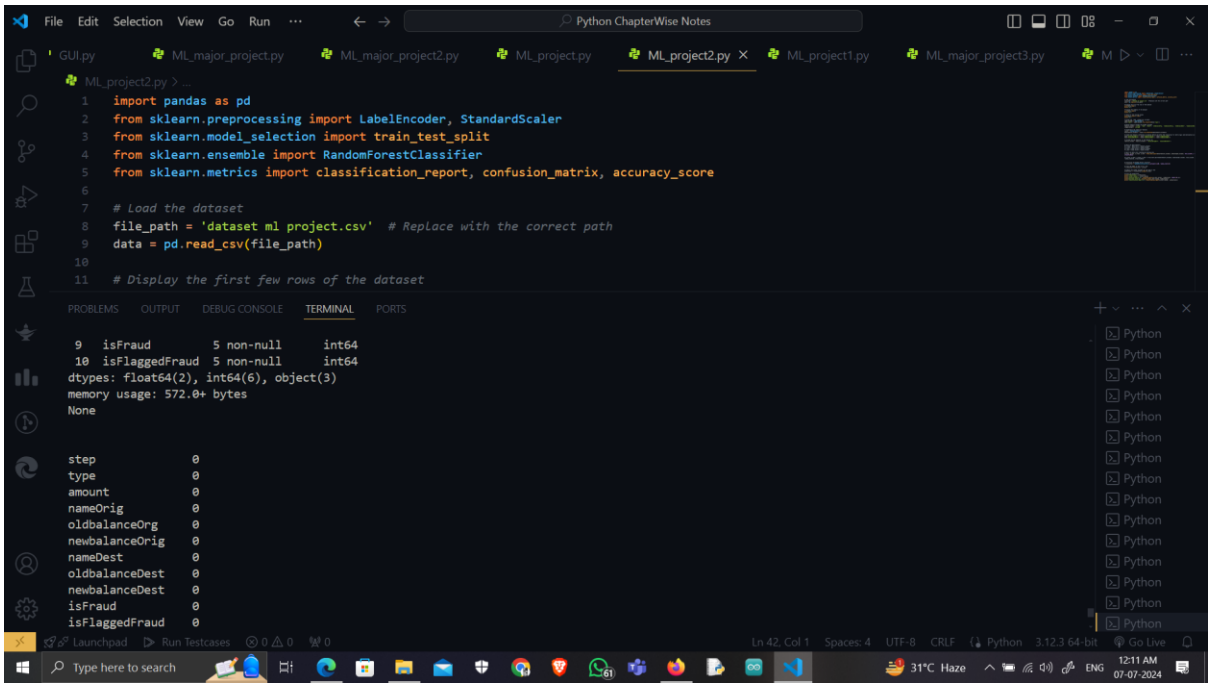
```
import pandas as pd
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score

# Load the dataset
file_path = 'dataset ml project.csv'  # Replace with the correct path but here
only name of the file is written because the file location is same as the code
location
data = pd.read_csv(file_path)

# Display the first few rows of the dataset
print(data.head())
print("\n")

# Display the summary of the dataset
print(data.info())
print("\n")

# Check for any missing values
print(data.isnull().sum())
print("\n")

# Encode the 'type' categorical variable
label_encoder = LabelEncoder()
data['type'] = label_encoder.fit_transform(data['type'])

# Define feature columns and target variable
feature_columns = ['step', 'type', 'amount', 'oldbalanceOrg',
'newbalanceOrig', 'oldbalanceDest', 'newbalanceDest']
target_column = 'isFraud'

# Standardize the numerical features
scaler = StandardScaler()
data[feature_columns] = scaler.fit_transform(data[feature_columns])

# Create new feature: difference in balance before and after the transaction
for both origin and destination accounts
data['balanceOrigDiff'] = data['newbalanceOrig'] - data['oldbalanceOrg']
data['balanceDestDiff'] = data['newbalanceDest'] - data['oldbalanceDest']

# Include the new features in the feature set
feature_columns = feature_columns + ['balanceOrigDiff', 'balanceDestDiff']


# Split the data manually
# X_train = data.loc[:2, feature_columns]
```

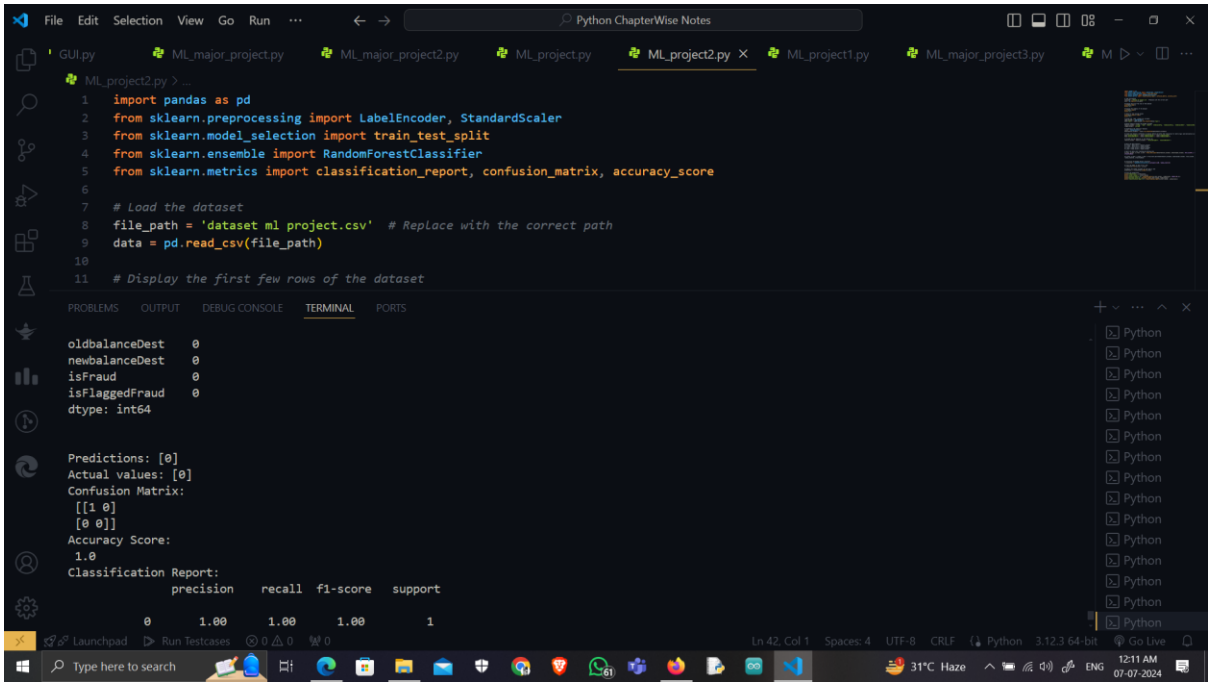
```
# y_train = data.loc[:2, target_column]
# X_test = data.loc[3:4, feature_columns]
# y_test = data.loc[3:4, target_column]

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(data[feature_columns],
data[target_column], test_size=0.2, random_state=42, stratify=None)

# X_train, X_test, y_train, y_test = train_test_split(data[feature_columns],
data[target_column], train_size=4, test_size=1, random_state=42,
stratify=None)


# Initialize the Random Forest classifier
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)

# Train the model on the first 3 rows
rf_classifier.fit(X_train, y_train)

# Predict the target variable for the next 2 rows
predictions = rf_classifier.predict(X_test)

# Print the predictions
print("Predictions:", predictions)
print("Actual values:", y_test.values)
print("Confusion Matrix:\n", confusion_matrix(y_test.values,  predictions,
labels=[0,1]))
print("Accuracy Score:\n", accuracy_score(y_test.values, predictions))
print("Classification Report:\n",
classification_report(y_test.values,  predictions))
```

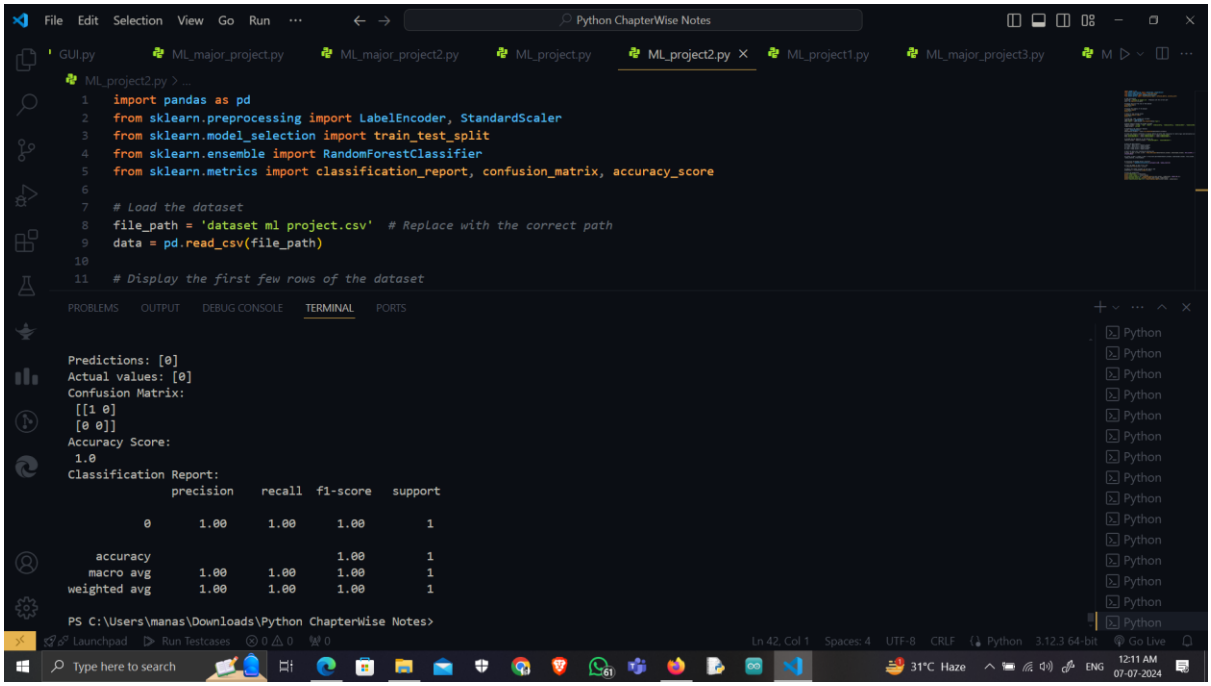
SCREENSHOTS OF OUTPUTS:


File Edit Selection View Go Run
Python ChapterWise Notes
GUI.py
ML_major_project.py
ML_major_project2.py
ML_project.py
ML_project2.py
ML_project1.py
ML_major_project3.py
ML_project2.py > ...
1 import pandas as pd
2 from sklearn.preprocessing import LabelEncoder, StandardScaler
3 from sklearn.model_selection import train_test_split
4 from sklearn.ensemble import RandomForestClassifier
5 from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
6
7 # Load the dataset
8 file_path = 'dataset ml project.csv' # Replace with the correct path
9 data = pd.read_csv(file_path)
10
11 # Display the first few rows of the dataset
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\manas\Downloads\Python ChapterWise Notes> & C:/Users/manas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/manas/Downloads/Python ChapterWise Notes/ML_project2.py"
step type amount nameOrig oldbalanceOrg newbalanceOrig nameDest oldbalanceDest newbalanceDest isFraud isFlaggedFraud
0 1 PAYMENT 9839.64 C1231006815 170136 160296.36 M1979787155 0 0 0 0
1 1 PAYMENT 1864.28 C1666544295 21249 19384.72 M2044282225 0 0 0 0
2 1 TRANSFER 181.00 C1305486145 181 0.00 C553264065 0 0 1 0
3 1 CASH_OUT 181.00 C840083671 181 0.00 C38997010 21182 0 1 0
4 1 PAYMENT 11668.14 C2048537720 41554 29885.86 M1230701703 0 0 0 0
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 11 columns):
Column Non-Null Count Dtype
--- ------ -------------- -----
0 step 5 non-null int64
1 type 5 non-null object
2 amount 5 non-null float64
Launchpad Run Testcases
Ln 42, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.3 64-bit Go Live
Type here to search
31°C Haze
ENG
12:11 AM
07-07-2024

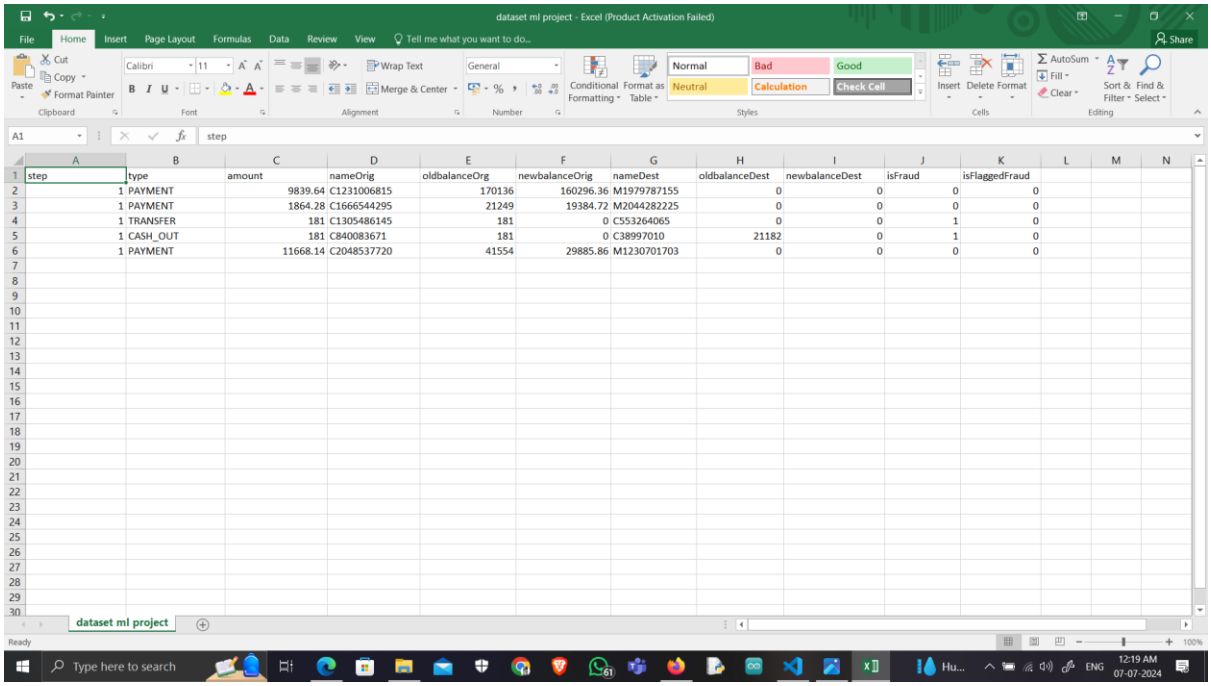

File Edit Selection View Go Run
Python ChapterWise Notes
GUI.py
ML_major_project.py
ML_major_project2.py
ML_project.py
ML_project2.py
ML_project1.py
ML_major_project3.py
ML_project2.py > ...
1 import pandas as pd
2 from sklearn.preprocessing import LabelEncoder, StandardScaler
3 from sklearn.model_selection import train_test_split
4 from sklearn.ensemble import RandomForestClassifier
5 from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
6
7 # Load the dataset
8 file_path = 'dataset ml project.csv' # Replace with the correct path
9 data = pd.read_csv(file_path)
10
11 # Display the first few rows of the dataset
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 11 columns):
Column Non-Null Count Dtype
--- ------ -------------- -----
0 step 5 non-null int64
1 type 5 non-null object
2 amount 5 non-null float64
3 nameOrig 5 non-null object
4 oldbalanceOrg 5 non-null int64
5 newbalanceOrig 5 non-null float64
6 nameDest 5 non-null object
7 oldbalanceDest 5 non-null int64
8 newbalanceDest 5 non-null int64
9 isFraud 5 non-null int64
10 isFlaggedFraud 5 non-null int64
dtypes: float64(2), int64(6), object(3)
memory usage: 572.0+ bytes
Launchpad Run Testcases
Ln 42, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.3 64-bit Go Live
Type here to search
31°C Haze
ENG
12:11 AM
07-07-2024


File Edit Selection View Go Run
Python ChapterWise Notes
GUI.py
ML_major_project.py
ML_major_project2.py
ML_project.py
ML_project2.py
ML_project1.py
ML_major_project3.py
ML_project2.py > ...
1 import pandas as pd
2 from sklearn.preprocessing import LabelEncoder, StandardScaler
3 from sklearn.model_selection import train_test_split
4 from sklearn.ensemble import RandomForestClassifier
5 from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
6
7 # Load the dataset
8 file_path = 'dataset ml project.csv' # Replace with the correct path
9 data = pd.read_csv(file_path)
10
11 # Display the first few rows of the dataset
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
9 isFraud 5 non-null int64
10 isFlaggedFraud 5 non-null int64
dtypes: float64(2), int64(6), object(3)
memory usage: 572.0+ bytes
None
step 0
type 0
amount 0
nameOrig 0
oldbalanceOrg 0
newbalanceOrig 0
nameDest 0
oldbalanceDest 0
newbalanceDest 0
isFraud 0
isFlaggedFraud 0
Launchpad Run Testcases
Ln 42, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.3 64-bit Go Live
Type here to search
31°C Haze
ENG
12:11 AM
07-07-2024

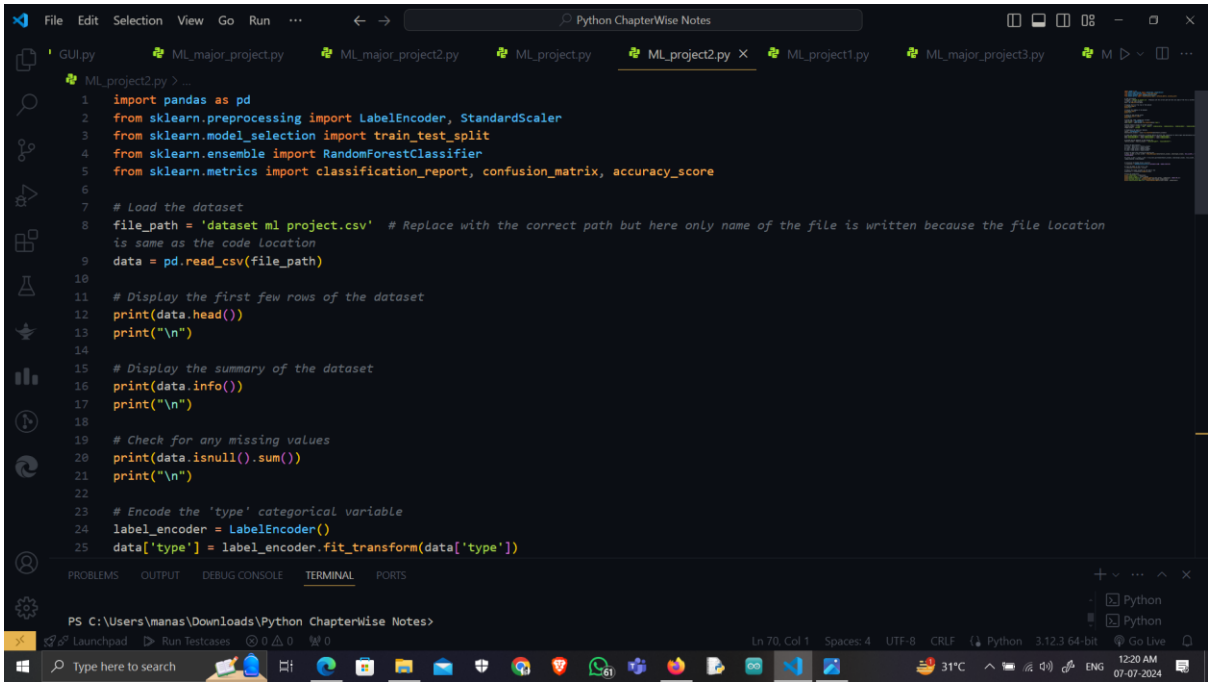

File Edit Selection View Go Run
Python ChapterWise Notes
GUI.py
ML_major_project.py
ML_major_project2.py
ML_project.py
ML_project2.py
ML_project1.py
ML_major_project3.py
ML_project2.py > ...
1 import pandas as pd
2 from sklearn.preprocessing import LabelEncoder, StandardScaler
3 from sklearn.model_selection import train_test_split
4 from sklearn.ensemble import RandomForestClassifier
5 from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
6
7 # Load the dataset
8 file_path = 'dataset ml project.csv' # Replace with the correct path
9 data = pd.read_csv(file_path)
10
11 # Display the first few rows of the dataset
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
oldbalanceDest 0
newbalanceDest 0
isFraud 0
isFlaggedFraud 0
dtype: int64
Predictions: [0]
Actual values: [0]
Confusion Matrix:
[[1 0]
[0 0]]
Accuracy Score:
1.0
Classification Report:
precision recall f1-score support
0 1.00 1.00 1.00 1
Launchpad Run Testcases
Ln 42, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.3 64-bit Go Live
Type here to search
31°C Haze
ENG
12:11 AM
07-07-2024



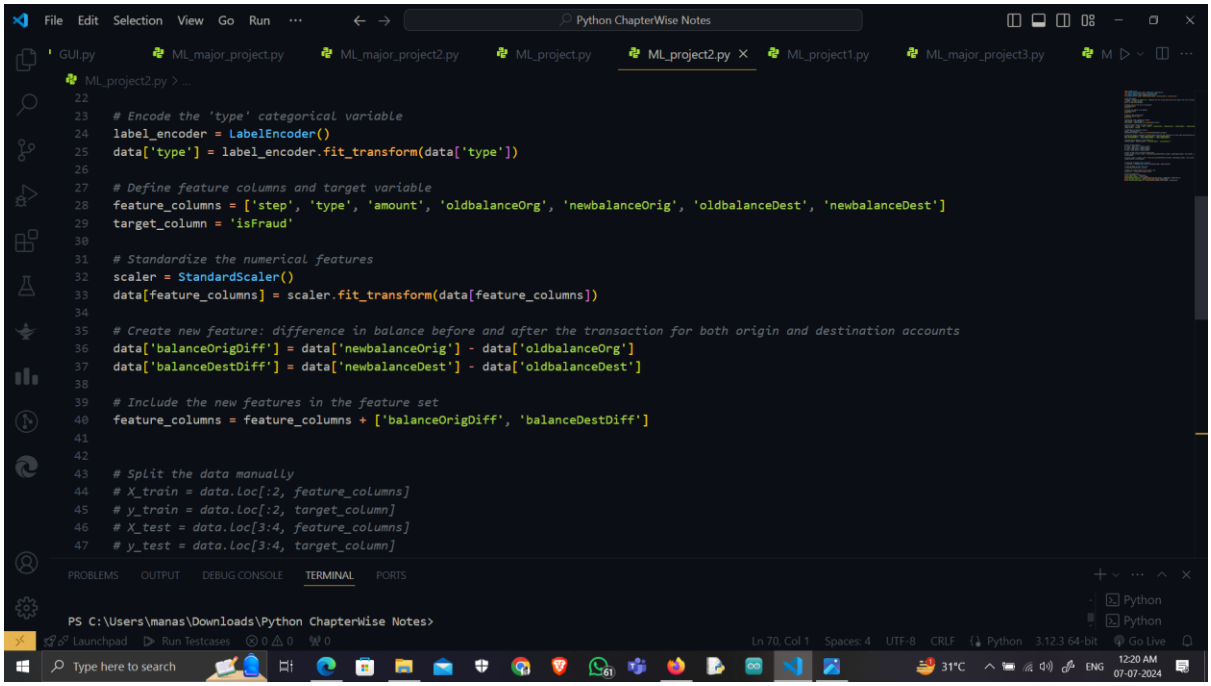
SCREENSHOTS OF DATASET:



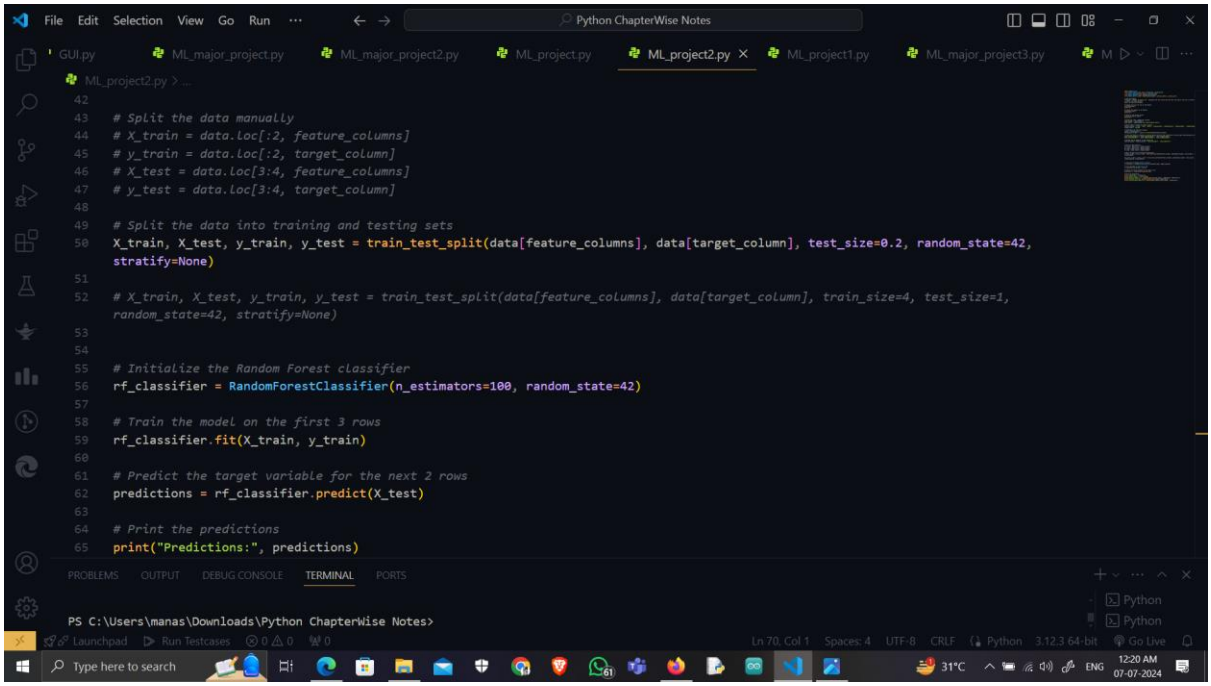
SCREENSHOTS OF CODE:

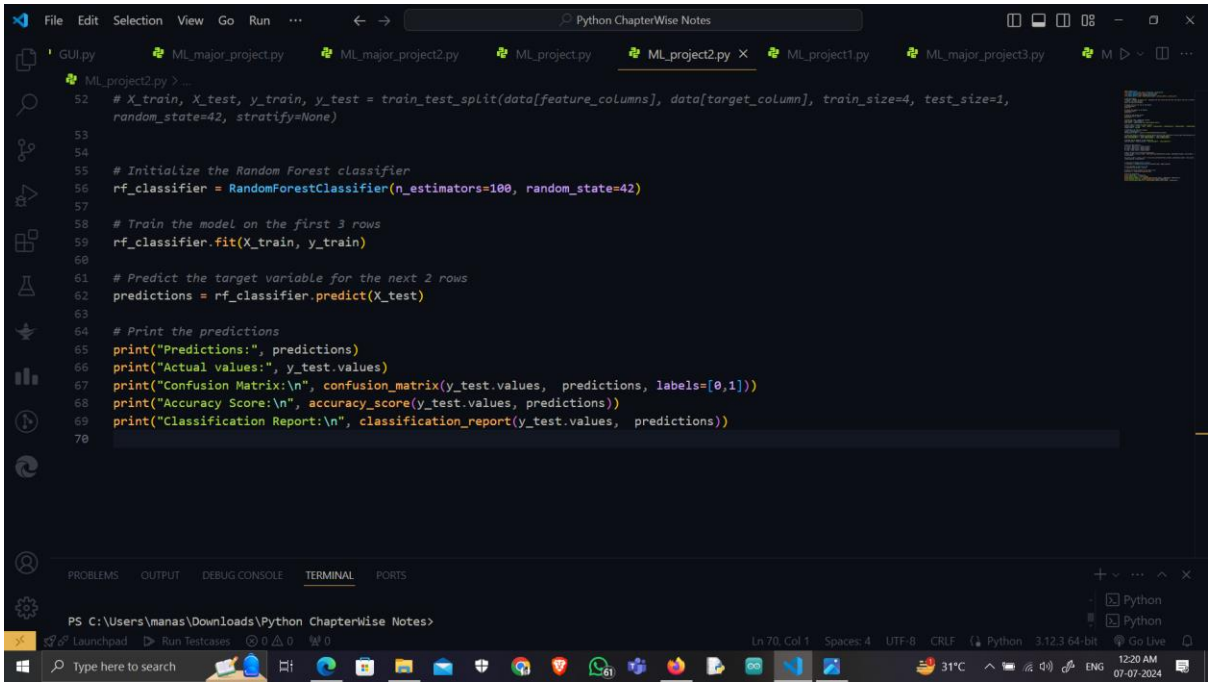


```python
import pandas as pd
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

# Load the dataset
file_path = 'dataset ml project.csv'  # Replace with the correct path but here only name of the file is written because the file location is same as the code location
data = pd.read_csv(file_path)

# Display the first few rows of the dataset
print(data.head())
print("\n")

# Display the summary of the dataset
print(data.info())
print("\n")

# Check for any missing values
print(data.isnull().sum())
print("\n")

# Encode the 'type' categorical variable
label_encoder = LabelEncoder()
data['type'] = label_encoder.fit_transform(data['type'])
```



```python
# Encode the 'type' categorical variable
label_encoder = LabelEncoder()
data['type'] = label_encoder.fit_transform(data['type'])

# Define feature columns and target variable
feature_columns = ['step', 'type', 'amount', 'oldbalanceOrg', 'newbalanceOrig', 'oldbalanceDest', 'newbalanceDest']
target_column = 'isFraud'

# Standardize the numerical features
scaler = StandardScaler()
data[feature_columns] = scaler.fit_transform(data[feature_columns])

# Create new feature: difference in balance before and after the transaction for both origin and destination accounts
data['balanceOrigDiff'] = data['newbalanceOrig'] - data['oldbalanceOrg']
data['balanceDestDiff'] = data['newbalanceDest'] - data['oldbalanceDest']

# Include the new features in the feature set
feature_columns = feature_columns + ['balanceOrigDiff', 'balanceDestDiff']


# Split the data manually
# X_train = data.loc[:2, feature_columns]
# y_train = data.loc[:2, target_column]
# X_test = data.loc[3:4, feature_columns]
# y_test = data.loc[3:4, target_column]
```


File Edit Selection View Go Run
Python ChapterWise Notes
GUI.py
ML_major_project.py
ML_major_project2.py
ML_project.py
ML_project2.py
ML_project1.py
ML_major_project3.py
ML_project2.py > ...
Split the data manually
X_train = data.loc[:2, feature_columns]
y_train = data.loc[:2, target_column]
X_test = data.loc[3:4, feature_columns]
y_test = data.loc[3:4, target_column]
Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(data[feature_columns], data[target_column], test_size=0.2, random_state=42, stratify=None)
# X_train, X_test, y_train, y_test = train_test_split(data[feature_columns], data[target_column], train_size=4, test_size=1, random_state=42, stratify=None)
# Initialize the Random Forest classifier
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
# Train the model on the first 3 rows
rf_classifier.fit(X_train, y_train)
# Predict the target variable for the next 2 rows
predictions = rf_classifier.predict(X_test)
# Print the predictions
print("Predictions:", predictions)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\manas\Downloads\Python ChapterWise Notes>
Launchpad Run Testcases
Ln 70, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.3 64-bit Go Live
Type here to search
31°C
ENG
12:20 AM
07-07-2024


File Edit Selection View Go Run
Python ChapterWise Notes
GUI.py
ML_major_project.py
ML_major_project2.py
ML_project.py
ML_project2.py
ML_project1.py
ML_major_project3.py
ML_project2.py > ...
# X_train, X_test, y_train, y_test = train_test_split(data[feature_columns], data[target_column], train_size=4, test_size=1, random_state=42, stratify=None)
# Initialize the Random Forest classifier
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
# Train the model on the first 3 rows
rf_classifier.fit(X_train, y_train)
# Predict the target variable for the next 2 rows
predictions = rf_classifier.predict(X_test)
# Print the predictions
print("Predictions:", predictions)
print("Actual values:", y_test.values)
print("Confusion Matrix:\n", confusion_matrix(y_test.values, predictions, labels=[0,1]))
print("Accuracy Score:\n", accuracy_score(y_test.values, predictions))
print("Classification Report:\n", classification_report(y_test.values, predictions))
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\manas\Downloads\Python ChapterWise Notes>
Launchpad Run Testcases
Ln 70, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.3 64-bit Go Live
Type here to search
31°C
ENG
12:20 AM
07-07-2024