Project Development Phase – Online Payments Fraud Detection

1. Introduction

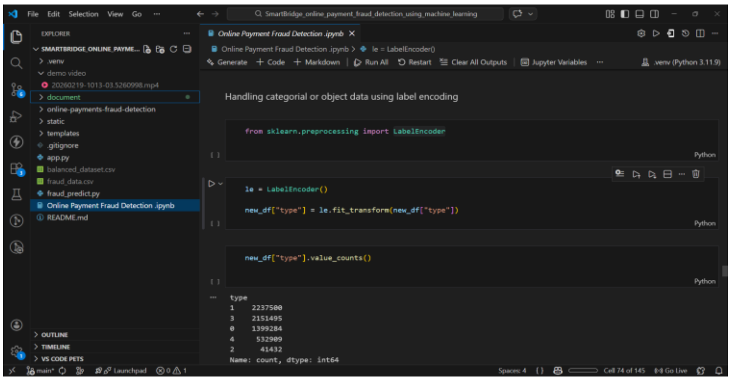
The **Project Development Phase** is the stage where the **concepts, design, and planning** are implemented into a **working system**. This phase includes **model building, training, testing, performance evaluation, and user interface development**.

2. Model Building and Training

2.1 Data Preprocessing

Before building the model, the data must be prepared:

- **Categorical Encoding:** Transaction type (type) is encoded using **LabelEncoder**.
- **Scaling:** Numerical features like amount, oldbalanceOrig, newbalanceOrig, oldbalanceDest, newbalanceDest are scaled using **StandardScaler**.
- **Handling Outliers:** Detected using IQR method and treated appropriately.



Handling categorial or object data using label encoding

```
from sklearn.preprocessing import LabelEncoder
```

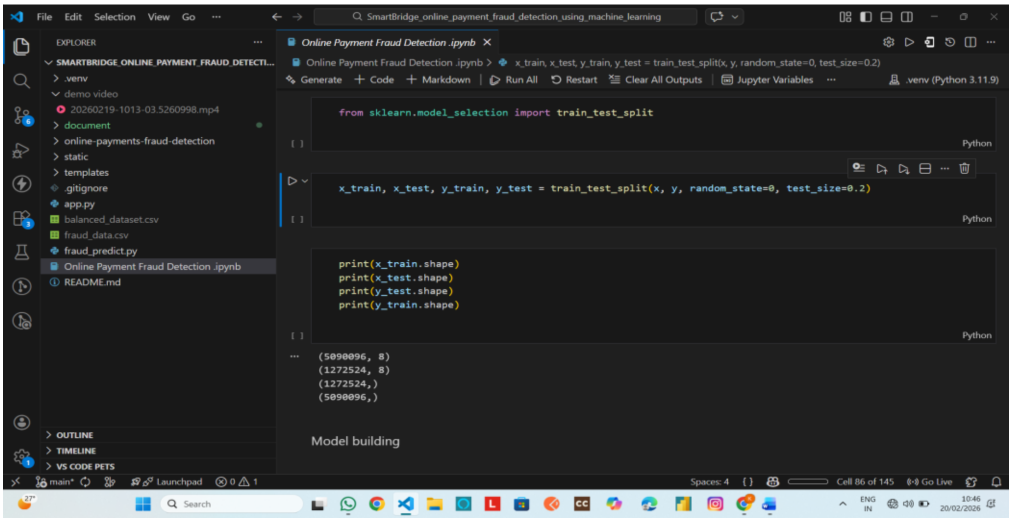
```
le = LabelEncoder()

new_df["type"] = le.fit_transform(new_df["type"])
```

```
new_df["type"].value_counts()
```

```
type
1    2237500
3    2151495
0    1399284
4     532909
2      41432
Name: count, dtype: int64
```

2.2 Splitting Dataset

- Split the dataset into **training (80%)** and **testing (20%)** for model evaluation.



```
from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=0, test_size=0.2)
```

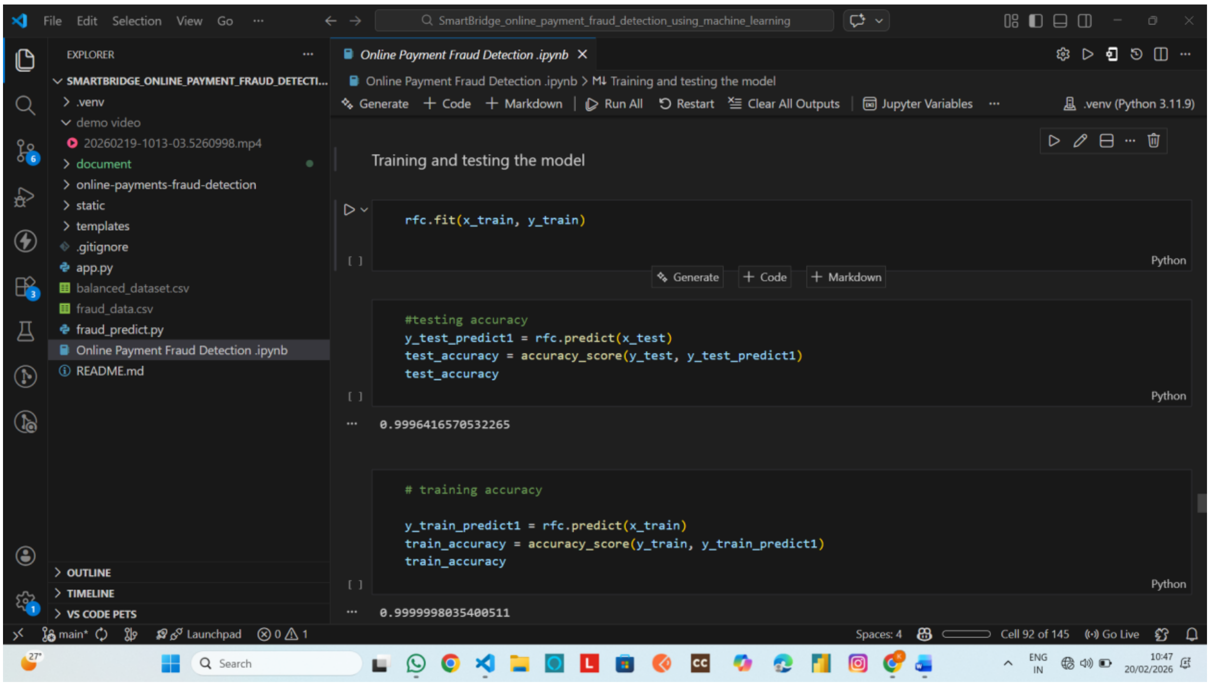
```
print(x_train.shape)
print(x_test.shape)
print(y_test.shape)
print(y_train.shape)
```

```
(5090096, 8)
(1272524, 8)
(1272524,)
(5090096,)
```

Model building

2.3 Model Training

- **Selected Algorithm:** Random Forest Classifier
- **Training:**



Training and testing the model

```
rfc.fit(x_train, y_train)
```

```
#testing accuracy
y_test_predict1 = rfc.predict(x_test)
test_accuracy = accuracy_score(y_test, y_test_predict1)
test_accuracy
```

```
0.9996416570532265
```

```
# training accuracy

y_train_predict1 = rfc.predict(x_train)
train_accuracy = accuracy_score(y_train, y_train_predict1)
train_accuracy
```

```
0.9999998035400511
```

Observation:

- The model achieved **high accuracy** in predicting legitimate vs. fraudulent transactions.
- Confusion matrix helps analyze **false positives and false negatives**.

3. Performance Testing

Performance testing ensures the model and system **respond quickly and accurately under expected loads**.

3.1 Metrics Used

- **Accuracy:** Percentage of correct predictions.
- **Precision:** Correct fraud predictions / Total predicted fraud.
- **Recall:** Correct fraud predictions / Total actual fraud.
- **F1-Score:** Harmonic mean of precision and recall.

Conclusion: Model performance is **satisfactory for real-time fraud detection**.

4. User Acceptance Testing (UAT)

- **Objective:** Ensure the system meets user expectations and requirements.

- **Process:**
    1. Users enter **transaction details** via the Flask web interface.
    2. The system predicts **fraudulent or legitimate transactions**.
    3. Users verify if results are **consistent with known transaction behavior**.

Criteria for UAT Success:

- Prediction output is **accurate**.
- Web interface is **user-friendly**.
- Input validation prevents **invalid transaction entries**.

5. Frontend Demonstration and Output

- **Home Page (home.html):** Form to input transaction details

Input Field	**Description**
Step	Transaction step/time
Type	Transaction type (CASH_IN, CASH_OUT, etc.)
Amount	Transaction amount
Old Balance Orig	Sender account balance before transaction
New Balance Orig	Sender account balance after transaction
Old Balance Dest	Receiver account balance before transaction
New Balance Dest	Receiver account balance after transaction

- **Submit Button:** Sends data to Flask backend for preprocessing and prediction.
- **Prediction Page (predict.html):** Displays results

Example Output:

Transaction Prediction: Fraudulent

or

Transaction Prediction: Legitimate

- **Additional Visualization (Optional):**
    - Countplot for transaction type vs. fraud
    - Distribution of transaction

6. Conclusion

- The **Project Development Phase** successfully implemented:
    - **Machine Learning model** for fraud detection.
    - **Performance testing** to ensure high accuracy and reliability.
    - **User Acceptance Testing** to validate system functionality.
    - **Frontend interface** for real-time prediction and result display.
- The system provides an **end-to-end solution** for detecting online payment fraud efficiently and effectively.