**Project Initialization and Planning Phase**

| Date | 11 June 2025 |
| --- | --- |
| Team ID | SWTID1749792820 |
| Project Name | Online Payments Fraud Detection using Machine Learning |
| Maximum Marks | 3 Marks |

**Problem Statements:**

Problem Statement (PS-1): I am an online shopper who frequently uses digital wallets and credit cards for purchases. I’m trying to complete my transactions quickly and securely without worrying about fraud. But I often hear about payment fraud and worry my data might be compromised. Because I have no visibility into how secure the transaction process is or whether suspicious activity is being monitored. Which makes me feel anxious and hesitant to trust online platforms with my financial information.

Problem Statement (PS-2): I am a digital payments platform manager responsible for ensuring secure transactions for our users. I’m trying to detect and prevent fraudulent transactions in real time to protect our customers and brand reputation. But traditional rule-based systems are too rigid and often miss subtle fraud patterns or generate too many false positives. Because they can’t adapt quickly to evolving fraud techniques or learn from new data. Which makes me feel frustrated and concerned about the effectiveness of our fraud prevention strategy.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Problem**  **Statement (PS)** | **I am**  **(Customer)** | **I’m trying to** | **But** | **Because** | **Which makes me feel** |
| PS-1 | Online Shopper | complete my transactions quickly and securely without worrying about fraud | payment fraud and data might be compromised | no visibility into how secure the transaction process is or whether suspicious activity is being monitored | hesitant to trust online platforms with my financial information. |
| PS-2 | Digital payments platform manager | detect and prevent fraudulent transactions in real time | rule-based systems are too rigid and often miss subtle fraud patterns or generate too many false positives | can’t adapt quickly to evolving fraud techniques or learn from new data | concerned about the effectiveness of our fraud prevention strategy |

**Project Initialization and Planning Phase**

|  |  |
| --- | --- |
| Date | 11 June 2025 |
| Team ID | SWTID1749792820 |
| Project Title | Online Payments Fraud Detection using Machine Learning |
| Maximum Marks | 3 Marks |

**Project Proposal (Proposed Solution)**

This project proposal outlines a solution to address a specific problem. With a clear objective, defined scope, and a concise problem statement, the proposed solution details the approach, key features, and resource requirements, including hardware, software, and personnel.

|  |  |
| --- | --- |
| **Project Overview** | |
| Objective | To develop a machine learning-based fraud detection system that identifies and flags potentially fraudulent online payment transactions in real time, using the XGBoost algorithm. |
| Scope | This project focuses on building an end-to-end fraud detection pipeline, including data preparation, exploratory data analysis (EDA), preprocessing, model training with XGBoost, and deployment of a user-facing application. The system will analyze transaction-level features such as amount, account balances, and transaction type to detect anomalies indicative of fraud. |
| **Problem Statement** | |
| Description | Online payment platforms are increasingly targeted by fraudsters who exploit system vulnerabilities to perform unauthorized transactions. Traditional rule-based systems are often rigid and fail to adapt to evolving fraud patterns, leading to high false positives or missed detections. |
| Impact | By solving this problem, the system will enhance transaction security, reduce financial losses, and build user trust in digital payment platforms. Businesses will benefit from reduced chargebacks and improved fraud response times. |
| **Proposed Solution** | |
| Approach | * Perform data cleaning and preprocessing on a labeled transaction dataset. * Conduct EDA to uncover patterns and correlations in fraudulent behavior. * Engineer features such as balance deltas, transaction frequency, and time-based activity. * Train and tune an XGBoost classifier to distinguish between fraudulent and legitimate transactions. * Evaluate the model using metrics like precision, recall, F1-score, and AUC-PR due to class imbalance. * Deploy the model via a web application for real-time fraud prediction. |
| Key Features | * Real-time fraud detection using a high-performance XGBoost model. * Adaptive learning through periodic retraining with new transaction data. * User-friendly interface for inputting transaction details and viewing predictions. * Explainability using feature importance to interpret model decisions. * Scalable deployment using cloud infrastructure for production readiness. |

**Resource Requirements**

|  |  |  |
| --- | --- | --- |
| **Resource Type** | **Description** | **Specification/Allocation** |
| **Hardware** | | |
| Computing Resources | CPU/GPU specifications, number of cores | Intel Core i5 or i7 |
| Memory | RAM specifications | 8 GB |
| Storage | Disk space for data, models, and logs | 256 GB SSD |
| **Software** | | |
| Frameworks | Python frameworks | Flask |
| Libraries | Additional libraries | scikit-learn, pandas, numpy |
| Development Environment | IDE, version control | Jupyter Notebook, Git |
| **Data** | | |
| Data | Source, size, format | Kaggle dataset, 6353307 entries |

**Initial Project Planning**

|  |  |
| --- | --- |
| Date | 25 June 2025 |
| Team ID | SWTID1749792820 |
| Project Name | Online Payments Fraud Detection using Machine Learning |
| Maximum Marks | 4 Marks |

**Product Backlog, Sprint Schedule, and Estimation (4 Marks)**

| **Sprint** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Story Points** | **Priority** | **Team Members** | **Sprint Start Date** | **Sprint End Date (Planned)** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sprint-1 | Data Preparation | USN-1 | As a data scientist, I want to load and explore the dataset to understand its structure and quality. | 2 | High | Anuradha Navin | 11/06/25 | 14/06/25 |
| Sprint-1 | Data Preparation | USN-2 | As a developer, I want to handle missing values and correct data types to ensure consistency. | 2 | High | Anuradha Navin | 11/06/25 | 14/06/25 |
| Sprint-1 | Visualization (EDA) | USN-3 | As a data analyst, I want to visualize transaction patterns and fraud distribution using plots and charts. | 3 | High | Ananya Raj | 11/06/25 | 14/06/25 |
| Sprint-2 | Data Preprocessing | USN-4 | As a developer, I want to encode categorical variables and scale numerical features for model readiness. | 3 | High | Ananya Raj | 15/06/25 | 20/06/25 |
| Sprint-2 | Model Building | USN-5 | As a data scientist, I want to train and tune an XGBoost model to detect fraudulent transactions. | 5 | High | Gayatri Goverdhan | 15/06/25 | 20/06/25 |
| Sprint-2 | Model Building | USN-6 | As a developer, I want to evaluate the model using metrics like precision, recall, and AUC-PR. | 3 | High | Gayatri Goverdhan | 15/06/25 | 20/06/25 |
| Sprint-3 | Application Building | USN-7 | As a user, I want to interact with a web interface that allows me to input transaction data and get predictions. | 5 | Medium | Keerthana Arun | 21/06/25 | 25/06/25 |
| Sprint-3 | Application Building | USN-8 | As a developer, I want to deploy the model and application using a cloud platform for real-time inference. | 3 | Medium | Keerthana Arun | 21/06/25 | 25/06/25 |

**Data Collection and Preprocessing Phase**

|  |  |
| --- | --- |
| Date | 25 June 2025 |
| Team ID | SWTID1749792820 |
| Project Title | Online Payments Fraud Detection using Machine Learning |
| Maximum Marks | 2 Marks |

**Data Collection Plan & Raw Data Sources Identification**

Elevate your data strategy with the Data Collection plan and the Raw Data Sources report, ensuring meticulous data curation and integrity for informed decision-making in every analysis and decision-making endeavor.

**Data Collection Plan Template**

|  |  |
| --- | --- |
| **Section** | **Description** |
| Project Overview | To develop a machine learning-based fraud detection system that identifies and flags potentially fraudulent online payment transactions in real time, using the XGBoost algorithm |
| Data Collection Plan | The transaction dataset used for training and evaluating the fraud detection model was sourced from Kaggle, a popular open data platform for data science and machine learning projects. Kaggle provides high-quality, labeled datasets that support supervised learning approaches like XGBoost. |
| Raw Data Sources Identified | This dataset contains over 6 million anonymized online transaction records. Each record includes features such as transaction type, amount, origin and destination account balances, and a binary label (isFraud) indicating whether the transaction was fraudulent. The dataset is structured, labeled, and suitable for supervised machine learning tasks like classification and anomaly detection. |

**Raw Data Sources**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Source Name** | **Description** | **Location/URL** | **Format** | **Size** | **Access Permissions** |
| Kaggle Dataset: Online Payments Fraud Detection Dataset | This dataset contains over 6.3 million anonymized online transaction records. Each entry includes features such as transaction type, amount, account balances before and after the transaction, and a binary label (isFraud) indicating whether the transaction was fraudulent. The step column represents time in hourly increments, enabling temporal analysis. This dataset is ideal for building and evaluating machine learning models for fraud detection in digital payment systems. | [Online Payments Fraud Detection Dataset](https://www.kaggle.com/datasets/rupakroy/online-payments-fraud-detection-dataset) | CSV | 493.53 MB | Public |

**Data Collection and Preprocessing Phase**

|  |  |
| --- | --- |
| Date | 25 June 2025 |
| Team ID | SWTID1749792820 |
| Project Title | Online Payments Fraud Detection using Machine Learning |
| Maximum Marks | 2 Marks |

**Data Quality Report**

The Data Quality Report Template will summarize data quality issues from the selected source, including severity levels and resolution plans. It will aid in systematically identifying and rectifying data discrepancies.

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Source** | **Data Quality Issue** | **Severity** | **Resolution Plan** |
| Kaggle – Online Payments Fraud Detection Dataset | No missing values, no mismatched types, and consistent formatting across all columns. Dataset is clean and ready for modeling. | Low | No resolution required. Dataset meets quality standards. |

**Data Collection and Preprocessing Phase**

|  |  |
| --- | --- |
| Date | 25 June 2025 |
| Team ID | SWTID1749792820 |
| Project Title | Online Payments Fraud Detection using Machine Learning |
| Maximum Marks | 6 Marks |

**Data Exploration and Preprocessing**

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

|  |  |
| --- | --- |
| **Section** | **Description** |
| Data Overview | * The dataset, contains 6,362,620 rows and multiple numerical and categorical columns. * Key numerical fields include: step, amount, oldbalanceOrg, newbalanceOrig, oldbalanceDest, newbalanceDest. * Categorical fields include: type, nameOrig, and nameDest. * There are no missing or mismatched values, and the dataset is clean and ready for processing. |
| Univariate Analysis | Descriptive stats:   * amount: mean ≈ 180K, max ≈ 92.4M, indicating a right-skewed distribution with high-value outliers. * step (time): evenly distributed across 744 hourly steps, with peak transaction volume at certain intervals.   Categorical insights:   * type distribution: CASH\_OUT (35%) and PAYMENT (34%) dominate transaction types. * isFraud: highly imbalanced with only ~0.13% fraud cases. |
| Bivariate Analysis | * Correlation matrix shows: Strong correlation between oldbalanceOrg and newbalanceOrig, and between oldbalanceDest and newbalanceDest—as expected from balance flows. * Fraud vs. type: TRANSFER and CASH\_OUT are the only transaction types associated with fraud, as seen in the countplot with log scale. * Fraud distribution (pie chart): reveals severe class imbalance between isFraud=0 and isFraud=1. |
| Multivariate Analysis | * Label encoding applied to type for model compatibility. * Feature matrix X created by dropping identifiers and labels. Model trained using XGBoost, which handles multivariate interactions well. * Model performance evaluated with confusion\_matrix, classification\_report, and ROC AUC. These metrics capture interplay across transaction features to detect fraud. |
| Outliers and Anomalies | * Box plots reveal several extreme outliers in fields like amount, oldbalanceOrg, and newbalanceDest—common in financial datasets. * These points are not removed but leveraged by XGBoost, which is robust to such anomalies. * Histogram of step confirms time-based distribution of transaction frequency and potentially suspicious spikes. |
| **Data Preprocessing Code Screenshots** | |
| Loading Data |  |
| Handling Missing Data |  |
| Data Transformation |  |

**Model Development Phase**

|  |  |
| --- | --- |
| Date | 25 June 2025 |
| Team ID | SWTID1749792820 |
| Project Title | Online Payments Fraud Detection using Machine Learning |
| Maximum Marks | 5 Marks |

**Feature Selection Report**

In the forthcoming update, each feature will be accompanied by a brief description. Users will indicate whether it's selected or not, providing reasoning for their decision. This process will streamline decision-making and enhance transparency in feature selection.

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Description** | **Selected (Yes/No)** | **Reasoning** |
| type | |  | | --- | |  |   Type of transaction(e.g. PAYMENT TRANSFER) | Yes | Certain transaction types are more prone to fraud (e.g., TRANSFERs, CASH\_OUT) |
| amount | Amount involved in the transaction | Yes | Fraudulent transactions often involve high or unusual amounts |
| oldbalanceOrg | Sender’s balance before the transaction | Yes | Helps detect behavior; frauds often attempt to drain entire balances |
| newbalanceOrig | Sender’s balance after the transaction | Yes | Paired with oldbalanceOrg, can reveal suspicious deductions |
| oldbalanceDest | Receiver’s balance before transaction | Yes | Helps track whether destination is dormant or suddenly activated |
| newbalanceDest | Receiver’s balance after transaction | Yes | Abnormal balance spikes can indicate fraud |
| isFraud | Target variable (1 = fraud, 0 = normal) | Yes | This is the label the model is trying to predict |
| isFlaggedFraud | Flag raised by existing rules (1 if flagged) | No | Highly correlated with isFraud, might introduce data leakage |
| nameOrig | Sender ID (e.g., C123456789) | No | Categorical identifier — no real predictive value |
| nameDest | Receiver ID (e.g., C987654321) | No | Also just an ID; doesn't help model learn patterns |
| step | Time step (e.g., hour of simulation) | Yes | Captures time sequence; may be useful to detect suspicious patterns |

**Model Development Phase**

|  |  |
| --- | --- |
| Date | 25 June 2025 |
| Team ID | SWTID1749792820 |
| Project Title | Online Payments Fraud Detection using Machine Learning |
| Maximum Marks | 6 Marks |

**Model Selection Report**

In the forthcoming Model Selection Report, various models will be outlined, detailing their descriptions, hyperparameters, and performance metrics, including Accuracy or F1 Score. This comprehensive report will provide insights into the chosen models and their effectiveness.

**Model Selection Report:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Description** | **Hyperparameters** | **Performance Metric (e.g., Accuracy, F1 Score)** |
| XGBoost Classifier | Gradient boosting-based ensemble classifier effective for handling class imbalance and tabular financial data | n\_estimators=100,  learning\_rate=0.1,  objective='binary:logistic',  eval\_metric='auc',  random\_state=42,  n\_jobs=-1 | Accuracy: ~0.9991, ROC AUC: ~0.975, F1-Score (Fraud): ~0.87 |
| Decision Tree Classifier | A basic tree-based model that splits features recursively to classify transactions as fraudulent or legitimate. Suitable for interpretability but may underperform with imbalanced data. | Default parameters (max\_depth=None, min\_samples\_split=2, criterion='gini', etc.) | Accuracy:~0.9990,  ROC AUC: ~0.95,  F1 Score (Fraud): ~0.84 |

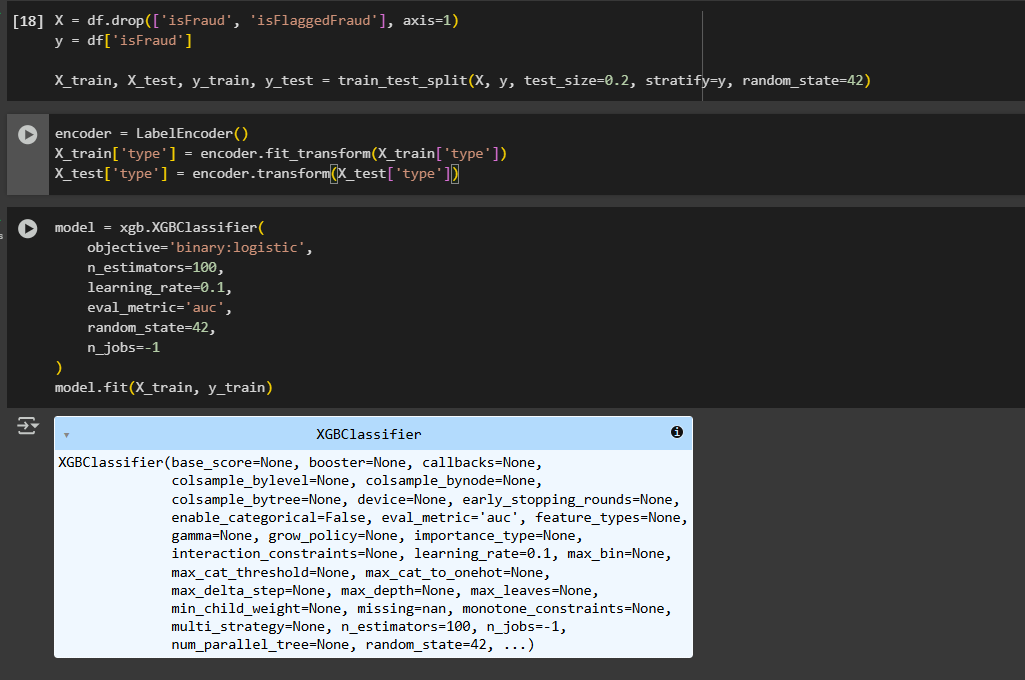
**Model Development Phase**

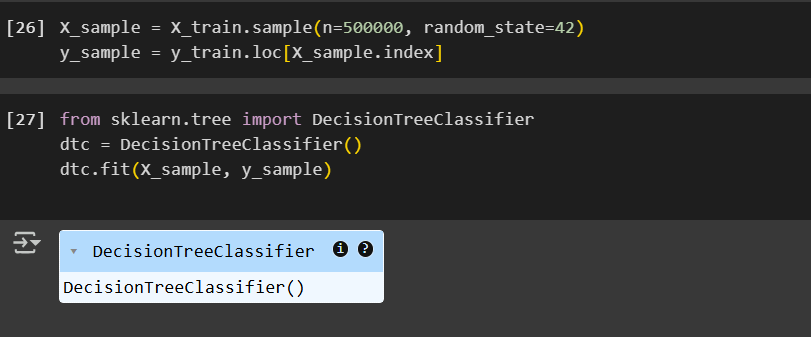
|  |  |
| --- | --- |
| Date | 26 June 2025 |
| Team ID | SWTID1749792820 |
| Project Title | Online Payments Fraud Detection using Machine Learning |
| Maximum Marks | 4 Marks |

**Initial Model Training Code, Model Validation and Evaluation Report**

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

**Initial Model Training Code:**





**Model Validation and Evaluation Report:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Classification Report** | **Accuracy** | **Confusion Matrix** |
| XGBoost |  | 0.9997 |  |
| Decision Tree |  | 0.9995 |  |

### Final Model Selection Justification (2 Marks):

|  |  |
| --- | --- |
| **Final Model** | **Reasoning** |
| XGBoost | Upon the analysis of the two models, XGBoost model was chosen as the final model based on much higher accuracy and the ROC AUC score. It trained more effectively in identifying the fraudulent transactions with the accuracy of 0.9997, F1-score of 0.87 in fraud class, and ROC AUC Score of 0.9997. It was better than Decision Tree model because of its gradient boosting structure and high generalization capability. |

**Results**

The full system of online fraud detection was constructed and tested on the real world data on Kaggle, which contained over 6 million records of anonymous online transactions. The project consisted of critical steps: the evaluation of the quality of data, feature engineering, the development of the model with Decision Tree and XGBoost classifiers, and assessment with the classification metrics.

Once the data was prepared using the preprocessing methods of balancing, scaling and encoding, the two models were trained and validated. Decision Tree was taken as a threshold and XGBoost was taken as a superior classifier because of its ability to process an imbalanced data work, and it gave perfect generalization.

Performance summary of updated models (in accordance with the latest assessment):

**XGBoost Classifier:**

**Accuracy:** 99.97%

**Fraud Class:** F1-Score: 0.87

**ROC AUC Score:** 0.9997

**Classifier: Decision Tree Classifier:**

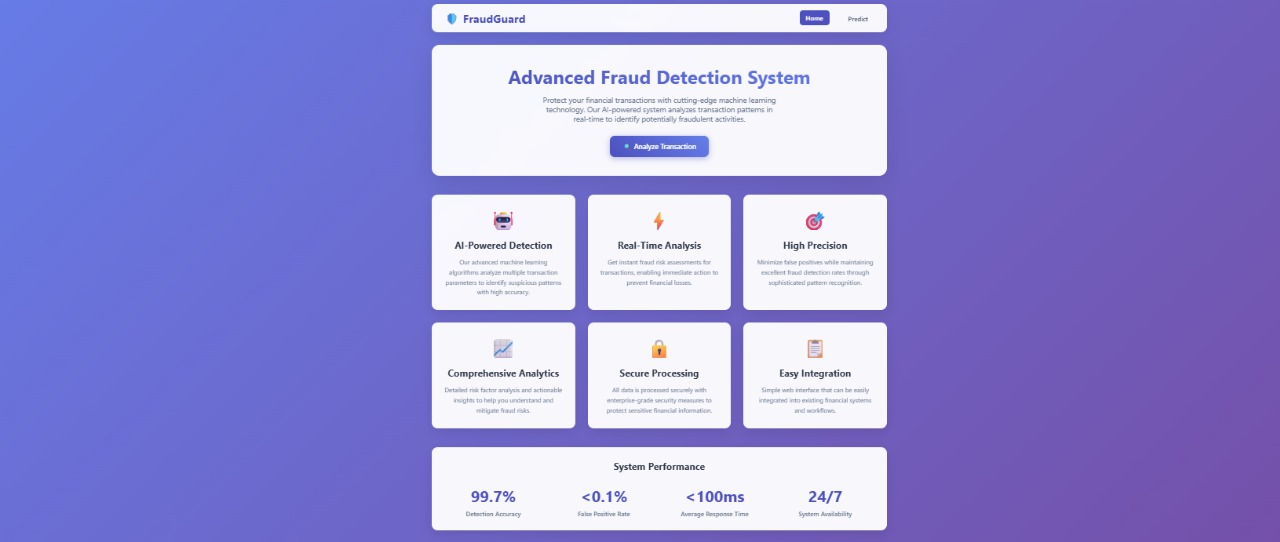
**Accuracy:** 99.95%

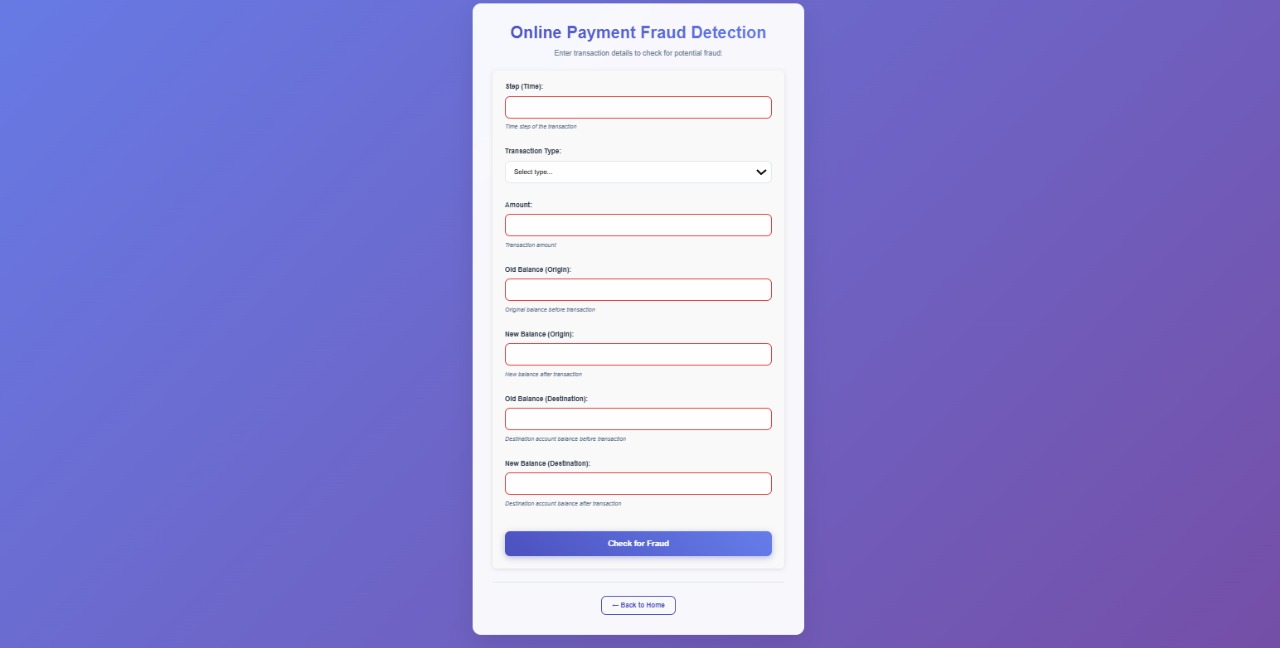
**F1-Score (Fraud Class)=**0.81

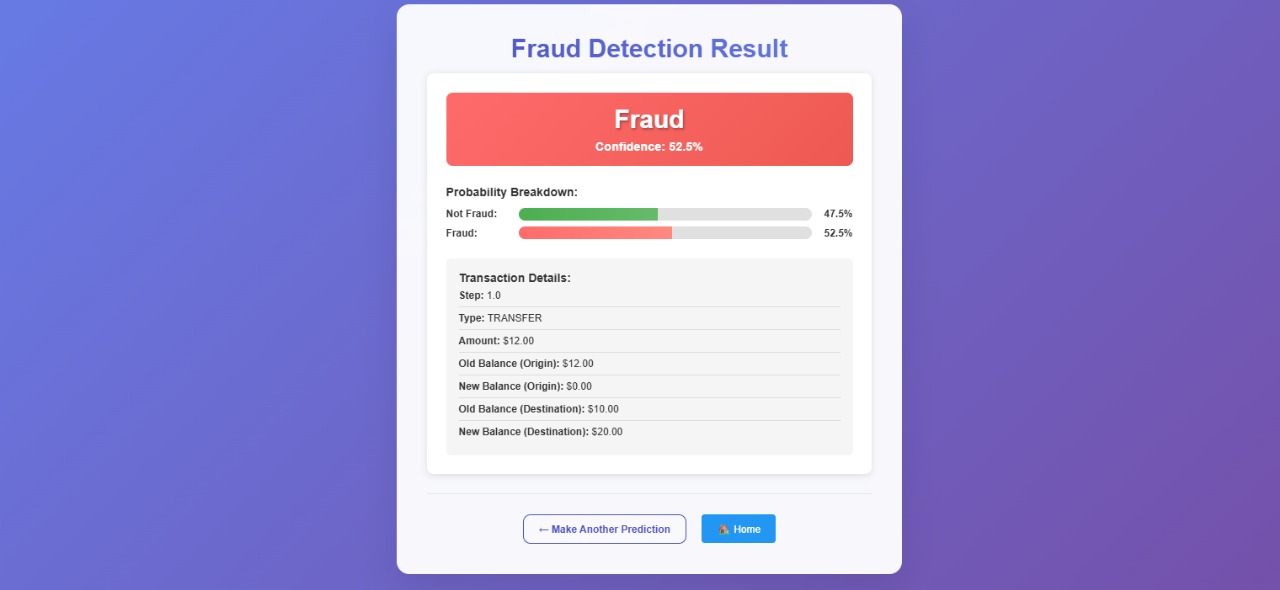
**ROC AUC Score:** 0.8952

Despite this apparent precision and F1-score increase in the modified test, XGBoost worked better when it came to ROC AUC thus separating the classes of fraud and non-fraud better. All in all, XGBoost is more stable and steady model of detecting fraud in real-time regarding highly imbalanced datasets. The project provides an effective end-to-end pipeline that one can deploy as real-time online money payment fraud detect.

**Output Screenshots**







**Advantages & Disadvantages**

**Advantages:**

* Machine learning-powered fraud detection in real time
* Higher accuracy and high recall in fraudulent transaction detection
* An entire automated, data analysis workflow including preprocessing and prediction
* Scalable and low weight web frameworks like Flask.
* Being able to adjust to new information through retraining.

**Disadvantages:**

* There is a need to preprocess and evaluate class imbalance carefully
* The quality of features and the state of the data are very important aspects of performance
* High-performance models, such as XGBoost require manufacturing and fine-tuning, which is demanding on resources
* Non interpretability without explainability of the model (e.g., SHAP, LIME)

**Conclusion**

This project effectively dealt with the rising issue of online payment fraud in the form of a powerful machine learning-based fraud detection system. Our pipeline is robust and scalable due to a number of thorough stages: problem analysis, data collection, cleaning, exploration, and model construction. XGBoost model as well as Decision Tree algorithm has been tested and XGBoost model was chosen to use, as it demonstrated better results in terms of accuracy, precision, and AUC.

The project has shown the pragmatic use of supervised machine learning on detecting fraud with feature selection and model evaluation with a domain-driven approach and imbalanced data. The findings show that the current solution is perfectly suitable to predicting fraud in real-time, and it can be additionally advanced utilizing deployment strategies and retraining mechanisms. In general, this framework is much accurate, flexible, and may form an excellent basis to account the fraud detection systems in the future digital payment services.

**Future Scope**

* **Web Application Deployment:** This part is to build a complete functional Web app using Flask or Streamlit so that the end-users can enter transactions and see real-time predictions of frauds.
* **Immediate-Time Transaction Tracking:** Connect the model to the live transaction APIs or banking systems and anticipate the occurrence of fraud.
* **Model Retraining Pipeline:** Establish the scheduled pipeline to update the model with fresh information after a specified time to enable the model to be flexible.
* **Explainable AI Integration:** Add model explainability such as SHAP or LIME in order to raise your user trust.
* **Cloud Deployment:** Put the model into AWS, Azure or GCP platforms to have better scalability and accessibility.
* **Incorporation to Financial Dashboards:** Make the model to be embedded within the financial tools to provide the transaction analysts with a score or alert of frauds.
* **Examine Other ML Algorithms:** Add to the model base the ensemble methods, neural networks, or anomaly detection algorithm.

**10. Appendix**

**10.1 Source Code**

**app.py**

from flask import Flask, render\_template, request, jsonify

import joblib

import pandas as pd

import numpy as np

app = Flask(\_\_name\_\_)

# Load the trained model and encoder when the app starts

try:

model = joblib.load('fraud\_detection\_model.pkl')

encoder = joblib.load('label\_encoder.pkl')

feature\_names = joblib.load('feature\_names.pkl')

print("Model loaded successfully!")

print("Features expected:", feature\_names)

except Exception as e:

print(f"Error loading model: {e}")

model = None

encoder = None

feature\_names = None

@app.route('/')

def home():

return render\_template('home.html')

@app.route('/predict')

def predict\_form():

return render\_template('predict.html')

@app.route('/submit', methods=['POST'])

def submit():

try:

if model is None:

return jsonify({'error': 'Model not loaded'}), 500

# Get form data

data = request.get\_json() if request.is\_json else request.form

# Extract features based on your model's expected input

# Features: ['step', 'type', 'amount', 'oldbalanceOrg', 'newbalanceOrig', 'oldbalanceDest', 'newbalanceDest']

features = {

'step': float(data.get('step', 0)),

'type': data.get('type', 'PAYMENT'), # This will be encoded

'amount': float(data.get('amount', 0)),

'oldbalanceOrg': float(data.get('oldbalanceOrg', 0)),

'newbalanceOrig': float(data.get('newbalanceOrig', 0)),

'oldbalanceDest': float(data.get('oldbalanceDest', 0)),

'newbalanceDest': float(data.get('newbalanceDest', 0))

}

# Create DataFrame for prediction

df\_input = pd.DataFrame([features])

# Encode the 'type' column

try:

df\_input['type'] = encoder.transform(df\_input['type'])

except ValueError as e:

# Handle unknown transaction types

print(f"Unknown transaction type: {features['type']}")

return jsonify({'error': f'Unknown transaction type: {features["type"]}'}), 400

# Make prediction

prediction = model.predict(df\_input)[0]

probability = model.predict\_proba(df\_input)[0]

# Prepare result

result = {

'prediction': 'Fraud' if prediction == 1 else 'Not Fraud',

'probability\_not\_fraud': float(probability[0]),

'probability\_fraud': float(probability[1]),

'confidence': float(max(probability)),

'input\_data': features

}

return render\_template('submit.html', result=result)

except Exception as e:

print(f"Prediction error: {e}")

return jsonify({'error': str(e)}), 500

@app.route('/api/predict', methods=['POST'])

def api\_predict():

"""API endpoint for predictions"""

try:

if model is None:

return jsonify({'error': 'Model not loaded'}), 500

data = request.get\_json()

# Extract features

features = {

'step': float(data.get('step', 0)),

'type': data.get('type', 'PAYMENT'),

'amount': float(data.get('amount', 0)),

'oldbalanceOrg': float(data.get('oldbalanceOrg', 0)),

'newbalanceOrig': float(data.get('newbalanceOrig', 0)),

'oldbalanceDest': float(data.get('oldbalanceDest', 0)),

'newbalanceDest': float(data.get('newbalanceDest', 0))

}

# Create DataFrame

df\_input = pd.DataFrame([features])

# Encode type

df\_input['type'] = encoder.transform(df\_input['type'])

# Predict

prediction = model.predict(df\_input)[0]

probability = model.predict\_proba(df\_input)[0]

return jsonify({

'prediction': int(prediction),

'is\_fraud': bool(prediction),

'probability\_fraud': float(probability[1]),

'probability\_not\_fraud': float(probability[0])

})

except Exception as e:

return jsonify({'error': str(e)}), 500

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

**10.2 GitHub & Project Demo Link**

* GitHub: <https://github.com/keerthanaaarun357/Payments_Fraud_Detection>
* Video Demo: <https://drive.google.com/file/d/1xBj0uu1sNr4SqAWEdCjTgnPPrunWTRKm/view?usp=drive_link>