Credit Card Fraud Detection System

Team Members : **Ronak Patel** **-** **IU2380820170**

Himanshu Patel **-** **IU2380820233**

Project Guide **:** **Mr. Vivek Dave**

Introduction

- Credit card fraud is a serious problem in the field of banking and finance.

- Millions of fake transactions cause huge losses each year.

- The project will detect fraudulent transactions with **Machine Learning** models.

- System combines **Data Analysis**, **Model Training**, **Deployment** , and **Visualization**.

- **The objective of this project is to:**

- Analyze credit card transactional data.

- Train ML models to detect fraud patterns.

- Deploy the best model in a user-friendly web application(**Streamlit**).

- Present clear visual insights using **Power BI** dashboards.

Project Profile

- **Project Title:** Credit Card Fraud Detection System.

- **Domain Area:** Machine Learning, Data Science, and FinTech.

- **Dataset: Kaggle** (Credit Card Transactions Dataset).

- **Frontend:** Streamlit Web Application.

- **Visualization:** Power BI Dashboard.

- **Objective:** Classifying transactions accurately, Reduce fraud risk, increase security, and improve trust in digital payments.

Scope of the System

- Detect fraudulent transactions with high accuracy.

- Handles large transaction datasets.

- Reduces risks of financial loss.

- Can be integrated with banking systems, e-commerce, fintech apps.

- MySQL for secure data storage & querying.

- Python EDA for data insights.

- Power BI dashboards for fraud trend visualization.

Modules

1 — **Data Collection Module:**

Fetch dataset from Kaggle, Store in MYSQL, and Load into Python for analysis.

2 — **Data Preprocessing Module:**

Data Cleaning, Handle missing values, Normalization, etc.

3 — **Model Training Module:**

Train ML models (Logistic Regression, Decision Tree, Random Forest, XGBoost, etc.).

4 — **Evaluation Module:**

Model comparison on Accuracy, Precision, Recall, F1-Score, ROC-AUC.

5 — **Deployment Module:**

Deploy best model with Streamlit application.(Users can input data and see prediction.)

6 — **Visualization Module:**

Create Power BI dashboards for fraud trends, fraud distribution, monthly fraud cases, etc.

Tools & Technology

- **Programming Language:** Python

- **Libraries Used:** Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, etc.

- **Model Deployment:** Streamlit (for web app)

- **Visualization:** Microsoft Power BI

- **Data Source:** Kaggle

Analysis

- **Challenge:** Dataset highly imbalanced.

- **Solution:** Used resampling techniques like SMOTE to balance data.

- EDA reveals key fraud patterns.

- Compared multiple ML models to find best performer.

- Achieved Confusion Matrix, ROC-AUC score, F1-score for evaluation.

- **Visual analysis with Power BI:**

- Fraud is most common in small transaction amounts.

- Fraud patterns differ across time of day.

Need for the System

- Increasing cybercrime and online fraud cases worldwide.

- Banks rely on Rule-based/Manual fraud detection which is Slow and Inefficient.

- Machine Learning provides Fast, Accurate and Automated Detection.

- ML-based system can adapt and learn from new fraud behavior.

- Protects customers and banks from financial and reputational loss.

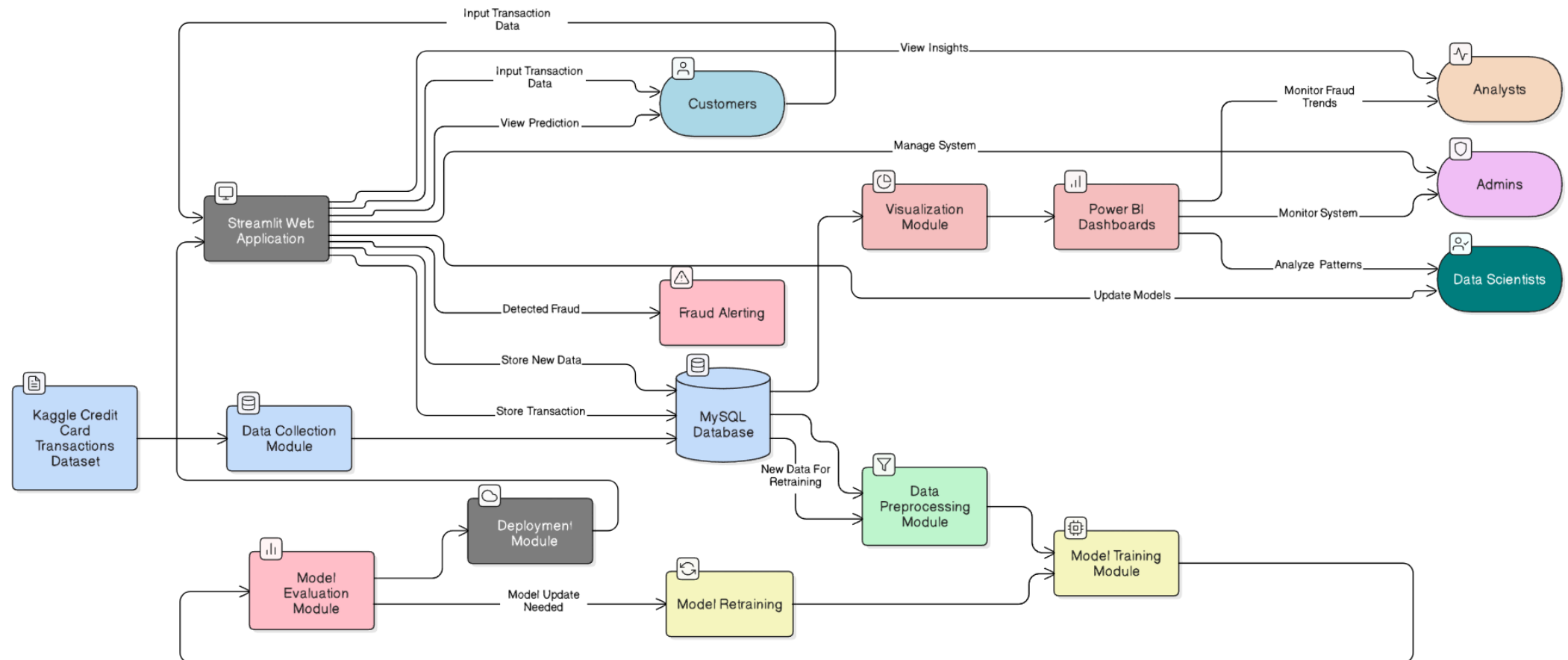- Provides real-time detection with better accuracy.

Existing System

- Traditional Fraud Detection systems rely on rule-based/manual checks:

- If transaction are greater than certain amount then flag as fraud.

- Works only for predefined rules.

- Cannot adapt to new fraud patterns.

- Limited accuracy and adaptability to new fraud patterns.

- Delay in fraud detection.
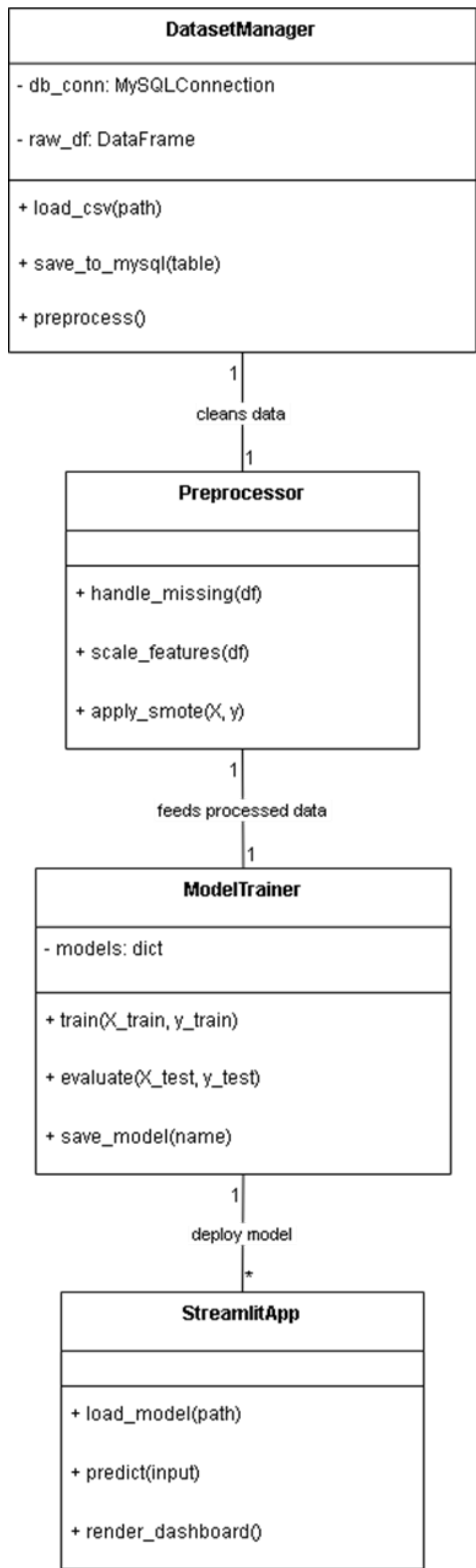
Need for the New System

- Uses Machine Learning instead of static rules.

- Machine Learning system automatically learns patterns.

- Detects fraud patterns in real-time with higher accuracy, precision, and recall.

- Reduces false alarms and detects actual frauds.

- Scalable, efficient, and adaptive to new fraud techniques.

- User-friendly Streamlit web application for user interaction.

- Power BI dashboards provide clear and easily understandable insights for effective decision-making.
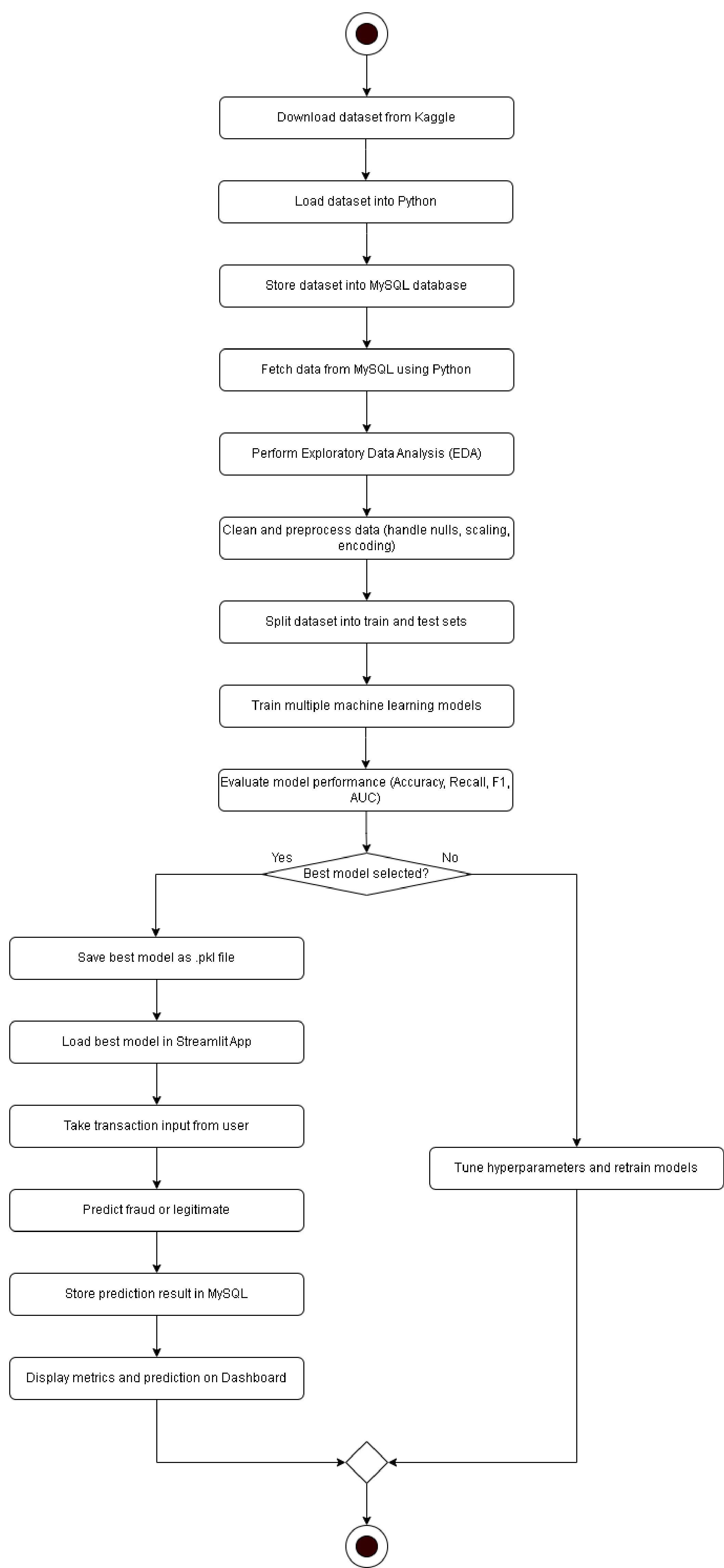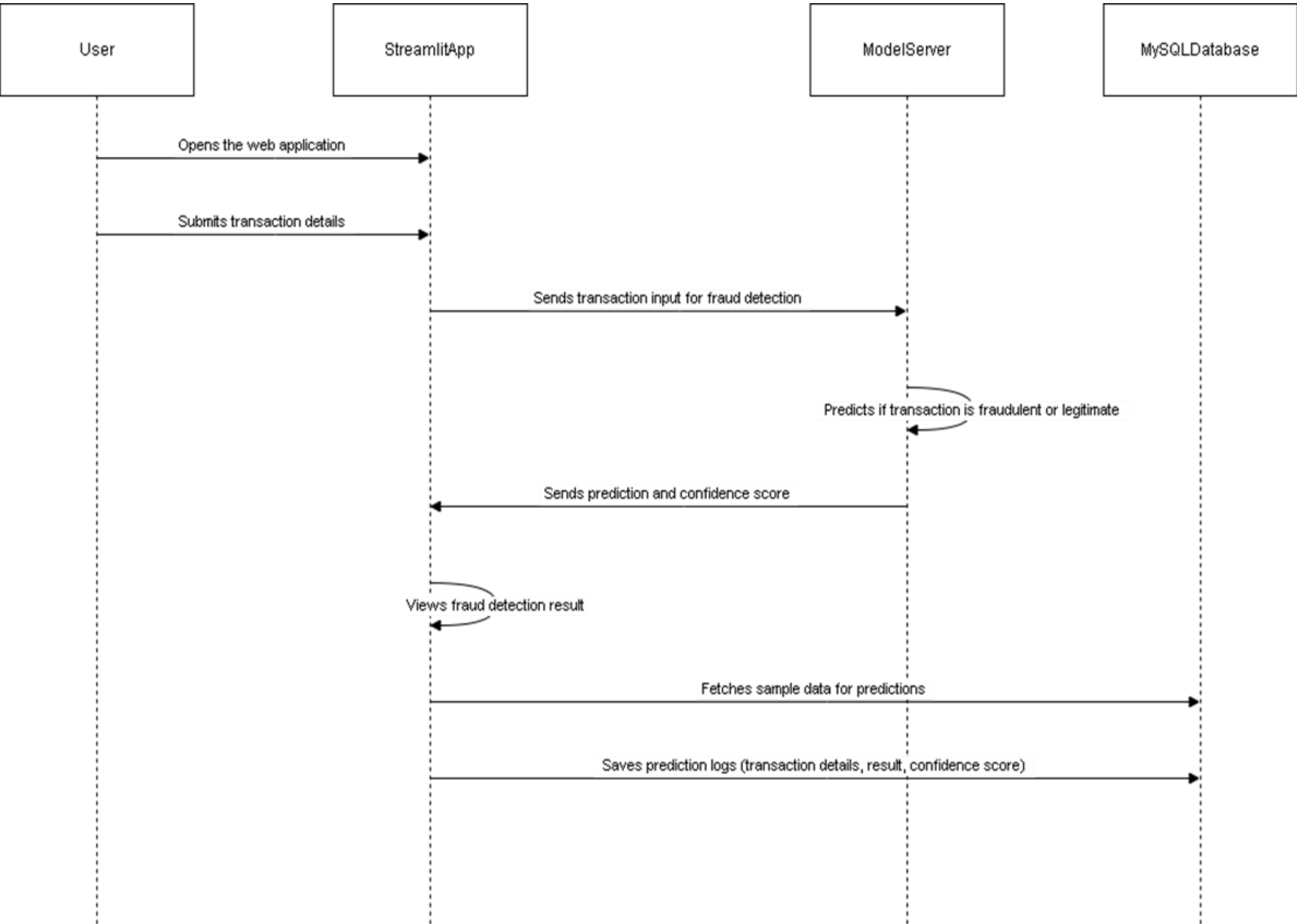
System Flow Diagram
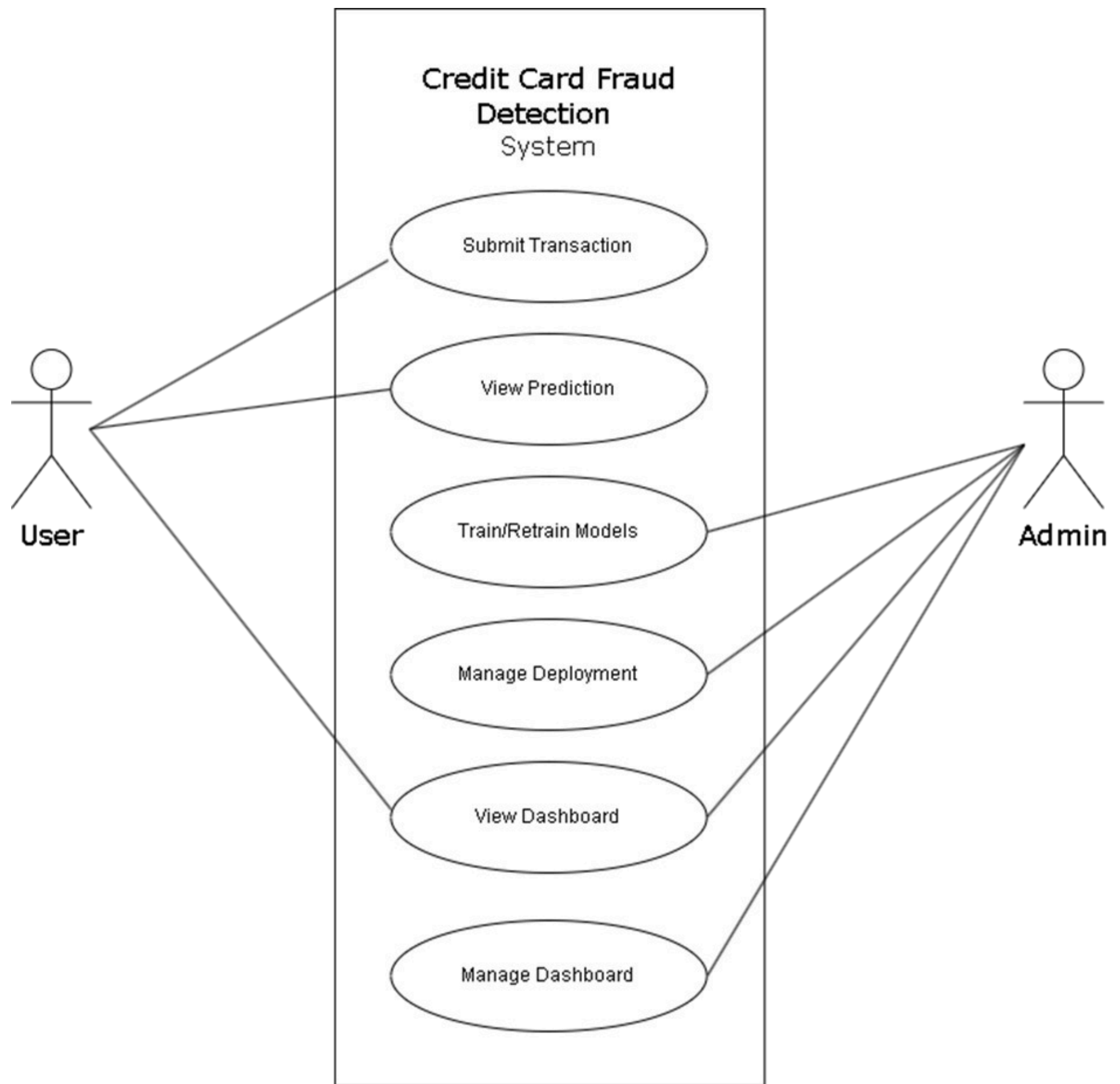
UML Diagrams:

Class Diagram-

DatasetManager

- db_conn: MySQLConnection
- raw_df: DataFrame

+ load_csv(path)
+ save_to_mysql(table)
+ preprocess()

1
cleans data
1

Preprocessor

+ handle_missing(df)
+ scale_features(df)
+ apply_smote(X, y)

1
feeds processed data
1

ModelTrainer

- models: dict

+ train(X_train, y_train)
+ evaluate(X_test, y_test)
+ save_model(name)

1
deploy model
*

StreamlitApp

+ load_model(path)
+ predict(input)
+ render_dashboard()

Activity Diagram-

```
        ●
        │
        ▼
┌───────────────────────────┐
│  Download dataset from Kaggle  │
└───────────────────────────┘
        │
        ▼
┌───────────────────────────┐
│   Load dataset into Python      │
└───────────────────────────┘
        │
        ▼
┌───────────────────────────┐
│ Store dataset into MySQL database │
└───────────────────────────┘
        │
        ▼
┌───────────────────────────┐
│  Fetch data from MySQL using Python │
└───────────────────────────┘
        │
        ▼
┌───────────────────────────┐
│ Perform Exploratory Data Analysis (EDA) │
└───────────────────────────┘
        │
        ▼
┌───────────────────────────┐
│ Clean and preprocess data (handle nulls, scaling, encoding) │
└───────────────────────────┘
        │
        ▼
┌───────────────────────────┐
│ Split dataset into train and test sets │
└───────────────────────────┘
        │
        ▼
┌───────────────────────────┐
│ Train multiple machine learning models │
└───────────────────────────┘
        │
        ▼
┌───────────────────────────┐
│ Evaluate model performance (Accuracy, Recall, F1, AUC) │
└───────────────────────────┘
        │
        ▼
```

Yes ◇ Best model selected? ◇ No

- Save best model as .pkl file
- Load best model in Streamlit App
- Take transaction input from user
- Predict fraud or legitimate
- Store prediction result in MySQL
- Display metrics and prediction on Dashboard

- Tune hyperparameters and retrain models

◇ → ●

Sequence Diagram-

Use Case Diagram-

Credit Card Fraud Detection System

- Submit Transaction
- View Prediction
- Train/Retrain Models
- Manage Deployment
- View Dashboard
- Manage Dashboard

User

Admin

Data Dictionary

Column Name	Description	Type
id	Unique transaction identifier	Integer
V1–V28	Anonymized numerical features (PCA components) from transaction data	Float
Amount	Transaction amount	Float
Class	Target variable — 1 = Fraud, 0 = Legitimate	Integer

Project Screenshots

💳 Credit Card Fraud Detection System

Predict fraudulent transactions using a trained **Random Forest** model with all 30 features.

● Single Prediction   📁 Bulk Prediction (CSV)   📊 Dashboard

Enter Transaction Details (All 30 Features)

Enter values for **Time, V1–V28, and Amount** below:

Time	V11	Amount
0.05	8.00	50000.00

V1	V12	V20
0.01	9.00	-5.00

V2	V13	V21
4.00	5.00	-8.00

V3	V14	V22
5.00	5.00	-2.00

8.00	9.00	0.00

V5	V16	V24
9.00	0.00	-8.00

V6	V17	V25
0.00	8.00	-8.00

V7	V18	V26
4.00	9.00	-9.00

V8	V19	V27
5.00	5.00	-60.00

| V9		V28
5.00		50.00

V10
8.00

🔍 Predict Fraud Status

💳 Credit Card Fraud Detection System

Predict fraudulent transactions using a trained **Random Forest** model with all 30 features.

◆ Single Prediction   📁 Bulk Prediction (CSV)   📊 Dashboard

Fraud Detection Dashboard

📊 Upload a CSV file with predictions

☁️ Drag and drop file here
Limit 200MB per file • CSV

[Browse files]

📄 fraud_predictions.csv 8.9KB   ✕

◆ **Fraud vs Legitimate Count**   ◆ **Fraud Proportion (%)**

Upload CSV File for Fraud Detection (must have 30 feature columns)
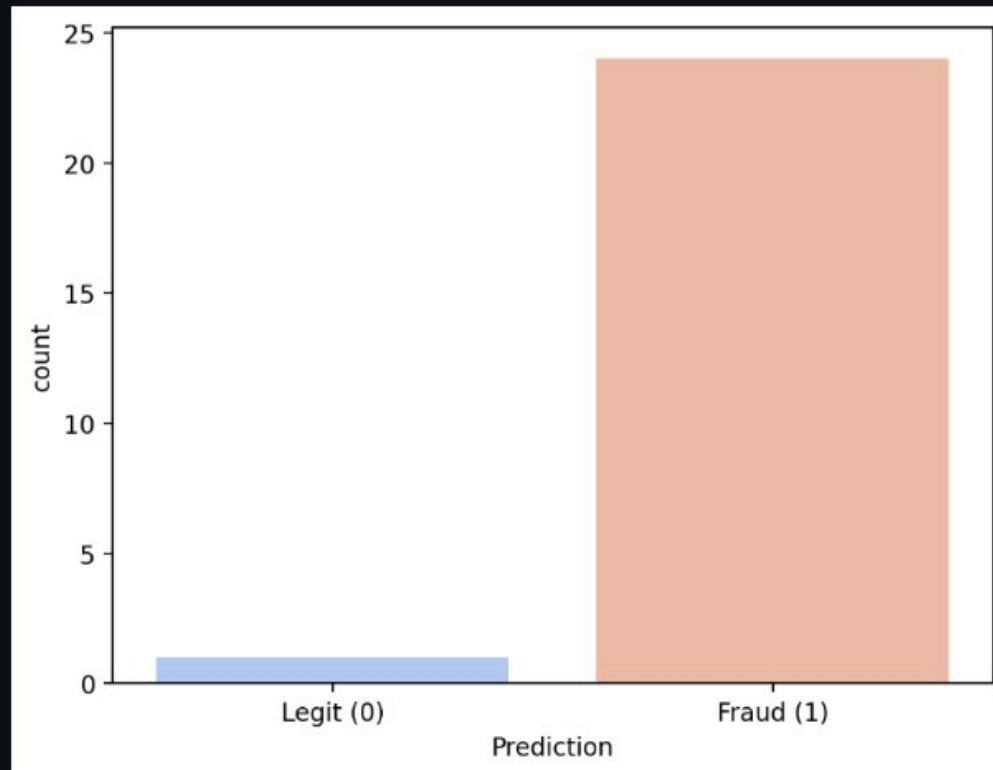
🟨 Upload your CSV file
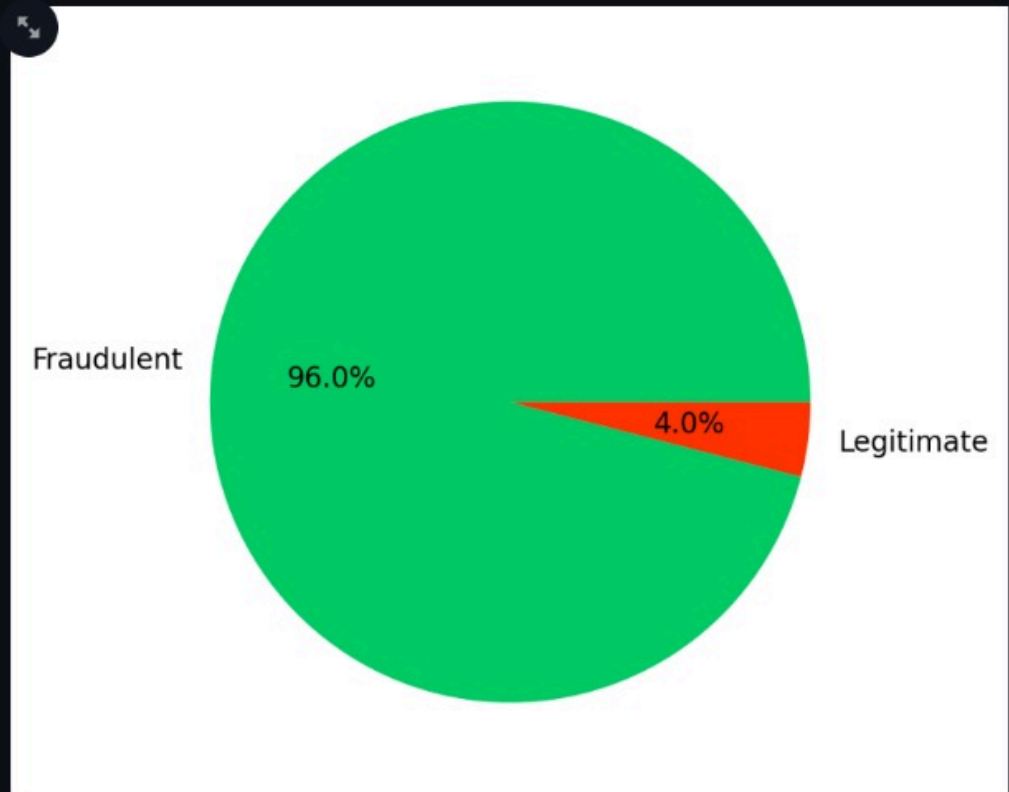
☁️ Drag and drop file here
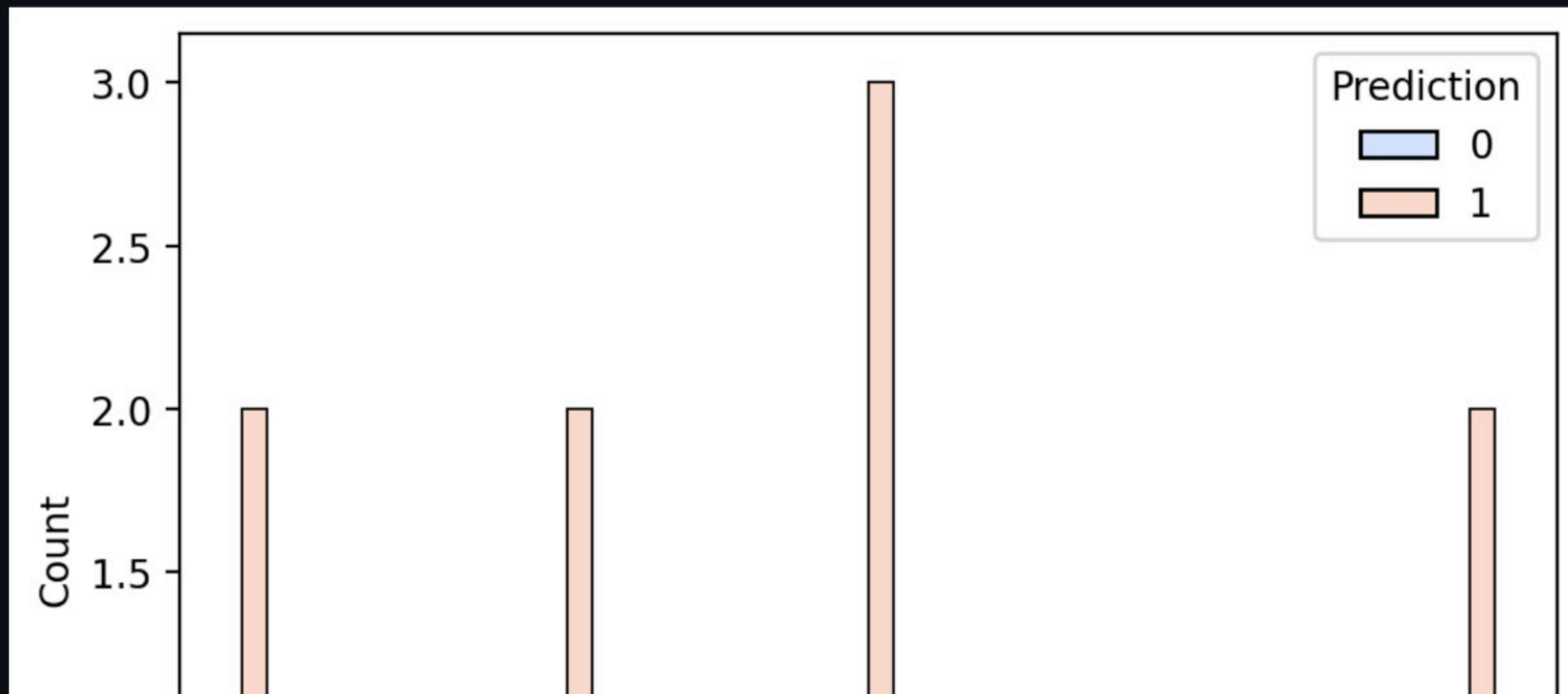Limit 200MB per file • CSV

[Browse files]
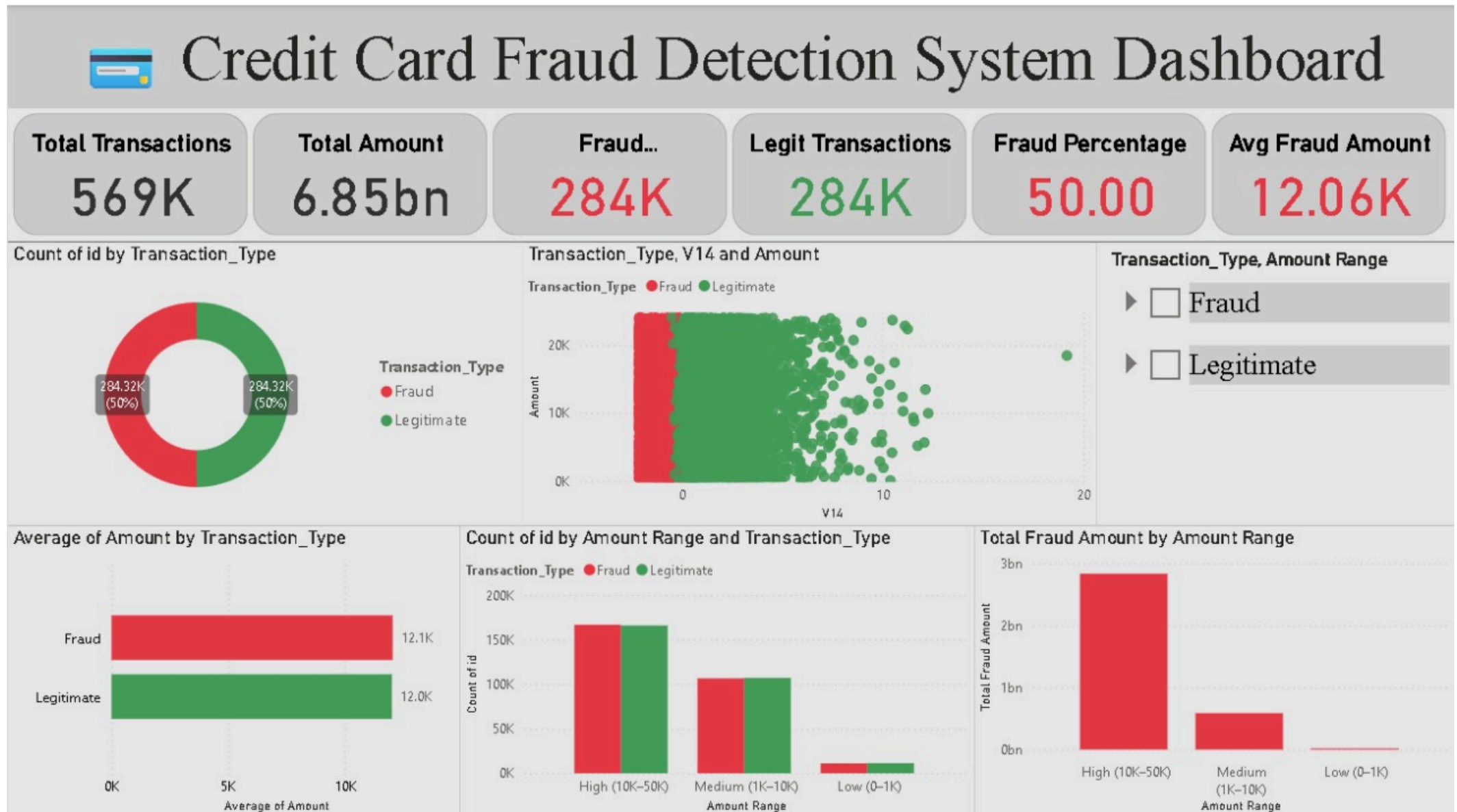
◆ Fraud vs Legitimate Count

◆ Fraud Proportion (%)

💰 Transaction Amount Distribution by Prediction

Power BI Dashboard

Test Plan and Test Cases:

Test Plan-

Test Types: Unit, Integration, System, Performance, Security.

Environment: Streamlit (frontend), MySQL (backend), Scikit-learn models.

Goal: Ensure reliability, accuracy, and data consistency across components.

Test Cases-

TC ID	Test Case Description	Expected Outcome	Actual Result
TC1	Load dataset (CSV → DataFrame)	Dataset loads successfully	Yes dataset loaded
TC2	Handle missing/invalid data	Cleaned and validated dataset	Dataset had been validated
TC3	Train ML model	Model trains without error	Yes all model have been trained properly
TC4	Evaluate model metrics	F1-Score > 0.90, ROC-AUC > 0.95	Yes we are getting accurate score >0.90
TC5	Streamlit prediction input	Displays "Fraud" or "Legit" correctly	Yes the output display fraud or not as 0 and 1.
TC6	Database logging	Transaction recorded with timestamp	Yes timestamp is given in user enter details section.
TC7	Performance under load	Prediction generated in < 2 seconds	Sometime it take time if the uploaded dataset is large.

Future Enhancements

1. Deep Learning Integration

2. Automated Retraining

3. Real-Time Fraud Detection

4. Automated Alert System

5. API Integration for Banks

6. Visualization Dashboard Upgrade

7. Blockchain Integration

References / Bibliography

1 **Kaggle**

Credit Card Fraud Detection Dataset

2 **Streamlit.io**

Building Interactive ML Apps

3 **Power BI Documentation**

Data Visualization & Dashboards

Thank You

For Your Time And Attention.