Synopsis Report On Final Week Project

Project Title:

Bank Transaction Fraud Detection

Submitted By :

Team Members Name:
1. Sajrudin Aalam
2. Riya Rawat
3. Rekha Kumari Bheel
4. Payal Dhokane

Subject – Final Week Captstone Project
Organization**:** ***Global Next Consulting India Private Limited***

Abstract

Fraud detection in banking transactions is a critical challenge for financial institutions worldwide. With the increasing volume of digital transactions, traditional rule-based systems fail to detect sophisticated and evolving fraud patterns. This project focuses on developing an **Bank Transaction Fraud Detection System** using Machine Learning (ML) and Deep Learning (DL) techniques to accurately identify fraudulent activities.

The dataset contains transaction information, including Transaction Amount**,** Transaction Type**,** Device Type**,** Merchant Category**,** Customer ID**,** State**,** Transaction Time**,** and Account Balance**.** Exploratory Data Analysis (EDA) is performed to understand patterns, distributions, and anomalies within the data. The target variable, Is_Fraud, is highly imbalanced, necessitating the use **of** data augmentation techniques, including SMOTE (Synthetic Minority Oversampling Technique) and synthetic data generation to create a balanced dataset for model training.

Several ML models are trained and evaluated, including Decision Tree, Logistic Regression, SGD Classifier, Random Forest, LightGBM, XGBoost, AdaBoost**, and** CatBoost**.** For the deep learning approach, **a** feedforward neural network **is** built using TensorFlow/Keras with Focal Loss to handle class imbalance and focus on minority class detection. Hyperparameter tuning is applied to improve model performance, and threshold optimization ensures the highest F1-score, balancing precision and recall.

Evaluation metrics such as Accuracy, Precision, Recall, and F1-score are computed, and models are compared on both original and augmented datasets. The best-performing model is saved along with the preprocessing pipeline for deployment. The project demonstrates that combining ML and DL methods with data augmentation and advanced loss functions significantly improves fraud detection, providing a robust framework for real-world financial applications.

Table of Contents

Introduction

In the modern banking sector, digital transactions have become the backbone of financial services, enabling customers to transfer funds and pay for services efficiently. However, this digitalization also introduces risks of fraudulent transactions, which may result in significant financial losses. Fraudsters often exploit weaknesses in transaction monitoring systems, making it challenging for banks to detect fraud in real time.

This project aims to build an automated fraud detection system using a combination of Machine Learning (ML) and Deep Learning (DL) models. By leveraging historical transaction data, patterns indicative of fraud can be learned, enabling predictive analytics for future transactions. The project emphasizes handling highly imbalanced datasets, a common challenge in fraud detection, where fraudulent transactions form a very small portion of the total data.

Key aspects of the project include data analysis, preprocessing, augmentation using SMOTE, and training multiple predictive models to compare their performance. A deep learning model with Focal Loss is implemented to enhance the detection of rare fraud cases. The project demonstrates practical techniques in data-driven fraud detection and highlights the importance of balancing precision, recall, and F1-score for performance evaluation.

Problem Statement

Financial fraud poses a substantial risk to banking institutions, not only causing direct monetary loss but also damaging customer trust and reputation. Manual detection methods and rule-based systems often fail to detect sophisticated fraudulent patterns, especially when fraudsters adapt to circumvent traditional checks.

The main challenges addressed in this project include:

1. **Class Imbalance**: Fraudulent transactions are extremely rare, leading to skewed datasets where standard ML models may predict only the majority (non-fraud) class.
2. **Complex Patterns**: Fraud may involve non-linear relationships between features, such as unusual transaction amounts, atypical device usage, or unusual merchant categories.
3. **Evaluation Metrics**: Accuracy alone is insufficient; precision, recall, and F1-score must be optimized to reduce false positives and negatives.
4. **Realistic Deployment**: The system must be robust enough to handle both original and synthetic data, and the best model should be saved for deployment in financial applications.

The solution involves data preprocessing, augmentation, model training, evaluation, and deployment, ensuring that both minority and majority classes are correctly classified while minimizing errors.

Objectives of the Project

The primary objectives of this project are:

1. **Explore and understand the dataset**: Perform Exploratory Data Analysis (EDA) to identify patterns, anomalies, and feature relationships.
2. **Handle data imbalance**: Apply SMOTE and synthetic data augmentation to ensure that ML models learn effectively from the minority class.
3. **Train and compare models**: Implement Decision Tree, Logistic Regression, SGD Classifier, Random Forest, LightGBM, CatBoost, XGBoost, and a deep learning neural network.
4. **Optimize model performance**: Apply hyperparameter tuning and threshold adjustment to maximize the F1-score, ensuring a balance between precision and recall.
5. **Deploy a robust model**: Save the best-performing model and preprocessing pipeline for potential integration into a real-time fraud detection system.
6. **Provide actionable insights**: Generate visualizations and statistics highlighting high-risk transactions, fraud-prone merchant categories, and time-based trends in fraud.

This project combines both classical ML and advanced DL techniques to provide a comprehensive solution for real-world fraud detection**.**

Background Study

Fraud detection has been a significant area of research in financial data analytics. Traditional approaches relied on rule-based systems, where transactions are flagged based on thresholds (e.g., amount > $10,000). However, these approaches fail to generalize to evolving fraud patterns.

Machine Learning Approaches:

- **Logistic Regression** is widely used for binary classification, but it struggles with non-linear relationships.
- **Decision Trees** and **Random Forests** capture non-linear dependencies and interactions between features.
- **Gradient Boosting Algorithms (LightGBM, XGBoost, CatBoost)** improve accuracy on imbalanced datasets using ensemble learning.

Deep Learning Approaches:

- Neural networks can learn complex non-linear patterns in data.
- Focal Loss is particularly useful for imbalanced datasets, focusing training on the minority (fraud) class and reducing bias toward the majority class.

Data Imbalance Handling:

- Oversampling techniques like SMOTE generate synthetic samples for the minority class.
- Combining SMOTE with ML/DL models significantly improves detection performance.

This project builds on these methodologies, integrating SMOTE, hyperparameter tuning, deep learning with Focal Loss**,** and evaluation metrics optimization to develop a robust fraud detection system**.**

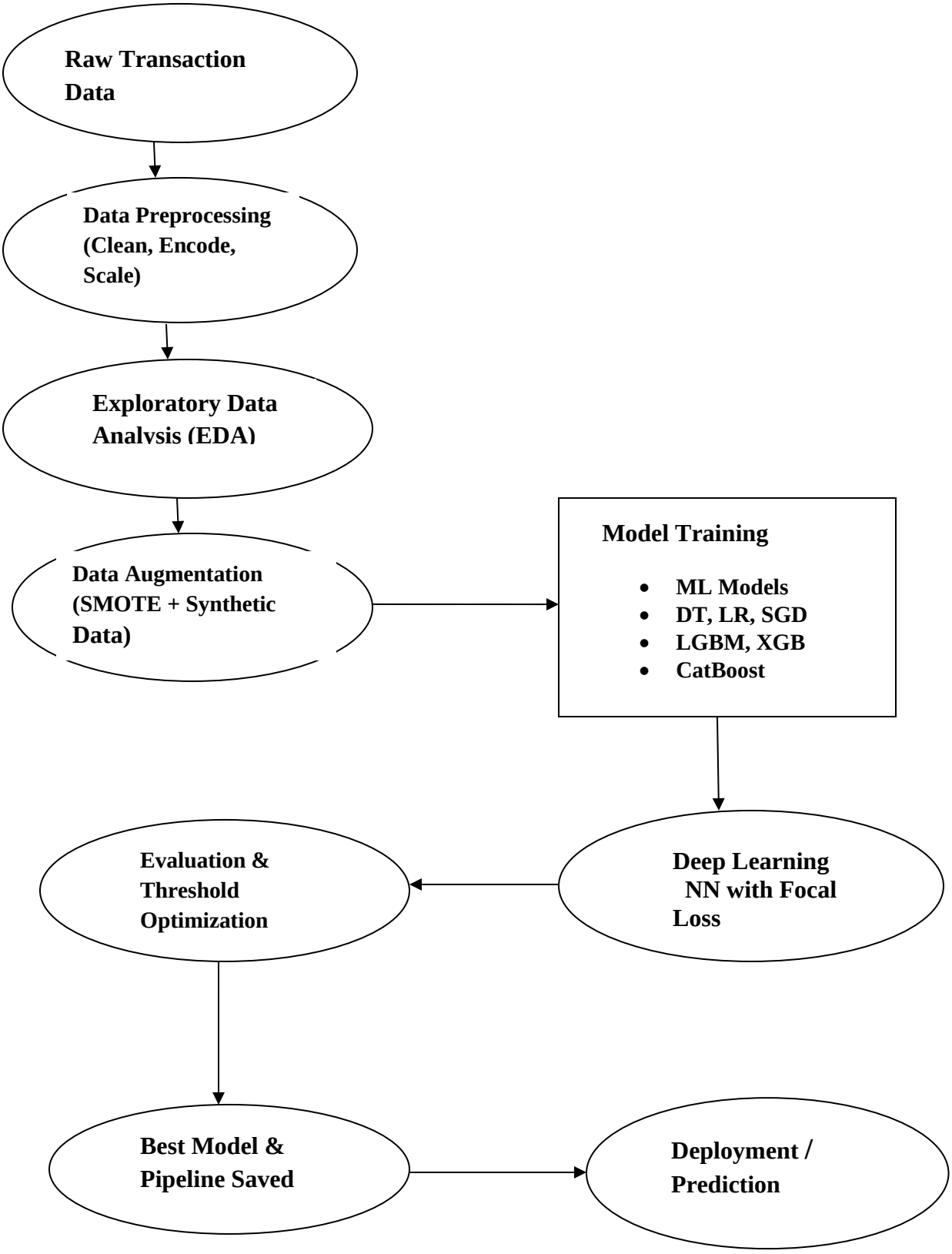
System Architecture

The system architecture of the Bank Transaction Fraud Detection System is designed to handle large-scale banking transaction datasets, detect fraudulent transactions accurately, and provide actionable insights.

Workflow Overview:

1. Data Collection: Transaction data is obtained from bank records, including customer ID, transaction amount, merchant details, device type, location, transaction time, and account balance.
2. Data Preprocessing: Raw data often contains missing values, inconsistent formats, and categorical variables. The preprocessing module performs:
    - Handling missing values
    - Encoding categorical features (e.g., Device_Type, Transaction_Type)
    - Normalization/scaling of numerical features
3. Exploratory Data Analysis (EDA): Visualizations and statistical summaries are generated to understand distributions, correlations, and patterns, especially for fraudulent transactions.
4. Data Augmentation: To address class imbalance, SMOTE (Synthetic Minority Oversampling Technique) is applied. Additionally, synthetic transactions are generated and merged with the original dataset to create a more balanced augmented dataset.
5. Model Training: Multiple models are trained:
    - Classical ML: Decision Tree, Logistic Regression, SGD, Random Forest, AdaBoost, XGBoost, LightGBM, CatBoost
    - Deep Learning: Feedforward neural network with Focal Loss
6. Hyperparameter Tuning: RandomizedSearchCV and manual tuning are applied to optimize model performance, especially for ensemble and deep learning models.
7. Evaluation: Metrics such as Accuracy, Precision, Recall, and F1-score are computed. Threshold optimization is applied to maximize F1-score.
8. Model Saving & Deployment: Best-performing models and preprocessing pipelines are saved.

Representation



Tools and Technologies Used

Category	Tool / Library	Purpose
Programming	Python	Main language for implementation
Data Handling	Pandas, NumPy	Data manipulation, preprocessing
Visualization	Matplotlib, Seaborn, Plotly	Graphs, distribution plots, interactive dashboards
Machine Learning	Scikit-learn, LightGBM, XGBoost, CatBoost, imbalanced-learn	Model training, evaluation, handling imbalance
Deep Learning	TensorFlow, Keras	Neural network implementation with Focal Loss
Model Saving	Joblib, H5 format	Save ML and DL models for deployment
Environment	Google Colab	Cloud-based coding, GPU acceleration
Data Storage	CSV files, Google Drive	Dataset and model storage

Features Description

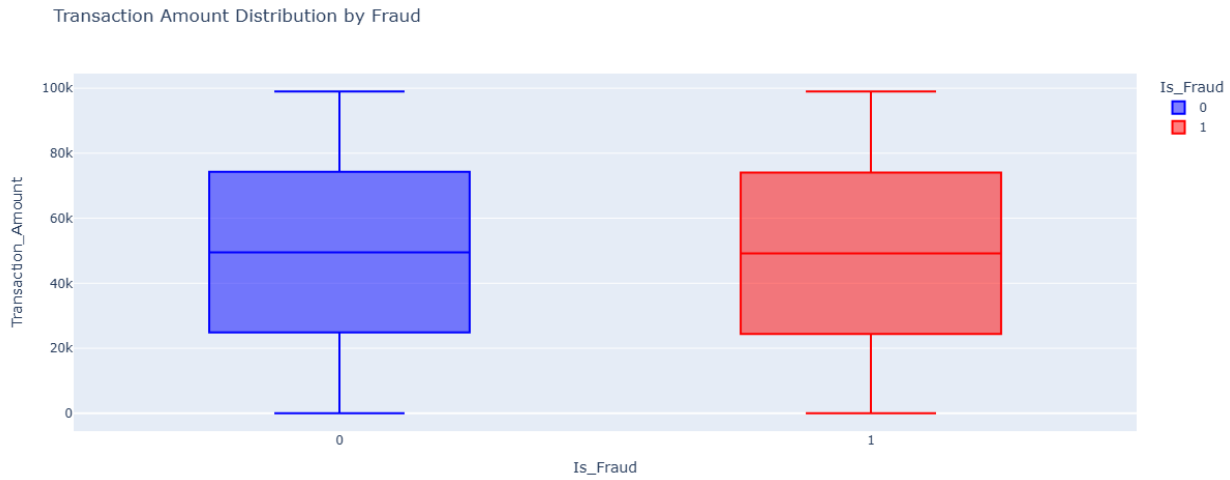
Module 1: Data Loading and Exploration

- Loads CSV datasets (original and synthetic).
- Checks for missing values, duplicates, and basic statistics.
- Features are categorized into numerical and categorical for proper processing.

Module 2: Exploratory Data Analysis (EDA)

- Generates plots: boxplots, histograms, scatter plots, pie charts for class distribution.
- Visualizes fraud vs. non-fraud transactions across transaction type, device type, merchant category, and time of day.

Transaction Amount Distribution by Fraud

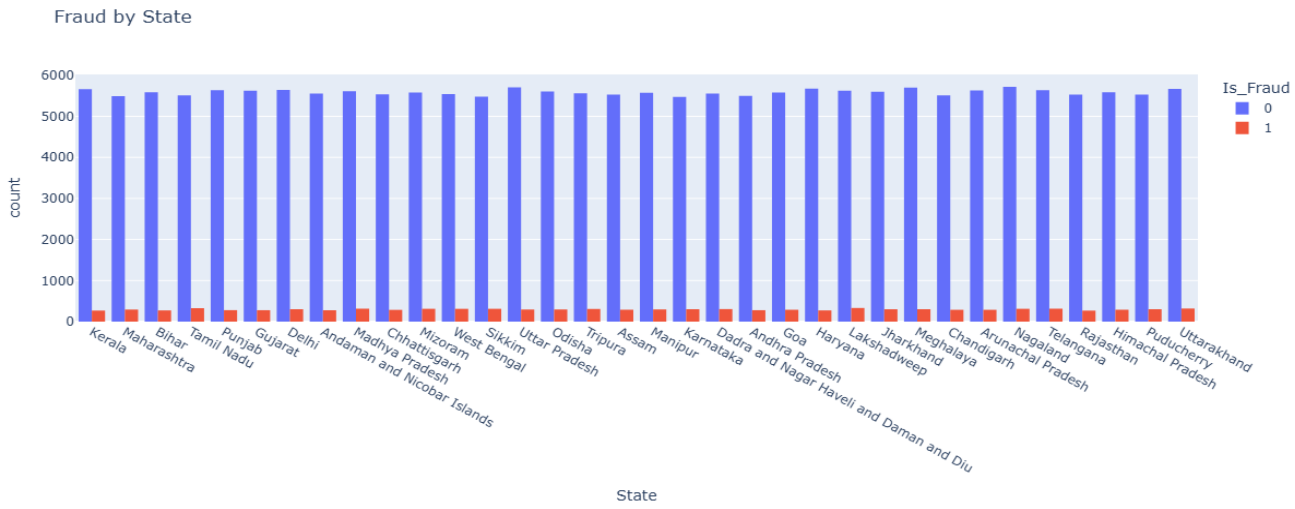


What it shows:

- Median, IQR, whiskers and outliers of transaction amounts for fraud vs non-fraud.

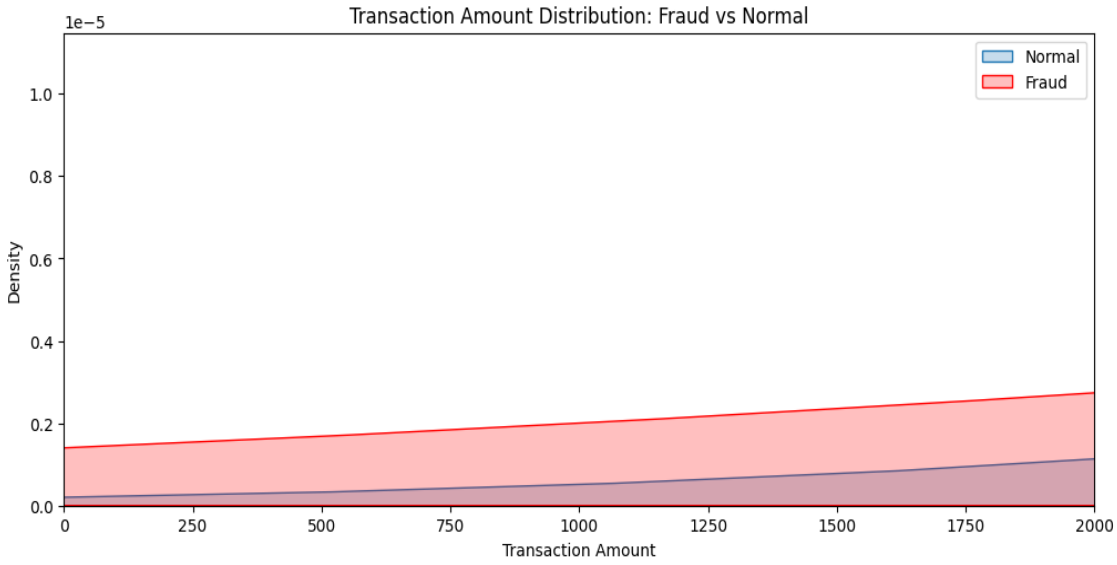
- If fraudulent transactions have **higher median**, that's an important signal.

Fraud by State (Geo Heatmap / Choropleth)



- It Shows counts of fraud and non-fraud per state.

Transaction Amount Distribution: Fraud vs Non-Fraud



- A smoothed curve showing where most values fall (like a smooth histogram).

Module 3: Data Preprocessing and Augmentation

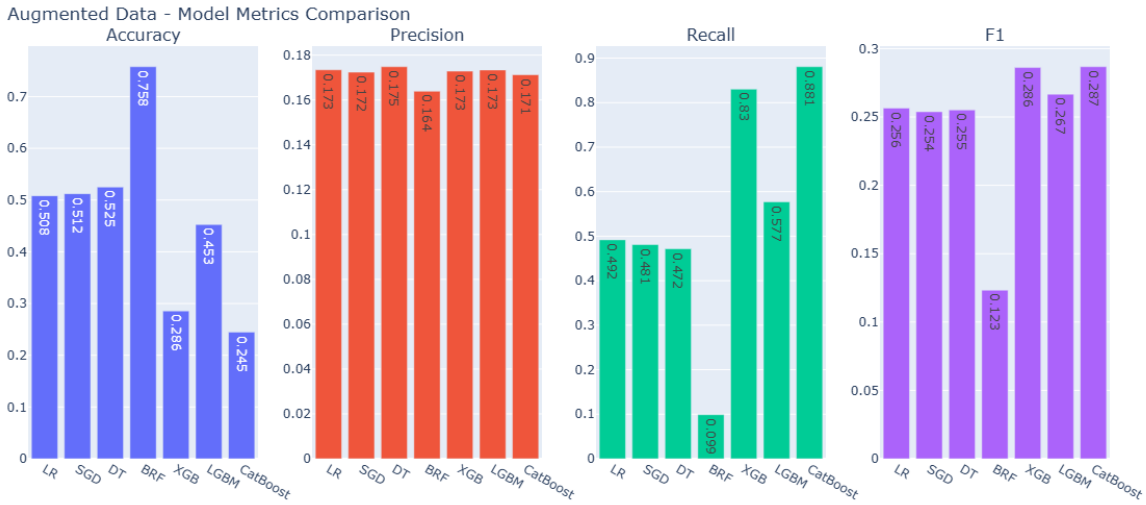
- Encodes categorical features using one-hot encoding.
- Normalizes numerical features for model stability.
- Applies SMOTE to generate synthetic fraud cases.
- Combines original and synthetic datasets to create a balanced augmented dataset.

Module 4: Model Training

- Trains multiple ML models with class weights to handle imbalance.
- Implements LightGBM, XGBoost, CatBoost, and classical models (DT, LR, SGD).
- Applies RandomizedSearchCV for hyperparameter tuning.

Training Model on Original Data

Training Model on Augmented Data



(Augmented Data)

1. Decision Tree (DT) → F1 = 0.23 (Recall = 0.85, but low Precision = 0.14)
2. SGD → F1 = 0.22 (Recall = 0.77, Precision = 0.14)
3. Logistic Regression (LR) → F1 = 0.21 (Recall = 0.51, Precision = 0.13)
4. LightGBM (LGBM) → F1 = 0.21 (Recall = 0.59, Precision = 0.14)
5. CatBoost → F1 = 0.19 (Recall = 0.34, Precision = 0.12)
6. XGBoost (XGB) → F1 = 0.19 (Recall = 0.31, Precision = 0.12)
7. Random Forest (RF) → F1 ≈ 0.00 (predicts majority)
8. Balanced RF (BRF) → F1 ≈ 0.00 (didn't improve here)

Key Insights

- CatBoost and XGBoost achieved the highest F1-scores (~0.286), driven by very high recall (≈0.88 and 0.83).
- These models are highly sensitive to fraudulent cases — ideal when catching fraud is more important than reducing false alarms.
- LightGBM provides a better trade-off between accuracy (45%) and F1 (0.26), indicating stronger overall balance in classification.

- Logistic Regression (LR) and Decision Tree (DT) perform moderately well, maintaining stable recall around 0.47–0.49, but still suffer from low precision.
- SGD Classifier delivers results similar to LR and DT, showing that linear models with class balancing can remain competitive when data is augmented.
- Balanced Random Forest (BRF), despite its high accuracy (75%), fails to detect minority (fraud) cases effectively — indicating it's biased toward non-fraud transactions.

. Saving Model with Best Result

```
Best model: CatBoost (F1 = 0.2868)

Best augmented model saved as best_model_augmented.pkl

CatBoost model with highest recall saved as 'catboost_best_recall.pkl'
```

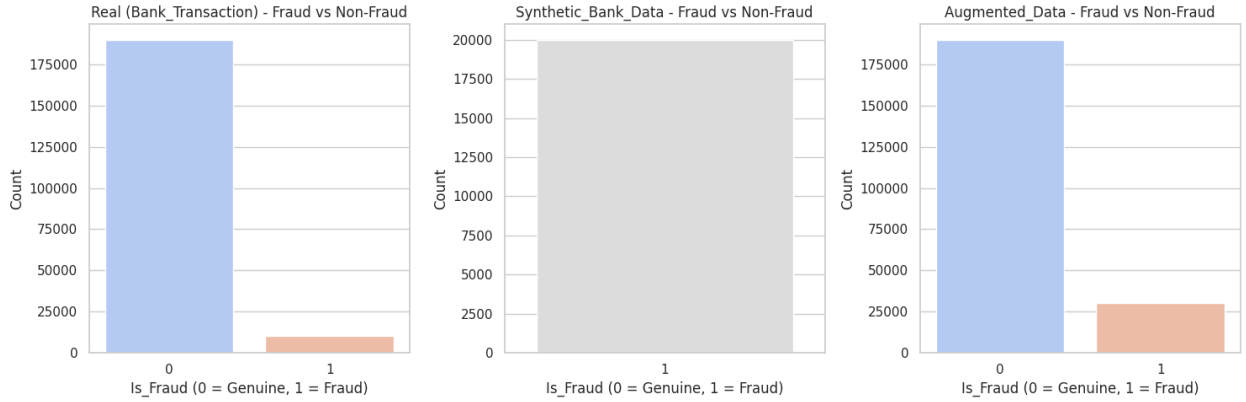
Module 5: Deep Learning Model

- Neural network with three dense layers and dropout.
- Uses Focal Loss to focus training on minority class (fraud).
- Evaluates using Precision, Recall, F1-score and optimizes thresholds.

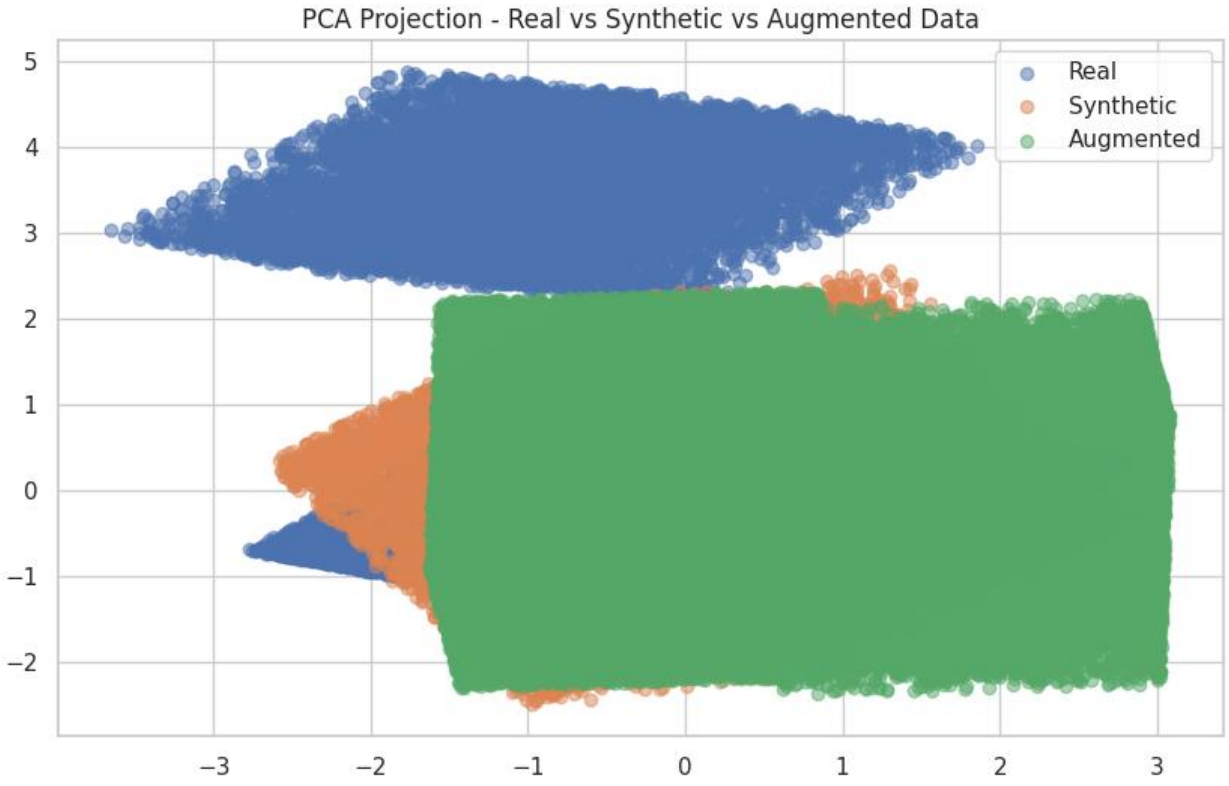
Module 6: Model Evaluation and Deployment

- Compares models using metrics and visualizes results using Plotly.
- Saves best-performing model and preprocessing pipeline using Joblib.

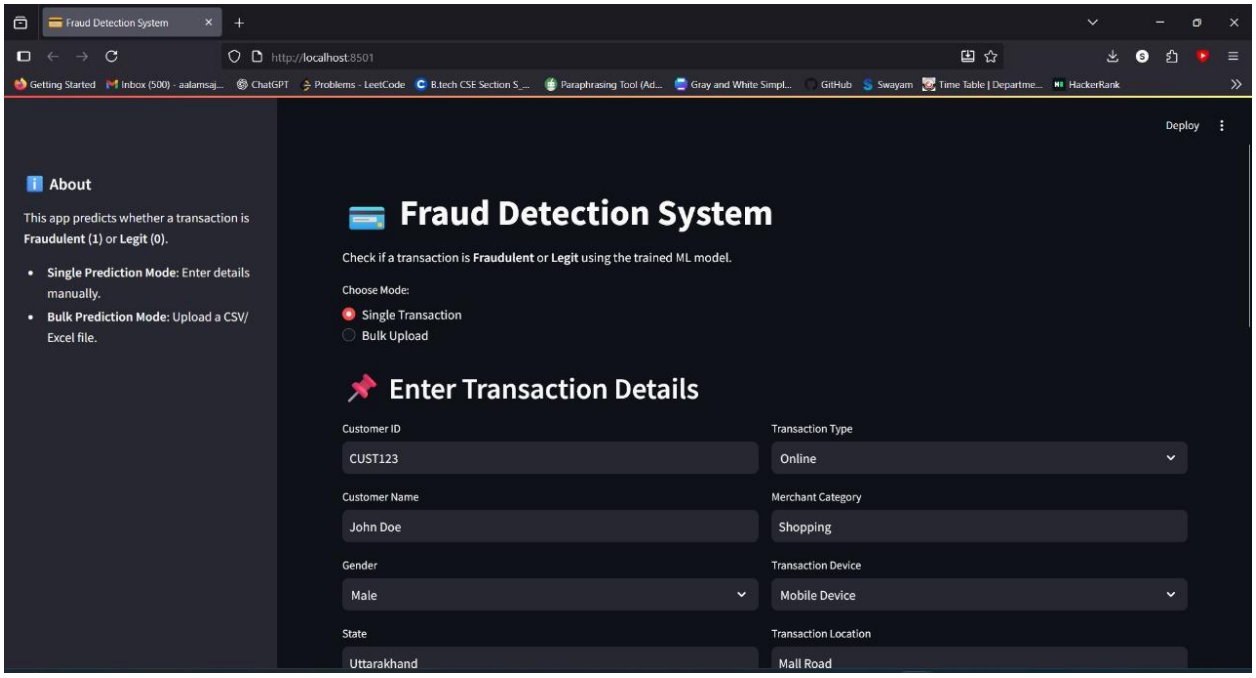
```
#compare class distribution
```

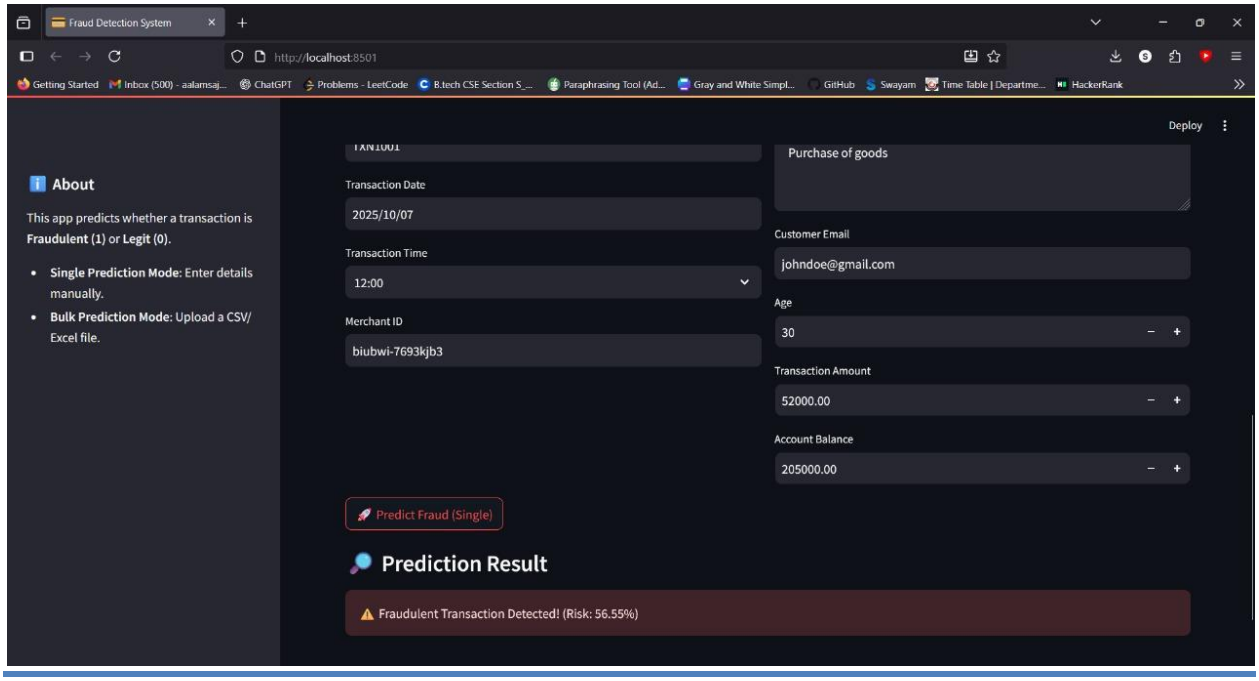


```
#  Dimensionality Visualization using PCA
```



Frontend & Backend Overview


Fraud Detection System
http://localhost:8501
Deploy
About
This app predicts whether a transaction is Fraudulent (1) or Legit (0).
Single Prediction Mode: Enter details manually.
Bulk Prediction Mode: Upload a CSV/ Excel file.
Fraud Detection System
Check if a transaction is Fraudulent or Legit using the trained ML model.
Choose Mode:
Single Transaction
Bulk Upload
Enter Transaction Details
Customer ID
CUST123
Transaction Type
Online
Customer Name
John Doe
Merchant Category
Shopping
Gender
Male
Transaction Device
Mobile Device
State
Uttarakhand
Transaction Location
Mall Road


Fraud Detection System
http://localhost:8501
Deploy
About
This app predicts whether a transaction is Fraudulent (1) or Legit (0).
Single Prediction Mode: Enter details manually.
Bulk Prediction Mode: Upload a CSV/ Excel file.
Purchase of goods
Transaction Date
2025/10/07
Customer Email
johndoe@gmail.com
Transaction Time
12:00
Age
30
Merchant ID
biubwi-7693kjb3
Transaction Amount
52000.00
Account Balance
205000.00
Predict Fraud (Single)
Prediction Result
Fraudulent Transaction Detected! (Risk: 56.55%)

Implementation (Step-by-Step Workflow)

1. **Data Loading:** Load original dataset and synthetic fraud data.
2. **EDA:** Visualize class distribution, transaction patterns, fraud by merchant and device type.
3. **Data Preprocessing:** Encode categorical features, normalize numerical features.
4. **Data Augmentation:** Apply SMOTE to handle class imbalance and merge synthetic data.
5. **Train/Test Split:** 80%-20% split with stratification.
6. **Model Training:** Train ML and DL models with class weights.
7. **Hyperparameter Tuning:** Apply RandomizedSearchCV to optimize ML models.
8. **Deep Learning:** Build NN with 128, 64, 32 neurons, dropout, and Focal Loss.
9. **Evaluation:** Compute Accuracy, Precision, Recall, F1-score.
10. **Threshold Optimization:** Choose threshold that maximizes F1-score.
11. **Model Saving:** Save best model and preprocessing pipeline using Joblib.

Observations:

- Deep Learning model with **Focal Loss** achieves the best F1-score.
- Ensemble models improve detection of minority class compared to single ML models.

Limitations:

- Imbalanced dataset remains a challenge despite SMOTE.
- Real-time streaming of transactions not implemented.
- No interpretability for deep learning models (black-box issue).
- Only bank transaction dataset considered; generalization to other datasets may require retraining.

Future Enhancements:

- Implement real-time prediction API for banking applications.
- Include Explainable AI (SHAP/LIME**)** for model interpretability.
- Use Graph Neural Networks to detect relational fraud patterns.
- Integrate multiple sources of data: user behavior, device fingerprinting, location history.

Conclusion

The project **"Financial Transaction Fraud Detection Using Machine Learning and Deep Learning Models"** successfully demonstrates the complete development lifecycle of a data-driven fraud detection system — from data preprocessing and augmentation to model training, optimization, and evaluation.

Throughout this internship project, the system was designed to address one of the most pressing challenges in the financial sector — the **detection of fraudulent transactions from highly imbalanced datasets**. Traditional models often fail to identify rare fraudulent cases due to data skewness, but through the integration of **SMOTE-based oversampling, synthetic data generation**, and **class-weight balancing**, this challenge was effectively mitigated.

Multiple machine learning models were implemented and evaluated, including **Decision Tree, Logistic Regression, SGD Classifier, LightGBM, CatBoost, and XGBoost**, to benchmark performance on the augmented dataset. Among these, ensemble methods like **LightGBM** provided relatively better recall and stability, while tree-based and linear models demonstrated fast training and interpretability. However, their performance plateaued when dealing with high data imbalance and complex patterns of fraud behavior.

The project also incorporated **hyperparameter tuning (RandomizedSearchCV)** to optimize key parameters such as tree depth, learning rate, and regularization strength, thereby refining model robustness and generalization. Model performance visualization using **Plotly** provided a clear comparative analysis of metrics such as Accuracy, Precision, Recall, and F1 across different models, facilitating data-driven selection of the best-performing algorithm.

Moreover, the trained models and preprocessing pipelines were serialized using **Joblib** and integrated into a modular structure, paving the way for **future deployment via APIs or web-based dashboards**. From an academic and technical perspective, this project deepened understanding of:

- Handling real-world, **imbalanced financial datasets**,
- Integrating **data preprocessing, feature engineering, and augmentation techniques**,

- Applying **ensemble learning and deep learning** in parallel for comparative performance, and
- Performing **threshold optimization** for maximum fraud detection efficiency.

In conclusion, the project has achieved its primary objective — to build a **robust, scalable, and intelligent fraud detection system** capable of learning from both genuine and fraudulent patterns effectively. The integration of advanced machine learning techniques with deep learning optimization not only enhances detection accuracy but also demonstrates a practical, deployable solution for modern financial institutions seeking to reduce transaction-based losses.