Report: Fraud Detection Models with Cost-Sensitive Learning and Financial Impact Analysis

**1. How Did We Proceed?**

To address the challenges posed by the highly imbalanced nature of fraud detection datasets, we employed several strategies:

**1.1 Cost-Sensitive Learning**

Fraudulent transactions are rare compared to legitimate ones, leading to class imbalance. To mitigate this issue, we implemented cost-sensitive learning techniques:

* **XGBoost** :
  + Used the **scale\_pos\_weight** parameter to adjust the weight of the positive class (fraudulent transactions) relative to the negative class.
  + Trained the model with parameters like **objective="binary:logistic"** and **eval\_metric="logloss"**.
  + Achieved improved recall for the minority class, demonstrating better detection of fraudulent transactions.
* **LightGBM** :
  + Enabled the **is\_unbalance=True** parameter to automatically adjust for class imbalance.
  + Demonstrated strong performance with minimal tuning, achieving high precision and recall for the majority class.
* **Bayesian Optimization for XGBoost** :
  + Used the **hyperopt** library to fine-tune hyperparameters such as **scale\_pos\_weight**, **max\_depth**, **learning\_rate**, and **n\_estimators**.
  + Bayesian optimization improved the ROC-AUC score and enhanced the model's ability to detect fraudulent transactions.

**1.2 Ensemble Methods**

To further improve performance, we combined multiple models using ensemble techniques:

* **Random Forest** :
  + Trained a Random Forest classifier with **class\_weight="balanced"** to handle class imbalance.
  + Evaluated its performance using metrics like precision, recall, and F1-score.
* **Voting Classifier** :
  + Combined the optimized XGBoost and Random Forest models into a Voting Classifier with soft voting.
  + Soft voting aggregates predicted probabilities from both models, resulting in better generalization and higher recall for the minority class.

**1.3 Margin Calculation**

To evaluate the financial impact of the models, we implemented a custom **calculate\_margin** function:

* **Definition of Margin** :
  + Margins were defined for different scenarios:
    - **True Negative (TN)** : A non-fraudulent transaction correctly accepted, generating profit.
    - **False Positive (FP)** : A non-fraudulent transaction incorrectly rejected, resulting in a loss.
    - **False Negative (FN)** : A fraudulent transaction incorrectly accepted, leading to significant losses.
    - **True Positive (TP)** : A fraudulent transaction correctly rejected, avoiding losses.
* **Threshold** :
  + A decision threshold determines how predicted probabilities are converted into binary predictions (fraudulent or non-fraudulent).
  + We iterated over a range of thresholds (e.g., 0.1 to 0.9) to find the threshold that maximized the total margin.
* **Final Margin** :
  + Applied the best threshold to convert predicted probabilities into binary predictions and calculated the final margin.

**2. Interpretation of Results**

**2.1 Key Findings**

* **Best Threshold with Ensemble Method** : 0.378
* **Best Total Margin with Ensemble Method** : €2,015,681.15
* **Best Threshold without Ensemble Method** : 0.704
* **Best Total Margin without Ensemble Method** : €2,008,401.96
* **Ensemble Model Performance** :
  + Precision: 0.99 (class 0), 0.32 (class 1)
  + Recall: 1.00 (class 0), 0.04 (class 1)
  + F1-Score: 0.99 (class 0), 0.08 (class 1)
  + Overall Accuracy: 0.99
* **Without Ensemble and optimization model performance**
  + Precision: 1.00 (class 0), 0.02 (class 1)
  + Recall: 0.68 (class 0), 0.67 (class 1)
  + F1-Score: 0.81 (class 0), 0.04 (class 1)
  + Overall Accuracy: 0.68
* **Without Ensemble and with Bayesian optimization model performance**
  + Precision: 0.99 (class 0), 0.10 (class 1)
  + Recall: 0.99 (class 0), 0.16 (class 1)
  + F1-Score: 0.99 (class 0), 0.12 (class 1)
  + Overall Accuracy: 0.98

Two notebooks were used for analysis:

1. **"With Ensemble Method Calculating Margin"** :
   * This notebook utilized an ensemble method combining XGBoost and Random Forest classifiers to maximize financial margins. The ensemble approach achieved a best threshold of 0.378 and a total margin of €2,015,681.15.
2. **"Without Ensemble Method Calculating Margin and Applying A/B Testing"** :
   * This notebook focused on individual models (XGBoost and LightGBM) and applied A/B testing to compare margins with and without fraud detection. It demonstrated the financial benefits of using fraud detection systems and highlighted the impact of varying fraud rates.

The ensemble model achieved high overall accuracy while balancing the trade-off between precision and recall for the minority class. The margin calculation confirmed that the model's decisions led to significant financial benefits.

**2.2 Comparing Margins for Fraud Detection Strategies (A/B Testing)**

We compared two scenarios to evaluate the financial impact of applying fraud detection:

1. **Group A (No Fraud Detection)** :
   * All transactions were accepted regardless of whether they were fraudulent or not.
   * **Margin** : €1,941,851.69
2. **Group B (With Fraud Detection)** :
   * The fraud detection model was applied to predict whether a transaction was fraudulent.
   * **Margin** : €1,982,455.21

**Conclusion from A/B Testing** :

* Applying the fraud detection model improved profitability by reducing losses due to fraud.
* The margin increased by €40,603.52 when using the fraud detection system.

**2.3 Simulating Transaction Fraud Rates and Calculating Margins**

To analyze the robustness of the model under varying fraud rates, we simulated scenarios with fraud rates of 1%, 5%, 10%, and 20%:

* **Fraud Rate: 1% → Simulated Margin: €2,109,040.78**
* **Fraud Rate: 5% → Simulated Margin: €1,274,532.98**
* **Fraud Rate: 10% → Simulated Margin: €242,417.24**
* **Fraud Rate: 20% → Simulated Margin: -€1,858,523.52**

**Interpretation** :

* As fraud rates increase, the importance of fraud detection becomes critical in mitigating financial losses.
* The model demonstrates robust performance at lower fraud rates but struggles to maintain profitability at higher fraud rates, highlighting the need for continuous improvement.

**3. Conclusion**

This project successfully implemented cost-sensitive learning and margin-based evaluation to address the challenges of fraud detection in imbalanced datasets.

By combining XGBoost, LightGBM, and ensemble methods, we developed robust models capable of detecting fraudulent transactions with high precision and financial impact. Future work could explore additional feature engineering, advanced ensemble techniques, and real-time deployment strategies to further enhance performance and scalability.