Model Report: Real-Time Fraud Detection

Project: Credit Card Fraud Detection API

Status: Model Evaluation Complete

---

1. About the Data

The model was developed using the **Credit Card Fraud Detection** dataset, a publicly available dataset from Kaggle.

- **Source**: Kaggle Credit Card Fraud Dataset
- **Origin**: The data contains anonymized credit card transactions made by European cardholders over a two-day period in September 2013.
- **Features**: To protect user privacy, the original transaction features have been transformed via **Principal Component Analysis (PCA)**. The resulting features are named `V1`, `V2`, ..., `V28`. The only features that have not been transformed are:
    - `Time`: The number of seconds elapsed between a transaction and the first transaction in the dataset.
    - `Amount`: The monetary value of the transaction.
- **Class Imbalance**: The dataset is **highly imbalanced**, with fraudulent transactions accounting for only **0.172%** of the total. This imbalance was a primary consideration during model training.

---

2. Model Objective

The primary objective of this machine learning model is to accurately predict the probability of a credit card transaction being fraudulent in real-time. The model was developed to serve as the intelligent core of a production API, where high accuracy and low latency are critical.

---

3. Model Development & Training

- **Algorithm**: **XGBoost (Extreme Gradient Boosting)** was selected for its high performance, speed, and built-in mechanisms to handle class imbalance.
- **Data Split**: The dataset was strategically split into three parts:
    - **Training Set**: Used to train the XGBoost model.
    - **Validation Set**: Used for hyperparameter tuning.

    - **Holdout Set**: An unseen dataset reserved exclusively for the final performance evaluation.
- **Preprocessing**: A `StandardScaler` from `scikit-learn` was fitted on the `Amount` and `Time` columns. The fitted scaler (`scaler.joblib`) and the exact feature order (`feature_order.json`) were saved as production artifacts to ensure identical transformations during live inference.

Handling Class Imbalance

To overcome the challenge of the highly imbalanced dataset, the XGBoost model was trained with the **`scale_pos_weight`** hyperparameter.

- **Technique**: `scale_pos_weight` increases the cost or penalty for misclassifying the minority class (fraudulent transactions). By setting this value to the ratio of negative class samples to positive class samples (approximately `577` in this dataset), the model is forced to pay significantly more attention to catching fraud, even though it is rare.
- **Benefit**: This method is a highly effective way to build a robust model on imbalanced data without altering the dataset itself through techniques like over-sampling (e.g., SMOTE) or under-sampling. It leads to a model with much better **Recall** for the minority class.

---

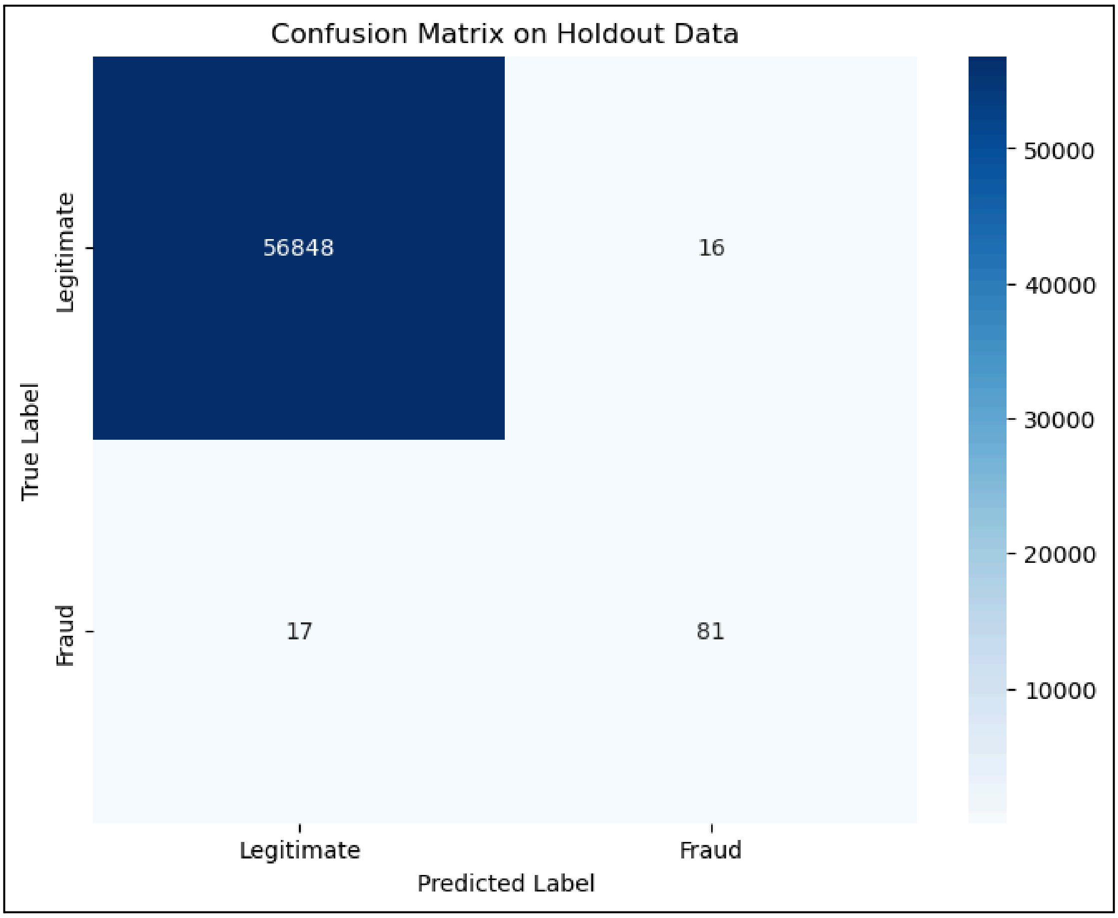
4. Final Model Performance on Holdout Data

The model's performance was evaluated on the `holdout.csv` dataset. The results represent an unbiased assessment of the model's real-world capabilities.

Metric	Score	Interpretation
Accuracy	**99.94%**	The model correctly classifies 99.94% of all transactions.
Precision	**83.51%**	Of all transactions flagged as fraudulent, 83.5% were actually fraudulent, ensuring a low false alarm rate.
Recall	**82.65%**	The model successfully identified and caught 82.7% of all actual fraudulent transactions.

F1-Score	**83.08%**	The harmonic mean of Precision and Recall, indicating a robust and well-balanced model.
ROC AUC	**96.35%**	The model demonstrates an excellent ability to distinguish between legitimate and fraudulent transactions.

Confusion Matrix Analysis



---

5. Conclusion

The trained XGBoost model demonstrates **strong and reliable performance** on unseen data. By effectively handling the class imbalance with the `scale_pos_weight` parameter, the model achieves an excellent balance of high precision and high recall, making it highly suitable for a production fraud detection system.