**Summary**

This project aims to build an accurate fraud detection system using real-world financial transaction data. By preprocessing the data (e.g., handling class imbalance using SMOTE) and training machine learning models like LightGBM, we achieved an F1-score of 0.94 in identifying fraudulent transactions. The project demonstrates the importance of feature engineering (e.g., balance differences) and effective class balancing in achieving high predictive accuracy.

**Introduction**

Fraud detection in financial transactions is a critical area of study that directly impacts the integrity and security of modern banking and payment systems. Fraudulent activities, such as unauthorized transfers or cash withdrawals, can result in significant financial losses for institutions and individuals alike. According to industry reports, financial fraud costs businesses billions of dollars annually, making it imperative to develop robust systems capable of identifying fraudulent transactions in real-time.

The challenge lies in the fact that fraudulent transactions are exceedingly rare, often making up less than 1% of total transactions in a dataset. This inherent class imbalance complicates the training process for machine learning models, which tend to favor the majority class unless special techniques are applied. Additionally, fraudulent behavior is highly dynamic, requiring models to adapt to evolving patterns and techniques used by fraudsters.

This project focuses on building an automated fraud detection system using machine learning techniques. By analyzing a dataset of financial transactions, the goal is to accurately classify transactions as fraudulent or non-fraudulent while addressing challenges such as class imbalance, feature engineering, and performance evaluation. Multiple machine learning models are explored, including LightGBM, Random Forest, and Support Vector Machines (SVM), and their performances are compared based on metrics such as precision, recall, F1-score, and accuracy.

**Dataset**

The dataset used for this project simulates real-world financial transactions, providing a robust foundation for building a fraud detection system. It contains detailed information about transactions, including account balances, transaction amounts, and transaction types. Below is a comprehensive breakdown of the dataset:

**Source and Attributes**

* **Source**: [Fraudulent Transactions Prediction](https://www.kaggle.com/datasets/vardhansiramdasu/fraudulent-transactions-prediction/datahttps:/www.kaggle.com/datasets/vardhansiramdasu/fraudulent-transactions-prediction/data).
* **Number of Records**: The dataset contains millions of rows, with each row representing a single transaction. A stratified subset of 100,000 records is used for experimentation.
* **Number of Features**: There are 11 features in the dataset, each contributing unique information about the transaction. Key features include:
  + **step**: Represents the time step of the transaction, which may correspond to hours or days since the start of the dataset.
  + **type**: Categorical variable representing the transaction type (e.g., CASH\_OUT, TRANSFER, PAYMENT, CASH\_IN).
  + **amount**: Numerical feature indicating the transaction amount.
  + **oldbalanceOrg and newbalanceOrig**: The originating account's balance before and after the transaction.
  + **oldbalanceDest and newbalanceDest**: The destination account's balance before and after the transaction.
  + **isFraud**: Target variable indicating whether a transaction is fraudulent (1) or not (0).
  + **isFlaggedFraud**: Indicates transactions flagged by the system for suspicious activity.

**Exploratory Data Analysis**

* Fraudulent transactions are rare in this dataset, creating a significant **class imbalance** where isFraud=1 constitutes less than 1% of the data. This poses a challenge for traditional machine learning models.
* The type feature shows that fraudulent transactions are most common in TRANSFER and CASH\_OUT types, aligning with real-world fraud trends.
* Many records have zero balances for the destination account (oldbalanceDest, newbalanceDest), indicating that feature engineering may be needed to extract meaningful patterns.

**Preprocessing Steps**

1. **Handling Class Imbalance**: The minority class (isFraud=1) is oversampled using **SMOTE**.
2. **Feature Encoding**: The type column is encoded using LabelEncoder to convert categorical values into numerical representations.
3. **Feature Scaling**: Numerical features (amount, oldbalanceOrg, etc.) are scaled using StandardScaler to ensure consistency and compatibility with machine learning models.
4. **Feature Selection**: Irrelevant columns (nameOrig, nameDest) are dropped to avoid potential feature leakage and reduce dimensionality.

**Methodology**

**Workflow Overview**

The project workflow includes the following stages:

**Data Preprocessing**:

* + Cleaning the data (e.g., removing missing values).
  + Encoding categorical variables and scaling numerical features.
  + Balancing the dataset using SMOTE to address class imbalance.

**Feature Engineering**:

* + Calculating balance differences (oldbalanceOrg - newbalanceOrig) and ratios (newbalanceOrig / oldbalanceOrg).
  + Adding transaction velocity (e.g., number of transactions per account in a given time window).

**Model Training and Selection**:

* + Training multiple models (LightGBM, Random Forest, SVM, Logistic Regression) and comparing their performance.
  + Using stratified train-test splits to ensure balanced distributions in training and testing datasets.

**Evaluation**:

* + Evaluating models using metrics like **precision**, **recall**, **F1-score**, and **accuracy**.
  + Analyzing results with confusion matrices and classification reports.

**Algorithms**

* **LightGBM**: A high-performance gradient boosting algorithm optimized for large datasets and imbalanced classes.
* **Random Forest**: An ensemble-based decision tree algorithm that excels at handling diverse feature types.
* **SVM**: A linear Support Vector Machine optimized for classification.
* **Logistic Regression**: A baseline linear classifier to evaluate model performance.

**Evaluation Metrics**

* **Precision**: Measures how many predicted fraud cases are truly fraudulent.
* **Recall**: Indicates how well the model identifies all actual fraud cases.
* **F1-Score**: Combines precision and recall, prioritizing balanced performance.
* **Confusion Matrix**: Visual representation of true positives, true negatives, false positives, and false negatives.

**Discussion**

**Key Findings**

* Fraudulent transactions are more prevalent in TRANSFER and CASH\_OUT types. This insight aligns with real-world fraud cases, where fraudsters often attempt to transfer funds to external accounts.
* Higher transaction amounts are more likely to be fraudulent, making the amount feature critical in identifying fraud.
* SMOTE significantly improved the model's performance by addressing the class imbalance.

**Challenges**

* **Class Imbalance**: Even after applying SMOTE, some models struggled with false positives due to the synthetic nature of the minority class.
* **Feature Engineering**: Features like balance differences and ratios improved performance, but additional domain-specific features could further enhance results.

**Conclusion**

This project successfully developed a fraud detection system using machine learning techniques. The LightGBM model achieved the highest performance, demonstrating its effectiveness in handling imbalanced datasets. Key takeaways include the importance of:

1. Addressing class imbalance using SMOTE.
2. Incorporating meaningful features (e.g., balance differences and transaction types).
3. Evaluating models with metrics tailored to imbalanced datasets (e.g., F1-score).

**Future Work**

Future iterations could focus on:

* Using advanced techniques like **deep learning** for feature extraction and prediction.
* Testing the system on real-world datasets to assess its generalizability.
* Improving interpretability using tools like **SHAP** or **LIME**.

**References**

* Kaggle Dataset: [Fraudulent Transactions Prediction](https://www.kaggle.com/datasets/vardhansiramdasu/fraudulent-transactions-prediction/datahttps:/www.kaggle.com/datasets/vardhansiramdasu/fraudulent-transactions-prediction/data).
* technical side: [YouTube course](https://www.youtube.com/watch?v=bY__YW-xknU&list=PLfFghEzKVmjsNtIRwErklMAN8nJmebB0I).