**Project Overview: Fraud Detection System**

1. **Introduction**

A *fraud transaction* is an unauthorized or deceptive financial activity, such as using stolen credit card details, identity theft, or falsified payments, intended to gain money or goods illegally. These transactions often appear as outliers compared to legitimate ones, exhibiting unusual patterns like high amounts, odd locations, or rapid frequency.

*Fraud detection* is the process of identifying and preventing such transactions using data analysis and machine learning. It involves analyzing transactional data for anomalies, patterns, or behaviors indicative of fraud (e.g., unusual spending, mismatched locations), often employing techniques like classification models (e.g., RandomForestClassifier), resampling for imbalanced data (e.g., SMOTETomek), and real-time monitoring to flag or block suspicious activities, minimizing financial losses while ensuring legitimate transactions proceed smoothly.

**What are we trying to find out?**  
The project aims to develop an effective fraud detection system by determining the best machine learning model to identify fraudulent transactions in a highly imbalanced dataset (0.56% fraud), focusing on maximizing recall and precision for the fraud class.

**What are we aiming to achieve?**  
Success is defined as deploying a model that accurately detects fraud (high recall to catch most fraud cases, high precision to minimize false positives) in real-time, with an F1-score above 0.90 on an imbalanced test set, ensuring practical usability for financial institutions with minimal operational disruption.

1. **Fraud detection**

The dataset was reduced by selecting data from 2019, focusing on the last quarter (November) due to potential fraud events. It was further narrowed by region, selecting the smallest region (250,000 rows), and irrelevant columns like CC number and date of birth were explored for useful data before being omitted. Over all 17 features were selected for the final modelling stage. The final dataset for modelling contains 347,746 after SMOTETomek model.

1. **What is the Likelihood of a Transaction Being Fraudulent?**

Criteria for Classifying a Transaction as "Fraudulent" in This Fraud Detection System:

* High-risk transaction category
* Unusually high transaction amount
* Abnormal transaction frequency
* Occurs during a high-risk time period

Only 0.56% of Transactions Are Identified as Fraudulent in the Dataset.

This Project Investigates the Factors Predicting Fraudulent Transactions:

Beyond the listed criteria, we will analyse the dataset to uncover patterns, such as location, issuer, or user behaviour, to understand and predict the likelihood of identifying fraudulent transactions.

1. **Data from the CSV**

The dataset consisted of 27 columns from two CSV files: credit\_card\_fraud.CSV and customers.CSV. The customers.CSV contained overlapping information with the main transaction’s dataset. They were merged using key identifiers such as trans\_date, trans\_num, and cc\_num, integrating transactional data with customer demographics. This structured approach enhances fraud detection by enabling a more comprehensive analysis of fraud patterns and key indicators.

No external data was used.

1. **Dataset Overview**

**Time Period:**

The original dataset spans transactions from January 1, 2019, to December 31, 2020. After reduction, the dataset focuses on transactions from November 1 to November 30, 2019, narrowing down to the 4th quarter of 2019 for analysis.

**Fraud Detection Target:**

The analysis focuses on detecting fraudulent transactions, with the target variable is\_fraud (0 = non-fraudulent, 1 = fraudulent).

**Key Influencing Factors**

Features like transaction category, amount (amt), transaction time, city population (city\_pop), and location show a strong correlation with fraud occurrence.

**Imbalance Challenge**

Fraudulent transactions make up only 0.56% of the dataset, requiring specialized techniques to balance the data during model training.

**Exploratory Data Analysis (EDA)**

Initial analysis included summary statistics for numerical variables (amt, city\_pop) and frequency distributions for categorical features (category).

Visualizations like histograms and bar plots helped identify trends and anomalies in the data.

Next, feature engineering and encoding will prepare the data, while SMOTE or Tomek Links will address class imbalance. Correlation analysis and advanced models will then enhance fraud detection.

**Data summery**

1. **Data Cleaning and Handling Missing Values**

**Raw Data Processing**

The dataset was filtered to focus on November 2019 (Q4), with new features like day and is\_weekend extracted from trans\_date to capture temporal fraud patterns.

**Outlier Handling**

EDA revealed extreme values in amt (max $16,498.31, mean $67.93) and city\_pop (max 2,504,700). While amt outliers were retained due to potential fraud relevance, city\_pop and amt were log-transformed to reduce skewness.

**Missing Values**

Key variables (amt, category, is\_fraud) had no significant missing data. Minor gaps in non-critical fields (job, dob) were negligible and left untreated.

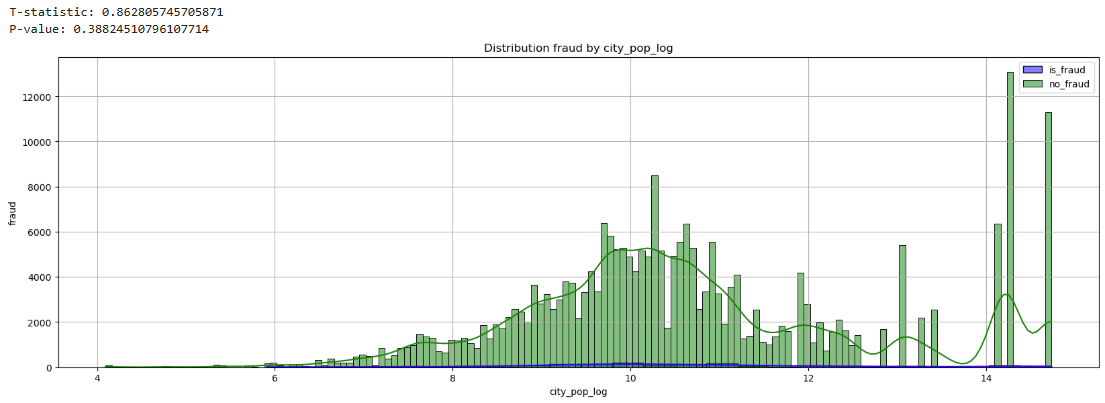
**T-test** - A t-test compares the means of two groups to determine if their differences are statistically significant. It was applied to is\_fraud against features like age, log-transformed city population, time category, day, and gender to identify meaningful patterns.

1. Fraudulent transactions are more common among younger individuals (30-40), while non-fraudulent transactions peak at 40-50. The overall age distribution is right-skewed (20-60). A low P-value (1.95e-11) confirms a significant difference in age, with fraud cases skewing younger. This suggests age may be a valuable feature for fraud detection.

A graph showing a line of green and blue

AI-generated content may be incorrect.

1. City population size does not significantly impact fraud occurrence (p-value = 0.388), as both fraud and non-fraud transactions peak at similar city\_pop\_log values (~10). Fraud remains rare overall. A possible issue with the city\_pop\_log scale should be clarified. More relevant fraud indicators may include transaction time, issuer, or category.



1. **Feature Engineering and Feature Selection**

**Creating new features from existing data:**

* Extracting information like prefixes, addresses, zip codes, etc.
* Extracting features from visual data.
* Geospatial Data.
* Extracting day, month, year, weekdays, holidays, and calculating time differences.
* Creating features like lags, moving averages, trends, and seasonality.

**Feature Selection –**

Eliminating irrelevant features is an essential part of feature selection, which helps improve model performance by removing noise, reducing overfitting, and speeding up training. In this project the following was used:

* **Lasso**: Shrinks coefficients of less important features to zero, effectively eliminating them for better model performance.
* **Ridge**: Reduces the magnitude of feature coefficients but doesn’t eliminate them. It helps manage multicollinearity.
* **LinearSVC**: A linear classifier that assigns zero weight to irrelevant features through regularization, helping with feature selection.
* **GradientBoostingClassifier**: Evaluates feature importance based on how much they improve model performance, allowing irrelevant features to be discarded.
* **RandomForestClassifier**: Ranks features by importance, making it easy to identify and remove irrelevant ones.

At the end of the process 17 features were selected, selecting variables with a sum of selections >= 4

1. **Model Development and Imbalance Handling**

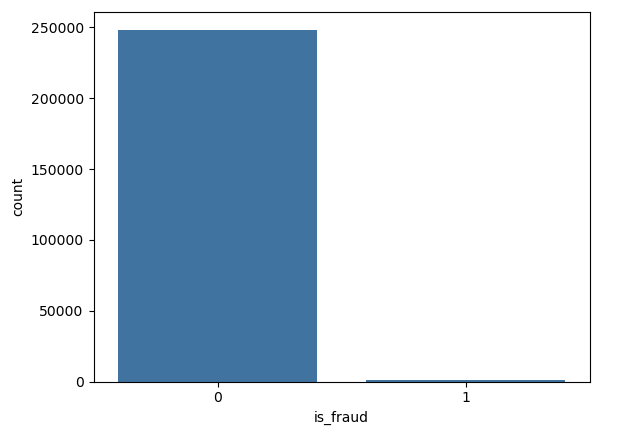
**Model Selection**

After completing the data preparation in earlier stages, we umade a split to train, test, and validation:

* Split into train+val and test sets (80% train+val, 20% test)
* Split train+val into train and val sets (75% train, 25% val from the train+val set)

**Imbalanced Data**

The target variable is is\_fraud, with only 0.56% of transactions being fraudulent. This severe class imbalance requires careful handling to maximize model accuracy and reduce errors in fraud detection.



**Imbalance Handling**

A RandomForestClassifier was trained on the imbalanced dataset. However, this approach failed to detect most fraudulent transactions, as shown by a low recall for the minority class.

Confusion Matrix: TP: 379, FP: 0, TN: 74518, FN: 38 (is\_fraud = 0 - 248391, 1 - 1391)

A diagram of different colored squares

AI-generated content may be incorrect.To address this, a combination of Under sampling and Oversampling was applied:

* **ROS (Random Over Sampling)**: Randomly duplicates instances from the minority class to balance the dataset.
* **RUS (Random Under Sampling)**: Randomly removes instances from the majority class to balance the dataset.
* **SMOTE (Synthetic Minority Over-sampling Technique)**: Generates synthetic samples for the minority class by interpolating between existing instances.
* **SMOTETomek**: Combines SMOTE with Tomek Links to generate synthetic samples and remove ambiguous instances, improving decision boundaries.

This approach aims to improve the model’s ability to detect fraudulent transactions while maintaining overall performance. The four models—SMOTE, SMOTETomek, RUS, and ROS—each have their strengths and weaknesses in handling imbalanced data for fraud detection. The **SMOTE model** performs well, with high accuracy, but its precision and recall for fraud detection are slightly lower compared to others. **SMOTETomek** improves upon SMOTE by achieving better precision and F1-Score for fraud detection, making it a more balanced model with fewer false positives. In contrast, the **RUS model** boosts recall significantly but at the cost of very low precision, resulting in a high number of false positives, which is not ideal for fraud detection. The **ROS model** strikes the best balance, offering both high precision and recall, leading to an excellent F1-Score and accuracy. Overall, **SMOTETomek** and **ROS** stand out as the most effective models for fraud detection, with **ROS** providing the best balance between precision and recall, while **SMOTETomek** offers a slightly better performance in precision.

Overall, SMOTETomek appears to be the most balanced and effective model.

**Final Model Implementation and Evaluation on the Fraud Detection Dataset**

**Dataset Overview**

The dataset used for this fraud detection project has been preprocessed and balanced using SMOTETomek, resulting in a dataset with 347,746 rows and 17 columns.

**Model Training**

Model training involved using multiple classifiers, including SVM, Logistic Regression, ADABoost, GBM, Decision Tree, and Random Forest. After performing hyperparameter tuning, the optimal parameters for the Random Forest classifier were identified as:

* Bootstrap = False
* max\_depth = 30
* min\_samples\_split = 5
* n\_estimators = 300
* random\_state = 1

The model was then evaluated on the test set (X\_test, y\_test).

**Conclusion**

The system includes validation steps like cross-validation and evaluation on a holdout test set to ensure robustness. However, the near-perfect validation performance (accuracy 0.9998) on a balanced set suggests potential overfitting or data leakage, however it can be explained due to the imbalanced process the dataset had to go through.

1. **Deployment of the Fraud Detection Model**

**Who It Benefits:**

This fraud detection algorithm is designed to benefit financial institutions, e-commerce platforms, and payment processors handling large transaction volumes. It is particularly valuable for organizations seeking to minimize financial losses due to fraudulent transactions while reducing the operational burden of investigating false positives. Stakeholders such as fraud analysts, risk management teams, and business owners will directly benefit from its implementation.

**Financial Institutions:**

Banks and credit card companies can reduce fraud-related losses by identifying 93% of fraudulent transactions, protecting customers and maintaining trust. The low false positive rate (4 out of 74,518 non-fraud cases) minimizes unnecessary holds or declines on legitimate transactions, improving customer experience.

**E-commerce Platforms:**

Online retailers can prevent chargebacks and fraudulent purchases, saving revenue and reducing operational costs. The system’s high precision ensures minimal disruption to genuine customer orders.

**Payment Processors:**

Companies like PayPal or Stripe can enhance their fraud detection capabilities, ensuring secure transactions for merchants and consumers while keeping investigation costs low due to fewer false positives.

**Fraud Analysts:**

The system prioritizes high-risk transactions for review, allowing analysts to focus on true fraud cases rather than sifting through numerous false positives, as seen with alternatives like RUS (1,568 false positives).

**How the System Works:**

It is designed to identify fraudulent transactions using a RandomForestClassifier with the SMOTETomek resampling technique to handle an imbalanced dataset.

**Here’s a concise overview of how the system operates:**

The fraud detection system processes a dataset with 17 selected features, using SMOTETomek to balance the highly imbalanced data (0.56% fraud). A RandomForestClassifier is trained on the resampled data, and hyperparameter tuning via RandomizedSearchCV. The system flags fraudulent transactions in real-time, achieving high recall (0.93046) and precision (0.98980) on an imbalanced test set (F1-score 0.95921). Validation on balanced and imbalanced sets ensures robustness, though potential overfitting or leakage is noted.

**Deployment and Operation:**

In practice, the system can be integrated into a transaction processing pipeline. For each incoming transaction, the system extracts the 17 features, preprocesses them, and feeds them into the trained model. The model outputs a prediction (is\_fraud = 1 for fraud, 0 for non-fraud), flagging suspicious transactions for further review while allowing legitimate ones to proceed. The system’s high recall ensures most fraud cases are caught, and its high precision minimizes unnecessary investigations, making it efficient for real-time use.

**Validation and Practical Considerations:**

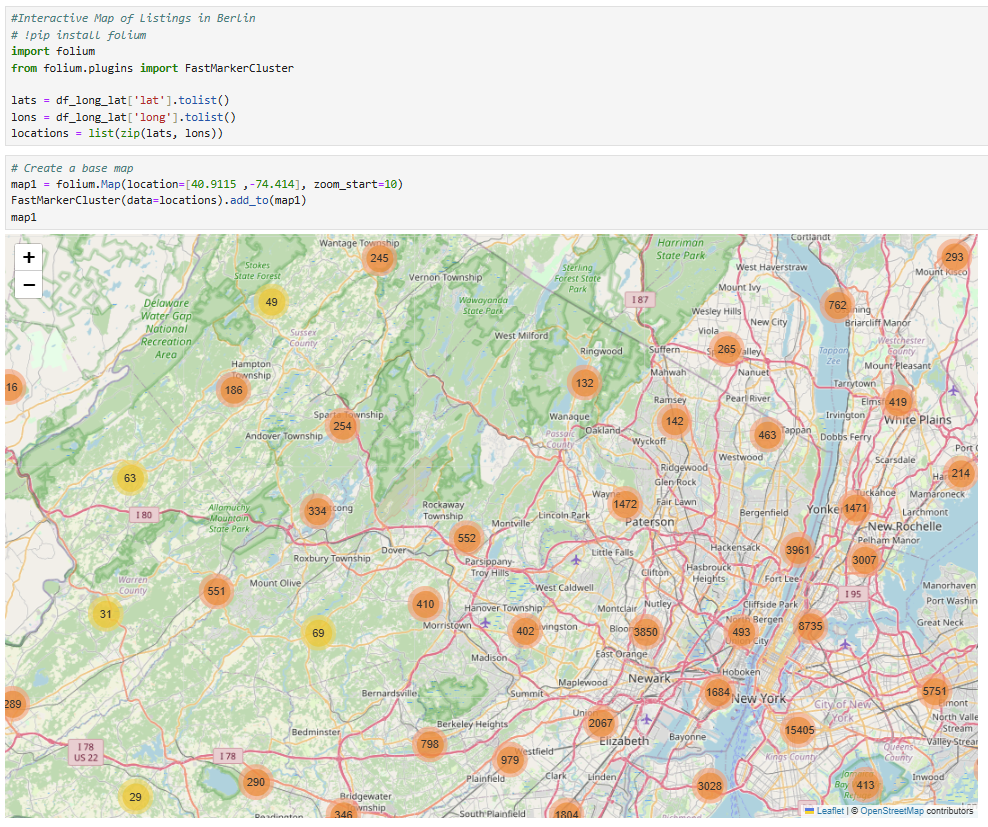
The system includes validation steps like cross-validation and evaluation on a holdout test set to ensure robustness. However, the near-perfect validation performance (accuracy 0.9998) on a balanced set suggests potential overfitting or data leakage, however it can be explained due to the imbalanced process the dataset had to go through.

In summary, the system works by preprocessing transactional data, balancing the training set with SMOTETomek, training a tuned RandomForestClassifier, and predicting fraud in real-time, offering a robust solution for fraud detection with a focus on high recall and precision.

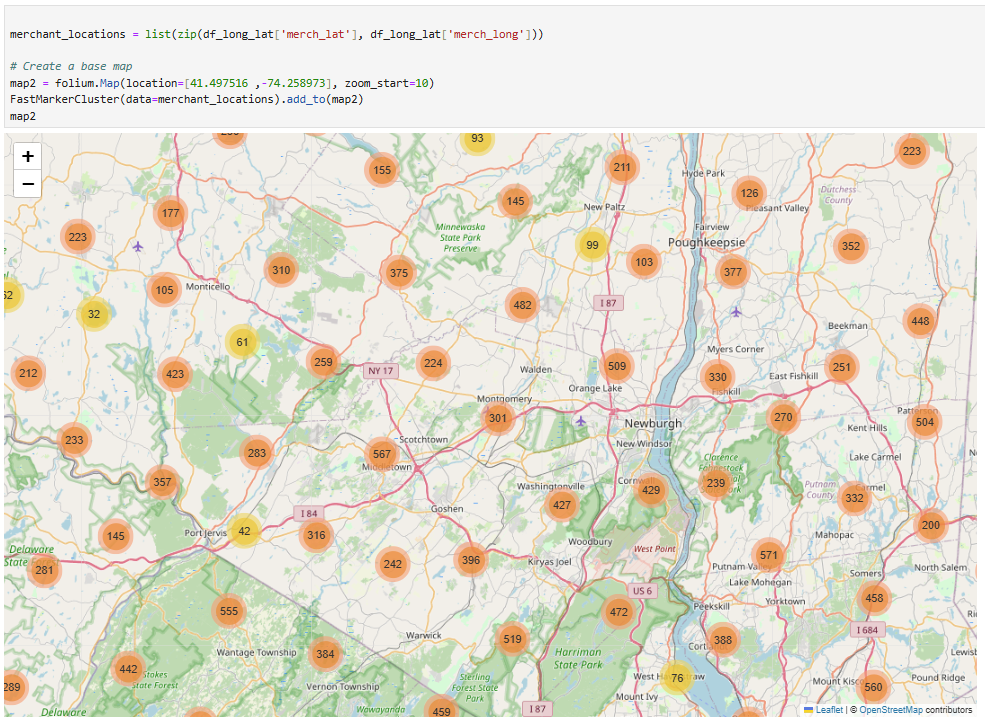
**Conclusion**

The deployment of this fraud detection model on cloud infrastructure will enable financial institutions to identify and prevent fraudulent transactions in real time, focusing on high-risk categories, times, and regions. By reducing false positives and improving detection accuracy, the model will enhance customer trust, reduce financial losses, and optimize fraud prevention efforts. Continuous monitoring and updates will ensure the model remains effective against evolving fraud patterns, making it a valuable tool for the financial industry.

Interactive Map – customers



Interactive Map – Merchant



Word Cloud –

A black background with green text

AI-generated content may be incorrect.