CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE, INDIA

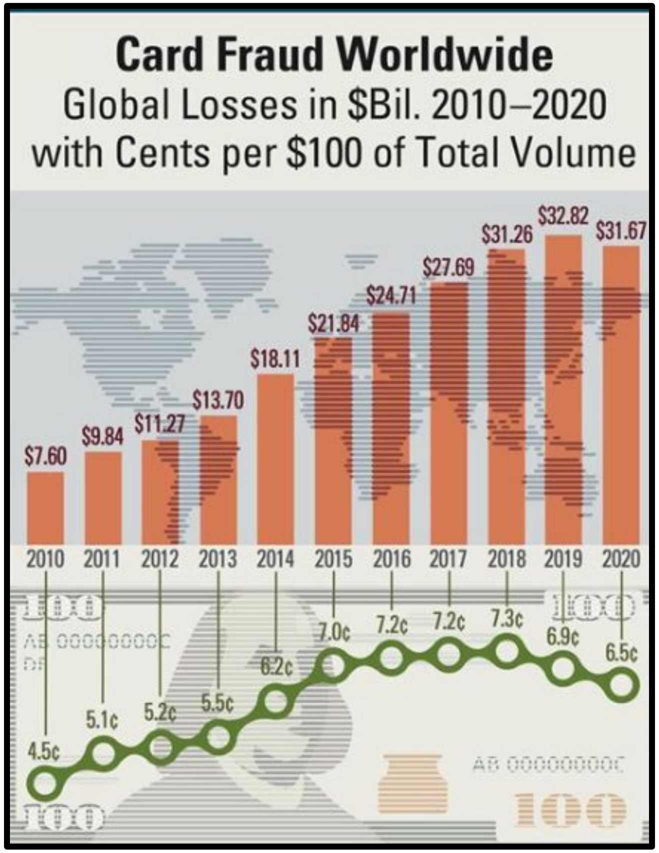
Christ(Deemed to be University)
Machine Learning
CAC 2 Project
on
Credit Card Fraud Detection Dataset

Submitted by
Annesha Naskar 2448306
Harshitha S 2448326
Kiran Guruv 2448333
Parameswaran A 2448343
Saranya M 2448356

Introduction:

Credit card fraud losses have surpassed **$32 billion**—and are still rising!

- With **digital transactions booming**, fraudsters are exploiting new vulnerabilities, threatening **banks and consumers**.
- **Traditional rule-based systems** struggle against rapidly evolving fraud tactics.
- **Machine Learning (ML)** offers a smarter, adaptive approach to detecting fraud.



Statistical Analysis

UNDERSTANDING THE DATA COLUMNS :

Total Rows : 1852394 ; Total Columns : 23

Numerical Type : trans_date_trans_time, cc_num, amt, lat, long, city_pop,dob, trans_num, merch_lat,merch_long, is_fraud, zip.

Categorical Type : merchant, category, first, last, gender, street, city, state,

Binary : is_fraud

KEY TAKEAWAYS:

1.The is_fraud column is the target variable, indicating whether a transaction is fraudulent or not.

2.The columns 'amt', 'city_pop', 'lat', 'long', 'merch_lat', 'merch_long', 'age'

requires scaling to ensure the better performance. And the Categorical values need to be encoded properly for machine learning models.

Transaction Trends by Central Tendencies : Mean

```
Mean of transaction_count: 626.4782
Mean of amt_deviation: 1.0000
Mean of amt: 70.2753
```

1.From Mean the mean transaction amount is found to be 626.
2.The deviation pattern can be found using the mean value of deviated amount to identify the fraudulent transaction.
3.The mean of the amount is used to identify the average transaction amount.

Median

```
Median of transaction_count: 661.0000
Median of amt_deviation: 0.6699
Median of amt: 47.5000
```

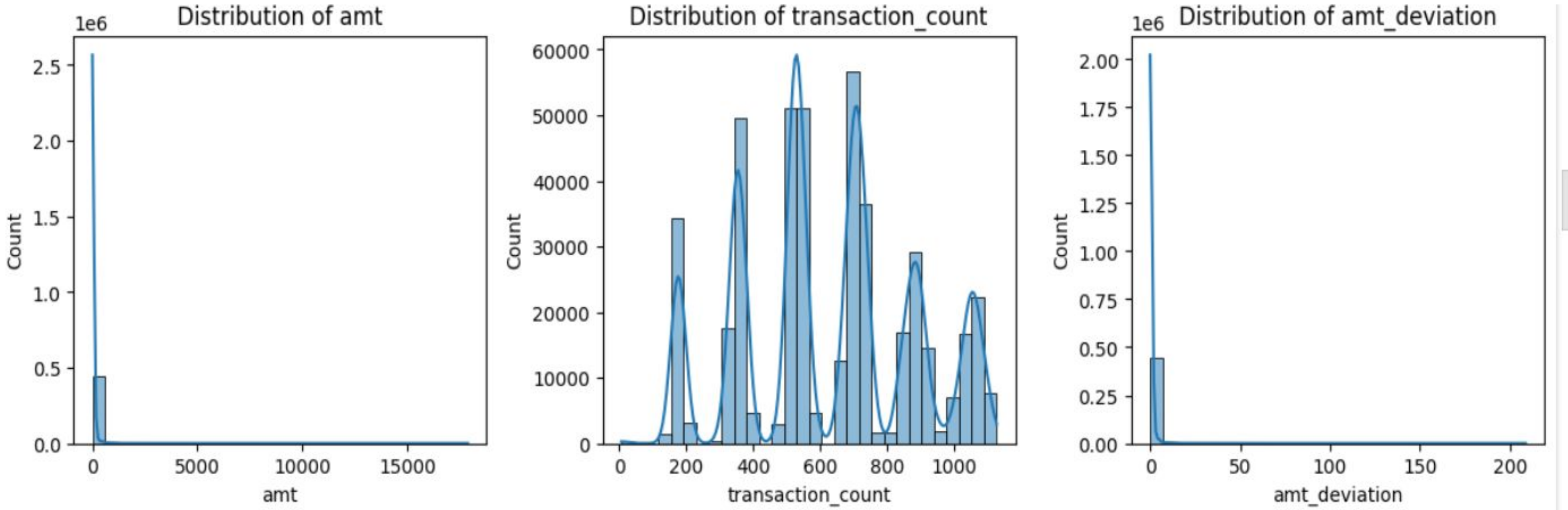
1. The median of the transaction_count implies that there are 661 transactions by 50% of people (or may be fewer than this), which is greater than the mean value (626.47) implies left skewed distribution means some customers have very low transaction counts.
2. The Low median value of the amt_deviation shows that most transactions are close to the expected value.
3. The median of amt says that half of the transactions is 47.50 or less and the mean(70.28) which is higher than the median means that it is a right skewed distribution.

Mode

```
Mode of category: gas_transport
Mode of merchant: fraud_Kilback LLC
Mode of gender: F
Mode of state: TX
```

1. Most transactions falls under gas_transport.
2. Since the most transactions appeared with merchant fraud_kilback LLC this might be fraudulent activity or business with high transaction frequency.
3. More transactions are associated with female more than Males.
4. The majority of transactions occurred in Texas (TX).

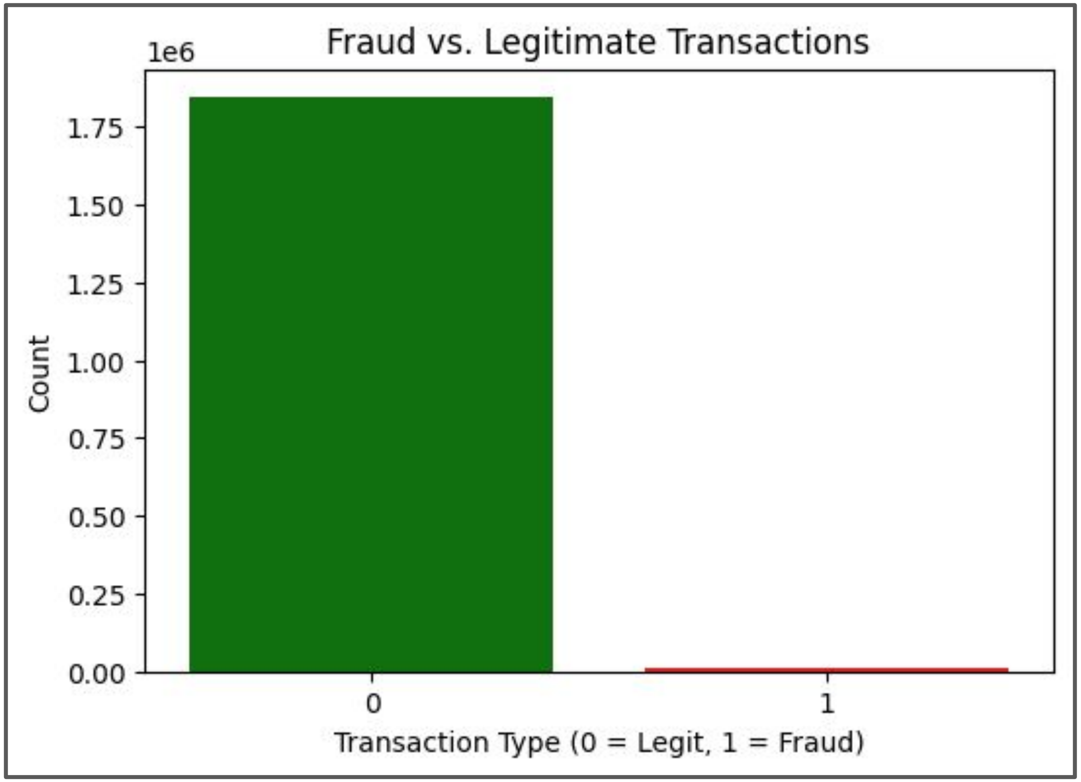
Visualizations on Data Distributions



1. Transaction_count multimodal [No significant skewness is observed]
2. Amt positively skewed [Implies potential outliers]
3. amt_deviation positively skewed [Most values are close to zero, with a few large outliers, indicating rare but significant spending deviations.

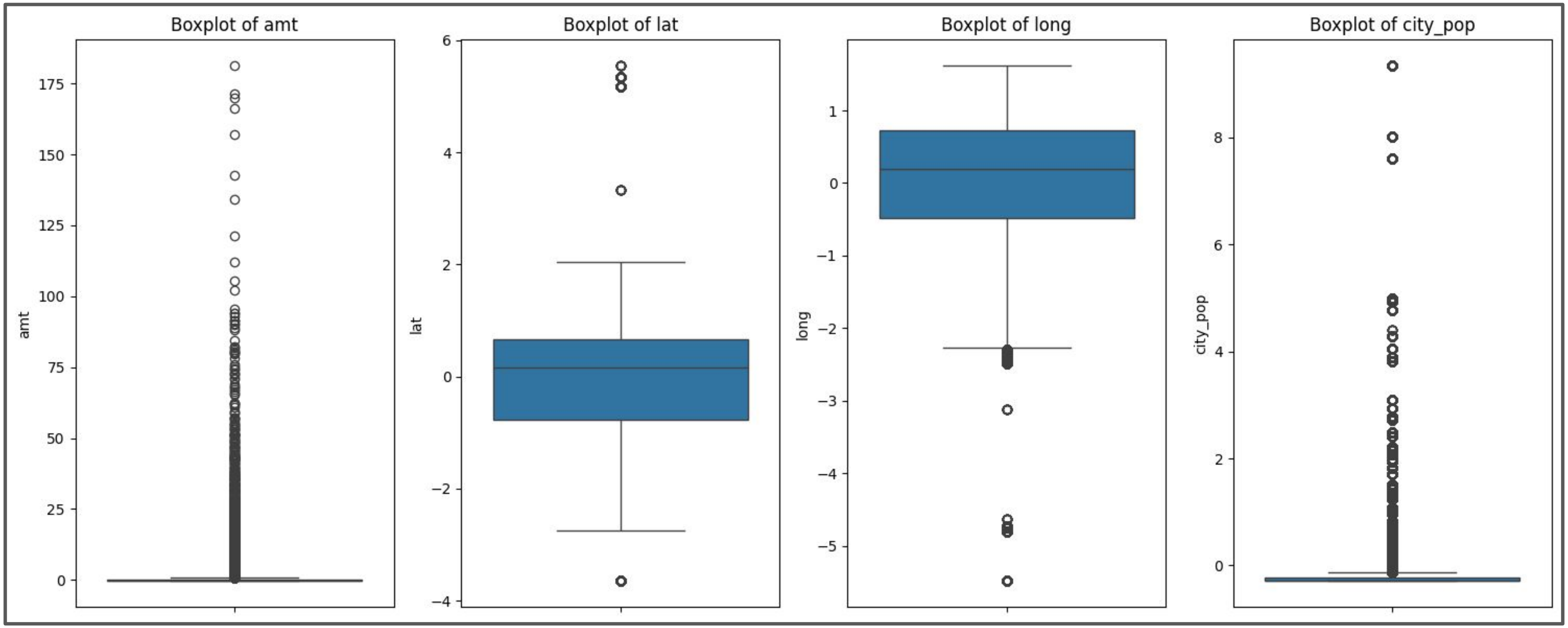
Exploratory Data Analysis

Before moving onto EDA, we preprocess the data. Dropping column 'Unnamed: 0',converting column 'trans_date_trans_time' to datetime format and extracting the important time based information. Then we extract 'age' from column 'dob'. Then we drop the missing values, perform Label Encoding for the categorical features 'merchant', 'category', 'gender', 'state', 'job' and scale the numerical features.

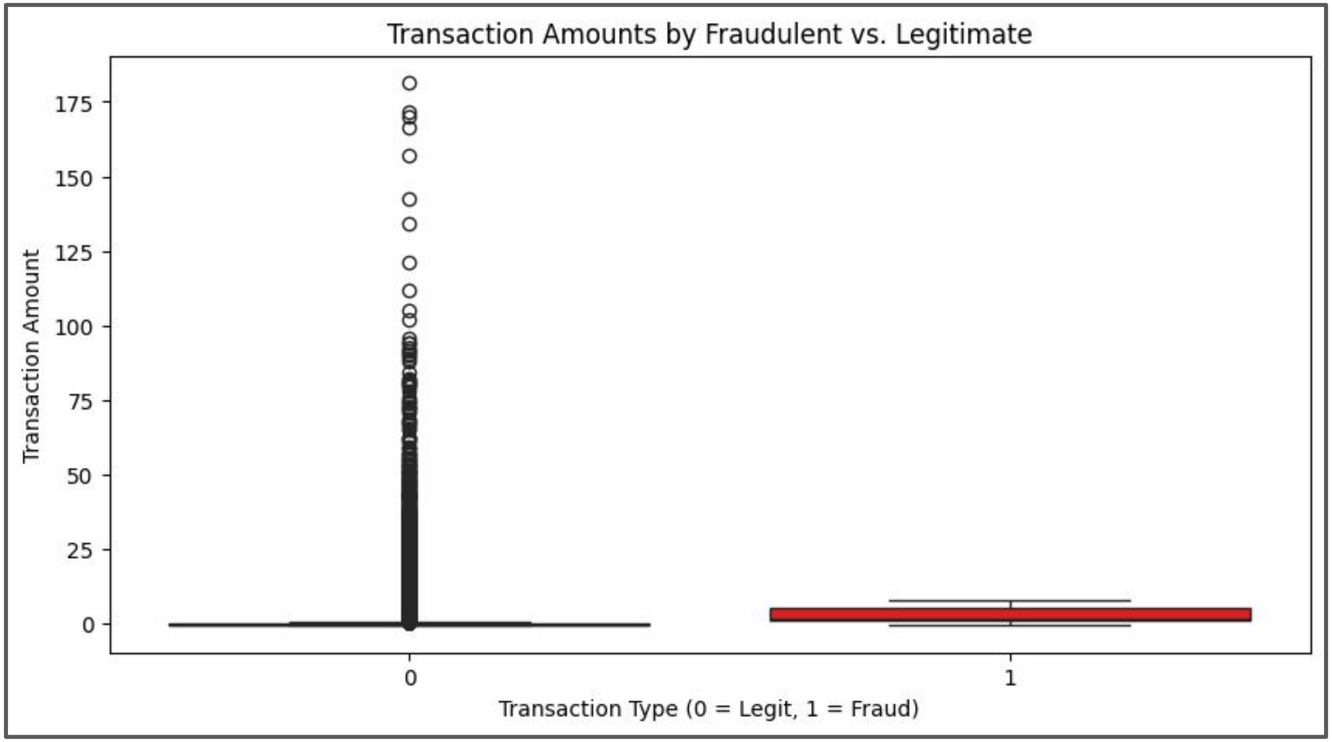


We can see that the dataset is imbalanced where most of the data denotes legitimate transaction.

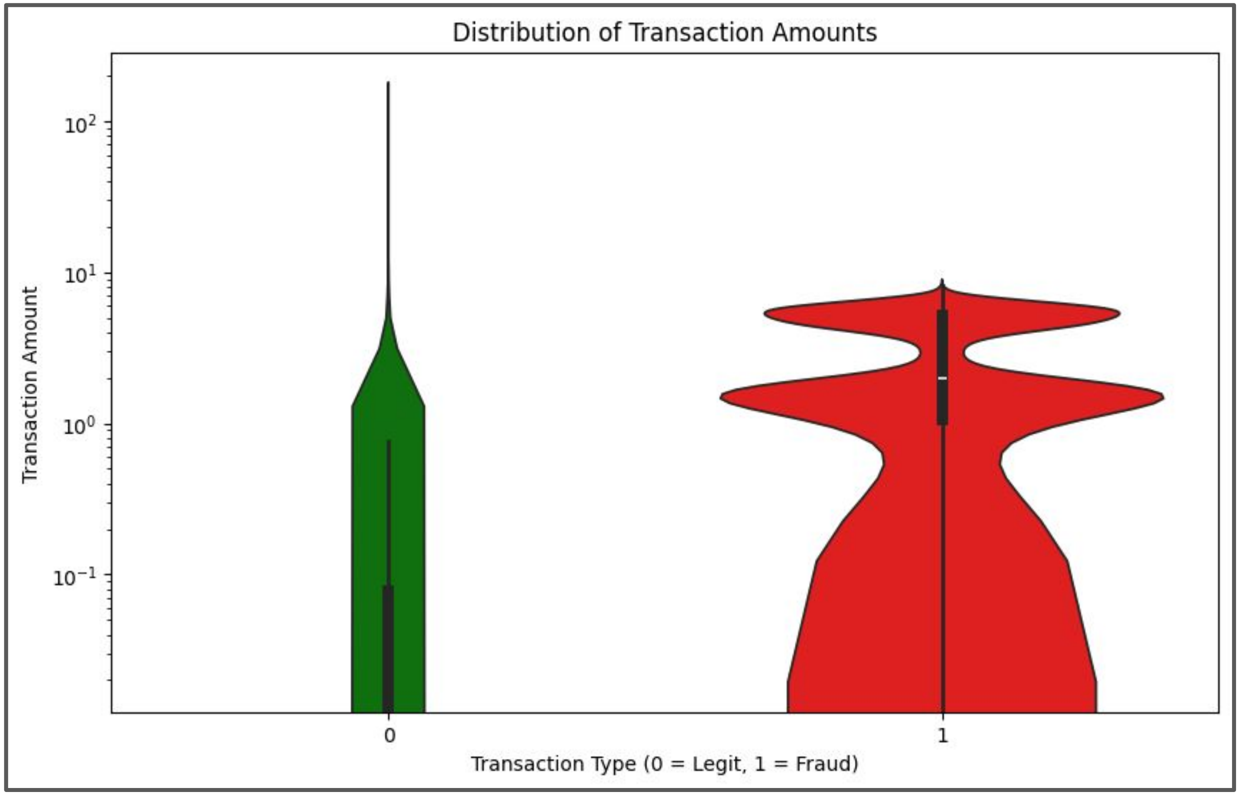
The bulk of transactions are low value, but there are numerous high value outliers. These high value transactions could be legitimate business transactions, but they might also contain fraudulent transactions, as fraudsters often try large amounts in a few attempts.
The latitude and longitude outliers suggest that some transactions occur in unexpected locations.
Most transactions occur in low population areas, but a significant number of outliers belong to highly populated cities.



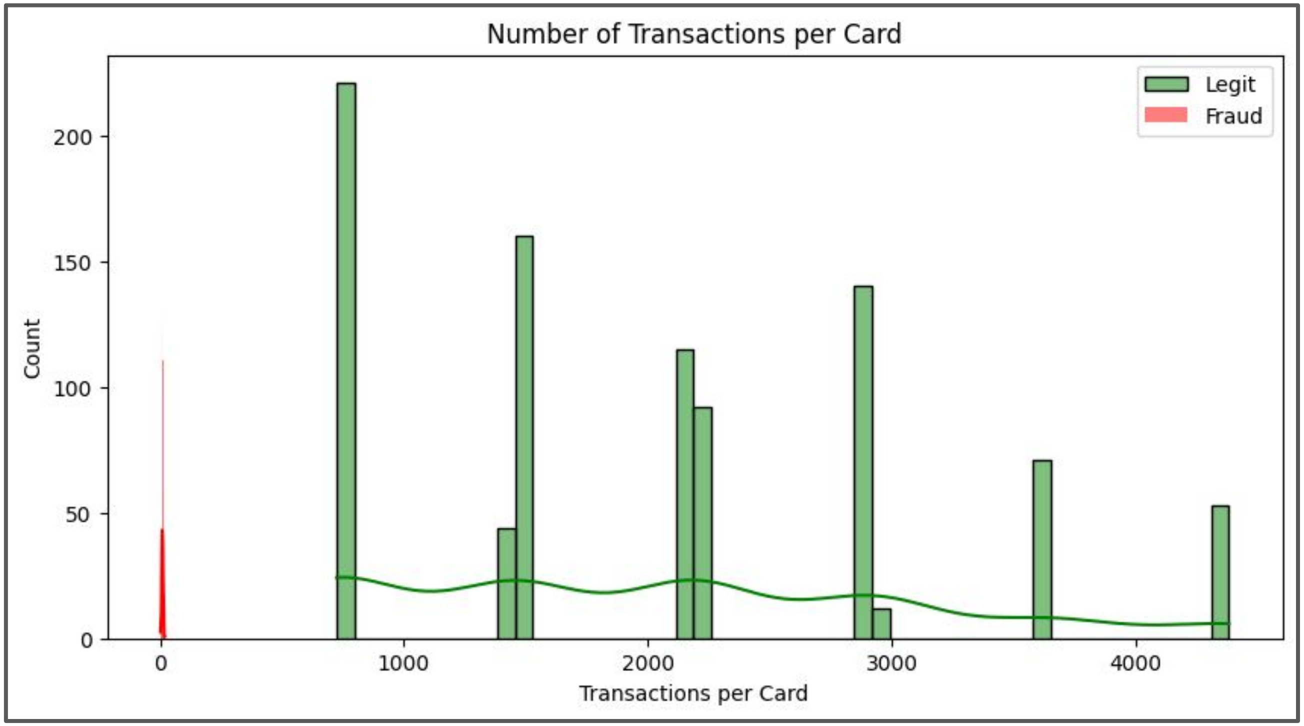
Fraudulent transactions tend to involve moderate amounts and do not show extreme outliers.
While legitimate transactions can vary significantly, including very large amounts.
This could indicate that fraudsters avoid very high amounts to stay undetected.



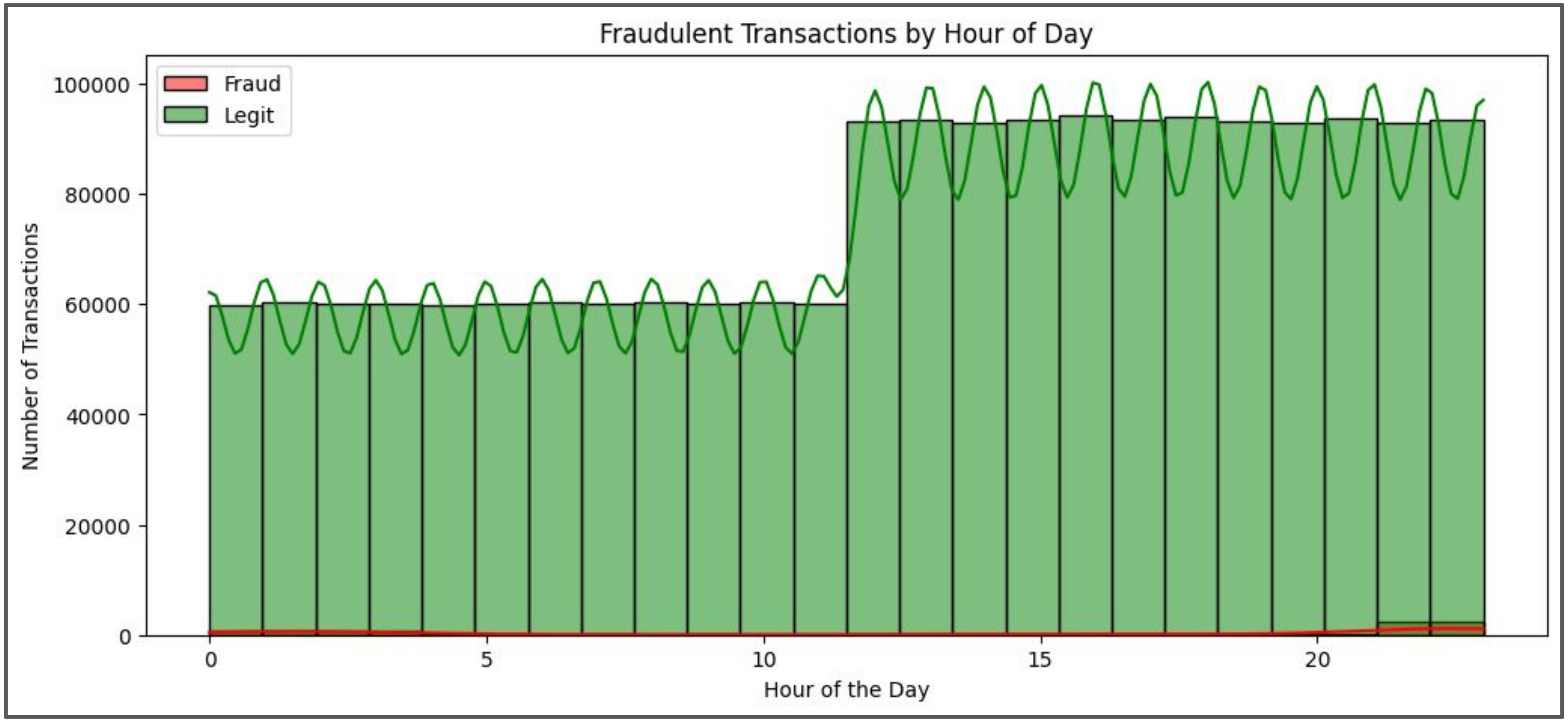
Compared to the boxplot this shows how the transactions are distributed.
The legitimate transaction distribution is heavily skewed towards very small transaction amounts and there are very few large transactions, but they do exist as seen in the long upper tail.
The fraudulent transaction distribution is more spread out compared to legitimate transactions and the plot suggests multiple peaks, indicating that fraudulent transactions may follow distinct patterns.



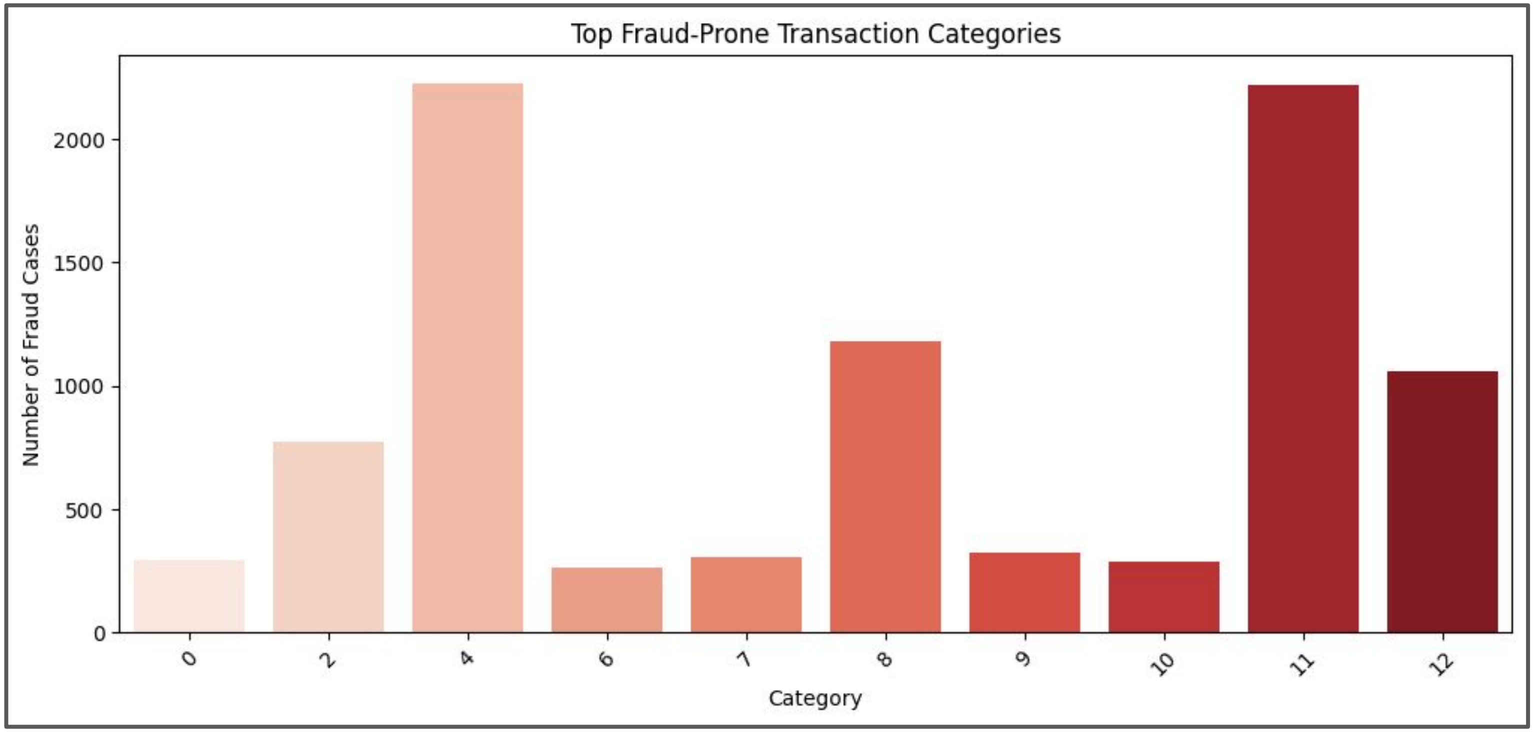
Legitimate cards tend to have more frequent transactions, which vary widely in volume. Fraudulent transactions are much more likely to occur on cards with very few transactions, which denotes that the fraudsters probably target rarely used cards



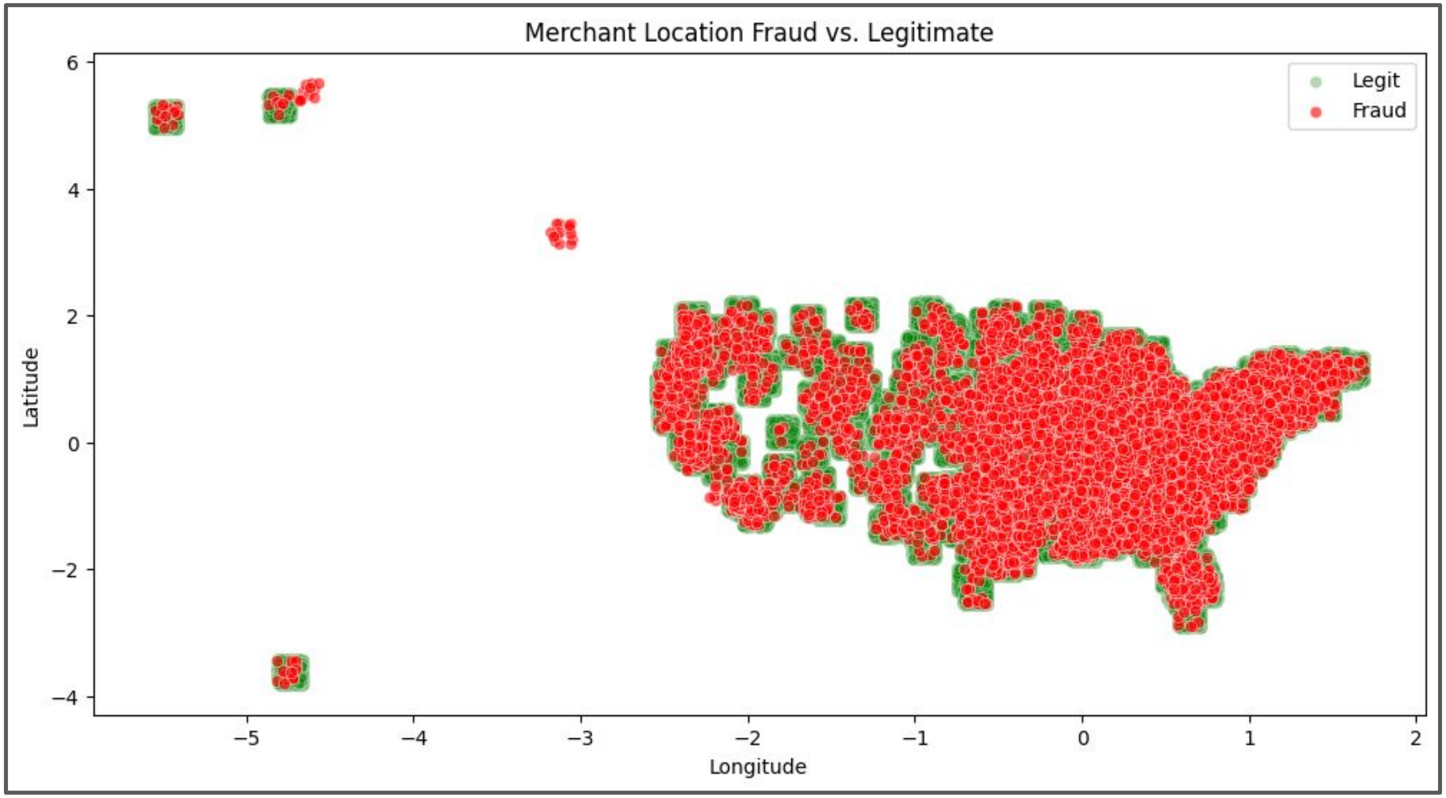
This shows that the number of legitimate transactions is relatively stable but increases significantly after 10th hour. Fraudulent transactions are very low in volume compared to legitimate ones so as to avoid detection but there is a slight increase in fraud cases during late night and early morning hours.



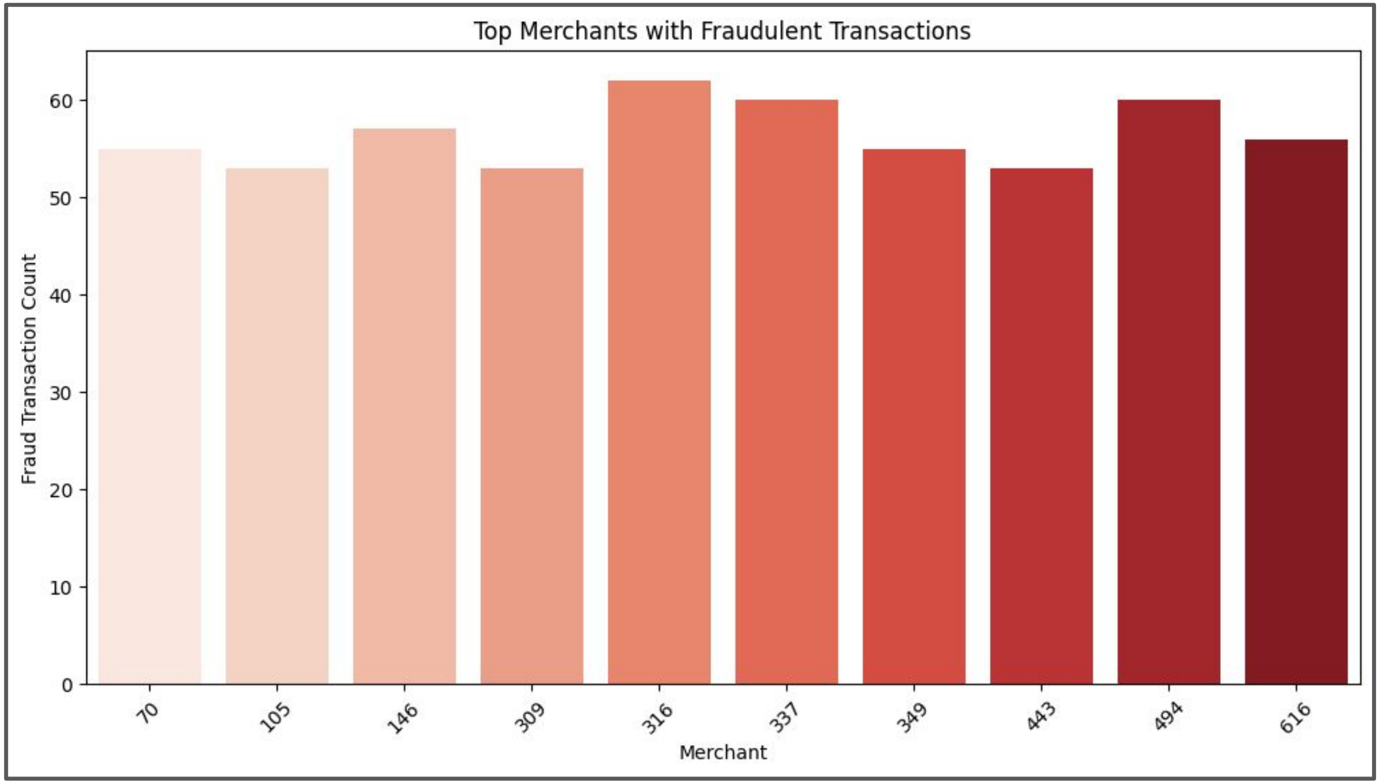
Fraud is not evenly distributed across categories—certain transaction types are much more likely to be fraudulent. Such as: `entertainment, gas_transport, grocery_pos, home, kids_pets, misc_net, misc_pos, personal_care, shopping_net, shopping_pos`



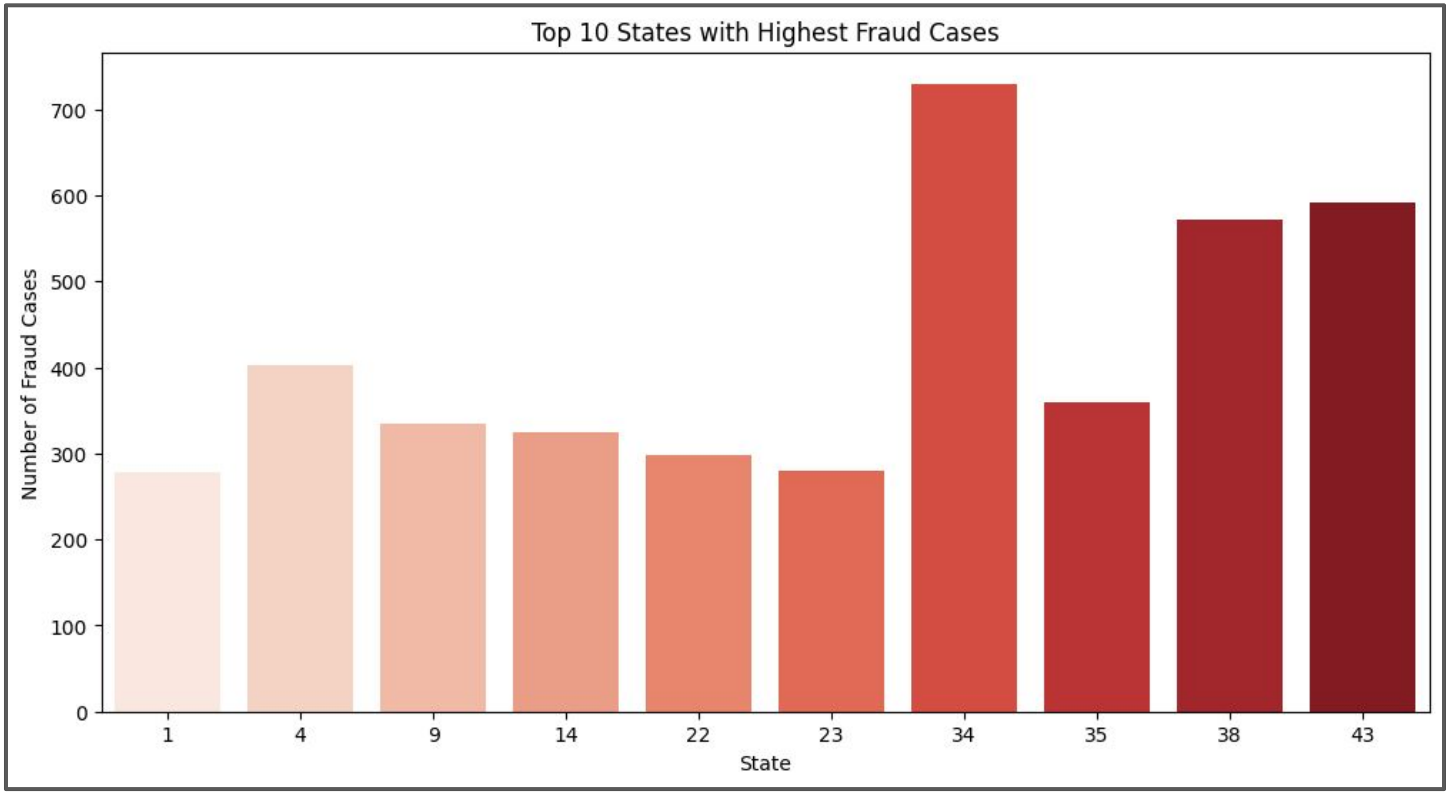
Fraudulent transactions (red) are densely packed across most merchant locations.
Legitimate transactions (green) appear surrounding or mixed within the fraud cases.



This chart represents the top merchants with the highest number of fraudulent transactions. The top merchants are: fraud_Boyer PLC, fraud_Cormier LLC, fraud_Doyle Ltd, fraud_KiehnEmmerich, fraud_Kilback LLC, fraud_KozeyBoehm, fraud_Kuhn LLC, fraud_Mosciski, Ziemann and Farrell, fraud_Rau and Sons, fraud_TerryHuel.



The top 10 states with highest fraud cases are:
AL, CA, FL, IL, MI, MN, NY, PA, TX



Data Preprocessing

- Dropped unnecessary columns (e.g., 'Unnamed: 0')
- Converted date columns to datetime format
- Extracted useful time based features
- Calculated age from date of birth (DOB)
- Handled missing values by dropping NA rows

Feature Engineering

- Extracted transaction time-based features (hour, day, month, weekday)
- Created 'age' from DOB
- Added transaction count per credit card
- Computed average transaction amount per card
- Calculated deviation from the mean transaction amount

Encoding & Scaling

- Encoded categorical variables (merchant-692, category-13, gender-2, state-50, job-496)
- Standardized numerical features (amount, city population, latitude/longitude, merchant location, age) using StandardScaler

Feature Engineering 2

- Flagged transactions occurring on weekends & night hours
- Computed geolocation based feature: distance between user and merchant
- Encoded transaction category and merchant for modeling

Model Selection and Training

Models used:

1. **Logistic Regression:** Simple, interpretable, and effective for binary classification.

2. **Random forest:** An ensemble model that reduces overfitting and handles nonlinear relationships.

3. **XGBoost:** Boosting based model that improves performance through gradient boosting.

Training Process

Training Process

Dataset Split:

✔ **80% Training** – Used to train the ML models.

✔ **20% Testing** – Used to evaluate model performance.

Evaluation Metrics Used:

✔ **Accuracy** – Measures overall correctness of predictions.

✔ **Precision & Recall** – Important for fraud detection (high recall = fewer missed fraud cases).

✔ **F1 Score** – Balances precision and recall.

Training the model

```
[30]
from sklearn.model_selection import train_test_split

# Selecting relevant features
features = ['transaction_count', 'amt_deviation', 'hour', 'day_of_week', 'is_weekend',
            'is_night', 'distance_from_home', 'category_encoded', 'merchant_encoded']
X = df[features]
y = df['is_fraud']

# Splitting into 80% training and 20% testing data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

print(f"Training Set Size: {X_train.shape}, Testing Set Size: {X_test.shape}")

```

```
Training Set Size: (1481915, 9), Testing Set Size: (370479, 9)
```

Model Selection

Model Training

[31]

```
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.metrics import classification_report

models = {
    "Logistic Regression": LogisticRegression(max_iter=1000),
    "Random Forest": RandomForestClassifier(n_estimators=100),
    "XGBoost": XGBClassifier(use_label_encoder=False, eval_metric='logloss')
}
```

Evaluation

[33]

```
for name, model in models.items():
    print(f" Training {name}...")
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    print(f" {name} Performance:")
    print(classification_report(y_test, y_pred))
    print("-" * 50)
```

Model Evaluation and Interpretation

We evaluated three models: **Logistic Regression, Random Forest, and XGBoost**. And the model performance analysis is:

1. **Logistic Regression**
   - Achieved **99% accuracy**, but performed **poorly on the minority class (class 1: Fraud)** with a precision and recall of **0.00**.
   - This indicates that the model is biased toward predicting the majority class (class 0: Legitimate), likely due to class imbalance.
2. **Random Forest**
   - Delivered **100% accuracy** with improved performance on class 1 (**Precision: 0.91, Recall: 0.40, F1 Score: 0.56**).
   - Although better at detecting the minority class, recall is still low, meaning many positive cases are being misclassified.
3. **XGBoost**
   - Similar to Random Forest, with a **macro average F1 Score of 0.72** and better recall.
   - Shows the best balance between precision and recall among all models.

Accuracy of the Model

```
Training Logistic Regression...
Logistic Regression Performance:
              precision    recall  f1-score   support

           0       0.99      1.00      1.00    368549
           1       0.00      0.00      0.00      1930

    accuracy                           0.99    370479
   macro avg       0.50      0.50      0.50    370479
weighted avg       0.99      0.99      0.99    370479

--------------------------------------------------
```

Logistic regression

```
Training Random Forest...
Random Forest Performance:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    368549
           1       0.91      0.40      0.56      1930

    accuracy                           1.00    370479
   macro avg       0.95      0.70      0.78    370479
weighted avg       1.00      1.00      1.00    370479

--------------------------------------------------
```

Random Forest

```
Training XGBoost...
XGBoost Performance:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    368549
           1       0.89      0.45      0.59      1930

    accuracy                           1.00    370479
   macro avg       0.94      0.72      0.80    370479
weighted avg       1.00      1.00      1.00    370479
```

XGBoost

Conclusion

- **Class Imbalance Issue:** Logistic Regression fails; tree based models perform better because we have more legitimate transaction data in the dataset.

- **XGBoost is the best model**, but recall still needs improvement.

- **Future Improvements:**

  - **Handle Class Imbalance** (SMOTE, undersampling).
  - **Threshold Tuning** to improve recall.
  - **Hyperparameter Optimization** for better model performance.

Final Takeaway: XGBoost performs best, but improving recall on class 1 is crucial!