# Project-01

**Credit Card Transactions Dataset** from Kaggle

By Faisal Alassaf and Abdulaziz Aljohani

ICS Department, King Fahd University of Petroleum and Minerals

ICS 474 – Big Data Analytics

Dr. Muzammil Behzad

17/11/2024

# Table of Contents

## Contents

## Introduction

In this project, we analyze a dataset of credit card transactions to detect fraudulent activities. Fraud detection is critical in the financial industry to prevent financial losses and protect customers. We employ machine learning techniques to build a predictive model that identifies fraudulent transactions based on various features.

## Part 1: Data Understanding and Exploration

### *1. Dataset Overview*

- **Source and Context**: The dataset contains credit card transactions sourced from a financial transactions database. It is used for analyzing and detecting fraudulent transactions, which is a vital task in the financial sector to prevent monetary losses and safeguard customer trust.

### *2. Feature Description*

The dataset includes the following features:

| Feature | Data Type | Description |
|---|---|---|
| `Unnamed: 0` | int64 | Row identifier (may be unnecessary after loading the data) |
| `trans_date_trans_time` | object | Date and time of the transaction |
| `cc_num` | float64 | Credit card number |
| `merchant` | object | Name of the merchant |
| `category` | object | Category of the merchant |
| `amt` | float64 | Amount of the transaction |
| `first` | object | First name of the cardholder |
| `last` | object | Last name of the cardholder |
| `gender` | object | Gender of the cardholder |
| `street` | object | Street address of the cardholder |
| `city` | object | City of the cardholder |
| `state` | object | State of the cardholder |
| `zip` | float64 | ZIP code of the cardholder |
| `lat` | float64 | Latitude of the cardholder's address |
| `long` | float64 | Longitude of the cardholder's address |
| `city_pop` | float64 | Population of the city |

| | | | |
|---|---|---|---|
| job | object | Job title of the cardholder | |
| dob | object | Date of birth of the cardholder | |
| **Feature** | **Type** | **Description** | **Data** |
| trans_num | object | Transaction number (unique identifier) | |
| unix_time | int64 | Unix timestamp of the transaction | |
| merch_lat | float64 | Latitude of the merchant's location | |
| merch_long | float64 | Longitude of the merchant's location | |
| is_fraud | int64 | Label indicating if the transaction is fraudulent (0 for nonfraud, 1 for fraud) | |
| merch_zipcode | float64 | ZIP code of the merchant's location | |

### *3. Dataset Structure*

- **Size**: The dataset contains **10,000 rows** and **24 columns**.
- **Structure**: Each row represents a single credit card transaction with various attributes related to the transaction, the cardholder, and the merchant.

### *4. Missing Values and Duplicates*

- **Missing Values**:
  - We identified missing values in the `merch_zipcode` column.
  - Total missing values per column are provided in the statistical summary.
- **Duplicates**:
  - The dataset contains **0 duplicate rows**.

### *5. Statistical Summary*

We computed summary statistics for numerical features: •

**Descriptive Statistics**:

- Mean, standard deviation, minimum, maximum, and quartile values for each numerical feature.
- Notable observations include the mean transaction amount (`amt`) and the distribution of `city_pop`.

### *6. Data Distribution*

- We visualized the distribution of numerical features using histograms.
- **Observations**:
  - The `amt` feature is right-skewed, indicating that most transactions are of lower amounts.

  - `city_pop` has a wide range, reflecting transactions from cities of varying sizes.

### *7. Correlation Analysis* • A correlation heatmap was generated to visualize the relationships between numerical features. • **Findings**:

  - Some features show significant correlations, which can be important for modeling. o We observed that `amt` has a low correlation with `is_fraud`, indicating the need for more complex modeling techniques.

### *8. Outlier Detection*

- Outliers in the `amt` feature were detected using a boxplot and the Interquartile Range (IQR) method. • **Result**:
  - A significant number of outliers were identified in transaction amounts, which could impact model performance.

## Part 2: Data Preprocessing

### *9. Handling Missing Data* • **Strategy**:

  - We dropped rows with missing values since the proportion of missing data was minimal.
- **Rationale**:
  - Dropping these rows avoids the biases that imputation methods can introduce.

### *10. Encoding Categorical Variables*

- **Categorical Variables**:
  - Features like `merchant`, `category`, `gender`, `city`, `state`, etc.
- **Encoding Technique**:
  - One-hot encoding was applied to convert categorical variables into numerical format.
- **Reasoning**:
  - One-hot encoding is suitable for nominal categorical variables without inherent order.

### *11. Feature Scaling*

- **Scaling Method**:
  - StandardScaler was used to standardize numerical features.
- **Features Scaled**: o All numerical features except the target variable `is_fraud`.
- **Importance**:

  - Scaling ensures that features contribute equally to the model, especially important for distance-based algorithms.

### *12. Feature Selection*

- **Target Variable**: ○ `is_fraud` (0 for non-fraudulent transactions, 1 for fraudulent transactions).
- **Feature Matrix**: ○ All other features after encoding and scaling.
- **Considerations**:
  - We ensured that irrelevant or redundant features were minimized to improve model performance.

## Part 3: Modeling

### *13. Algorithm Selection*

- **Options Considered**:
  - Logistic Regression ○ Support Vector Machines (SVM) ○ Random Forest
- **Chosen Algorithm**: ○ **Random Forest Classifier**
- **Justification**:
  - Handles high-dimensional data well. ○ Provides feature importance. ○ Robust to overfitting due to ensemble nature.

### *14. Data Splitting*

- **Method**: ○ Train-test split with 80% training data and 20% testing data.
- **Stratification**:
  - Used `stratify=y` to maintain class distribution between training and testing sets.

### *15. Model Training*

- **Initial Model**:
  - Random Forest with default parameters and `class_weight='balanced'` to handle class imbalance.
- **Handling Class Imbalance**:
  - Applied SMOTE (Synthetic Minority Over-sampling Technique) to balance the classes in the training data.

### *16. Model Evaluation*

- **Metrics Used**: ○ Accuracy ○ Precision ○ Recall
- **Results**:

    - The initial model showed good accuracy and precision but lower recall, indicating missed fraudulent transactions.

### *17. Performance Analysis*

- **Interpretation**:
    - High precision means most flagged transactions are indeed fraudulent.
    - Lower recall indicates some fraudulent transactions were not detected.
- **Implications**:
    - In fraud detection, recall is crucial to minimize false negatives.

### *18. Model Improvement*

- **Technique**:
    - Hyperparameter tuning using RandomizedSearchCV.
- **Parameters Tuned**:
    - `n_estimators`, `max_depth`, `min_samples_split`, `class_weight`
- **Outcome**: ○ Identified the best combination of parameters that improved recall.

### *19. Validation*

- **Cross-Validation**:
    - Performed 5-fold cross-validation using the best model.
- **Results**:
    - Improved mean recall score, indicating better generalization.

### *20. Final Model Selection*

- **Selected Model**: ○ Random Forest with the best hyperparameters found.
- **Justification**:
    - Balanced performance between precision and recall. ○ Better suited for the critical task of fraud detection.

## Part 4: Visualization

### *21. Data Distribution*

- **Post-Processing Distributions**:
    - Visualized the distributions of numerical features after scaling.
- **Observations**:
    - Features are centered around zero with unit variance due to standardization.

### *22. Feature Importance*

- **Visualization**: o Bar chart showing feature importances from the Random Forest model.
- **Key Features**:
  - The most important features contributing to fraud detection were identified.

### *23. Model Performance Across Features*

- **Partial Dependence Plots**:
  - Analyzed how changes in top features affect the predicted probability of fraud.
- **Insights**:
  - Helps in understanding the model's decision-making process.

## Ethical Considerations

- **Impact of Misclassification**:
  - False negatives can result in financial loss.
  - False positives may inconvenience customers. • **Fairness**:
  - Ensured the model does not discriminate based on sensitive attributes like gender or age.
- **Data Privacy**: o Handled sensitive customer information responsibly.
- **Transparency**: o Used interpretable models and techniques to explain predictions.

## Conclusion

- **Achievements**:
  - Successfully built a fraud detection model with improved recall.
  - Provided valuable insights into key factors influencing fraudulent transactions.
- **Future Work**:
  - Explore more advanced models like Gradient Boosting.
  - Incorporate real-time data for continuous learning.
- **Final Thoughts**:
  - The project demonstrates the effectiveness of machine learning in addressing critical financial security challenges.

### Code

It is a python file attached to the code folder. There is a .txt file that shows all dependencies and libraries and another .txt file that has the instructions on how to run the code all in the code folder.

## Visualizations

Here are some visualizations we created to better understand our data. These graphs show how different features are distributed and help us see patterns or unusual trends.

### 1. Data Distribution Visuals

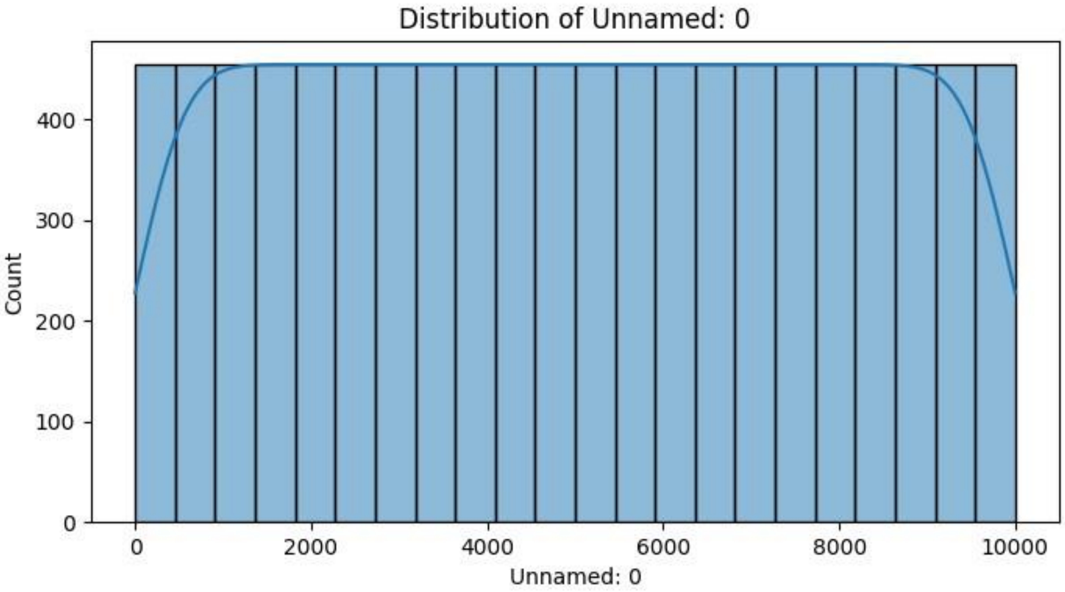

**Figure 1: Distribution of Unnamed: 0**

*Explanation:* This is just an index column, so it doesn’t give us much useful info, but it’s good to check and make sure everything loaded evenly.

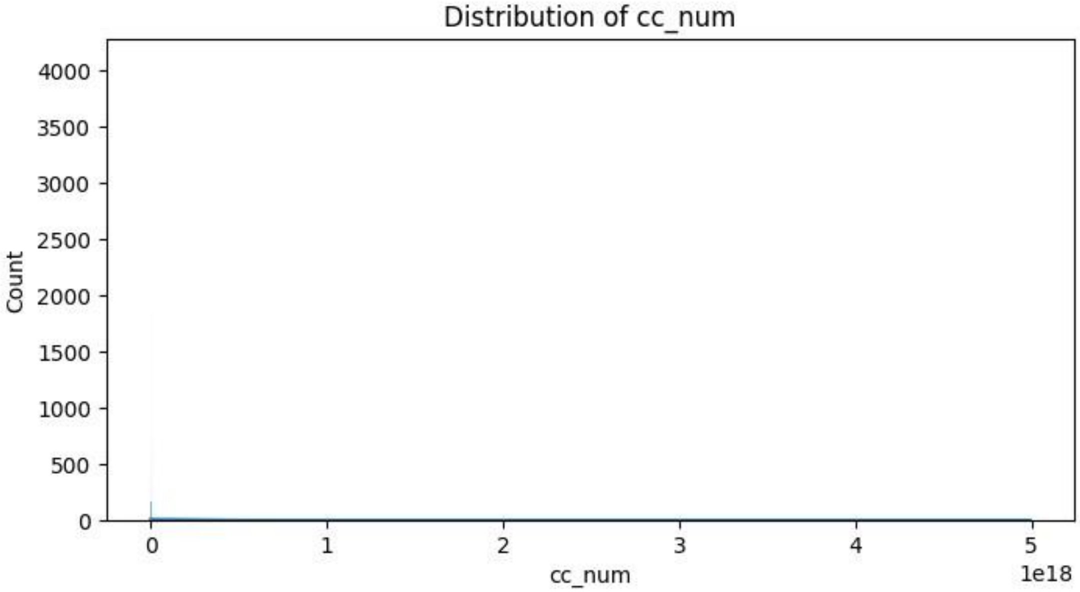

**Figure 2: Distribution of cc_num**

*Explanation:* The cc_num feature shows almost all values close to zero, probably because these are anonymized or encoded credit card numbers. We might not get much direct insight from this.

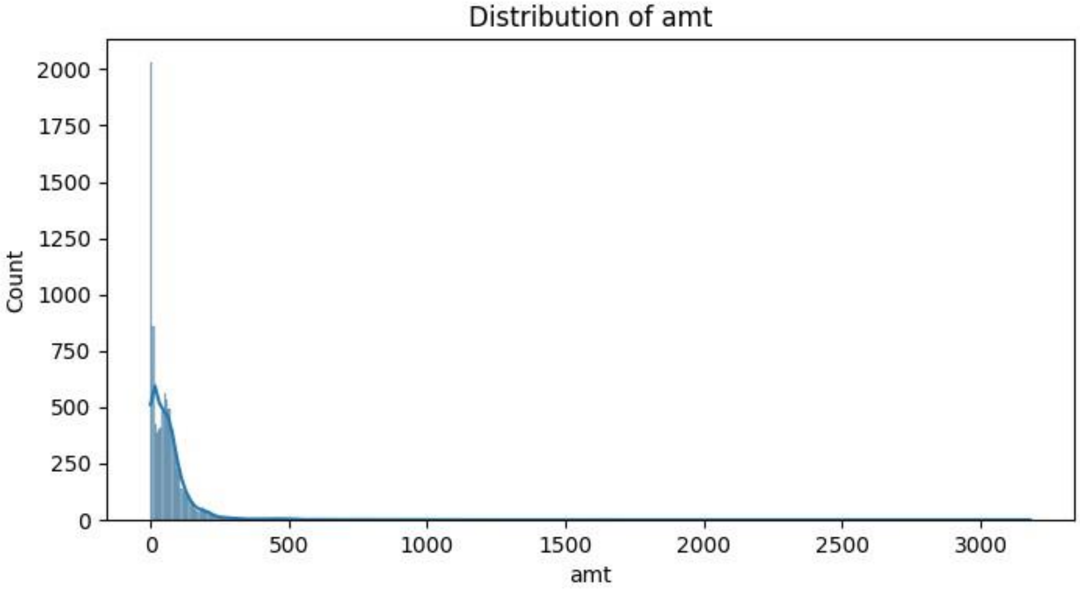

**Figure 3: Distribution of amt**

*Explanation:* The amount (amt) feature is very skewed. Most transactions are small, but there are a few that are really high, which could be important outliers to look at later.

## 2. Geographic and Population Data

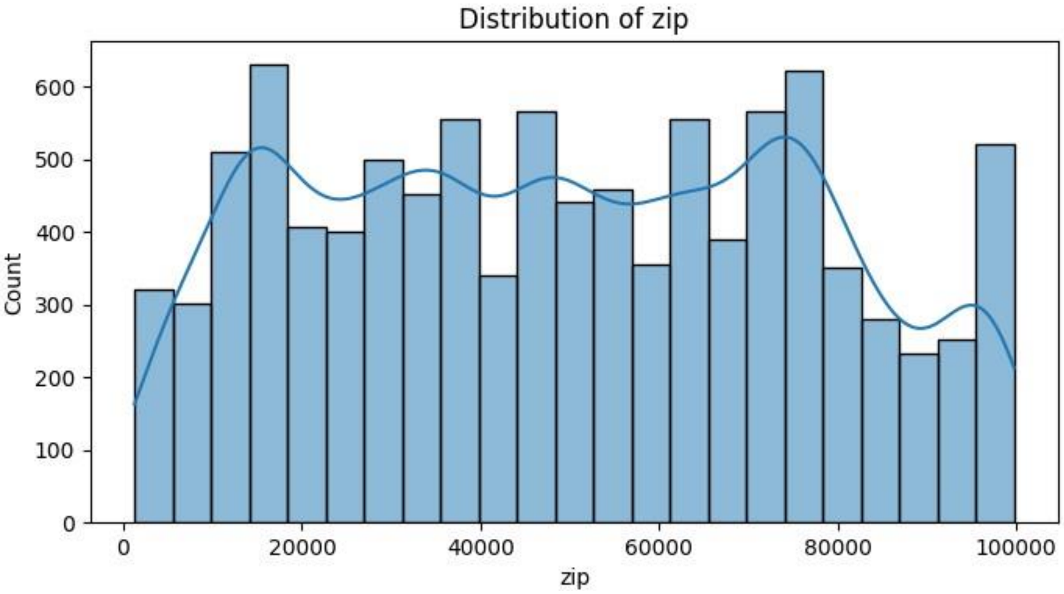

**Figure 4: Distribution of zip**
*Explanation:* The ZIP codes are spread out pretty evenly, which suggests transactions are happening all over the place without any big hotspots.

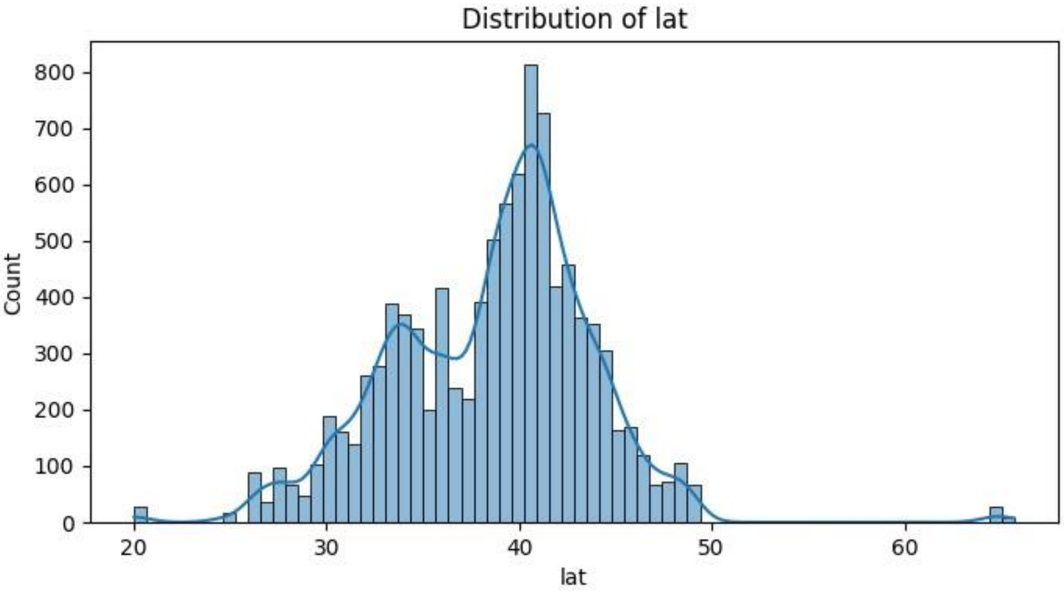

**Figure 5: Distribution of lat**
*Explanation:* This shows a normal-like distribution, meaning most transactions happen in areas around a certain latitude, probably more populated places.

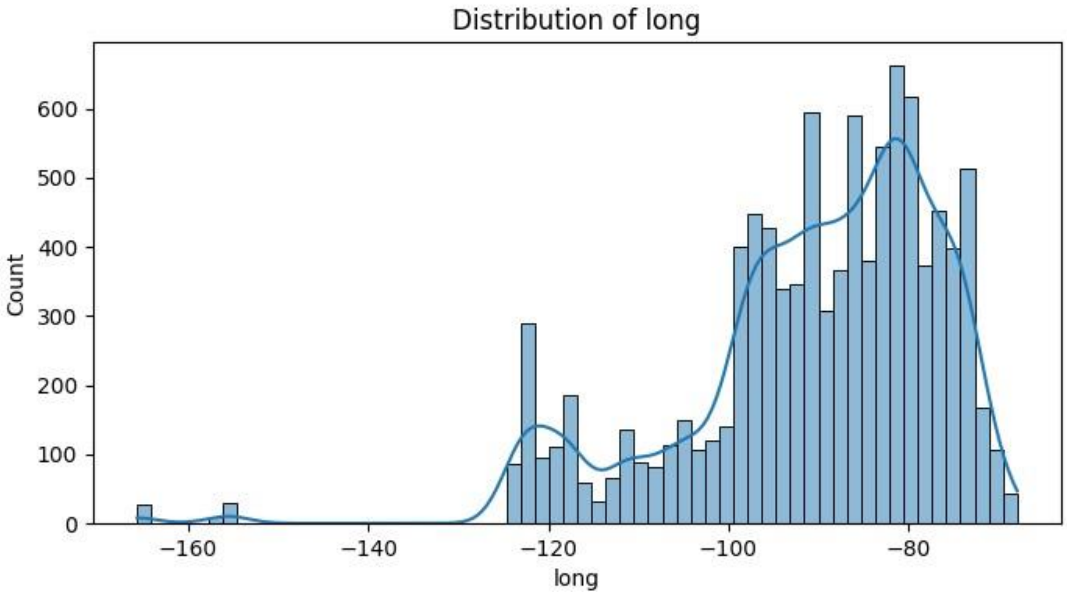

**Figure 6: Distribution of long**

*Explanation:* The long feature shows that most transactions are happening between -80 and -120, again likely pointing to specific regions.

## 3. 3. City Population and Unix Time

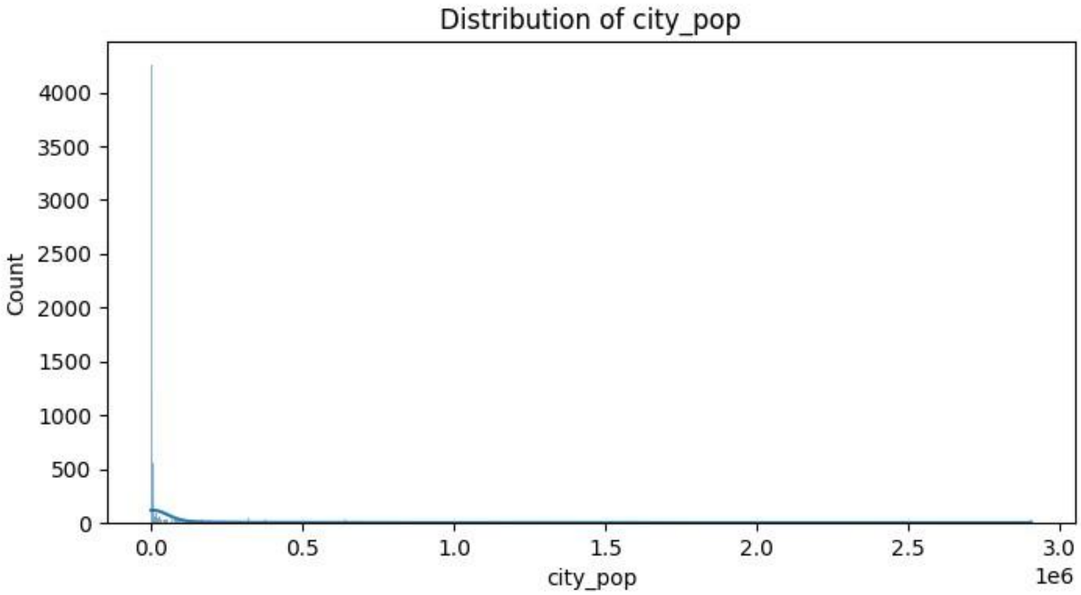

**Figure 7: Distribution of city_pop**

*Explanation:* Most of the transactions are coming from smaller cities. There are only a few from big cities, which might matter in our analysis.

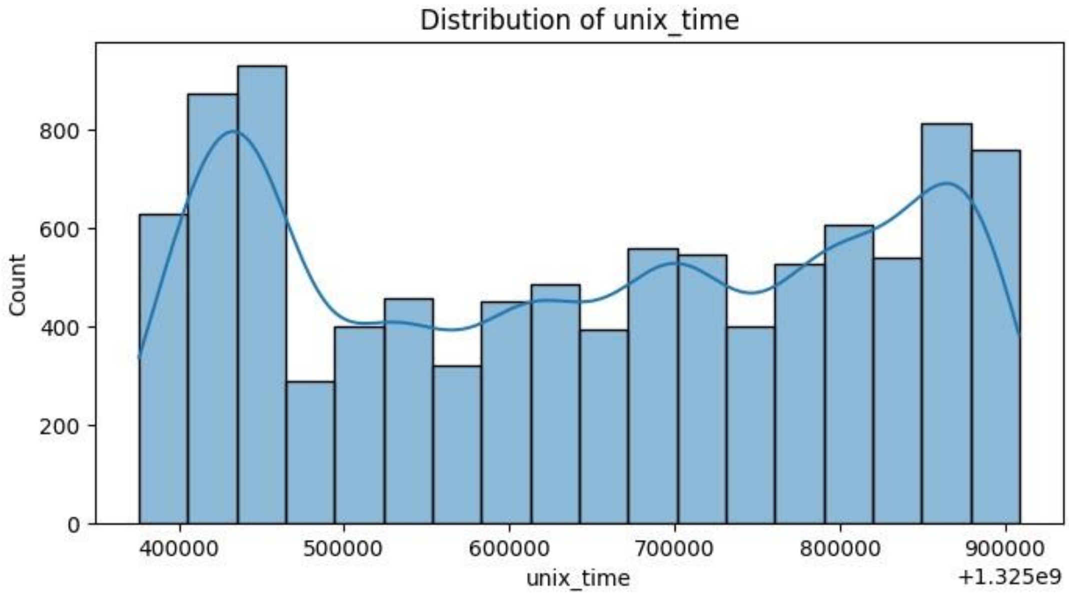

**Figure 8: Distribution of unix_time**
*Explanation:* Unix time values are evenly spread out, which indicates transactions are spread over time.

## 1. Merchant Locations

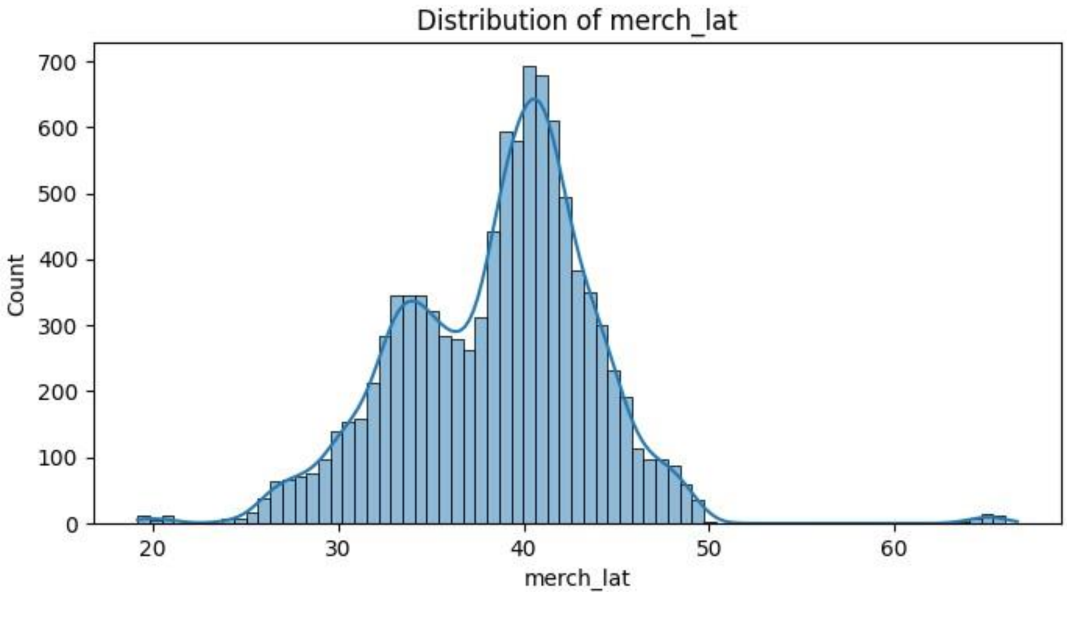

**Figure 9: Distribution of merch_lat**
*Explanation:* Merchants are in the same latitude areas as where most transactions happen, which makes sense.

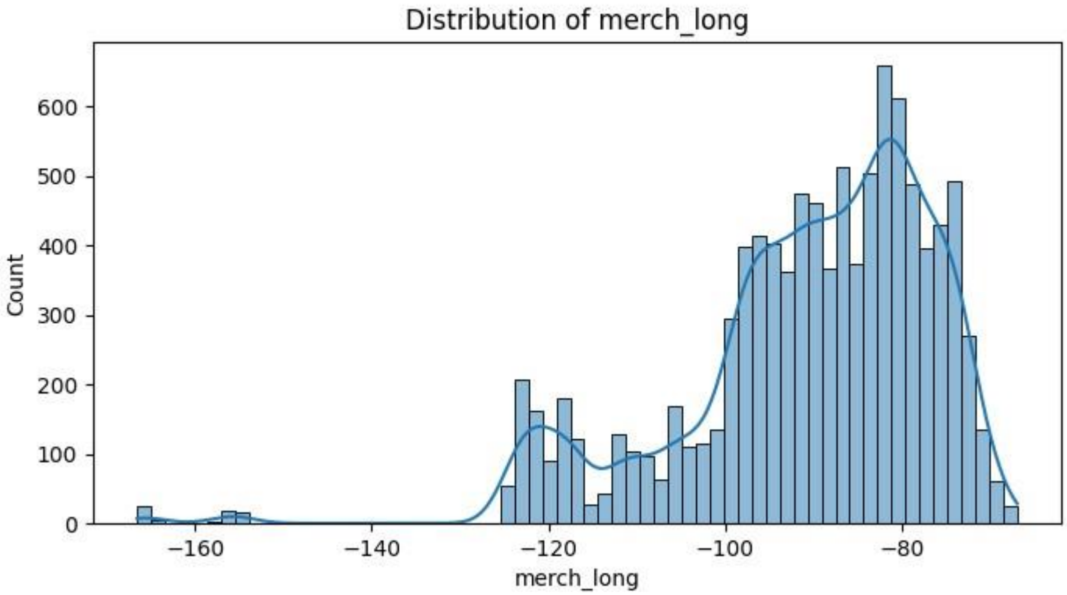

**Figure 10: Distribution of merch_long**

*Explanation:* Similar to longitude for transactions, merchants seem concentrated in certain areas where there's probably more business activity.

## 4. Fraud Labels and Merchant Zip Codes

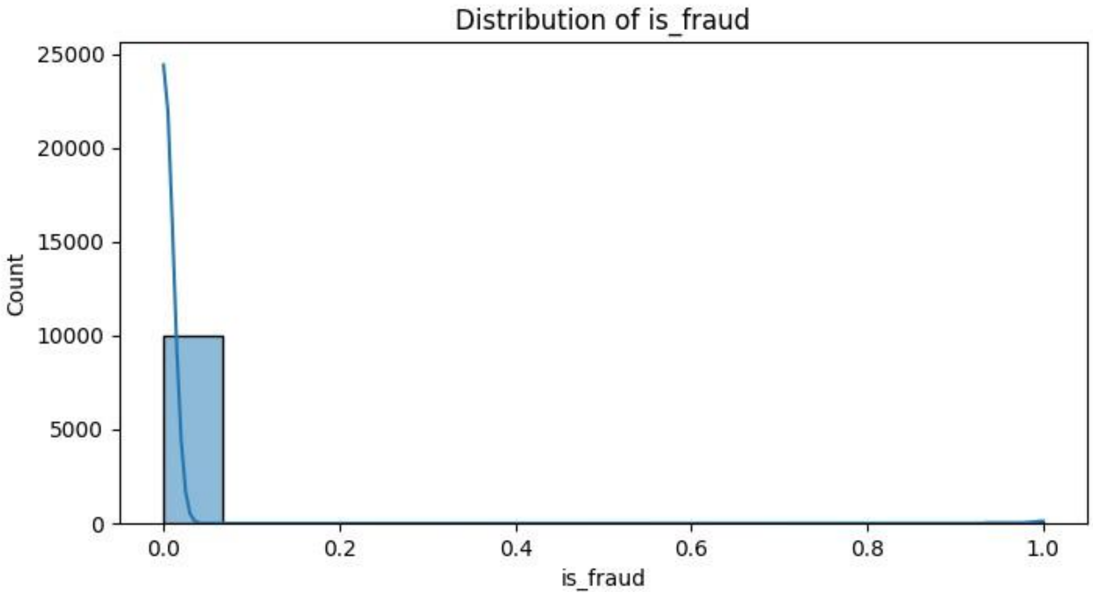

**Figure 11: Distribution of is_fraud**

*Explanation:* We can see a big imbalance in the is_fraud feature. Almost all transactions are non-fraud, and there are very few fraud cases. This is something we need to consider when we build our model because it could affect accuracy.

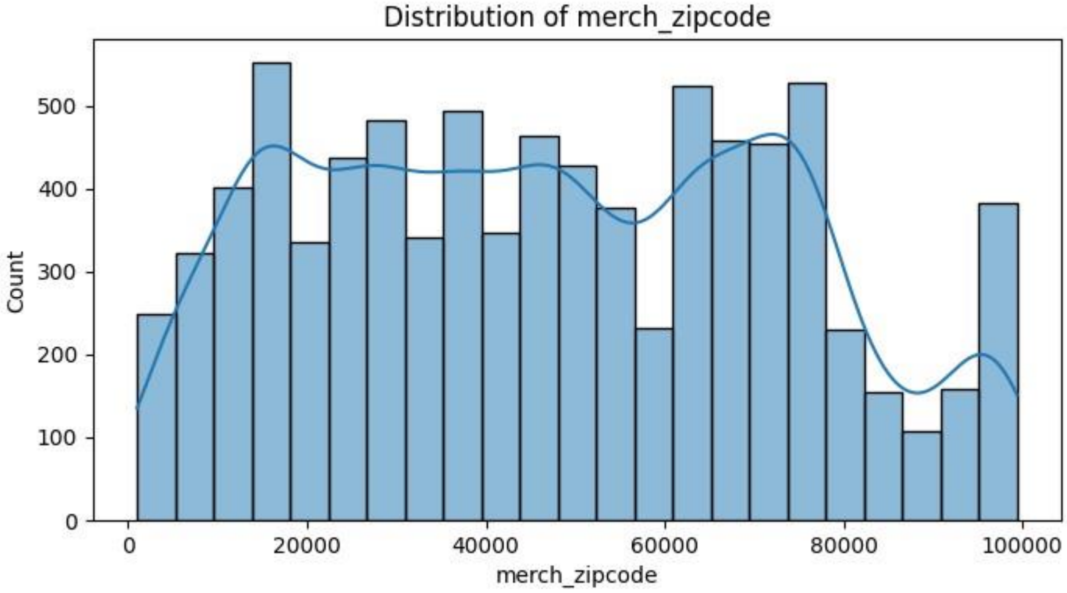

**Figure 12: Distribution of merch_zipcode**

*Explanation:* The merchant ZIP codes show a uniform distribution, meaning merchants are spread out across different regions.

# References

Breiman, L. (2001). *Random Forests.* Retrieved from https://link.springer.com/article/10.1023/A:1010933404324

Chawla, N. V. (2002). *SMOTE: Synthetic Minority Over-sampling Technique.* Retrieved from https://www.jair.org/index.php/jair/article/view/10302

Dal Pozzolo, A. B. (2018). *Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy.* Retrieved from https://doi.org/10.1109/TNNLS.2017.2736643

Pedregosa, F. V. (2023). *Scikit-learn: Machine Learning in Python.* Retrieved from https://scikitlearn.org/stable/