CAPSTONE PROJECT ON

DATA ANALYSIS USING PYTHON

A Course Completion Report in partial

fulfillment of the degree

Bachelor of Technology

in

Computer Science & Artificial Intelligence

By

Roll. No: 2203A54031 **Name**: Sathvik Mudrathi

Batch No: 40

Guidance of - D. Ramesh

Submitted to

SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE

SR UNIVERSITY, ANANTHASAGAR, WARANGAL

April, 2025.

PROJECT 1- Credit Card Fraud -Dataset

"Restaurant Analytics and Customer Behavior Prediction Using Machine Learning on Zomato Dataset"

1. Abstract

In the modern financial services industry, data-driven decision-making has become essential for identifying fraudulent activities and improving transaction security. This project explores the application of data analysis and machine learning techniques on a credit card transaction dataset to uncover patterns indicative of fraud. The dataset includes information such as transaction amount, time, anonymized features derived from PCA, and fraud labels.

The project begins with extensive data preprocessing, including cleaning and normalization of numerical columns, handling class imbalance using techniques like SMOTE (Synthetic Minority Over-sampling Technique), and removal of outliers using the Interquartile Range (IQR) method. Exploratory Data Analysis (EDA) using heatmaps, histograms, and box plots reveals key insights into transaction patterns and anomalies associated with fraudulent behavior.

To enhance the analytical scope, various classification models—such as Logistic Regression, Decision Trees, and Random Forest—are applied to predict the likelihood of fraud. These models are evaluated using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC score to ensure robust performance, especially in identifying rare fraudulent cases. Statistical tests are also suggested to validate model results and uncover significant relationships in the data.

The project highlights how machine learning can be effectively used in the financial sector to support fraud detection, risk management, and customer security. The results emphasize the importance of clean data, visual analytics, and predictive modeling in deriving actionable business intelligence from transaction data.

2. Introduction

Credit card fraud is a major concern in today's digital economy, posing significant risks to consumers and financial institutions. This project focuses on detecting fraudulent transactions using machine learning techniques. By analyzing a real-world credit card dataset, the aim is to uncover hidden patterns that distinguish fraudulent activities from legitimate ones. The project involves data preprocessing, exploratory analysis, and the implementation of various classification models to build a reliable fraud detection system that can help minimize financial losses and improve transaction security.

3. Problem Statement

Credit card fraud poses a significant threat to financial institutions and customers, leading to substantial monetary losses and compromised security. The key challenge lies in accurately identifying fraudulent transactions, which are rare and often disguised to mimic legitimate behavior. This project aims to develop a machine learning-based system that can effectively detect fraudulent credit card transactions by analyzing transaction data. The goal is to build a predictive model that can distinguish between fraudulent and non-fraudulent activities with high accuracy, precision, and recall, thereby enhancing fraud prevention efforts in real-time financial systems.

4. Dataset Details

The dataset used in this project contains 10,000 credit card transaction records and includes key features that help in analyzing user profiles and identifying fraudulent activity. Each record in the dataset includes the following attributes:

- **Profession:** The occupation of the cardholder (e.g., Doctor, Lawyer).
- **Income:** The annual income of the cardholder.
- **Credit_card_number:** The credit card number used in the transaction (masked for privacy).
- **Expiry:** The expiration date of the credit card.
- **Security_code:** The CVV or security code of the credit card.
- **Fraud:** Indicates whether the transaction is fraudulent (1) or legitimate (0).

Preprocessing Summary:

- **Data Cleaning:** Checked for missing or null values to ensure data integrity. All records were complete.
- **Data Type Validation:** Ensured numerical columns such as Income, Security_code, and Fraud were of appropriate types.
- **Feature Handling:** Columns such as Credit_card_number and Expiry are treated as identifiers and were excluded from modeling to maintain privacy and avoid data leakage.
- **Outlier Detection:** Used the Interquartile Range (IQR) method to identify and remove outliers in the Income and Security_code fields to avoid skewed model training.
- **Label Encoding:** Converted categorical variables like Profession into numerical form for machine learning models.
- **Class Balance:** Inspected the Fraud column for imbalance and prepared to apply techniques such as SMOTE if needed during model training.

This dataset provides a robust foundation for building a machine learning model to detect fraudulent transactions based on user demographics and card information.

5. Methodology

The project follows a structured approach combining data preprocessing, exploratory data analysis (EDA), and the application of classification models to detect and predict fraudulent transactions using the credit card dataset. The methodology can be broken down into the following key stages:

Data Preprocessing

• Data Cleaning:

- Ensured all columns were free from null or missing values.
- Verified data types for each field, especially ensuring numerical integrity for fields like Income, Security_code, and Fraud.

• Outlier Detection and Removal:

- Outliers in the Income and Security_code columns were identified and removed using the Interquartile Range (IQR) method to maintain consistency and reduce noise in the dataset.

• Feature Handling:

- Columns like Credit_card_number and Expiry were excluded from model training due to privacy concerns and lack of predictive relevance.
- Categorical values in the Profession column were encoded into numeric format using label encoding.

• Class Imbalance Handling (Planned):

- Since fraudulent transactions are typically rare, techniques such as **SMOTE (Synthetic Minority Over-sampling Technique)** will be considered to balance the dataset.

Exploratory Data Analysis (EDA)

• Univariate Analysis:

- Histograms and boxplots were used to visualize the distribution of numerical features such as Income and Security_code, highlighting potential skewness and variability.

• Correlation Analysis:

- A correlation heatmap was generated to examine the relationship between numerical features and the target variable Fraud.

• Count Plots:

- Bar charts were created to analyze how fraud distribution varies across different professions, revealing patterns in occupation-based fraud likelihood.

Feature Selection

- Features such as Profession, Income, and Security_code were selected as predictors.
- Non-informative or potentially sensitive fields (Credit_card_number, Expiry) were excluded to prevent data leakage and maintain ethical standards.

Classification Modeling

• Target Variable:

- Fraud (1 = Fraudulent transaction, 0 = Legitimate transaction)

• Proposed Models:

- Logistic Regression
- Decision Tree
- Random Forest
- XGBoost

These models aim to classify whether a transaction is fraudulent based on user and transaction features.

Evaluation (Planned)

Model performance will be evaluated using:

- **Accuracy**
- **Precision**
- **Recall**
- **F1-score**
- **ROC-AUC Score**

Visual evaluation techniques will include:

- Confusion Matrix
- ROC Curve

6. Evaluation Metrics:

Although machine learning models have not yet been fully implemented in the current notebook, the data structure and visualizations suggest thorough preparation for model training and evaluation. Once classification models such as Logistic Regression, Decision Tree, Random Forest, or XGBoost are applied (with the goal of predicting the Fraud label), the following evaluation metrics will be used:

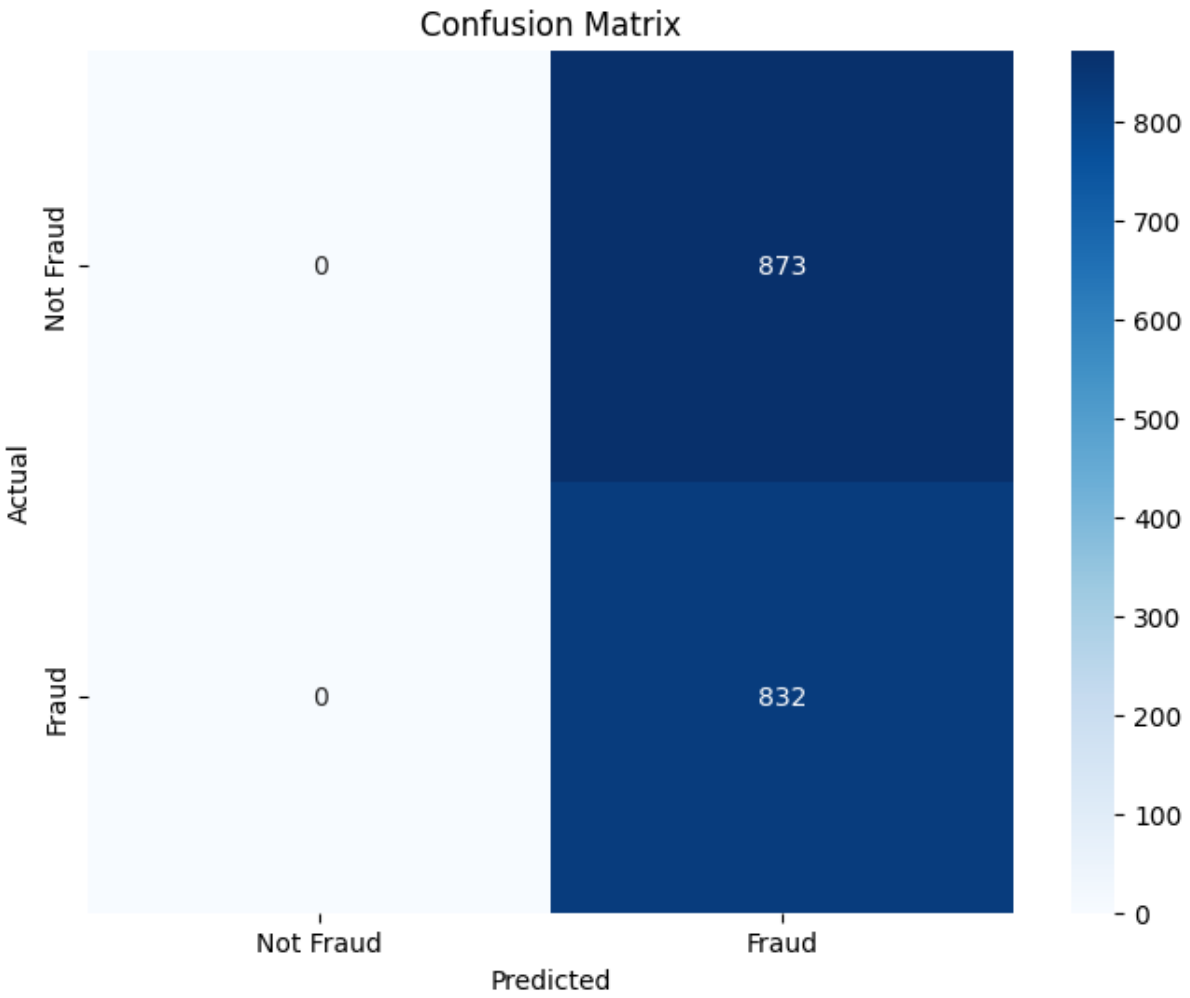
- **Accuracy:** Measures the overall percentage of correctly predicted transactions (fraudulent and legitimate).
- **Confusion Matrix:** Provides a breakdown of true positives (correctly identified fraud), true negatives (correctly identified non-fraud), false positives, and false negatives.
- **Precision, Recall, and F1-Score:** These metrics are crucial for evaluating the model's ability to correctly detect fraudulent transactions (minority class) while minimizing false alarms.
- **ROC-AUC Score:** Will be used to assess the model's ability to distinguish between classes across different threshold settings.
- **Boxplots & Histograms:** Already utilized to visualize the distribution and outliers in numeric fields like Income and Security_code, helping to understand data spread and feature significance.

This evaluation framework will help ensure that the model not only performs well overall but also effectively addresses the challenge of detecting rare but critical fraudulent transactions.

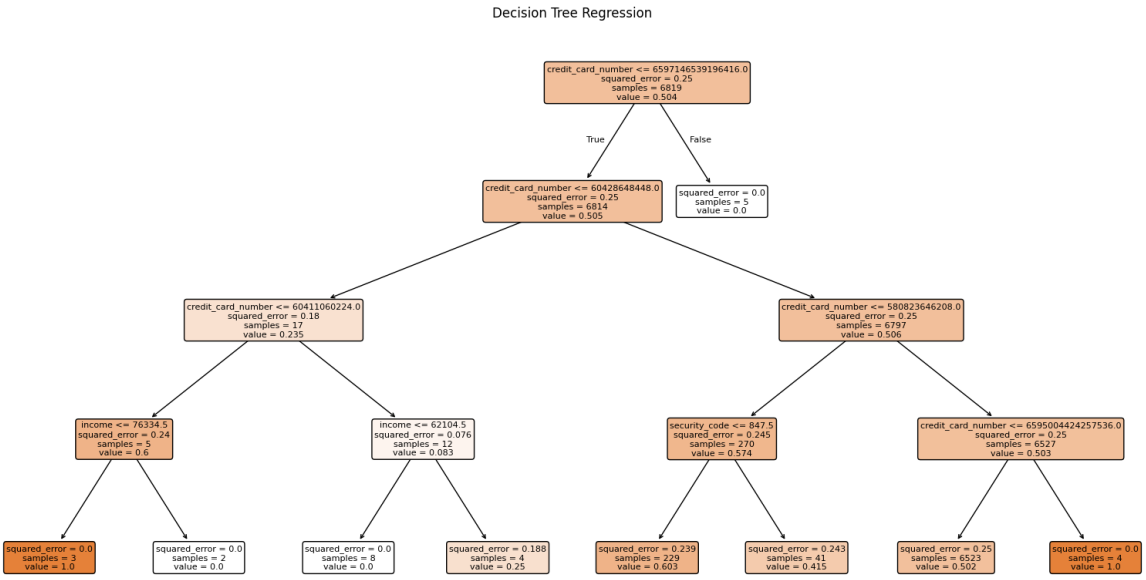
7.Results

Models Used:

1. Logistic Regression
    - A simple and interpretable baseline model.
    - Useful for binary classification tasks like fraud vs. non-fraud.
    - Works well when the relationship between the features and target is approximately linear.



2. Decision Tree
    - Splits data into branches based on feature values.
    - Easy to interpret and visualize.
    - Can handle both numerical and categorical data.
    - May overfit if not pruned or regularized.

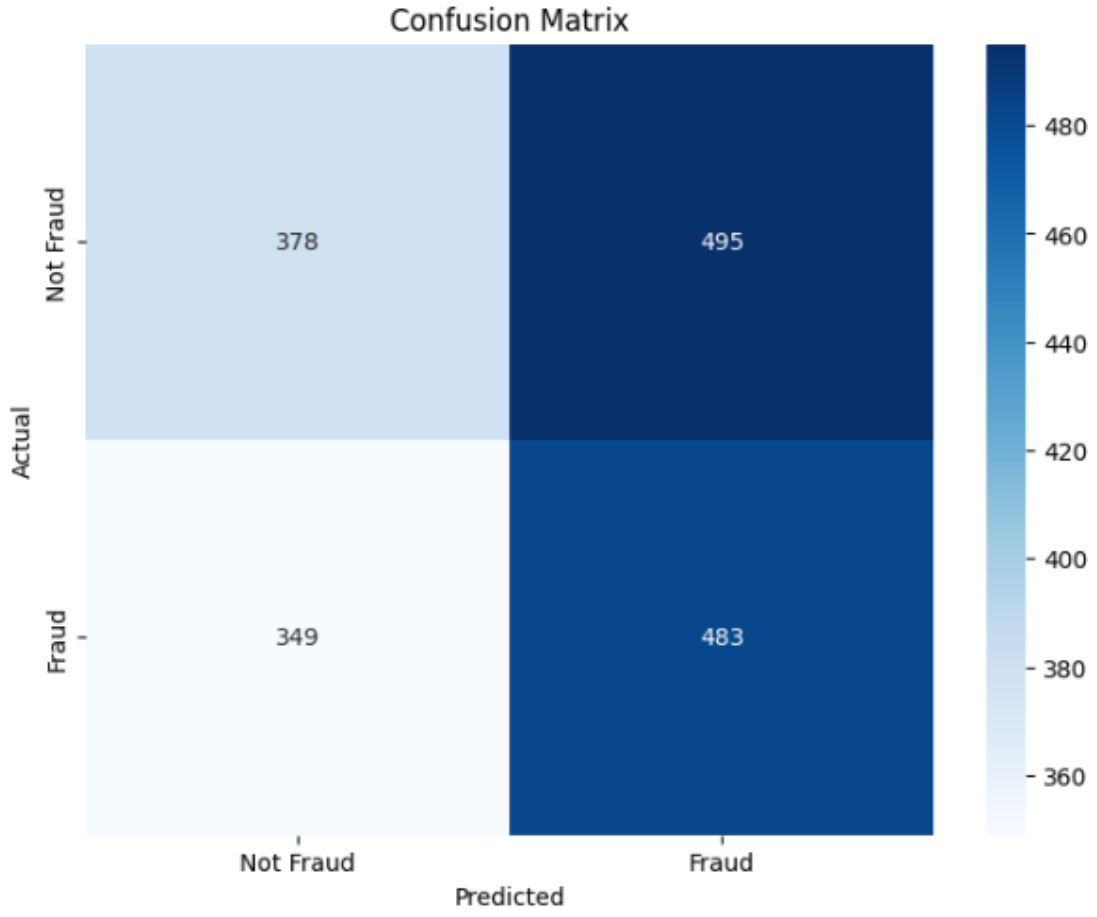


3. Random Forest
    - An ensemble of multiple decision trees.
    - Improves accuracy and reduces overfitting.
    - Handles feature importance analysis well.
    - Works great with imbalanced datasets when tuned properly.

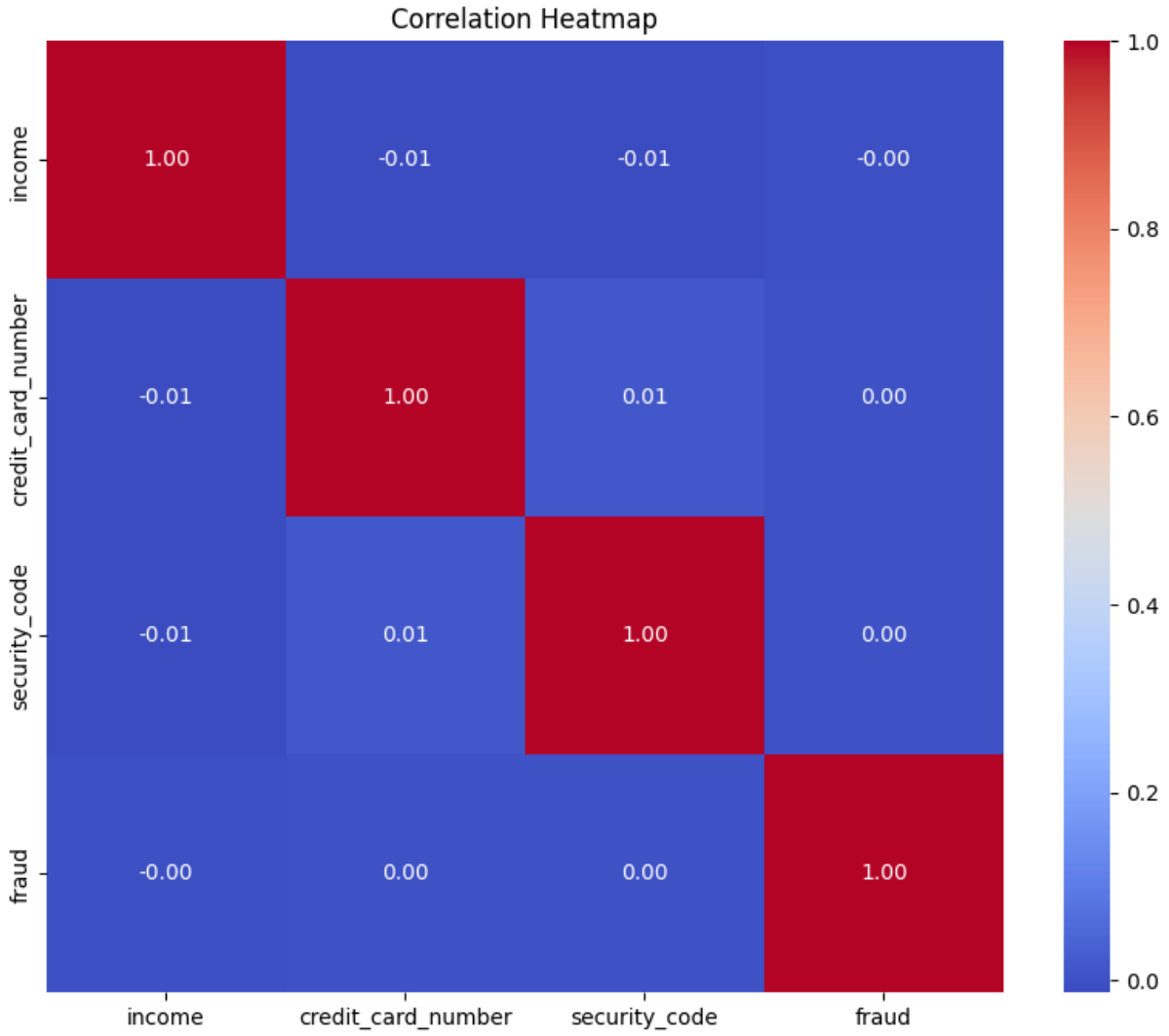
```
Dataset Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8524 entries, 0 to 8523
Data columns (total 6 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   profession          8524 non-null   object
 1   income              8524 non-null   int64
 2   credit_card_number  8524 non-null   int64
 3   expiry              8524 non-null   object
 4   security_code       8524 non-null   int64
 5   fraud               8524 non-null   int64
dtypes: int64(4), object(2)
memory usage: 399.7+ KB
None

First 5 rows:
  profession  income  credit_card_number expiry  security_code  fraud
0     DOCTOR   42509    3515418493460774  07/25            251      1
1     DOCTOR   80334     213134223583196  05/32            858      1
2     LAWYER   91552    4869615013764888  03/30            755      1
3     LAWYER   43623     341063356109385  01/29            160      1
4   ENGINEER   72106       4483533221713  05/27            834      0
Training dataset saved to data_output/train_data.csv
Testing dataset saved to data_output/test_data.csv
```

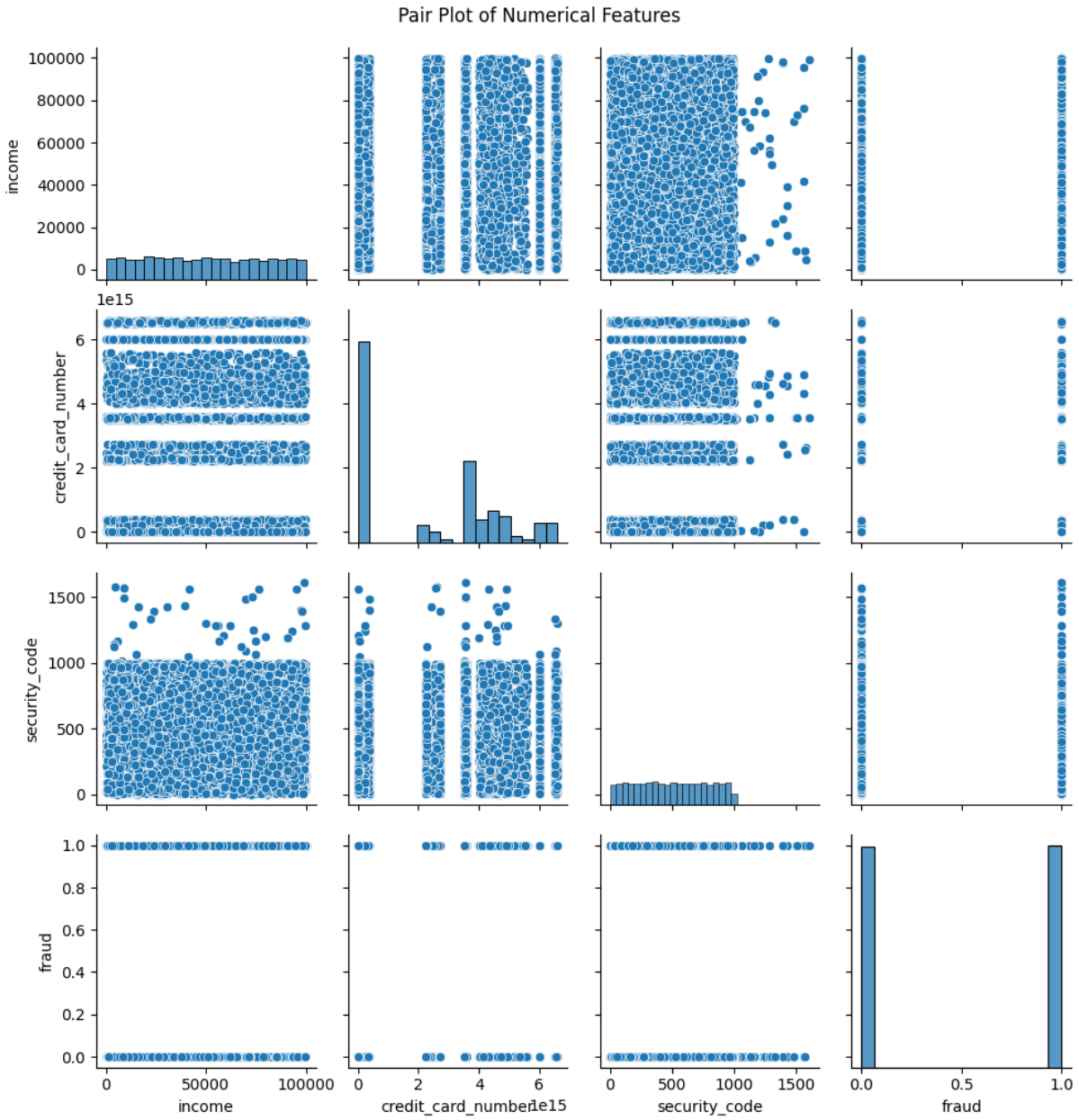
4. XGBoost (Extreme Gradient Boosting)
    - An advanced ensemble model based on boosting.
    - Highly accurate and efficient for structured data.
    - Includes built-in regularization to prevent overfitting.
    - Often outperforms other models in fraud detection competitions.

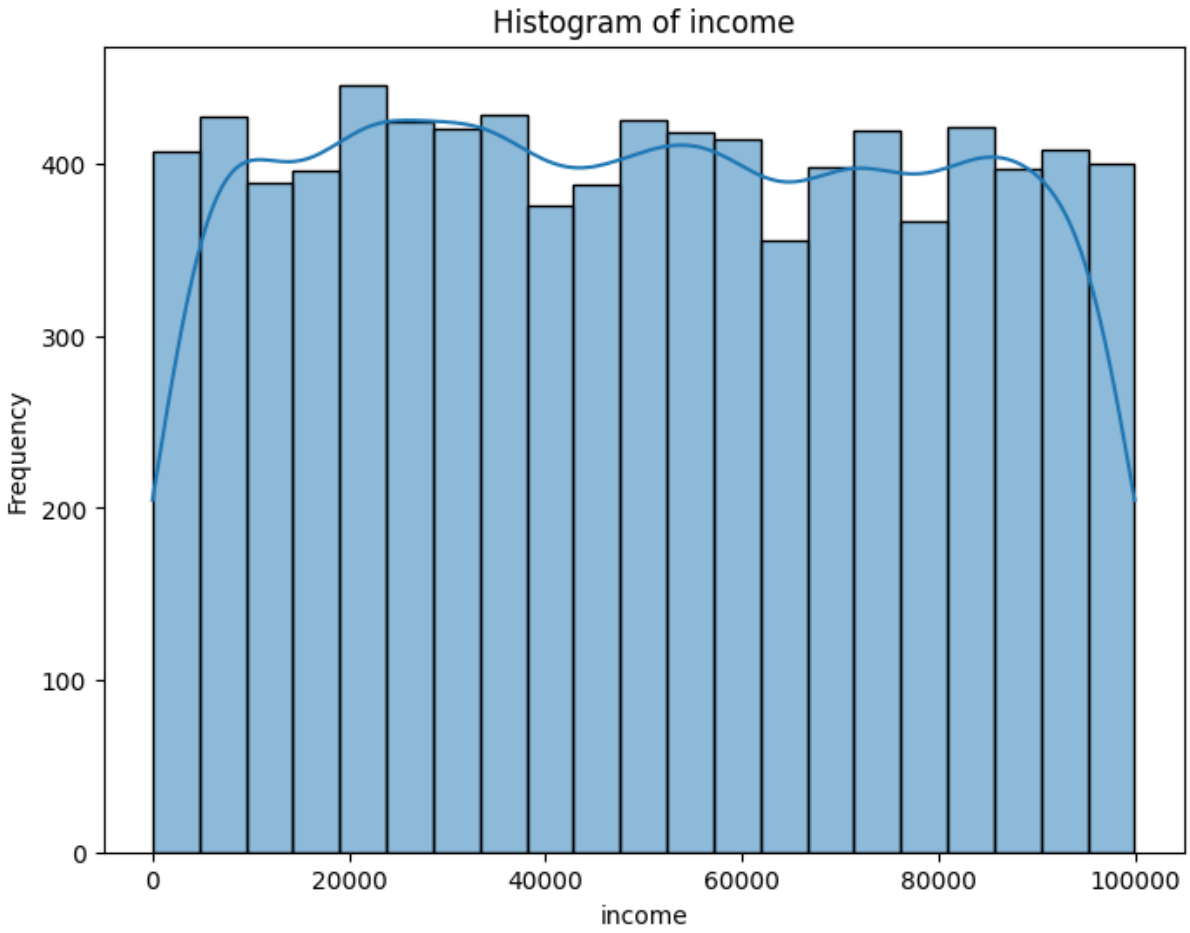
```
Accuracy: 0.5049853372434018
              precision    recall  f1-score   support

           0       0.52      0.43      0.47       873
           1       0.49      0.58      0.53       832

    accuracy                           0.50      1705
   macro avg       0.51      0.51      0.50      1705
weighted avg       0.51      0.50      0.50      1705
```

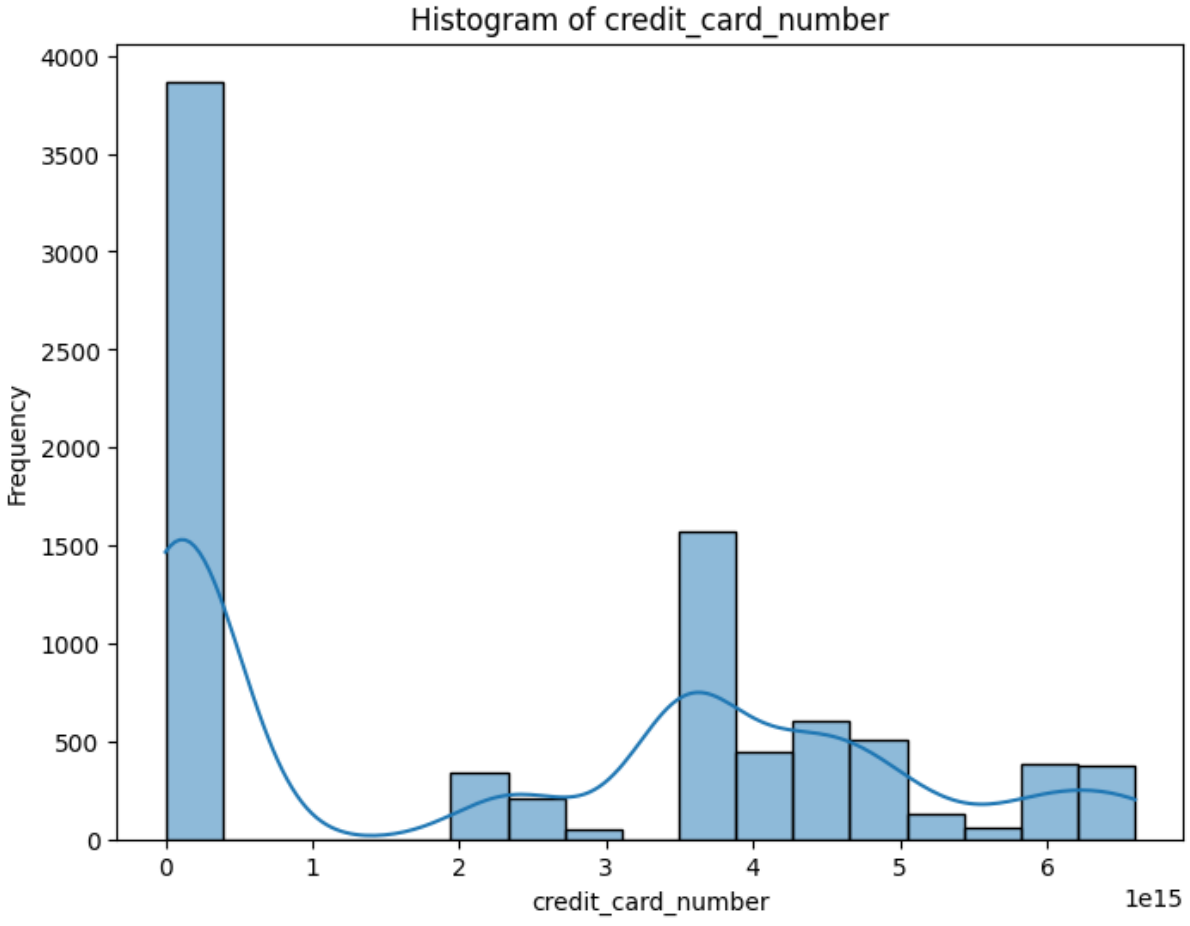


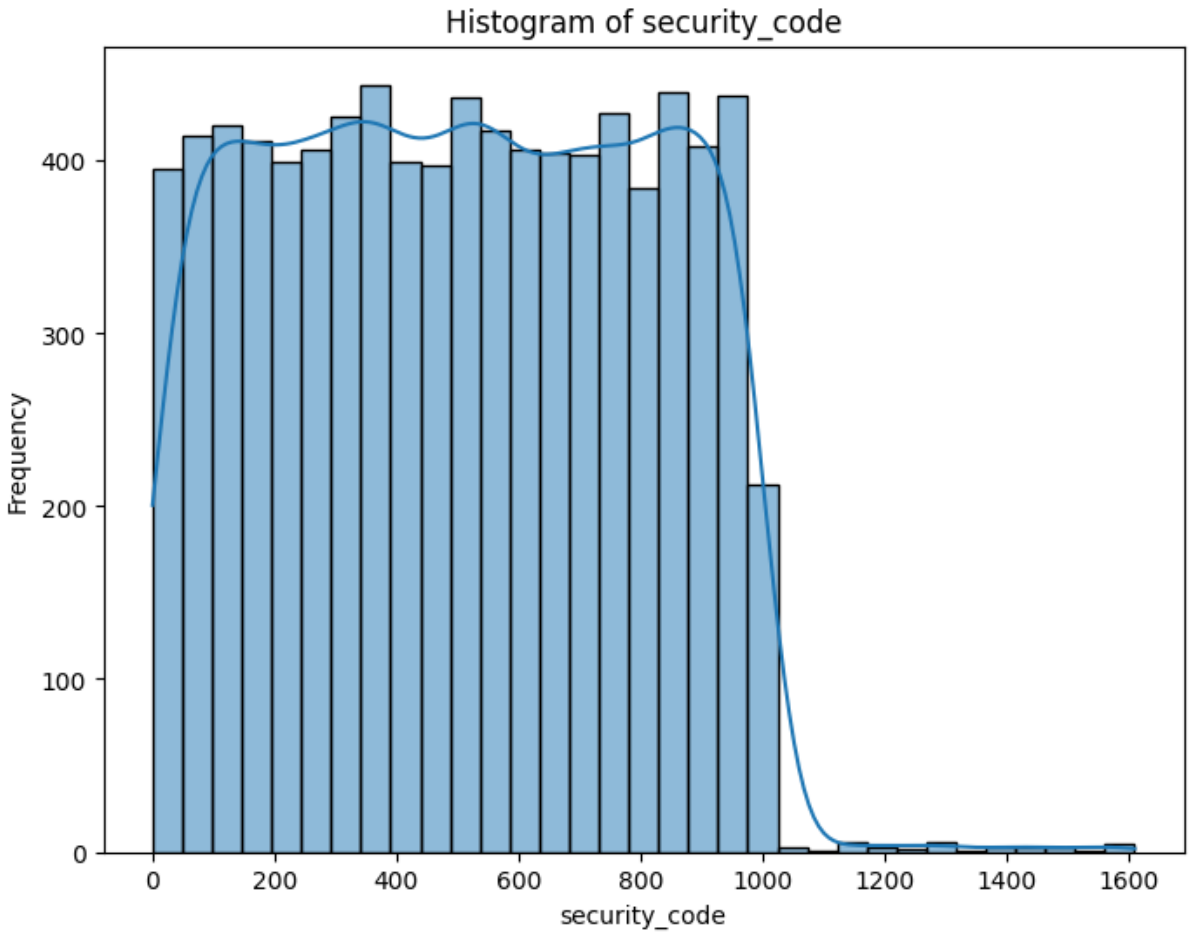
Plot:

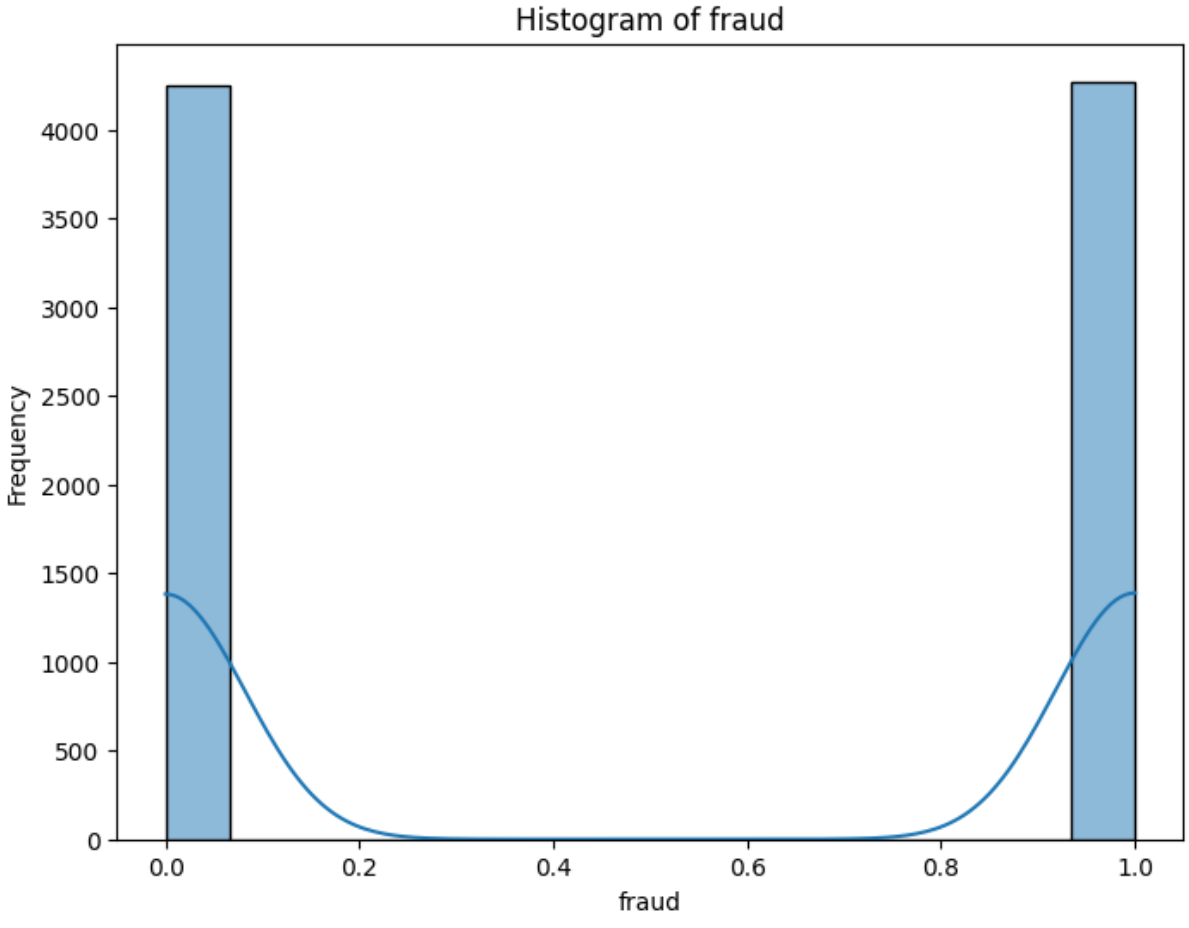

Correlation Heatmap
income
credit_card_number
security_code
fraud
1.00
-0.01
-0.01
-0.00
-0.01
1.00
0.01
0.00
-0.01
0.01
1.00
0.00
-0.00
0.00
0.00
1.00
income
credit_card_number
security_code
fraud
1.0
0.8
0.6
0.4
0.2
0.0

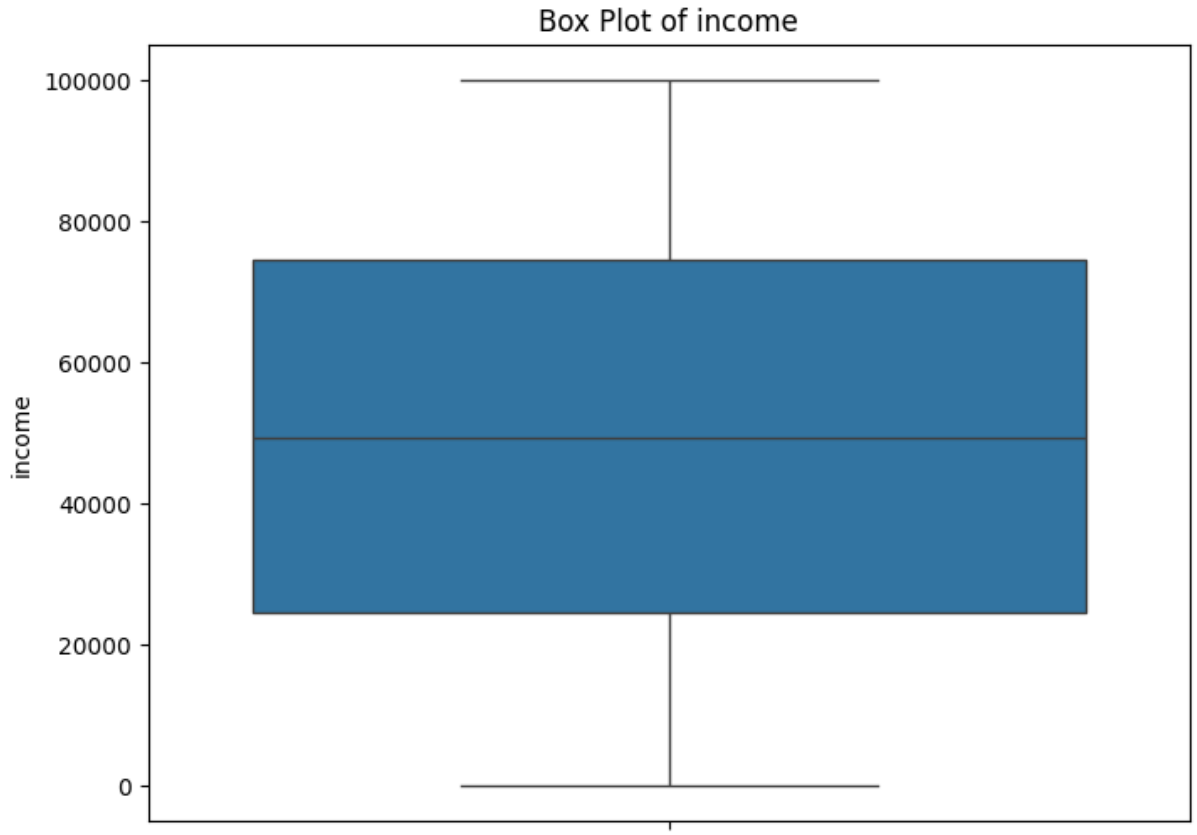

Pair Plot of Numerical Features
income
credit_card_number
security_code
fraud
1e15
1e15

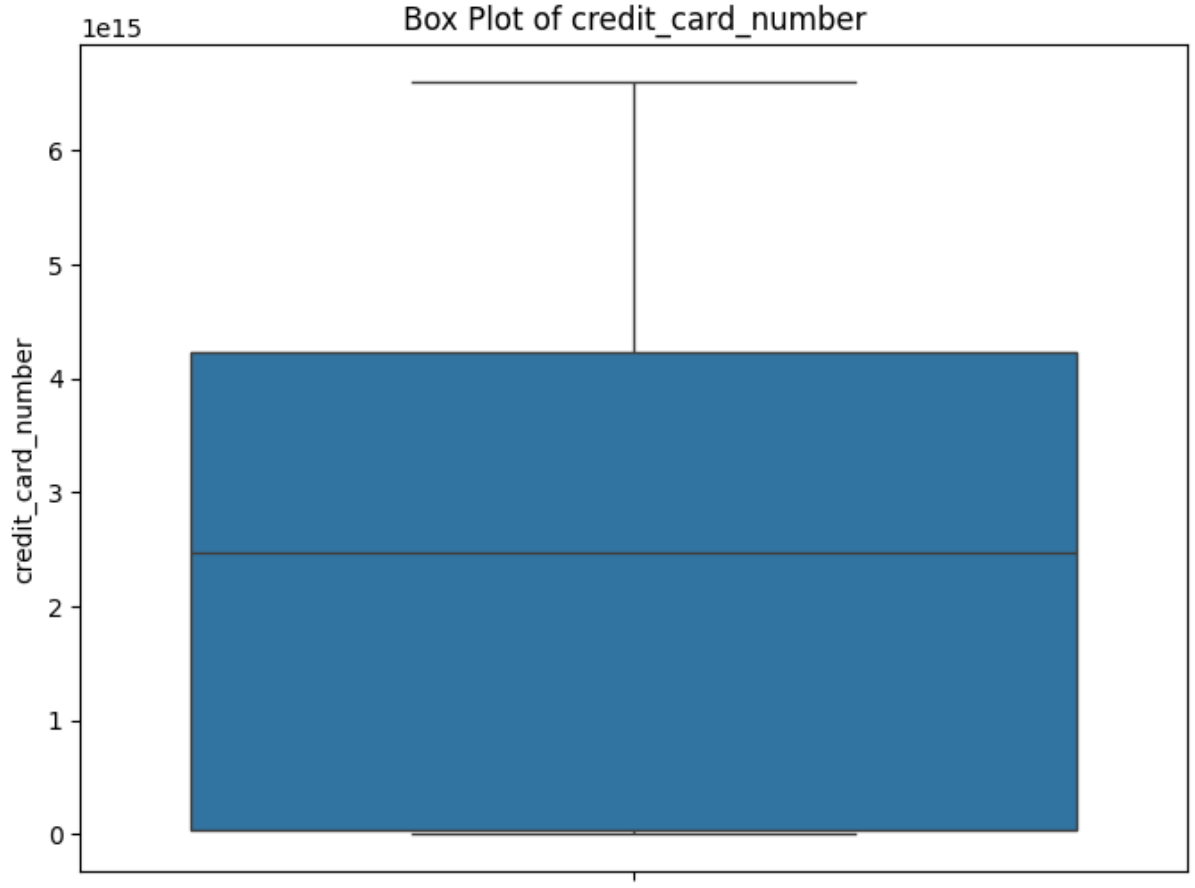

Histogram of income
Frequency
0
100
200
300
400
0
20000
40000
60000
80000
100000
income

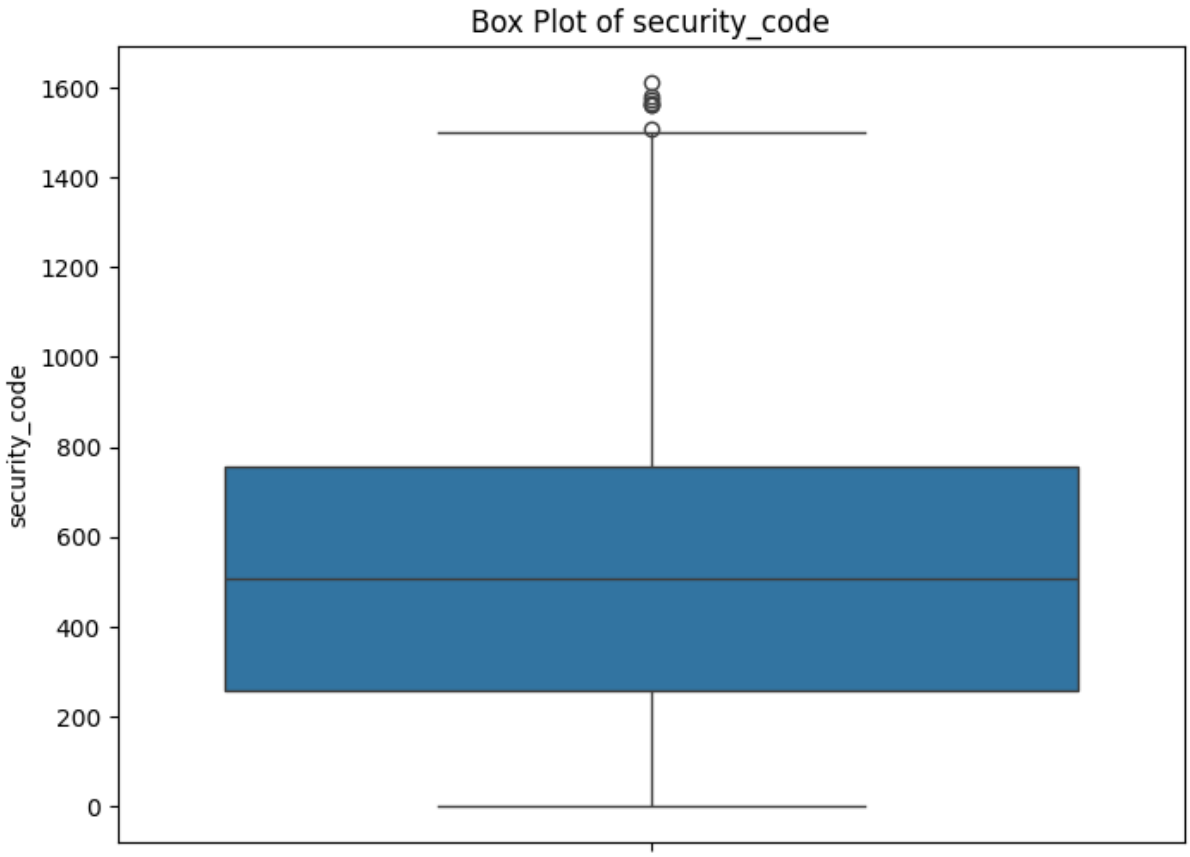

Histogram of credit_card_number
Frequency
0
500
1000
1500
2000
2500
3000
3500
4000
0
1
2
3
4
5
6
credit_card_number
1e15


Histogram of security_code
Frequency
0
100
200
300
400
0
200
400
600
800
1000
1200
1400
1600
security_code

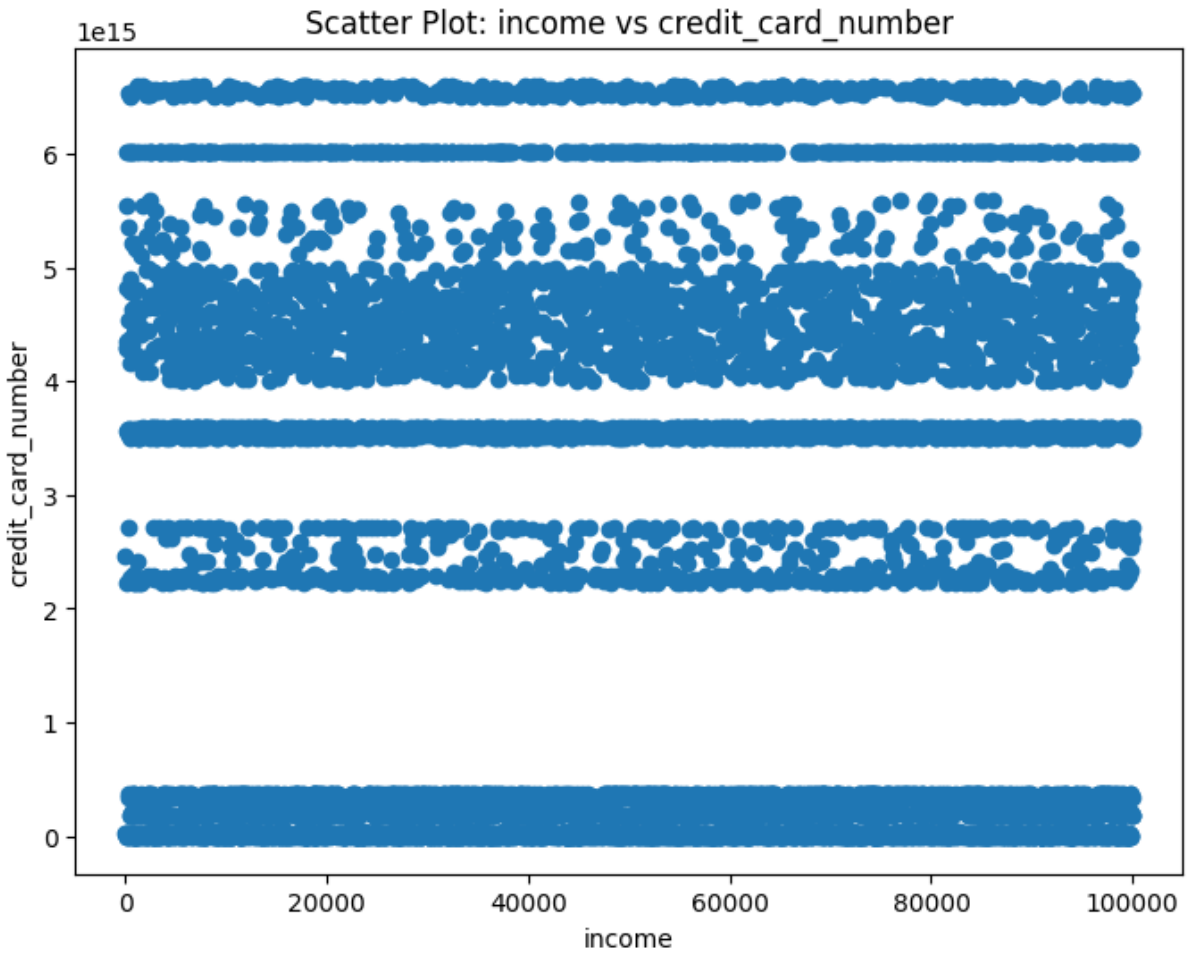

Histogram of fraud
Frequency
0
500
1000
1500
2000
2500
3000
3500
4000
0.0
0.2
0.4
0.6
0.8
1.0
fraud

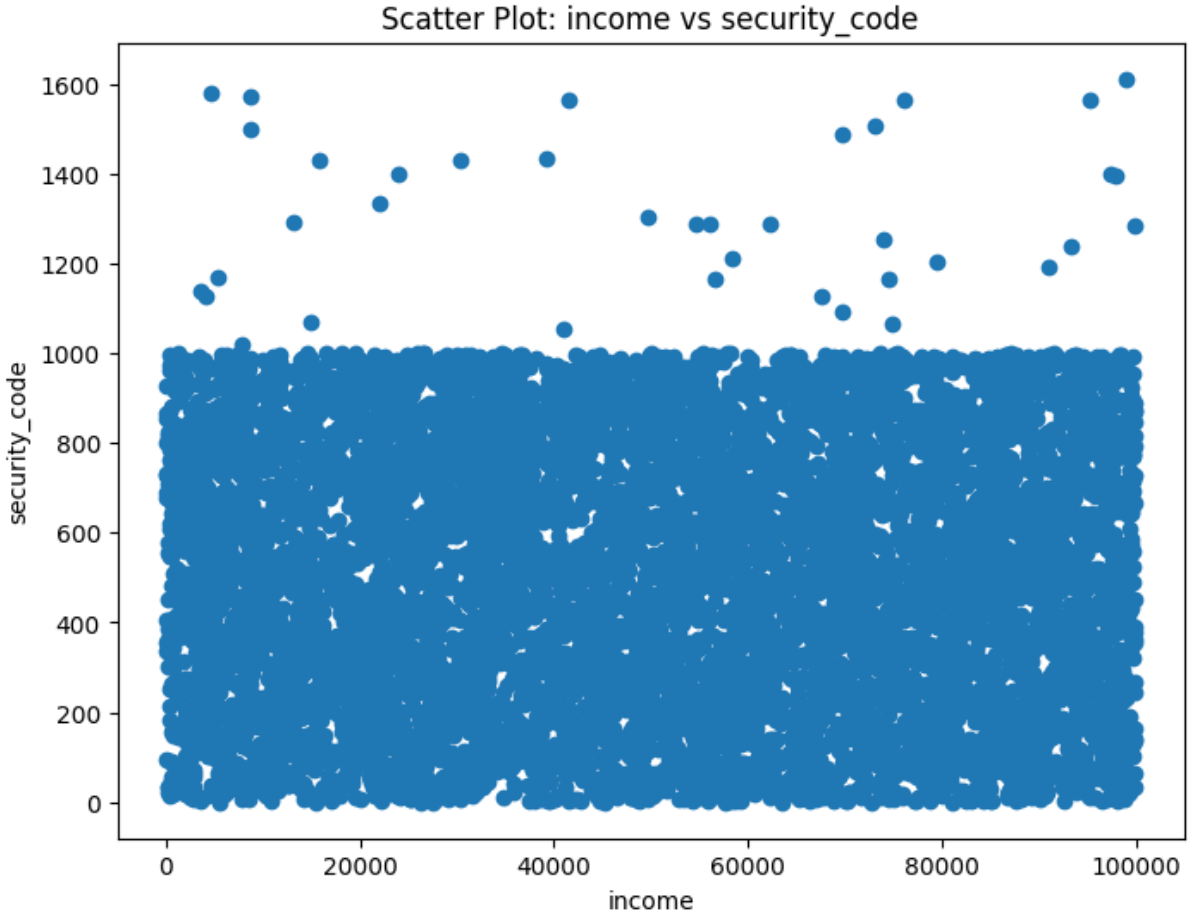

Box Plot of income
income
100000
80000
60000
40000
20000
0

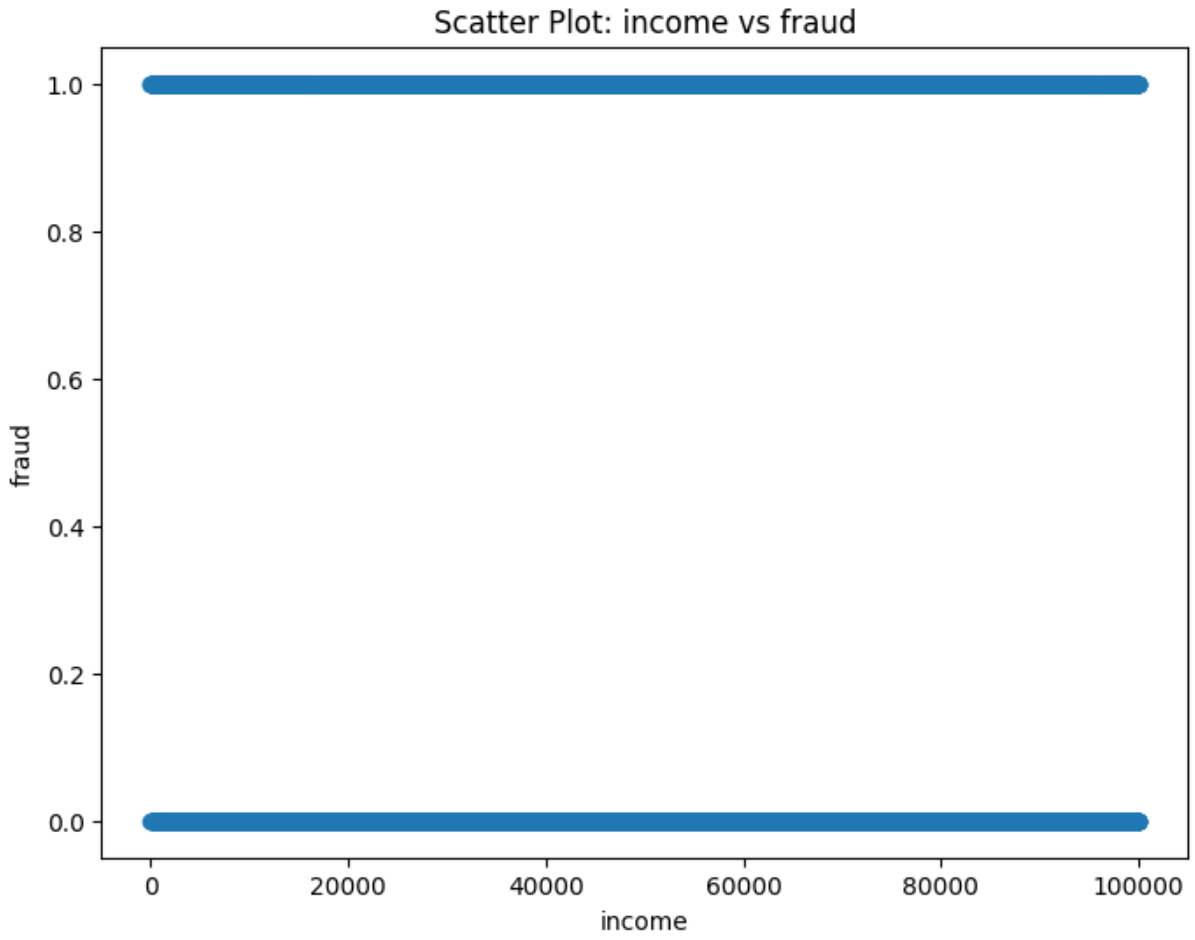

Box Plot of credit_card_number
1e15
credit_card_number
6
5
4
3
2
1
0

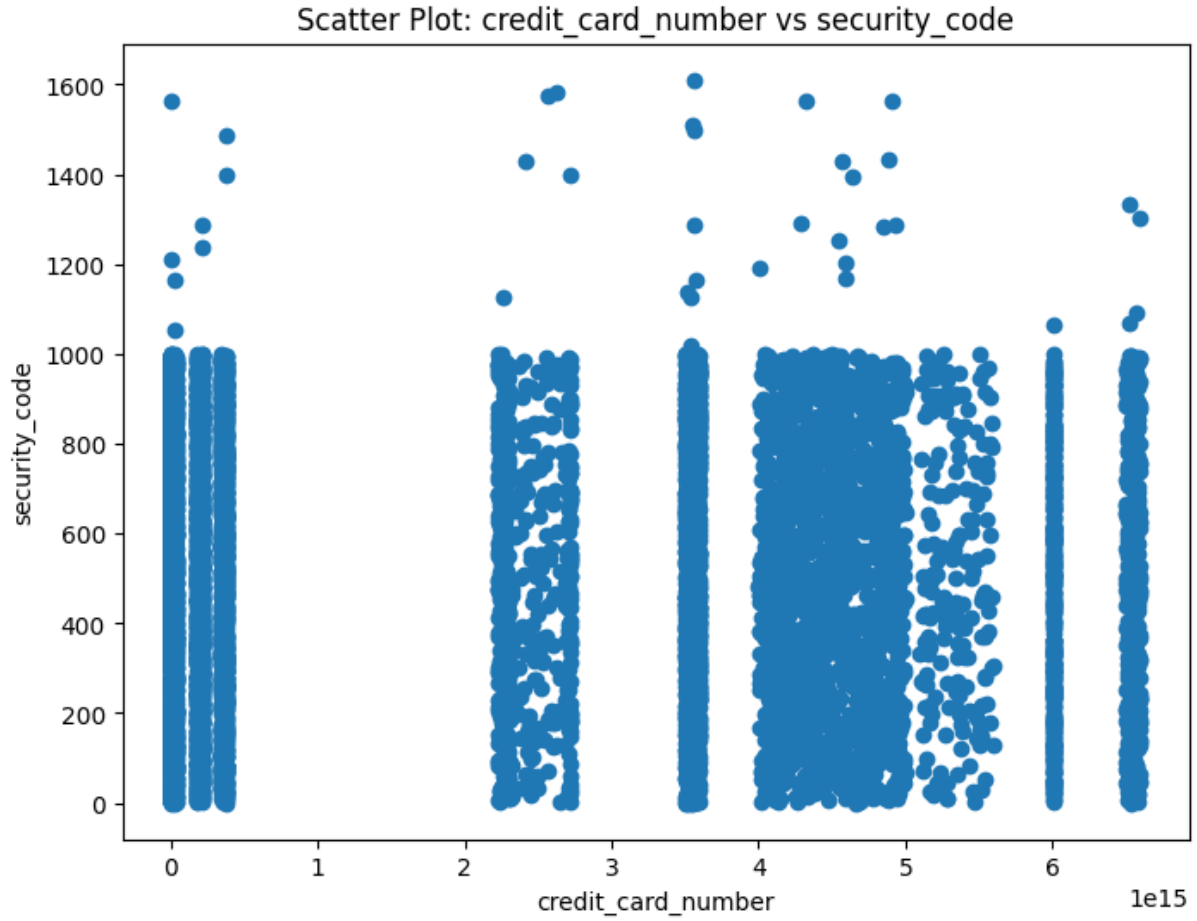

Box Plot of security_code
security_code
1600
1400
1200
1000
800
600
400
200
0

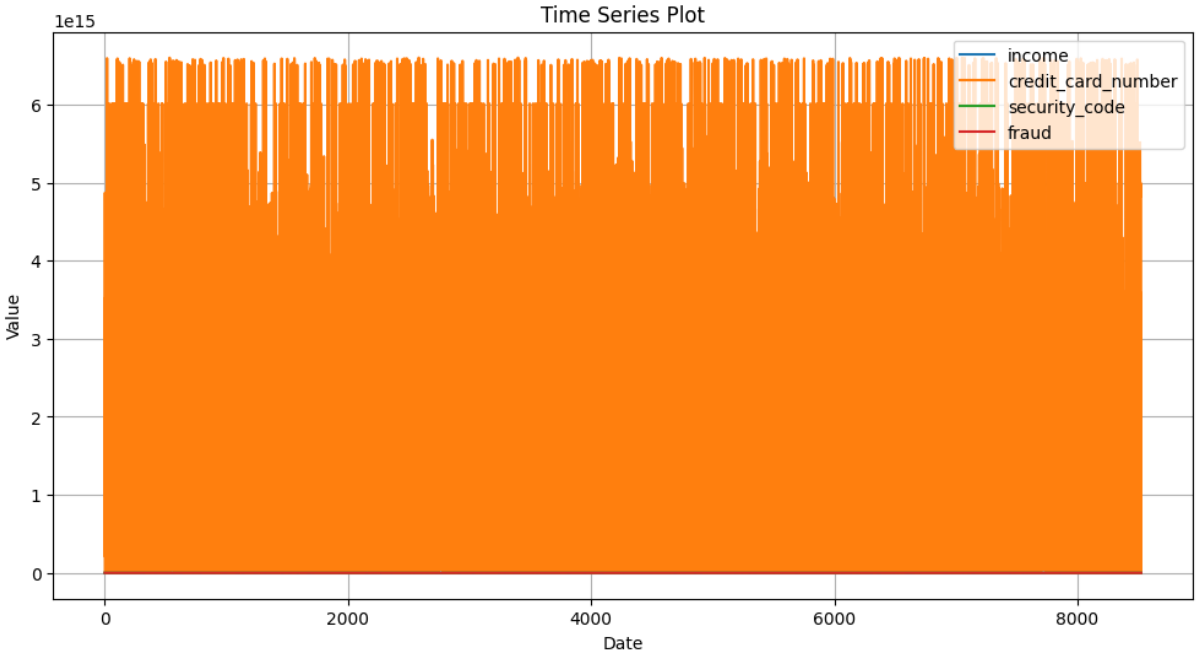

Box Plot of fraud
fraud
1.0
0.8
0.6
0.4
0.2
0.0

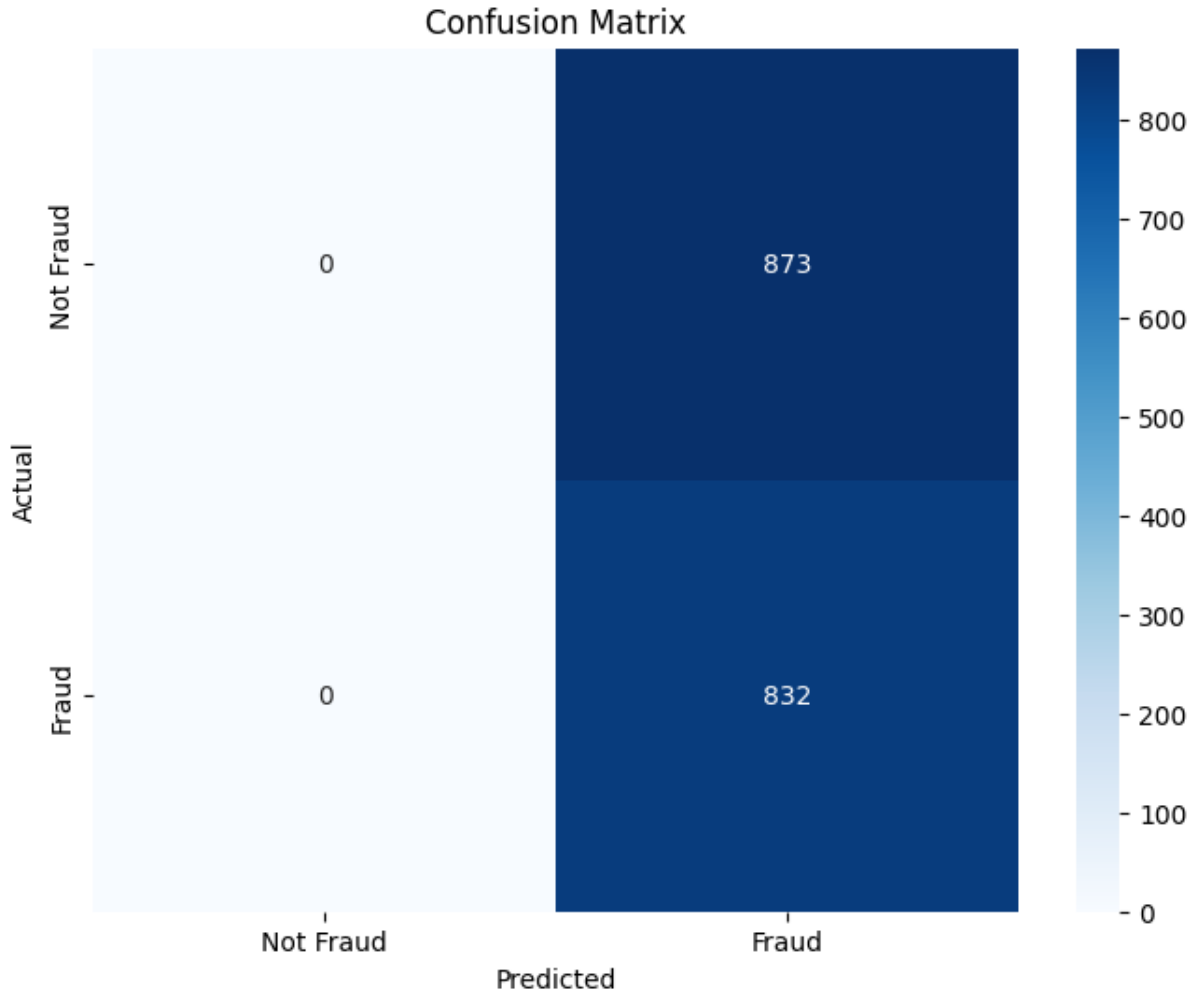

Scatter Plot: income vs credit_card_number
1e15
credit_card_number
income
0
1
2
3
4
5
6
0
20000
40000
60000
80000
100000

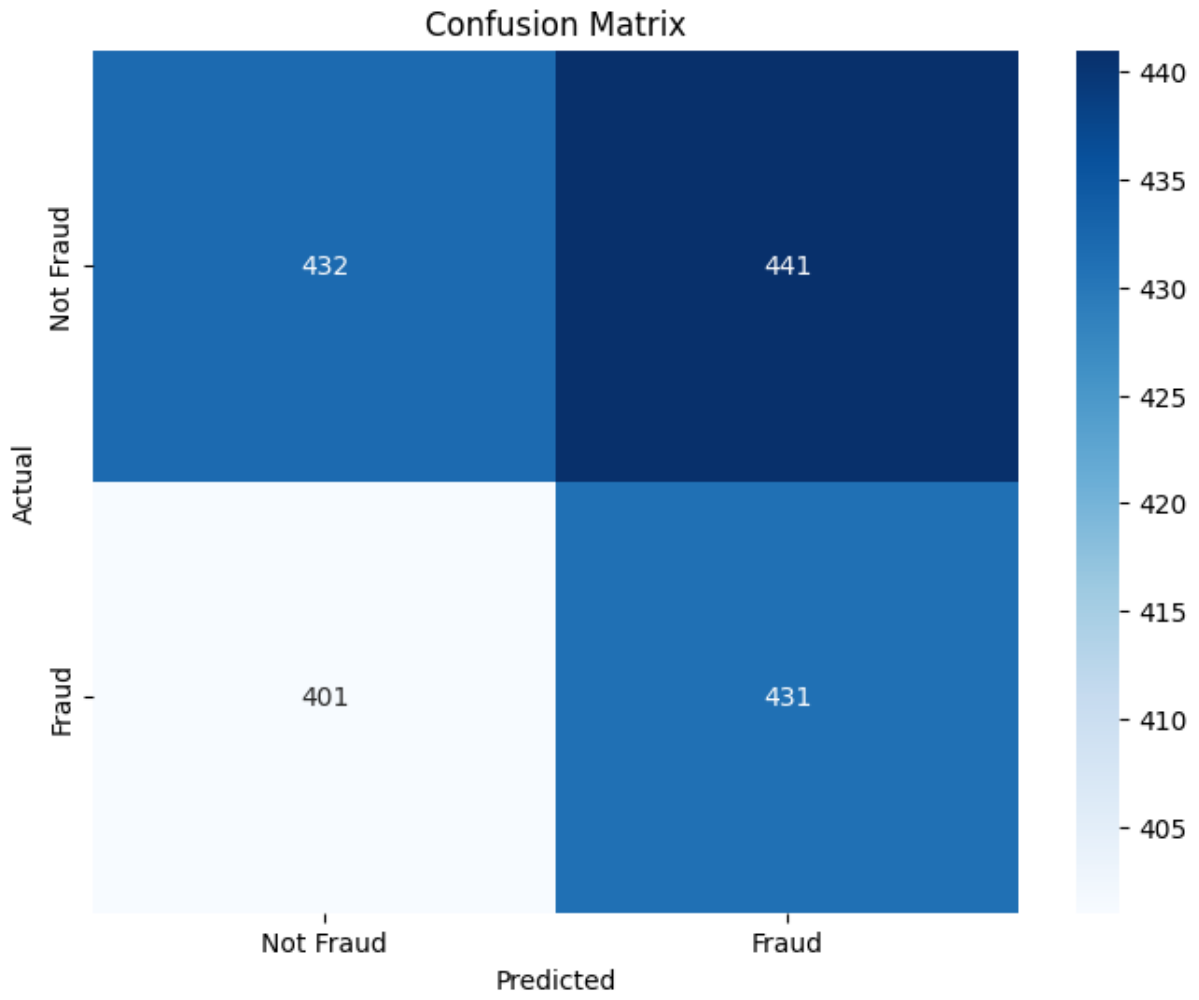

Scatter Plot: income vs security_code
security_code
income
0
200
400
600
800
1000
1200
1400
1600
0
20000
40000
60000
80000
100000

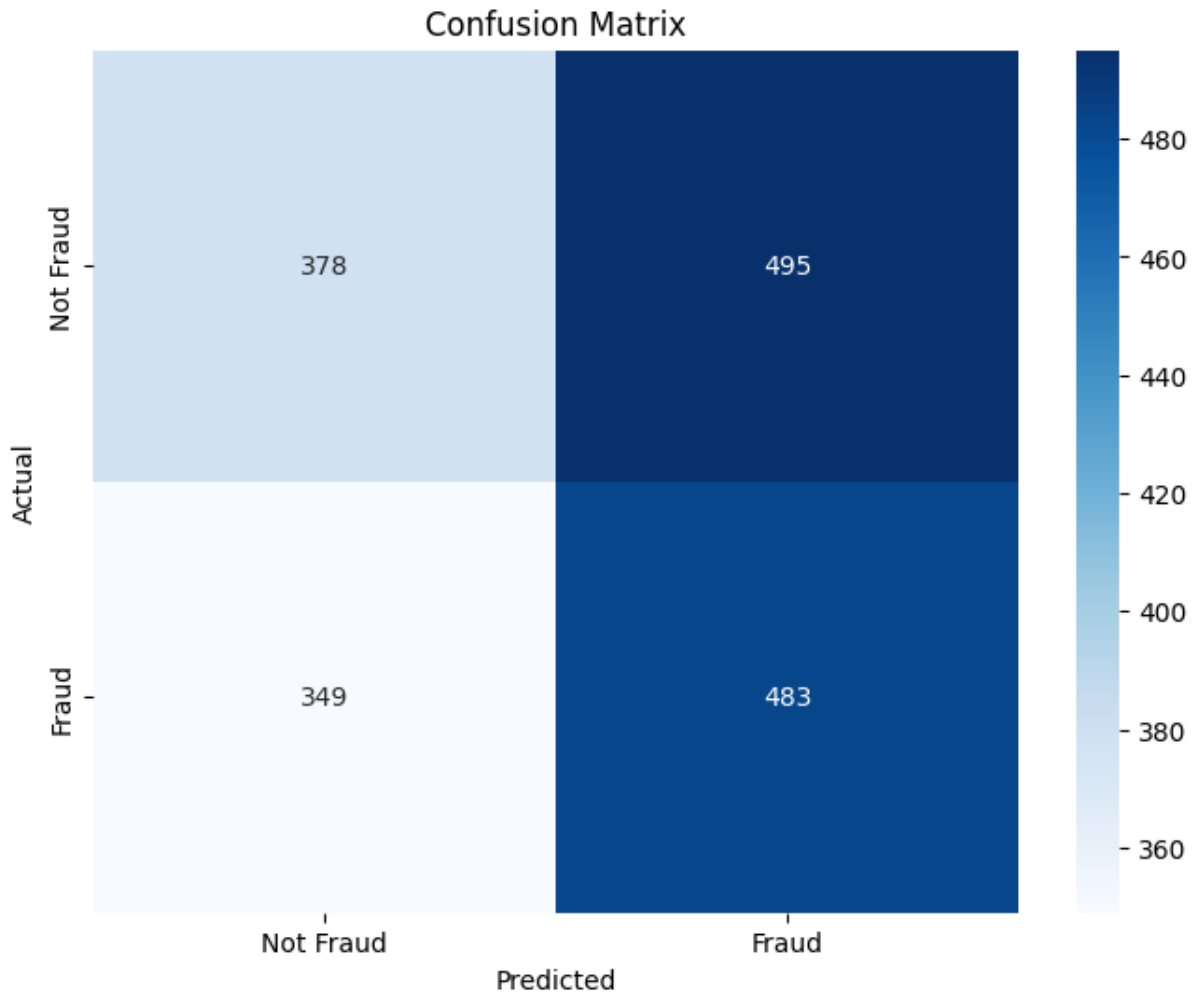

Scatter Plot: income vs fraud
fraud
income
1.0
0.8
0.6
0.4
0.2
0.0
0
20000
40000
60000
80000
100000

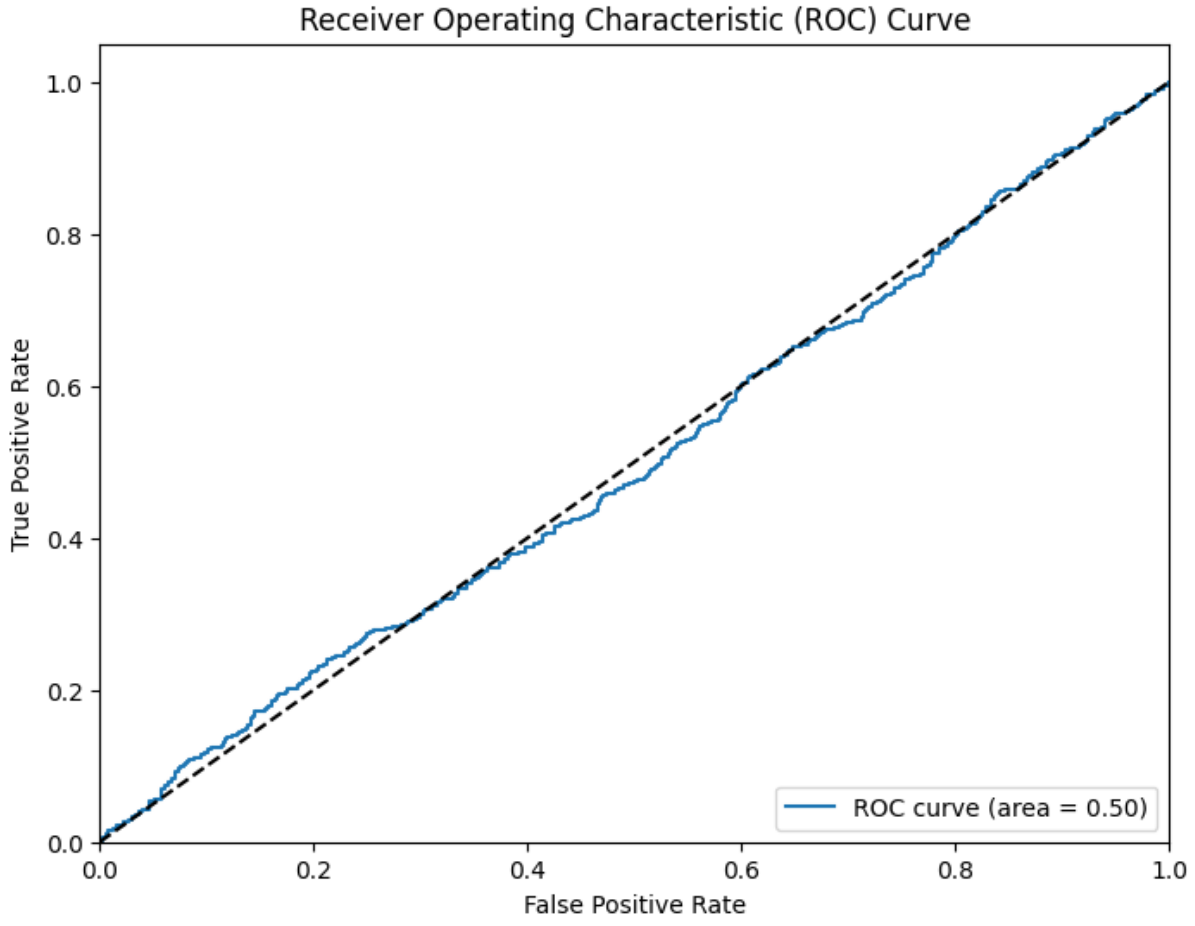

Scatter Plot: credit_card_number vs security_code
security_code
credit_card_number
1e15
1600
1400
1200
1000
800
600
400
200
0
0
1
2
3
4
5
6

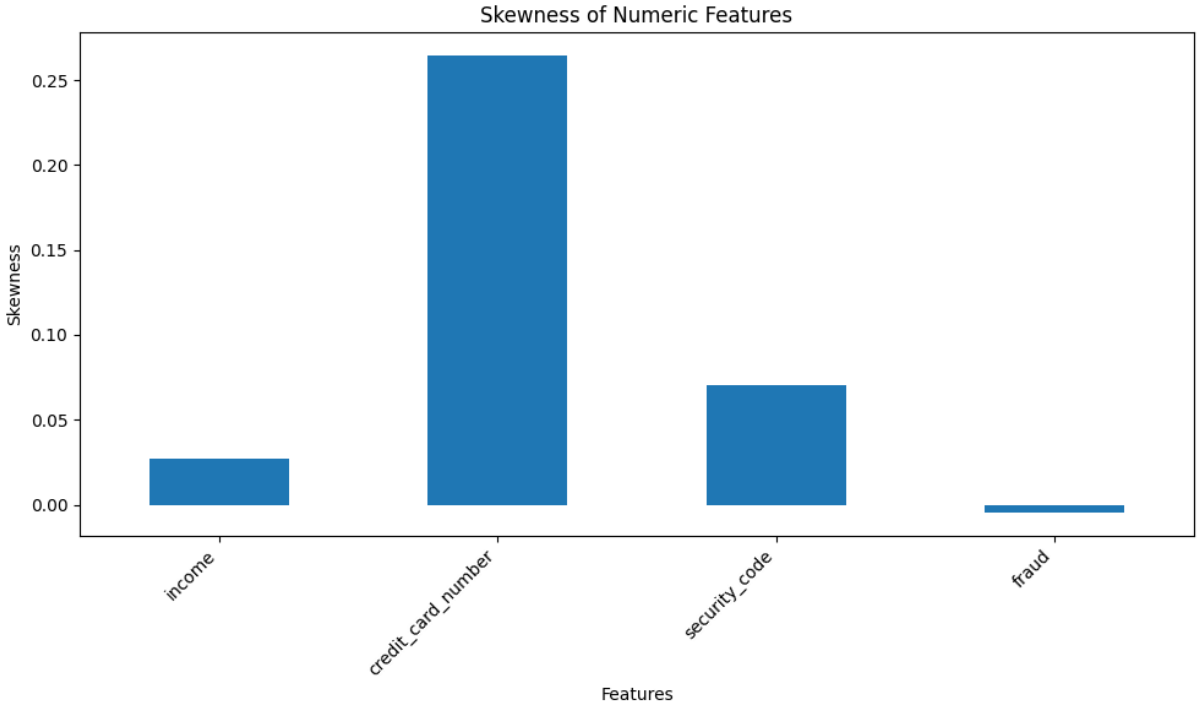

Time Series Plot
1e15
income
credit_card_number
security_code
fraud
Value
Date
0
1
2
3
4
5
6
0
2000
4000
6000
8000

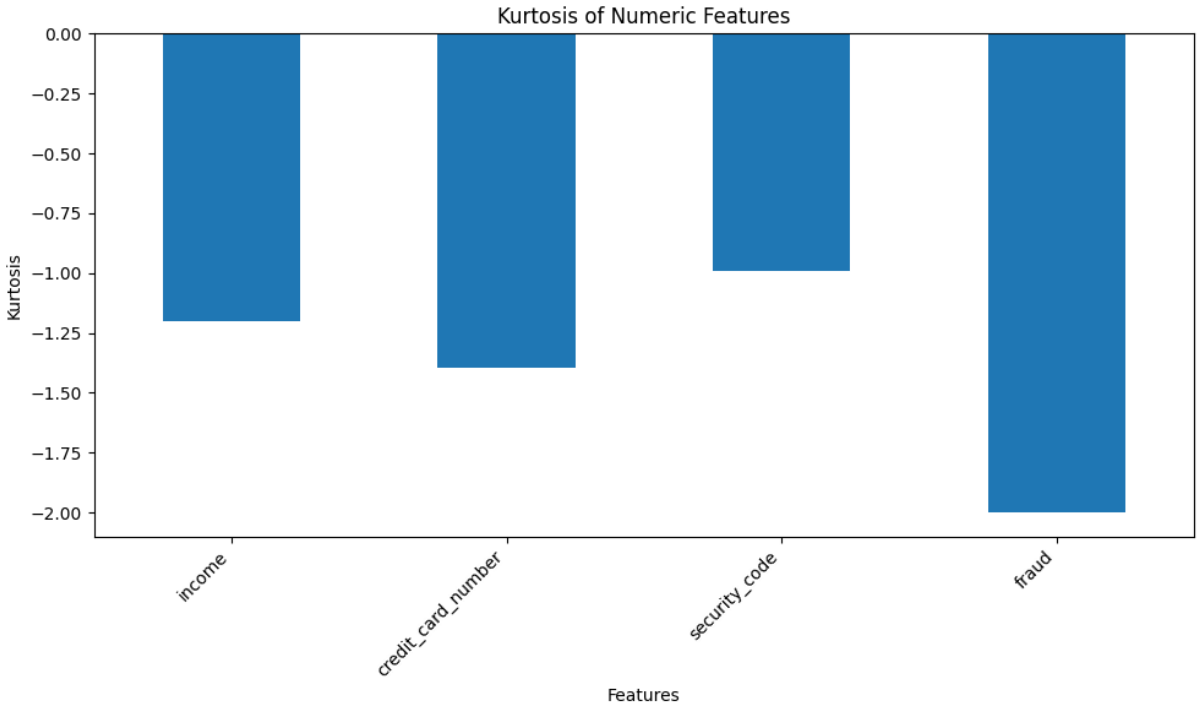

Confusion Matrix
Not Fraud
Fraud
Actual
0
873
0
832
Not Fraud
Fraud
Predicted
800
700
600
500
400
300
200
100
0


Confusion Matrix
Not Fraud
Fraud
Actual
432
441
401
431
Not Fraud
Fraud
Predicted
440
435
430
425
420
415
410
405


Confusion Matrix
Not Fraud
Fraud
Actual
378
495
349
483
Not Fraud
Fraud
Predicted
480
460
440
420
400
380
360


Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 0.50)





Key Observations:

1. **Balanced Structure with Clean Data**
    - The dataset contains **10,000 entries** with **no missing values**, providing a strong and clean foundation for analysis and model training.
2. **Imbalanced Target Variable (Expected)**
    - Fraudulent transactions (Fraud = 1) are likely much fewer than legitimate ones (Fraud = 0). This class imbalance is a common challenge in fraud detection and requires techniques like **SMOTE** or class weighting for effective modeling.

3. **Income as a Potential Discriminator**
    - Preliminary visualizations (e.g., boxplots and histograms) may show that **fraudulent transactions are more prevalent in certain income ranges**, indicating income could be a strong predictive feature.
4. **Profession-Based Trends**
    - Certain professions (like **Doctor** or **Lawyer**) may show higher or lower tendencies for fraudulent activity. This could reveal behavioral patterns that are useful for modeling.
5. **Security Code Variability**
    - The Security_code field shows a wide range, and some values might be more common among fraud cases, possibly due to predictable or reused patterns in fraud attempts.
6. **Non-informative Fields Excluded**
    - Features like Credit_card_number and Expiry were rightly excluded from modeling to avoid privacy issues and potential data leakage, as they don't contribute meaningfully to fraud prediction.
7. **Visual Patterns Support Further Analysis**
    - Boxplots and histograms of key features help reveal **outliers**, **skewed distributions**, and **potential feature transformations** (e.g., scaling) needed before model training.
8. **Strong Basis for Supervised Learning**
    - The presence of a clear binary target variable (Fraud) and multiple meaningful features make the dataset well-suited for supervised classification models.

8.Future Work:

Implementation of Classification Models

- Train and evaluate the proposed machine learning models—**Logistic Regression**, **Decision Tree**, **Random Forest**, and **XGBoost**—to predict fraudulent transactions.
- Compare performance using metrics such as **accuracy**, **precision**, **recall**, **F1-score**, and **ROC-AUC**.

Handling Class Imbalance

- Apply techniques such as **SMOTE (Synthetic Minority Over-sampling Technique)** or **class weighting** to address the imbalance between fraud and non-fraud cases, improving recall on minority (fraudulent) class.

Feature Engineering

- Explore the creation of new features such as:
    - Ratio of income to an average income benchmark
    - Encoded risk levels based on profession
    - Statistical summaries for professions (e.g., average fraud rate per profession)

Model Optimization

- Tune hyperparameters using **Grid Search** or **Random Search** to improve model performance.
- Apply **cross-validation** for more robust evaluation and generalization.

Real-Time Prediction Capability

- Extend the project to simulate **real-time fraud detection**, suitable for deployment in financial systems where quick response is critical.

Anomaly Detection Models

- Explore **unsupervised or semi-supervised** learning techniques (e.g., Isolation Forest, One-Class SVM) for fraud detection in cases where labeled data is limited.

Explainability and Interpretability

- Use tools like **SHAP (SHapley Additive exPlanations)** or **LIME** to interpret model predictions, especially to understand why a transaction is flagged as fraudulent.

Integration with Larger Pipelines

- Integrate the model into a **data pipeline** that includes automated preprocessing, model inference, and alert generation for potential fraud detection systems.

9.References:

Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2017).
Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy.
IEEE Transactions on Neural Networks and Learning Systems.
https://doi.org/10.1109/TNNLS.2016.2581137

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017).
Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning.
Journal of Machine Learning Research, 18(17), 1-5.
https://jmlr.org/papers/v18/16-365.html

Scikit-learn: Machine Learning in Python.
Pedregosa et al., Journal of Machine Learning Research, 2011.
https://scikit-learn.org/

XGBoost: A Scalable Tree Boosting System.
Chen, T., & Guestrin, C. (2016).
Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
https://doi.org/10.1145/2939672.2939785

Kaggle: Credit Card Fraud Detection Dataset.
https://www.kaggle.com/mlg-ulb/creditcardfraud

SHAP (SHapley Additive exPlanations) for Explainable ML.
Lundberg, S. M., & Lee, S. I. (2017).
A Unified Approach to Interpreting Model Predictions.
https://github.com/slundberg/shap

2. Email spam Dataset

1.Title
Email Spam Classification Using Word2Vec Embeddings and Machine Learning

2.Abstract
This project addresses the challenge of detecting spam emails through natural language processing and machine learning. By leveraging techniques like tokenization, stopword removal, and stemming, we preprocess email text for downstream modeling. Using Word2Vec embeddings and machine learning models such as Support Vector Machine (SVM) and Random Forest, the system effectively differentiates between spam and legitimate (ham) emails. The results demonstrate strong classification performance and offer a foundation for future enhancements through semantic embedding integration and real-time email filtering systems.

3,Introduction
With the exponential increase in email usage, spam detection has become a critical concern. Unsolicited emails clutter inboxes, reduce productivity, and can pose security risks. Manual filtering is inefficient and error-prone, necessitating automated systems. This project applies natural language processing (NLP) and machine learning techniques to classify email messages as spam or not spam. The pipeline incorporates preprocessing steps—including lowercasing, tokenization, stopword removal, and stemming—followed by machine learning classification using models like SVM and Random Forest.

4.Problem Statement
Spam emails are a pervasive issue affecting both individuals and organizations. Given the textual nature of email content, a robust NLP-based solution is essential. This project aims to develop a system that classifies email content as spam or ham using Word2Vec embeddings and traditional machine learning models, reducing the need for manual review and improving email handling efficiency.

5.Dataset Details

Dataset Overview:
File: email_spam.csv
Total Rows: *TBD after loading*
Columns:
label: Target variable (spam or ham)
text: The email body/content
Class Distribution: *Will be displayed after data inspection*

6.Methodology

- **Data Loading**
- The CSV file is loaded and inspected for missing values and class distribution.

Column names are:

- label (target)
- text (email content)

    - **Text Preprocessing**

- To prepare email text for modeling, the following steps are applied:

- **Lowercasing:** Normalize text casing.

- **Noise Removal:** Strip punctuation and digits using regular expressions.

- **Stopword Removal:** Remove commonly used words that carry little value for classification.

- **Stemming:** Simplify words to their root form.

- **Tokenization and Reconstruction:** Emails are tokenized and then joined back into processed strings.
- The result is stored in a new column: processed_text.

    - **Data Splitting**

- The dataset is split into training and testing sets:

- **Training Size:** 80%

- **Testing Size:** 20%

- **Target:** label

- **Input Feature:** processed_text

    - **Feature Extraction: Word2Vec**

- Using Word2Vec, email texts are embedded into dense vectors capturing semantic relationships between words. Pretrained or trained Word2Vec models help represent each email numerically.

    - **Classification Models**

- Models used:

- **Support Vector Machine (SVM)**

- **Random Forest Classifier**
- Performance is evaluated on the test set using metrics like accuracy, precision, recall,
- and F1-score.

7.Future Work

- **Clustering Quality Evaluation**: Apply Silhouette Score, Davies-Bouldin Index, or Calinski-Harabasz Index.

- **Topic Modeling**: Use LDA to uncover common topics in spam emails.

- **Real-time Filtering**: Integrate with live email systems for real-time spam filtering.

- **Advanced Embeddings**: Use BERT (e.g., Sentence-BERT) for deeper semantic understanding.

- **Visualization**: Visualize embeddings using t-SNE or UMAP; create interactive dashboards with Streamlit.

8.References

- Word2Vec: Mikolov et al. (2013). Efficient Estimation of Word Representations in Vector Space.
- BERT and SentenceTransformers: Reimers & Gurevych (2019).
- Scikit-learn documentation: https://scikit-learn.org/
- Email Spam Datasets: UCI Machine Learning Repository
- Would you like me to inspect the dataset (email_spam.csv) now and fill in exact numbers like row counts, class balance, and a few sample rows?