Credit Card Fraud Detection System

A Project Report

Submitted by

Paras Bhosale	**112103025**
Spandan Jathar	**112103054**
Abhishek Kakade	**112103063**

in Fulfillment of the Requirements for Course

Data Science Lab - TY (Computer Engineering)

Under the guidance of

Dr. Y. V. Haribhakta

COEP Technological University

DEPARTMENT OF COMPUTER ENGINEERING AND INFORMATION TECHNOLOGY,

COEP Technological University

CERTIFICATE

Certified that this project, titled "Credit Card Fraud Detection System" has been successfully completed by

Paras Bhosale	**112103025**
Spandan Jathar	**112103054**
Abhishek Kakade	**112103063**

and is approved for the partial fulfilment of the requirements for the completion of course "Data Science Lab".

SIGNATURE

Dr. Y. V. Haribhakta

Project Guide

Department of Computer Engineering

and Information Technology,

COEP Technological University,

Shivajinagar, Pune - 5.

Abstract

As the world is rapidly moving towards digitization and money transactions are becoming cashless, the use of credit cards has rapidly increased. The fraud activities associated with it have also been increasing which leads to a huge loss to the financial institutions. Therefore, weneed to analyze and detect the fraudulent transaction from the non-fraudulent ones. In this wepresent a comprehensive review of various methods used to detect credit card frauds. Here weimplement different machine learning algorithms on an imbalanced dataset such as logistic regression, naïvebayes, random forest with ensemble classifiers using boosting technique. An extensive review is done on the existing and proposed models for credit card fraud detection and has done a comparative study on these techniques. So Different classification models are applied to the data and the model performance is evaluated on the basis of quantitative measurements such as accuracy, precision, recall, f1 score, support, confusion matrix.

Contents

List of Figures

Chapter 1

Introduction

With the development and popularity of credit cards, more and more people hold credit cards in current society. According to a credit card processing company, there are 2.8 billion credit cards worldwide. However, this widespread usage also attracts fraudulent activities, leading to financial losses for both cardholders and financial institutions. Therefore, the development of effective credit card fraud detection systems is crucial to safeguarding the integrity of financial transactions. This project aims to explore techniques for detecting and preventing credit card fraud, ultimately contributing to enhanced security in financial transactions.

Chapter 2

Problem Statement

The primary problem this project addresses is the increasing occurrences of credit card fraud, which leads to significant challenges to financial institutions and cardholders. Fraudsters employ techniques to exploit vulnerabilities in the payment systems, making it imperative to develop robust fraud detection mechanisms. Challenges associated with this problem include the dynamic nature of fraudulent activities, the need for real-time detection, and the balance between accuracy and computational efficiency in fraud detection algorithms.

Chapter 3

Objectives

- Develop an efficient credit card fraud detection system which is capable of identifying fraud transactions in real-time.
- Explore various machine learning techniques for fraud detection, including supervised and unsupervised learning algorithms.
- Implement a scalable and adaptable solution that can accommodate evolving fraud patterns and adapt to new data trends.

Chapter 4

Pipeline

4.1 Data Collection

- **Data Sources Identification:** Identify sources of transaction data, including payment gateways,and online transactions.
- **Data Acquisition:** Implement processes to collect transaction data from various sources and store it in a centralized repository.

4.2 Data Preprocessing

- **Data Cleaning:** Handle missing values, outliers, and inconsistencies in the transaction data.
- **Data Transformation:** Standardize data formats, normalize numerical values, and encode categorical variables.
- **Feature Engineering:** Extract relevant features from the transaction data, such as transaction amount, timestamp, and merchant category.

4.3 Exploratory Data Analysis (EDA)

- **Data Visualization:** Explore the distribution and relationships between different features using visualizations such as histograms, scatter plots, and heatmaps.
- **Statistical Analysis:** Conduct statistical tests and analyses to identify patterns, trends, and anomalies in the transaction data.
- **Correlation Analysis:** Determined correlations between features and the target variable (fraud or non-fraud) to identify predictive relationships.

4.4 Model Selection

- **Algorithm Selection:** Choose suitable machine learning algorithms for fraud detection, such as logistic regression, decision trees, random forests, or gradient boosting.

4.5 Model Development

- **Feature Scaling:** Scale numerical features to ensure uniformity and improve model performance.
- **Hyperparameter Tuning:** Optimize model hyperparameters using techniques such as grid search or random search to improve model accuracy.

4.6 Model Training

- **Data Splitting:** Split the preprocessed data into training, validation, and test sets using techniques such as stratified sampling to maintain class balance.

- **Model Training:** Train the selected machine learning models on the training data using appropriate algorithms and hyperparameters.

4.7 Model Evaluation

- **Performance Metrics Calculation:** Evaluate model performance on the validation set using predefined evaluation metrics.
- **Cross-Validation:** Perform cross-validation to assess model generalization and robustness.
- **Model Comparison:** Compare the performance of different models and select the best-performing model for deployment.

4.8 Application Development

- **Deployment Architecture:** Design the deployment architecture for the fraud detection system, considering scalability, reliability, and real-time processing requirements.
- **Integration:** Integrate the trained model into the production environment, ensuring seamless communication with data sources and downstream systems.
- **Alerting Mechanism:** Implement an alerting mechanism to notify stakeholders when potentially fraudulent transactions are detected in real-time.

4.9 Maintenance and Updates

- **Model Monitoring:** Continuously monitor the performance of the deployed model in production, tracking key metrics such as detection rate and false positive rate.

- **Data Drift Detection:** Detect and handle data drift by periodically re-evaluating model performance and retraining the model with updated data.
- **Model Updates:** Regularly update the deployed model with new data and retrained versions to adapt to evolving fraud patterns and maintain effectiveness over time.

Chapter 5

System Design

5.1 Overview

The credit card fraud detection system aims to identify and prevent fraudulent transactions in real-time, ensuring the security of cardholder accounts and minimizing financial losses. This system leverages machine learning techniques, specifically logistic regression and decision trees, to analyze transaction data and detect fraudulent patterns.

5.2 Data Sources and Ingestion

Transaction data is sourced from kaggle , including online transactions, and mobile payments. Data ingestion processes collect and aggregate transaction data in real-time, ensuring timely analysis and detection of fraudulent activity.

5.3 Model Training

Logistic regression and decision tree models are trained on historical transaction data using supervised learning techniques. The training process involves optimizing model parameters using techniques such as gradient descent for logistic regression and tree pruning for decision trees.

5.4 Monitoring and Maintenance

The system is continuously monitored for performance and accuracy. Model performance is regularly evaluated using metrics such as precision, recall, and F1-score. Models are retrained periodically using updated transaction data to adapt to evolving fraud patterns.

Chapter 6

Methodology

The project implementation involves data preprocessing , model training , and evaluation. Techniques such as logistic regression, decision trees , random forests, and neural networks will be explored for their effectiveness in fraud detection. Feature scaling, dimensionality reduction, and outlier detection methods will be employed to enhance the performance of the models. Python programming language and libraries such as pandas, numpy, TensorFlow, matplotlib, and Keras will be utilized for implementation.

Chapter 7

Implementation

7.1 Development Environment Setup

- **Tools and Frameworks:** We use Python programming.
- **Libraries:** Necessary libraries such as scikit-learn for machine learning, Pandas for data manipulation, and Matplotlib for visualization are installed and configured.

7.2 Data Retrieval Implementation

- **Data Sources:** We retrieve transaction data from various sources such as payment readymade dataset and transaction logs.

7.3 Exploratory Data Analysis (EDA)

- **Automated Analysis:** Python scripts are used to automate data analysis tasks.
- **Visualization Tools:** We leverage libraries like Matplotlib and Seaborn to generate statistical plots and charts for identifying patterns and anomalies in transaction data.

7.4 Model Configuration and Training

- **Model Selection:** We choose machine learning algorithms such as logistic regression and decision trees for fraud detection.
- **Parameter Tuning:** Models are configured with initial parameters and trained using historical transaction data. Parameters are adjusted based on ongoing analysis to improve accuracy.

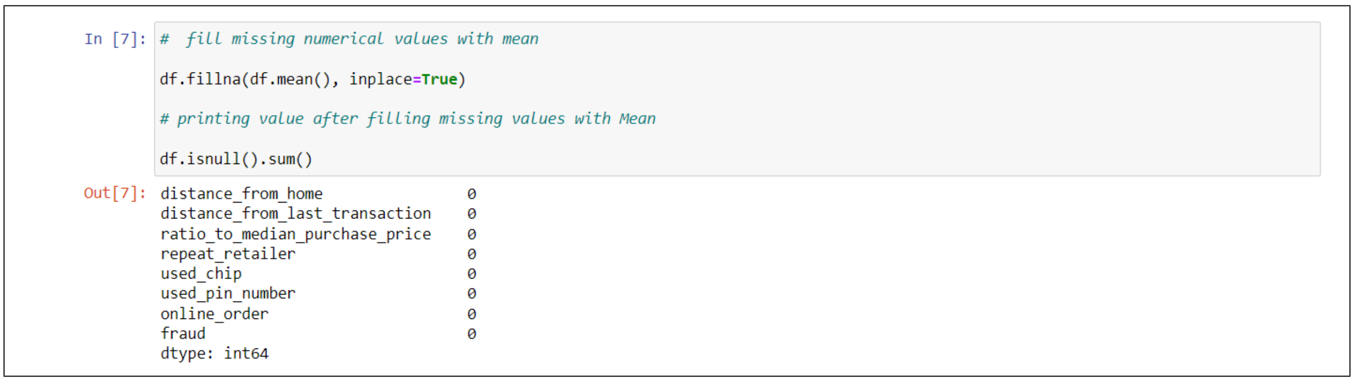
7.5 Model Validation and Optimization

- **Validation Phase:** Trained models are tested against unseen data to evaluate their performance.
- **Optimization Strategies:** We optimize models to enhance their performance and efficiency, ensuring accurate fraud detection while minimizing false positives.

Chapter 8

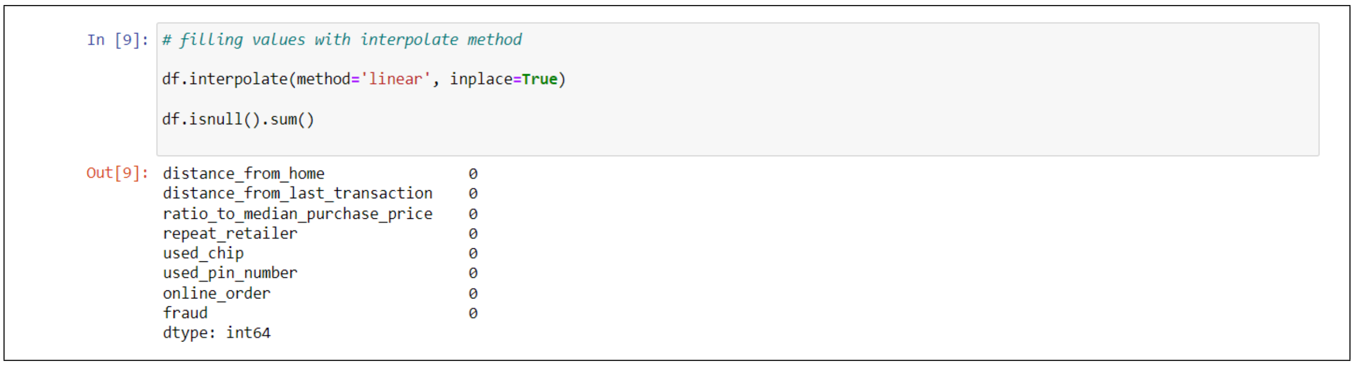
Results and Analysis

8.1 Handling The Missing Values



```
In [7]: #  fill missing numerical values with mean

        df.fillna(df.mean(), inplace=True)

        # printing value after filling missing values with Mean

        df.isnull().sum()

Out[7]: distance_from_home                0
        distance_from_last_transaction    0
        ratio_to_median_purchase_price    0
        repeat_retailer                   0
        used_chip                         0
        used_pin_number                   0
        online_order                      0
        fraud                             0
        dtype: int64
```

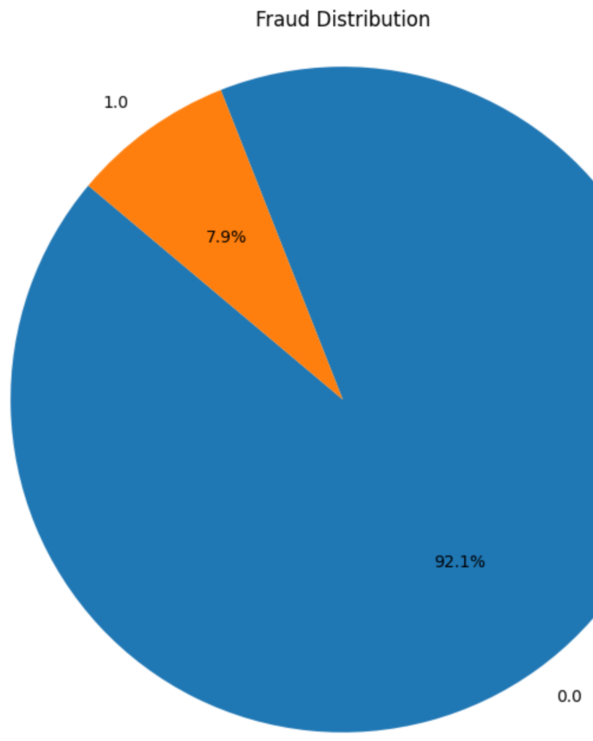
Figure 8.1: Handle_null_1



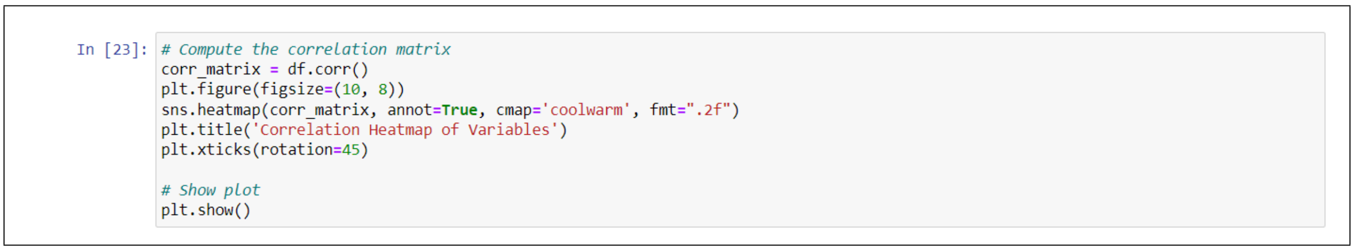
```
In [9]: # filling values with interpolate method

        df.interpolate(method='linear', inplace=True)

        df.isnull().sum()

Out[9]: distance_from_home                0
        distance_from_last_transaction    0
        ratio_to_median_purchase_price    0
        repeat_retailer                   0
        used_chip                         0
        used_pin_number                   0
        online_order                      0
        fraud                             0
        dtype: int64
```

Figure 8.2: Handle_null_2

8.2 Exploratory Data Analysis



8.3 Correlation Matrix Showing Relationships Between Features



```
In [23]: # Compute the correlation matrix
         corr_matrix = df.corr()
         plt.figure(figsize=(10, 8))
         sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f")
         plt.title('Correlation Heatmap of Variables')
         plt.xticks(rotation=45)

         # Show plot
         plt.show()
```

Figure 8.3: Heatmap Code

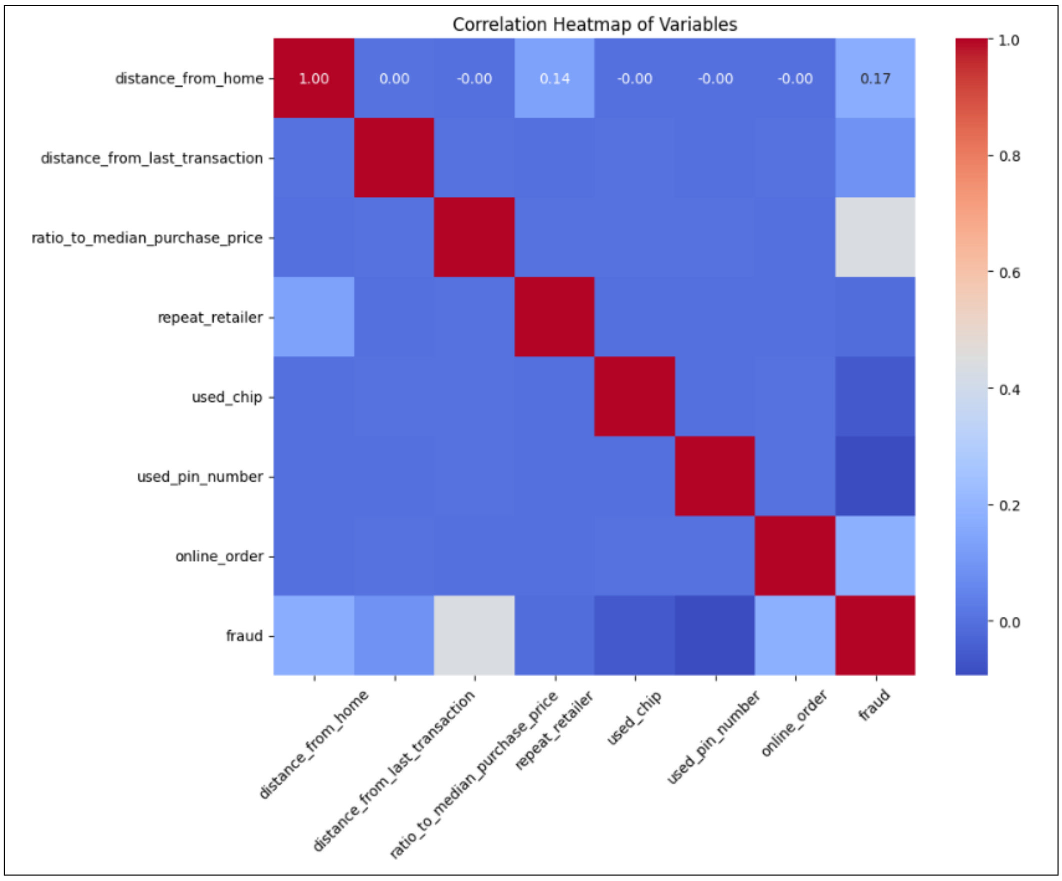


Figure 8.4: Heatmap

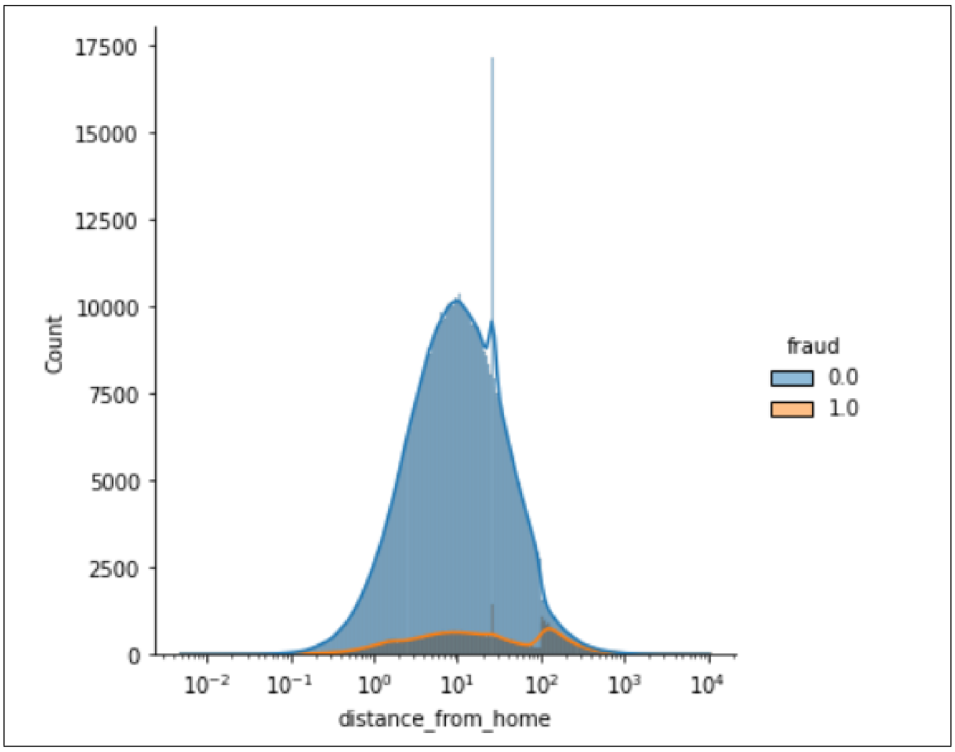


Figure 8.5: distance_from_home Vs fraud plot

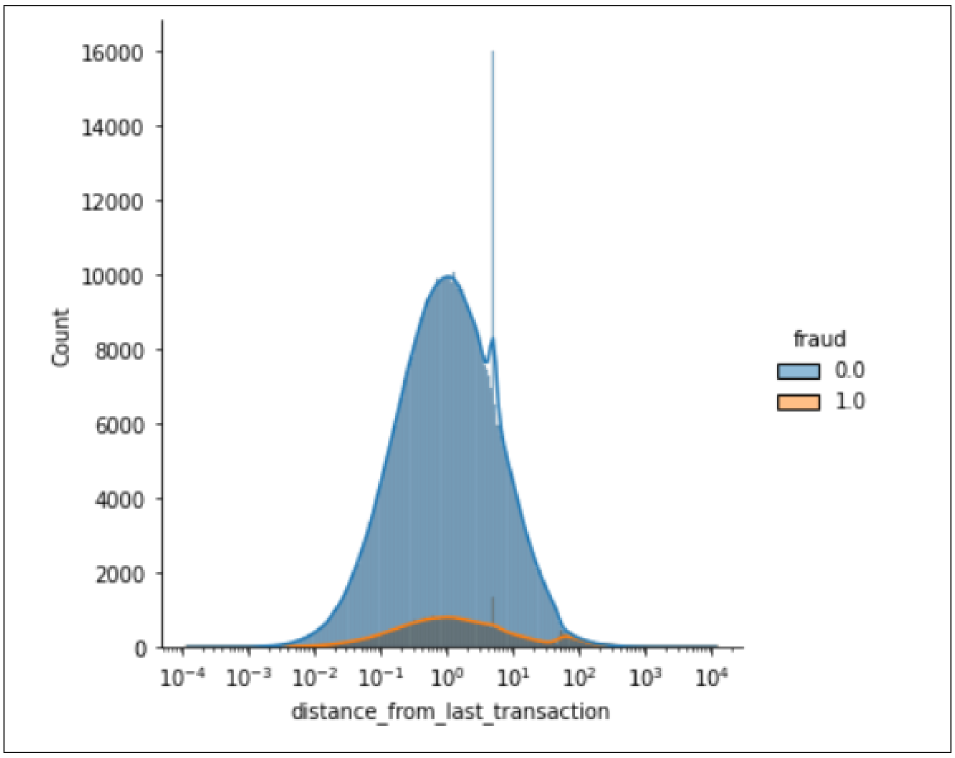


Figure 8.6: distance_from_last_transaction Vs fraud plot

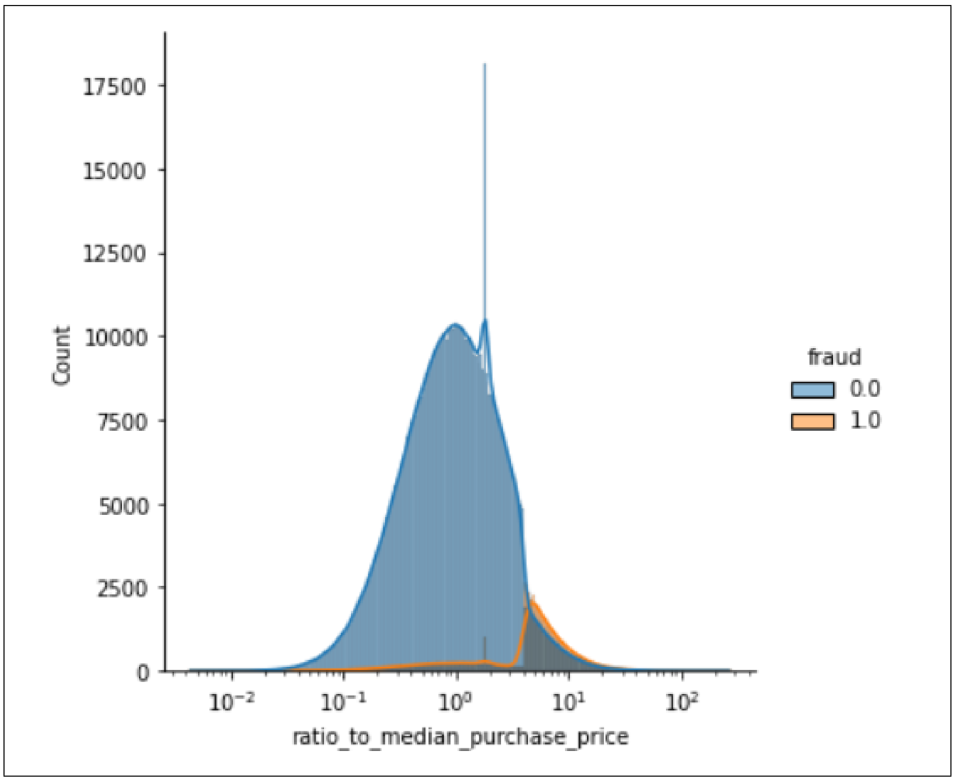


Figure 8.7: ratio_to_median_purchase_price Vs fraud plot

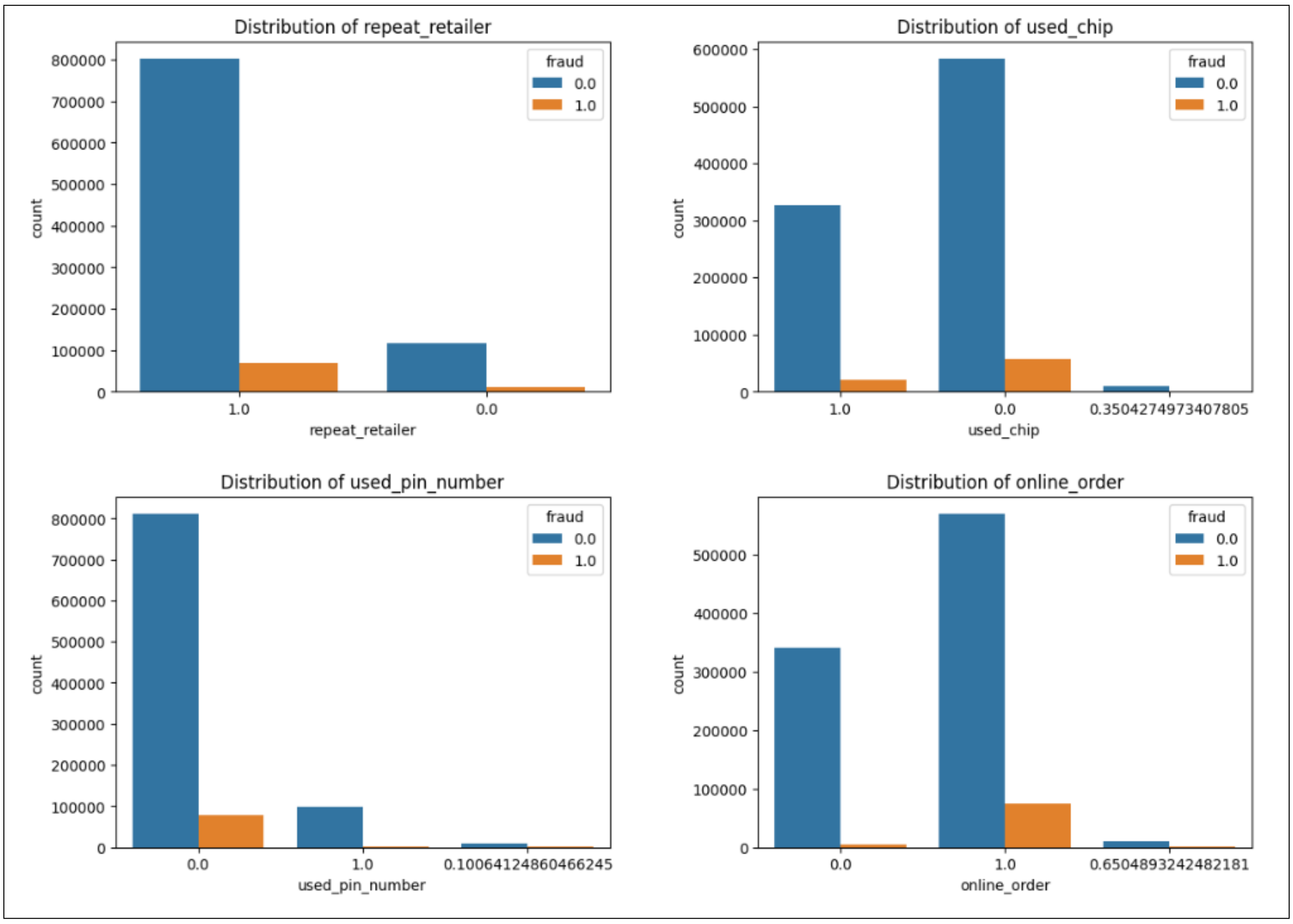


Figure 8.8: Distribution of repeat_retailer, used_chip, used_pin_number, online_order

- Most of the fraud transactions occurred with repeated retail transactions.
- Most of the fraud transactions didn't use chip to be successful. (They must have preferred online media).
- There is no fraud transaction using pin number (Prefer using pin while withdrawing money).
- Most of the fraud transactions have occurred in the online orders (Beware of risks in internet banking).

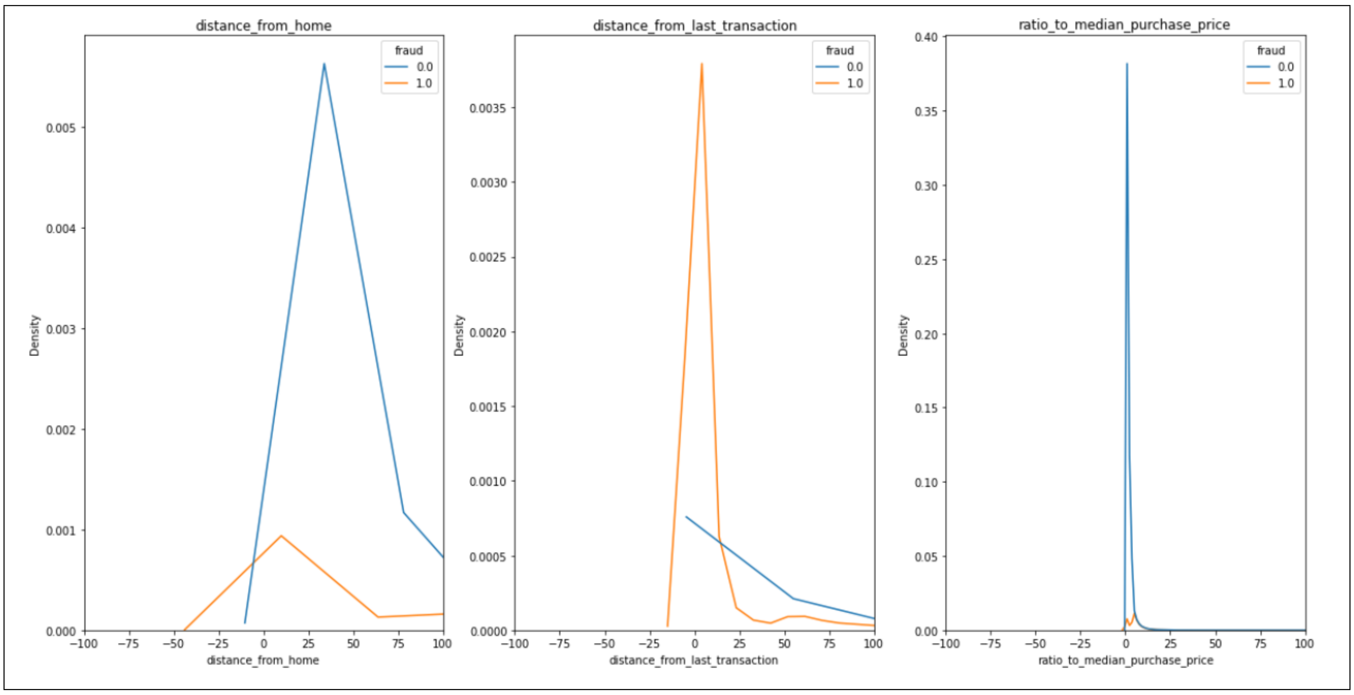
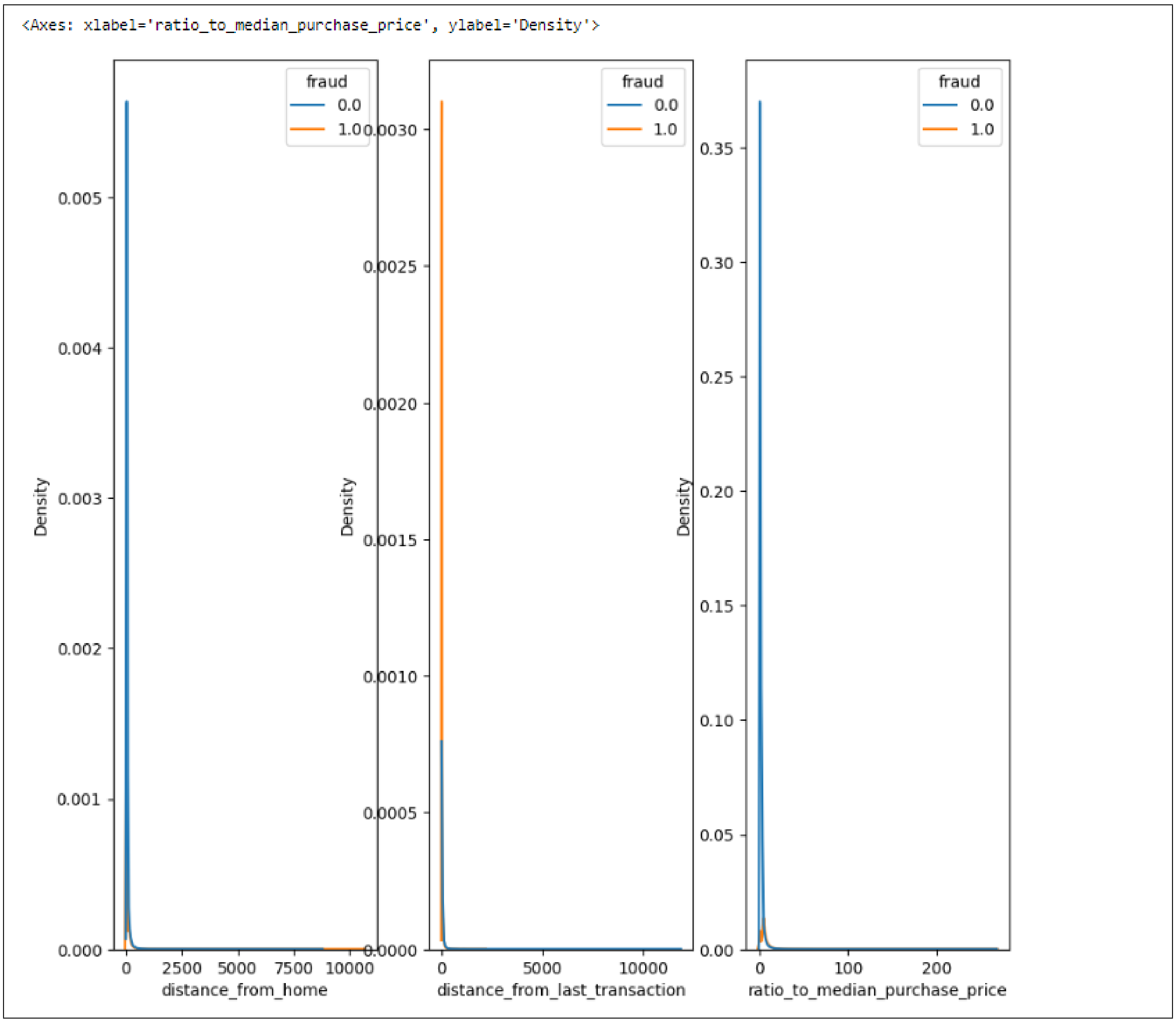


Figure 8.9: KDEplot$_1$

It is really insightful to see that most of the fradulent card transaction occurs within a range of 50km from the victim and also it can be seen that for most of the fradulent cases, the ratio_to_median_purchase is nearly equalfor both fradulent and genuine transactions, which means, the fraud doesn't lets the victim feel wrong by withdrawing a heavy amount of money at once, rather small transactions will be made. Insighful with numerical features yet.

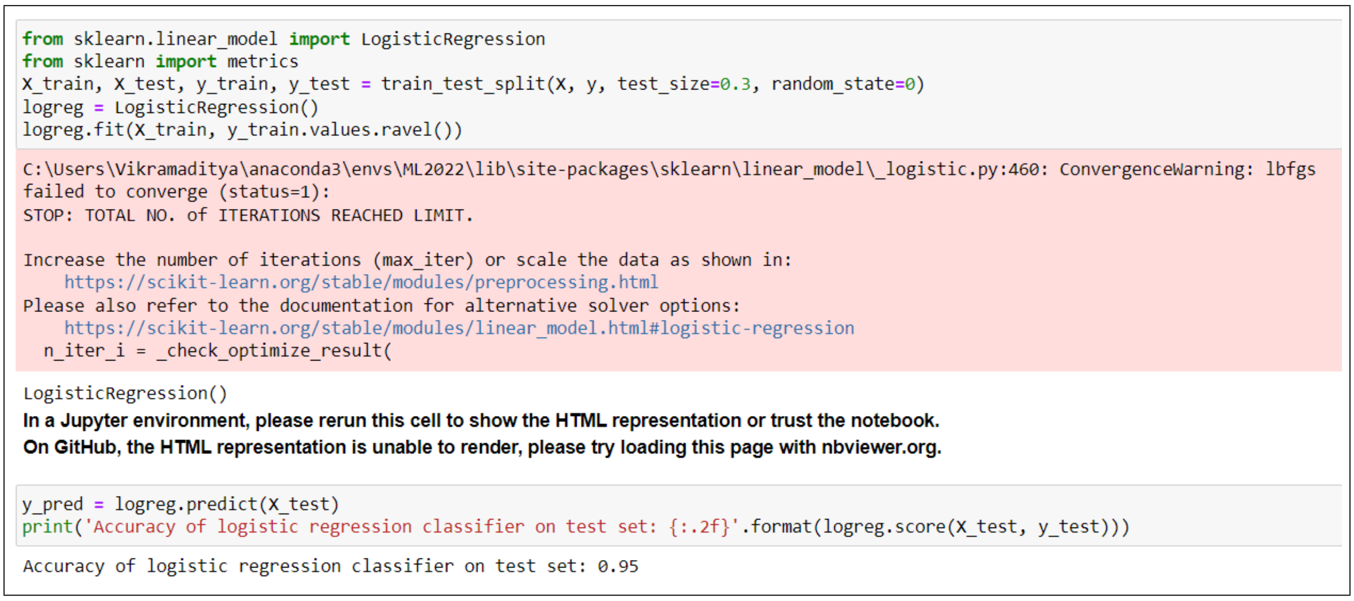


```
<Axes: xlabel='ratio_to_median_purchase_price', ylabel='Density'>
```

Figure 8.10: KDEplot$_2$

From the above plots it is clearly visible that all the features are highly SKEWED to one side. Lets in the believe in the fact, people use cards for their daily needs, obviously the distances are near and frequent. But also there are people or fraud transactions happening from a far off place. Now lets check the standard deviation of each : numerical_features = ['distance_from_home','distance_from_last_transaction' ,'ratio_to_median_purchase_price']

8.4 Model Selection and Performance Metrics



```
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
logreg = LogisticRegression()
logreg.fit(X_train, y_train.values.ravel())
```

```
C:\Users\Vikramaditya\anaconda3\envs\ML2022\lib\site-packages\sklearn\linear_model\_logistic.py:460: ConvergenceWarning: lbfgs
failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
LogisticRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
y_pred = logreg.predict(X_test)
print('Accuracy of logistic regression classifier on test set: {:.2f}'.format(logreg.score(X_test, y_test)))
```

```
Accuracy of logistic regression classifier on test set: 0.95
```

Figure 8.11: Code for Model

95% is a pretty decent accuracy score, although advancements will be ongoing to make it further. Now lets have a look at the False positive cases. This is very important as these are the cases where the transactions are suspected although they are genuine. For this purpose we will be using a simple confusion matrix.

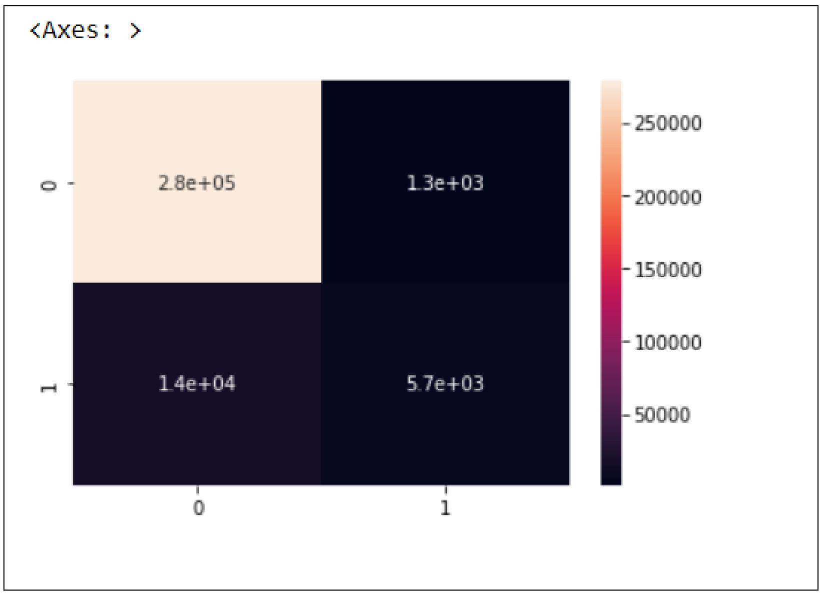
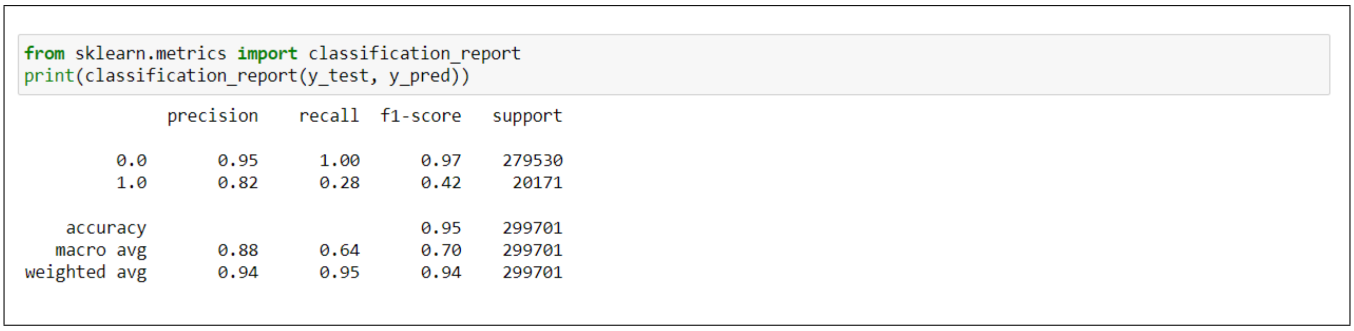


Figure 8.12: Confusion Matrix

We can se only 1.4e+03 i.e 1400 transactions were False positive cases, and it is very very less compared to our entire dataset. Great! So, logistic regression worked pretty well in terms of precision also. Still lets get the mathematical stats.

8.5 Result:



```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

         0.0       0.95      1.00      0.97    279530
         1.0       0.82      0.28      0.42     20171

    accuracy                           0.95    299701
   macro avg       0.88      0.64      0.70    299701
weighted avg       0.94      0.95      0.94    299701
```

Figure 8.13: Classification Report

- **Precision:** Precision is the ratio of correctly predicted positive observations to the total predicted positives.

  In this case:

  - Precision for class 0.0: 95% of the instances predicted as class 0.0 were actually class 0.0.
  - Precision for class 1.0: Only 82% of the instances predicted as class 1.0 were actually class 1.0.

- **Recall (Sensitivity):** Recall is the ratio of correctly predicted positive observations to the all observations in actual class.

  In this case:

  - Recall for class 0.0: 100% of the instances that were actually class 0.0 were predicted as class 0.0.
  - Recall for class 1.0: Only 28% of the instances that were actually class 1.0 were predicted as class 1.0.

- **F1-score:** The F1-score is the weighted average of Precision and Recall. It is a good way to show that a classifier has a good value for both recall and precision.

In this case:

  - F1-score for class 0.0: 97%
  - F1-score for class 1.0: 42%

- **Support:** Support is the number of actual occurrences of the class in the specified dataset. In this case, there are 279530 instances of class 0.0 and 20171 instances of class 1.0.
- **Accuracy:** Overall, the accuracy of the model is 95%, meaning 95% of the predictions made by the model are correct.
- **Macro avg:** This is the average of precision, recall, and F1-score for both classes. It gives equal weight to both classes regardless of their size.

  In this case:

  - Macro avg precision is 88.5%
  - Macro avg recall is 64%
  - Macro avg F1-score is 69.5%
- **Weighted avg:** This is the weighted average of precision, recall, and F1-score for both classes. It considers the number of occurrences of each class for calculation, thus favoring the majority class.

  In this case:

  - Weighted avg precision is about 93.84%
  - Weighted avg recall is about 94.85%
  - Weighted avg F1-score is about 93.97%

Chapter 9

Conclusion

With the popularity of credit cards, cash has gradually faded out of public life, and credit card transactions have become one of the most common transaction methods worldwide. Overall, the principal target of this study is to figure out what variables are relative to credit card fraud and use logistic regression as the primary statistical model and three supervised machine learning models to predict the probability of fraud occurring after cleaning and transforming the significant independent variables. The original seven independent variables contain binary variables, like repeat retailer, used chip, pin number, and online order, and three numerical variables. As can be seen from the Exploratory Data Analysis section, the higher or more outrageous the amount of credit card transactions, and the farther the transaction takes place from home or from the place where the previous transaction is, the more likely it is a fraud transaction. Also, online orders are more prone to fraud. The use of pins and chips can effectively reduce the occurrence of fraud. Additionally, the correlation plot indicates that RR has no relationship with fraud. The bar plot of the fraud statistics proved that credit card is generally a safe purchase. After all, the fraud rate is only 8.74one out of every ten credit card transactions. However, the number of frauds remains particularly terrifying due to the large base of credit card transactions worldwide, such as 10 billion.

Chapter 10

Future Work

Detection, we did end up creating a system that can, with enough time and data, get very close to that goal. As with any such project, there is some room for improvement here. The very nature of this project allows for multiple algorithms to be integrated together asmodules and their results can be combined to increase the accuracy of the final result. This model can further be improved with the addition of more algorithms into it. However, the output of these algorithms needs to be in the same format as the others. Once that condition is satisfied, the modules are easy to add as done in the code. This provides a great degree of modularity and versatility to the project. More room for improvement can be found in the dataset. As demonstrated before, the precision of the algorithms increases when the size of dataset is increased. Hence, more data will surely make the model more accurate in detecting frauds and reduce the number of false positives. However, this requires official support from the banks themselves.

Chapter 11

References

- https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdfdoi=0419c275f05841
  d87ab9a4c9767a4f997b61a50e
- https://www.researchgate.net/profile/Sarika-Jain-
  2/publication/332264296_A_comparative_analysis_of_various_credit_card_fraud_d
  etection_techniques/links/5d46e13f92851cd046a0b3ec/A-comparative-analysis-of-various-
  credit-card-fraud-detection-techniques.pdf

Chapter 12

Appendix

https://www.kaggle.com/datasets/kartik2112/fraud-detection