**AI TOOLS & TECHNIQUES FOR CYBERSECURITY**

Year 2 Semester 4

**SCHOOL OF INFOCOMM TECHNOLOGY**

Diploma in Cyber Security & Digital Forensics

**Assignment Writeup**

|  |  |
| --- | --- |
| **Tutorial Group:** | **P01** |
| **Tutor:** | **Mohamed Saifulamri Omar** |
| **Date of Submission:** | **02/02/2025** |

|  |  |  |
| --- | --- | --- |
| **Student Number** | **Student Name** | **Grade** |
| **S10257626H** | **Nicholas Ng** |  |
| **S10258327C** | **Joshua Leong** |  |

[1. Introduction 3](#_Toc189430714)

[2. Problem Statement 3](#_Toc189430715)

[3. Dataset Used 4](#_Toc189430716)

[4. Group Member Contributions 6](#_Toc189430717)

[5. Data Research 7](#_Toc189430718)

[5.1 Data Validation & Cleaning 7](#_Toc189430719)

[5.2 Exploratory Data Analysis 12](#_Toc189430720)

[5.3 Feature Engineering 14](#_Toc189430721)

[5.4 EDA of continuous numerical features 15](#_Toc189430722)

[5.5 Logistic Regression 16](#_Toc189430723)

[5.5.1 Encoding of non-numeric data & further cleaning 16](#_Toc189430724)

[5.5.2 Training of the Model 17](#_Toc189430725)

[5.6 Model Evaluation 18](#_Toc189430726)

[6. Findings 22](#_Toc189430727)

[7. Conclusion 23](#_Toc189430728)

[8. Appendix 24](#_Toc189430729)

[9 References 32](#_Toc189430730)

# 1. Introduction

A credit card is a thin rectangular piece of plastic or metal issued by a bank or financial services company that allows cardholders to borrow funds to pay for goods and services with merchants that accept cards for payment. Credit cards impose the condition that cardholders pay back the borrowed money, plus any applicable interest and any additional agreed-upon charges, either in full by the billing date or overtime.

With the widespread adoption of digital payments and e-commerce, credit card fraud has become an increasingly prevalent threat in today's interconnected world. This form of fraud encompasses various deceptive practices aimed at unlawfully obtaining financial information or resources associated with credit cards.

Tactics such as card skimming, phishing, online transaction fraud and online frauds exploit vulnerabilities in payment systems and take advantage of lack of awareness, jeopardizing the security and financial well-being of individuals and businesses alike. This introduction explores the methods and impacts of credit card fraud, highlighting the importance of vigilance and robust security measures in safeguarding against malicious activities in the digital age.

Credit card frauds can happen anywhere and at any time, anyone is a victim if they are not careful enough and do not practice safe practices when dealing/using credit cards. In this document, we will be going through a dataset which shows many credit card transactions containing legitimate and fraudulent transactions from the duration 1st Jan 2019 - 31st Dec 2020. It covers the credit cards of 1000 customers doing transactions with a pool of 800 merchants.

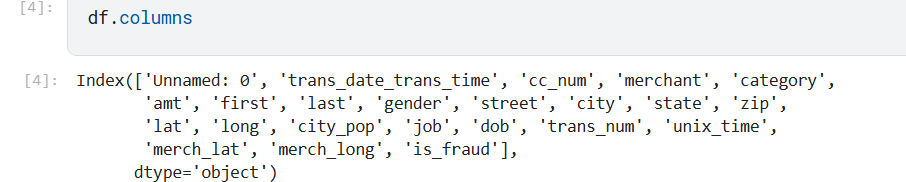
# 2. Problem Statement

With the rising number of credit card frauds recently, it is imperative that everyone is aware of the risks of these attacks. Hence, we aim to develop a machine learning model that can predict with an accuracy of at least 90% when a transaction is a possible fraud or not. We will be conducting supervised machine learning by providing the program with an open-source dataset containing a list of over 1 million previous credit card transactions, containing both legitimate and fraudulent transactions after cleaning and organising the data of irregularities and unnecessary data (e.g. gender, street, etc.). As the number of victims falling for such fraud, we aim to complete machine learning in 2 weeks.

# 3. Dataset Used

The dataset that we have chosen is **fraudTrain.csv** that has been downloaded from Kaggle. It contains information about credit card fraud so that we can understand what type of transactions and any potential signs that the transaction is a fraud. This dataset contains fraudulent transactions in the past which can be used for us to train the machine and detect fraud relating to credit card transactions. In the dataset, there are 23 columns of information:

1. *Unnamed:0*
   1. *This column describes which row we are in for each transaction. The first row has the value of “0” while the second row has the value of “1”*
2. *trans\_date\_trans\_time*
   1. *This column has the information of which date the transaction occurred and the time when the transaction occurred. For example: “2019-01-01 00:00:18”*
3. *cc\_num*
   1. *This column contains information about credit card numbers. For example: “2703186189652095”*
4. *merchant*
   1. *This column contains information about the business that is accepting credit card transactions/payments for their goods and services. For example: “fraud\_Rippin, Kub and Mann”*
5. *category*
   1. *This column contains information about the reason for the transaction. For example: “grocery\_pos”*
6. *amt*
   1. *This column contains information about how much money was transacted. For example: “4.97”*
7. *first*
   1. *This column contains information about the credit card owner’s first name. For example: “Jennifer”*
8. *last*
   1. *This column contains information about the credit card owner’s last name. For example: “Banks”*
9. *gender*
   1. *This column contains information about the credit card owner’s gender. For example: “F”*
10. *street*
    1. *This column contains information about the credit card owner’s home details. For example: “561 Perry Cove”*
11. *city*
    1. *This column contains information about the credit card owner’s city details. For example: “ Morovian Falls”*
12. *state*
    1. *This column contains information about the credit card owner’s state details. For example: “NC”*
13. *zip*
    1. *This column contains information about the credit card owner’s zip details. For example: “28654”*
14. *lat*
    1. *This column contains information about the credit card owner’s location latitude. For example: “36.0788”*
15. *long*
    1. *This column contains information about the credit card owner’s location longitude. For example: “-81.1781”*
16. *city\_pop*
    1. *This column contains information about credit card owners’ city population. For example: “3495”*
17. *job*
    1. *This column contains information about credit card owners’ job. For example: “Psychologist, counselling”*
18. *dob*
    1. *This column contains information about credit card owners’ date of birth. For example: “1988-03-09"*
    2. *In YYYY-MM-DD format*
19. *trans\_num*
    1. *This column contains information about the unique transaction number for every transaction. For example: “0b242abb623afc578575680df30655b9”*
20. *unix\_time* 
    1. *This column contains information about time of transaction in Unix (usually unique). For example: “1325376018”*
21. *merch\_lat*
    1. *This column contains information about merchant latitude. For example: “36.011293”*
22. *merch\_long*
    1. *This column contains information about the merchant longitude. For example: “-82.048315”*
23. *is\_fraud*
    1. *This column contains information on whether the transaction is fraud or no. For example: “0”*
    2. *1 – Is fraud, 0 – Is not a fraud*



# 4. Group Member Contributions

|  |  |
| --- | --- |
| Nicholas Ng | Joshua Leong |
| * 1) Introduction * 3) Dataset Used * 5.1) Data Validation and Cleaning * 5.2) Exploratory data analysis * 5.3) Feature Engineering * 5.4) EDA of continuous numerical features | * 2) Problem Statement * 5.5) Logistic Regression * 5.5.1 + 5.5.2) Encoding of non-numeric data and further cleaning + Training of the Model * 5.6) Model Evaluation * 6) Findings * 7) Conclusion |
| Coded from the start till in the picture shown. | Coded from the picture till the end. |

# 5. Data Research

## 5.1 Data Validation & Cleaning

Data validation is crucial in machine learning because poor quality data can lead to inaccurate model predictions, essentially resulting in a "garbage in, garbage out" scenario; by validating data before training a model, we have to ensure the data is clean, consistent, and suitable for learning, ultimately leading to a more reliable and accurate machine learning model.

To validate data, we have to ensure that there are no missing values in any columns or rows.

To check for missing values, we used .isnull().sum() function

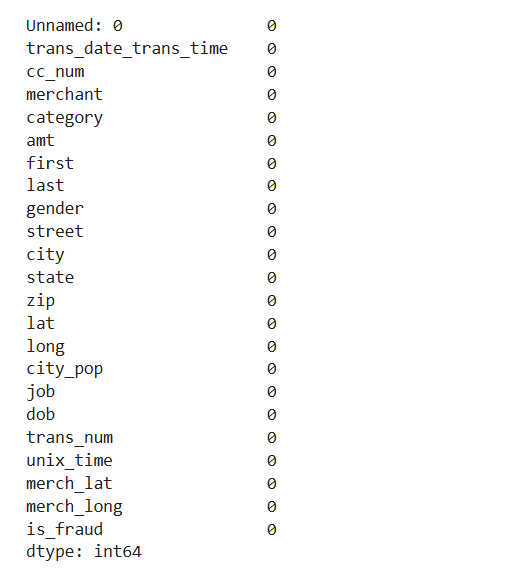


Figure 1.

From Figure 1. above, it shows that there are no missing values. We also have to check for duplicated rows in the data set with .duplicated().sum()

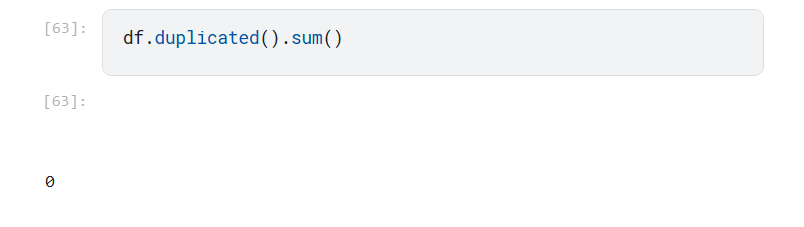


Figure 2.

From Figure 2, we can see that there are no duplicated rows. We also have to ensure that each column data type is appropriate for Exploratory Data Analysis (EDA) with the command .info().

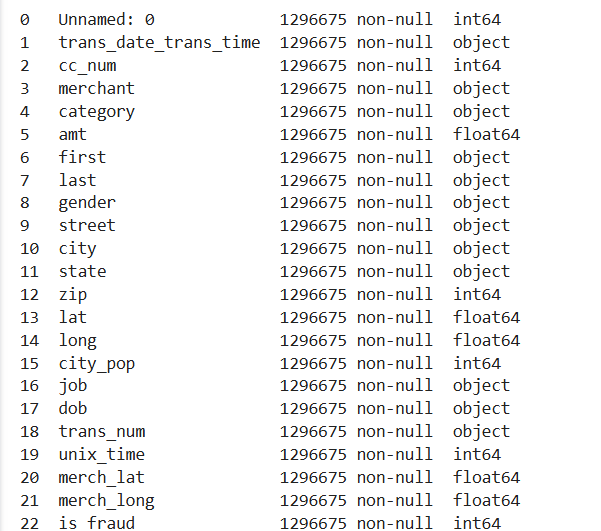


Figure 3

From Figure 3, we can see that all data type is appropriate except for trans\_date\_trans\_time and dob (date of birth), we would have to change that to DateTime Format instead of object format. After that, we would have to check that each column have the same data type with .nunique

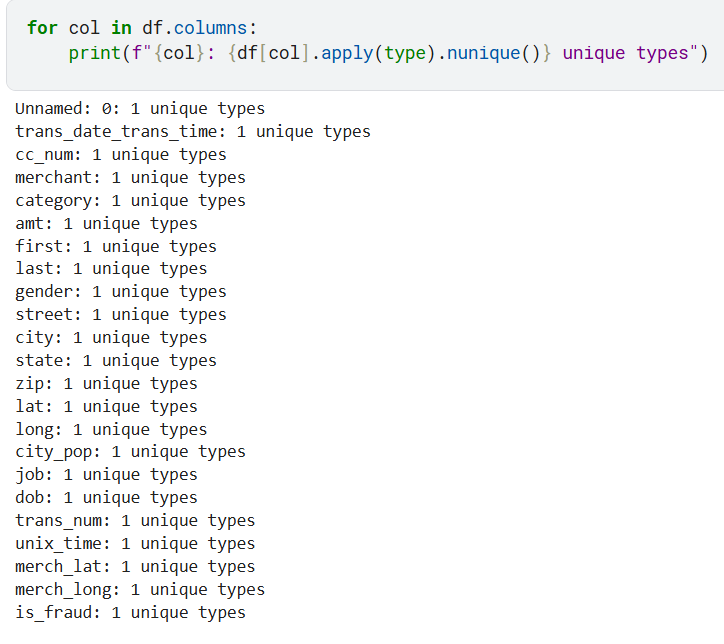


Figure 4

Figure 4 shows that each column have only 1 type of data which is the same data type. Now we will have to check for columns with constant values, as they may be our target variable and gather more information about the dataset.

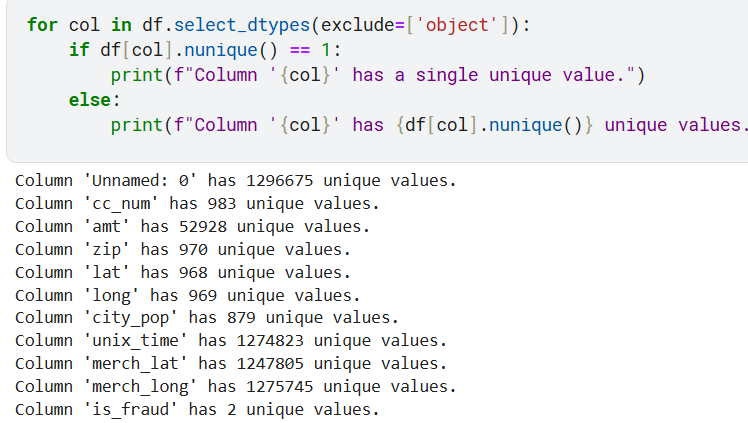


Figure 5

From Figure 5, we can see that “is\_fraud” has 2 unique value, 1 or 0. Figure 5 only shows column of numerical values, now we will search for all columns with all type of data types that include objects.

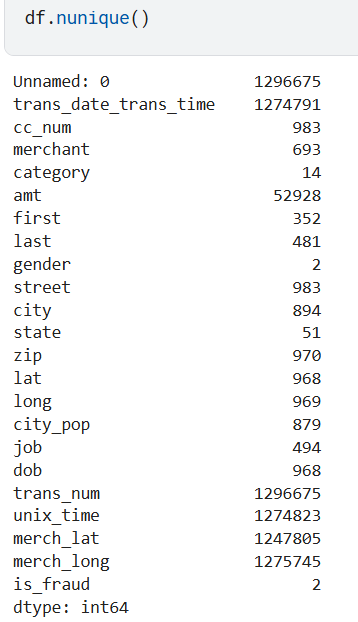
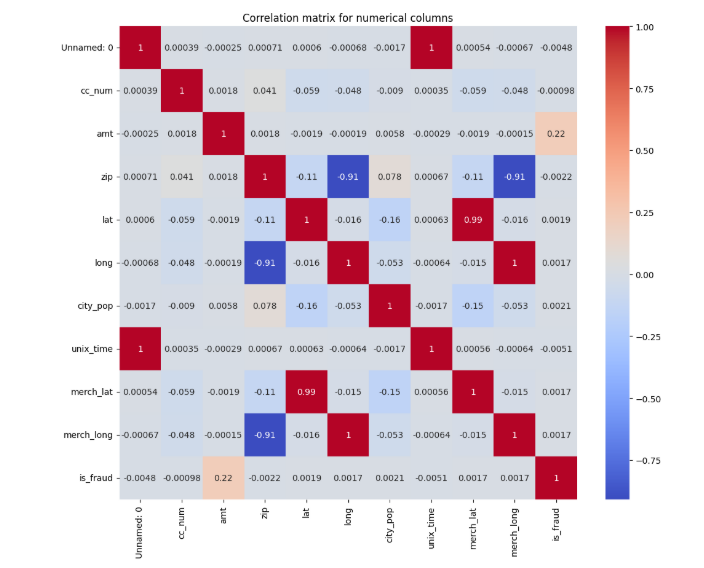


Figure 6

Figure 6 shows more information about the different columns. Transactions only happen in 51 states, 2 genders and more. Now we will check the correlation of the values.

Figure 7

## 5.2 Exploratory Data Analysis

Now we will begin on our EDA. Exploratory data analysis (EDA) is done to gain a fundamental understanding of a dataset before making any assumptions, allowing us to identify patterns, outliers, relationships between variables, and potential issues within the data, which helps guide further analysis and model building by revealing insights that might otherwise be missed; essentially, it's a crucial first step to ensure our analysis is based on a solid understanding of the data itself. From EDA, we are able to uncover patterns and relationships of trend and correlation between variables and help us to generate hypotheses. We could also validate our assumptions like high transaction amounts can lead to be a potential fraud activity

Firstly, we create a copy of dataset for eda showcasing to avoid tampering with the original dataset. (Refer to Appendix A)

Next, we convert the trans\_date\_trans\_time and dob from object datatype to datetime datatype. (Refer to Appendix B + C)

We would also create a more detailed age group column, and categorise the ages to different categories (Refer to Appendix D)

We would also calculate distance between cardholder and merchant as the distance between those 2 might give us a pattern. For example, longer distance may suggest fraudulent activity. (Refer to Appendix E)

We would also create bins for city population as continuous data is grouped into discrete intervals (bins) to simplify complex data, reduce noise, identify patterns, and make data visualization easier to interpret, especially when dealing with large datasets or distributions with outliers. (Refer to Appendix F)

After creating our columns, we would drop unwanted columns (Refer to Appendix G)

After preparing the data for EDA, we would start off by finding the distribution of the target variable of “is\_fraud”. This is to show if the data is balanced or imbalanced, and from the figure shown in Appendix H, we can see that most transactions are not fraudulent, which can suggest that the dataset is imbalanced. (Refer to Appendix H)

So now we know that there are way more non-fraudulent transactions, let's find out the distribution of transaction amounts, whether most of it is low, high or is it balanced? From Appendix I, we can see that there are a large number of low transactions and few large transactions, which could indicate that large transactions are highly likely to be fraudulent transactions. (Refer to Appendix I)

Now we will find out the average amount transaction that are non-fraudulent and fraudulent to see if fraudulent transactions tend to be on a higher amount. From the figure in Appendix J, we can tell that fraudulent transaction shows a higher average amount. (Refer to Appendix J)

We learn the amount in a transaction can possibly indicate whether it is fraud or not, we will now check whether the days of the week can give us information and clues whether fraud activity happens. Does fraudulent transaction happen on weekends more, or weekdays more? From the figure in Appendix K, we can tell those fraudulent transactions occur more on weekends and Mondays compared to the rest of the days, which can tell us to look out for transactions more on those days. (Refer to Appendix K)

Other than days and amount, age of cardholder may also contribute to the factor of whether transaction is fraudulent or not. So we will do a correlation and from the figure in Appendix L, we can tell that high number of fraud transactions happens within people of age 30 to age 40. (Refer to Appendix L)

Time can also be a factor of whether a transaction is fraud or not, hence, we would check number of fraud transactions that happens during different parts of the day. As we can see in the figure in Appendix M, fraudulent transactions happens very often in night time compared to the rest of the day. (Refer to Appendix M)

We now know fraud occur more often at night, so which hour of the night does it usually happen on? We can see from the figure in Appendix N, fraudulent transactions are happening more on 22nd hour and also happen quite a few from midnight to 4th hour. (Refer to Appendix N)

Distance may also be an additional factor of whether the transaction is fraud or not. Hence, we would also check and as seen from the figure in Appendix O, fraudulent transactions typically happen when the distance between cardholders and merchant are between 64km to 80km distance. (Refer to Appendix O)

We will also check if city population affects rate of fraudulent activity. From the figure seen in Appendix P, we can tell that most fraudulent activity happens in city with lower population. (Refer to Appendix P)

## 5.3 Feature Engineering

We would have to perform and carry out feature engineering to potentially:

1. Improve a model's predictive performance
2. reduce computational or data needs
3. Improve interpretability of the results

We have grouped the columns data into 4 different categories:

* Numerical Features
* Discrete Numerical Features
* Continuous Numerical Features
* Categorical Features

(Refer to Appendix Q)

Some feature engineering steps that we did are:

* Extracting date time features from trans\_date\_trans\_time, extracting the hour, day of week, month, and whether it is weekend or not (Refer to appendix R).
* Creating categorical features (part\_of\_day) based on transaction hour (Refer to Appendix S)
* Creating City population bins. (Refer to Appendix F)

So, with these features engineered, we can come up with figures in Appendix K, M and N

## 5.4 EDA of continuous numerical features

Now we can plot graphs of continuous numerical features and find out more of the data whether is it skewed or any outliers present

Refer to Appendix T for boxplots.

Refer to Appendix U for histograms

From the information above, the amt column and city\_pop are highly skewed.

## 5.5 Logistic Regression

After cleaning up and removing irrelevant data from the dataset, we can move on to creating a model that can detect if a transaction is fraudulent.

As the results of each prediction is either true or false (fraud or not fraud), we will use Logistic Regression to create our model.

### 5.5.1 Encoding of non-numeric data & further cleaning

Before we can create the model, we will have to encode all values in the new dataset to be numeric as Logistic Regression only accepts numerical values. To do so, we used One-Hot Encoding. One Hot Encoding involves taking the values from a single column (e.g. day\_of\_WEEK) and creating multiple columns for each value, each containing boolean values only, which the model can also detect as numeric values (0 or 1).

The dataset had its non-numeric columns encoded and unnecessary columns (“city\_pop” and “distance”) dropped. After this, we can finally move on to creating the predictive model.

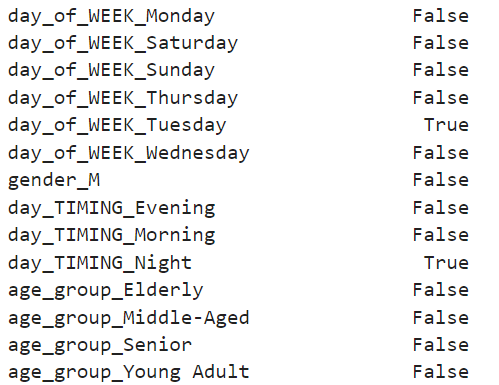
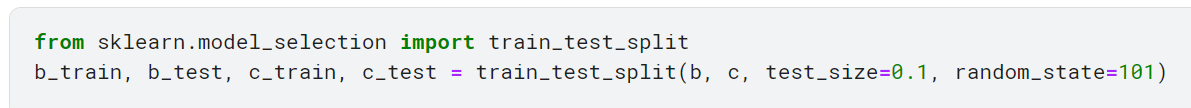


Fig 8: Example of columns created from previous column values

### 5.5.2 Training of the Model

First, we create testing and training data splits, the data that will be used to train the model’s predictive ability and test it.

Fig 9: Code for creating testing and training data splits

Next, we normalize the data to ensure more consistent and efficient learning for the model, before creating a logistic model and training it.

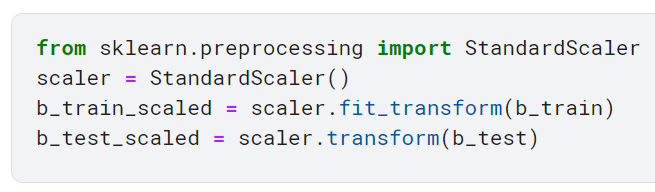


Fig 10: Code for scaling the data

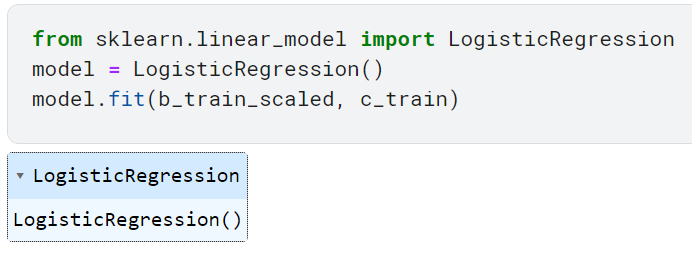


Fig 11: Code for creating the Logistic Regression model

After creating and training the model, we will run a test for the model to make predictions on the testing data set. We will see from the figure below that the model predicts a total of 162 fraud out of 129668 cases in the test.

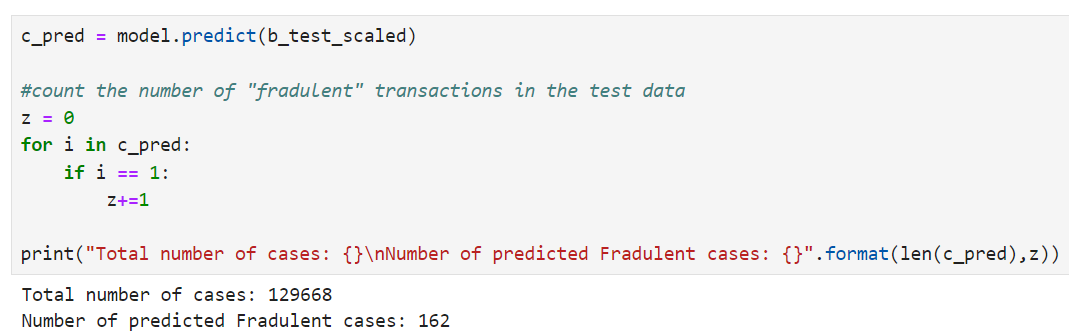


Fig 12: Code for model prediction and calculating number of predicted frauds

## 5.6 Model Evaluation

After successfully training and testing the model, we will move on to evaluating the overall accuracy of the model’s predictions. The most common way to do so is using a Confusion Matrix, which shows the number of cases the model correctly or incorrectly predicted.

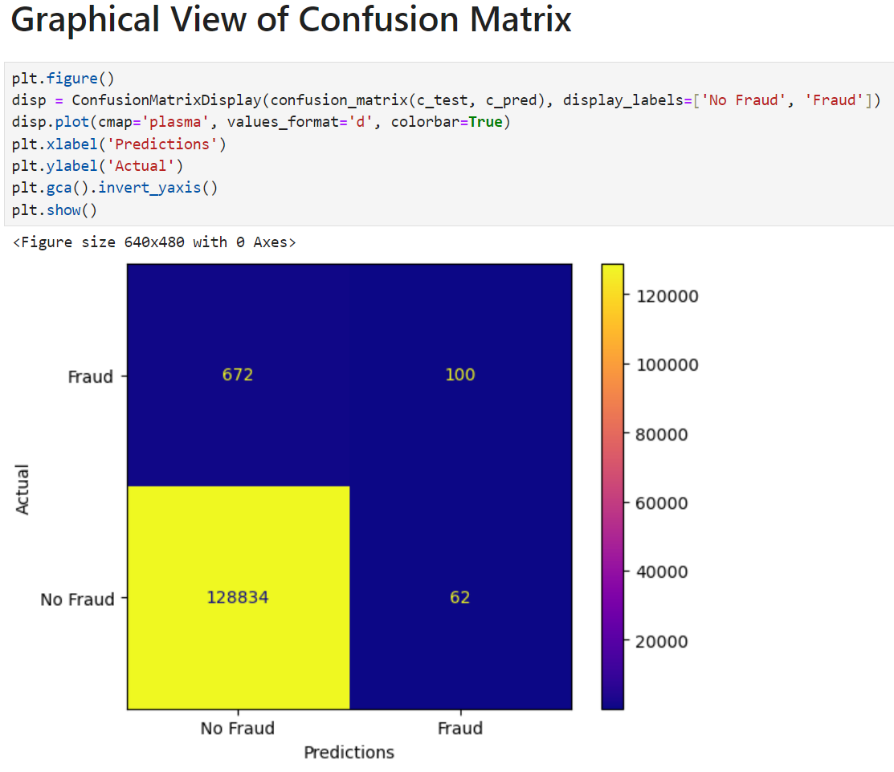


Fig 13: Confusion Matrix showing the accuracy of the model

From the above figure, we can see that out of the 162 predicted frauds only 100 of them are actual frauds. Additionally, there are 672 other frauds that were not even detected by the model. A better way of viewing these statistics is through the classification report below.

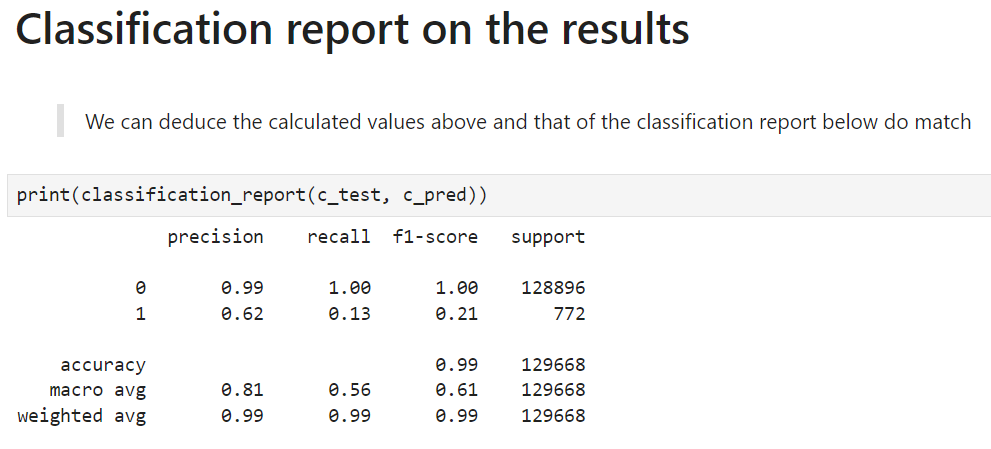


Fig 14: Classification report of the first test

From the classification report above we can see that the precision of the model is not bad, having 62% of the predicted values be accurate. However, the recall of the model is extremely low at 13%. This means that only 13% of the frauds that exist in the test data were detected. The f1-score, which displays the overall balance between precision and recall, is also low (21%), indicating that the model is not extremely reliable yet. As a result, we will have to optimize the model to be better at recognising fraud while maintaining a balanced f1-score.

First, we create the model again, this time setting a class\_weight parameter as “balanced”. Doing this makes the model prioritize identifying lower frequency class (fraud in this case as only a few cases in the test data are fraud) over higher frequency class (not fraud). The results below show a significant increase in recall from 13% to 87%. However, the precision of the model has also significantly decreased, causing a massive imbalance resulting in the f1-score being extremely low. Since we want not just a high recall, but a good overall f1 scored. We will have to change the class\_weight to change the prioritisation of the model.

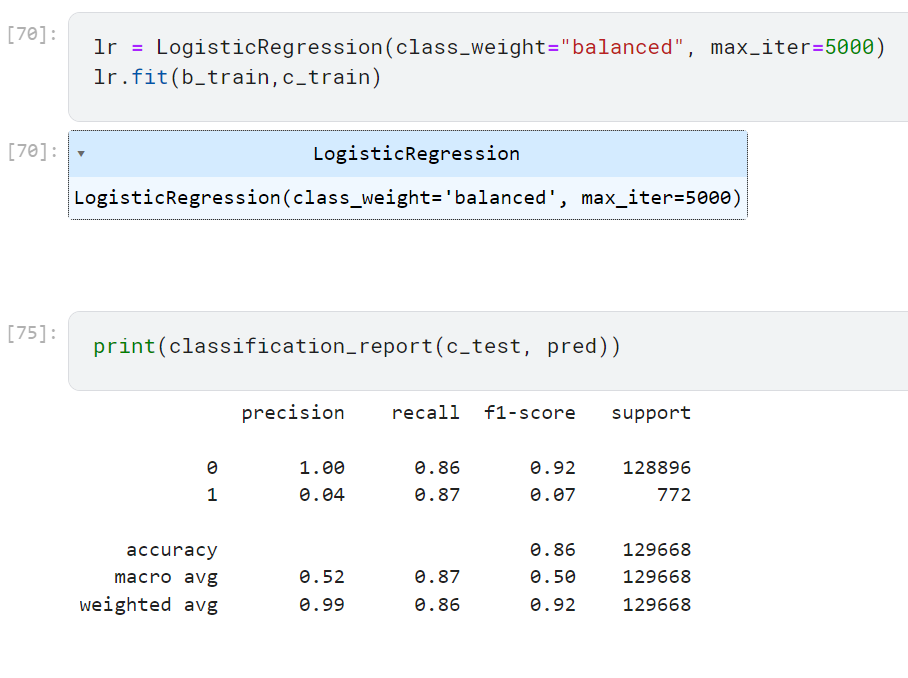


Fig 15: Optimising the model by implementing class\_weight

Below are three figures, showing three of the many values that were changed in class\_weight and their corresponding classification report. Basically, the value {0:1,1:10} in the class\_weight specifies how much weight we want to put on a specific class (0 for not fraud and 1 for fraud). Since the model originally did not detect fraud cases, we focus on increasing the weight of the “1” class until a desired result is obtained.

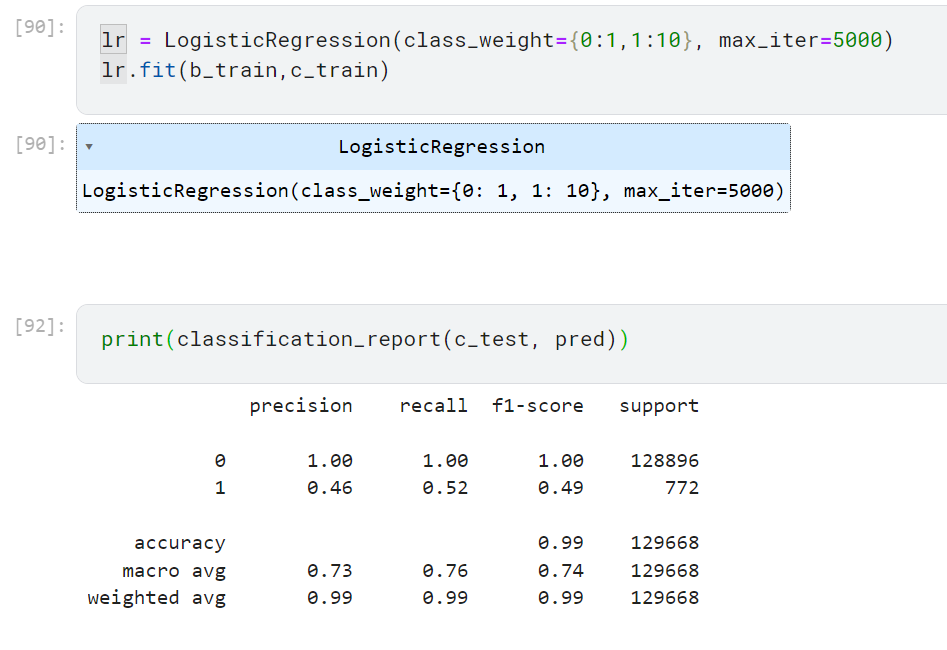


Fig 16a: Optimising by changing class\_weight of “1” to 10

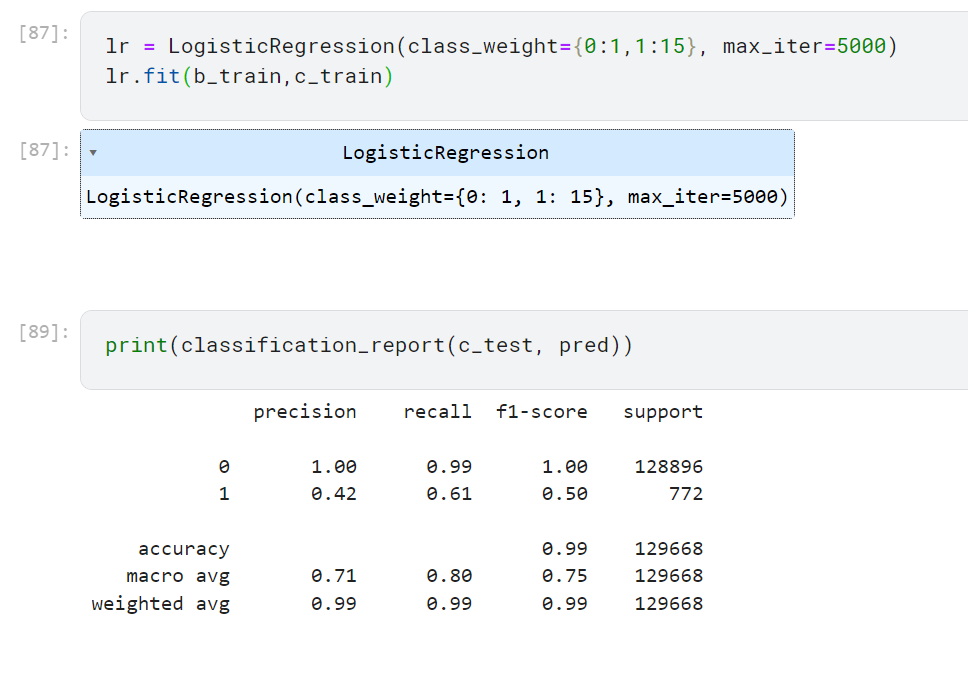


Fig 16b: Optimising by changing class\_weight of “1” to 15

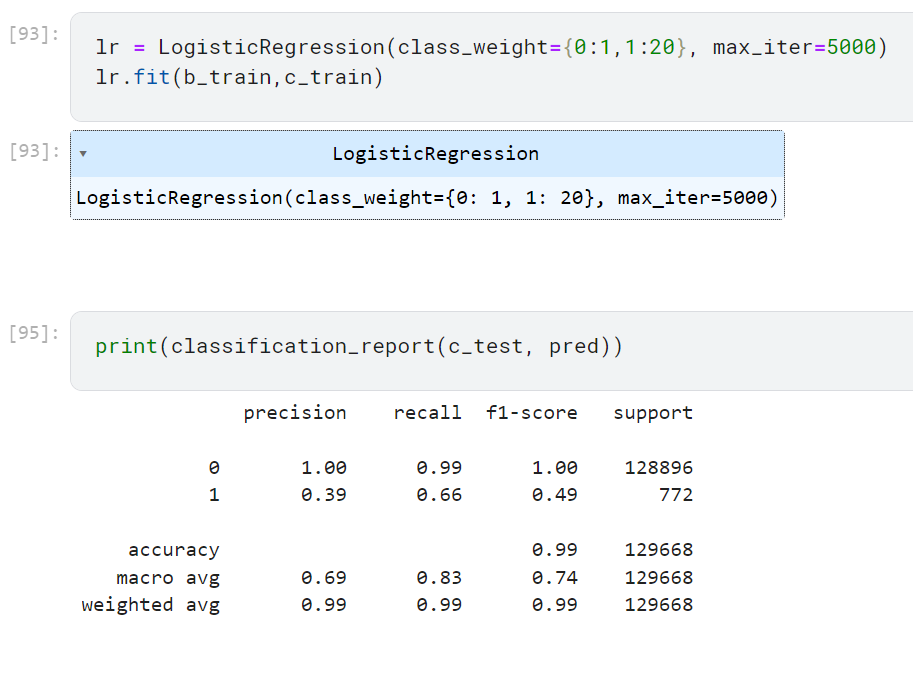


Fig 16c: Optimising by changing class\_weight of “1” to 20

From the above figures, we can see that the highest the f1 score reaches is when the class\_weight of “1” is 15. Other values of “1” gave lower f1 scores, hence the most balanced this model’s accuracy can be is 0.50, with a detection frequency for frauds (recall) at 66% and percentage of correctly detected predictions (precision) being 39%.

# 6. Findings

From the various tests as conducted above, we have discovered various results from the model:

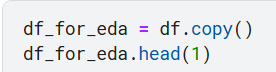
* Model Evaluation:
  + Accuracy: While the accuracy of the model from the classification report was always shown to be high at 99%, this is misleading because of the class imbalance, especially in predicting fraud
  + Precision: Precision of the model originally was high at 62%, however through the various tests conducted it is seen that the precision decreases as the recall increases, eventually settling at 42% for a desired f1 score. (refer to figure 16)
  + Recall: The original recall of the model was extremely low at only 13%, but through the tests conducted we were able to increase the recall to 61% despite the precision decreasing as a result.
  + F1 Score: While obtaining a high recall is desirable, it is more important to focus on balancing out the precision with the recall. Especially in this case where the best recall value is 61%, despite being lower than our desired recall (90%), this was the best we could get while maintaining a good f1 score of 50%
* Confusion matrix:
  + There was a total of 128896 non-fraud and 772 fraud cases.
  + The model correctly detected 468 frauds, falsely detected 640 non-frauds, and missed 304 actual fraud cases.
* Model’s strengths:
  + The model was able to correctly detect a majority, almost 100% of the non-fraud cases due to the vast number of non-fraud cases in the dataset.
* Adjusting of the Model:
  + Through experimentation of the class\_weight values, we were able to obtain an acceptable value for recall (61%) with the highest possible f1 score discovered (0.5) for this model. While this decreased the precision to 42%, obtaining a balanced f1 score is more desirable than just either precision or recall.
* Overall Impact on Consumers:
  + A recall of 0.61% will correctly identify more than half of attempted fraud, which will benefit consumers as less as victims of fraudulent transactions.
  + A low precision of 42%, however, will result in multiple false alarms, which may result in complaints from consumers.

# 7. Conclusion

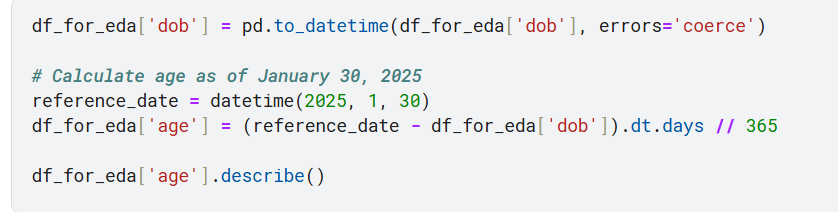
Overall, in this project, we were successfully able to create a model that can predict if a transaction is fraudulent or not using logistic regression. While the model can accurately predict non-fraudulent transactions, it is more challenging for it to predict actual fraud, resulting in a stark contrast between the precision and recall for non-fraud and fraud (e.g. Non-fraud Precision: 100%, Fraud Precision: 42%). While we were able to obtain a good balance between the precision and recall for fraudulent prediction (f1 score: 50%), it came at the cost of lowering both the precision and recall of the model as they were seemingly inversely related to one another (e.g. precision would decrease when recall increases). This may result in some fraud bypassing the detection and several false alarms.

As we were unable to achieve the desired recall of 90% without causing major imbalance with precision, further improvements, research and possibly a different approach will have to be made in the future to create a model that can be relied on by all online consumers and keep the online market a safe a trusted place.

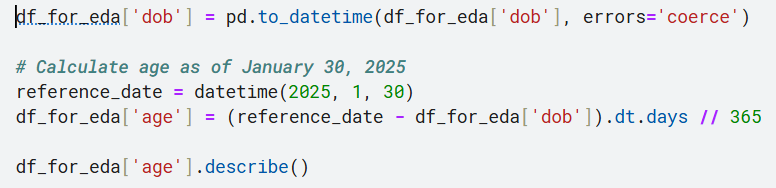
# 8. Appendix



Appendix A



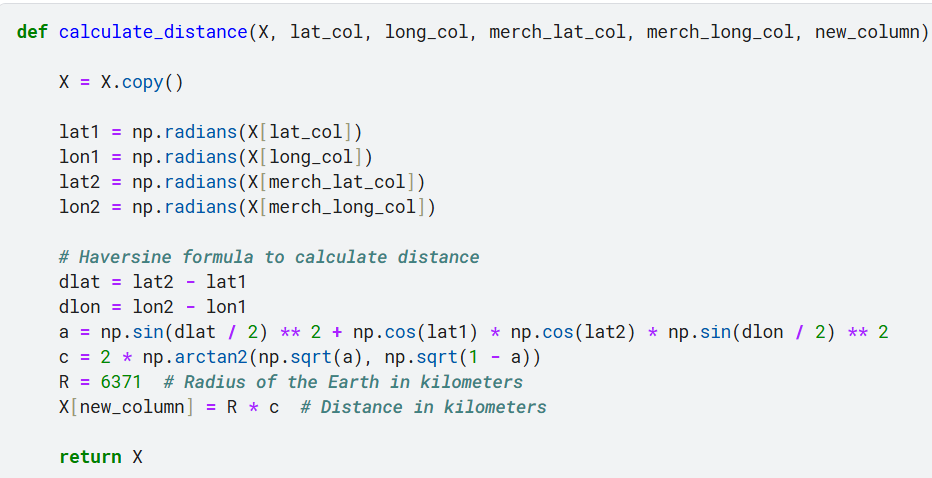
Appendix B

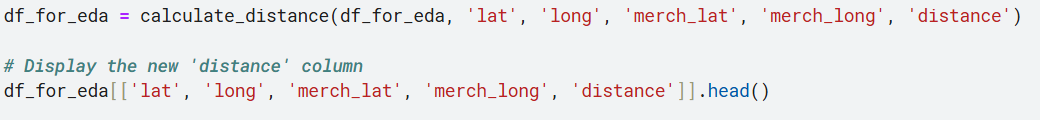


Appendix C

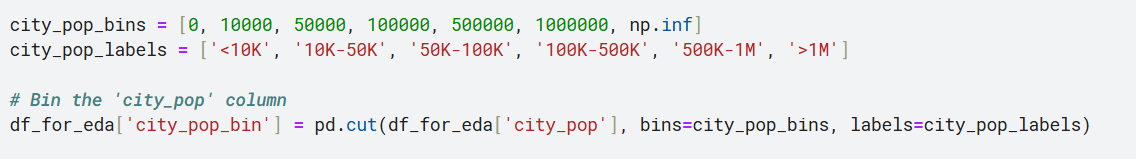


Appendix D





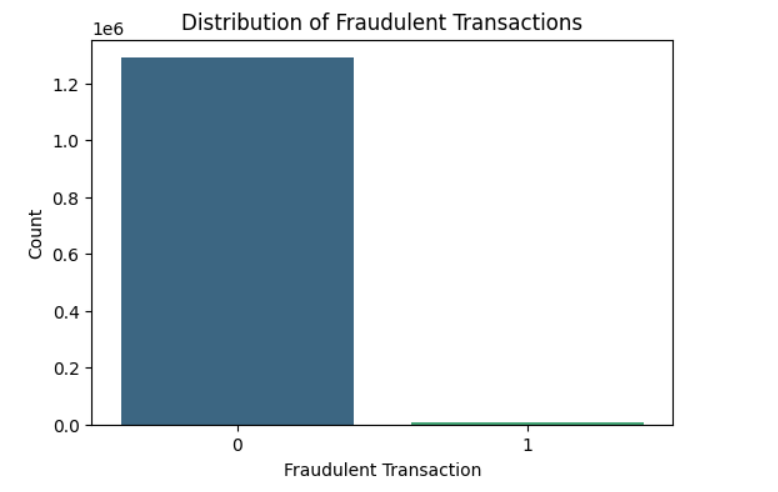
Appendix E



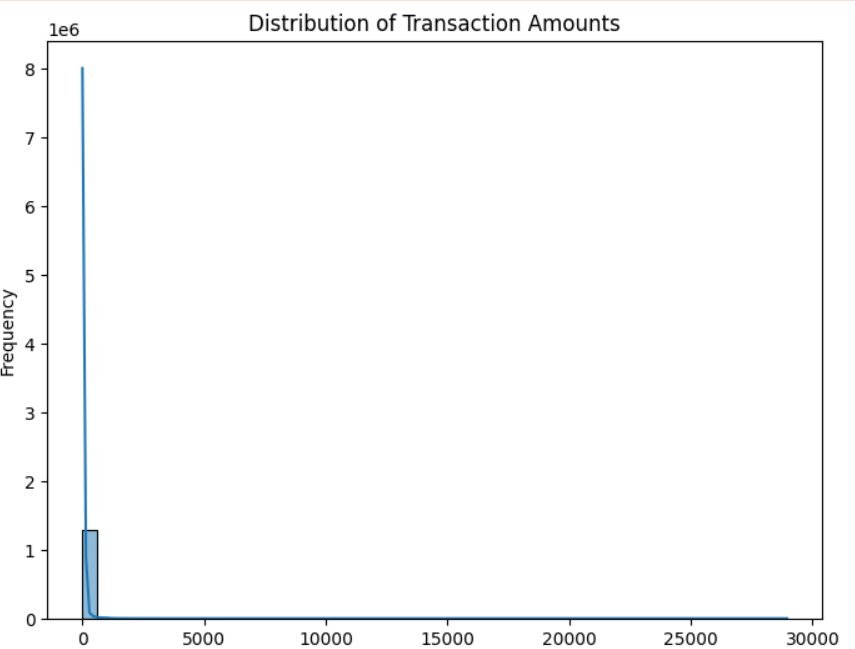
Appendix F



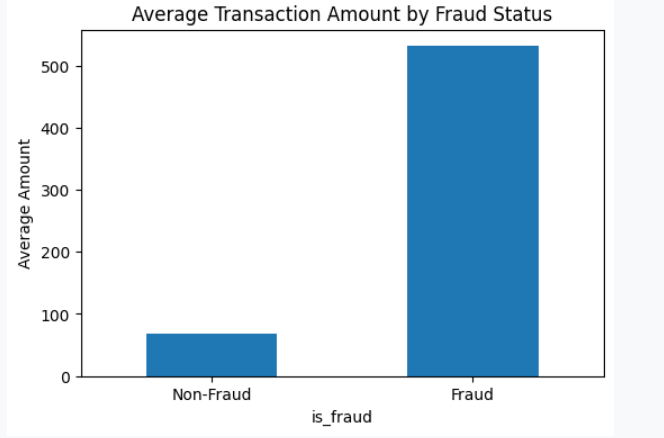
Appendix G

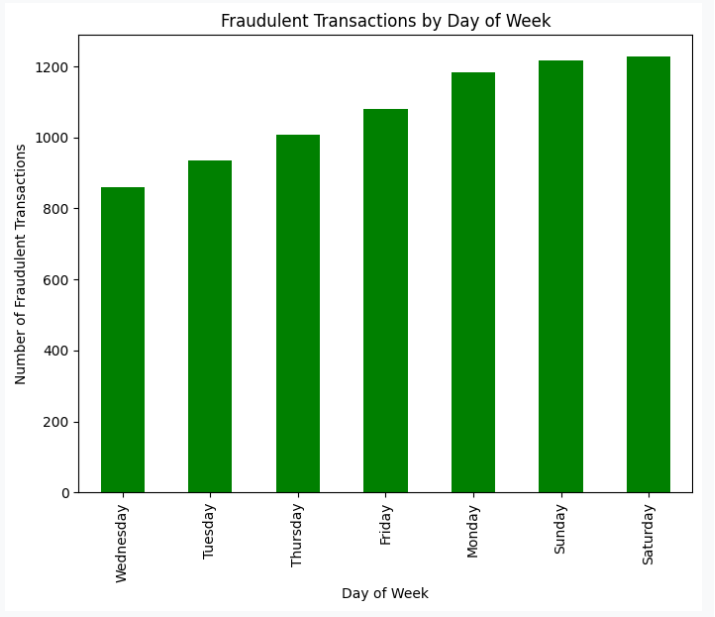


Appendix H

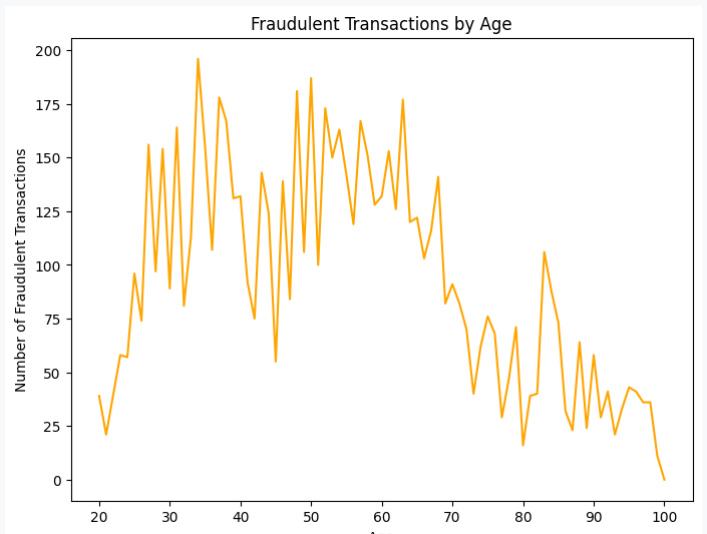


Appendix I

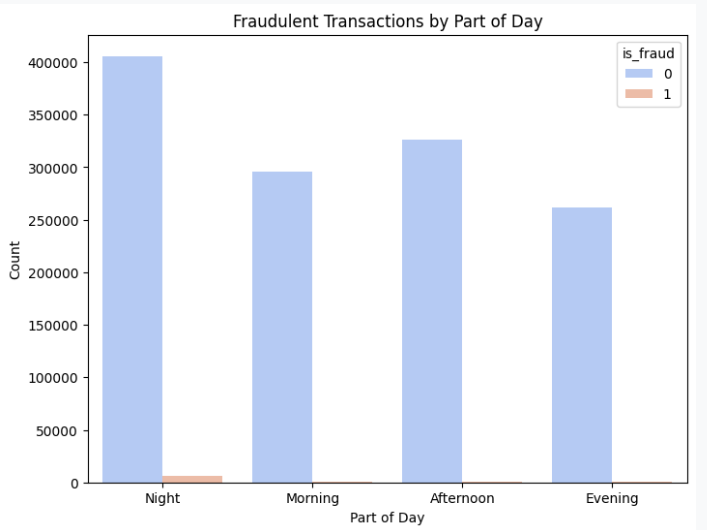
  
Appendix J



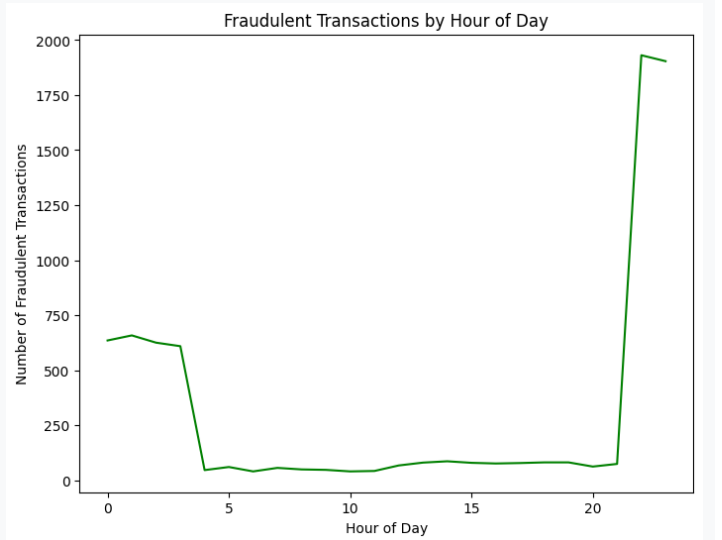
Appendix K



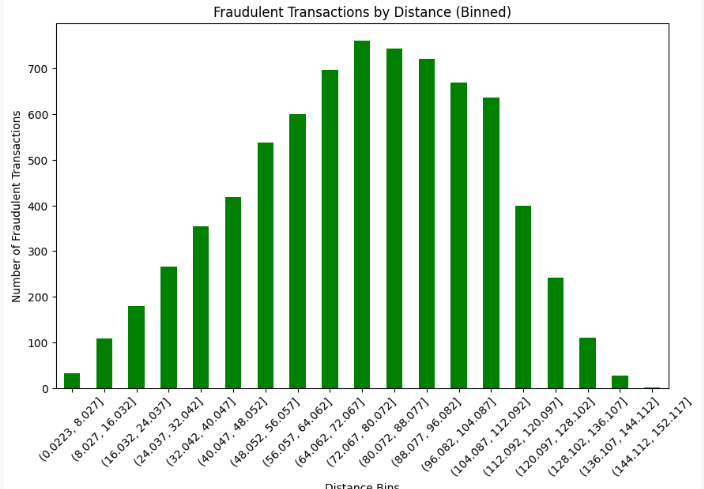
Appendix L



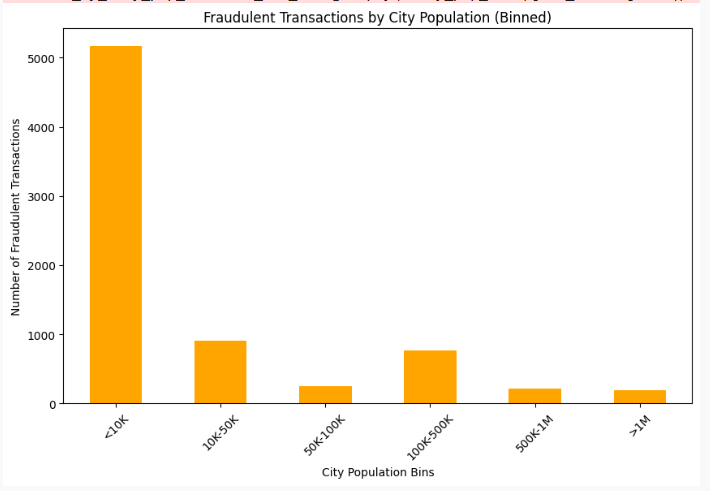
Appendix M



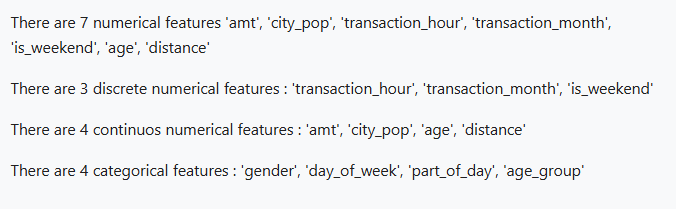
Appendix N



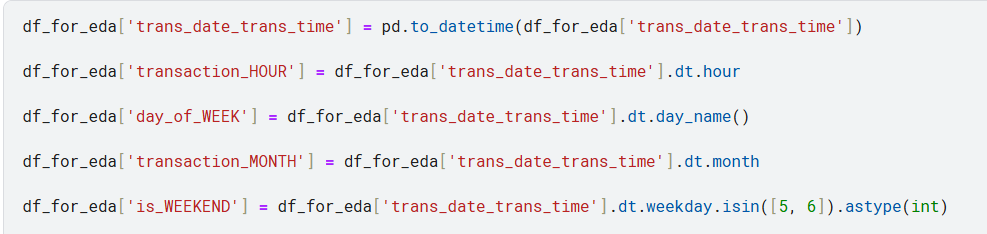
Appendix O

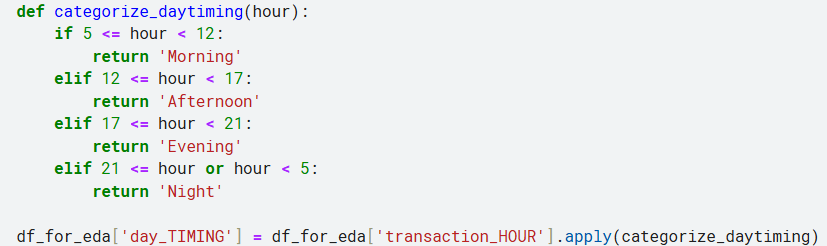


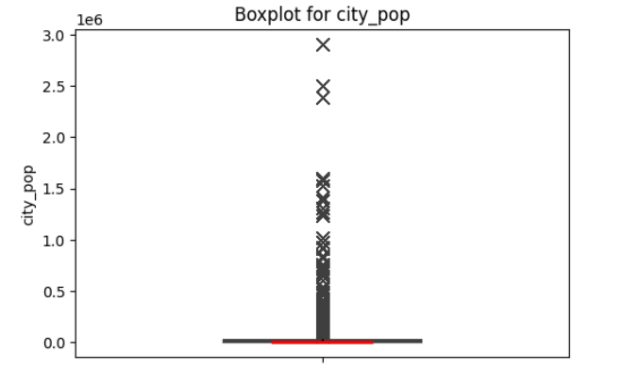
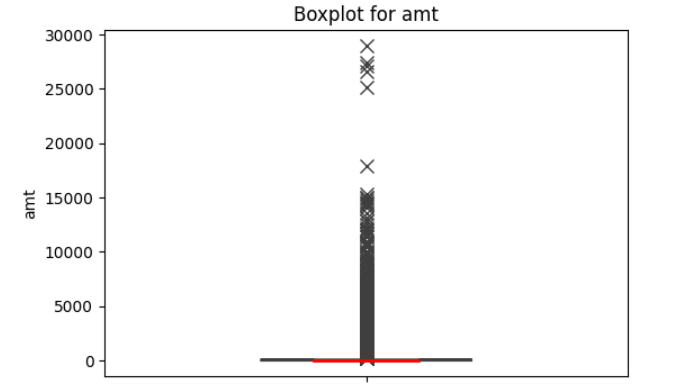
Appendix P

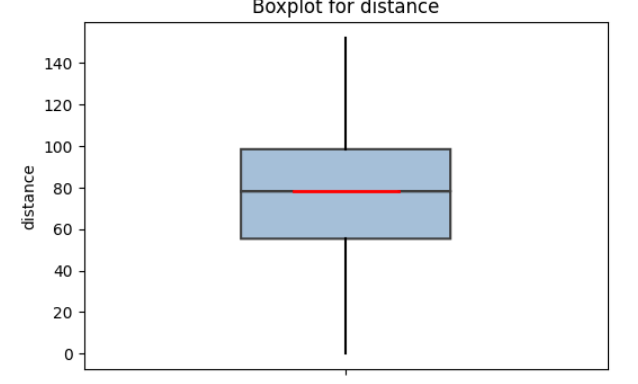
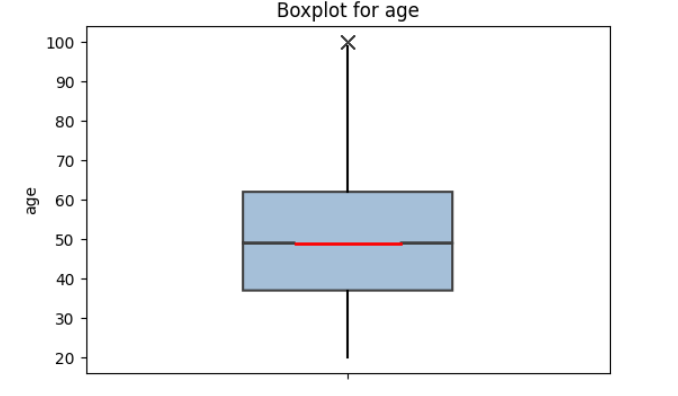


Appendix Q

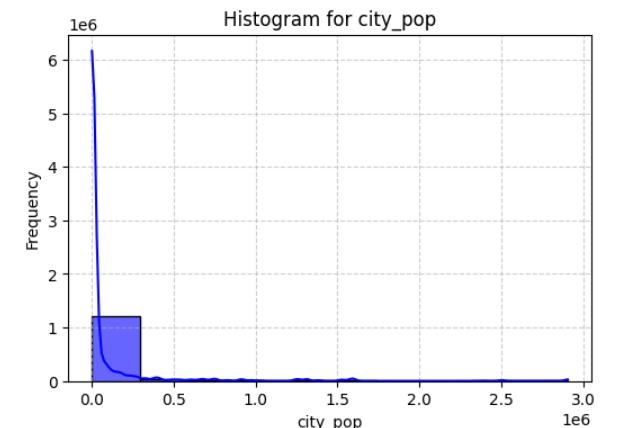
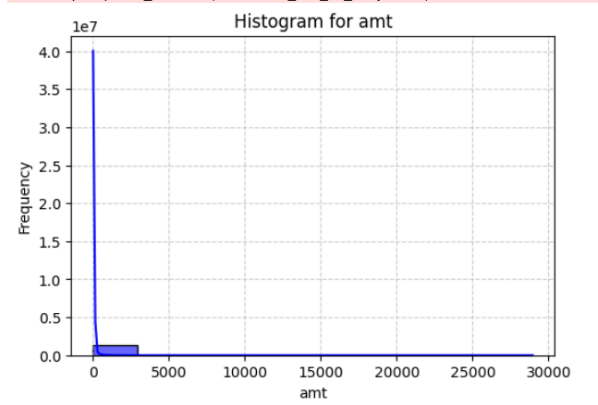
Appendix R

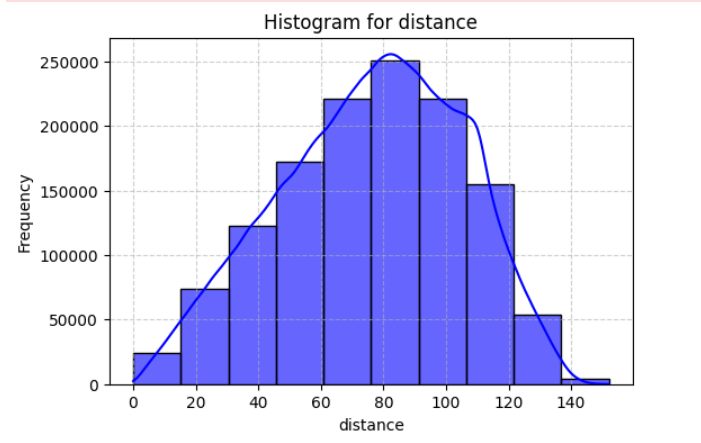
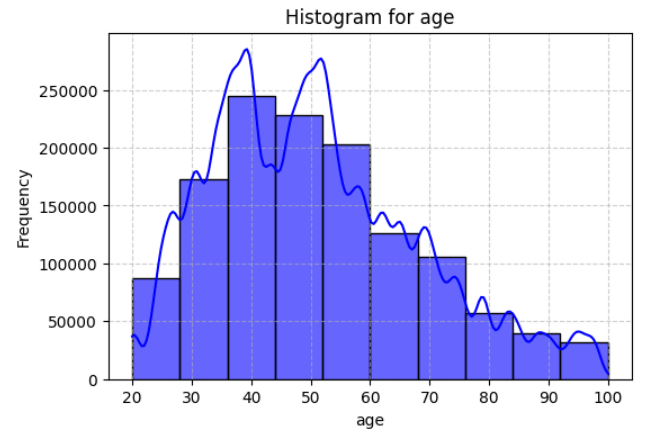
Appendix S





Appendix T





Appendix U

# 9 References

Abhijeet Kumar. (2021, November 22). *Why is Data Validation Crucial for Long-term Data Success*. Sigmoid. <https://www.sigmoid.com/blogs/data-validation/>

Bloomenthal, A. (2023, May 6). *How Credit Cards Work*. Investopedia. <https://www.investopedia.com/terms/c/creditcard.asp>

IBM. (2021, October 4). *Exploratory Data Analysis*. Ibm.com. <https://www.ibm.com/think/topics/exploratory-data-analysis>

samanthawolfe. (2023, October 27). *How to Prevent Debit Card and Credit Card Fraud - Midwest Bank*. Midwest Bank. <https://midwest.bank/news/2023/10/how-to-prevent-debit-card-credit-card-fraud/>

Square. (2022). *What Is Credit Card Processing and How It Works*. Square. <https://squareup.com/ca/en/townsquare/credit-card-processing>

they /them, M. S. (2021, February 19). *What is Feature Engineering? - Towards Data Science*. Medium; Towards Data Science. <https://medium.com/towards-data-science/what-is-feature-engineering-bfd25b2b26b2>