

**DATA MINING**

*Analysis of transactional data in order to predict fraudulent transactions.*

*Adithya Vinod*

*Sneha Kumble Manohara*

**School of Graduate Professional Studies**

MPS/MS in Data Analytics

August, 2022

# Document Control

## Work carried out by:

|  |  |  |
| --- | --- | --- |
| **Name** | **Email Address** | **Other** |
| Adithya Vinod | avv5502@psu.edu |  |
| Sneha Kumble Manohara | skm6622@psu.edu |  |
|  |  |  |

Revision Sheet

|  |  |  |
| --- | --- | --- |
| **Release No.** | **Date** | **Revision Description** |
| 1. | 09/04/2022 | Defined the research goals, the queries to address, and discussed the source of the data collected to address the research queries. |
| 2. | 09/04/2022 | Created data dictionary for dataset |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Note**: For the weekly submission deadline of the project assignment refer to the syllabus in Canvas. Each week’s assignment is for 10 points and the project assignment is worth 35% of your final grade. Tasks for each week has to be completed and documented within this document

**Table of Contents**

[**Document Control** 2](#_Toc89640382)

[Work carried out by: 2](#_Toc89640383)

[Revision Sheet 2](#_Toc89640384)

[**Project Assignment 1** 4](#_Toc89640385)

[Research Goals and Business Understanding 4](#_Toc89640386)

[**Project Assignment 2** 5](#_Toc89640387)

[Data Dictionary for Fraudulent Transaction Data 5](#_Toc89640388)

[**Project Assignment 3** 9](#_Toc89640387)

[**Project Assignment 4** 11](#_Toc89640387)

[**Project Assignment 5** 12](#_Toc89640387)

[**Project Assignment 6** 14](#_Toc89640387)

**Project Assignment 1**

**Research Goal:**

Exploring and understanding transactional data in order to predict fraudulent transactions using Frequent Pattern Mining using Apriori Growth (Reference 1). Predicting fraudulent transactions helps improve the reputation of the merchant as well as the service provider, which in turn benefits the financial sector.

**Business questions:**

a. What kind of products are bought during fraudulent transactions?

b. Is there a range of amounts that are targeted by fraudsters?

c. Do fraudulent transactions have any relation with the locality?

d. Is there any relation with age of the card holder and fraudulent transaction?

**Dataset:**

The dataset for the above problem statement can be found on Kaggle. The link to the dataset is: <https://www.kaggle.com/datasets/kartik2112/fraud-detection>

**Project Assignment 2**

The dataset used in this project contains legitimate and fraud transaction from the duration 1st Jan 2019 – 31st Dec 2020. It covers credit cards of 1000 customers doing transactions with a pool of 800 merchants.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Name** | **Data Type** | **Data Classification** | **Description** |
| trans\_date\_trans\_time | DateTime | Time series | Transaction date and time |
| cc\_num | Numeric | Nominal | Credit Card Number |
| Merchant | Categorical | Nominal | Merchant |
| Category | Categorical | Nominal | Transaction category |
| amt | Numeric | Interval | Transaction amount |
| First | Categorical | Nominal | First Name of the card holder |
| Last | Categorical | Nominal | Last Name of the card holder |
| Gender | Categorical | Nominal | Gender of the card holder |
| Street | Categorical | Nominal | Card Holder’s address: Street |
| City | Categorical | Nominal | Card Holder’s address: City |
| State | Categorical | Nominal | Card Holder’s address: State |
| Zip | Numeric | Nominal | Card Holder’s address: Zip Code |
| Lat | Categorical | Interval | Card Holder’s address: Latitude |
| Long | Categorical | Interval | Card Holder’s address: Longitude |
| City\_pop | Numeric | Interval | City population |
| Job | Categorical | Nominal | Job of the card holder |
| Dob | DateTime | Time series | Date of birth |
| trans\_num | Numeric | Nominal | Transaction Number |
| unix\_time | DateTime | Time series | Transaction unix time |
| merch\_lat | Categorical | Interval | Merchant’s location latitude |
| merch\_long | Categorical | Interval | Merchant’s location longitude |
| is\_fraud | Binomial | Nominal | Is the transaction Fraudulent |

**Project Assignment 3**

As a part of data cleaning, the first thing we noticed was that there were very few rows with NA and came up to just 0.001% of the entire dataset, which we decided to drop it. We also noticed that the quality of the data was not satisfactory. The data included a lot of outliers, skewness and the interpretability were low.

FEATURE REMOVAL:

There were variables in the dataset that wouldn’t be very useful in classifying whether the transaction was a fraud or not. These variables included:

1. cc\_num – credit card number
2. first – first name
3. last – last name
4. street – street name
5. trans\_num – transaction ID
6. unix\_time – unix time

FEATURE CREATION:

There were some columns that needed pre-processing such as, adding a new column for age from the DOB column, a new column of distance was added based on the information in column merch\_long and merch\_lat in order to check if fraudulent transactions have any relation with the locality, few of the numerical variables were converted to categorical variables based on taking 33rd and 66th percentile in order to increase the interpretability.

OUTLIERS:

Before outlier removal: After outlier removal:

Chart, box and whisker chart

Description automatically generated Chart

Description automatically generated

The Age column had a few outliers which were detected using the box and whisker plot. Since the number of outliers were very minimal these records were removed.

Since the problem we are trying to solve is an imbalanced classification problem, not all outliers can be removed as it could be an important feature in classification.

For example, the column that represents the amount of the transaction contains a lot of outliers. But cleaning these outliers would possibly result in misbalancing the already imbalanced dataset. Because of which this column was kept unchanged.

Chart

Description automatically generated

SKEWNESS:

We noticed that couple of the numeric variables we were interested were skewed heavily. To reduce the skewness, we implemented box-cox transformation and transformed the data.

Skewness before transformation: Skewness after transformation:

Table

Description automatically generated with medium confidence Text

Description automatically generated

The variable “*is\_fraud*” is the output variable and hence wasn’t considered for the transformation.

Skewness before transformation:

Chart

Description automatically generated

Skewness after transformation:

Chart, histogram, box and whisker chart

Description automatically generated

The transaction data before data cleaning:

Graphical user interface

Description automatically generated

The transaction data after data cleaning:

Graphical user interface, application

Description automatically generated

**Project Assignment 4**

Feature selection is a fundamental step in many machine learning pipelines. You dispose of a bunch of features, and you want to select only the relevant ones and to discard the others. The aim is simplifying the problem by removing not useful features which would introduce unnecessary noise.

Boruta is a smart algorithm dating back to 2010 designed to automatically perform feature selection on a dataset. We will be using BorutaPy, which is a version for Boruta for Python.

For the dataset in use, we consider the 9 numerical variables from which a few, if any could be removed from further analysis.

Using Boruta on the dataset for the 9 numerical variable gives us the following result:

1. Number of Confirmed variables: 7, namely:
   1. lat
   2. long
   3. city\_pop
   4. merch\_lat
   5. merch\_long
   6. age
   7. amt
2. Number of Tentative variables: 0
3. Number of Rejected variables: 2, namely:
   1. cc\_number
   2. zip

Hence, we will not be considering cc\_number (Credit Card Number) and zip (Zip code) for further analysis.

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated

From the above plot, we can see that amt is the most correlated to is\_fraud and hence it becomes the most important predictor. We can also see that cc\_num has the least correlation with respect to is\_fraud. Hence we can remove cc\_num from further analysis.

# **Project Assignment 5**

To address our research queries, we will be using certain statistical as well as data mining techniques. The main objective of this project is to see if there is any pattern of activities of the fraudulent users. One data mining technique that helps us identify the patterns in transaction would be Apriori method.

The main aim of this project is to identify what kind of products are most targeted by fraudsters. Understanding this would help merchants provide necessary security measures.

To categorize the amounts that are targeted by fraudulent, we have clustered the amount based on percentile. The amount will be categorized as Highly Expensive, Sparsely Expensive and Moderately Expensive.

The relation with fraudulent transactions and locality can be calculated by taking the distance between the card holder’s latitude-longitude position and transaction latitude-longitude position. The distance will also be categorized as Far Away Distance, Nearby Distance and Moderate Distance through clustering.

|  |  |  |
| --- | --- | --- |
| **Query** | **Data Mining Technique** | **Statistics** |
| What kind of products are bought during fraudulent transactions? | Apriori/Association Rule Mining Algorithm. |  |
| Is there a range of amounts that are targeted by fraudsters? | Apriori/Association Rule Mining Algorithm. | Categorizing the amount based on the percentile |
| Do fraudulent transactions have any relation with the locality? | Apriori/Association Rule Mining Algorithm. | Categorizing the distance based on the percentile |
| Is there any relation with age of the card holder and fraudulent transaction? | Apriori/Association Rule Mining Algorithm. | Categorizing the distance based on the percentile |

# **Project Assignment 6**

After performing Apriori Association rule mining Algorithm, we found out that most of the fraudulent transaction involve not a lot of physical theft of the card, but with credit card credentials for online purchases.

Significantly higher orders than the usual user transactions were observed. There also seems to be no relation with locality of the card holder and transaction location, which tells us that these transactions are done online.

Further, we also found that most of the fraudulent transactions were done on old-aged card holders. The involvement of this huge number of transactions makes it more vulnerable to attacks from hackers.

Hence it is necessary to get the most from these hackers, identify their pattern and match it with the new users to detect fraudulent transactions.

|  |  |  |
| --- | --- | --- |
| **Query** | **Data Mining Technique** | **Findings** |
| What kind of products are bought during fraudulent transactions? | Apriori/Association Rule Mining Algorithm | **Most of the products that were bought during fraudulent transaction were during online shopping and grocery shopping online.** |
| Is there a range of amounts that are targeted by fraudsters? | Apriori/Association Rule Mining Algorithm | **The amount spent during fraudulent transaction involved highly expensive products** |
| Do fraudulent transactions have any relation with the locality? | Apriori/Association Rule Mining Algorithm | **There seems to be no relation with the locality of the card holder and the fraudster’s location.** |
| Is there any relation with age of the card holder and fraudulent transaction? | Apriori/Association Rule Mining Algorithm | **Most of the fraudulent transactions were done on old aged card holders.** |

**What kind of products are bought during fraudulent transactions?**

**Data Mining Technique**: Boruta, Apriori Algorithm

**Evaluation**: Support, Lift, Confidence

Table

Description automatically generated

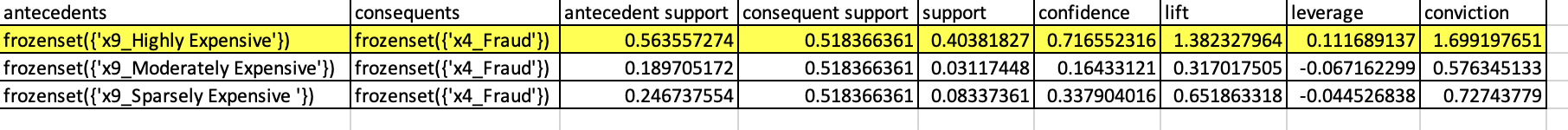
**Assumption**: The following Apriori rules were constructed, assuming a minimum Support value of 1% because fraudulent transactions occur in very low number of transactions.

**Justification**: From the above set of rules, it’s quite clear that most of the fraudulent transactions involve the purchase of groceries or products that are purchased online. This result comes at no shock as most of credit card transactions that occur in the 21st century are predominantly focused on online shopping and groceries.

**Is there a range of amounts that are targeted by fraudsters?**

**Data Mining Technique**: Boruta, Apriori Algorithm

**Evaluation**: Support, Lift, Confidence



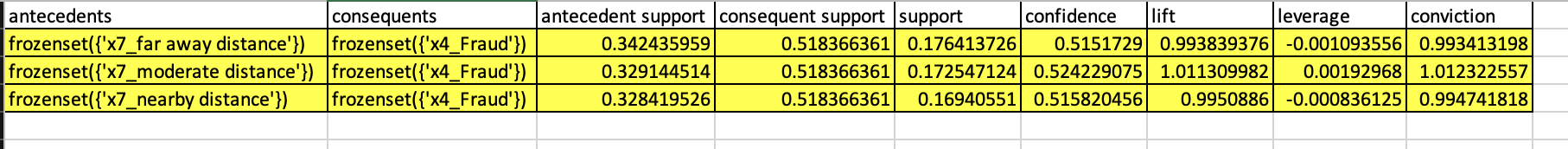
**Assumption**: The following Apriori rules were constructed, assuming a minimum Support value of 1% because fraudulent transactions occur in very low number of transactions.

**Justification**: Following set of rules are constructed considering Cost of transaction in mind. We can see that rules with the highest support values belong to those fraudulent transactions that involve highly expensive products. This is yet another finding that makes sense, as fraudulent transactions are targeted towards expensive brands and products.

**Do fraudulent transactions have any relation with the locality?**

**Data Mining Technique**: Boruta, Apriori Algorithm

**Evaluation**: Support, Lift, Confidence



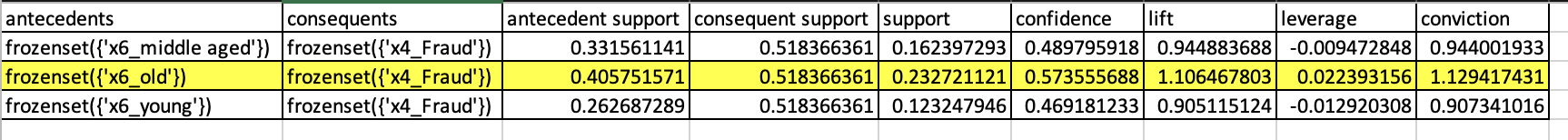
**Assumption**: The following Apriori rules were constructed, assuming a minimum Support value of 1% because fraudulent transactions occur in very low number of transactions.

**Justification**: Distance here refers to the distance between the merchant and the residence of the credit card holder. Here we can see that there is a mix of distances that make up fraudulent transaction. We can see nearby\_distance as well as far\_away\_distances with similar support , confidence and lift values, suggesting that distance between the merchant and the residence of the credit card holder is not a factor that influences fraudulent transactions. This comes as a shock because fraudsters are brave enough to try and make fraudulent transactions from near the merchant and the residence of the card holder.

**Is there any relation with age of the card holder and fraudulent transactions?**

**Data Mining Technique**: Boruta, Apriori Algorithm

**Evaluation**: Support, Lift, Confidence



**Assumption**: The following Apriori rules were constructed, assuming a minimum Support value of 1% because fraudulent transactions occur in very low number of transactions.

**Justification**: Age of the card holder was considered to verify if older customers can keep up with technology and the security features that revolve around possessing a credit card. From the above support values, we can clearly see that older customers are considered easy targets by fraudsters to conduct fraudulent transactions as they aren’t able to keep pace with technology.

REFERENCES:

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

http://ijsart.com/Home/IssueDetail/33241

APENDIX:

<https://colab.research.google.com/drive/1x-6rJypAvXDPg0gSQWse7BroS-aCiflQ#scrollTo=aaMvYvkPmyni>

Generating new Variables:

def calculate\_age(dob):

year\_now = datetime.datetime.now().year

dob\_year = int(dob[:4])

age = year\_now - dob\_year

age\_list.append(age)

distance = []

for i in range(len(df\_new)):

lat1 = df\_new['lat'][i]

lon1 = df\_new['long'][i]

lat2 = df\_new['merch\_lat'][i]

lon2 = df\_new['merch\_long'][i]

coords\_1 = (lat1,lon1)

coords\_2 = (lat2,lon2)

dist = geopy.distance.geodesic(coords\_1, coords\_2).miles

distance.append(dist)

Finding Correlation:

df\_numeric = df\_new[['cc\_num','amt','zip','lat','long','city\_pop','merch\_lat','merch\_long','Age','is\_fraud']]

df\_numeric.corr()

plt.figure(figsize=(16, 6))

heatmap = sns.heatmap(df\_numeric.corr(),vmin=-1, vmax=1, annot=True)

heatmap.set\_title('Correlation Heatmap', fontdict={'fontsize':12}, pad=14);

Dealing with Skewness:

amt=df\_new['amt']

sns.distplot(amt, hist=False, kde=True)

transformed\_amt\_data, best\_lambda = boxcox(amt)

sns.distplot(transformed\_amt\_data, hist=False, kde=True)

print('lambda:', best\_lambda)

amt[0:5]

df\_new['amt']=transformed\_amt\_data

transformed\_amt\_data[0:5]

city\_pop=df\_new['city\_pop']

sns.distplot(city\_pop, hist=False, kde=True)

transformed\_pop\_data, best\_lambda = boxcox(city\_pop)

sns.distplot(transformed\_pop\_data, hist=False, kde=True)

print('lambda:', best\_lambda)

amt[0:5]

df\_new['city\_pop']=transformed\_pop\_data

transformed\_pop\_data[0:5]

df\_new.skew()

Data Transformation:

q66,q33 = np.percentile(df\_new['distance'],[66,33])

distance\_category = []

for distance in df\_new['distance']:

if distance >= q66:

distance\_category.append('far away distance')

elif distance <= q33:

distance\_category.append('nearby distance')

elif distance > q33 and distance < q66:

distance\_category.append('moderate distance')

df\_new['Distance Category'] = distance\_category

df\_new['Distance Category'].value\_counts()

Boruta:

forest = RandomForestClassifier(n\_jobs=-1, class\_weight='balanced', max\_depth=5)

forest.fit(df\_numeric\_x, list(df\_new['is\_fraud']))

feat\_selector = BorutaPy(forest, n\_estimators='auto', verbose=2, random\_state=1)

# find all relevant features

feat\_selector.fit(df\_numeric\_x.values, df\_new['is\_fraud'].values)

# check selected features

feat\_selector.support\_

# check ranking of features

feat\_selector.ranking\_

One Hot Encoding:

from sklearn.preprocessing import OneHotEncoder

from sklearn.preprocessing import LabelEncoder

from mlxtend.preprocessing import TransactionEncoder

ohenc = OneHotEncoder()

ohenc.fit(df\_cat\_fraud)

#ohenc.fit(df\_cat)

df\_ohenc = ohenc.transform(df\_cat\_fraud)

Frequent Itemsets:

from mlxtend.frequent\_patterns import apriori, association\_rules

# Extracting the most frequest itemsets via Mlxtend.

# The length column has been added to increase ease of filtering.

frequent\_itemsets = apriori(df\_final, min\_support=0.01, use\_colnames=True)

frequent\_itemsets['length'] = frequent\_itemsets['itemsets'].apply(lambda x: len(x))

# printing the frequent itemset

frequent\_itemsets