**OPIM 5604 PREDICTIVE MODELING**

**PROJECT 2**

**Credit Card Fraud Detection Report**

**Executive Summary:**

This report presents an analysis of credit card fraud detection using logistic regression, a decision tree, a boosted tree, a bootstrap forest, a neural network, a discriminant, a KNN, and a Naive Bayes. The dataset consists of transaction data including various features such as transaction date, customer information, merchant details, transaction amount, and fraud label. The goal is to build predictive models for identifying fraud transactions.

Dataset Source: This dataset is taken from <https://www.kaggle.com/datasets/kelvinkelue/credit-card-fraud-prediction?select=fraud+test.csv>

**Exploring the Data:**

Our data has 23 columns and 555,719 rows. Below are the details.

**Column 1:** “Column 1” => Exclude

Serial number of data not important for prediction.

A screenshot of a computer

Description automatically generated

**Column 2:** “trans\_date\_trans\_time” => Include

Looking at the data we will be splitting it into date and time, making two different columns. Also, the transactions range from 21/06/2020 to 31/12/2020 (Approx. 6 months).

Steps performed:

* Split the column with the delimiter “<space>” breaking down it into 3 new columns from **Cols > Utilities > Text to Column**.
* Combine the Time and AM/PM column together from **Cols > Utilities > Combine column**.
* Deleting the extra columns and renaming the newly formed columns.

A screenshot of a computer

Description automatically generated

Now we have 2 new column –

* **trans\_date** - having 194 unique dates which is approx. to 6 months as mentioned above. As the date won’t be a good predictor, we will be excluding it.
* **trans\_time** **- we will be grouping them as per daytime (like Morning, Afternoon, Evening and Night) later** so we will have 4 values only to make our model less complex.

**Column 3:** “cc\_num” => Exclude

Unique customer identification numbers are not important for prediction. The number consists of “e” value which after expanding also are not giving the full string making this column inconsiderable.

A screenshot of a computer

Description automatically generated

**Column 4:** “merchant” => Include

This column consists of 693 unique merchants. All the merchants have “Fraud\_” as the prefix in their name. So, we will be trimming it from the string. **Edit > Search > Find**.

A screenshot of a computer

Description automatically generated

Also, as there are a lot of unique merchants, so have to modify this column (if we are using it).

*//Below analysis is done to see if we should consider this column in our predictor, as it is difficult to modify.*

*// It is done after modifying all other columns*

A screenshot of a computer

Description automatically generated

As we can see merchant is Ranked 2 in our predictor screening. So we have to use this column and in order to do this **we will be modifying as much as possible later.**

**Column 5:** “category” => Include

This column shows the category of the transaction type (e.g., personal, childcare). We have 14 different categories in this. We will be keeping it as it is.

A screenshot of a computer screen

Description automatically generated

**Column 6:** “amt” => Include

This column represents the transaction amount. As per Summary Statistics, the mean transaction amount is $69.39. **Also, there may be some extreme values (outliers) which we will be checking in later**.

A screenshot of a computer screen

Description automatically generated

**Column 7:** “first” => Exclude

First name of cardholder not important for prediction

**Column 8:** “last” => Exclude

Last name of cardholder not important for prediction

**Column 9:** “gender” => Include

This column represents the gender of the cardholder. We have 2 categories in it I.e., Male and Female. The ratio of male and female in over data set is approx. 11:9 respectively.

A screenshot of a graph

Description automatically generated

**Column 10:** “street” => Exclude

This column represents the street address of the cardholder. We have an unorganized set of data in this column which if included will only make our model more complex. So, we will be excluding it.

**Column 11:** “City” => Include

This column represents the city of residence of the cardholder. As we will be using state for our modeling, we will be excluding this column.

**Column 12:** “state” => Include

This column represents the state of residence of the cardholder. Here we will be **tabulating the % of fraud and then we will recode them as high or low fraud rate state**.

**Column 13:** “zip” => Exclude

This column represents the zip code of residence of the cardholder. As we will be using state for our modeling, we will be excluding this column to avoid complexity.

**Column 14:** “lat” => Include

Latitude of cardholder's location. As we observed the latitude and longitude of cardholder and merchants are almost similar for no fraud.

For fraud detection, the distance between the cardholder's location and the merchant's location can be a valuable feature. Transactions with unusually long distances might be more likely to be fraudulent. (unless its online payment)

So, **we will be calculating The Great-Circle distance for it later and make a new column for it.**

**Column 15:** “long” => Include

Longitude of cardholder's location. As mentioned above, we will be calculating The Great-Circle distance for it later and make a new column for it.

**Column 16:** “city\_pop” => Include

Population of the cardholder's city.

A screenshot of a computer screen

Description automatically generated

**Column 17:** “job” => Include

Cardholder's job title. Here we have 478 unique categories of jobs. We will be later **grouping them as per fraud percentage** to reduce model complexicity.

**Column 18:** “dob” => Include

Date of birth of cardholder. This can be modified to get the age of the cardholder. Which can be correlated with the fraud, like if the is a certain age group being targeted. So, **we will be modifying it later**.

**Column 19:** “trans\_num” => Exclude

Unique transaction identifier

**Column 20:** “unix\_time” => Exclude

Transaction timestamp (Unix format), As we have transaction time we will not be using it.

**Column 21:** “merch\_lat” => Include

Latitude of merchant location. As mentioned above, we will be calculating The Great-Circle distance for it later and make a new column for it.

**Column 22:** “merch\_long” => Include

Longitude of merchant location. As mentioned above, we will be calculating The Great-Circle distance for it later and make a new column for it.

**Column 23:** “is\_fraud” => Include (Target Variable)

Fraudulent transaction indicator (1 = fraud, 0 = legitimate). This is the target variable for classification purposes.

**Data Preprocessing:**

1. **Explore missing values:** There are no missing values found in the data.

A screenshot of a computer

Description automatically generated

1. **Exploring NA values:** Tried to find NA values with all possible NA values such as ‘NA’, ‘N/A’, <blank> and ‘Not Available’. No NA values found.

A screenshot of a computer

Description automatically generated

1. **Data Types:**

trans\_time\_new, merchant\_fraud\_level, category, gender, job\_new – Nominal

amt, distance, city\_pop(Transform), age – Continuous

No changes are required for our predictive variables, but we need to change our target variable to categorical.

A screenshot of a computer

Description automatically generated

1. **Modifying columns:**

* **trans\_time**
  + Firstly, we divided the time in 24 different categories (as per hours), reducing it from 1,400+ categories.

A screenshot of a computer

Description automatically generated

* + Then we are recoding and re-grouping them as per daytime like Morning, Afternoon, Evening, and Night.

A screenshot of a computer

Description automatically generated

* + Delete the old columns and rename the new one.
* **lat, long, merch\_lat, merch\_long**
  + Creating a new JSL script that calculates the distance between two points.

*As I was not aware of the concept of Great-Circle distance, have taken the help of google for this part.*

A screenshot of a computer program

Description automatically generated

* + Then setting our data table variables accordingly with the formula.

A screenshot of a computer

Description automatically generated

* + Excluding and hiding the previous 4 columns.
* **state**
  + Exploring the % of fraud as per state.
  + Grouping them into severe, moderate, low, and no fraud states.
    - As minimum and maximum % of fraud per state is 0% and 1.66% respectively, we will make three equal partitions. i.e.,

0% - No

0.01 – 0.55% - Low

0.55 – 1.10% - Moderate

1.10 – 1.68% - Sever

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* **dob**
  + Making a new column to calculate age for better understanding the data.

A screenshot of a computer

Description automatically generated

* + Now we have ages from 19 to 99.

A graph of a number and a number

Description automatically generated with medium confidence

* **job**
  + Checking the fraud % of all the job types.
  + Grouping them together as per below.
    - Group 1 – 0%
    - Group 2 – 0-1%
    - Group 3 – 1-2%
    - Group 4 – 2-3%
    - Group 5 – 3-4%
    - Group 6 – 5-6%
    - Group 7 – 6-99% (as only 2 jobs were there)
    - Group 8 – 100%

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* **Merchant**
  + Distribution of fraud per merchant is as below. (using log scale)
  + After checking the fraud % for each merchant, the minimum and maximum % are 0 and 2.17 respectively.
  + So, we will be dividing them into 22 different levels of fraud. Like –

0 – 0.1

* 1. – 0.2
  2. – 0.3
  3. – 0.4

….. soon, till

2.1 – 2.2

A pink and blue background with black and white lines

Description automatically generated

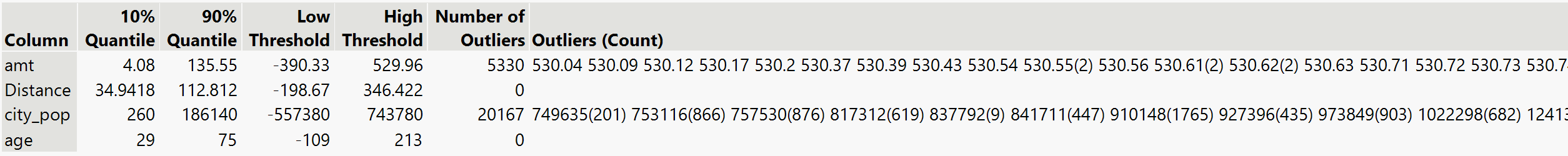
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

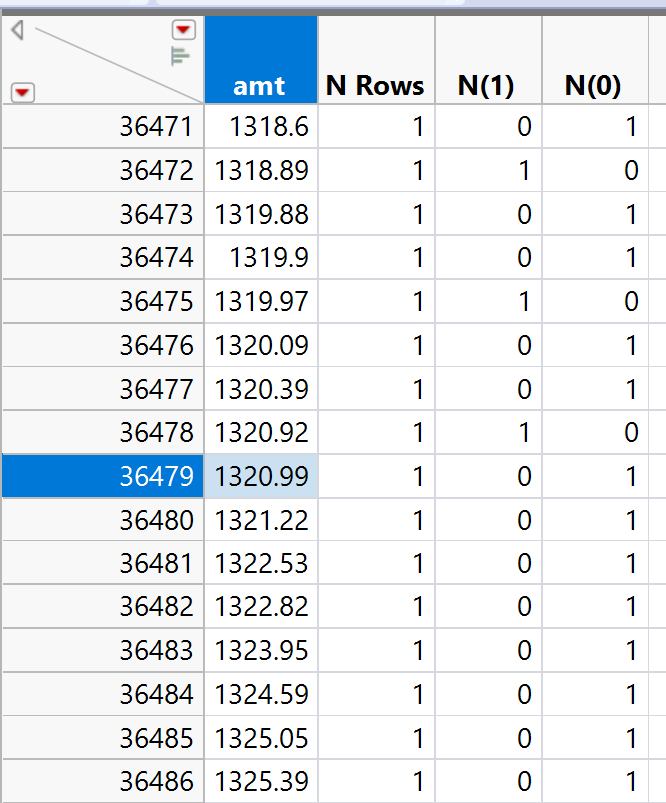
1. **Exploring Outliers**
   * There are no outliers in distance and age but there are many in amt and city\_pop.
   * For city\_pop we will be transforming the column.
     1. After that there are also some outliers, but we won’t be deleting those columns as these values are of big city (like New York, etc). So, will leave it. A screenshot of a computer

        Description automatically generated

A screenshot of a computer

Description automatically generated

* + For amt there are extreme values so we will be removing some row, so it won’t disturb our models as some models are sensitive to outliers.
    1. Exploring this column, there are no fraud recorded for amount more than $1,320. So, we will excluding the 780 row records for amount value more than $1,320. As these are extreme values.



A screen shot of a graph

Description automatically generated

**Data Partition:**

As dataset is not balanced, is fraud (1) is < 1% (0.39% in total), we will be doing oversampling using **Stratified Split Balanced Add-In**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Add-Ins:**

Setting random seed to ensure that results are reproducible.

**A screenshot of a computer

Description automatically generated**

**Modeling:**

As our target variable is a categorical variable, we will be doing logistic regression, a decision tree, a boosted tree, a bootstrap forest, a neural network, a discriminant, a KNN, and a Naive Bayes.

1. Logistic Regression: As some of the variables were not contributing a lot (from the PValue) to our target variable, **we removed them and checked the model performance, which resulted in increasing the true positive**.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a graph

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Decision Tree: With “Go” we got 47 splits with a true positive of 265 on validation. **On Prune we see the best split at 41**.

\*As small tree and leaf report were to big to include here, we have saved it in the data table.

A screenshot of a computer

Description automatically generated

A graph of a number of splits

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A graph on a computer screen

Description automatically generated

1. Boosted Tree: **We tried different settings for boosted trees, like the default one, some related splits per tree, etc.** Below is the best setting for the maximum true positives.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a graph

Description automatically generated

1. Bootstrap Forest: Same as boosted tree, **we tried different settings for bootstrap forest, like the default one, etc.** Below is the best setting for the maximum true positives.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a graph

Description automatically generated

1. Neural Network: Here also we have tried different setting approaches to get tot the best result.

When including all the predictor variables Model NTanH(5)NTanH2(5) is the best setting –

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a graph

Description automatically generated

Now as Neural network doesn’t have any variable selection process of its own, according to the p values we will be removing some of the low contribution variables.

After trying some, the performance was not better than the above one.

1. Discriminant:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a screen

Description automatically generated

A graph with a red line

Description automatically generated

1. KNN:

A graph with a line and a red line

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Naïve Bayes:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a graph

Description automatically generated

**Model Comparision:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated