UECS3213 UECS3453 UECS3483 DATA MINING ASSIGNMENT	
Programme(s)	Bachelor Of Science (Honours) Applied Mathematics with Computing Bachelor Of Engineering (Honours) Electronics (Computer Networking) Bachelor Of Science (Honours) Software Engineering
Trimester	January 2023
Course Lecturer	Dr. Fatimah Audah Md. Zaki
Submission Date	18 April 2023 (Week 12)
Submission Platform	WBLE

Student Information

Group no.	Student Name	Student ID	Programme Code	Signature	Final mark
6	Hannah Tan E-Ling	2005143	AM		
	Ho Jie Kai	2005840	AM		
	Lim Yen Khai	2005211	SE		

Course Learning Outcomes Assessed
CLO2: Create programming solutions using data mining techniques for given problem
CLO3: Evaluate performance of data mining solutions for a given problem
CLO4: Construct a data mining project as a team

1.0 Data Understanding

1.1 Dataset and Source of Dataset

Source: Kaggle

Kaggle is one of the platforms for data scientists to host and share the data projects, to learn from others to see how others work with the dataset and improve their skills and knowledge. Kaggle can also allow access to a large collection of publicly available datasets.

Dataset: Car Kick

When purchasing a used car at an auto auction, auto dealerships face a challenge. The auto dealership must determine whether the vehicle has serious issues and prevent it from being sold to customers. These unfortunate purchases are referred to as "kicks" in the auto industry.

The features of a kicked car may include tampered odometers, mechanical issues that the dealer is unable to address, problems to obtain the vehicle title from the seller, or some other unforeseen problem. The reason dealers tend to prevent “kicked cars” from being sold to customers is that they are very costly. Since "kick cars" are vehicles with serious problems, dealers must fix them at a high cost, including such transportation costs, throw-away repair work, and market losses when reselling the vehicle.

Researchers have to figure out which cars have a higher risk of being “kicked” so that dealerships can provide the best choices to their customers and prevent financial loss of auto dealership . The main purpose of this dataset is to predict if the car purchased at an auto auction is a “kick” (cars that are bad buys).

1.2 Summary of Variables in Dataset

There are a total 31 of columns and 67,212 of rows in the dataset. There are 19 columns in the dataset for numerical data and 12 for categorical data.

Variable Name	**Description of the Variable Name**
PurchDate	Purchase Date
VehYear	Year the car was produced
VehicleAge	Age of the car
VehOdo	How far the car has driven in km
MMRAcquisitionAuctionAveragePrice	Price of the car when is was bought at auction, average
MMRAcquisitionAuctionCleanPrice	Price of the car when it was bought at auction, before fees
MMRAcquisitionRetailAveragePrice	Price of the car when is was bought in a retail store, average
MMRAcquisitionRetailCleanPrice	Price of the car when is was bought in a retail store, before fees
MMRCurrentAuctionAveragePrice	Current price of the car at auction, average
MMRCurrentAuctionCleanPrice	Current price of the car at auction, before fees
VehBCost	Base price of the car

WarrantyCost	Cost of car warranty
Auction	Location of auction
Make	Producer of the car (automobile manufacturer)
Model	Model of the car
Trim	Trim level of the car (a version of a particular model with a particular configuration) The different trim levels offer varieties to the exterior and interior elements of a particular model, in addition to performance options, technologies, and even safety options. While many manufacturers today offer customers a wide array of trims, some models are still offered in a single configuration. Knowing how to differentiate trim levels will help to make a better choice. Some common trims: ● CE: Classic/Custom edition ● D/DL/DX: Deluxe ● EX/X: Extra

	● GL: Grade level ● GLE: Grade Level Extra ● GT: Grand Touring ● LE/LX: Luxury ● LTD: Limited ● S: Sport/Special/Standard ● SE: Sport edition/Special edition/Special equipment ● SL: Standard level ● T: Touring edition
SubModel	Submodel of the car
Color	Color of the car
Transmission	Type of transmission in the car (Manual / Auto)
WheelTypeID	ID of the wheelID (1 = Alloy, 2 = Covers, 3= Special)
WheelType	Type of wheel (Alloys, Covers, Special)

Nationality	Nationality of the car (American, Other, Other ASIAN, Top Line ASIAN)
Size	Size of the car (Compat, Crossover, Large, Large SUV, Large Truck, Medium, Medium SUV, Small SUV, Small Truck, Specialty, Sports, Van)
TopThreeAmericanName	Whether the car is from one of the three largest manufacturers in America (CHRYSLER, FORD, GM, Other)
BYRNO	Car registration number
VNZIP1	Car ZIP number (A garaging ZIP code or address refers to where you park your car the majority of the time, whether it's in a driveway, garage or along the street. Typically, that's where you live, at your primary residence, and so the ZIP code in your address would be the garaging ZIP.)
VNST	Car navigation system
IsOnlineSale	Whether the sale was online or not (0 = not sale, 1 = sale)
Class	Class of the car (0 = car in good condition, 1 = car purchased is a “kick”

Table 1.1: Variable name and its description.

1.3. Explanation on Dataset

Since the dataset is about to predict if the car purchased is “kick”, the target variable here is the “CLASS” column in the dataset where 0 represents a non-“kick” car and 1 represents a “kick” car.

“PurchDate” is representing the purchase date from the auction. The data object in this attribute is difficult to understand and lack prior documentation on its interpretability. Hence, “PurchDate” is recommended to be dropped from the dataset.

From the dataset, the column “VehYear” and “VehicleAge” represent the same attribute. Knowing that “VehYear” refers to Year the car was produced and “VehicleAge” refers to the Age of the car from Table 1.1. The "VehYear" was recorded between 2001 and 2010. While "VehicleAge" was recorded as a number ranging from 0 to 9. For example, if the "VehicleAge" is zero, the "VehYear" ranges from 2009 to 2010, indicating that the vehicle is brand new. If the "VehicleAge" is 1, the "VehYear" is from 2008 to 2009, indicating that the car is the second newest. Then, if the "VehicleAge" is 8, the "VehYear" is 2001-2002. If "VehicleAge" equals 9, "VehYear" equals 2001. As a result, it is clear that the "VehYear" and "VehicleAge" represent the year of the car. Since the “VehYear” is easier understood, suggest that “VehYear” remain in the dataset and “VehicleAge” be removed from the dataset. This suggestion will be confirmed after the correlation shown below.

According to Table 1.1, “WheelTypeID” and “WheelType”represent the same thing. “WheelTypeID” is just a more simple way to identify the “WheelType”, where 1 represents Alloy, 2 represents the Covers, and 3 represents Special. The “WheelType” will be used in the dataset since it is a more direct way to observe the data.

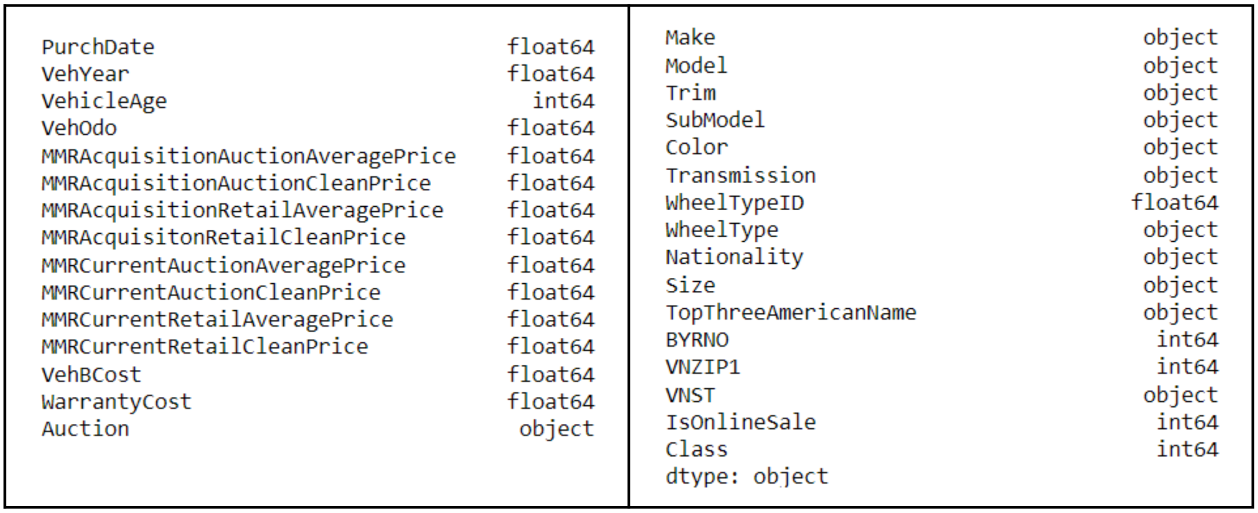
The "VNST" attribute contains uninformative values due to lack of documentation. It is known to refer to a car’s navigation system. But while the abbreviation of values are provided, their full names of the data object are not. However, there might be some relationship between other variables. It is necessary to visualize this variable in order to determine whether or not it has a relationship.

Other variables look independent and cannot see the hidden relationship. Data Mining has to be used here to exact the useful information from the dataset to know the relationship and the pattern.

2.0 Data Exploration

This section provides a thorough explanation of the findings from exploring the dataset, complemented by a set of visual and written descriptive statistics and summaries of the dataset, and points out any key features, patterns and/or trends found.

Our dataset consists of 67,211 rows and 31 columns, with initial data types as follows

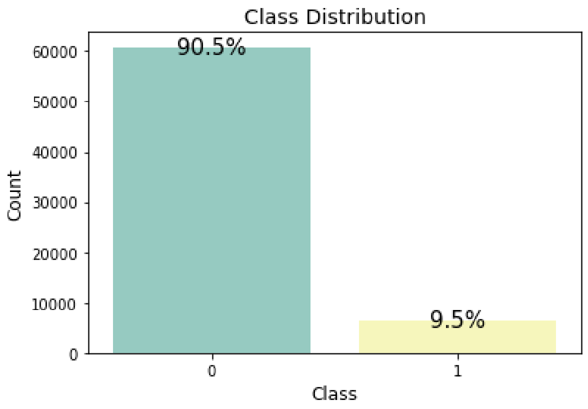


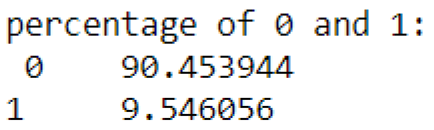
```
PurchDate                            float64
VehYear                              float64
VehicleAge                             int64
VehOdo                               float64
MMRAcquisitionAuctionAveragePrice    float64
MMRAcquisitionAuctionCleanPrice      float64
MMRAcquisitionRetailAveragePrice     float64
MMRAcquisitonRetailCleanPrice        float64
MMRCurrentAuctionAveragePrice        float64
MMRCurrentAuctionCleanPrice          float64
MMRCurrentRetailAveragePrice         float64
MMRCurrentRetailCleanPrice           float64
VehBCost                             float64
WarrantyCost                         float64
Auction                               object
```

```
Make                                  object
Model                                 object
Trim                                  object
SubModel                              object
Color                                 object
Transmission                          object
WheelTypeID                          float64
WheelType                             object
Nationality                           object
Size                                  object
TopThreeAmericanName                  object
BYRNO                                  int64
VNZIP1                                 int64
VNST                                  object
IsOnlineSale                           int64
Class                                  int64
dtype: object
```

Figure: Datatypes of columns

A simple plot on the target variable, Class, was made and it is shown that 90% of the dataset is classified under value ‘0’, that is, non-kick. This demonstrates that the dataset is highly imbalanced.



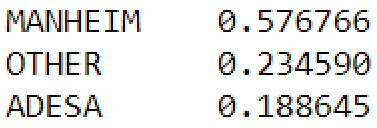


```
percentage of 0 and 1:
 0    90.453944
1     9.546056
```

Figure: Percentage distribution of class values

2.1 Data Exploration on Categorical Data

Under categorical variables, the relevant columns of this subset include ‘Auction’, ‘Model’, ‘Trim’, ‘Submodel’, ‘Colour’, ‘Transmission’, ‘WheelType’ and ‘WheelTypeID’, ‘Nationality’, ‘Size’, ‘Make’, ‘TopThreeAmericanName’, ‘VNST’, ‘IsOnlineSale’ and ‘VNZIP1’. The distribution of data in terms of these categorical variables were investigated, and interesting outputs were generated. Firstly, 58% of cars purchased were from MANEIM, while only 19% of cars were purchased from ADESA.



```
MANHEIM    0.576766
OTHER      0.234590
ADESA      0.188645
```

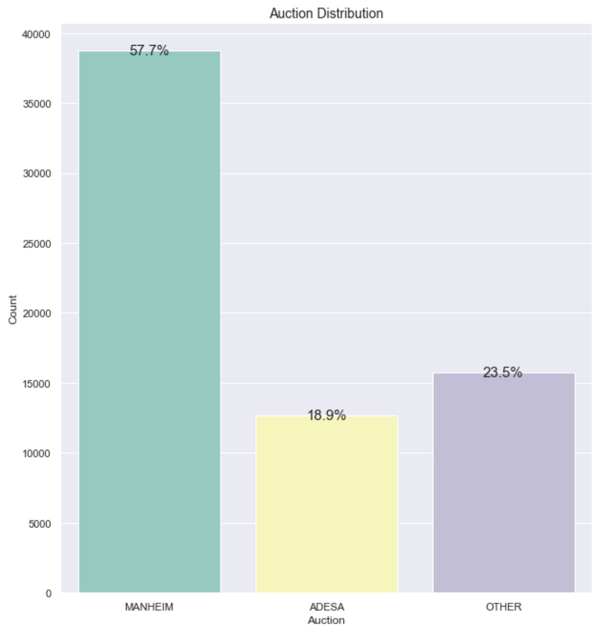


Figure: Percentage distribution of Auction values

Columns ‘Model’, ‘Trim’ and ‘Submodel’ consisted of over a hundred distinct values, with the ‘Model’ column having 953 rows, followed by ‘Submodel’, with 823 rows, and ‘Trim’, with 133 rows. These numbers indicate that further binning is required under the data processing section to assess more distinct categorizations with fewer distinct row values.

In terms of colour preferences, the top 3 colors of cars purchased were silver, white and blue, indicating that more muted colours were preferred as compared to more vibrant colours like purple and orange.

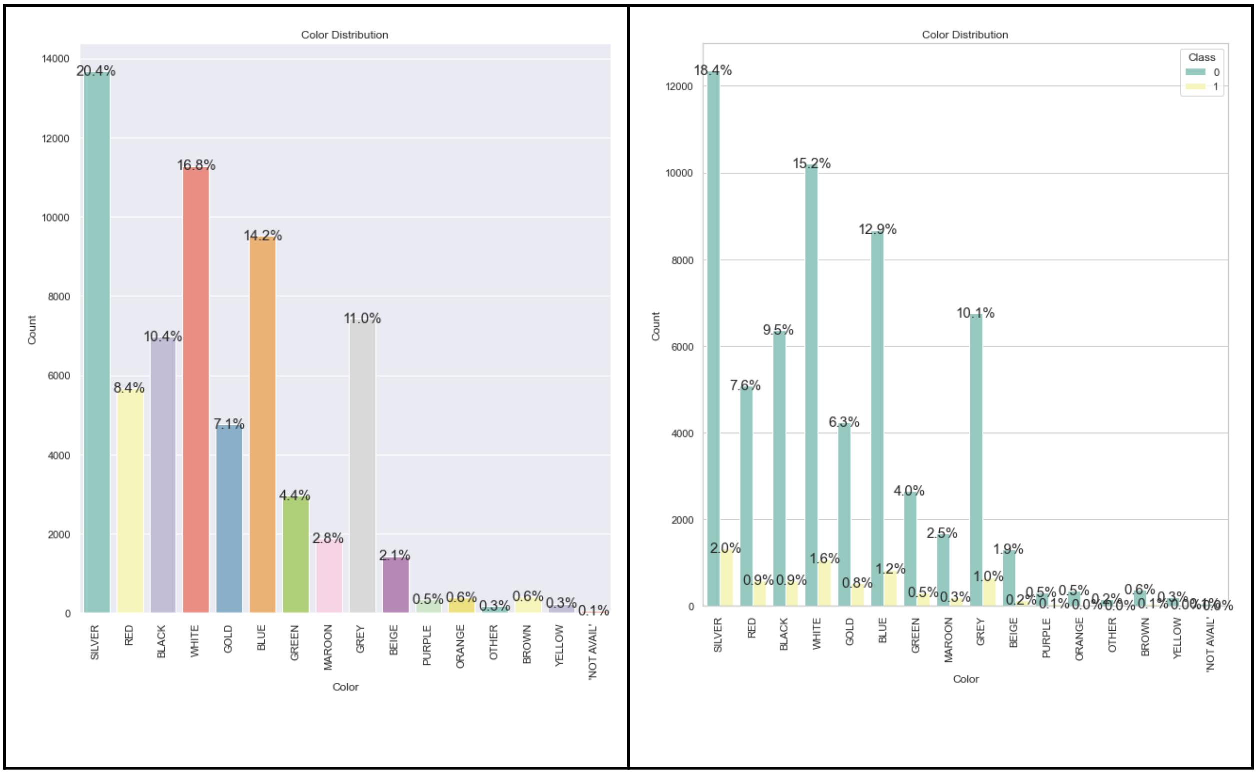
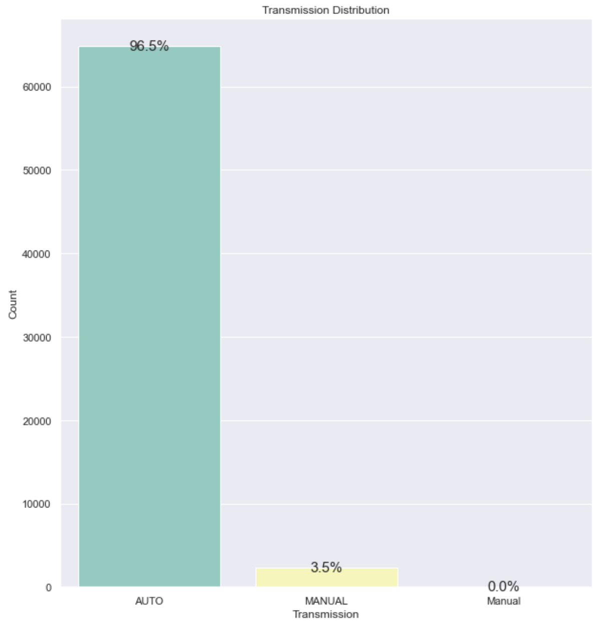
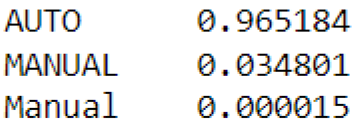


Figure: Percentage distribution of color values

When the ‘Transmission’ column was investigated, spelling inconsistencies were revealed, which were adjusted in the data cleaning section. But generally, 97% of cars purchased were auto, while the remaining cars are manual.

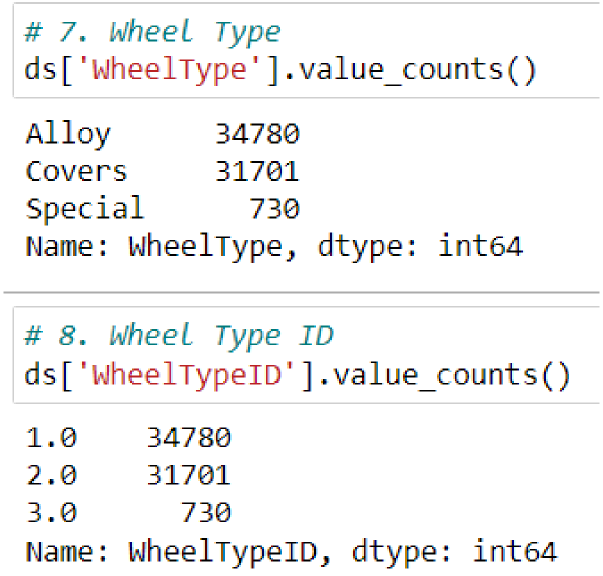




```
AUTO      0.965184
MANUAL    0.034801
Manual    0.000015
```

Figure: Percentage distribution of Transmission values

Columns ‘WheelType’ and ‘WheelTypeID’ were proven to be referring to the same attribute, as their corresponding frequencies for the different column values are the same. Alloy and Covers consisted about 98% of data, with the minimal exception of Special wheel type cars.



```
# 7. Wheel Type
ds['WheelType'].value_counts()
```

```
Alloy      34780
Covers     31701
Special      730
Name: WheelType, dtype: int64
```

```
# 8. Wheel Type ID
ds['WheelTypeID'].value_counts()
```

```
1.0    34780
2.0    31701
3.0      730
Name: WheelTypeID, dtype: int64
```

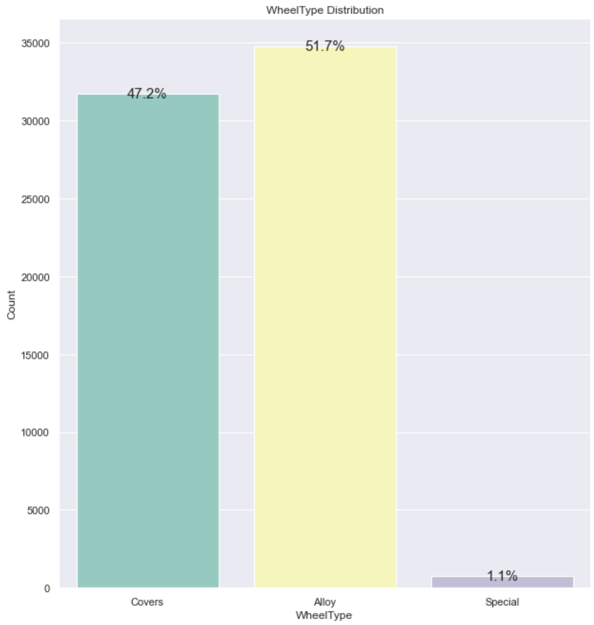
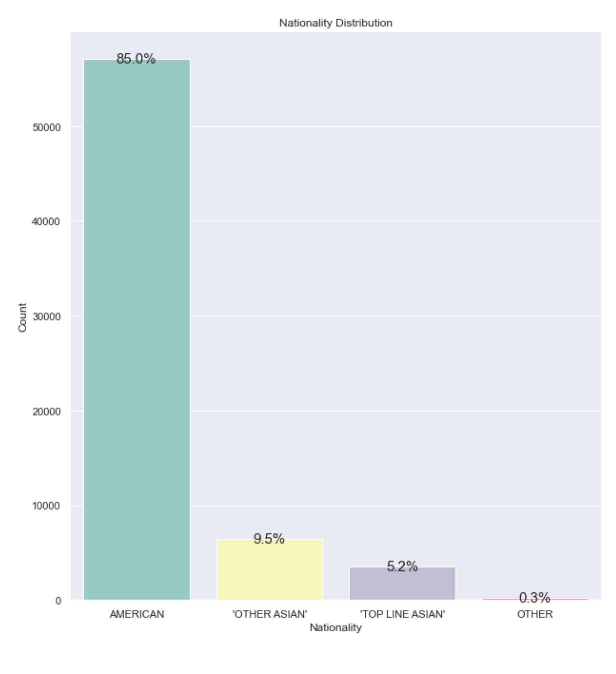
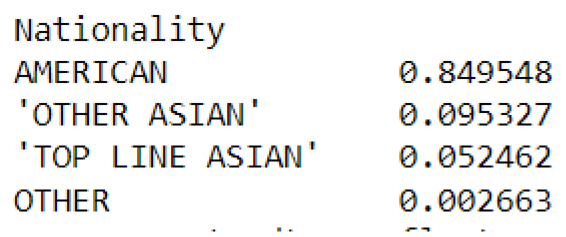


Figure: Numerical similarities between WheelType & WheeltypeID; Percentage distribution of WheelType values

Investigating the cars' nationality, it can be seen that 85% of cars purchased are of American nationality, while the remaining are either Asian or some other nationality. This is consistent to the location of research, which indicates that local cars are more common.

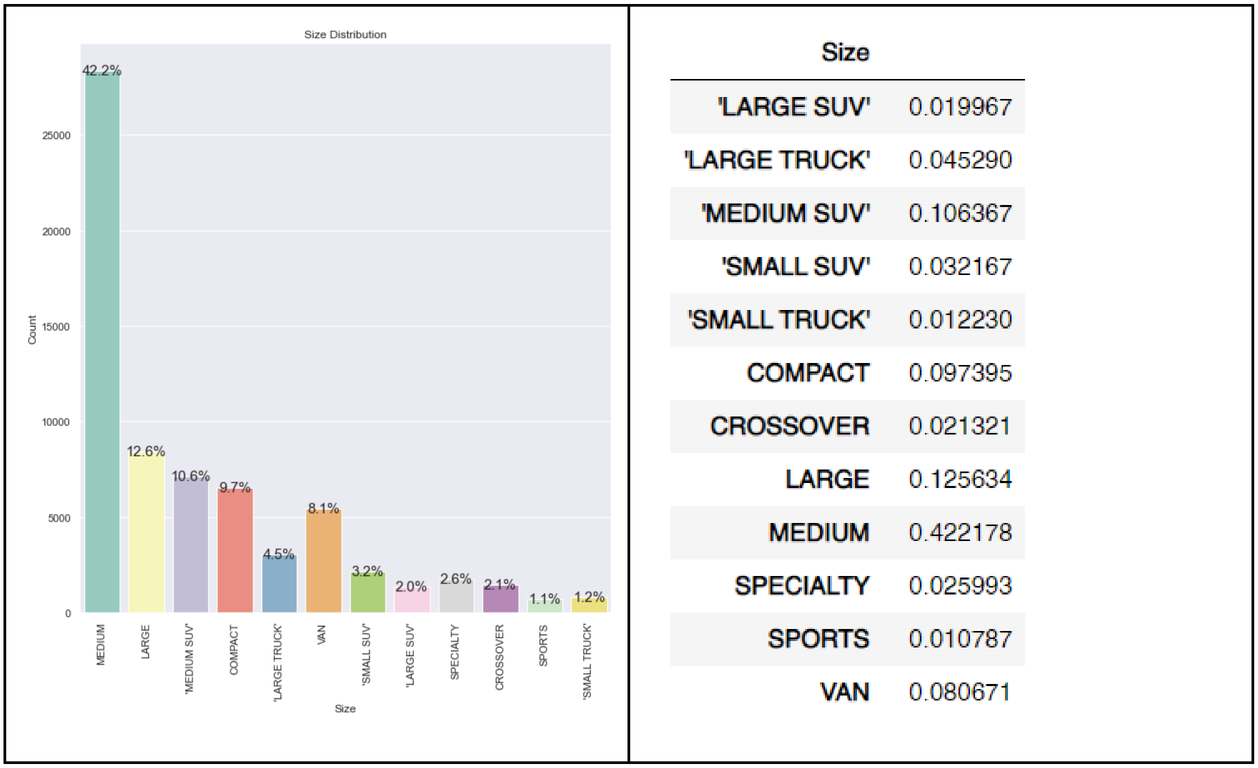




```
Nationality
AMERICAN                0.849548
'OTHER ASIAN'           0.095327
'TOP LINE ASIAN'        0.052462
OTHER                   0.002663
```

Figure: Percentage distribution of Nationality values

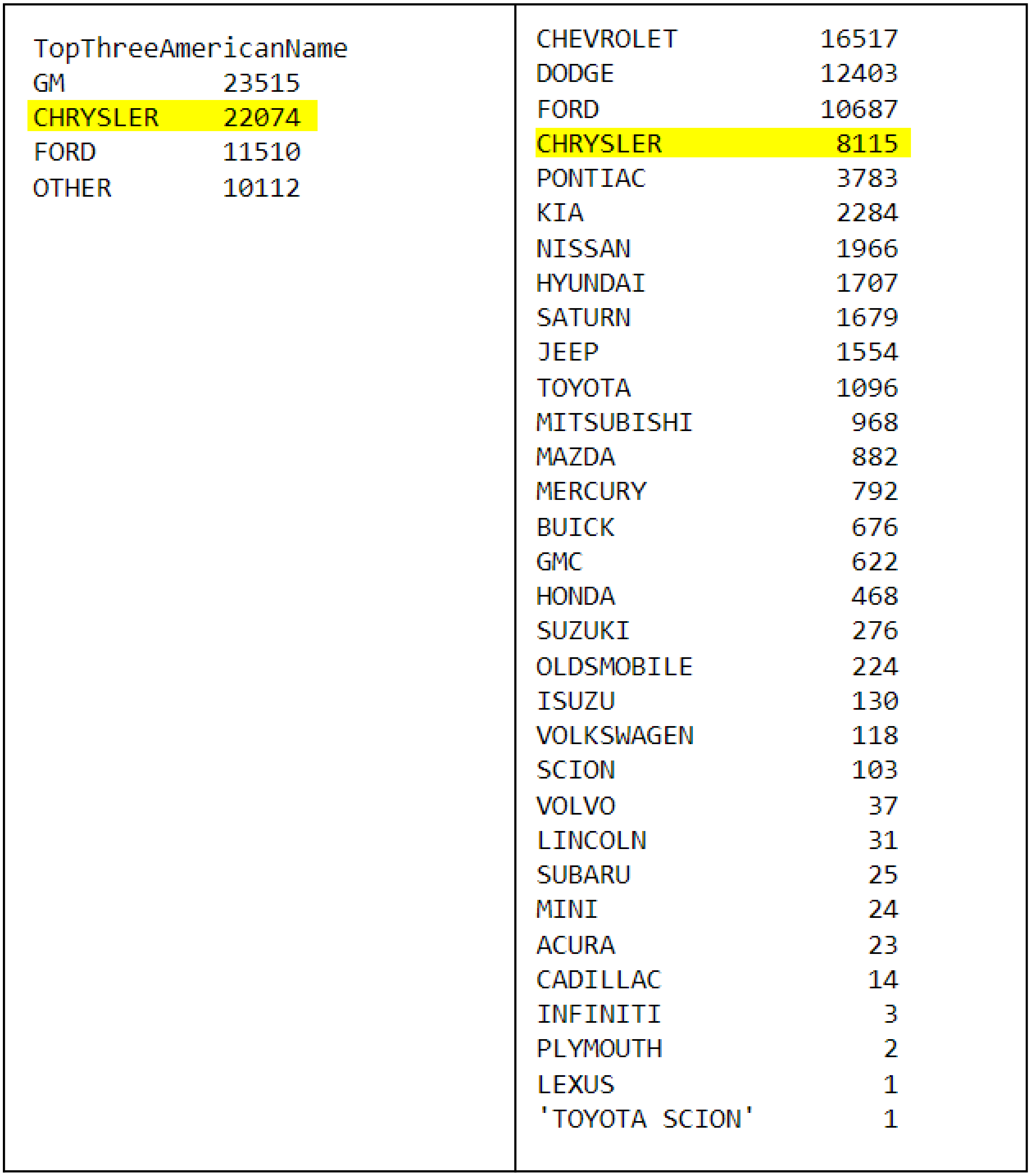
The car size distribution shows that 42.2% of cars are medium sized, while the rarest of car sizes are sports cars, consisting of only 1.1%. Upon observation, the values of this column require cleaning for consistency, in which single quotes ‘’ should be removed so as to improve cleanliness of data.



Size	
'LARGE SUV'	0.019967
'LARGE TRUCK'	0.045290
'MEDIUM SUV'	0.106367
'SMALL SUV'	0.032167
'SMALL TRUCK'	0.012230
COMPACT	0.097395
CROSSOVER	0.021321
LARGE	0.125634
MEDIUM	0.422178
SPECIALTY	0.025993
SPORTS	0.010787
VAN	0.080671

Figure: Percentage distribution of Size values

The ‘TopThreeAmericanName’ column is meant to be a derivation of the ‘Make’ column, which groups values from make into only one of 4 groups. However, when comparing overlapping values, the number of matches for the same manufacturer name was inconsistent. This is illustrated below:



```
TopThreeAmericanName
GM              23515
CHRYSLER        22074
FORD            11510
OTHER           10112
```

```
CHEVROLET         16517
DODGE             12403
FORD              10687
CHRYSLER           8115
PONTIAC            3783
KIA                2284
NISSAN             1966
HYUNDAI            1707
SATURN             1679
JEEP               1554
TOYOTA             1096
MITSUBISHI          968
MAZDA               882
MERCURY             792
BUICK               676
GMC                 622
HONDA               468
SUZUKI              276
OLDSMOBILE          224
ISUZU               130
VOLKSWAGEN          118
SCION               103
VOLVO                37
LINCOLN              31
SUBARU               25
MINI                 24
ACURA                23
CADILLAC             14
INFINITI              3
PLYMOUTH              2
LEXUS                 1
'TOYOTA SCION'        1
```

Figure: Absolute value comparison between TopThreeAmericalBrand and Make

Number of CHRYSLER's in 'TopThreeAmericanName' = 22074. While the number of CHRYSLER's in 'Make' = 8115. This suggests that one of the two columns are incorrect, and to prevent loss of information through generalization, the ‘Make’ column, from which

‘TopThreeAmericanName’ was supposedly derived should be maintained, while the former should be dropped.

Exploring the column with car navigation systems, it can be seen that the cars purchased are distributed into over 30 different navigation systems, with TX consisting of the highest percentage (19%).

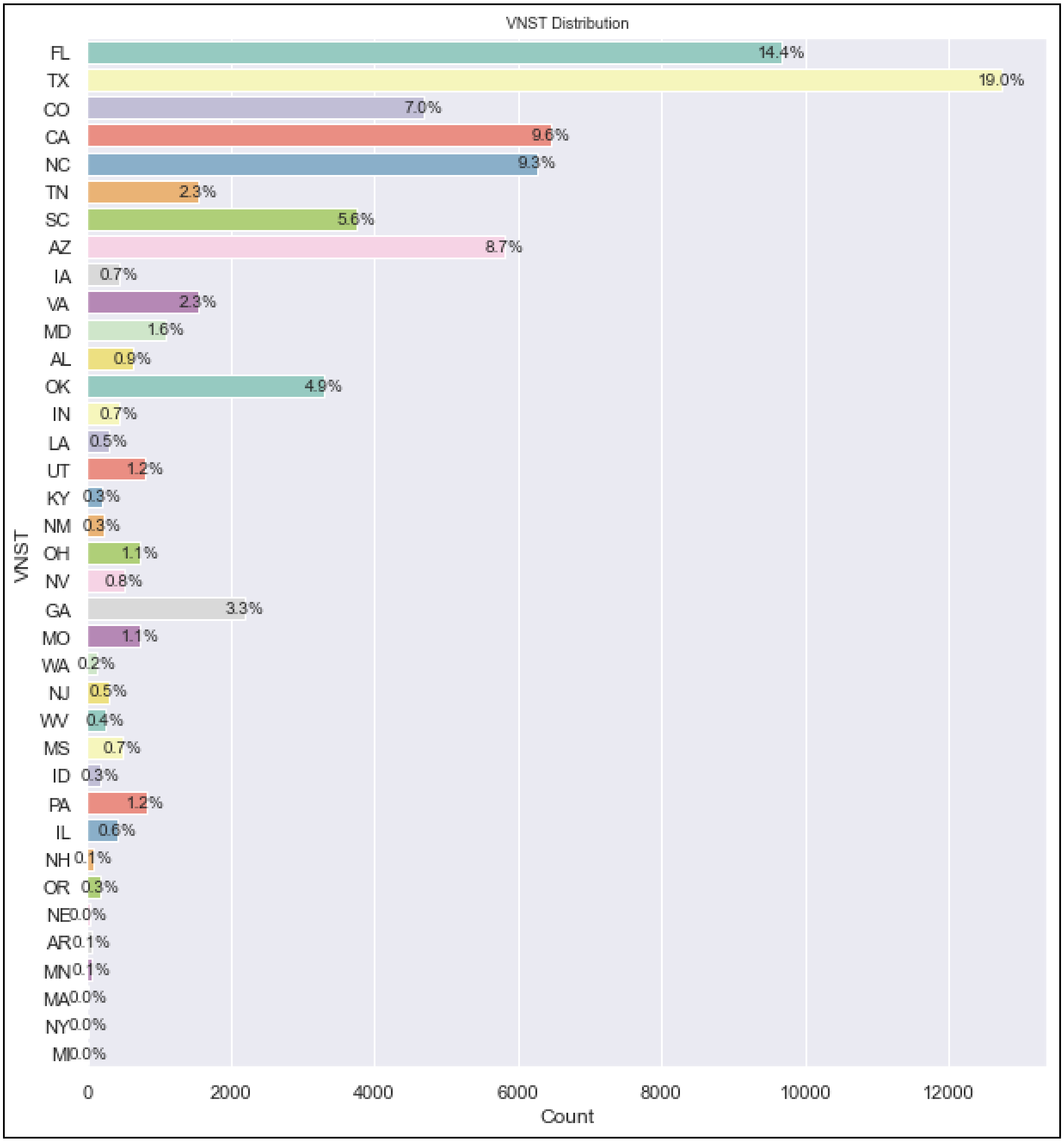


Figure: Percentage distribution of VNIST values

Observing if cars were purchased from an online sale, it is displayed that most cars were not purchased from a sale. Only a scarce amount of 2.5% cars were purchased through an online sale. Uncertainty regarding the legitimacy of purchasing the car online could be a sign of kicked cars, Further investigations on this column of data show that relatively, 8% (134/(1550+134)*100) of cars bought through an online sale were kicked cars, whereas 9.6% (6282/(6282+59245)*100) of the cars bought physically were kicked cars. These roughly similar figures suggest that our hypothesis relating illegitimacy of purchasing a car through an online sale with the classification of a kicked car is rejected.

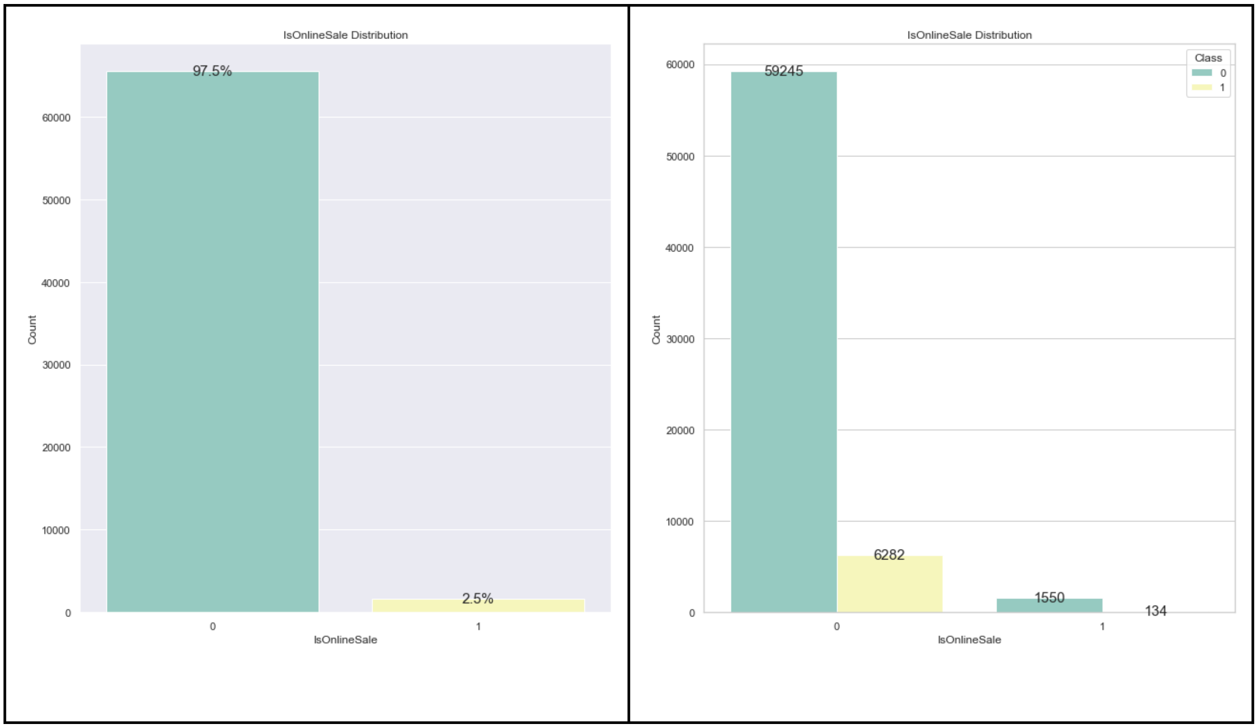
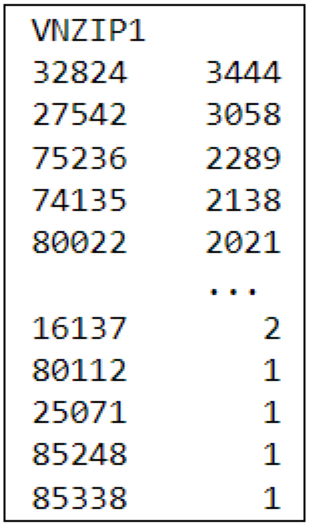


Figure: Percentage distribution of IsOnlineSale values; Percentage distribution of IsOnlineSale values with respect to class values

The 'VNZIP1' column contains 150 distinct values, arranged in this manner. The large number of distinct values imply that further binning is required in the clearing step for further investigation.



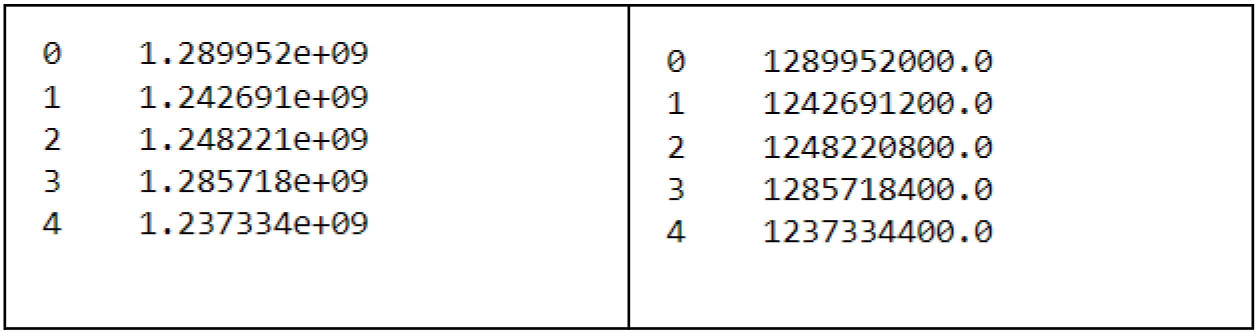
```
VNZIP1
32824    3444
27542    3058
75236    2289
74135    2138
80022    2021
         ...
16137       2
80112       1
25071       1
85248       1
85338       1
```

Figure: VNZIP1 data distribution

2.2 Data Exploration on Numerical Data

Under numerical variables, the relevant columns of this subset include ‘PurchDate’, ‘VehYear’, ‘VehicleAge’, ‘VehOdo’, ‘MMRAcquisitionAuctionAveragePrice’, ‘MMRAcquisitionAuctionCleanPrice’, ‘MMRAcquisitionRetailAveragePrice’, ‘MMRAcquisitonRetailCleanPrice’, ‘MMRCurrentAuctionAveragePrice’, ‘MMRCurrentAuctionCleanPrice’, ‘MMRCurrentRetailAveragePrice’, ‘MMRCurrentRetailCleanPrice’, ‘VehBCost’, and ‘WarrantyCost’. The distribution of data in terms of these numerical variables were investigated.

Firstly, the format of the ‘PurchDate’ values was puzzling, and with lack of documentation on this, the interpretation of this column of data was undetermined and so provided no useful information. Thus, this column should be dropped in the cleaning section.



```
0    1.289952e+09
1    1.242691e+09
2    1.248221e+09
3    1.285718e+09
4    1.237334e+09
```

```
0    1289952000.0
1    1242691200.0
2    1248220800.0
3    1285718400.0
4    1237334400.0
```

Figure: Original and updated datatype of PurchDate values

Plotting the graph for year of vehicle production, and graph for vehicle age, it was deduced that the two were obviously correlated to each other, as the latter could be determined from the former. This was demonstrated in the horizontal mirroring effect of the graphs on each other, i.e. notice how VehYear is left skewed while VehicleAge is right skewed.

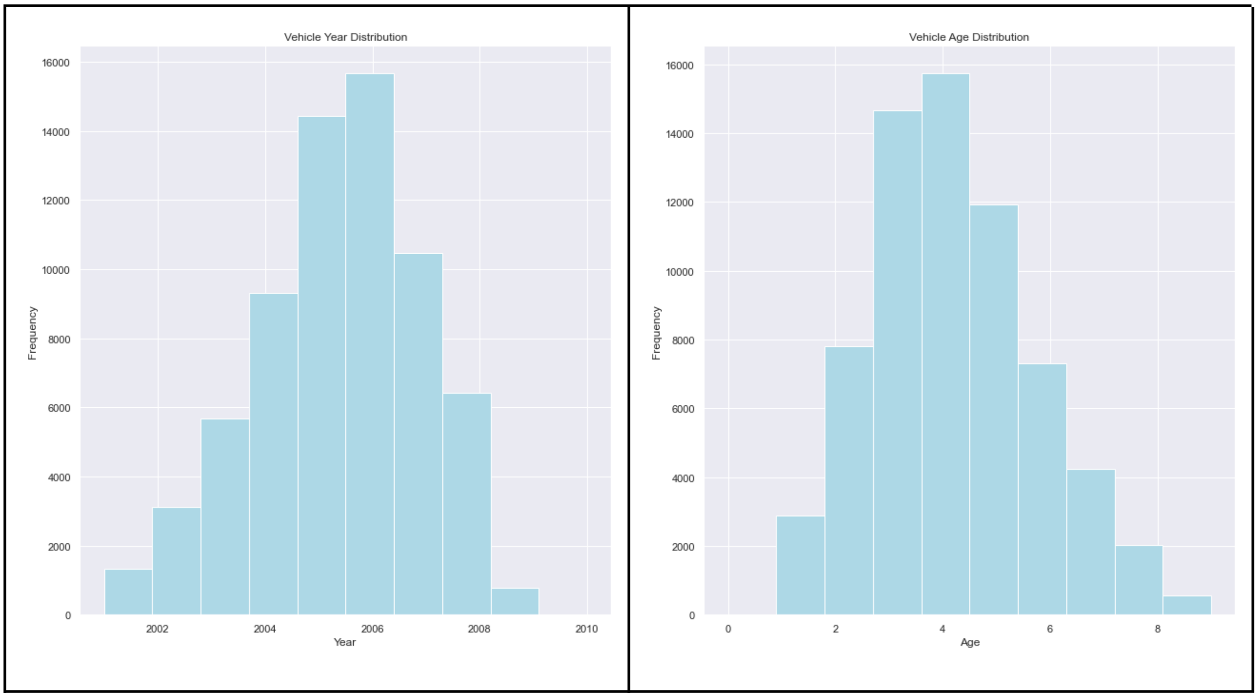


Figure: Distribution of Vehicle year and Vehicle age

The values of the vehicle odometer readings show a left-tailed graph, where more cars purchased were driven over longer distances around 80,000km. Prior hypothesis was that larger vehicle ages of lower ODO would be kicked as it would be suspected that car odometer was tempered with. However deeper digging revealed vehicles with higher ODOs are kicked, despite vehicle age. Therefore the hypothesis is invalidated.

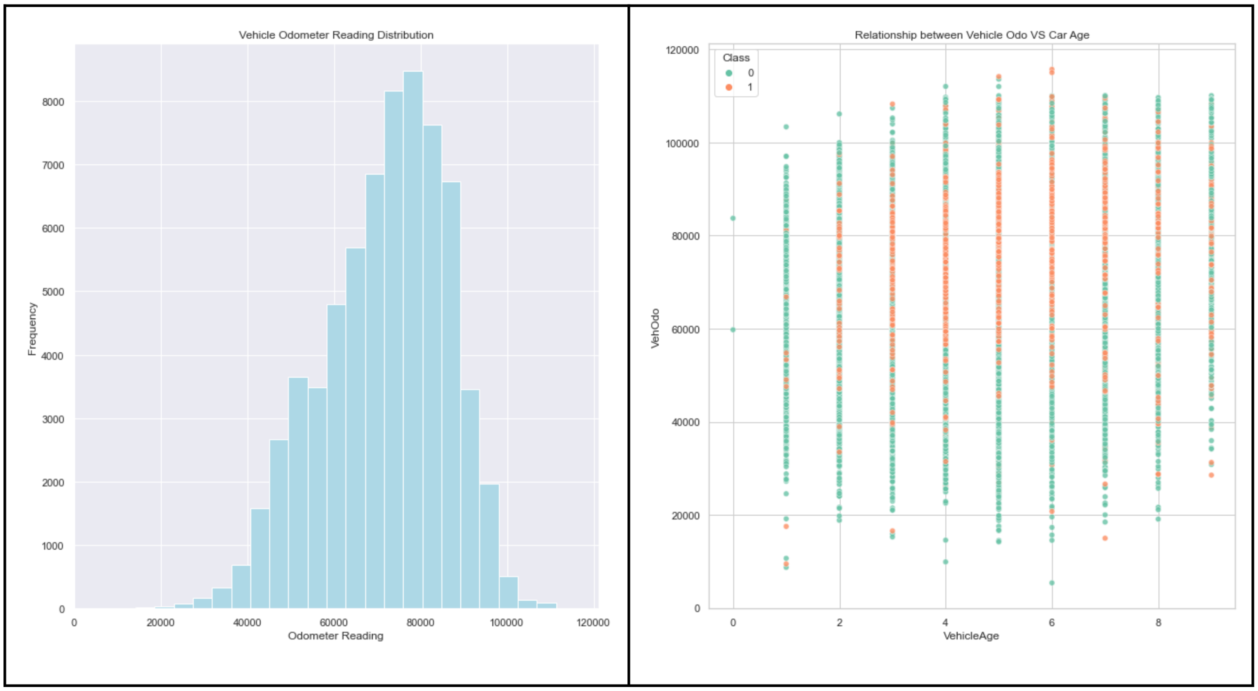


Figure: Distribution of vehicle Odometer readings; Scatter plot between VehOdo and VehicleAge

Next, it was suspected, by mere means of analyzing the meaning of the following column names, that these columns are correlated to each other. By plotting their individual distributions, the shape of data seems to be similar, with the only difference in left or right shift. Specifically, the cleaned prices are generally a positive shift of the average prices. The retail prices are also a positive shift of its corresponding acquisition or current prices.

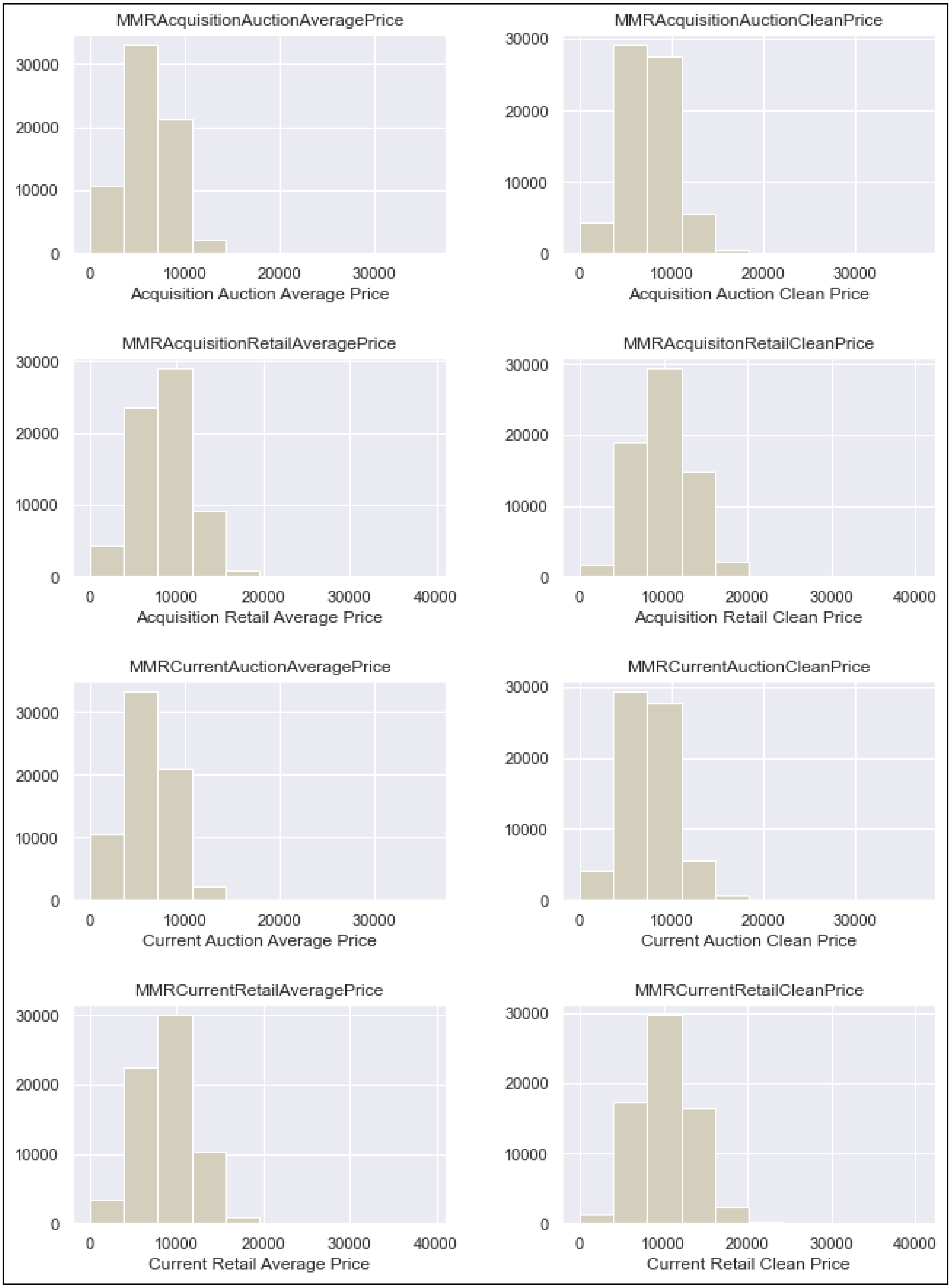

MMRAcquisitionAuctionAveragePrice
30000
20000
10000
0
0
10000
20000
30000
Acquisition Auction Average Price
MMRAcquisitionAuctionCleanPrice
30000
20000
10000
0
0
10000
20000
30000
Acquisition Auction Clean Price
MMRAcquisitionRetailAveragePrice
30000
20000
10000
0
0
10000
20000
30000
40000
Acquisition Retail Average Price
MMRAcquisitonRetailCleanPrice
30000
20000
10000
0
0
10000
20000
30000
40000
Acquisition Retail Clean Price
MMRCurrentAuctionAveragePrice
30000
20000
10000
0
0
10000
20000
30000
Current Auction Average Price
MMRCurrentAuctionCleanPrice
30000
20000
10000
0
0
10000
20000
30000
Current Auction Clean Price
MMRCurrentRetailAveragePrice
30000
20000
10000
0
0
10000
20000
30000
40000
Current Retail Average Price
MMRCurrentRetailCleanPrice
30000
20000
10000
0
0
10000
20000
30000
40000
Current Retail Clean Price

Figure: Distribution of ‘MMR’ type attributes

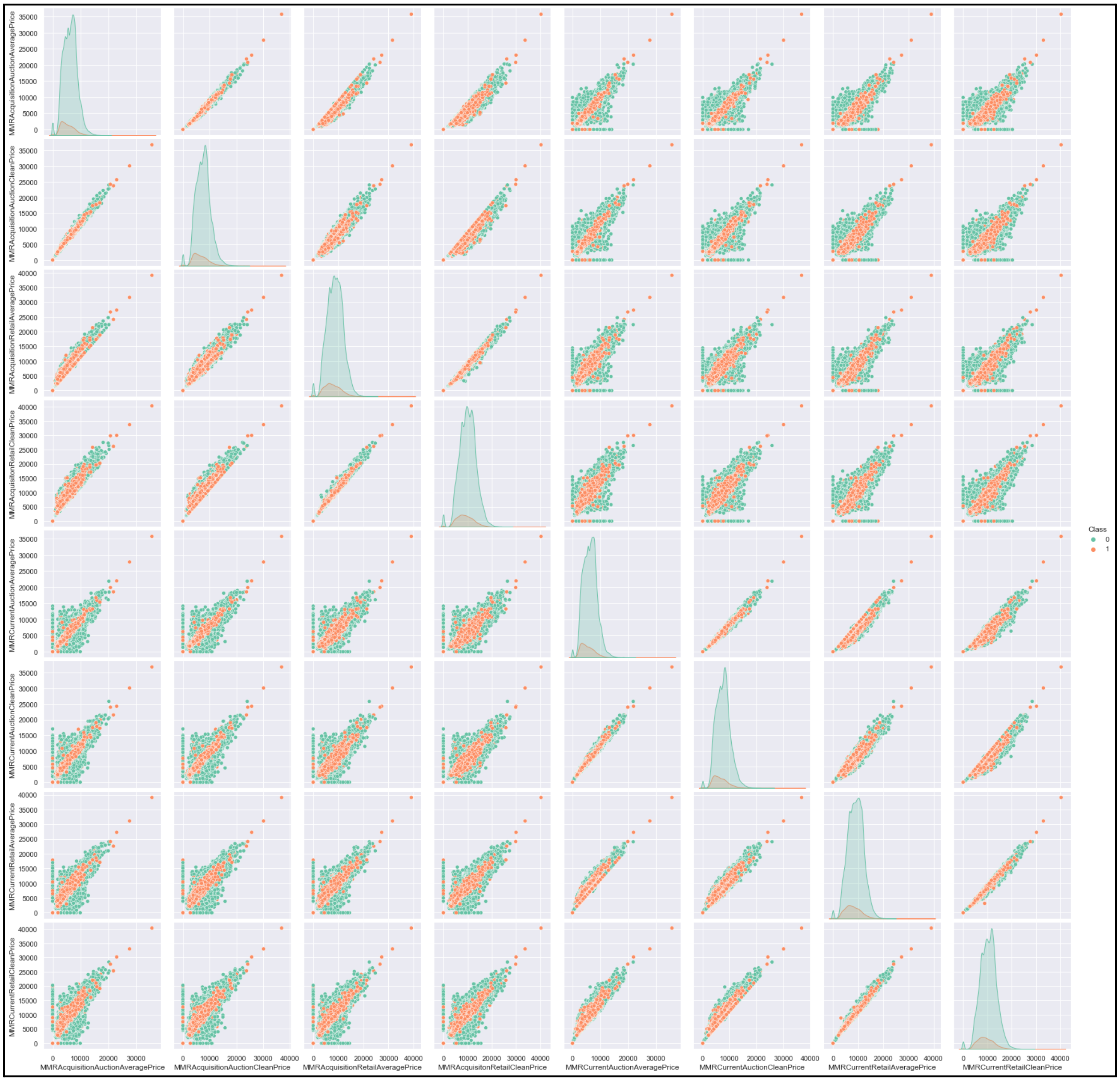


Figure: Pairplot of all ‘MMR’ type attributes

Creating a pairplot for the aforementioned columns show a very strong correlation among them. This is further supported by the correlation matrix results as shown below, where a prominent yellow box represent the correlations between stipulated columns.

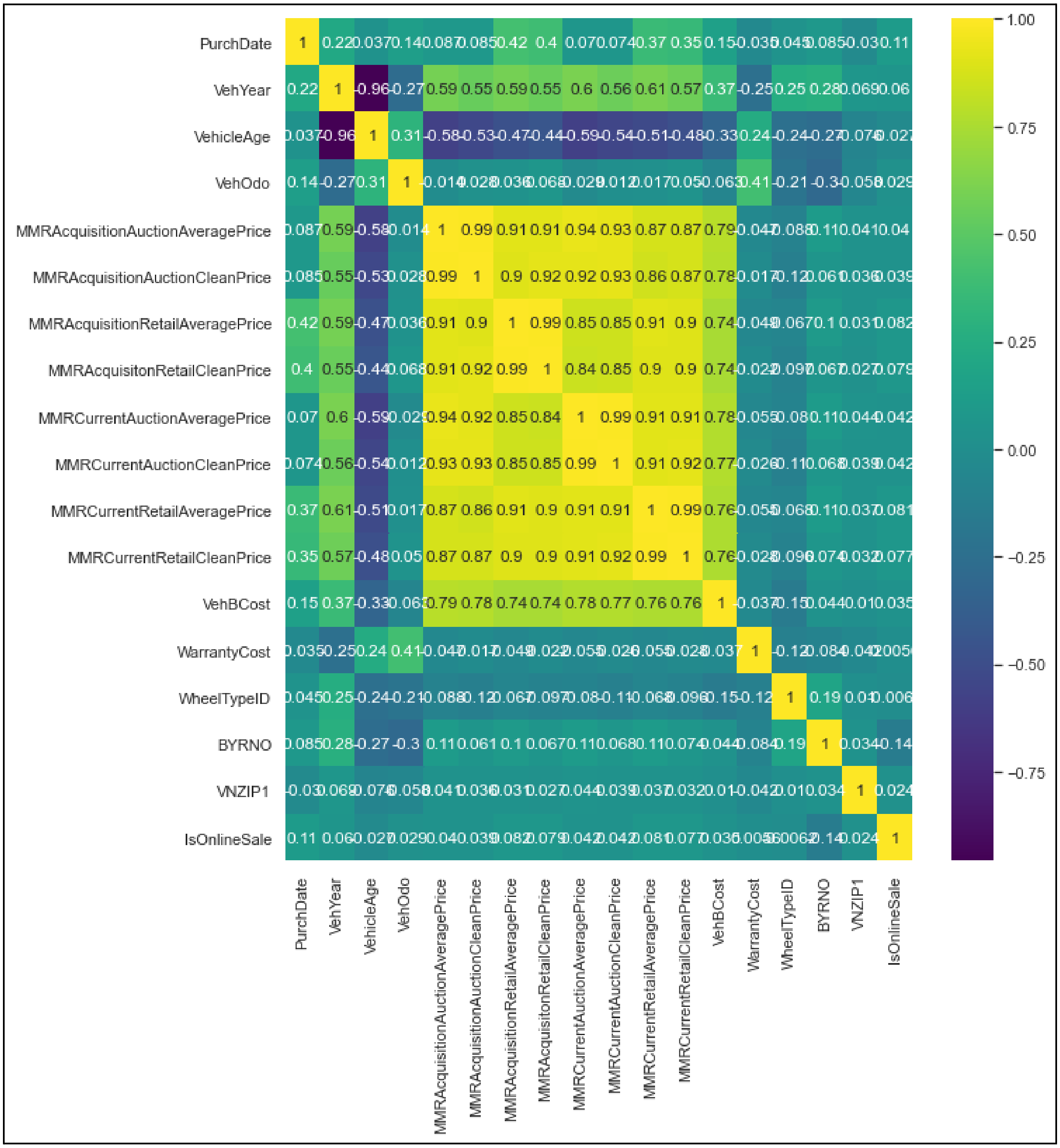


Figure: Correlation matrix of numerical data

Cleaning of data is required in this section to reduce the dimensionality of the dataset.

Vehicle Buying price was investigated and results show that the buying price center around 7000 per car, for which currency was not documented so it is unknown, but most likely USD.

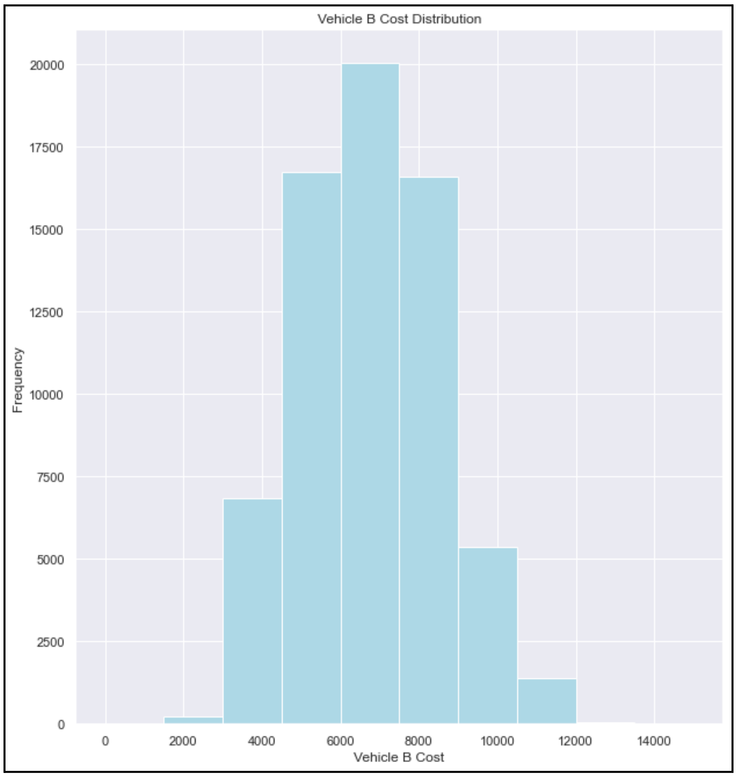


Figure: Distribution of VehBCost values

Warranty cost, on the other hand, is right skewed, with most common cost price between 1000 - 2000, about 20% of the vehicle buying price.

2.3 Comparisons with Respect to Class

There were generally no significant differences between the pattern of distribution of data among kick vs non-kick cars. Ones with notable differences are mentioned hare, while the rest are omitted.

Odometer readings of kicked cars are generally higher ODO. This means that cars that were less used are more considered to be good buys.

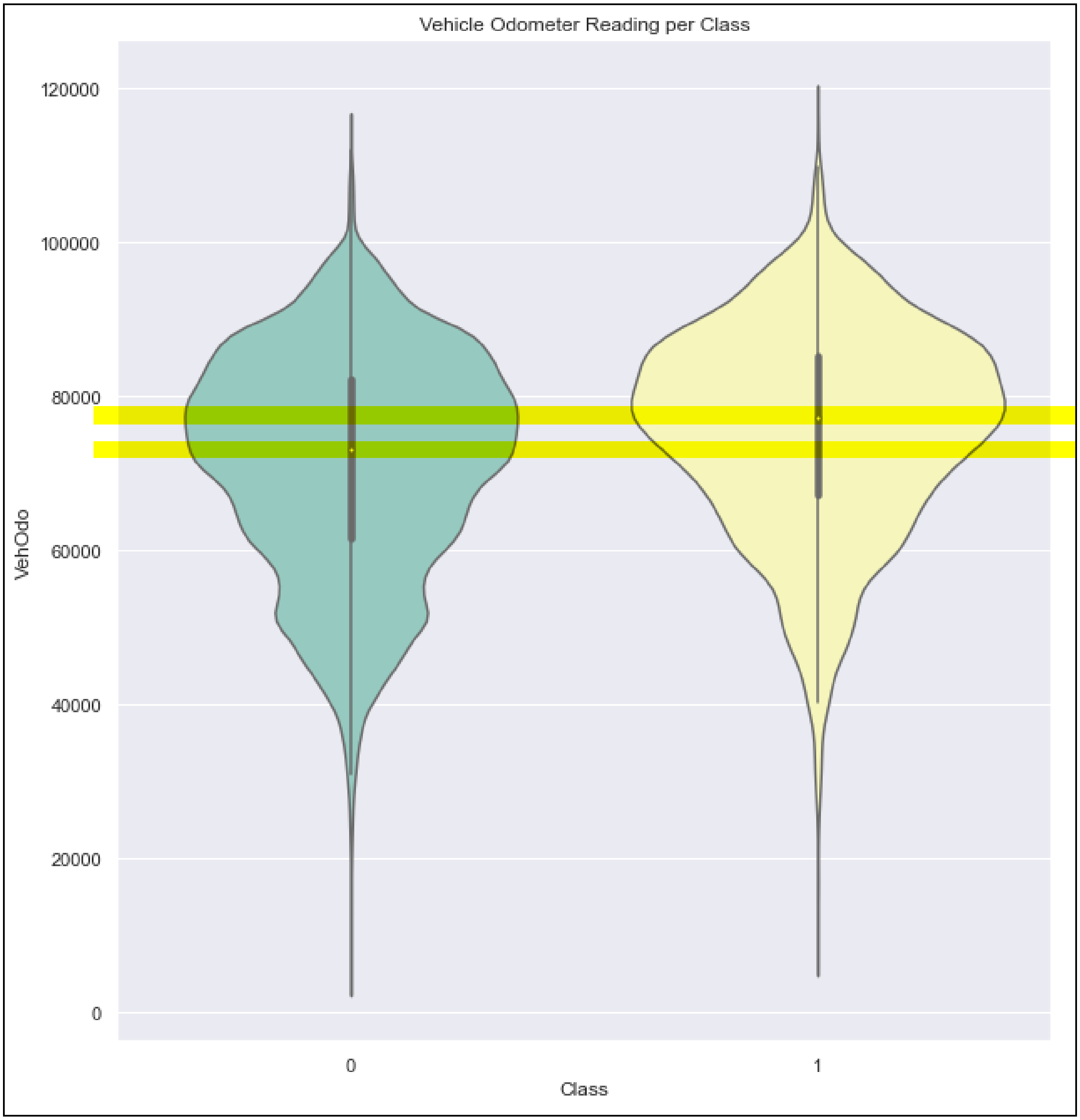


Figure: Odometer reading comparison between kicked and non-kicked cars

Investigating the split between kick vs non-kick cars by 'Make' showed that 66.7% of cars manufactured by INFINITI are kicked. The second highest kicked car manufacturer is POLYMOUTH, with 50% of kicked cars. Conversely, cars manufactured by TOYOTA SCION, LEXUS and VOLVO are 100% non-kicked, indicating that cars of these brands are preferred.

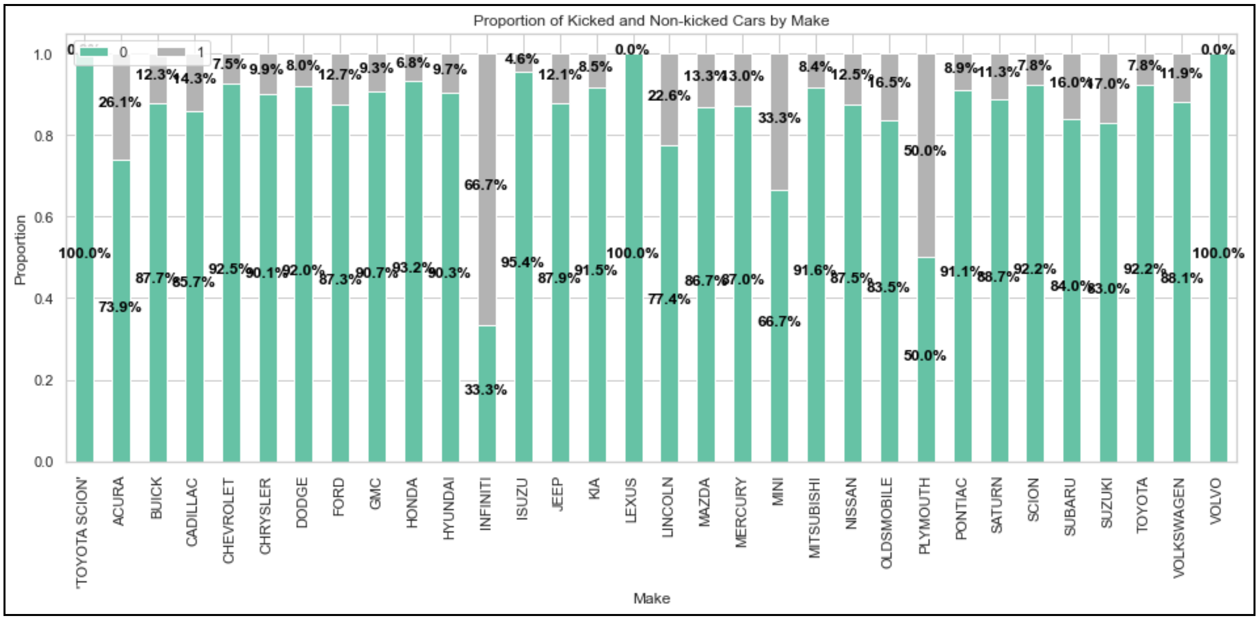


Figure: Percentage proportion of kicked and non-kicked cars’ Make values

2.4 Null Values and Outlier Handling

The dataset was checked for null values and outliers. The dataset was clean of any null value, but had a significant number of outliers. The outliers in this dataset were replaced by a median value of the dataset excluding outliers for the relevant columns.

Below shows one of the examples for this instance

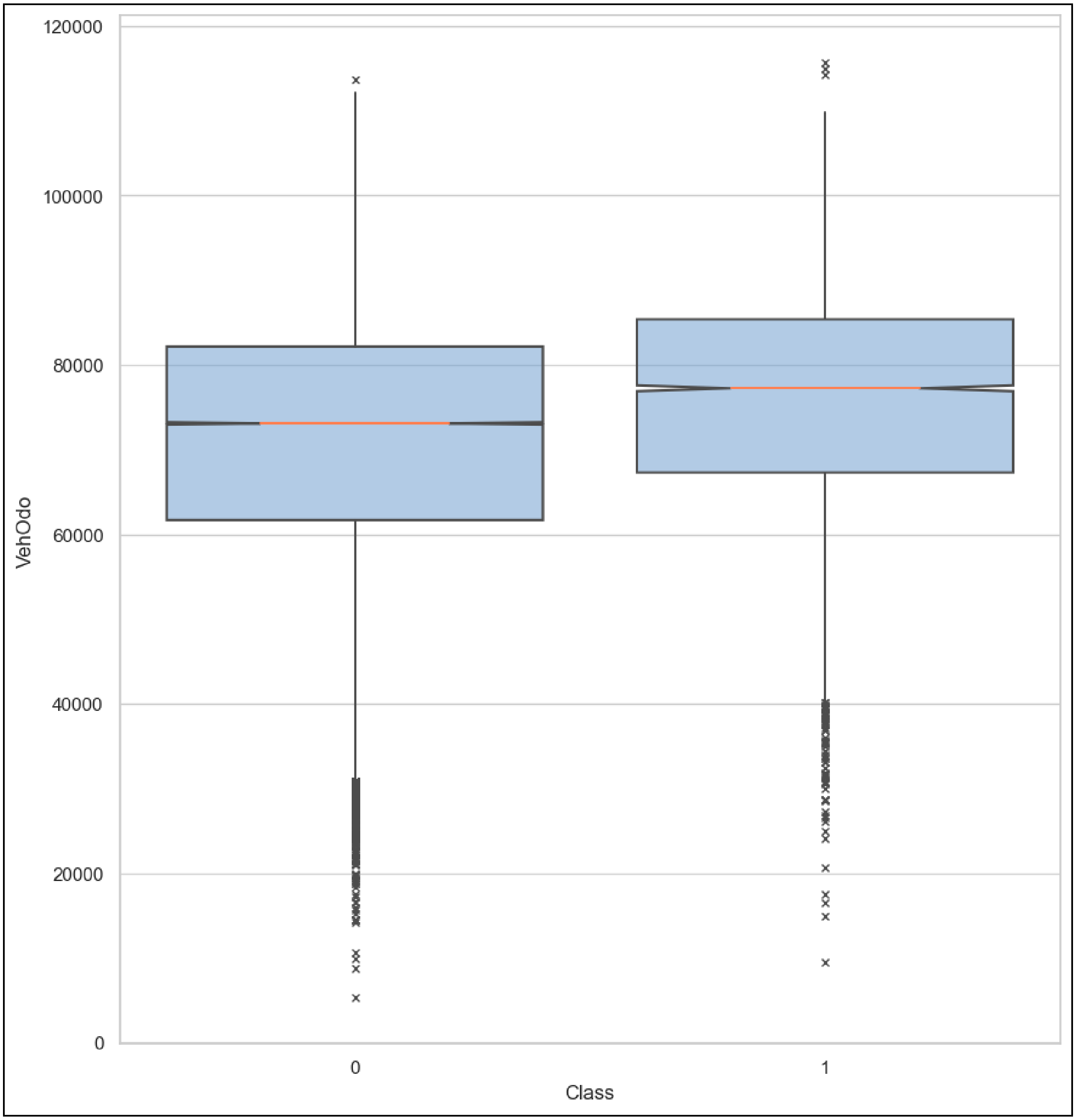
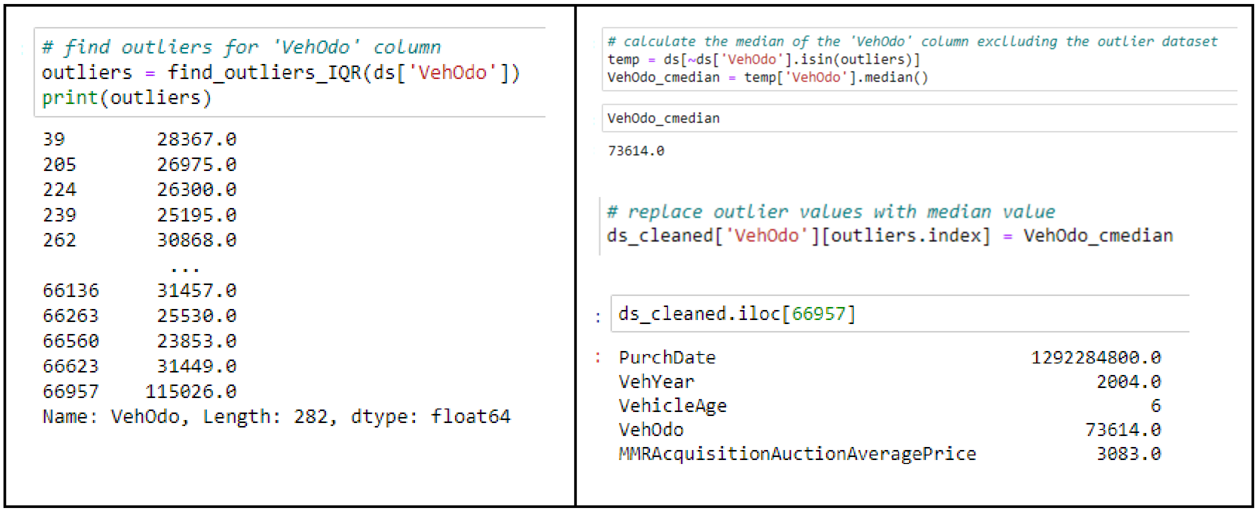


Figure: Boxplot of VehOdo values



```
# find outliers for 'VehOdo' column
outliers = find_outliers_IQR(ds['VehOdo'])
print(outliers)
```

```
39        28367.0
205       26975.0
224       26300.0
239       25195.0
262       30868.0
           ...   
66136     31457.0
66263     25530.0
66560     23853.0
66623     31449.0
66957    115026.0
Name: VehOdo, Length: 282, dtype: float64
```

```
# calculate the median of the 'VehOdo' column exclluding the outlier dataset
temp = ds[~ds['VehOdo'].isin(outliers)]
VehOdo_cmedian = temp['VehOdo'].median()
```

```
VehOdo_cmedian
```

```
73614.0
```

```
# replace outlier values with median value
ds_cleaned['VehOdo'][outliers.index] = VehOdo_cmedian
```

```
ds_cleaned.iloc[66957]
```

```
PurchDate                            1292284800.0
VehYear                                    2004.0
VehicleAge                                      6
VehOdo                                    73614.0
MMRAcquisitionAuctionAveragePrice          3083.0
```

Figure: Procedure to alter output values with median values

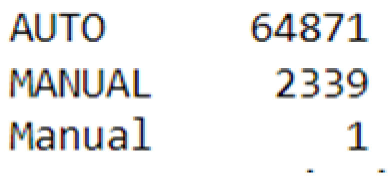
Thus, this procedure of replacing outliers is repeated for numerical columns ‘MMRAcquisitionAuctionCleanPrice’, ‘MMRCurrentRetailCleanPrice’, ‘VehBCost’, and ‘WarrantyCost’. The remaining of the ‘MMR’ columns were not adjusted for outliers as the decision for dropping them were made with the exchange of creating a new column of data, which is discussed in further detail in the next section.

3.0 Data Preprocessing

3.1 Data Cleaning

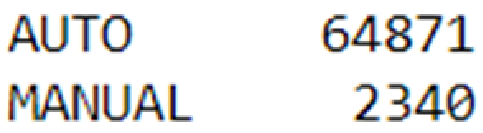
As previously stated, 'PurchDate' must be removed because it is uninterpretable. Furthermore, because 'VehYear' provides a more direct way to represent the data, 'VehAge' is opted be removed from the collection. Analysing 'WheelType' and 'WheelTypeID, it is decided to drop one of the two attributes. Since 'WheelType' has more direct information than 'WheelTypeID,' 'WheelTypeID' is removed from the dataset. There are inconsistencies between columns 'Make' and 'TopThreeAmericanName,' which divide rows into Ford, Chrysler, GM, or other vehicle makers groups. To avoid such inconsistencies as mentioned in the earlier section, 'TopThreeAmericanName' should be eliminated. Data cleaning needed to deal with those data that have missing value, incomplete data, noisy data, inconsistent data and intentional data. Data inconsistency prevails in categorical data, although there are no missing values. Actions should be taken accordingly.

In ‘Transmission’, there is only one inconsistent data, which is Manual, so can substitute the only Manual into MANUAL.



```
AUTO         64871
MANUAL        2339
Manual           1
```

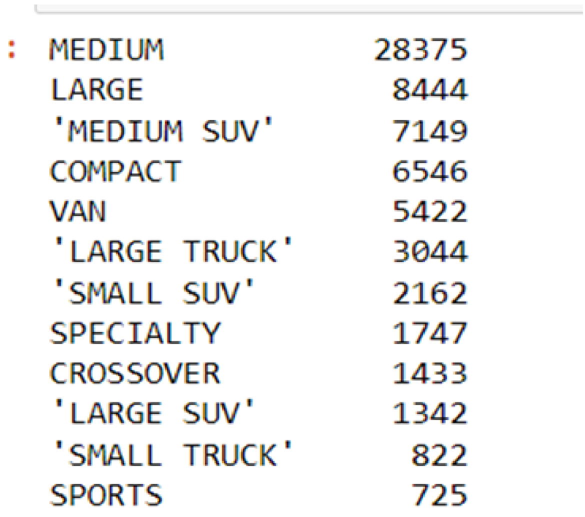
Figure: Raw data from ‘Transmission’



```
AUTO         64871
MANUAL        2340
```

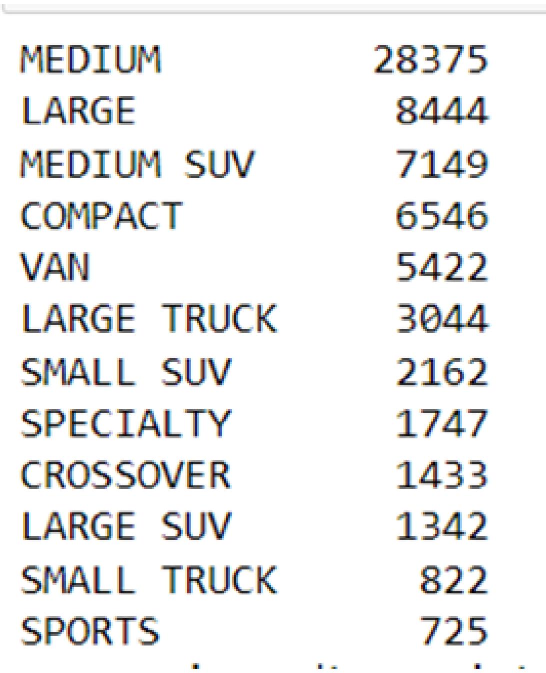
Figure: Data from ‘Transmission’

There are inconsistent data in ‘Size’, ‘Model’, ‘SubModel’, and ‘Nationality’. Some of the data objects have single quotes, but some of them do not. To make it consistent, it was decided to remove all the single quotes from the model.



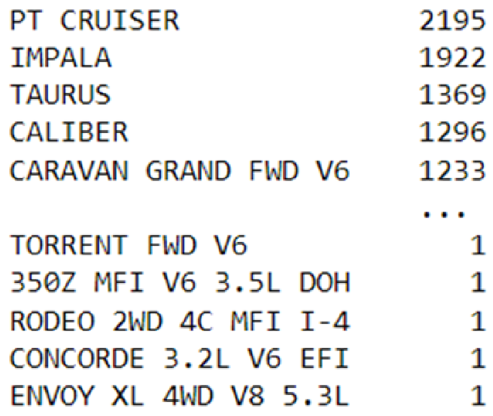
```
: MEDIUM            28375
  LARGE              8444
  'MEDIUM SUV'       7149
  COMPACT            6546
  VAN                5422
  'LARGE TRUCK'      3044
  'SMALL SUV'        2162
  SPECIALTY          1747
  CROSSOVER          1433
  'LARGE SUV'        1342
  'SMALL TRUCK'       822
  SPORTS              725
```

Figure: Raw data from ‘Size’

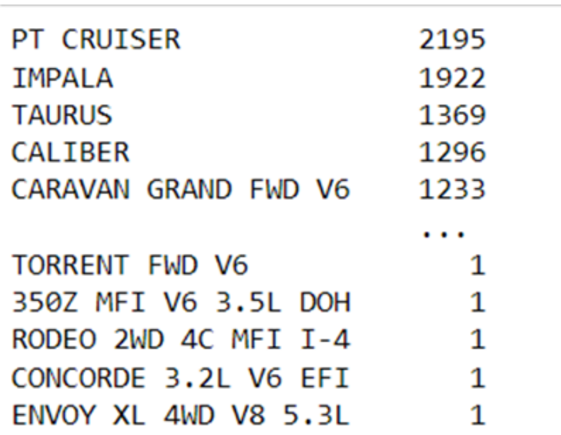


```
MEDIUM         28375
LARGE           8444
MEDIUM SUV      7149
COMPACT         6546
VAN             5422
LARGE TRUCK     3044
SMALL SUV       2162
SPECIALTY       1747
CROSSOVER       1433
LARGE SUV       1342
SMALL TRUCK      822
SPORTS           725
```

Figure: After removing single quote from ‘Size’



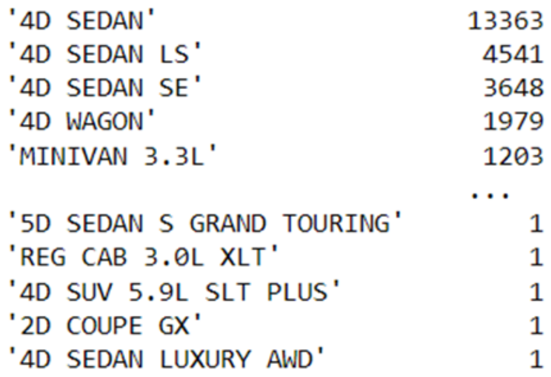
```
PT CRUISER                2195
IMPALA                    1922
TAURUS                    1369
CALIBER                   1296
CARAVAN GRAND FWD V6      1233
                          ...
TORRENT FWD V6               1
350Z MFI V6 3.5L DOH         1
RODEO 2WD 4C MFI I-4         1
CONCORDE 3.2L V6 EFI         1
ENVOY XL 4WD V8 5.3L         1
```



```
PT CRUISER                2195
IMPALA                    1922
TAURUS                    1369
CALIBER                   1296
CARAVAN GRAND FWD V6      1233
                          ...
TORRENT FWD V6               1
350Z MFI V6 3.5L DOH         1
RODEO 2WD 4C MFI I-4         1
CONCORDE 3.2L V6 EFI         1
ENVOY XL 4WD V8 5.3L         1
```

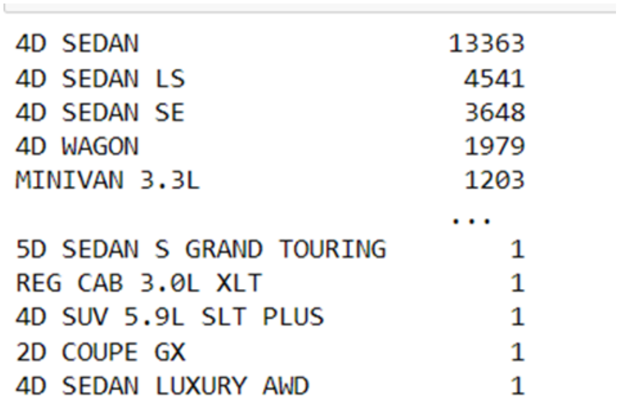
Figure: Raw data from ‘Model’

Figure: Raw data from ‘Model’



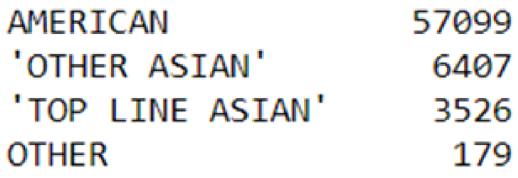
```
'4D SEDAN'                      13363
'4D SEDAN LS'                    4541
'4D SEDAN SE'                    3648
'4D WAGON'                       1979
'MINIVAN 3.3L'                   1203
                                 ...
'5D SEDAN S GRAND TOURING'          1
'REG CAB 3.0L XLT'                  1
'4D SUV 5.9L SLT PLUS'              1
'2D COUPE GX'                       1
'4D SEDAN LUXURY AWD'               1
```

Figure: Raw data from ‘SubModel’



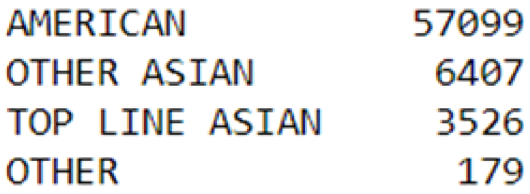
```
4D SEDAN                       13363
4D SEDAN LS                     4541
4D SEDAN SE                     3648
4D WAGON                        1979
MINIVAN 3.3L                    1203
                                 ...
5D SEDAN S GRAND TOURING           1
REG CAB 3.0L XLT                   1
4D SUV 5.9L SLT PLUS               1
2D COUPE GX                        1
4D SEDAN LUXURY AWD                1
```

Figure: Raw data from ‘SubModel’



```
AMERICAN             57099
'OTHER ASIAN'         6407
'TOP LINE ASIAN'      3526
OTHER                  179
```

Figure: Raw data from ‘Nationality’



```
AMERICAN             57099
OTHER ASIAN           6407
TOP LINE ASIAN        3526
OTHER                  179
```

Figure: Raw data from ‘Nationality’

After data cleaning for categorical data, there are 28 attributes and 67211 rows in the remaining dataset.

As for numerical data, the cleaned prices of the ‘MMR’ type features were found to be a positive shift from the average prices. With confidence from the results of correlation studies, the decision to drop all ‘MMR’ columns except the cleansed retail value of acquisition and current prices. Then, for the prospective value of the car, establish a new attribute, 'potential_value' = 'MMRCurrentRetailCleanPrice' - 'MMRAcquisitonRetailCleanPrice'.The magnitude of this discrepancy indicates the car's relative prospective value. Additionally, 'BYRNO' represents the car's registration number, which serves as an ID in this collection. It is useless for identifying whether the car has been "kicked," hence it should be removed from the dataset. The remaining dataset comprised 19 columns and 67211 rows after cleaning for numerical values.

3.2 Data Binning on Categorical Data

By reducing the number of unique values in a column, there can be more significant differences in values, rather than values being too spread out. While the default dimension is 10, the dimension for the data object should be in the range of 3 to 10. We do not need to manually bin categorical data because it is already in group form. However, some of the categorical data have more than ten dimensions. As a result, we can examine the relationship for those categorical data that have multiple dimensions with the target variable to see whether there is any association between them. Those categorical data with several dimensions such as 'Make,' 'Model,' 'Trim,' 'SubModel,' 'Color,' 'Size,' and 'VNST'. Since 'VNZIP1' has too many unique values, which is a similar case with continuous data. Thus, we will group them by using equal cut that same as other continuous data. Using the Pearson's Correlation Coefficient scale, compare these category data to the target variable, Class.

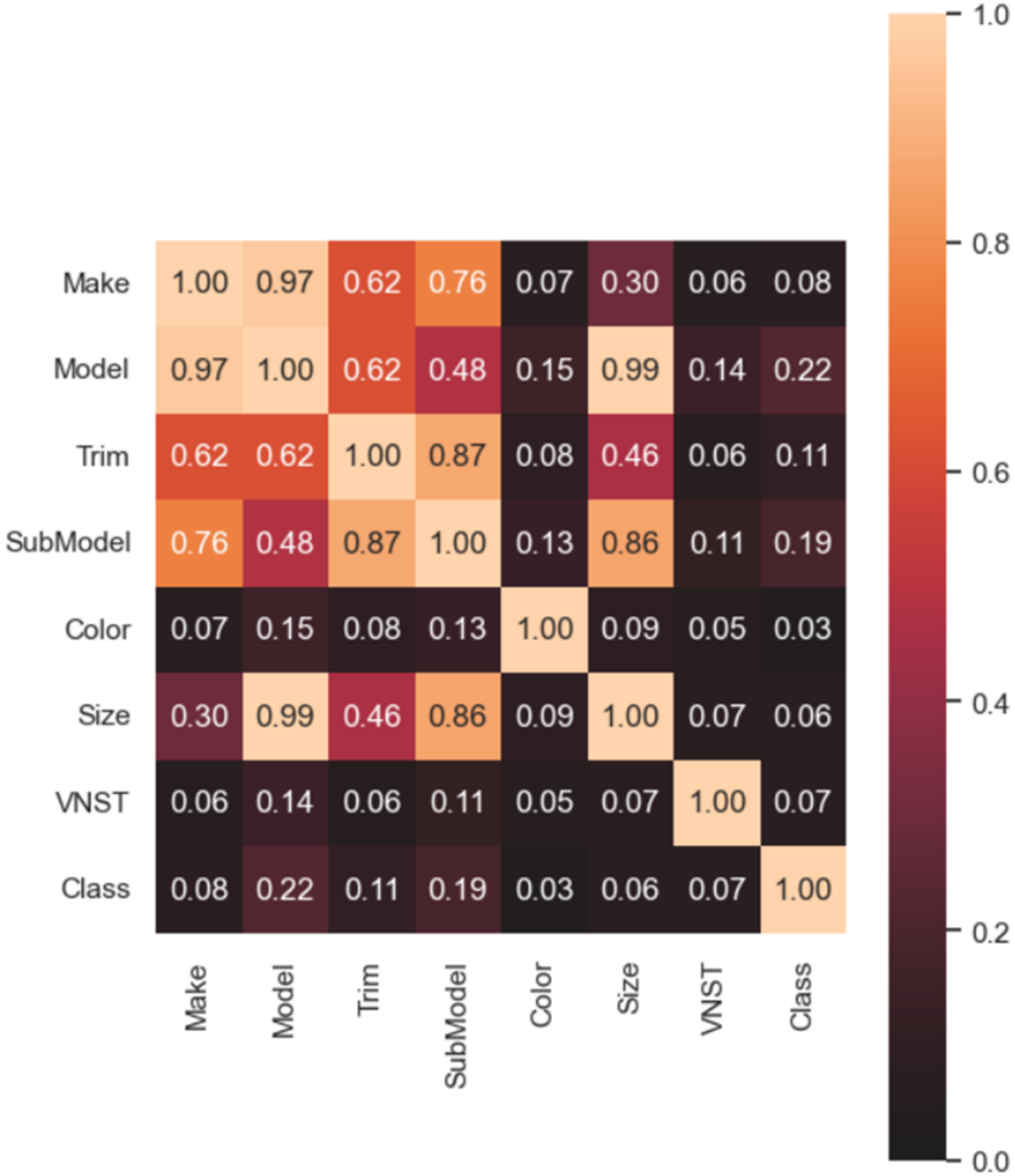
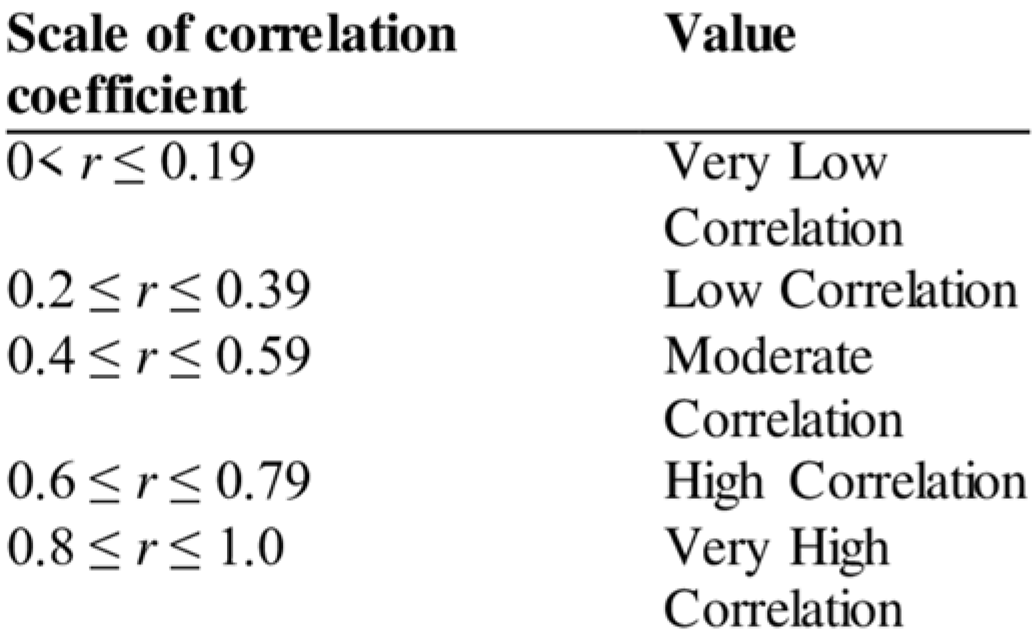


Figure: Visualization correlation coefficient for many dimension’s categorical data

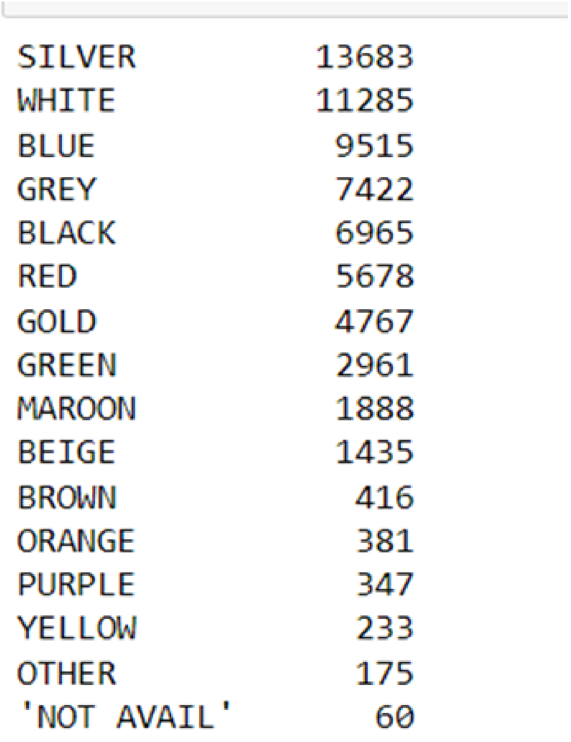


Scale of correlation coefficient	Value
$0< r \leq 0.19$	Very Low Correlation
$0.2 \leq r \leq 0.39$	Low Correlation
$0.4 \leq r \leq 0.59$	Moderate Correlation
$0.6 \leq r \leq 0.79$	High Correlation
$0.8 \leq r \leq 1.0$	Very High Correlation

Figure: The scale of Pearson's Correlation Coefficient

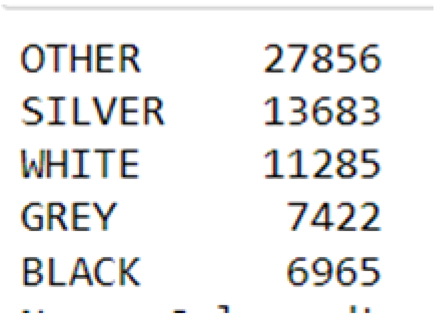
Data mining will exclude very highly correlated data since it will cause the model to overfit. The visualisation above shows that 'Model' and 'Make' are highly connected; consider dropping one of them. Since 'Model' is more connected to the target variable, 'Class,' the 'Make' variable can be removed from the dataset. Furthermore, because 'Model' and 'Size' are highly associated, one of them must be removed from the dataset. As 'Model' is better connected to the target variable, 'Class,' the 'Size' variable should be removed from the dataset. 'Trim' and 'SubModel' are highly correlated; because 'SubModel' is more associated to the target variable, 'Trim' should be removed from the dataset. As 'SubModel' and 'Size' are highly correlated, 'SubModel' will be retained in the dataset because it has a higher corelation coefficient with the target variable. After removing the very highly correlated categorical data, the dataset was reduced to 16 characteristics and 67211 rows.

'Model,' 'SubModel,' 'Color,' and 'VNST' are the left category data that contain several dimensions. In forecasting the target variable, there is no predictive function for 'VNST'. As a result, it has been removed from the dataset. According to global studies, neutral and simple colours such as black, white, silver, or grey guarantee an excellent resale value and easy maintaining them. As a result, the non-neutral colour in a group is classified as OTHER.



```
SILVER          13683
WHITE           11285
BLUE             9515
GREY             7422
BLACK            6965
RED              5678
GOLD             4767
GREEN            2961
MAROON           1888
BEIGE            1435
BROWN             416
ORANGE            381
PURPLE            347
YELLOW            233
OTHER             175
'NOT AVAIL'        60
```

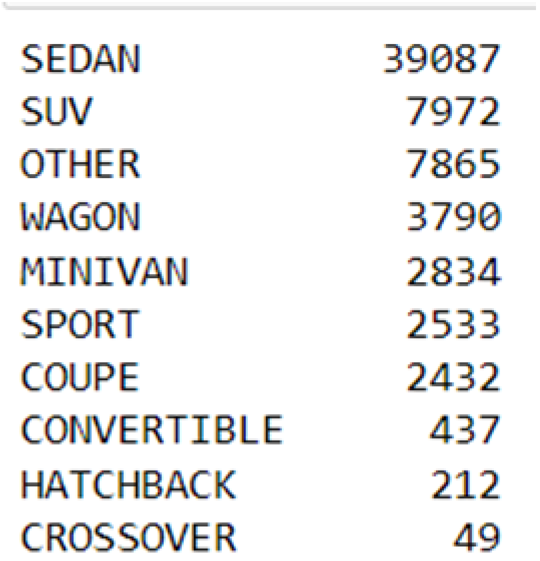
Figure: Raw ‘Color’



```
OTHER     27856
SILVER    13683
WHITE     11285
GREY       7422
BLACK      6965
```

Figure: After grouping non-neutral color into OTHER

Due to the fact that 'Model' and 'SubModel' have numerous dimensions, it is known from the statement that 'SubModel' is a subset of 'Model'. As a result, we intend to remove both of them from the dataset and replace them with a new attribute called 'CarTyp’,' which represents the type of car. According to research, the following car types are available: SUV, HATCHBACK, CROSSOVER, CONVERTIBLE, SEDAN, SPORT, COUPE, MINIVAN, WAGON, and others. The dimensions of the type of car are as follows:

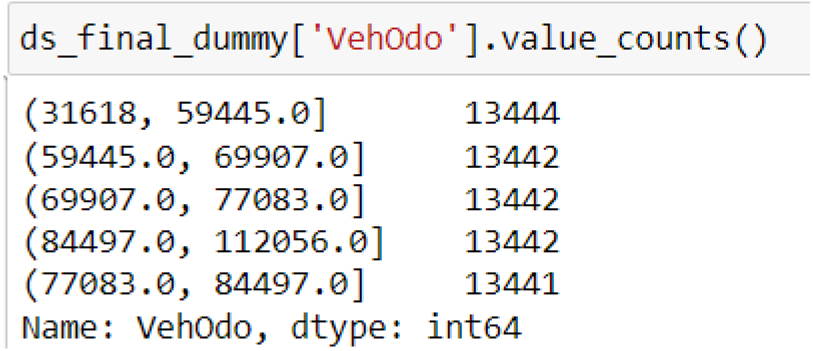


```
SEDAN           39087
SUV              7972
OTHER            7865
WAGON            3790
MINIVAN          2834
SPORT            2533
COUPE            2432
CONVERTIBLE       437
HATCHBACK         212
CROSSOVER          49
```

Figure: Dimension of ‘CarType’

3.3 Data Binning on Continuous Numerical Data

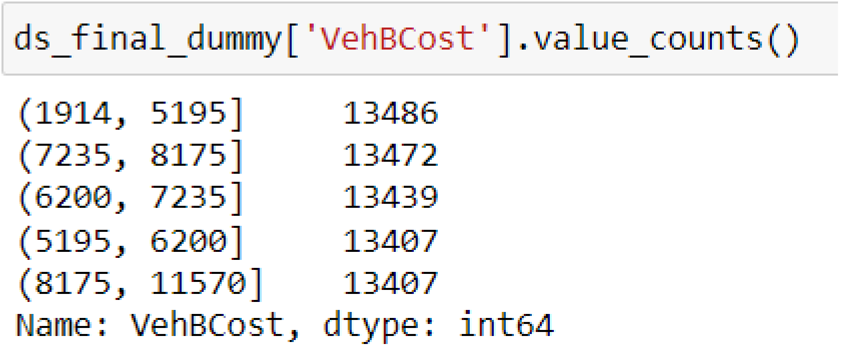
To decrease the dimensions of continuous data, an equal cut for 5 groups is employed here for all continuous data including 'VehOdo', 'VehBCost', 'WarrantyCost', 'VNZIP1', and 'Potential_Value'.



```
ds_final_dummy['VehOdo'].value_counts()
```

```
(31618, 59445.0]       13444
(59445.0, 69907.0]     13442
(69907.0, 77083.0]     13442
(84497.0, 112056.0]    13442
(77083.0, 84497.0]     13441
Name: VehOdo, dtype: int64
```

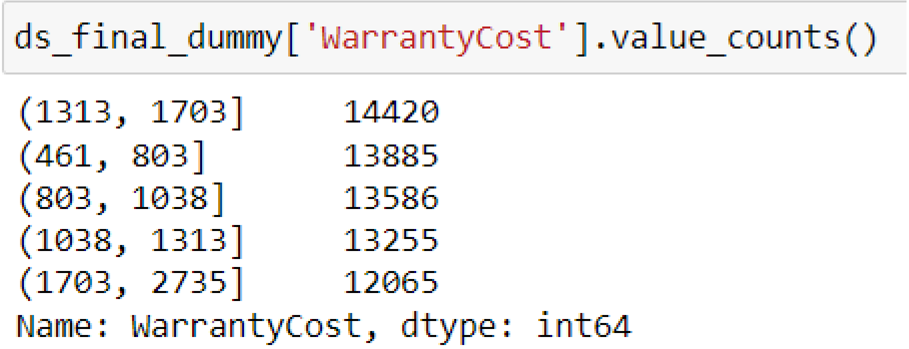
Figure: Equal cut and binning for ‘VehOdo’



```
ds_final_dummy['VehBCost'].value_counts()
```

```
(1914, 5195]     13486
(7235, 8175]     13472
(6200, 7235]     13439
(5195, 6200]     13407
(8175, 11570]    13407
Name: VehBCost, dtype: int64
```

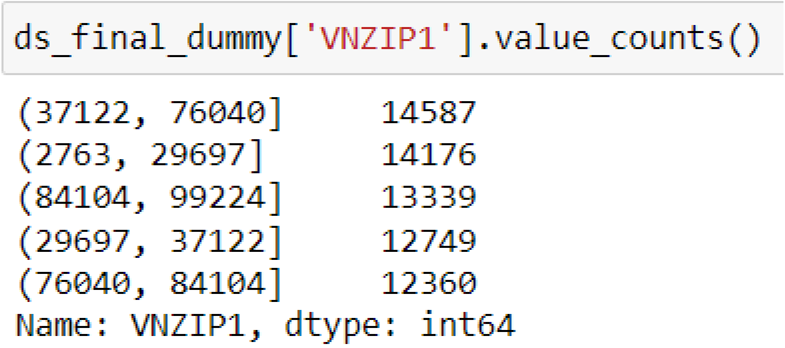
Figure: Equal cut and binning for ‘VehBCost’



```
ds_final_dummy['WarrantyCost'].value_counts()
```

```
(1313, 1703]     14420
(461, 803]       13885
(803, 1038]      13586
(1038, 1313]     13255
(1703, 2735]     12065
Name: WarrantyCost, dtype: int64
```

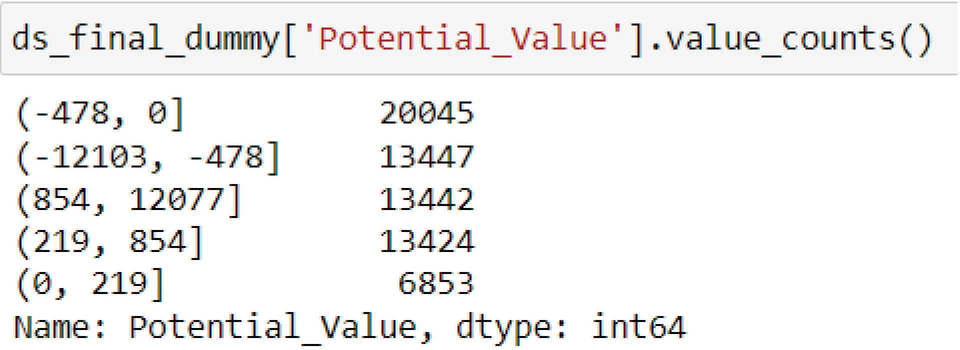
Figure: Equal cut and binning for ‘WarrantyCost’



```
ds_final_dummy['VNZIP1'].value_counts()
```

```
(37122, 76040]     14587
(2763, 29697]      14176
(84104, 99224]     13339
(29697, 37122]     12749
(76040, 84104]     12360
Name: VNZIP1, dtype: int64
```

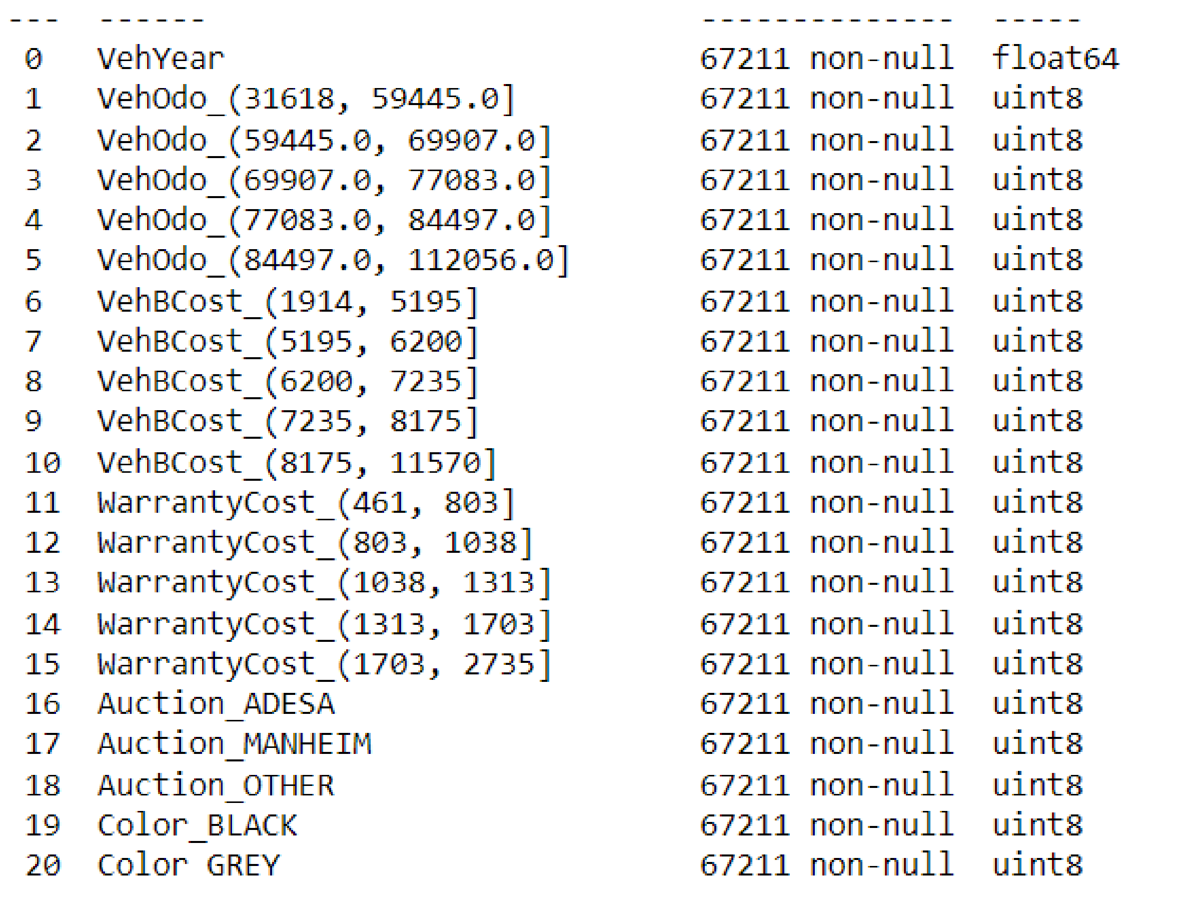
Figure: Equal cut and binning for ‘VNZIP1’



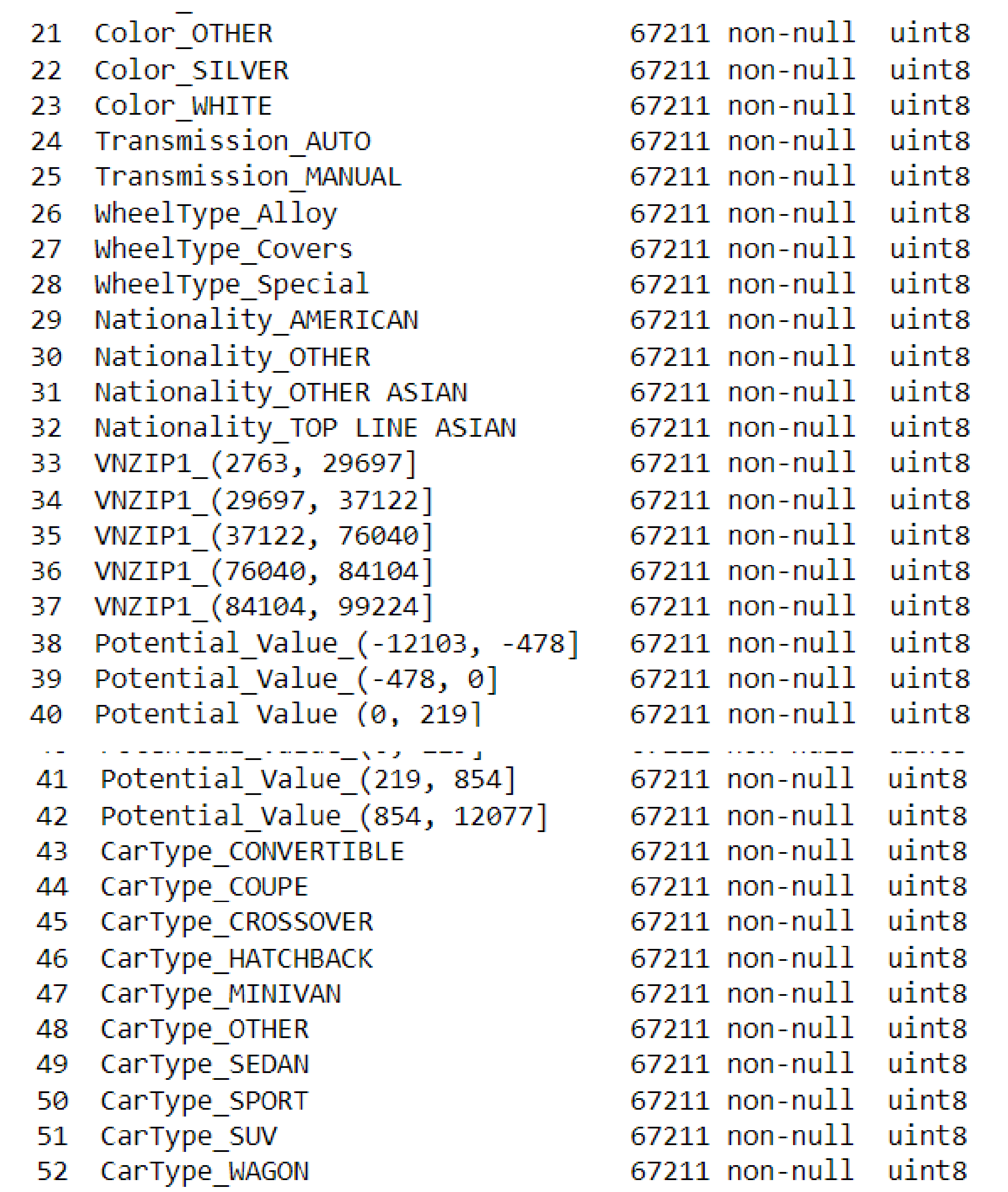
```
ds_final_dummy['Potential_Value'].value_counts()
```

```
(-478, 0]         20045
(-12103, -478]    13447
(854, 12077]      13442
(219, 854]        13424
(0, 219]           6853
Name: Potential_Value, dtype: int64
```

Figure: Equal cut and binning for ‘Potential_Value’



```
---  ------                          --------------  -----
 0   VehYear                         67211 non-null  float64
 1   VehOdo_(31618, 59445.0]         67211 non-null  uint8
 2   VehOdo_(59445.0, 69907.0]       67211 non-null  uint8
 3   VehOdo_(69907.0, 77083.0]       67211 non-null  uint8
 4   VehOdo_(77083.0, 84497.0]       67211 non-null  uint8
 5   VehOdo_(84497.0, 112056.0]      67211 non-null  uint8
 6   VehBCost_(1914, 5195]           67211 non-null  uint8
 7   VehBCost_(5195, 6200]           67211 non-null  uint8
 8   VehBCost_(6200, 7235]           67211 non-null  uint8
 9   VehBCost_(7235, 8175]           67211 non-null  uint8
 10  VehBCost_(8175, 11570]          67211 non-null  uint8
 11  WarrantyCost_(461, 803]         67211 non-null  uint8
 12  WarrantyCost_(803, 1038]        67211 non-null  uint8
 13  WarrantyCost_(1038, 1313]       67211 non-null  uint8
 14  WarrantyCost_(1313, 1703]       67211 non-null  uint8
 15  WarrantyCost_(1703, 2735]       67211 non-null  uint8
 16  Auction_ADESA                   67211 non-null  uint8
 17  Auction_MANHEIM                 67211 non-null  uint8
 18  Auction_OTHER                   67211 non-null  uint8
 19  Color_BLACK                     67211 non-null  uint8
 20  Color_GREY                      67211 non-null  uint8
```

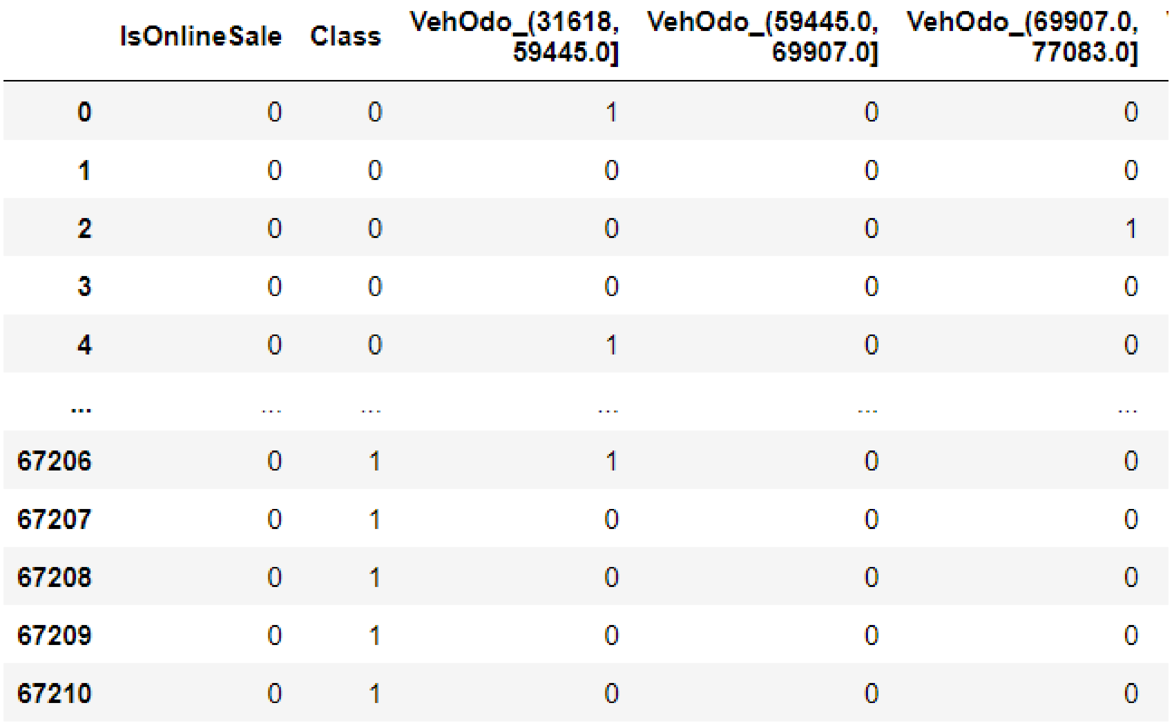


```
 21  Color_OTHER                        67211 non-null  uint8
 22  Color_SILVER                       67211 non-null  uint8
 23  Color_WHITE                        67211 non-null  uint8
 24  Transmission_AUTO                  67211 non-null  uint8
 25  Transmission_MANUAL                67211 non-null  uint8
 26  WheelType_Alloy                    67211 non-null  uint8
 27  WheelType_Covers                   67211 non-null  uint8
 28  WheelType_Special                  67211 non-null  uint8
 29  Nationality_AMERICAN               67211 non-null  uint8
 30  Nationality_OTHER                  67211 non-null  uint8
 31  Nationality_OTHER ASIAN            67211 non-null  uint8
 32  Nationality_TOP LINE ASIAN         67211 non-null  uint8
 33  VNZIP1_(2763, 29697]               67211 non-null  uint8
 34  VNZIP1_(29697, 37122]              67211 non-null  uint8
 35  VNZIP1_(37122, 76040]              67211 non-null  uint8
 36  VNZIP1_(76040, 84104]              67211 non-null  uint8
 37  VNZIP1_(84104, 99224]              67211 non-null  uint8
 38  Potential_Value_(-12103, -478]     67211 non-null  uint8
 39  Potential_Value_(-478, 0]          67211 non-null  uint8
 40  Potential Value (0, 219]           67211 non-null  uint8
 41  Potential_Value_(219, 854]         67211 non-null  uint8
 42  Potential_Value_(854, 12077]       67211 non-null  uint8
 43  CarType_CONVERTIBLE                67211 non-null  uint8
 44  CarType_COUPE                      67211 non-null  uint8
 45  CarType_CROSSOVER                  67211 non-null  uint8
 46  CarType_HATCHBACK                  67211 non-null  uint8
 47  CarType_MINIVAN                    67211 non-null  uint8
 48  CarType_OTHER                      67211 non-null  uint8
 49  CarType_SEDAN                      67211 non-null  uint8
 50  CarType_SPORT                      67211 non-null  uint8
 51  CarType_SUV                        67211 non-null  uint8
 52  CarType_WAGON                      67211 non-null  uint8
```

Figure: All variable including categorical and numerical data after binning

3.4 Feature Engineering

Converting our column data into dummy variables of 0s and 1s. Therefore, the dataset now have 53 columns and 67,211 rows.

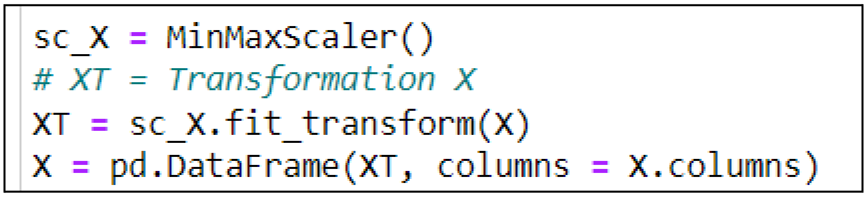


	IsOnlineSale	Class	VehOdo_(31618, 59445.0]	VehOdo_(59445.0, 69907.0]	VehOdo_(69907.0, 77083.0]
0	0	0	1	0	0
1	0	0	0	0	0
2	0	0	0	0	1
3	0	0	0	0	0
4	0	0	1	0	0
...	...	...	...	...	...
67206	0	1	1	0	0
67207	0	1	0	0	0
67208	0	1	0	0	0
67209	0	1	0	0	0
67210	0	1	0	0	0

Figure: After putting all variables into dummy variable

3.5 Feature Transformation / Feature Normalization

Transform data by using normalization using MinMaxScale.

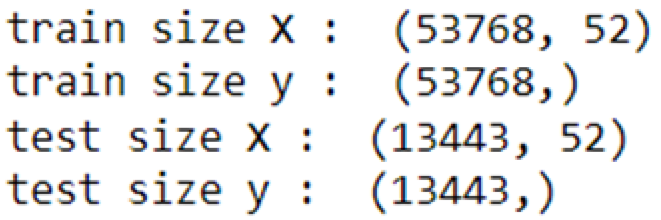


```
sc_X = MinMaxScaler()
# XT = Transformation X
XT = sc_X.fit_transform(X)
X = pd.DataFrame(XT, columns = X.columns)
```

Figure: Normalization process

3.6 Train-Test Splitting

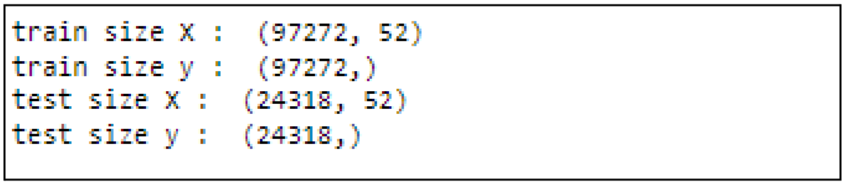
The data set is obtained through random sampling, then split randomly by 80% as a training set and 20% as a testing set. This is used as our preliminary train and test set.



```
train size X :  (53768, 52)
train size y :  (53768,)
test size X :  (13443, 52)
test size y :  (13443,)
```

Figure: Training and testing set (random sampling)

However, due to the imbalance in data, oversampling of the minority class and undersampling of the majority class is carried out to train and test the data mining models in an advanced stage.



```
train size X :  (97272, 52)
train size y :  (97272,)
test size X :  (24318, 52)
test size y :  (24318,)
```

Figure: Training and testing set (oversampling and undersampling)

4.0 Data Mining Models

To decide on which modeling technique would be most appropriate for our data set, the previously mentioned split of training and testing data was used to train and test 5 different data classification models which are: decision tree, neural network, linear and radial basis function (RBF) support vector machine (SVM) and k-Nearest Neighbors (kNN). For each model, a confusion matrix is used to display the results, which are then evaluated on accuracy, precision, recall and f1-score. The negative class represents “not kicked” data, while the positive class represents “kicked” data.

4.1 Preliminary Training and Testing

In the preliminary testing stage, the randomly sampled data with similar distribution to the original data is used to train and test the models.

4.1.1 Decision Tree

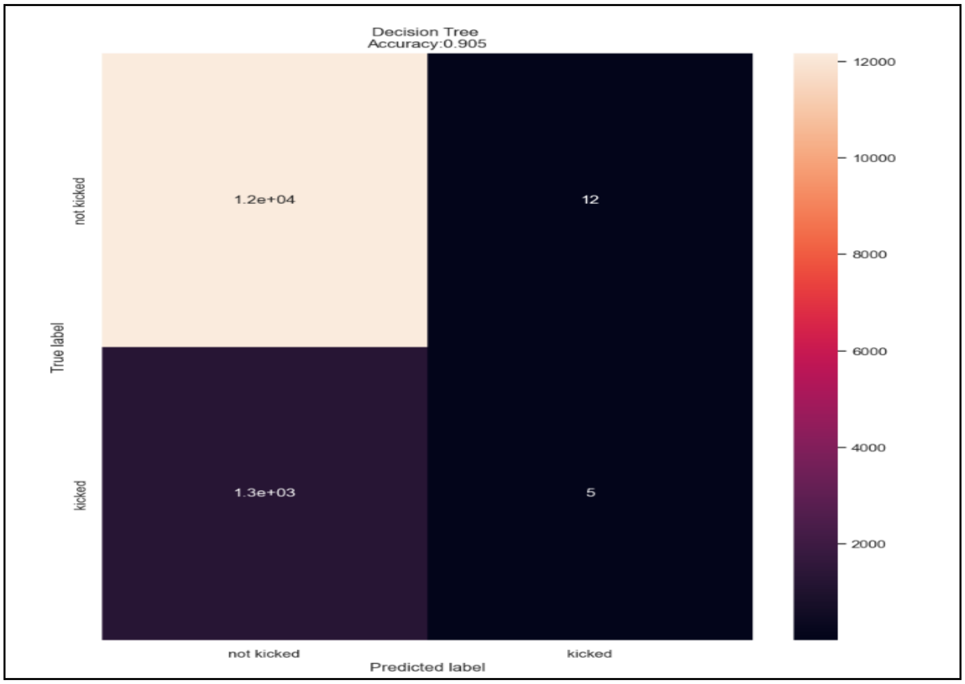
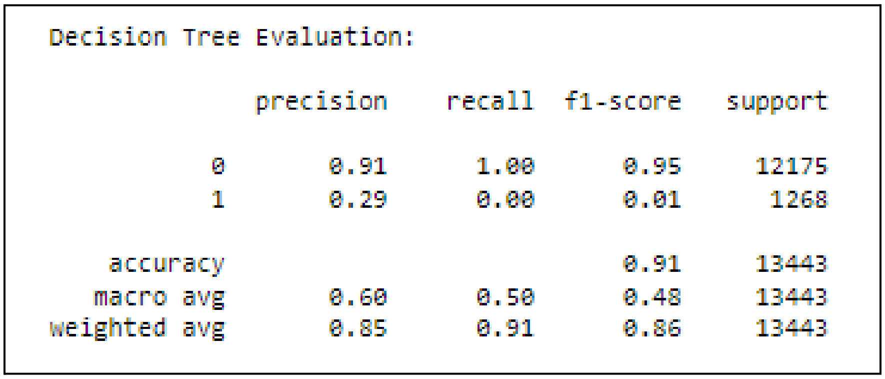


Figure: Confusion Matrix for Decision Tree

In this case, a decision tree of 5 levels of depth was used. The new confusion matrix showed that the decision tree model correctly classified 12163 out of 13443 observations in the "not kicked" class, while it only correctly classified 5 out of 1268 observations in the "kicked" class. However, the model misclassified 1263 observations from the "kicked" class as "not kicked" (false negatives), and 12 observations from the "not kicked" class as "kicked" (false positives).

The accuracy of the model can be calculated as (12163 + 5) / (12163 + 12 + 1263 + 5) = 0.905, or 90.5%. However, since the data is still heavily imbalanced towards the "not kicked" class, the accuracy metric may not be the best evaluation metric to use. Therefore, further analysis is required.

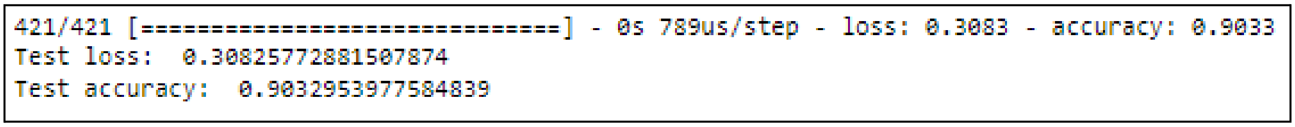


```
Decision Tree Evaluation:

              precision    recall  f1-score   support

           0       0.91      1.00      0.95     12175
           1       0.29      0.00      0.01      1268

    accuracy                           0.91     13443
   macro avg       0.60      0.50      0.48     13443
weighted avg       0.85      0.91      0.86     13443
```

Figure: Decision tree evaluation report

From this report, we can see that the model has a high precision (0.91) and high accuracy (0.91) for the negative class (0), indicating that when the model predicts an instance as not kicked, it is usually correct. Recall measures the proportion of true positive predictions out of all actual positive instances. However, the recall for the positive class (1) is very low (0.00), indicating that the model is not good at identifying the positive class, and that it is missing a lot of the actual positive instances. The macro-averaged F1-score is 0.48, indicating poor overall performance, while the weighted average F1-score is 0.86 only because it takes account of the class imbalance, giving more weight to the “not kicked” class.

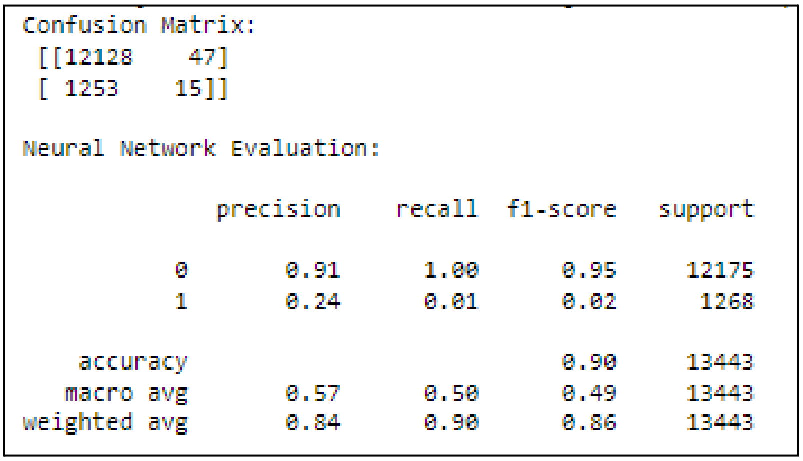
4.1.2 Neural Network



```
421/421 [==============================] - 0s 789us/step - loss: 0.3083 - accuracy: 0.9033
Test loss:  0.30825772881507874
Test accuracy:  0.9032953977584839
```

Figure: Test results obtained from 10 epochs

The training results of the neural network return a high accuracy of 0.903, which is 90.3%. This accuracy score is similar to the decision tree model, therefore further analysis is done to evaluate its performance.

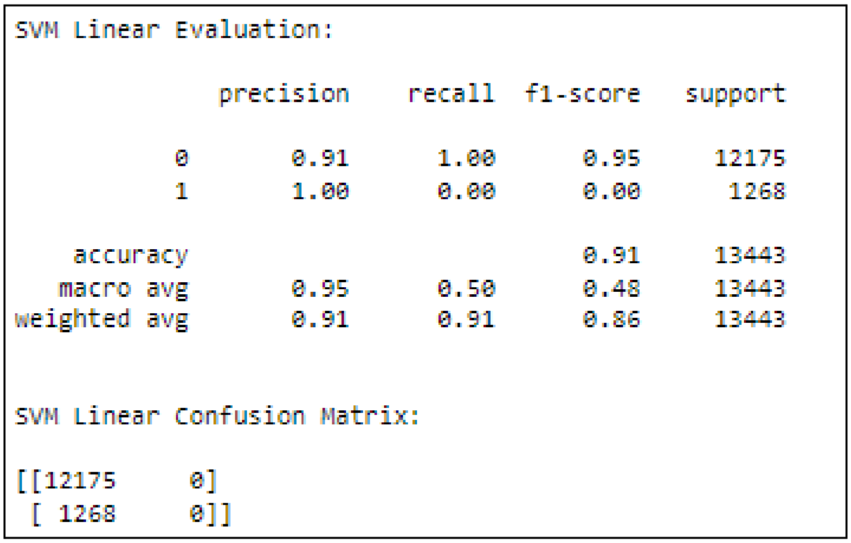


```
Confusion Matrix:
 [[12128    47]
 [ 1253    15]]

Neural Network Evaluation:

              precision    recall  f1-score   support

           0       0.91      1.00      0.95     12175
           1       0.24      0.01      0.02      1268

    accuracy                           0.90     13443
   macro avg       0.57      0.50      0.49     13443
weighted avg       0.84      0.90      0.86     13443
```

Figure: neural network evaluation report

The above figure shows that the neural network had similar performance to the decision tree, where the accuracy for determining negative classes is high, but is low for positive classes. In this case, the precision for the positive class is 0.24, meaning that only 24% of the predicted positive instances were actually positive. The recall for the positive class is 0.01, meaning that only 1% of the actual positive instances were correctly predicted as "kicked". The F1 score is a weighted average of precision and recall that combines the two measures into a single number. In this case, the F1 score for the positive class is very low at 0.02. Similar to the decision tree model, the macro average F1 score is 0.49, which is quite low, indicating that the model does not perform well, despite its high accuracy.

4.1.3 Linear SVM



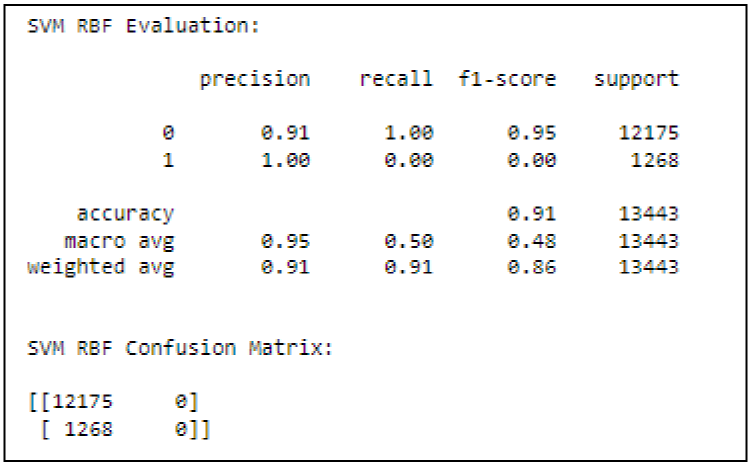
```
SVM Linear Evaluation:

              precision    recall  f1-score   support

           0       0.91      1.00      0.95     12175
           1       1.00      0.00      0.00      1268

    accuracy                           0.91     13443
   macro avg       0.95      0.50      0.48     13443
weighted avg       0.91      0.91      0.86     13443


SVM Linear Confusion Matrix:

[[12175     0]
 [ 1268     0]]
```

Figure: Linear SVM evaluation report

The linear SVM evaluation shows that the model is predicting “not kicked” classes with high precision and recall, but is predicting “kicked” classes with low precision and recall. This means that the model is correctly identifying examples that don't belong to the category, but is poor at identifying examples that do belong to the category.

The confusion matrix for the linear SVM evaluation confirms this: all examples belonging to the “kicked” class are being misclassified as “not kicked”. This proves that the linear SVM is showing similar patterns to the previous 2 models, where the model’s accuracy value is only influenced by the sheer imbalance of classes.

4.1.4 RBF SVM



```
SVM RBF Evaluation:

              precision    recall  f1-score   support

           0       0.91      1.00      0.95     12175
           1       1.00      0.00      0.00      1268

    accuracy                           0.91     13443
   macro avg       0.95      0.50      0.48     13443
weighted avg       0.91      0.91      0.86     13443


SVM RBF Confusion Matrix:

[[12175     0]
 [ 1268     0]]
```

Figure: RBF SVM evaluation report

The RBF SVM evaluation shows that the model is also predicting class 0 with high precision and recall, but is predicting class 1 with very low precision and recall. This means that the model is not doing a good job at identifying either class.

In summary, both models are doing well at predicting examples that belong to the “non kicked” class, but are unable to accurately predict data that belong to the “kicked” class. The RBF SVM model is doing even worse than the linear SVM model at predicting the positive class. However, from the similarities in inability to predict positive classes across all models, this difference between the RBF and linear SVM models is likely insignificant.

4.1.5 kNN Classification

As a preliminary stage, a range of K values between 10 to 31 were tested on 2 metrics, accuracy score and error rate.

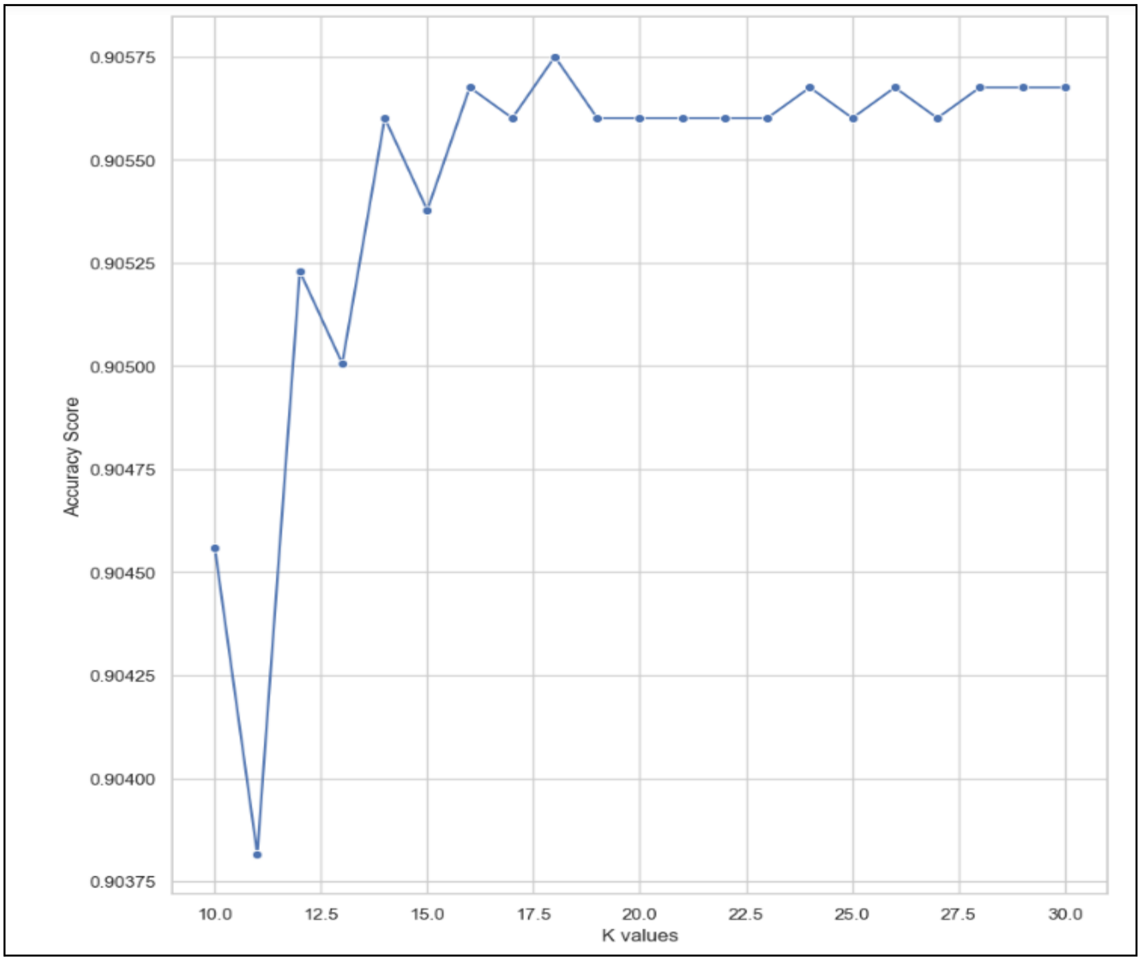


Figure: K values against accuracy score

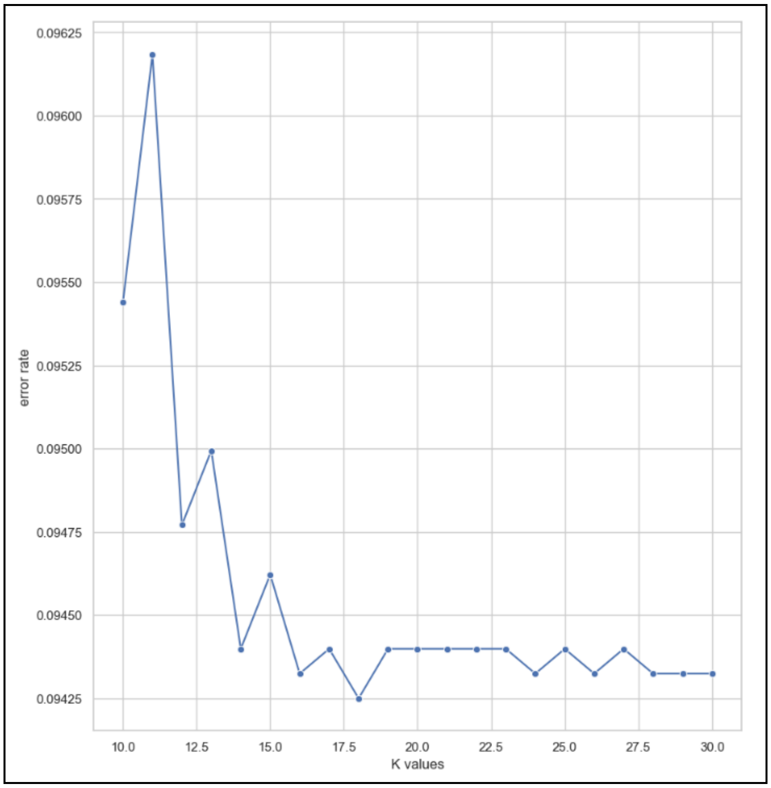
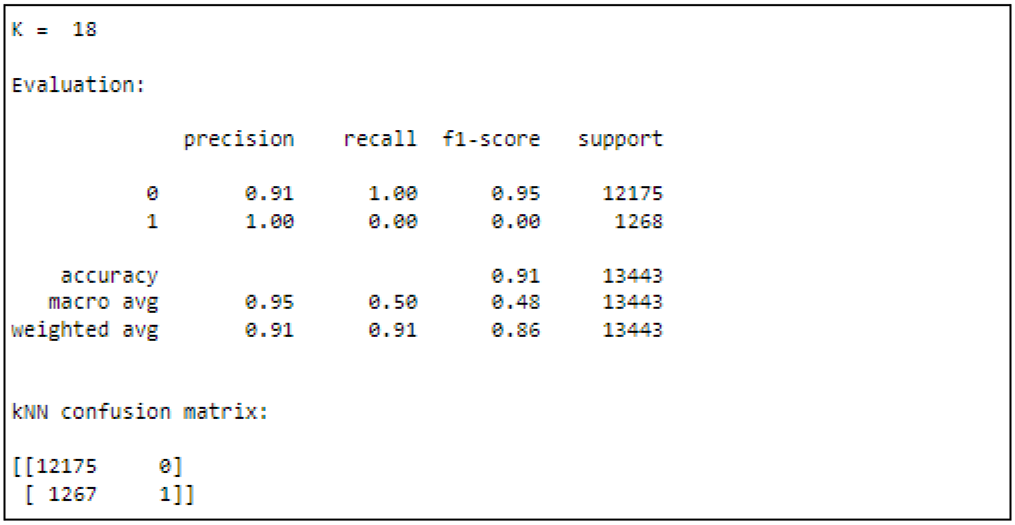


Figure: K values against error rate

From these results, we can conclude that the most fitting K value is 18, as it shows the highest accuracy score and lowest error rate. We then use the K value of 18 to train and test another kNN model.



```
K =  18

Evaluation:

              precision    recall  f1-score   support

           0       0.91      1.00      0.95     12175
           1       1.00      0.00      0.00      1268

    accuracy                           0.91     13443
   macro avg       0.95      0.50      0.48     13443
weighted avg       0.91      0.91      0.86     13443


kNN confusion matrix:

[[12175     0]
 [ 1267     1]]
```

Figure: KNN evaluation report

From this evaluation of the final kNN model, it can be seen that the same trends that occurred within the previous models still persist here, where the precision and recall for negative classes are high, and the F1-score accuracy of the model is high as well. Though the precision for the positive class is seen as high in this scenario, the recall and F1-score are extremely low, meaning that out of all the positive classes, only 1 was correctly identified. This implies that this model is also of poor performance.

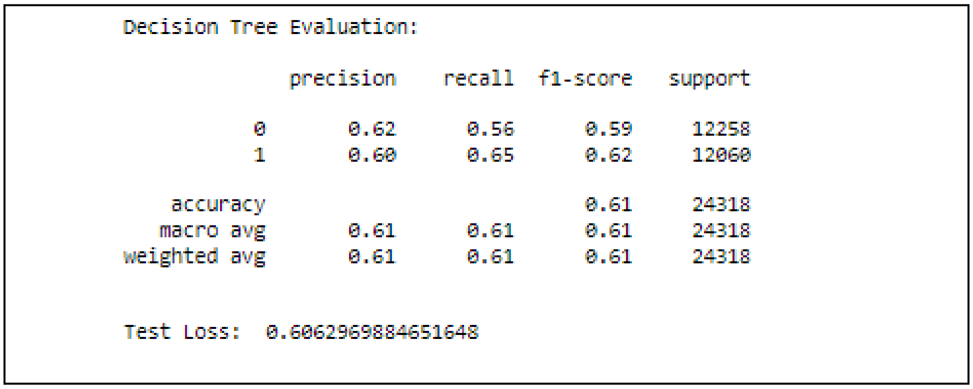
4.1.6 Overall Observations

Across the 5 models, similar accuracy scores were obtained, and all evaluations returned similar numerical results, implying that there is no major difference in the performance of each model. From these results, the hypothesis drawn is that the high accuracy scores of all models are due to the strong imbalance between the negative and positive classes, with the negative classes taking up 90% of the data set, leaving a small portion of positive class data. This is proven in the recall scores of the positive class being close to 0, which means that the model is unable to correctly identify any of the positive classes. Not only is this due to the large data imbalance, but it also ties back to what was hypothesized in the data exploration stage, where there was no recognizable pattern or trend to the data that belonged to the positive class, which is likely what is causing the model's failure to identify positive class data.

4.2 Advanced Training and Testing

Once we have tested the models with the preliminary data, we use a set of data that has been sampled to overcome the imbalance between the negative and positive classes, by oversampling the minority class and undersampling the majority class to make the distribution close to 50:50. In testing, it is found that both the linear and RBF SVM model's computational requirements are likely too high, as our devices are unable to obtain results from the SVM models. Thus, we have excluded it from this stage.

4.2.1 Decision Tree



```
Decision Tree Evaluation:

              precision    recall  f1-score   support

           0       0.62      0.56      0.59     12258
           1       0.60      0.65      0.62     12060

    accuracy                           0.61     24318
   macro avg       0.61      0.61      0.61     24318
weighted avg       0.61      0.61      0.61     24318


Test Loss:  0.6062969884651648
```

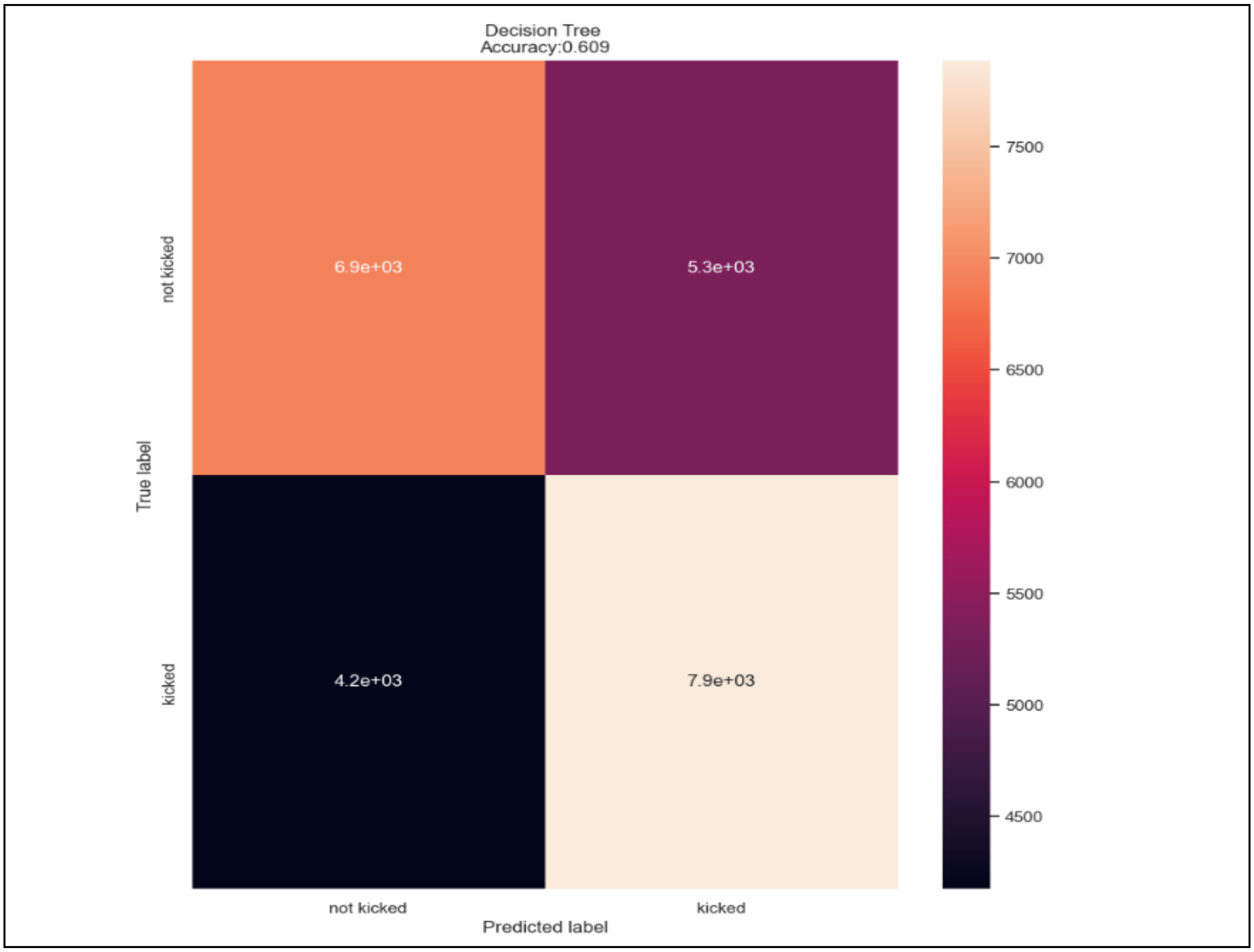
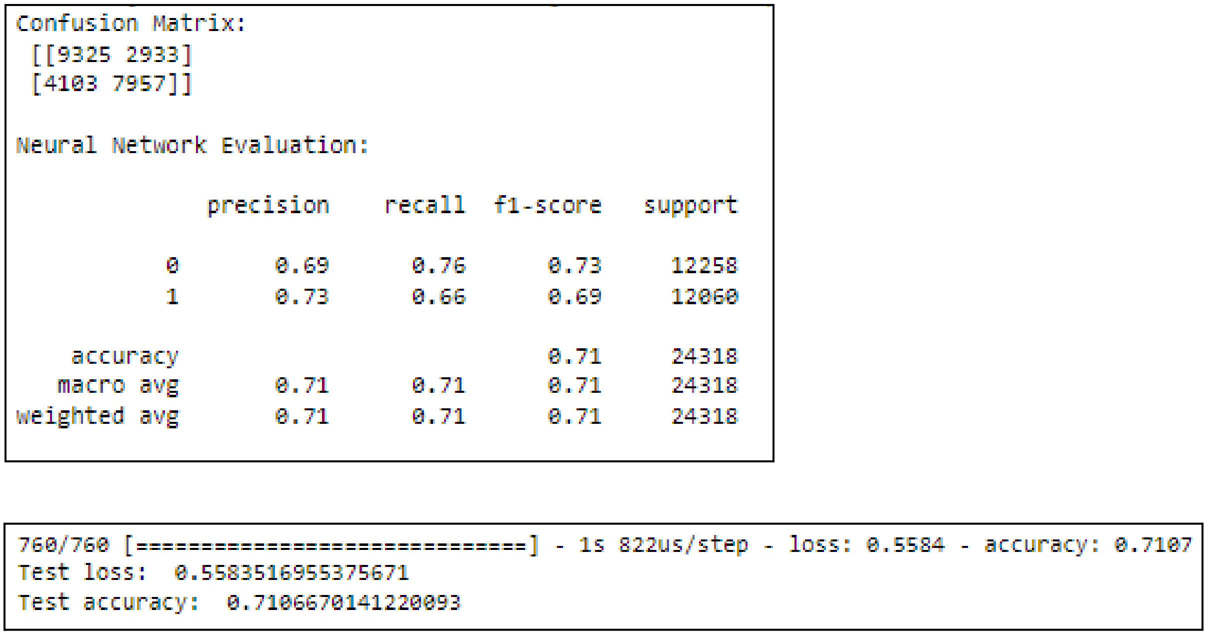


Figure: Decision tree confusion matrix

The evaluation report shows that the precision and recall scores for both classes improved slightly compared to the previous evaluation, but the overall accuracy is still only 60%. The macro average F1-score also improved slightly to 0.6, indicating that the model's performance is only slightly better than random chance, while the test loss is also very high at 60.6%. These results suggest that oversampling and undersampling alone may not be sufficient to improve the model's performance.

4.2.2 Neural Network



```
Confusion Matrix:
 [[9325 2933]
 [4103 7957]]

Neural Network Evaluation:

              precision    recall  f1-score   support

           0       0.69      0.76      0.73     12258
           1       0.73      0.66      0.69     12060

    accuracy                           0.71     24318
   macro avg       0.71      0.71      0.71     24318
weighted avg       0.71      0.71      0.71     24318
```

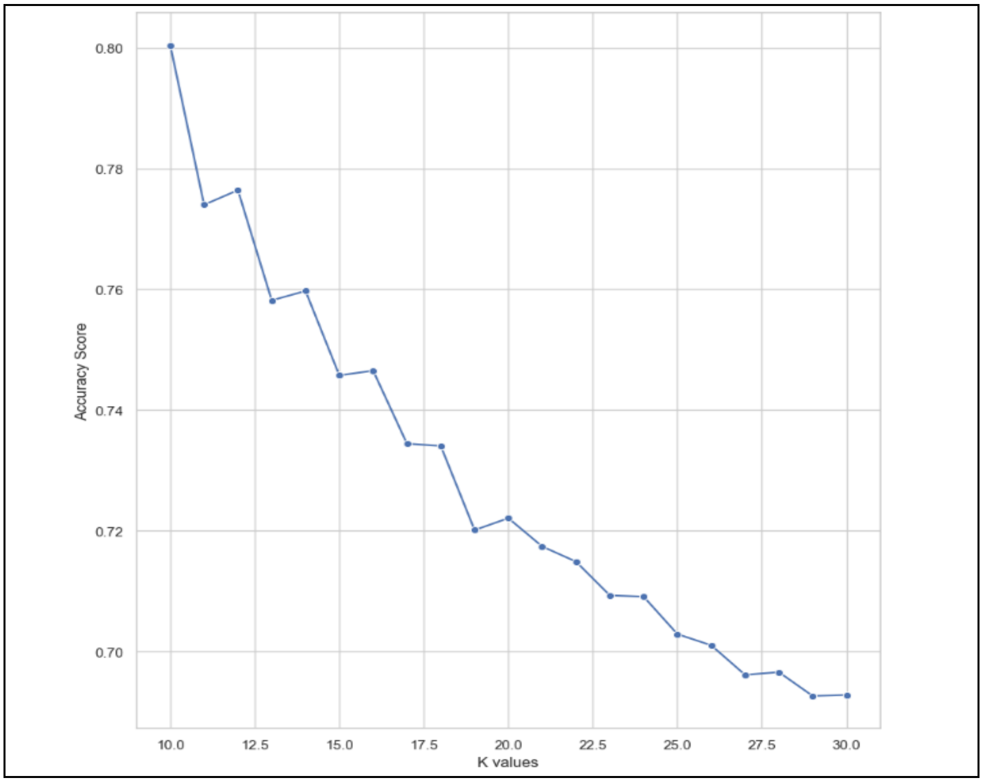
```
760/760 [==============================] - 1s 822us/step - loss: 0.5584 - accuracy: 0.7107
Test loss:  0.5583516955375671
Test accuracy:  0.7106670141220093
```

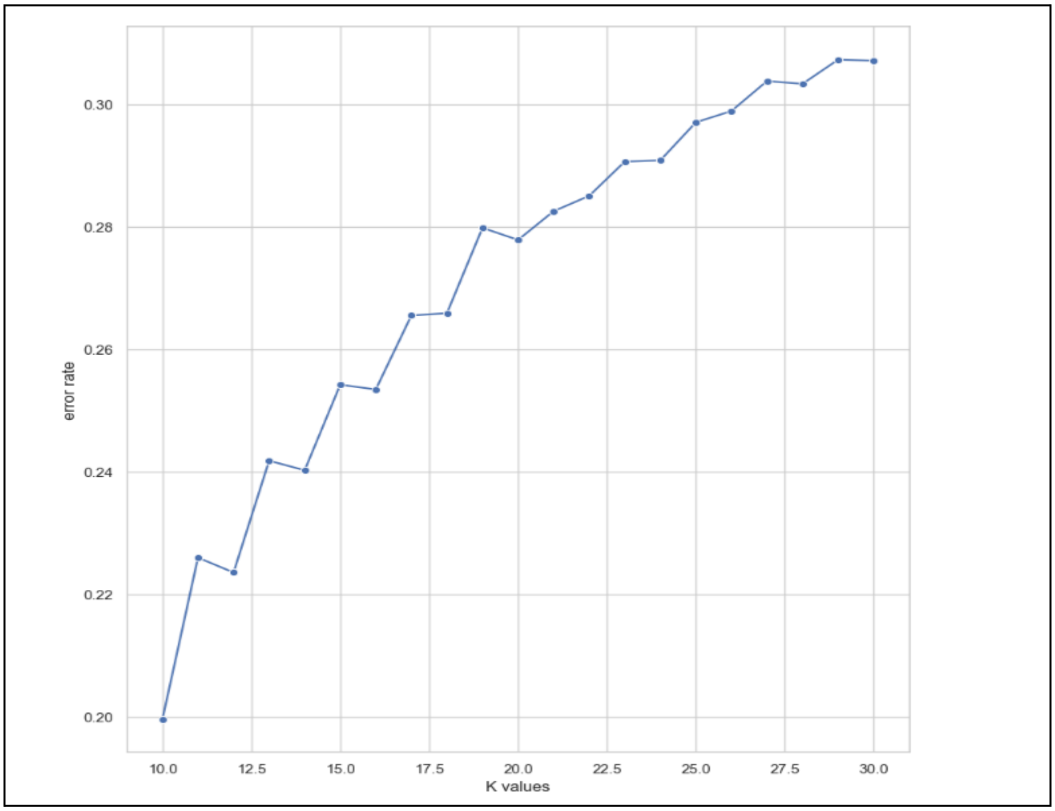
Figure: Neural network performance after 20 epochs

The neural network’s performance also underwent a slight improvement in performance, slightly overtaking the decision tree. However, although the accuracy, precision, recall and F1-score have improved, the test loss is high, at 55.8%. This indicates that the model’s predictions are incorrect 55.8% of the time, which is not a desirable result.

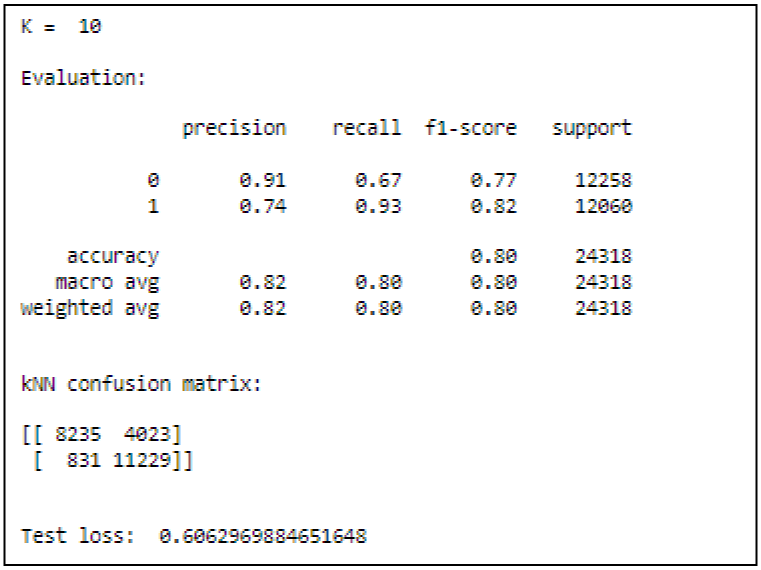
4.2.3 kNN Classification

Similarly to the preliminary stage, the data set is first used to determine the best K-value, based on accuracy score and error rate.





From the graphs, it is clear that a K-value of 10 is the most suitable.



```
K =  10

Evaluation:

              precision    recall  f1-score   support

           0       0.91      0.67      0.77     12258
           1       0.74      0.93      0.82     12060

    accuracy                           0.80     24318
   macro avg       0.82      0.80      0.80     24318
weighted avg       0.82      0.80      0.80     24318


kNN confusion matrix:

[[ 8235  4023]
 [  831 11229]]


Test loss:  0.6062969884651648
```

The kNN model shows the best performance out of the 3 models after training and testing with the balanced data set, showing the highest accuracy so far at 80%, and significantly improved values to precision, recall, and F1-score. The confusion matrix shows that the model has correctly classified 11,229 instances belonging to the minority class (class 1), but has incorrectly classified 831 instances belonging to the majority class (class 0) as belonging to the minority class, which is a stark contrast to the results obtained from the preliminary stage. However, the test loss is found to be even higher than the neural network, at 60.6%.

4.3 Model Summary

Overall, there are significant improvements to the performance of the models after being tested with balanced datasets compared to when tested with randomly sampled datasets. Based on accuracy, precision, recall and F1-score, the kNN classification model is by far the best performer, followed by the neural network and the decision tree. However, the test losses in all 3 of the models used in the advanced modeling stage are high, indicating that the models are only performing well on identifying 1 of the classes, further confirming the hypothesis that there is a lack of trends that determine the class of a tuple.

5.0 Research Questions

5.1 Research Questions and Justifications

Auto dealerships face a challenge when purchasing a used car at an auto auction. The dealership must determine whether or not the vehicle has serious flaws and prevent it from being sold to customers. The primary goal of this dataset is to predict whether or not a car purchased at an auto auction is a "kick" (car with serious problems). In dealing with this dataset, some research questions come to mind:

1. Do older cars with lower odometer readings show higher chances of being kicked?

   This question is posed as we would like to determine if tampered odometers is a determining factor of kicked cars. By having supporting evidence to this, we can then determine the range of values of odometer reading - age values that would cause the likelihood of a kick. However, from analyzing our dataset, there was no relationship supporting this hypothesis, as shown in section 2.1, as the outcome of kicks increase with age regardless of odometer values. Thus, the answer to this question is no, older cars with lower odometer readings do not show higher chances of being kicked.

2. Are cars bought through an online sale susceptible to illegitimate dealings and thus make the car a kick?

   This question is important in determining whether a dealership should even consider purchasing a car through an online sale. Purchasing a car through an online sale could be potentially costly should those cars be advertised falsely, with little room for verification of authenticity due to the virtual nature of this dealing. If it is so, then the kicked car percentage on cars bought through an online sale should be high. This was however, not the case, as it was found that 8% and 9.6% of cars were classified as kicked cars from cars not purchased from an online sale and cars purchased from an online sale respectively. Since the proportion of kicked cars between the two categories are not significantly different, we cannot conclude that cars bought through an online sale have a significantly higher chance of being a kick.

3. Is there a trend to the color of kicked cars?

   Something as simple as a car’s body color could have a significant impact on the purchase value of a car. We want to know what the ‘safe’ colors are in purchasing a car, so that the chances of buying a kicked car is avoided. From our analysis in section 2.1, it was found that colors of more neutral, low-key colors comprised of the top 3 non-kicked proportion of cars. Namely, silver, white and blue are considered ‘safe’ colors. Conversely, more jarring colors such as purple, orange, and brown had higher proportions of kicks. Thus, those colors should generally be avoided to prevent purchasing a kicked car.

5.4 Provide meaningful insight and conclusion from the data analysis

The dealership must decide whether or not the vehicle has significant problems and prevent it from being sold to customers. These bad purchases are known as "kicks" in the auto industry. Knowing that the historical dataset has a high probability of not being kicked out of Data Exploration which means that the most of the cars purchased were considered “best buy”.

Under categorical variable, the relevant columns of this subset include 'Auction', 'Model', 'Trim', 'Submodel', 'Colour', 'Transmission', 'WheelType' and 'WheelTypeID', 'Nationality', 'Size', 'Make', 'TopThreeAmericanName', 'VNST', 'IsOnlineSale' and 'VNZIP1'. After data cleaning, there are only 8 columns consider significant in predicting whether the car purchased is "best buy" which include 'VehYear', 'Auction', 'Color', 'Transmission', 'WheelType', 'Nationality' , 'IsOnlineSale' and 'CarType'.

- For 'Auction', MANEIM has the highest car purchase compared to ADESA and OTHER.
- Due to the fact that 'Model' and 'SubModel' have numerous dimensions, it is known from the statement that 'SubModel' is a subset of 'Model'. As a result, we intend to remove both of them from the dataset and replace them with a new attribute called 'CarType', which represents the type of car. According to research, the following car types are available: SUV, HATCHBACK, CROSSOVER, CONVERTIBLE, SEDAN, SPORT, COUPE,

MINIVAN, WAGON, and others. Finding that SEDAN type was the highest car purchased and being sold by an auto dealer.

- For 'Colour' from data exploration, knowing that the top 3 colors of cars purchased were silver, white and blue. However, according to global studies, neutral and simple colors such as black, white, silver, or grey guarantee an excellent resale value and easy maintenance. Hence, in the Data Cleaning part, the non-neutral color in a group is classified as OTHER.
- For 'Transmission', most of the cars are AUTO which has about 97% of cars purchased were auto.
- ‘WheelType’ and ‘WheelTypeID’ were proven to be referring to the same attribute, as their corresponding frequencies for the different column values are the same which shows that the highest car purchased is using Alloy type wheel.
- For 'Nationality', it can be seen that 85% of cars purchased are of American nationality, while the remaining are either Asian or some other nationality.
- Only a small amount of 2.5% cars were purchased through an online sale.

Under numerical variables, the relevant columns of this subset include ‘PurchDate’, ‘VehYear’, ‘VehicleAge’, ‘VehOdo’, ‘MMRAcquisitionAuctionAveragePrice’, ‘MMRAcquisitionAuctionCleanPrice’, ‘MMRAcquisitionRetailAveragePrice’, ‘MMRAcquisitonRetailCleanPrice’, ‘MMRCurrentAuctionAveragePrice’, ‘MMRCurrentAuctionCleanPrice’, ‘MMRCurrentRetailAveragePrice’, ‘MMRCurrentRetailCleanPrice’, ‘VehBCost’, and ‘WarrantyCost’. After Data Exploration and Data Cleaning, there are 5 attributes left which are 'VehOdo', 'VehBCost', 'WarrantyCost', 'VNZIP1' and 'Potential_Value'.

- For 'VehOdo', which indicates how far the car has driven in km, according to Data Exploration, the larger the vehicle age, the larger the 'VehOdo'.
- For the prospective value of the car, establish a new attribute, 'potential_value' = 'MMRCurrentRetailCleanPrice' - 'MMRAcquisitonRetailCleanPrice'.The magnitude of this discrepancy indicates the car's relative prospective value.

- Vehicle Buying price was investigated and results show that the buying price center around 7000 per car, for which currency was not documented so it is unknown, but most likely USD.
- Warranty cost, on the other hand, is right skewed, with most common cost price between 1000 - 2000, about 20% of the vehicle buying price.

Deliverables

Your submission should include the following:

- A Jupyter notebook (or equivalent) with your code, comments, and visualizations.
- A report based on the outlined tasks, summarizing your findings, including a discussion of your approach, key insights, and any limitations or areas for further exploration.
- A brief video presentation (max. 5 minutes) that explains your approach and summarizes your findings.

Marking Rubric

The assignment will be graded out of 120 points; 40 marks for each CLO. The final score will be 30%, which is 10% for each CLO. The marks are based on the following criteria, as outlined in the tasks:

CLO	Criteria	Poor	Satisfactory	Excellent	Marks
2	Data Understanding: **(20 marks)** • Identifies the dataset and describes its source (5m) • Summarizes the variables in the dataset (5m) • Identifies potential issues or limitations with the dataset (5m) • Demonstrates understanding of the dataset and its context (5m)				
	Data Description **(20 marks)** • Provides descriptive statistics and summaries of the dataset (5m) • Identifies key features, patterns, and trends in the data (5m) • Uses appropriate statistical and visual methods to summarize the data (5m) • Demonstrates understanding of the data and its properties (5m)				
3	Data Preprocessing **(20 marks)** • Performs data cleaning, transformation, and normalization (5m) • Selects and justifies data sampling techniques (5m) • Removes any irrelevant data or outliers (5m) • Demonstrates understanding of data preprocessing techniques (5m)				

	Visualization **(20 marks)** • Creates effective and informative visualizations to summarize the data (5m) • Uses appropriate chart types and labeling to convey information (5m) • Provides clear and accurate data labels, titles, and legends (5m) • Demonstrates understanding of data visualization techniques (5m)				
4	Research Questions **(20 marks)** • Identifies and justifies research questions to address using the dataset (5m) • Develops and applies appropriate data mining methods to answer research questions (5m) • Evaluates the data mining model to ensure that the model is accurate and reliable (5m) • Provides meaningful insights and conclusions from the data analysis (5m)				
	Presentation **(20 marks)** • Presents the analysis in a clear, organized, and engaging manner (5m) • Uses appropriate language, tone, and style for the intended audience (5m) • Follows appropriate formatting and structure for the presentation (5m) • Demonstrates good communication skills (5m)				

Note: Poor 0-2, Satisfactory 3, Excellent 4-5

Submission Guidelines

Submit your assignment as a compressed file (zip or tar) that includes the Jupyter notebook, report, and video presentation. Name the file using your group number (e.g., Group1.zip). Submit your assignment through the course's WBLE. The deadline for submission 18 April 2023. Late submissions will not be accepted.