Carvana, Don’t Get Kicked

Muhammad Umair Naeem

This project aims to create a predictive model for the purchase of a different cars. Thanks to this model the buyer will reduce the possibility to buy a bad car that will be difficult to sell in the market.

The dataset consisted of all the data of cars that have been sold between 2000 and 2010 and all of the data have been provided by the American company Carvana that has published this competition on Kaggle.

All the car in the dataset have different characteristics, foreach auto we have the type, which kind of wheel the vehicle has, the mileage of the car, the color, the price of the purchase and other information useful or not which will be used for create a good predictive model.

Our analysis is divided in four different parts: Data Understanding, Clustering, Associations rules, Classifications.

- **Data Understanding**: In the first part, we have analyzed the training set, the data quality, data semantics, missing values and outliers.

- **Clustering**: In this part, we have analyzed any redundancies between the attributes and some relations between that.

- **Association Rules**: This task is about finding frequent patterns and association rules inside the dataset for a better prediction of the class.

- **Classification**: In this final task, we have built different prediction models using different algorithms and we choose the best between these different models and made some comments about our decision.

Data Understanding

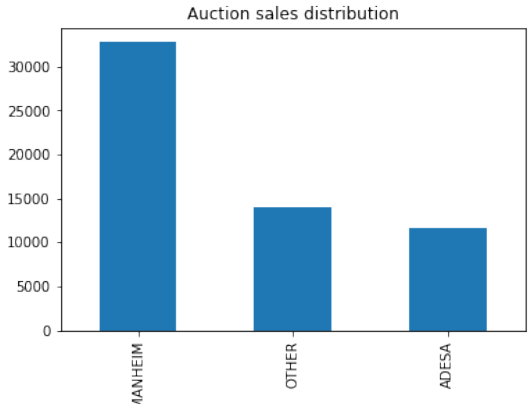
In this part, we will assess all the attributes of the dataset. We will evaluate the distribution and the relevance of all the attributes with respect to the goal of our project and the correlation between them.
During our analysis, we encounter some attributes that have missing values and outliers, so, we used different approaches to manage these missing values, for instance, replacing the missing values with the average value of the attributes, eliminating some missing values where the distribution of the value don't change and eliminating the outliers.
We have also eliminated some attributes which are irrelevant to our goal and we transformed some attributes for a better data preparation.

- **RefId:** is an integer attribute that show the unique identification number assigned to every vehicle. It is unrelated to our goal, because it doesn't say anything about the good qualities of the purchase, so we have eliminated it.

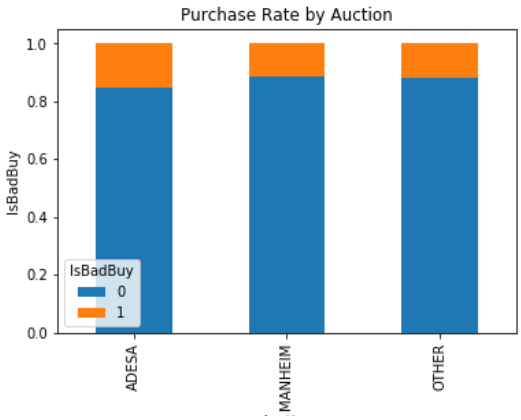
- **IsBadBuy:** is a discrete attribute which takes binary values (0 or 1): 0 is a good purchase, 1 is a bad purchase.

- **PurchDate:** is an object data which show the auction purchase date of the vehicle. There are no missing values for this attribute.

- **Auction:** is an object attribute which have three values, ADHESA, MANHEIM and OTHER. We decide to transform them into categorical data, 1, 2, 3. In the graphs we can see the different distribution of the variables of Auction and the correlation between the different values with the attribute IsBadBuy.

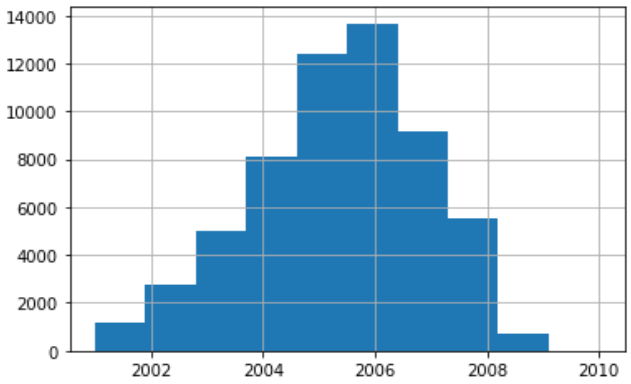


Graph 1- Auction Distribution



Graph 2- Purchase Rate by Auction

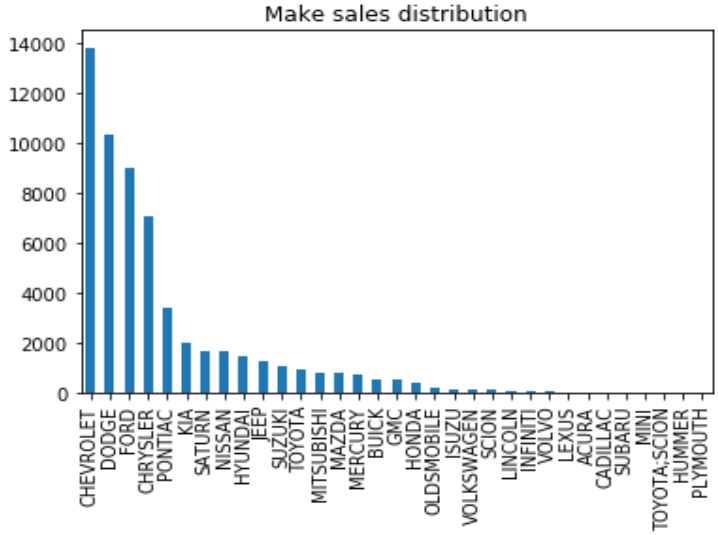
- **VehYear:** is an integer attribute that explain when the vehicle was manufactured. In graph 3 we can see the different distribution of the value.



Graph 3 – Number of vehicles by year

- **VehicleAge:** is an integer attribute that provides the actual age of vehicle, we eliminated VehicleAge it can be obtained by evaluating the difference between PurchDate and VehYear, so it is redundant.

- **Make:** is an object attribute which shows the vehicle brand.



Graph 4 – Vehicle brand distribution

- **Model:** is an object attribute and it is correlated with the Make.

- **Trim:** is an object attribute that show the additional features that one can choose to add in his vehicle. There are 1911 missing values of this attribute. So, we decided to replace the missing values with an undefined value, "?", The reason behind this decision is that we do not want to eliminate the missing values as they are missing in a random order and even if we try to eliminate these missing value it will force us to eliminate the entire row and the entire row is consisted on the data of other attributes.

- **SubModel:** is an object attribute. It provides more precise information about the vehicle and it is correlated with Make and Model. This attribute has 7 missing values and we decided to eliminate the missing values because the attributes Color, Transmission, WheelTypeID, WheelType and Trim are in the same row and have missing values.

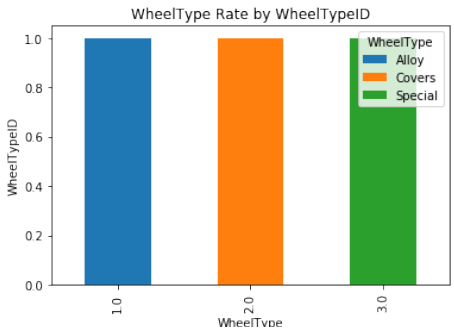
- **Color:** is an object attribute. This attribute has 7 missing values similarly to other attributes explained previously in the section of SubModel.

- **Transmission:** is an object attribute which have, other than 7 missing values explained in the previous sections, only 1 missing value. We have decided to eliminate this 1 missing value as it would not affect the quality of our data. Transmission have only two types of data which is MANUAL and AUTO so we decide to transform them into categorical data, 1 and 2.

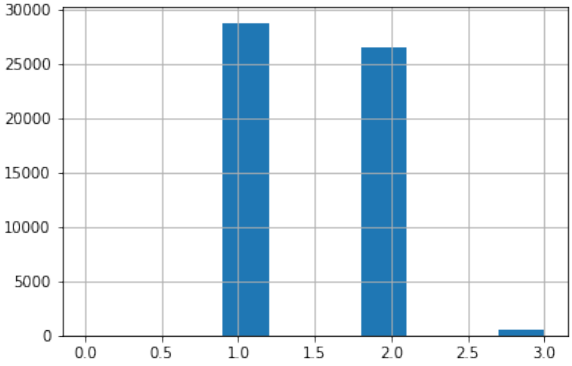
- **WheelTypeID:** is an integer attribute while **WheelType** is an object attribute, they are homogeneous attributes. So, we are keeping the former as it has categorical data (1, 2, 3) and it is easier to manage such data and eliminating the latter. There are 2574 missing values of WheelTypeID which we decided to ignore and replace with an out of range value -10. We took this decision because the missing values are missing randomly, and it would not be wise to eliminate the entire row which consists on the data of the other attributes.

WheelType WheelTypeID	Alloy	Covers	Special
1.0	1.0	0.0	0.0
2.0	0.0	1.0	0.0
3.0	0.0	0.0	1.0

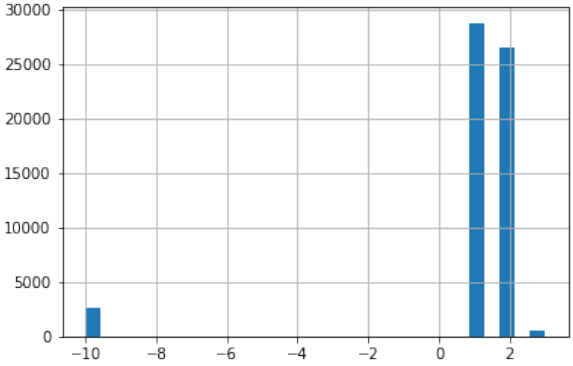
Graph 5 – Correlation between WheelTypeID and WheelType attributes



Graph 6 – WheelType rate by WheelTypeID

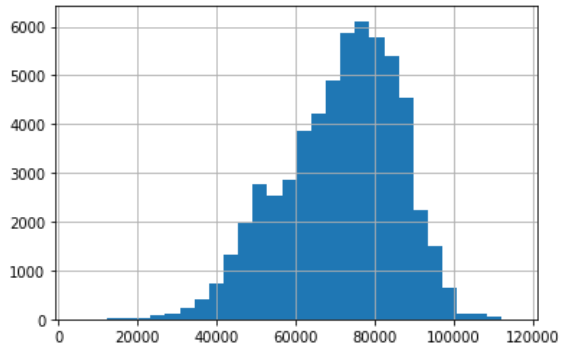


Graph 7 – WheelTypeID initial distribution

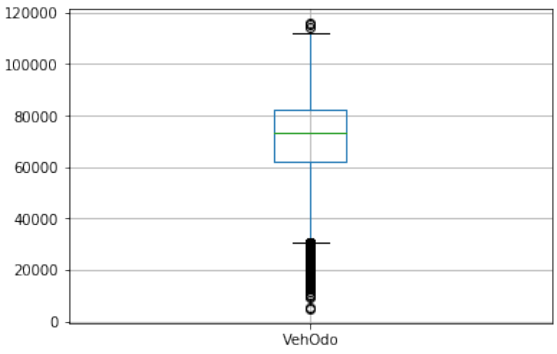


Graph 8 – WheelTypeID distribution after changes

- **VehOdo:** is an integer attribute which provides the odometer reading. We have eliminated all the values which are greater than 11600 and less than 3000, because these values are outliers.

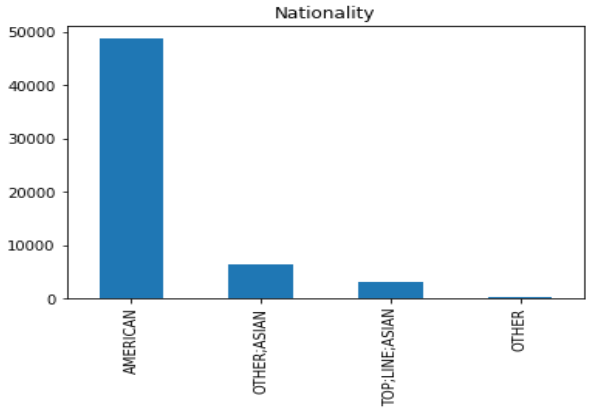


Graph 9 – VehOdo distribution



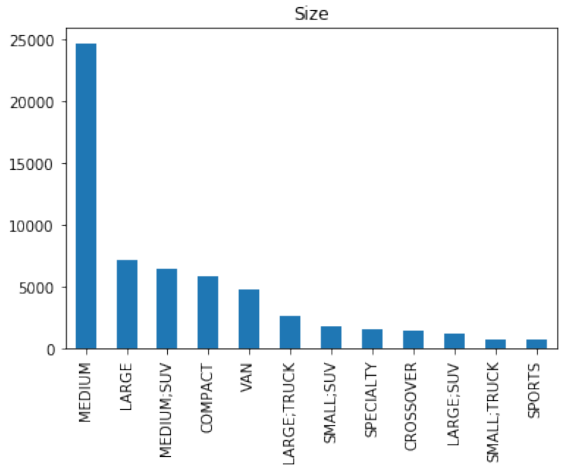
Graph 10 – VehOdo boxplot

- **Nationality:** is an object attribute that shows the country where the vehicle was manufactured. This attribute can be transformed into a categorical data because it has only four types of data (AMERICAN, OTHER; ASIAN, TOP; LINE; ASIAN, OTHER). OTHER has only 152 occurrences, so we eliminate it as an outlier. It has 4 missing values, we decided to fill them because we can know the Nationality from the attribute 'Make'. We decide to transform the remaining other 3 types (AMERICAN, OTHER; ASIAN, TOP; LINE; ASIAN) into categorical dates, 1, 2, 3.

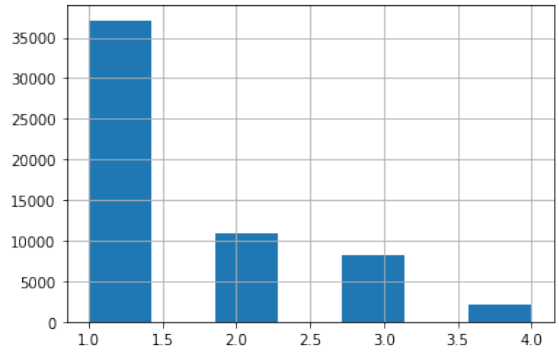


Graph 11 – Nationality distribution

- **Size:** is an object attribute which identifies the size of the vehicle. Data shows that there is a perfect pairwise correlation between Model and Size and a strong correlation between Size and SubModel, so, because of this fact we decided to eliminate the Model and SubModel as they are more difficult to manage as compared to the Size and these values are not classifiable. Size has 4 missing values and we decided to fill them as we can observe the Size from the Model. Size has 12 types, some of them are redundant. In particular, we decide to group MEDIUM; SUV, CROSSOVER and VAN types into MEDIUM type, LARGE; SUV and LARGE;TRUCK, into LARGE type, COMPACT, SMALL;SUV and SMALL;TRUCK into the new type SMALL, SPORTS and SPECIALTY into new type SPORT. So now we have MEDIUM, LARGE, SMALL and SPORT types that we have transformed into categorical data, 1, 2, 3, 4.



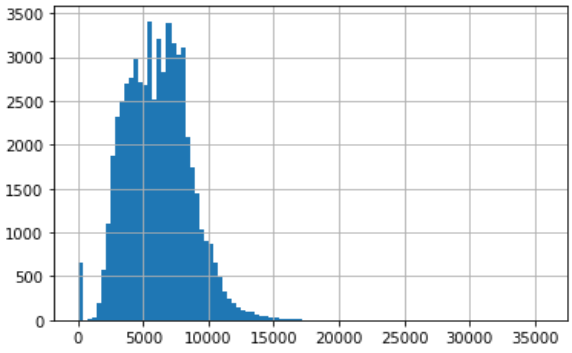
Graph 12 – Size initial distribution



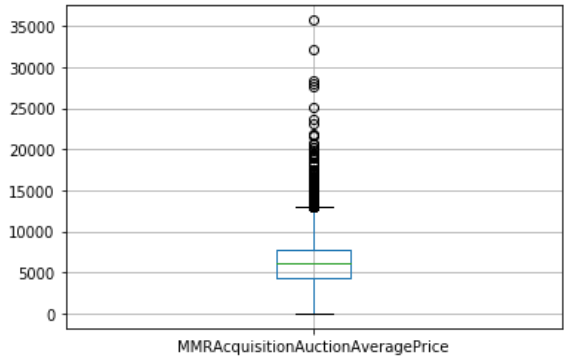
Graph 13 – Distribution of Size after grouping

- **TopThreeAmericanName:** is an object attribute that show if the manufacturer belongs top three American manufacturers. It has 4 missing values, which can be filled as we can retrieve TopThreeAmericanName from the attribute 'Make'. It has 4 types of values, GM, CHRYSLER, FORD, OTHER, we decide to eliminate this attribute because we already have the Nationality and Make attributes that convince us about the redundancy of TopThreeAmericanName.

- **MMRAcquisitionAuctionAveragePrice** is a numerical attribute which provides us the acquisition price of the vehicle in average condition at the time of purchase. It has 13 missing values which will be replaced with the average value. We have eliminated the values which are greater than 13500 as they are outliers. There are some missing values that has a 0 as the value, so we decided to eliminate the rows of such missing values.

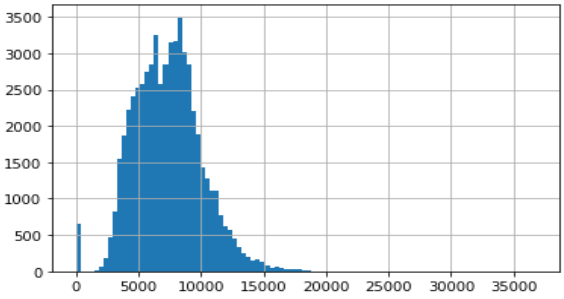


Graph 14 – MMRAAAP distribution

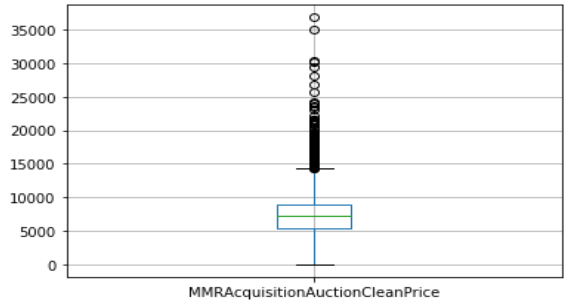


Graph 15 – MMRAAAP boxplot

- **MMRAcquisitionAuctionCleanPrice:** is a numerical attribute assigned to the acquisition price of the vehicle in the above Average condition at the time of purchase. It has 13 missing values which we have replaced with the average value. We decided to eliminate the values which are greater than 14500 as they are outliers.There are some missing values that has a 0 as the value, so we decided to eliminate the rows of such missing values.

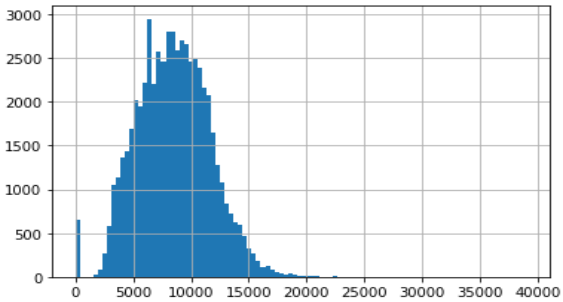


Graph 16 – MMRAACP distribution

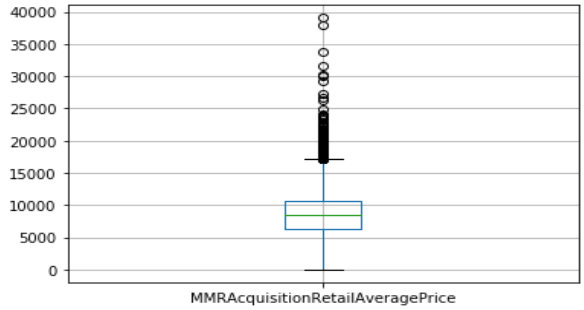


Graph 17 – MMRAACP boxplot

- **MMRAcquisitionRetailAveragePrice:** is a numerical attribute which explains the acquisition price for the vehicle in the retail market in average condition at the time of purchase. It has 13 missing values which have been replaced by taking average value. We decided to eliminate the values which are greater than 17000 as they are outliers. There are some missing values that has a 0 as the value, so we decided to eliminate the rows of such missing values.

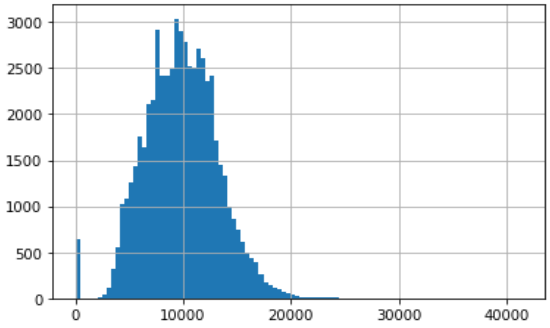


Graph 18 – MMRARAP distribution

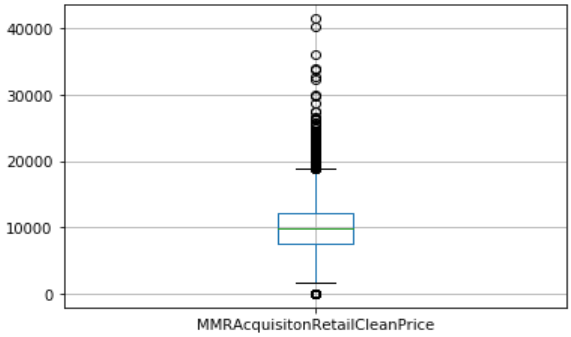


Graph 19 – MMRARAP boxplot

- **MMRAcquisitonRetailCleanPrice:** is a numerical attribute which provides the acquisition price for the vehicle in the retail market in above average condition at the time of purchase. There are 13 missing values which we decided to replace with the average value. We eliminated the values greater than 19000 and less than 300 as they are outliers. There are some missing values that has a 0 as the value, so we decided to eliminate the rows of these missing values

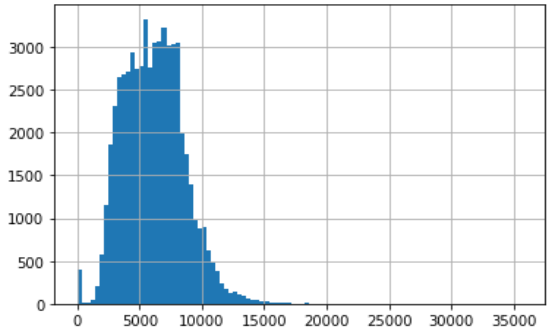


Graph 20 – MMRARCP distribution

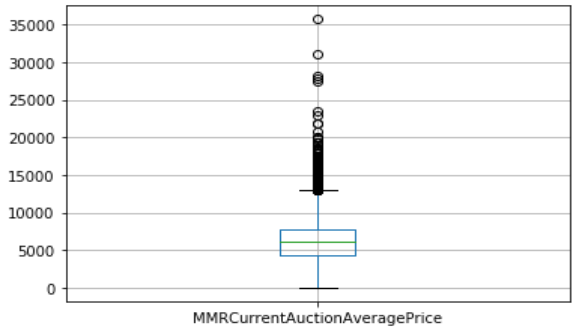


Graph 21 – MMRARCP boxplot

- **MMRCurrentAuctionAveragePrice:** is a numerical attribute which provides the acquisition price for the vehicle in average condition as of current day. It has 245 missing values that will be replaced with the average value. The values which are greater than 13 000 will be eliminated as they are outliers. There are some missing values that has a 0 as the value, so we decided to eliminate the rows of these missing values.

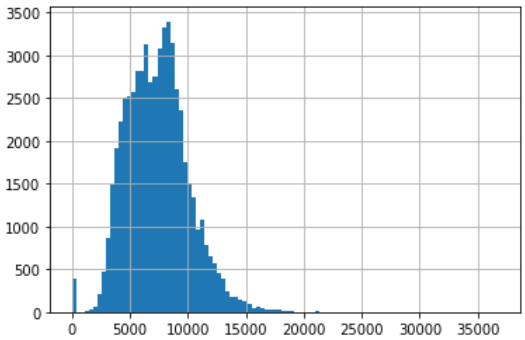


Graph 22 – MMRCAAP distribution

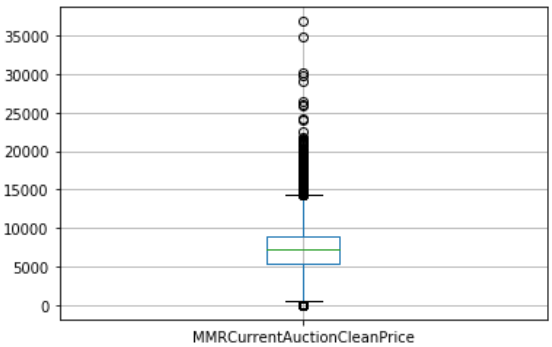


Graph 23 – MMRCAAP boxplot

- **MMRCurrentAuctionCleanPrice:** is a numerical attribute which describes the acquisition price of the vehicle in the above condition as of current day. There are 245 missing values of this attribute and we decided to replace with the average value. We decided to eliminate the values which are greater than 14500 as they are outliers. We have also eliminated the rows where there are missing values that has a 0 as the value.

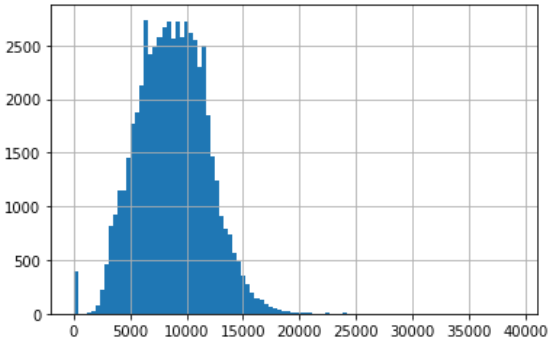


Graph 24 – MMRCACP distribution

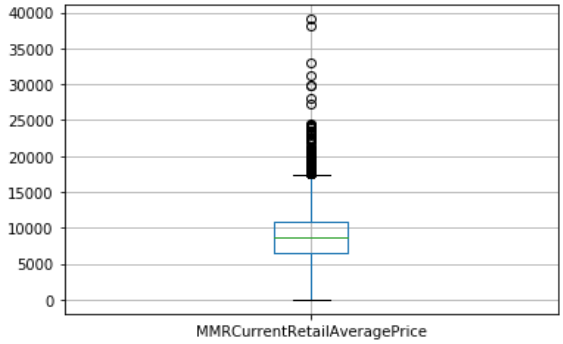


Graph 25 – MMRCACP boxplot

- **MMRCurrentRetailAveragePrice:** is a numerical attribute which describes the acquisition price of the vehicle in the retail market in average condition as of current day. It has 245 missing values which we decided to replace with the average value. We eliminated the values greater than 17000 and less than 100 as they are outliers. There are some missing values that has a 0 as the value, so we decided to eliminate the rows of these missing values.

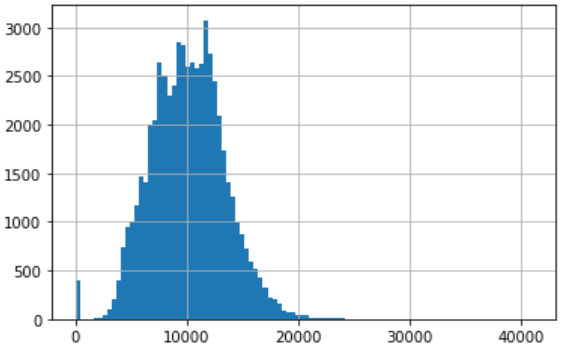


Graph 26 – MMRCRAP distribution

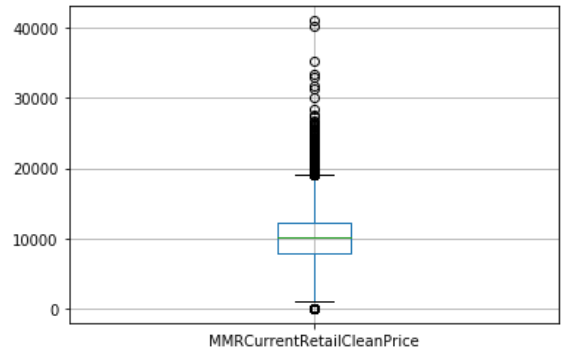


Graph 27 – MMRCRAP boxplot

- **MMRCurrentRetailCleanPrice:** is a numerical attribute which represents the acquisition price of the vehicle in the retail market in above average condition as of current day. It has 245 missing values which have been replaced with the average value. We decided to eliminate the values greater than 19500 and less than 100 as they are outliers. There are some missing values that has a 0 as the value, so we decided to eliminate the rows of these missing values.



Graph 28 – MMRCRCP distribution



Graph 29 – MMRCRCP boxplot

- **PRIMEUNIT:** is an object attribute which identifies if the vehicle would have a higher demand than a standard purchase. This attribute consisted on 55695 missing values and that is a lot of missing data so it. It is near to impossible to replace these missing values, so we decided to eliminate the entire attribute PRIMEUNIT.

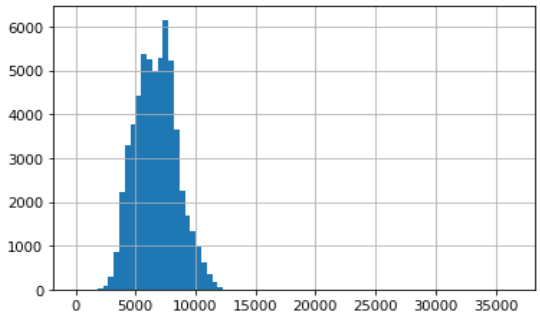
- **AUCGUART:** is an object attribute that represents the guarantee level provided by the auction for the vehicle (Green light - guaranteed/arbitrable, Yellow Light - caution/issue, Red light - sold as it is). This attribute has 55695 missing values and because of such a huge amount of data it cannot be replaced or ignored so we decided to eliminate this entire attribute.

- **BYRNO:** is a numerical attribute which shows the unique number assigned to the buyer that purchased the vehicle. This attribute does not describe anything relevant to assess the qualities of the purchase, so we decided to eliminate this attribute.

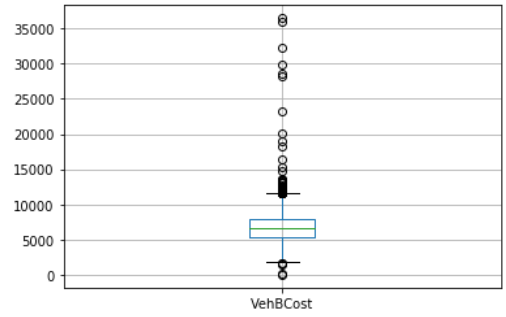
- **VNZIP** is a numerical attribute which describes the zip code to identify the place where the car was purchased.

- **VNST** is an object attribute that represents the state where the car was purchased. There is a strong correlation between VNST and VNZIP. We decide to keep VNST because it is easier to manage and VNZIP is too much specific for our work.

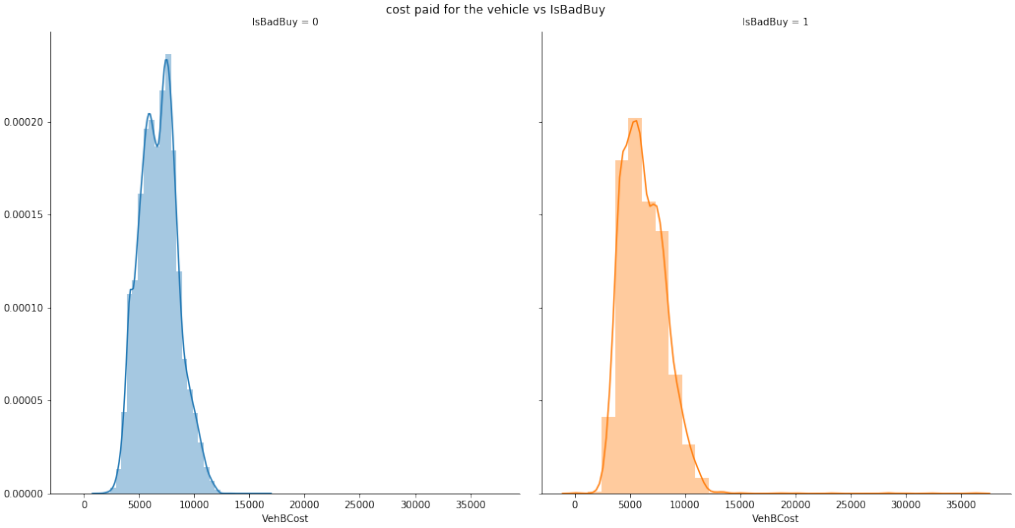
- **VehBCost:** is a numerical attribute which describes the cost paid for the vehicle at the time of purchase. The outliers exist when the values are greater than 11500 and less than 2000. So, they are eliminated.



Graph 30 – VehBCost distribution



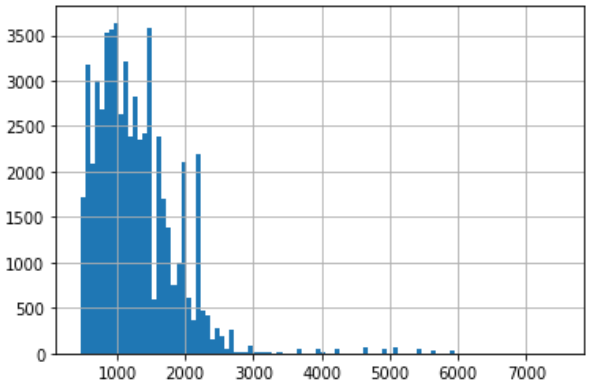
Graph 31 – VehBCost boxplot



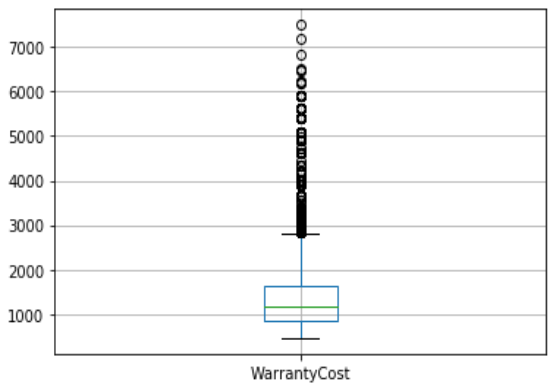
Graph 32 – Distribution of VehBCost related to good and bad purchase

- **IsOnlineSale:** is a discrete attribute which takes binary values (0 or 1), 1 mean that it has been sold online, 0 wasn't sold online. There are 56879 rows which have a 0 value and there are 1499 rows which have 1 value. This attribute does not consist of any missing value.

- **WarrantyCost:** is an integer attribute which explains the Warranty price. The values of this attribute were contaminated with semicolons and initially the type of this attribute was object. Semicolons are meaningless for our purposes, so we had to eliminate them. We have also changed the type of this attribute from object to integer. The outliers exist when the values are greater than 2800. So, they are eliminated.



Graph 33 – WarrantyCost distribution



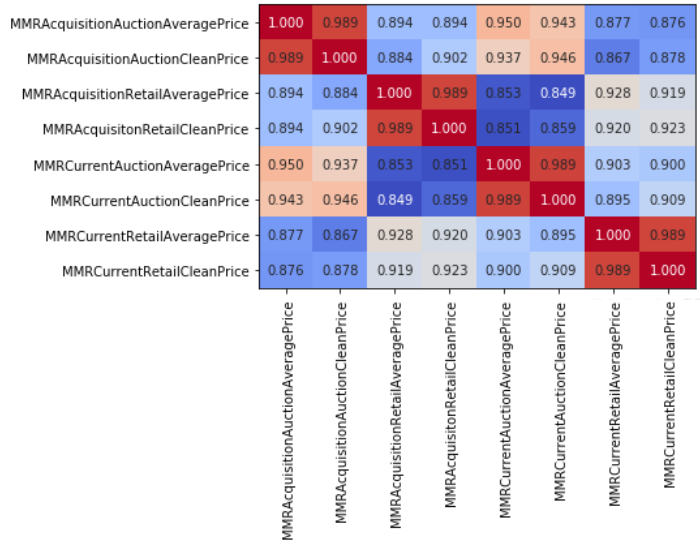
Graph 34 – WarrantyCost boxplot

Attributes elimination

After Studying, analyzing, evaluating and treating all the attributes, we decided to eliminate the entire columns of below-mentioned attributes:

- MMRAcquisitionAuctionCleanPrice
- MMRAcquisitionRetailAveragePrice
- MMRAcquisitonRetailCleanPrice
- MMRCurrentAuctionCleanPrice
- MMRCurrentRetailAveragePrice
- MMRTCurrentRetailCleanPrice

These attributes can be seen as couples as they are providing similar values. The reason to eliminate the entire is that we can clearly observe a strong correlation between the four couples of AveragePrice and CleanPrice and currently we are focusing on the auction market instead of retail market.



Graph 35 - Correlation between the MMR attributes

Then we have also decided to eliminate the attributes SubModel and Models, these two attributes are useless for our purpose because they are too much specific and we can find all the important information about the type of the car form the attribute Make and Size.

So, at the end of the first part we are only left with these attributes:
IsBadBuy, Auction, VehYear, VehicleAge, Make, Trim, Color, Transmission, WheelTypeID, VehOdo, Nationality, Size, MMRAcquisitionAuctionAveragePrice, MMRCurrentAuctionAveragePrice, VNST, VehBCost, isOnlineSale and WarrantyCost.

Clustering

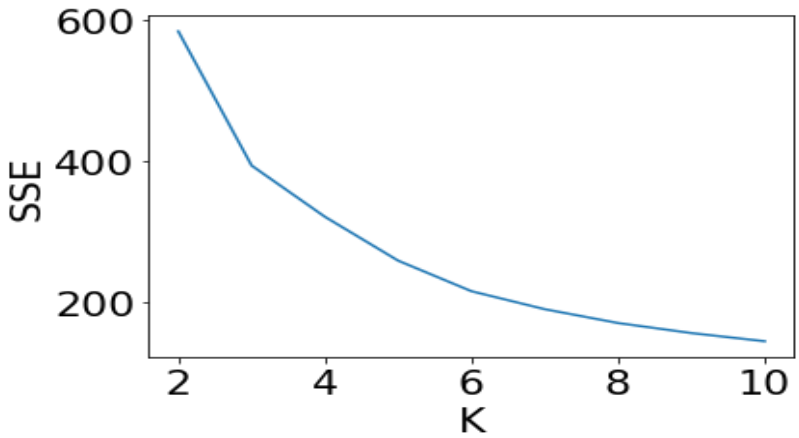
For the cluster analysis, we choose to use the following attributes:

- MMRAcquisitionAuctionAveragePrice;
- WarrantyCost;
- VehBCost.

The reason to use the above mentioned attributes is that they are numerical values and provide ust he values in terms of dollars. We ran three different types of clustering algorithm for analyze the different results and understand some correlation between the attributes used.

• K-Mean

The first algorithm used is K-means. First of all, using the standard normalization, we have normalized all the data used in this case. Then we ran the K-Means from two to eleven in order to find the best value of K. We ran K fifty times and we have obtained the graph representing the relationship between K and SSE. After that we observed from the graph that the knee point on the K-Means is near to 4 because the curve of graph is stabilized around that point.
So, we ran the algorithm using K equal to 4, 5 and 6. Analyzing the distribution of the clusters we see that using K equal to 5 or 6, the result will lead to create similar, insignificant and definitely to redundant clusters, so we decided to take 4. We minimized the error of K-Means in relation with the number of clusters.



Graph 36 – Graph for the best K

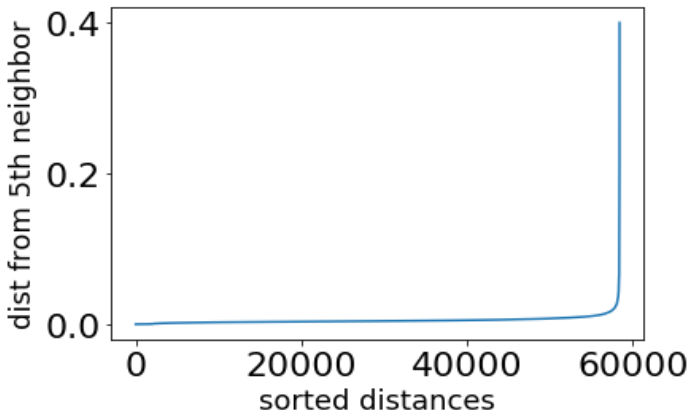
- **Correlation between WarrantyCost and VehBCost:** The correlation between WarrantyCost and VehBCost is “0.03”. The distribution of the values is not well separated in the graph, but we can easily observe four different clusters and then we can observe that when the WarrantyCost is higher, the density is lower.

- **Correlation between WarrantyCost and MMRAcquisitionAuctionAveragePrice**: The correlation between the WarrantyCost and MMRAcquisitionAuctionAveragePrice is “0.05”. The distribution of values in the graph is not well separated. We can see four different clusters and when the WarrantyCost is lower, the density is higher.

- **Correlation between MMRAcquisitionAuctionAveragePrice and VehBCost:** The Correlation between MMRAcquisitionAuctionAveragePrice and VehBCost is “0.079”. As we can observe in the cluster that these attributes are related to each other because the distribution is not well separated. We can see two clusters, but they are not distinguishable.

- **DB Scan**

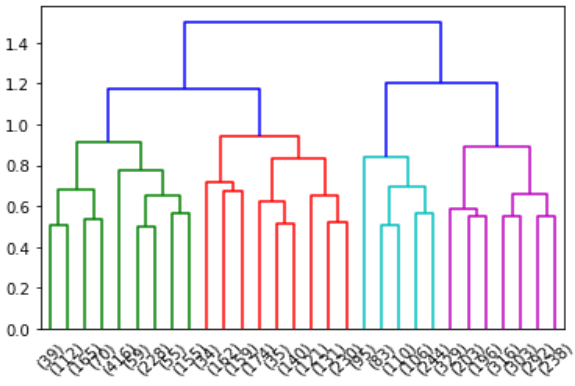
The second algorithm used in this task is the DB Scan algorithm. This algorithm is based on the density of the points and it use this density to build the different clusters. First of all, we have chosen the number of minimum points for make a cluster and the radius of neighborhood around a point X. The values chosen for MinPoints is 5 and the value of Epsilon in 0.03, based to the graph 35.
After we have run the DB Scan algorithm, we observed that it is useless in our case because this method is based on density, we use attributes that are very near to each other, so the result is consisted of one big cluster and some outliers. So, this type of clustering does not provide us any information about the different correlation between the attributes.



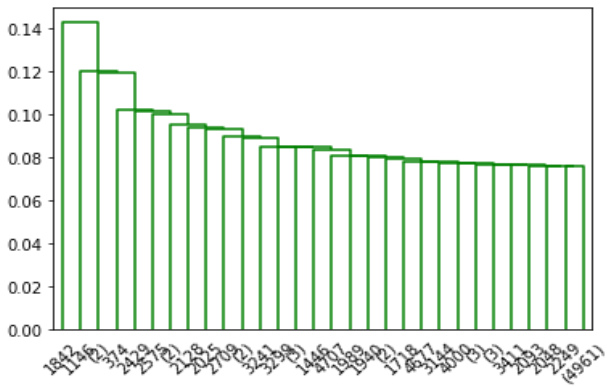
Graph 37 – Graph for the best MinPoints

- **Hierarchical Clustering:**

The last algorithm used for the clustering is the Hierarchical algorithm. This algorithm is computationally heavy. Because it is taking a lot of time to process the whole dataset with Hierarchical algorithm, we had used the K-Means algorithm for reducing the dimension of the dataset to 5000 values, taking K equal to 5000. Using this method, we have been able to reduce the whole dimension of the dataset without modify the distribution of the attributes in the data. After this transformation we ran the hierarchical algorithm on the point that has been normalized using the MinMax normalization. We used the Complete Link method and the Single Link method. As we can see the Complete Link method is the way more significative. In fact, in the Single Link, we have obtained just one big single cluster, that is useless for our purpose. Otherwise, the Complete Link is useful for us, doing a cut at 1.0, as the Graph 36 show us, we obtain 4 different cluster. We have used also the Average Distance method, but the result of this way was no more relevant than the Complete Link.



Graph 36 - MinMax Complete Link



Graph 37 - MinMax Single Link

Association Rules Mining

Data Preparation

For finding patterns we must have categorical data because with a lot of continuous data like in our case, doing it will take a plenty of time. So, we are going to transform the continuous attributes such as VehBCost, WarrantyCost, VehOdo and MMRAcquisitionAuctionAveragePrice, into categorical data and grouping them into single values that represent a range of the previous data. We divided this data into 15 groups and found the number of occurrences within the range of each group. The number of occurrences according to the range of each group can be observed in the table mentioned below:

VehBCost	VehOdo	MMRAAAP	WarrantyCost
[2000 – 2633) 54	[5368 – 12724) 4	[884 – 1723) 172	[462 – 617) 4797
[2633 – 3266) 435	[12724 – 20081) 22	[1723 – 2563) 1861	[617 – 773) 6169
[3266 – 3900) 1576	[20081 – 27437) 100	[2563 – 3403) 5090	[773 – 929) 8234
[3900 – 4533) 4290	[27437 – 34794) 348	[3403 – 4242) 6038	[929 – 1085) 4953
[4533 – 5166) 4930	[34794 – 42151) 1179	[4242 – 5082) 6648	[1085 – 1241) 6722
[5166 – 5800) 6796	[42151 – 49507) 3333	[5082 – 5922) 6861	[1241 – 1396) 6652
[5800 – 6433) 7319	[49507 – 56864) 5156	[5922 – 6762) 6880	[1396 – 1552) 4699
[6433 – 7066) 6590	[56864 – 64220) 6644	[6762 – 7601) 7624	[1552 – 1708) 3937
[7066 – 7700) 7932	[64220 – 71577) 8871	[7601 – 8441) 6460	[1708 – 1864) 2217
[7700 – 8333) 7112	[71577 – 78934) 11524	[8441 – 9281) 3858	[1864 – 2020) 3038
[8333 – 8966) 3981	[78934 – 86290) 10593	[9281 – 10121) 2219	[2020 – 2175) 2639
[8966 – 9600) 2273	[86290 – 93647) 6360	[10121 – 10960) 1508	[2175 – 2331) 994
[9600 – 10233) 1640	[93647 – 101003) 1945	[10960 – 11800) 662	[2331 – 2487) 679
[10233 – 10866) 947	[101003 – 108360) 184	[11800 – 12640) 303	[2487 – 2643) 291
[10866 – 11509) 423	[108360 – 115867) 35	[12640 – 13492) 114	[2643 – 2801) 277

Tab 1 – Division of the continues attributes used in the Associations task

In order to find the frequent, closed and maximal itemset we decided to remap all the categorical data. For the simplification and clear comprehension of the apriori results we have transformed the number into the corresponding value. We decided to use apriori algorithm in order to find the frequent, closed and maximal itemset. The minimum support is 10% and the minimum number of items in the itemset is 1.

Support	Frequent Itemset	Closed Itemset	Maximal Itemset
90,00%	- Transmission = 2 (Supp = 0.96) - IsOnlineSale = 0 (Supp = 0.97) - Transmission = 2 & IsOnlineSale = 0 (Supp = 0.94)	-Transmission = 2 (Supp = 0.96) - IsOnlineSale = 0 (Supp = 0.97) -Transmission = 2 & IsOnlineSale = 0 (Supp = 0.94)	
80,00%	- IsBadBuy = 0 (Supp = 0.85) - Nationality = 1 (Supp = 0.84) - Nationality = 1 & Transmission = 2 (Supp = 0.82) - Nationality = 1 & IsOnlineSale = 0 (Supp = 0.82) - IsBadBuy = 0 (Supp = 0.88) - IsBadBuy = 0 & Transmission = 2 (Supp = 0.85) - IsBadBuy = 0 & Transmission = 2 & IsOnlineSale (Supp = 0.82) - IsBadBuy = 0 && IsOnlineSale = 0 (Supp = 0.85) - Transmission = 2 (Supp = 0.96) - IsOnlineSale = 0 (Supp = 0.97) - Transmission = 2 & IsOnlineSale = 0 (Supp = 0.94)	- IsBadBuy = 0 (Supp = 0.85) - Nationality = 1 (Supp = 0.84) - Nationality = 1 & Transmission = 2 (Supp = 0.82) - Nationality = 1 & IsOnlineSale = 0 (Supp = 0.82) - IsBadBuy = 0 (Supp = 0.88) - IsBadBuy = 0 & Transmission = 2 (Supp = 0.85) - IsBadBuy = 0 & Transmission = 2 && IsOnlineSale (Supp = 0.82) - IsBadBuy = 0 & IsOnlineSale = 0 (Supp = 0.85) - Transmission = 2 (Supp = 0.96) - IsOnlineSale = 0 (Supp = 0.97) - Transmission = 2 & IsOnlineSale = 0 (Supp = 0.94)	
70,00%	- Nationality = 1 (Supp = 0.84) - Nationality = 1 & IsBadBuy = 0 (Supp = 0.74) - Nationality = 1 & IsBadBuy = 0 & Transmission = 2 (Supp = 0.72) - Nationality = 1 & IsBadBuy = 0 & IsOnlineSale = 0 (Supp = 0.72) - Nationality = 1 & Transmission = 2 (Supp = 0.82) - Nationality = 1 & Transmission = 2 & IsOnlineSale = 0 (Supp = 0.79) - Nationality = 1 & IsOnlineSale = 0 (Supp = 0.82) - IsBadBuy = 0 (Supp = 0.85) - IsBadBuy = 0 & Transmission = 2 (Supp = 0.85) - IsBadBuy = 0 & Transmission = 2 & IsOnlineSale (Supp = 0.82) - IsBadBuy = 0 & IsOnlineSale = 0 (Supp = 0.85) - Transmission = 2 (Supp = 0.96) - IsOnlineSale = 0 (Supp = 0.97) - Transmission = 2 & IsOnlineSale = 0 (Supp = 0.94)	- Nationality = 1 (Supp = 0.84) - Nationality = 1 & IsBadBuy = 0 (Supp = 0.74) - Nationality = 1 & IsBadBuy = 0 && Transmission = 2 (Supp = 0.72) - Nationality = 1 & IsBadBuy = 0 && IsOnlineSale = 0 (Supp = 0.72) - Nationality = 1 & Transmission = 2 (Supp = 0.82) - Nationality = 1 & Transmission = 2 && IsOnlineSale = 0 (Supp = 0.79) - Nationality = 1 & IsOnlineSale = 0 (Supp = 0.82) - IsBadBuy = 0 (Supp = 0.85) - IsBadBuy = 0 & Transmission = 2 (Supp = 0.85) - IsBadBuy = 0 & Transmission = 2 & IsOnlineSale (Supp = 0.82) - IsBadBuy = 0 & IsOnlineSale = 0 (Supp = 0.85) - Transmission = 2 (Supp = 0.96) - IsOnlineSale = 0 (Supp = 0.97) - Transmission = 2 & IsOnlineSale = 0 (Supp = 0.94)	

Tab 2 – Frequent, Closed and Maximal itemset extract from the dataset

After applying the apriori algorithm, we divided the itemset into three subsets. The first subset consisted on all itemset with support greater than 90%. The second subset consisted on all the itemset with support greater than 80%. The third subset consisted on all the itemset with support greater than 70%. It can be observed in the table that all the itemset are distinguished with respect to their minimum required support. In the table we can see the transmission auto and American nationality has appeared frequently. We have observed that the frequent itemset and closed itemset are same. These frequent and closed itemset can been in the above table. We applied apriori algorithm and this algorithm could not provide us any maximal itemset.

Having a lot of association, we choose to keep only the more significant of them, below mentioned:

1. Americans vehicles have AUTO transmission.
2. Other,Asian vehicles that have AUTO transmission are LARGE in Size.
3. Ford vehicles have Alloy WheelType.
4. Vehicles which comes from Florida are sell at Manheim auction.
5. Black colored vehicles which are not sold online are good buy.
6. When TopThreeAmericanName is CHRYSLER and Nationality is AMERICAN and Transmission is AUTO the size

Association Rules	Confidence	Lift
Nationality = 1 → Transmission = 2	0,97	1,01
(Nationality = 2, Transmission = 2) → Size = 1	0,78	1,23
Make = FORD → WheelTypeID = 1,0	0,72	1,47
VNST = FL → Auction = 2	0,77	1,37
(Color = Black, IsOnlineSale = 0) → IsBadBuy = 0	0,89	1,01
IsBadBuy = 0 → Nationality = 1	0,84	1
(Make = CHRYSLER, Nationality = 1 & Transmission = 2) → Size = 1	0,79	1,24
(Nationality = 1, Make = FORD) → WheelTypeID = 1.0	0,71	1,46
(Nationality = 1, Trim = LS, Auction = 2) → Make = CHEVROLET	0,97	4,1

Tab 3 – Association rules in the dataset

We are unable to find any conclusive results from the rules which indicate that either it is a good purchase or a bad purchase. So, in order to have conclusive results we decided to restrict the rules with only IsBadBuy = 0 as the results of the rule.

Rules	Confidence	Lift
(Nationality = 0, Transmission = 2) ->IsBadBuy = 0	0.88	1.00
(VehYear = 2007, Transmission = 2) => IsBadBuy = 0	0.93	1.05
(WheelTypeID = 2.0, Size = 1) => IsBadBuy = 0	0.92	1.05
(Color = White, Transmission = 2) => IsBadBuy = 0	0.87	1.00
(Make = CHEVROLET) => IsBadBuy = 0	0.90	1.03
(Auction = 1, IsOnlineSale = 0) => IsBadBuy = 0	0.85	0.96
(VehOdo = [64220.8, 71577.4], Transmission =0) => IsBadBuy = 0	0.89	1.01
(Make = FORD, Transmission = 2, IsOnlineSale = 0) => IsBadBuy = 0	0.84	0.96

Tab 4 – Association rules in dataset with IsBadBuy as a result

Replacing missing values using association rules

The point of the association rules is that we can observe that one character or a correlation of more characters can lead to a particular type of attribute. So, if there are some missing values about an attribute but there are other values in the same rows that are linked by a rule of association that lead to a particular value of the missing attribute, we can deduce the missing values using this association rules.
In this case we have found some helpful association rules that can fill some missing values.

We can replace 920 missing values that belong to the WheelTypeID attribute using this association rules that we have found:

- Make = 'FORD' → WheelTypeID = 1 (410 replaced)
- VehYear = 2008 → WheelTypeID = 2 (181 replaced)
- Size = 2 → WheelTypeID = 1 (275 replaced)
- VehYear = 2007 & Make = 'CHEVROLET' → WheelTypeID = 2 (54 replaced)

We can also replace 130 missing values that belong to the Trim attribute with the same method:

- Make = 'PONTIAC' → Trim = 2.0 (130 replaced)

Classification

In this task we will analyze the different decision tree obtained from different algorithms. This tree will be used then for the final algorithm which will predict the good or the bad purchase.

First of all, we have decided which attributes to use for building the decision tree, so starting from the cleaned dataset obtained from the data understanding we have selected different attributes. These attributes have been chosen because they are categorical data, they minimized the classification error analyzed by the features selection, and because, starting for the association rules, they look like the most meaningful attributes in the whole dataset. The attributes chosen are: VehicleAge, Transmission, WheelTypeID, Size, Nationality, VehOdo, MMRAcquisitionAuctionAveragePrice and VehBCost. On the attributes VehOdo, MMRAcquisitionAuctionAveragePrice and VehBCost we had to do a division in the range of the attributes for modify these attributes from continues data to a categorical data. We have used the same transformation used in the association task, so we have divided the range of this attributes in 15 different cuts. The division of these attributes can be watched at page 13 of this relation.

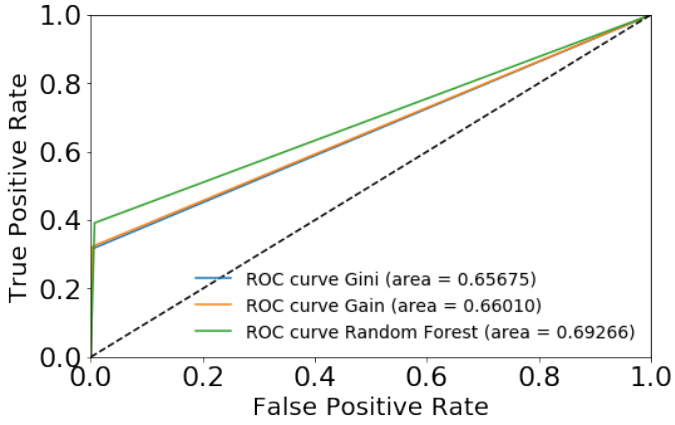
Starting from that we build the classification tree, initially without pruning. We have already seen that most of the leaves of the tree has few numbers of values, so the pruning for avoid the overfitting could be recommended.

These are the results of the different classification algorithm used on the dataset. The first algorithm used is the Decision Tree Learned using Gini Index as metrics and then also with Information Gain as metrics. The second classification algorithm used is the Random Forest.

Algorithm	Accuracy	Precision	Recall	F-Measure	TP	FP	TN	FN
Gini Decision Tree Learner	0,914	0,91	1,0	0,95	49268	149	2178	4703
Gain Decision Tree Learner	0,914	0,91	1,0	0,95	48782	149	2224	4657
Random Forest	0,912	0,92	0,99	0,96	49089	328	2697	4184

Tab 5 – Results of the different algorithms

As we can see, the Gini Decision Tree algorithm has the almost same values of the Gain Decision Tree while the Random Forest give us quite better results as we can also see on the chart below.



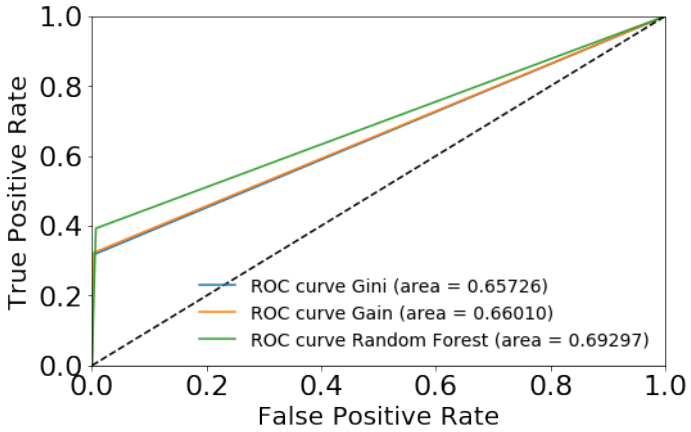
Graph 38 - Comparison of the roc curves using different algorithms

Doing the classification, we have seen that changing the value of the initially missing value in WheelTypeID, the result of the algorithms would change in a positive way, so we have gone back to the data understanding and we have seen a correlation between this missing values and the bad purchase. As well as we see that the 75% of the rows with WheelTypeID equal to 0.0 in are bad purchase. So, we have decided to try these algorithms replacing the missing values not changed in the association task with the value 0.0. These are the results:

Algorithm	Accuracy	Precision	Recall	F-Measure	TP	FP	TN	FN
Gini Decision Tree Learner	0,914	0,91	1,0	0,95	49268	149	2185	4696
Gain Decision Tree Learner	0,914	0,91	1,0	0,95	49268	149	2224	4657
Random Forest	0,912	0,92	0,99	0,96	49084	333	2702	4179

Tab 6 - Results of the different algorithms with the modify on WheelTypeID

As we can see from the table the results are quite the same, but in this case the precision for the bad purchase is a little better. In both case the best algorithm between Gini, Gain and Random Forest is the random forest as we can also see form the roc chart.



Graph 39 - Comparison of the roc curves using different algorithms and replacing the missing values

For the tree without pruning we can see that the best tree solution is the Decision Tree Learner using Gain as metrics. The accuracy in that case is equal to 0,914 which is equal to the Decision Tree Learner using Gini as metrics but seeing the confusion matrix we can see the that the algorithm with Gain give us the best results.
Otherwise changing the missing values of WheelTypeID with 0.0 the results don't change enough for justify this choice, so we have decided to continue using the dataset come out from the Data Understanding.

Pruning

Seeing the results of the previously tree we can see the problem with overfitting, so for avoid this problem we have gone to apply the pruning to the tree already built, post-pruning, and also to try some modify before the building of the tree, pre-pruning.
The results of the pruned tree are not better than the not pruned tree. For example, for the pre-pruning we have apply the pruning shrinking the minimum number of samples for leaf, starting from 100 and walking down until 5. The result doesn't go better, rather the accuracy decreases and the value in the confusion matrix are quite the same.

	Gain no Pruning	Gain post-pruning
Accuracy	**0,914**	**0,890**
Precision	**0,91**	**0,89**
Recall	**1,0**	**0,99**
F-Measure	**0,95**	**0,94**
True Positive	**48782**	**49042**
False Positive	**149**	**375**
True Negative	**2224**	**1079**
False Negative	**4657**	**5802**

Tab 7 – Comparison between pruned and non pruned tree

In the table above we can see the difference between the algorithm Tree Decision Learner using Gain without pruning (on the left) and the same algorithm using a post-pruning reducing the minimum number of samples for leaf equal to 25. As we have already seen the results are worse in the pruning tree.
Also in the pre-pruning the results are similar to those in the table. We tried the pre-pruning shrinking the maximal depth of the tree from 5 to 9, but no one of the attempts give us better results than the not pruned tree.

Conclusion

From the results of the classification task we can pull out some different conclusion.
For example, we have some threshold in the attributes MMRAcquisitionAuctionAveragePrice and VehOdo, above this threshold the 83% of the purchase are bad while beside this the purchase are good purchase.
The best Classification Algorithm for this dataset is the Random Forest without apply the pruning.
All the algorithm gives us a good result for accuracy and precision, but the confusion matrix is a little bit unbalanced. The matrix is very good for the results about the good purchase but is not precise for the bad purchase. This can be explained by the big disparity of bad and good purchase in the training dataset.
In fact, applying again all the classification task on a balanced training set we can see that the results are really better.

	Gain Unbalanced	Gain Balanced
Accuracy	**0,914**	**0,910**
Precision	**0,91**	**0,986**
Recall	**1,0**	**0,845**
F-Measure	**0,95**	**0,91**
True Positive	**48782**	**2465**
False Positive	**149**	**35**
True Negative	**2224**	**2086**
False Negative	**4657**	**414**

Tab 8 – Comparison between balanced and unbalanced tree

In this table we can see the result of the Tree Decision Learner using Gain on a balanced dataset. The results for Accuracy, Precision, Recall and F-Measure are quite the same, but we have obtained a really better confusion matrix.