UNIVERSITY OF PISA
COMPUTER SCIENCE

Carvana: Is A Bad Buy?

Data Mining Project

Alessandro Cudazzo
Giulia Volpi
Flavia Achena
Aleksandra Maslennikova

ACADEMIC YEAR 2019/2020

1. Introduction

Carvana is a start-up business launched by a well-established American company. The goal is to change completely the way people buy, finance, and trade their used vehicles by replacing physical infrastructure with technology and top of the line scientific models. In this report, we will show our analysis based on the dataset published on kaggle.com for the *Data Mining 2019/2020 Project*. The aim is to build a model to advise future customers whether a purchase could be a good or bad buy.
The report is structured as follows: in Section 2, the available data have been analyzed. In particular, we reflected on data semantics and distribution, evaluated data quality, etc. In Section 3, we have searched for potential clusters in the data; we have reported the results of three different clustering approaches: K-means, DBSCAN, and Hierarchical. In Section 4, we have discussed the most interesting frequent patterns and rules extracted from the data. While in Section 5, a prediction model built with the Decision Tree is presented, whereas Section 6 is devoted to conclusions.

2. Data Understanding

Categorical		**Numerical**	
Discrete	**Binary**	**Continuous**	**Discrete**

`RefId`, `Auction`, `Make`, `Model`, `Trim`, `SubModel`, `Color`, `Transmission`, `WheelTypeID`, `WheelType`, `Nationality`, `Size`, `TopThreeAmericanName`, `PRIMEUNIT`, `AUCGUART`, `BYRNO`, `PurchDate` `VNZIP1`, `VNST`	`IsBadBuy`
	`IsOnlineSale`
		`MMRAcquisitionAuctionAveragePrice`, `MMRAcquisitionAuctionCleanPrice`, `MMRAcquisitionRetailAveragePrice`, `MMRAcquisitonRetailCleanPrice`, `MMRCurrentAuctionAveragePrice`, `MMRCurrentAuctionCleanPrice`, `MMRCurrentRetailAveragePrice`, `MMRCurrentRetailCleanPrice`, `VehBCost`	`WarrantyCost`
			`VehYear`
			`VehicleAge`
			`VehOdo`

Table 2.1: *Categorical and numerical attributes*

The Carvana Dataset consists of 58386 records described by 34 features, which 21 are categorical and 13 numerical (see Tab. 2.1). In the provided documentation there were also some attributes (`KickDate`, `AcquisitionType`) not present in the dataset and it was impossible to recover (from other sources) or derive them (from the attributes already provided).

2.1 Data semantics

From the semantics point of view, the most interesting features are MMR ones:

- `MMRAcquisitionAuctionAveragePrice`, `MMRAcquisitionRetailCleanPrice`, `MMRAcquisitionAuctionCleanPrice`, `MMRAcquisitionRetailAveragePrice`
- `MMRCurrentAuctionAveragePrice`, `MMRCurrentAuctionCleanPrice`, `MMRCurrentRetailAveragePrice`, `MMRCurrentRetailCleanPrice`

`MMR` stands for `Manheim Market Report`, which calculates a price for a car model in general: not the price paid for the particular car, but the overall average cost of similar vehicles. This estimated price varies depending upon different market conditions. `Acquisition` indicates the price at the moment of purchase, `Current`, on the other hand, at the time the dataset was released. `Auction` refers to the auction price, `Retail` suggests the price which the customer is willing to pay at the dealership center. Lastly, `Clean` indicates the vehicle in good conditions and `Average` refers to the vehicle that may have some mechanical or cosmetic problems, that may require reconditioning.

All the other features can be understood with a brief description, as shown in the following Tab. 2.2 .

Feature	Description
RefId	is the unique (sequential) number assigned to vehicles
IsBadBuy	identifies if the kicked vehicle was an avoidable purchase
PurchDate	indicates the date the vehicle was purchased at auction
VehYear	indicates the manufacturer's year of the vehicle
VehicleAge	indicates the years elapsed since the manufacturer's year
Make, Model, Trim, SubModel, Color, Transmission, Size	represent vehicle's brand characteristics and are mostly self-explanatory. E.g. Make provides the models manufactured, SubModel more detailed Model's specifications, Transmission: Auto or Manual.
WheelTypeID, WheelType	identify the type of the vehicle wheel, the first is the id of the type indicated by the second attribute
VehOdo	indicates the vehicles odometer reading
Nationality	refers to the manufacturer's country
TopThreeAmericanName	identifies if the manufacturer is one of the top three American manufacturers
PRIMEUNIT	identifies if the vehicle would have a higher demand than a standard purchase.
AUCGUART	refers to the level guarantee provided by auction for the vehicle (the value Green is less risky and Red the riskiest).
BYRNO	is the unique number assigned to the buyer that purchased the vehicle.
VNZIP1, VNST	are the zipcode and the state where the vehicle was purchased.
VehBCost	is the acquisition cost paid for the vehicle at time of purchase.
IsOnlineSale	identifies if the vehicle was originally purchased online.
WarrantyCost	is the warranty price (term=36month and millage=36K).

Table 2.2: *Description of the features*

2.2 Distribution of the variables and statistics

In this part, we analyze statistics and distributions of the features. The data has been visualized to get some intuitions about the characteristics of the relevant features both individually and between them. First, we discuss the distribution of the most relevant ones in detail, then, all the others.

IsBadBuy is a binary attribute that can assume the values 0 (good purchased, $87,7\%$) and 1 (bad purchased, $12,3\%$). As we can see in Fig 2.1, the attribute has a very unbalanced distribution. The same has been found for **Transmission** and **Nationality**. The first one can assume the values of AUTO and MANUAL and vehicles with automatic transmission are the 96.5% of the data set. The second one has the values AMERICAN, TOP LINE ASIAN, OTHER ASIAN and OTHER. The majority of the vehicles are American (83.6%), the Asian are 16.1% and the remaining 0.3% are European brands (OTHER).

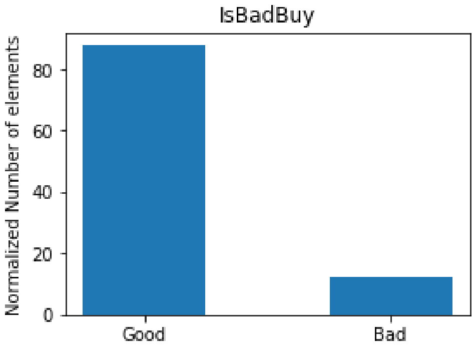
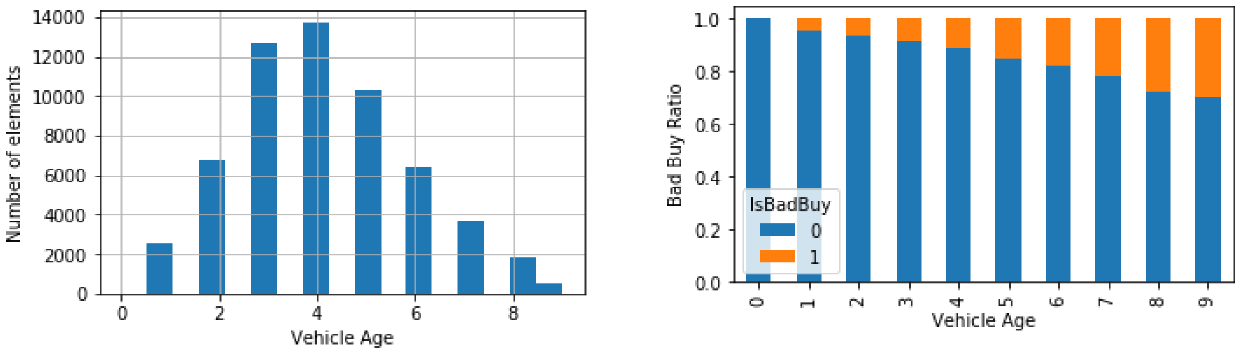


Figure 2.1: *Distribution of IsBadBuy.*

VehicleAge represents the age of the vehicles and varies from 1 to 9 years, where those of 4 years are the most frequent (23%). As we can see, the distribution is mainly Gaussian (Fig 2.2a). As expected, the percentage of incorrect purchases increases with the age of the vehicle (Fig 2.2b).

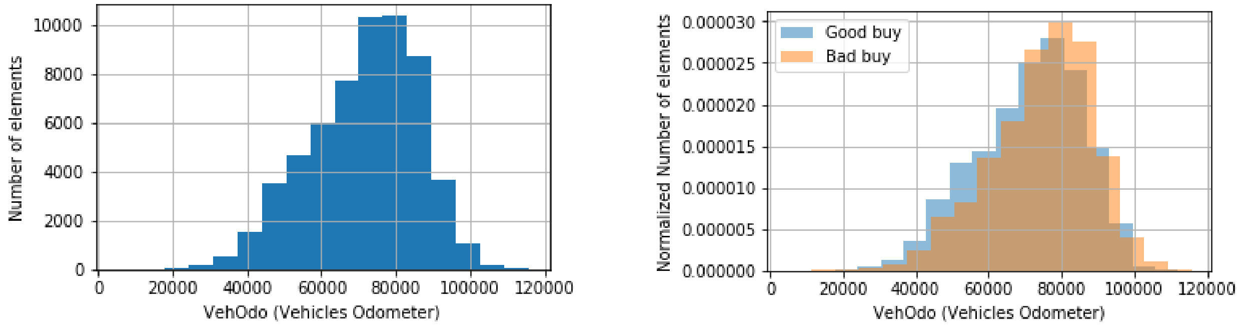


(a) *Density of elements per year.*

(b) *Vehicle age and bad buy ratio.*

Figure 2.2: *Distribution of VehicleAge*

VehOdo is a numerical attribute that can assume values between 4825 and 115717 (average mileage is 71024 km). Even if the distribution is slightly negative skewed (Fig. 2.3a), it reflects the information found in the VehAge's one, which means that high-mileage cars are the most frequent ones (old and middle ages cars). Linking this data to the purchase, we can see that, as easily assumed, cars with high mileage correspond more to a bad purchase (Fig. 2.3b).

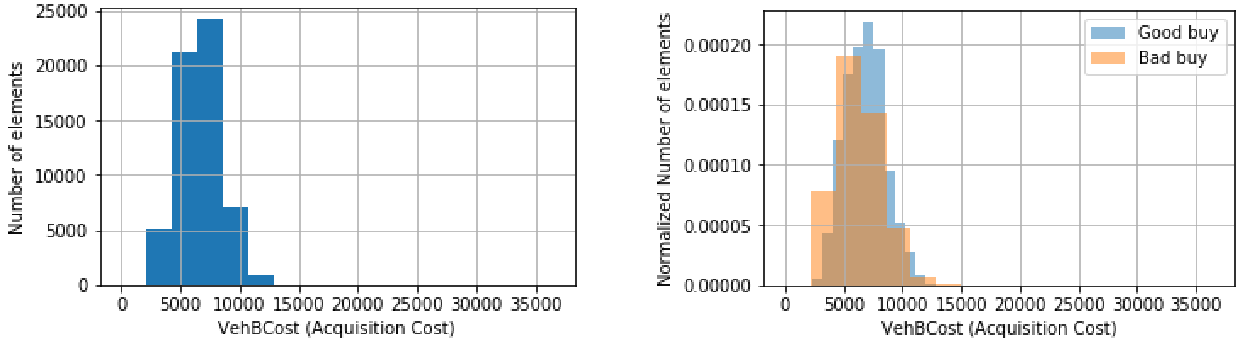


(a) *Histogram of VehOdo.*

(b) *Overlapping histograms for bad and good buy.*

Figure 2.3: *Distribution of VehOdo*

VehBCost is a numerical attribute that can assume values between 1$ and 36485$. It's unexpected that a car, even at auction, can be sold for $1, it's probably a wrong value. The positively skewed distribution gives us an indication of outliers (Fig. 2.4a). But it's important to note that as the cost of the vehicle increases, the possibility of it being a bad purchase decreases (Fig. 2.4b).

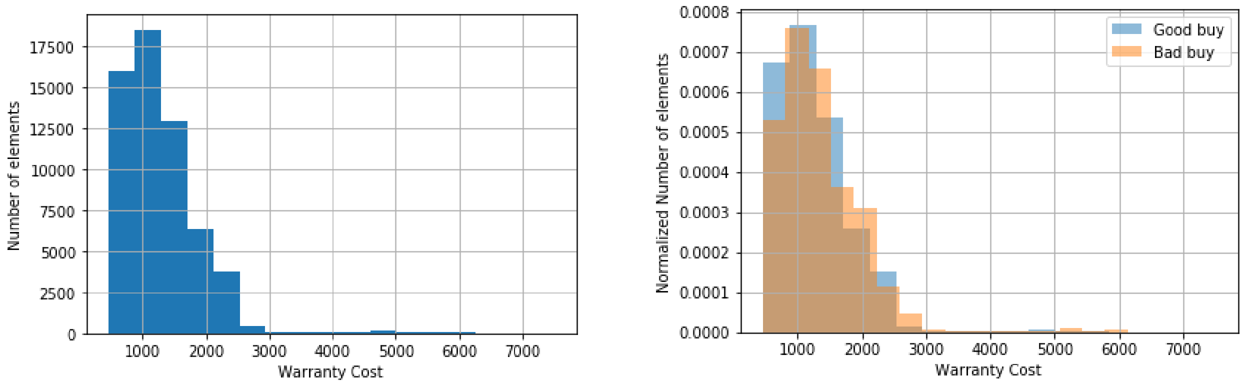


(a) *Histogram of VehBCost.*

(b) *Overlapping histograms for bad and good buy.*

Figure 2.4: *Distribution of VehBCost.*

Warranty Cost is a numerical attribute that can assume values between 462 and 7498. For most vehicles, the cost is between 1000 and 2000 and the distribution is positively skewed (Fig. 2.5a), which means that also in this feature there are some outliers. As far as the relationship with the purchase, we can see that as the Warranty Cost increases, the bad buys also increase. So, paying more for a warranty does not ensure a good buy. (Fig. 2.5b)



(a) *Histogram of Warranty Cost.*

(b) *Overlapping histograms for bad and good buy.*

Figure 2.5: *Distribution of Warranty Cost*

VNST is a categorical attribute that can assume 37 different values, each value is the abbreviation of an American state. As shown in Fig. 2.6, the market of used vehicles is more thriving in the south part of the USA. Indeed, the colour gradation reveals how many cars have been sold, for a range between 2.000 cars and more than 10.000. The state with the biggest number of purchased vehicles is Texas (18.6%) followed by Florida (14.2%).

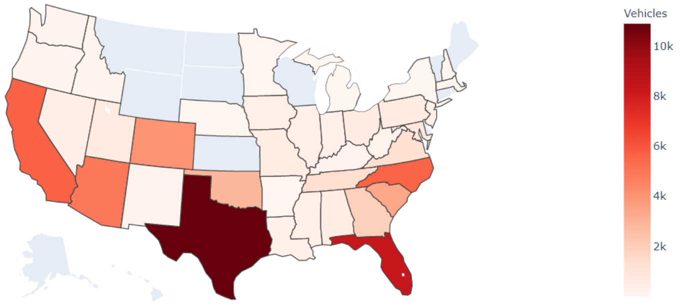


Figure 2.6: *This map reveals, with the colour gradation, how many cars have been sold in a specific state.*

Finally, in the cases of other features nothing special has been observed about distribution, but we can take into analysis the basic statistical data about the numerical features (see Tab. 2.3): mean, standard deviation, 25%, 50%, 75% percentiles, minimum and maximum values. Since we have already discussed some of them, we can focus here on MMR features. The average price of acquisition at auction is 6128.13$, while the average price current (at the time the dataset was released) is 6131.67$, so we can already see that on average there isn't a sharp rise in prices. What is interesting is the slight increase in the price of Clean, i.e. vehicles in good condition, which means that on average similar machines without damage and well maintained have a higher market price than when they were actually bought. Considering the minimum and maximum of all 8 MMR features, the presence of min equal to 0 or max over 35.000, if the average is much lower, make us understand as of now that we'll have to manage outliers or eliminate them, as we have seen before in `VehBCost`.

So, the previews features showed how unbalanced our dataset is and the other categorical features (see Tab. 2.4) reflect the information found above. Indeed, the most purchased car is American, has an automatic transmission and is a good buy, besides being sold by Manheim (the world’s largest wholesale auto auction), belonging to the Chevrolet’s car manufacturer, having 4 doors (Sedan), being medium-sized.

Feature name	Mean	Std	Min	Max	25%	50%	75%
MMRAAAP	6128.13	2456.63	0	35722	4273	6097	7765
MMRAACP	7372.91	2715.51	0	36859	5409	7308	9017
MMRARAP	8497.29	3151.11	0	39080	6279	8448	10652
MMRARCP	9851.77	3378.84	0	41482	7501	9798	12084
MMRCAAP	6131.67	2432.17	0	35722	4275	6063	7737
MMRCACP	7389.96	2682.31	0	36859	5415	7311	9014
MMRCRAP	8776.07	3086.37	0	39080	6538	8733	10910
MMRCRCP	10145.23	3304.64	0	41062	7788	10103	12309
VehBCost	6730.01	1762.08	1	36485	5430	6700	7900
VehYear	2005.34	1.73	2001	2010	2004	2005	2007
VehicleAge	4.18	1.713	0	9	3	4	5
VehOdo	71478.09	14591.23	4825	115717	61785	73359	82427
WarrantyCost	1276.11	598.89	462	7498	837	1155	1623

Table 2.3: *Statistics for numerical data with mean, Std, minimum, maximum values and percentiles (25%, 50%, 75%) which give an idea about the distribution of values.*

Feature name	Unique values	Most frequent	2° most freq.	3° most freq.
RefId	58386	-	-	-
IsBadBuy	2	0	-	-
PurchDate	517	11/23/2010	2/25/2009	10/13/2010
Auction	3	MANHEIM	OTHER	ADESA
Make	33	CHEVROLET	DODGE	FORD
Model	1029	PT CRUISER	IMPALA	TAURUS
SubModel	839	4D SEDAN	4D SEDAN LS	4D SEDAN SE
Trim	131	Bas	LS	SE
Color	16	SILVER	WHITE	BLUE
Transmission	3	AUTO	MANUAL	Manual
Size	12	MEDIUM	LARGE	MEDIUM SUV
WheelTypeID	4	1.0	2.0	3.0
WheelType	3	Alloy	Covers	Special
Nationality	4	AMERICAN	OTHER ASIAN	TOP LINE ASIAN
TopThreeAN	4	GM	CHRYSLER	FORD
PRIMEUNIT	2	NO	YES	-
AUCGUART	2	GREEN	RED	-
BYRNO	72	99761	18880	835
VNZIP1	152	32824	27542	75236
VNST	37	TX	FL	CA
IsOnlineSale	2	0	1	-

Table 2.4: *Statistics for categorical and boolean data that shows the number of unique values and the top 3 most frequent values for each attribute. With the symbol ’-’ has been specified’ where there is no a specific frequent value.*

2.3 Assessing data quality

In this part, we have checked the data for errors, missing values and outliers, which otherwise would affect the quality of data and could negatively influence the analysis.

2.3.1 Errors in the data

From the qualitative analysis of the attributes some errors and inaccuracies have emerged:

- The manufacturer (`Make` attribute) `TOYOTA SCION` doesn't exist on the market, Scion is the subbrand of Toyota's corporation, so the value has been manually corrected in `SCION`.
- A value of `Transmission` was written in lower case (`Manual`) while in the most cases was reported in uppercase (`MANUAL`). The formal error was corrected manually since there were few occurrences.

2.3.2 Missing values

The dataset has some missing values problems. We can see in Tab. 2.5 the number of them for each feature and the percentage concerning all the records. The analysis, based on the distribution of the missing values over the data set, led us to use different strategies for their management.

Attribute name	Missing values	%
`Trim`	1911	3.27
`SubModel`	7	0.01
`Color`	7	0.01
`Transmission`	8	0.01
`WheelTypeID`	2573	4.41
`WheelType`	2577	4.41
`Nationality`	4	0.00
`Size`	4	0.00
`TopThreeAmericanName`	4	0.00
`MMRAcquisitionAuctionAveragePrice`	13	0.02
`MMRAcquisitionAuctionCleanPrice`	13	0.02
`MMRAcquisitionRetailAveragePrice`	13	0.02
`MMRAcquisitonRetailCleanPrice`	13	0.02
`MMRCurrentAuctionAveragePrice`	245	0.42
`MMRCurrentAuctionCleanPrice`	245	0.42
`MMRCurrentRetailAveragePrice`	245	0.42
`MMRCurrentRetailCleanPrice`	245	0.42
`PRIMEUNIT`	55703	95.40
`AUCGUART`	55703	95.40

Table 2.5: *Features with missing values.*

- `Color`: has 7 missing values and 72 values equal to `NOT AVAIL`, we replaced them with the mode.
- `Transmission`: has been replaced with the mode of the feature.
- `WheelTypeID`: we have applied a recursively groupby substitution. We have grouped all the records by (`Make`, `Model`, `SubModel`) then accordingly and then we replaced the missing value of the attribute with the mode of each group.
- `Nationality`: to fill the missing values we searched on the internet the nationality of the brand (value of `Make`).
- `Size`: we have grouped all the records by `Make` and `Model` and then we replaced the missing value of this attribute with the mode of each group.
- `MMRs`: we grouped all the records by `Model` and then we replaced the missing values with the median of each group.

We will clarify below, in Subsection 2.6, the several reason why the untreated features with missing values were dropped.

2.3.3 Outliers

The outliers in the dataset were detected through the use of a BoxPlot (some are shown in Fig. 2.7), which highlights the values that do not fall within the interquartile range. In particular, we found outliers for attributes, `MMRAcquisitionAuctionAveragePrice`, `WarrantyCost`, `VehicleAge`, `VehOdo` and `VehBCost`. For all the records with `IsBadBuy` equals to `0` we calculated the low and high quartiles of the data distribution, and then we dropped every row where the value was lower or higher than the quartiles, then we did the same for `IsBadBuy` equals to `1`.

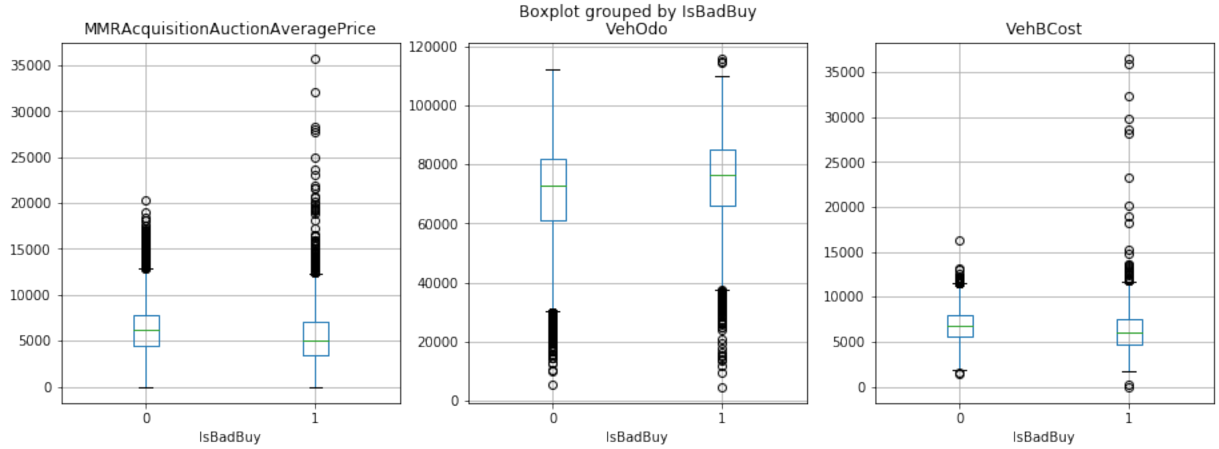


Figure 2.7: *Boxplots of some features.*

2.4 Features creation

`Model` and `SubModel` attributes contain a lot of technical information, but many are useless. So we decided to extract the most helpful information to create a new attribute: **BodyStyle**. This new categorical feature describes the body style of a vehicle and can assume values of `Sedan` $(58,1\%)$, `SUV` $(14,6\%)$, `Van` $(7,7\%)$, `Wagon` $(5,7\%)$, `Truck` $(5,4\%)$, `Crossover` (4%), `Coupe` $(3,6\%)$, `Convertible` $(0,7\%)$, `Hatchback` $(0,3\%)$ and `Jeep`. As we can see from Fig. 2.8, `Convertible` and `Coupe` are the body style with the highest percentage of bad buy.

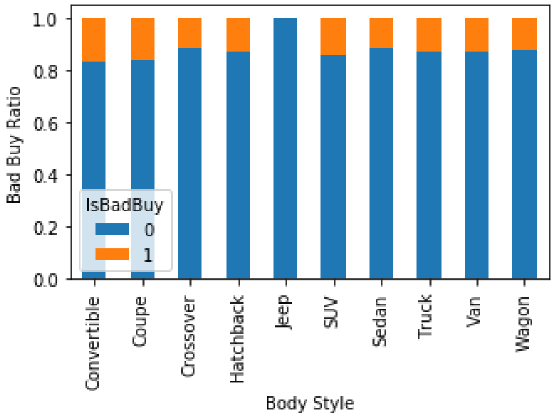


Figure 2.8: *Distribution of BodyStyle and Bad Buy Ratio.*

From the `Model` we also extracted the **ModelSeries** feature. The initial `Model` attribute can be described as car modification, the created `ModelSeries` is a generalization of foresaid (`IMPALA 3.5L V6 SFI F` → `IMPALA`). The `Model` attribute has 1029 different values, `ModelSeries` - 245. The most popular car by `Model` is Chrysler PT Cruiser (1845 entries), by `ModelSeries` - Chevrolet Impala (3879 entries).

2.5 Variables transformations

In order to compact the data set and simplify the analysis we have applied a label encoder to all the categorical features except for the following ones where the following pre-processing operations were performed:

- `Size`: We have compacted the values into three categories and then transformed the strings in numerical values: (`COMPACT`,`SPORT`)→`SMALL`→ `0`, (`CROSSOVER`, `MEDIUM SUV`, `SMALL SUV` and `SPECIALTY`)→ `MEDIUM`→ `1` and (`LARGE SUV`, `TRUCK` and `VAN`)→`LARGE`→ `2`.
- `Nationality`: conversion of `TOP LINE ASIAN` and `OTHER ASIAN` into `ASIAN` because the initial

values didn't contain additional information about nationality. Subsequently the values have been transformed into numerical values: `AMERICAN` → `0`, `ASIAN` → `1` and `OTHER` → `2`.

- `Transmission`: The strings have been transformed into numerical values: `AUTO` → `0` and `MANUAL` → `1`.
- `BodyStyle`: The strings also have been transformed into numerical values: `Sedan` → `0` and `SUV` → `1`, `Van` → `2`, `Wagon` → `3`, `Truck` → `4`, `Crossover` → `5`, `Coupe` → `6`, `Convertible` → `7`, `Hatchback` → `8` and `Jeep` → `9`.

2.6 Pairwise correlations and features selection

We computed the correlation matrix (Fig. 2.9) to have an understanding of the pairwise relations between features. From the graph above and from the all considerations presented before in this Section, we can draw the following conclusions:

- `RefId`: is a irrelevant information for data analysis.
- `VNZIP1`: is a redundant information because the features `VNZIP1` and `VNST` both describe the location of the auction where the car is bought, so we kept just `VNST` and dropped the zipcode.
- `VehYear` and `PurchDate`: are redundant information because we have `VehicleAge`.

$$\textit{VehicleAge} = \textit{PurchDate - VehYear}$$

- `TopThreeAmericanName`: is a redundant information because, we already have `Make` that identifies the manufacturer.
- `BYRNO`: is an irrelevant information for data analysis.

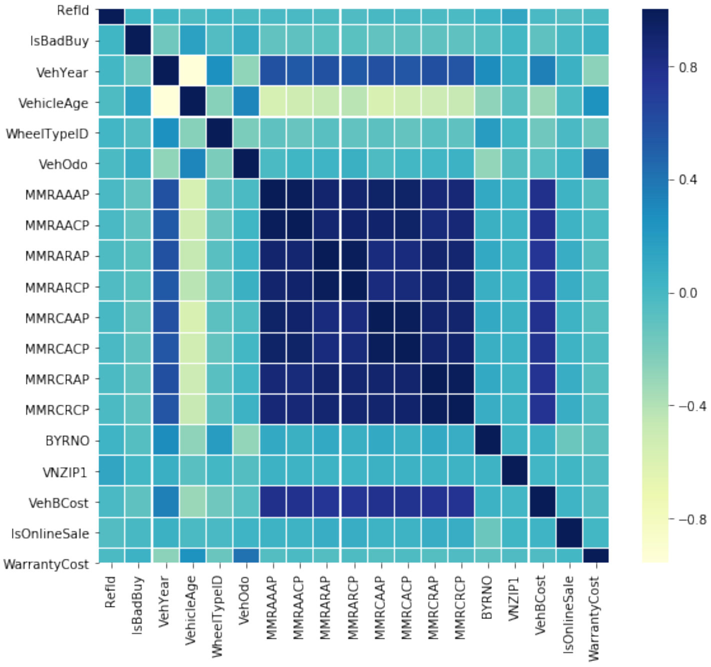


Figure 2.9: *Correlation Matrix.*

- `Auction`: unnecessary for analysis, which also presents 40% of the values set in `OTHER`.
- `SubModel`, `Model` and `Trim`: are superfluous and too detailed.
- `IsOnlineSale`: unnecessary for analysis.
- `WheelType`: is redundant because we already have `WheelTypeID`.
- `PRIMEUNIT` and `AUCGUART`: mostly have null values, only 4% of the values are assigned, so they must be removed.
- We saw that all the MMRs variables are strictly correlated, so we decided to remove 7 of them and leave only `MMRAcquisitionAuctionAveragePrice` because it's the market's cost at the moment of the purchase and we believed that is the most relevant aspect for our aim.

After Data Cleaning phase we get 55495 entries and 13 original features: **IsBadBuy, VehicleAge, Make, Color, Transmission, WheelTypeID, VehOdo, Nationality, Size, MMRAcquisitionAuctionAveragePrice, VNST, VehBCost, WarrantyCost** - and 2 new created features: **BodyStyle** and **ModelSeries**. So 15 features in total.

3. Clustering

In this section of the project, we work with 3 variables that represent the financial aspects of the vehicle: typical market cost (`MMRAcquisitionAuctionAveragePrice`), actual vehicle cost (`VehBCost`) and warranty cost (`WarrantyCost`). The purpose was to examine if among the financial aspects exists some kind of internal grouping.
To find the best grouping for our dataset, we tried three algorithms that use different cluster models: K-means (centroid model), DBSCAN (density model), and Hierarchical clustering (connectivity model). For each algorithm, we first searched for the best model parameters; after that,we have evaluated the model quality. For all the clustering algorithms, the *standard scaling* and *euclidean distance* have been used, respectively, as normalization method and as evaluation method of the distance matrix.

3.1 K-Means

3.1.1 Best K value

In the K-means algorithm, we need to choose only one hyperparameter: *K* - used to specify the number of clusters to be identified. To take the best value of *K*, we plotted the approximate SSE value for the different number of clusters (from 2 to 50) and chose the best range of K, using the *knee method*. Analyzing Fig. 3.1, we saw that the knee point is located near 5, so as possible K-s we considered the values between 4 and 6. To select the best one from this interval, we used the mean silhouette measure for different K-s. The results were: 0,31 for k=4, 0,28 for k=5 and 0,29 for k=6 - so the biggest separation distance between the clusters was with K=4.

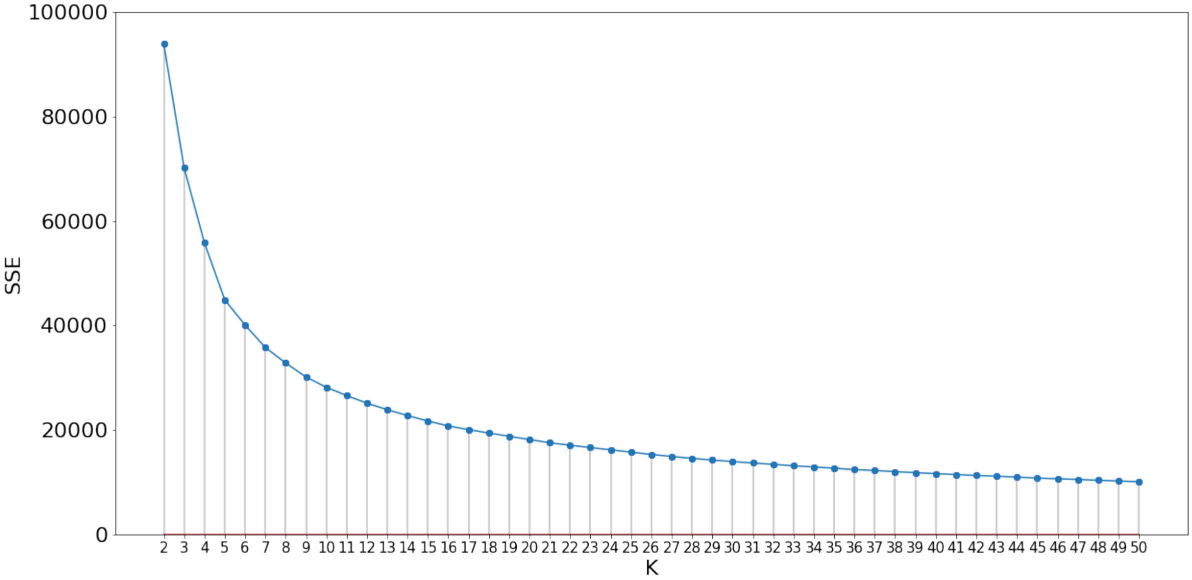
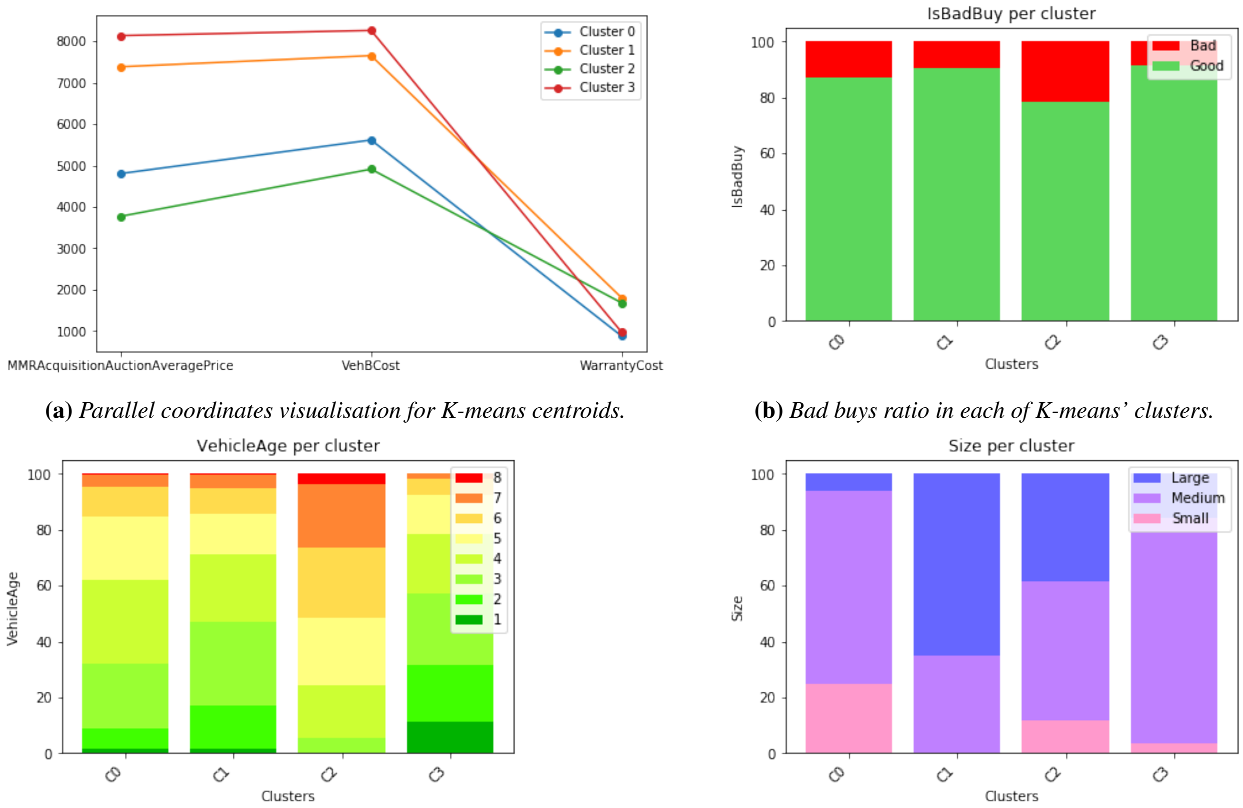


Figure 3.1: *SSE versus number of clusters (K) for the data. It was used to identify the right number of clusters via the knee method.*

3.1.2 Clusters analysis

From the study of the cluster interactions, we can see that for each attribute the centroids are well-separated into two groups: two clusters with higher values, two with smaller ones (Fig. 3.2a). Interestingly, each of these `WarrantyCost` groups is composed of one cluster with higher vehicle cost and one with a smaller one.

Each cluster contains a significant amount of bad buys (Fig. 3.2b), but the third cluster contains the highest ratio. This indicates that the cluster analysis can't be used as an efficient predictive model of the target attribute.



(a) *Parallel coordinates visualisation for K-means centroids.*

(b) *Bad buys ratio in each of K-means' clusters.*

(c) *Vehicle Age ratio in each of K-means' clusters.*

(d) *Vehicles' size ratio in each of K-means' clusters.*

Figure 3.2: *Distribution of the variables inside the K-means clusters, used to analyse their semantic properties.*

Based on the analysis of variables' distribution inside the clusters (see Fig. 3.2), we concluded that each cluster identifies different car categories with the distinguishing properties. **Cluster 2** contains 8680 cars (smallest cluster) and corresponds to the oldest cars with low cost, high warranty and with an odometer average of 67139. This means that old cars cost less but their warranty cost with respect to all others is higher, because in the case of any breakage the maintenance cost will be enormous, consequently this class has the major number of bad buys. **Cluster 3** contains 16090 cars and corresponds to bigger cost, smaller warranty, the lowest average age, an odometer average of 79233. So, relatively new cars usually require less maintenance, therefore they initially cost is bigger but warranty is less expensive, and potentially this car class must have the least amount of bad buys. The remaining two classes represent middle-aged cars. All else being equal, the smaller cars cost and warranty are usually smaller than the larger cars ones: **Cluster 0** contains 18032 cars (the biggest cluster) that are cheap, low warranty, small and medium sized vehicles with an odometer average of 66789; **Cluster 1** contains 11667 cars that are more expensive, high warranty, medium and large sized with an odometer average of 77969.

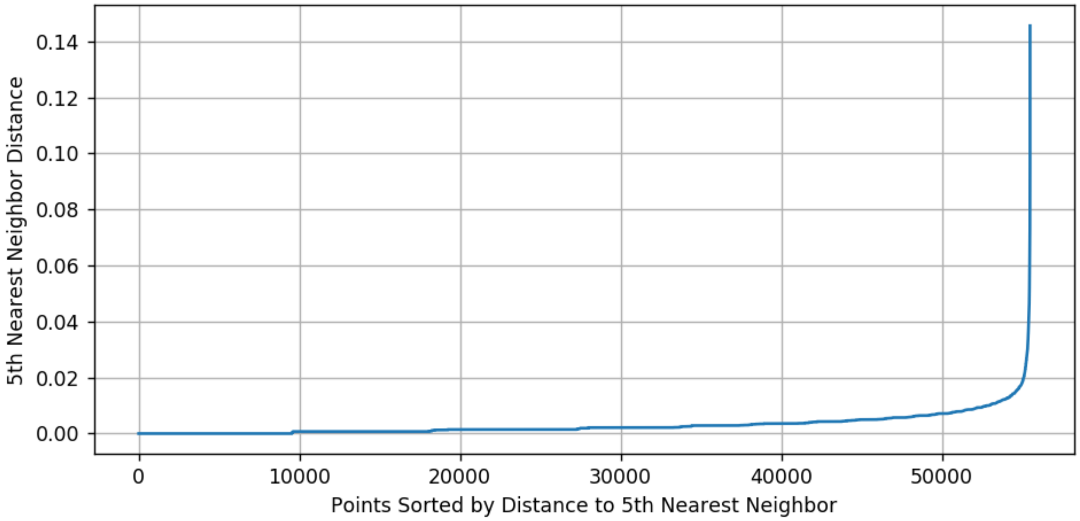


Figure 3.3: *Distance from 5th neighbour versus points sorted by distance to 5th nearest neighbor, used to identify the right number of Eps.*

3.2 DBSCAN

3.2.1 Clustering parameters

To run DBSCAN, we need to choose the values of two parameters: density measure (Eps) and a specified number of points (MinPts) within Eps. To select the best value of Eps, we used the knee method, plotting, for MinPts=5, the sorted distances from the 5th point (Fig. 3.3). From the plot, we pointed out that the best Eps must be between 0.12 and 0.20. Then, we took 9 points within this interval, and for each of these points, we searched for the best value of MinPts (considered MinPts were in the range between 2 and 50). The best silhouette score was with Eps=0.20 and MinPts=16.

3.2.2 Clusters analysis

The results of the clustering can be seen in Fig. 3.4. The clustering, which we received using this configuration, gave us 5 clusters (plus the noise points, the dark violet points). The clusters' sizes are **51503**, **139**, **20**, **33**, and **18**; there are **2756** noise points. The clustering is very unbalanced as almost all of the points belong to the same cluster. In other words we can say that the algorithm returns a single large cluster and for this reason DBSCAN is not considered efficient.

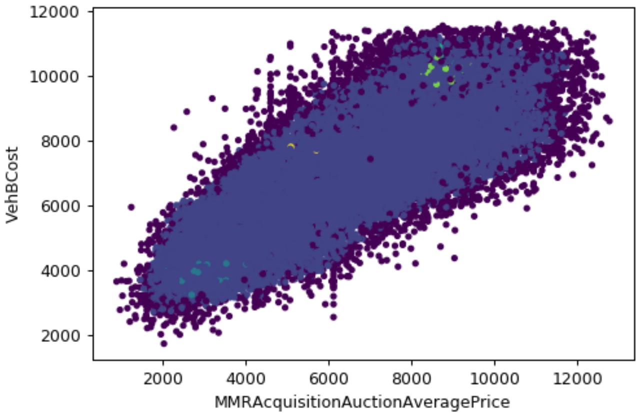
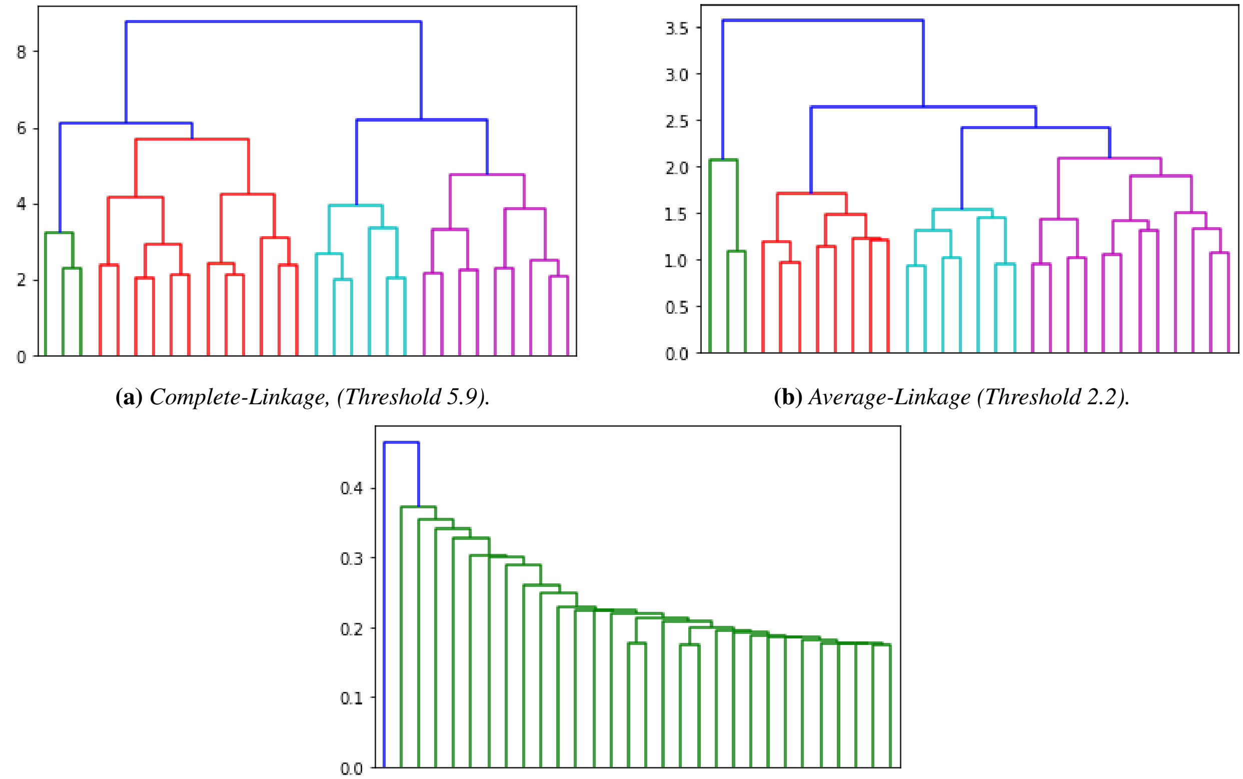


Figure 3.4: *DBSCAN Result.*

3.3 Hierarchical

For the Hierarchical clustering, we need to determine two hyperparameters - connection criteria and dissimilarity threshold. We tried the algorithm with different connection criteria (*single*, *complete*, *average*) and for each result examined different threshold to get a balanced number of clusters. We can see (Fig. 3.5c) that using the Single as connection criteria, we get the results similar to DBSCAN with a threshold equal to 0.4: one huge cluster that contains all the entries in the database - which can't possibly give as a new perspective on the hidden data structure.



(a) *Complete-Linkage, (Threshold 5.9).*

(b) *Average-Linkage (Threshold 2.2).*

(c) *Single-Linkage (Threshold 0.4).*

Figure 3.5: *Hierarchical clustering methods.*

Complete and Average methods with threshold respectively at 5.9 and 0.4, on the other hand, gave us clusters of more balanced size (Tab. 3.1). During the careful study of Complete and Average methods' results (Fig. 3.5a and Fig. 3.5b, respectively), we concluded that both of these clusterings also represent the same car categories, which we discussed in Section 3.1.2, that are mostly based on the age of the vehicle and its current state.

Method	Clusters size
Complete	15817, 14707, 9654, 15316
Average	719, 23918, 9458, 21399
Single	55493, 1

Table 3.1: *Size of the clusters produced by different connection criteria.*

Both of these algorithms did a really good job of finding the semantic class, that was previously called - old cars. Average gave us the cluster of only very old cars (only cars older than 5 years, with low medium vehicle cost (2136 on average) and high warranty (mean - 1597)), but a lot of old cars were left out (the size of this cluster is only 719 vehicles) and ended up in other clusters. 98% of the cars in this cluster were also presented in the K-means cluster which we assigned as old cars, but these cars only compose 8% of this K-means cluster. Complete, on the other hand, have a lot of vehicles in the older

cars cluster (9654 entries), but a lot of them, in our definition, might not be so old (still the oldest class with higher warranty cost, but the low vehicle cost or the biggest number of bad buys, which is usually associated with older cars, is not so articulated even though it is still present). The concordance of these clusters with respect to the K-means older car-cluster is around 80%.
Considering the new car's cluster, both of the algorithms have a lot of vehicles in attributed clusters, but the limits of new car definition are not strictly respected (21399 cars in this class for Average and 15187 for Complete) - the cars that in both algorithms (K-means and one of the Hierarchical) were assigned with the newer cars cluster form approximately 45% for Complete and 60% for Average.
The fact that for both of these algorithms the most reasonable number of clusters, which represented almost the same semantic categories, was also 4, just proved our judgment about the optimal number of K in the K-means algorithm.

3.4 Clustering comparison

Between all the clustering techniques we have presented in this chapter, K-Means emerges victorious, since it has generated the most significant clusters. The Hierarchical clustering, with *complete* and *average*, confirmed the results found with the K-Means, instead DBSCAN turned out to be inefficient for this dataset.

4. Association Rules Mining

In this section, we would like to discuss the results and consequences obtained during the association rules mining phase. The first step was data preparation, then we extracted the most frequent patterns, after that we studied obtained association rules. The goal was to apply derived association rules to resolve the missing value problem and to construct the prediction model of the target class.

4.1 Preliminary data preparation

Before making any special data preparation for the association rule mining, we followed all the data cleaning steps discussed in the Data Understanding section. Except one! We didn't substitute the missing values, because their replacement was one of the goals for the rules extraction.
To successfully apply the pattern extraction algorithms we should transform our quantitative features into categorical ones. `VehicleAge` is almost categorical variable (it has only 8 possible values), but all the others have thousands of possible unique values, so we replaced the actual value with the interval to which it belongs. The intervals has been achieved by clustering these 4 variables: `VehOdo`, `WarrantyCost`, `VehBCost` and `MMRAcquisitionAuctionAveragePrice`. You can see the results in the Tab. 4.1.

VehBCost	VehOdo	MMRAAAP	WarrantyCost
[1720; 3815]	[30212; 45443]	[884; 3619]	[462; 728]
[3820; 5745]	[45449; 61627]	[3620; 6609]	[754; 1223]
[5750; 7450]	[61630; 71437]	[6610; 10416]	[1241; 1808]
[7455; 9815]	[71439; 91679]	[10417; 12951]	[1857; 2282]
[9820; 11645]	[91683; 112029]	-	[2322; 2838]

Table 4.1: *Interval distribution for numerical data.*

As the method of the clustering we chose hierarchical one. In this particular case we saw that hierarchical clustering gives us clusters that have ranges of almost equal length, but the size of the clusters are not the same. On the other hand, K-means give us clusters of the same size but the range differ in length a lot.

We thought that in real life the car clusters don't have the same number of options (there is a lot of cars in medium range and only few super expensive or super cheap ones), but people start their search from the amount of money that they have, so the key factor should be the range, not the size of the cluster.

4.2 Frequent patterns

The Apriori algorithm has been applied to this prepared data with different support values and different frequent item sets' types: frequent, closed and maximal.
In the Tab. 4.2 and Tab. 4.3 you can see different frequents patterns with different support levels (frequent and closed gave us the same results for these support levels, so both of them are reported in the same column). We didn't insert the patterns with smaller support only because, since the number of patterns grows with the decrease of support, the item sets started to get too large to be presented in this general overview. Even the patterns with low support can be of interest, so for the association rules mining we chose the threshold of 10% for the support. As you can see, by analysing frequent patterns with a high level of support we got only some general remarks that we already knew, such as that the dataset mostly consists of American medium sized cars with automatic transmission, and that it is usually profitable to buy such cars. Almost 43% of the cars correspond to all these features listed above.

Support	**Frequent - Closed**
90% - 100%	**1)** {Transmission: Auto}, (*supp=0.97*)
80% - 90%	**2)** {Good Buy}, (*supp=0.88*) **3)** {Good Buy, Transmission: Auto}, (*supp=0.85*) **4)** {Nationality: American}, (*supp=0.84*) **5)** {Nationality: American, Transmission: Auto}, (*supp=0.81*)
70% - 80%	**6)** {Nationality: American, Good Buy}, (*supp=0.74*) **7)** {Nationality: American, Good Buy, Transmission: Auto}, (*supp=0.72*)
60% - 70%	**8)** {Size: Medium}, (*supp=0.62*) **9)** {Size: Medium, Transmission: Auto}, (*supp=0.60*)

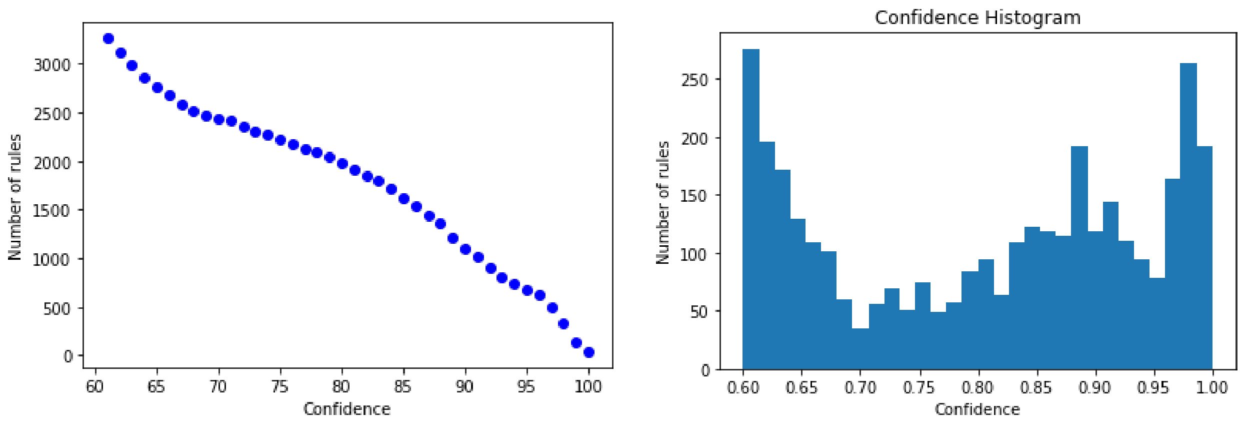
Table 4.2: *Frequent patterns with high support for frequent and closed itemset.*

Support	**Maximal**
90% - 100%	**1)** {Transmission: Auto}, (*supp=0.97*)
80% - 90%	**2)** {Good Buy, Transmission: Auto}, (*supp=0.85*) **3)** {Nationality: American, Transmission: Auto}, (*supp=0.81*)
70% - 80%	**4)** {Nationality: American, Good Buy, Transmission: Auto}, (*supp=0.72*)
60% - 70%	**5)** {Size: Medium, Transmission: Auto}, (*supp=0.60*)

Table 4.3: *Frequent patterns with high support for maximal itemset.*

4.3 Association rules

We extracted the association rules using 60% as confidence threshold, since the confidence indicates how often the rule is true, we are not interested in rules that are true less than 60% of the time. Using selected support and confidence thresholds we got 3495 rules but, as both Fig. 4.1a and 4.1b show, we have too many rules even with a really high confidence to report all of them here.

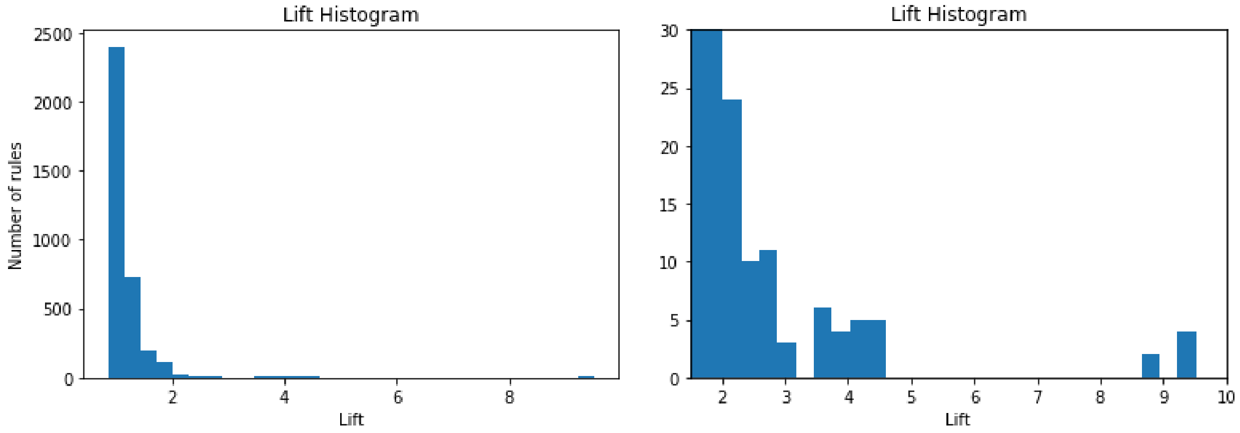


(a) *Rules variation with reference to the confidence.*

(b) *The distribution of rules' confidence.*

Figure 4.1: *Dependency of the number of rules from confidence.*

Furthermore, the distribution of lift is uneven: lots of rules have a lift of approximately 1 (Fig. 4.2a), which means that they are just casual co-occurrences; and a small number of significant rules, whose distribution of lift you can see more in detail on the second plot (Fig. 4.2b).



(a) *Histogram of the lift distribution, x-axis the lift and on the y-axis the number of rules (between 0 and 2500).*

(b) *A zoomed lift distribution, x-axis the lift and on the y-axis the number of rules (between 0 and 30).*

Figure 4.2: *Dependency of the number of rules from lift.*

From the 3495 initial rules we have selected the most significant ones, not only by using lift but taking into account our aim, namely to find as many useful information as possibles. We know indeed that the confidence of rules from the same itemset has an anti-monotone property, so we list below only 10 rules generated from different itemsets, sorted by lift.

- {Make: Chevrolet, Size: Large, Nationality: American, Transmission: Auto} ⟹{ModelSeries: Impala}, conf = 0.66 and lift = 9.51
- {Size: Large, BodyStyle: Sedan, Nationality: American, Good Buy, Transmission: Auto} ⟹ {ModelSeries: Impala}, conf =0.60, lift = 8.68
- {WarrantyCost: [462; 728], BodyStyle: Sedan, Good Buy, Transmission: Auto} ⟹{Nationality: Asian}, conf = 0.73 lift = 4.52

- {WarrantyCost: [462; 728], Size: Medium} $\Longrightarrow${Nationality: Asian}, conf = 0.70 lift = 4.33
- {WarrantyCost: [462; 728], WheelTypeID: 2.0, Good Buy} $\Longrightarrow${Nationality: Asian}, conf = 0.62, lift = 3.84
- {WarrantyCost: [1857; 2282], Nationality: American, Good Buy, Transmission: Auto} $\Longrightarrow${Size: Large}, conf = 0.79 lift = 2.93
- {MMR: [884; 3619], BodyStyle: Sedan} $\Longrightarrow${VehBCost: [3820; 5745]}, conf = 0.76 lift = 2.80
- {MMR: [884; 3619], Good Buy} $\Longrightarrow${VehBCost: [3820; 5745]}, conf = 0.76 lift = 2.78
- {MMR: [884; 3619], Nationality: American, Transmission: Auto} $\Longrightarrow${VehBCost: [3820; 5745]}, conf = 0.75 lift = 2.75
- {WarrantyCost: [1857; 2282], Nationality: American, Good Buy, Transmission: Auto} $\Longrightarrow${Make: Chevrolet}, conf = 0.61 lift = 2.56

All these rules confirmed that the 2 new features - ModelSeries and BodyStyle - has proved themselves useful.

The most popular large American sedan is Chevrolet Impala (60% of the market). Medium cars or sedans with low warranty costs with the probability of 70% are Asian. Instead, a good American car with auto transmission whose warranty cost is high (in the 4th interval from 5) is large in 79% of the cases and in 61% it is Chevrolet. Also, we can see that a buyer in 75% of cases pays more than estimate MMR cost for the sedans, American cars and good cars in general.

4.4 Missing values replacement

The features that have missing values are: Color, Transmission, WheelTypeID, Nationality and Size.

For **Color**, we had 79 missing values but we couldn't extract the rules with confidence bigger than 60%. The first rule considering Color that we found had the confidence of 15%. The conclusion is that association rules can't help us replace the Color's missing values.

For **Transmission**, we had 7 missing values and we found rules for each case, and according to the rules we replaced missing values with {Transmission:Auto} with a confidence higher than 98%.

For **WheelTypeID**, we had 2374 missing values and for 73 of them we didn't found rules. All the other has been substituted with {WheelTypeID: 1.0} or {WheelTypeID: 2.0}, with a confidence between 60% and 75%.

For **Nationality**, we had 4 missing values and according to the rules we replaced them all with {Nationality: American}, with a confidence bigger than 95%.

For **Size**, we also had 4 missing values and we substituted all of them with {Size: Medium}, with a confidence between 67% and 88%.

4.5 Target variable prediction

We used our rules to construct the target variable prediction model. The model's accuracy is 87%. On the first gaze, it seems to be a good result but actually the model always assigns the same class, the Good Buy, to all the entries; and we have such accuracy only because of the high unbalance of the data.

Moreover, this model doesn't yield Bad Buy in any potential case: it doesn't have not even one rule that says that the result must be Bad Buy. We have to assume a threshold of confidence as 20 to find at least one Bad Buy rule, which means that the rule's accuracy will be approximately 20% (obviously too small).

5. Classification

In this Section, we have applied the classification techniques in order to predict if the car purchased was a bad buy or not. This kind of task requires a supervised classification model and we decided to use the *Decision Tree*, that is commonly drawn using flowchart symbols as it is easier for many to read and understand.
The Carvana company provided us two data set: Training and Test set. As we have already seen from the previous analyses carried out in Subsection 2.2, the training set is highly unbalanced with respect to the target attribute object of the classification (`IsBadBuy`), therefore to try to solve this problem we used two different techniques:

- **Undersampling**: under-samples the majority class randomly and uniformly.
- **Oversampling**: in this case the Synthetic Minority Oversampling Technique (SMOTE) was performed, SMOTE allows to create synthetic observations of the minority class by finding the k-nearest-neighbors for minority class observations (finding similar observations) and then randomly choosing one of the k-nearest-neighbors and using it to create a similar, but randomly tweaked, new observation.

these two techniques decrease/increase the overall data set in order to allow us to get a balanced data set.

5.1 Model Selection

The goal of this step is to select a set of candidate models for the given data and then choose the best model that is able to generalize the data for our task. First, we have decided on a validation scheme, then we performed several tests with the intention to select the relevant features useful for the decision tree and finally the model selection phase with a random grid search for the hyperparameters optimization.

Validation Schema
First, we divided the data set (original training set) into random training (70% of data) and validation (30% of data) subsets by using a stratified splitting that allowed us to preserve the percentage of samples for each class. From now on we will refer to these sets as external training and validation sets. This external validation set has not been used initially but its purpose would be to compare the models found with oversampling and undersampling techniques. Then, for the model selection, we performed a random grid search with 3-folds stratified cross-validation on external training set. The result of 3-folds is an internal set of training and validation. The Undersampling and Oversampling techniques will be performed only on the internal training set and leave the internal validation set untouched (this is useful for the oversampling because if we upsample a dataset before splitting it into a training and validation set, we could end up with the same observation in both sets).

Relevant Feature
Now, we wanted to find out if there are features that are not relevant to those already selected in Section 2. Therefore, we trained some Decision Tree, with undersampling and oversampling methods, and we observed which features were most relevant for the classification task. Turned out that `Make` and `Colors` are not relevant and for this reason we have not considered them. After that, we wanted to check if some of the eliminated `MMRs` could be useful for classification and some tests have validated this intuition. The most significant features are:

- **Categorical**: `Auction`, `Size`, `VNST`, `WheelTypeID`
- **Numerical**: `VehicleAge`, `VehOdo`, `WarrantyCost`, `VehBCost`, `MMRAcquisitionAuctionAveragePrice`, `MMRCurrentRetailAveragePrice`, `MMRAcquisitionRetailAveragePrice`, `MMRCurrentAuctionAveragePrice`

Hyperparameters Optimization

The hyperparameters optimization phase was carried out by combining the validation scheme, described above, and the range of parameters in the Tab. 5.1 for the random grid search. As goodness criterion we decided to use the value of *ROC AUC* because the accuracy alone is not a good evaluation option with class-imbalanced data sets. Tab. 5.2 shows the best results found with the undersampling and oversampling techniques. So, in order to compare all five models found, we trained them with the external training set (using the associated sampling technique) and compared them with the external validation set (with a class balance similar to the original dataset). Tab. 5.3 shows the results of the comparison. Finally, we have chosen `Model 1` as final model that has the best `ROC AUC` value on validation and Fig. 5.1 shows the decision tree obtained from it.

Random Grid Search		
Hyperparameters	**Undersampling**	**Oversampling**
criterion	gini, entropy	gini, entropy
max_depth	None, 2 - 20	None, 2 - 20
min_sample_split	2 - 40	2 - 40
min_sample_leaf	2 - 40	2 - 40
min_impurity_decrease	0.	[0.75e-6, 0.5e-6, 1e-6]

Table 5.1: *Hyper-parameters for the random grid search.*

Best Models				
	Model 1	**Model 2**	**Model 3**	**Model 4**
Sampling	Undersam.	Undersam.	Oversam.	Oversam.
criterion	gini	entropy	gini	gini
max_depth	5	5	8	8
min_sample_split	26	22	29	13
min_sample_leaf	25	31	9	3
min_impurity_decrease	0	0	1e-06	5e-07
TR roc_auc	0.678±0.004	0.676±0.002	0.679±0.003	0.679±0.003
TR Accuracy	0.565±0.034	0.588±0.017	0.873±0.003	0.873±0.003
TR F1 Score	0.282±0.002	0.281±0.001	0.208±0.007	0.210±0.008
VL roc_auc	0.668±0.006	0.668±0.003	0.658±0.007	0.658±0.007
VL Accuracy	0.563±0.033	0.585±0.025	0.871± 0.001	0.871± 0.001
VL F1 Score	0.278±0.008	0.278±0.000	0.200±0.009	0.200±0.009

Table 5.2: *Hyper-parameters for the random grid search.*

Scoring on the common Valdiation set						
Model	**TR - ROC auc**	**TR - Accuracy**	**TR - F1**	**VL - ROC auc**	**VL - Accuracy**	**VL - F1**
Model 1	0.629	0.629	0.645	0.626	0.588	0.284
Model 2	0.628	0.628	0.645	0.625	0.585	0.282
Model 3	0.882	0.882	0.871	0.561	0.871	0.223
Model 4	0.882	0.882	0.871	0.561	0.870	0.223

Table 5.3: *Best model for undersampling vs oversampling.*

We can observe some aspects on the results (see table 5.3). The undersampling dominates the oversampling technique on the results but models 1 and 2 seem to be subject to the underfitting problem. Regarding models 3 and 4 the accuracy is very high on the validation set but the F1 score and the ROC AUC remain very low. This means that those models have learned very well to classify a good buy but they are inadequate to determine whether a vehicle could be a bad buy (this confirms that accuracy alone is not a good metric for unbalanced data sets). Moreover, accuracy, F1 score and ROC AUC on the training set are very high compared to the validation set. This could indicate overfitting using the oversampling technique.

5.2 Model Assessment

After choosing the final model we proceeded to estimate the prediction error on the test set provided by Carvana to measure the quality of the chosen model. The result is shown in the Tab. 5.4, the confusion matrix and ROC Curve are reported, respectively, in Fig. 5.2a and in Fig. 5.2b

	ROC AUC score	Accuracy	F1 Score	Precision	Recall
Model 1	0.6105	0.567	0.272	0.17	0.67

Table 5.4: *Decision tree evaluation.*

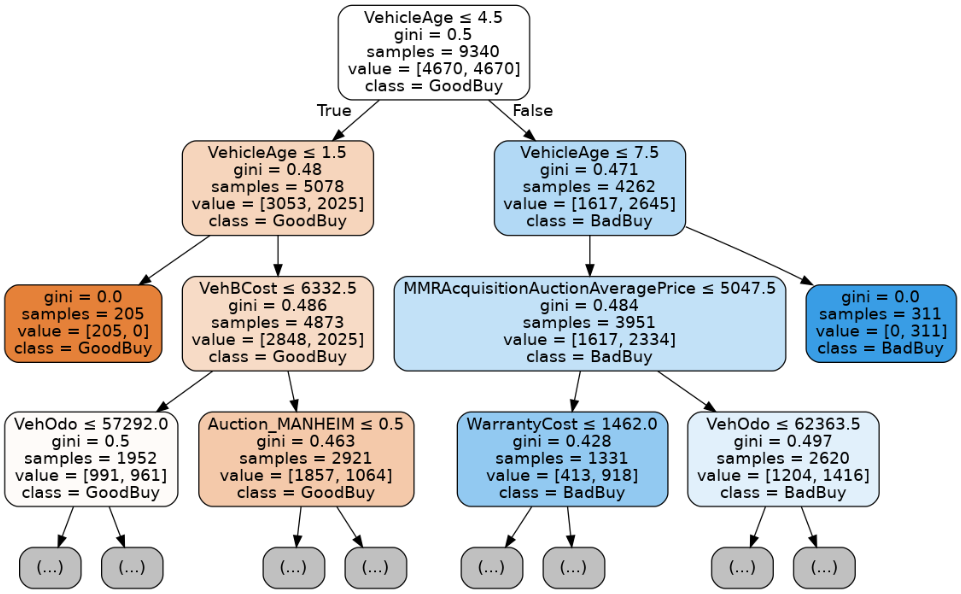
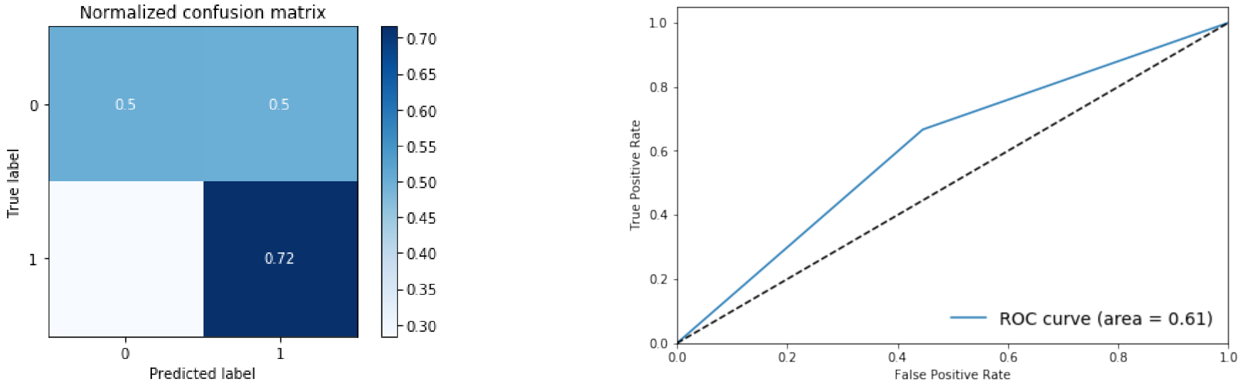


Figure 5.1: *Decision Tree obtained from model selection and a cut at depth 3.*



(a) *Normalized Confusion Matrix.*

(b) *ROC Curve.*

Figure 5.2: *Confusion Matrix and ROC curve evaluated for the final Decision Tree model on the Test Set.*

5.3 Classification results

We presented two different techniques for the management of the unbalanced dataset: oversampling and undersampling. The first provided some overfitting models, the second, that has better results, is subject to underfitting. Indeed, the area value obtained under the roc curve is not a good value (0.61). Therefore the model obtained will not lead to a good classifier and cannot be used by the company to predict whether a car will be a good or bad purchase but it's interesting to note that the most significant features (see Fig. 5.1) for making a classification are: `MMRAcquisitionAuctionAveragePrice`, `VehicleAge`, `VehBCost`, `VehOdo`, `WarrantyCost` and `Auction`. From the interpretation of the decision tree we can only derive these results (which may not reflect reality):

- bad buy could be cars with age more then 7.5 and good buy could be cars with age less then 1.5 will be classified as a good buy;
- by cutting the decision tree at level 2 we could see that a good buy could be cars with age more then 1.5, age less then 4.5 and cost less then 6332.5
- by cutting the decision tree at depth 2 we could see that a bad buy could be cars with age more then 4.5, age less then 7.5 and an acquisition auction average price less then 5047.5

other indicative classification rules could be found by visiting the tree to its maximum depth (5).

6. Conclusions

In summary:

- The data understanding and pre-processing phase has been very challenging, because of the features imbalance and the presence of missing values. Moreover, as we are not experts in cars, we had to inform ourselves in order to understand the detailed values of our features. We went from 34 initial variables to 13 variables and 2 variables created by us. This work allowed us to understand from the beginning that our dataset is very unbalanced and that any subsequent analysis should have taken this knowledge into account.

- As far as clustering is concerned, the K-means algorithm proved to be the best and its result has been confirmed by the Hierarchical one. Instead DBSCAN turned out to be inefficient for this dataset.

- The pattern mining phase allowed us to find rules that would help us in replacing missing values with a certain confidence, for some features very high around 98%, for others between 60% and 80%. Our predict model has achieved 87% of accuracy, even if we have such accuracy only because of the high unbalance of the data.

- Finally, concerning the classification unfortunately the model found using the Decision Tree, fails to give us precise and safe answers.

So, as regards future work: in order to improve the results found, it might be of interest applying additional classification models (random forest, naive bayes, knn, neural network, SVM) and understand the performances of these models on the provided dataset. We could then understand if the limits reached in the results of the classification are mainly related to the structure of the dataset (so it requires to collect additional records and better balance the data) or if the model used in this report is not efficient.