Piyush Tada
Sharath Chandra
Marta Fioravanti
Irene Parlanti

2019 - 2020
Data Mining
Pedreschi - Monreale
Università di Pisa

Index

Index **1**

1. Data understanding **2**
1.1. Data semantics 2
1.2. Distribution of the variables and statistics 2
1.3. Assessing data quality 4
1.4. Variables transformations 4
1.5. Pairwise correlations and eventual elimination of redundant variables 5

2. Clustering **6**
2.1. K-means 12
2.1.1. Best value of k 14
2.2. DBSCAN 14
2.3. Hierarchical 14
Conclusions 15

3. Association Rules Mining **15**
3.1. Frequent patterns extraction 17
3.2. Association Rules extraction 18

4. Classification **20**
4.1. Features selection 20
4.2. Decision Trees Learning 21
4.3. Decision Tree Interpretation 21
4.4. Decision trees validation with test and training set 21
4.5. Discussion of the best prediction model 22

1. Data understanding

1.1. Data semantics

“Carvana dataset” is a repository of used vehicles. Each car is provided with many information concerning its standard equipment, current characteristics and price, among others. In total we are provided with 32 attributes/features. We took some time to understand each attribute and listed in this report. This is the most important step to acquire the best result. So, all of us worked on the Data Understanding part and tried in their own approach. Finally we sorted among our understanding and obtained final Dataset.
After we agreed with list of attributes, some are listed below in Table 1.

Table 1: Attributes distribution

Type	Attributes
Binary	IsBadBuy,Transmission, WheelTypeID,IsOnlineSale.
Categorical	Auction, VehYear,Age, Make, model, nationality.
Continuous	all MMR Acquisition, Current attributes.

1.2. Distribution of the variables and statistics

Data visualization was essential to evaluate the distribution of the following variables. Variables that we considered : IsBadBuy, VehicleAge, Make, Transmission, VehOdo, Size, MMRAcquistionAuctionAveragePrice, MMRCurrentAuctionAveragePrice, Vnzip, VehBCost, WarrentyCost.
The 8 variables concerning the acquisition/current prices at the auction and on the retail market have very similar distributions.We agreed on taking few variables like, MMRAcquistionAuctionAveragePrice and MMRCurrentAuctionAveragePrice.

Initially, The MMR values are having minimum of 0 which is impossible to buy a car for 0 price. As we noticed and replaced with the mean of the column to get the minimum value. And also trimmed the bounds of the values by deleting some of the unwanted values from our large dataset. Result , will not start from 0 as in figure 1.1.

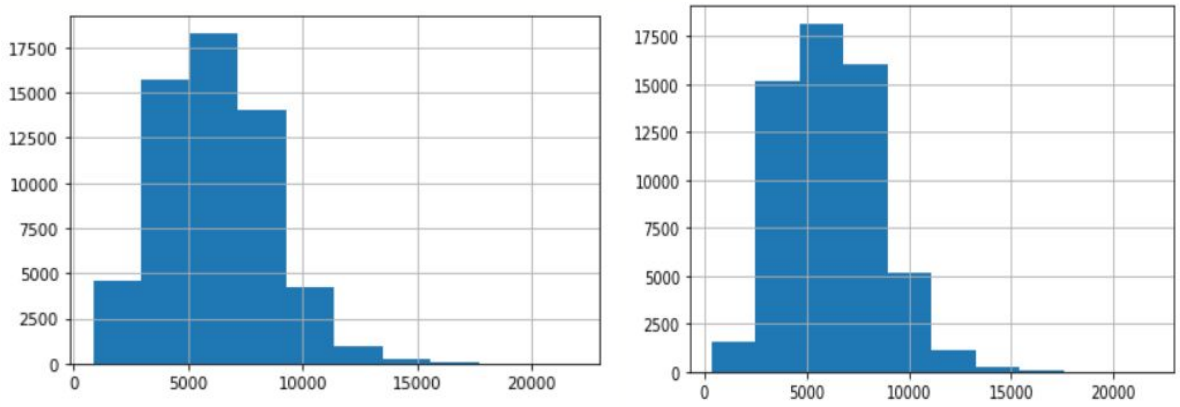


Figure 1.1 Histogram of MMRAcquistionAuctionAveragePrice and MMRCurrentActionAveragePrice

To make MMR values more interesting we cleaned them also.

It is possible to observe that the majority of the vehicle prices goes from $5000 to $15000 with respect to the acquisition price at the auction. Indeed, the distribution of the retail prices have a tail end around the $20000.

In figure 1.2, which is data after removing outliers. the outliers are observed in box plot next to the histogram.

VehBCost:

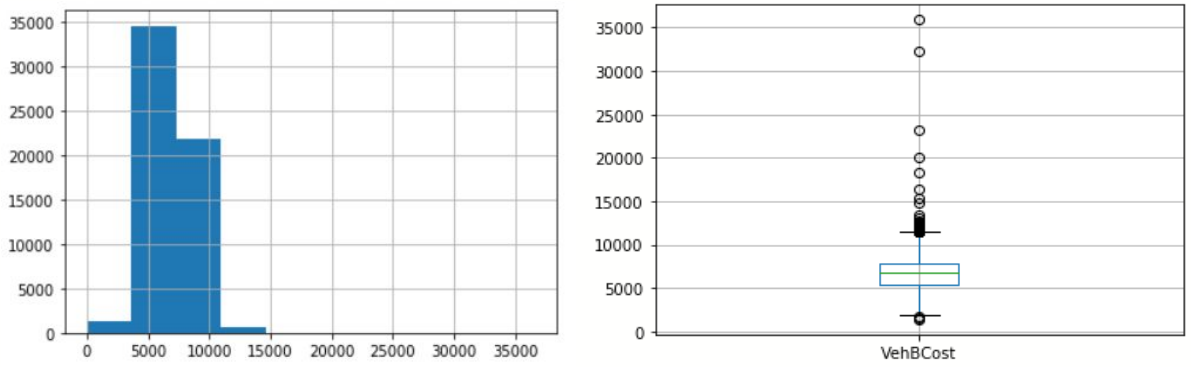


Figure 1.2 Histogram and Boxplot of VehBCost

Both the distribution of MMRAcquisitionAuctionAveragePrice and VehBCost variables were analysed compared to a good/bad purchase situation in Figure 1.3. As it can be seen in the figure below, the two variables are somehow correlated with a huge presence of good purchases.

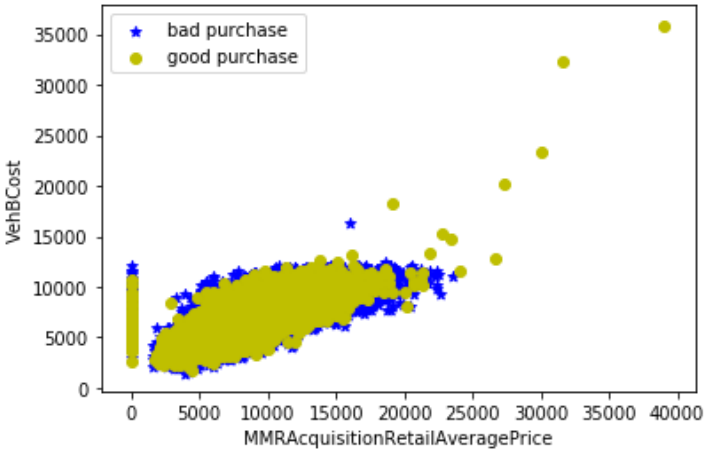


Figure 1.3 Distribution of MMR and VehBCost with respect to IsBadBuy

1.3. Assessing data quality

At a first glance, it appeared that there was a missing column (*Kickdate*) in the dataset that was instead described in the Carvana Dictionary file. Therefore, it was verified whether it was merged with the neighbouring columns but it emerged that there were no *Kickdate* values.

We also checked if the number of distinct observations for each variable was actually equal to that expected. Therefore, *IsOnLineSale* and *IsBadBuy* variables did actually assume binary values.Then, from a comparison of values it resulted that *VehYear* and *VehAge* were consistent.

Honestly, there were many missing values which we rectified them almost every part of the sub module by checking the boxplot accordingly. But our interest we into some of the main variables of dataset, which could give us the best results.
We also verified if a model (*Model*) was always associated with one and only one vehicle manufacturer (*Make*) and we found out that 'Neon' model appears both in Chrysler and in Plymouth. After few researches, comparing images of both models, it emerged that they were the same physical vehicle. That was not a problem because we also found out that Chrysler acquired Plymouth continuing the production of Neon.

1.4. Variables transformations

The transformation and manipulation of variables are done according to the module, since various modules require various transformations. As some only need integer, and some need string.

For instance, IsBadBuy, and IsOnline variables are transformed to boolean type, the MMR minimum values are replaced with mean values of the column, some

categorical data are transformed to integer mapping each record to unique range of numbers. Date is resolved to required format.

IsBadBuy and IsOnline variables was converted to boolean. WheeTypeId was normalized as float. PurchDate variable was consistent but in order to use it in Python in comparison with all the other variables it was converted in required date format. Categorical attributes were converted into int type mapping each record to a unique number.

1.5. Pairwise correlations and eventual elimination of redundant variables

The final Dataframe with drastic change in number of variables with removed unwanted columns because they were highly correlated. Among all columns we selected MMRCurrentActionAveragePrice and VehBCost with correlation of ~ 0.77, which is pretty good. As previously said, the 8 variables concerning *MMRAcquisition* and *MMRCurrent* prices appeared to be very similar in distribution, we calculated the mean and plotted the values to evaluate whether they were redundant and in that case how to deal with it. Plots showed that there were any drastic changes in the values distribution. Some variables are eliminated, a couple of them are listed below: *Nationalty, primeunit* have been deleted as they have unwanted information . The latter was kept as the year information only. As well as, *WheelTypeID* conveyed the same information as *WheelType*.

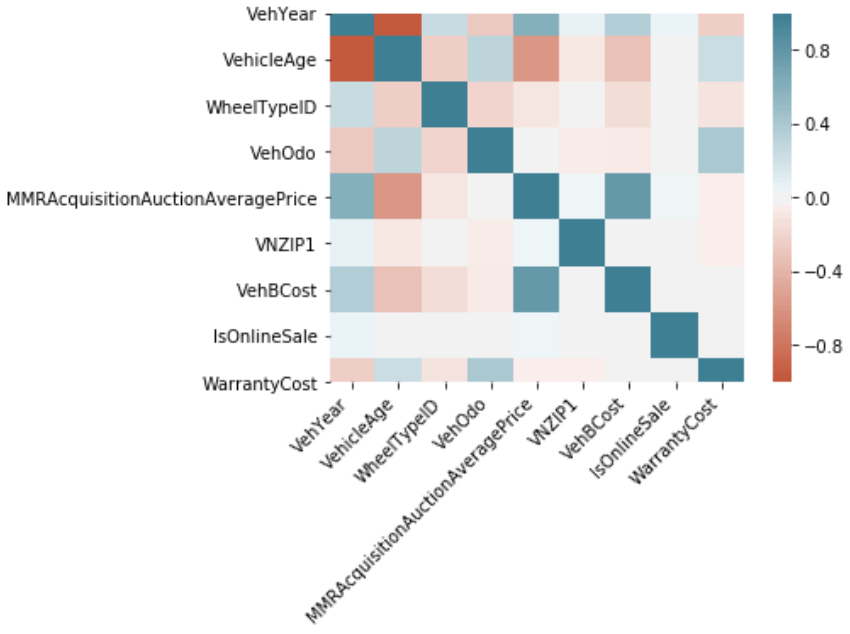


Figure 1.3 Correlation plot

2. Clustering

For Clustering analysis we used K-Means, DBSCAN, and Hierarchical clustering in which we experimented by taking randomly between many pairs of attributes and decides to use them. Before clustering we preprocessed the dataset and made it easy to make clusters in less amount of time.

Different kinds of clustering algorithm may allow to find possible regularities in the distribution of data and then going deeper in the analysis. Before moving forward the implementation of distinct clustering techniques some preprocessing operations were needed.

The attributes the analysis was focused on are the following:

- VehOdo;
- MMRAcquisitionAuctionAveragePrice;
- MMRCurrentAuctionAveragePrice;
- VehBCost;
- WarrantyCost.

Moreover, some normalization adjustments were needed in order to deal with possible ranges of different attributes and avoid consequent bias. For this purpose we adopted MinMax scaler. Not only was it useful to have all features on the same scale but also to better display data distributions when plotted.

We observed the distribution of the attributes without considering the predictive class IsBadBuy.

To select the best attributes we need to graph Scatter Matrix (figure 2.1) of all the variables that are left after reduction. We Selected MMRCurrentAuctionAveragePrice and VehBCost from scatter matrix and performed k-means clustering.

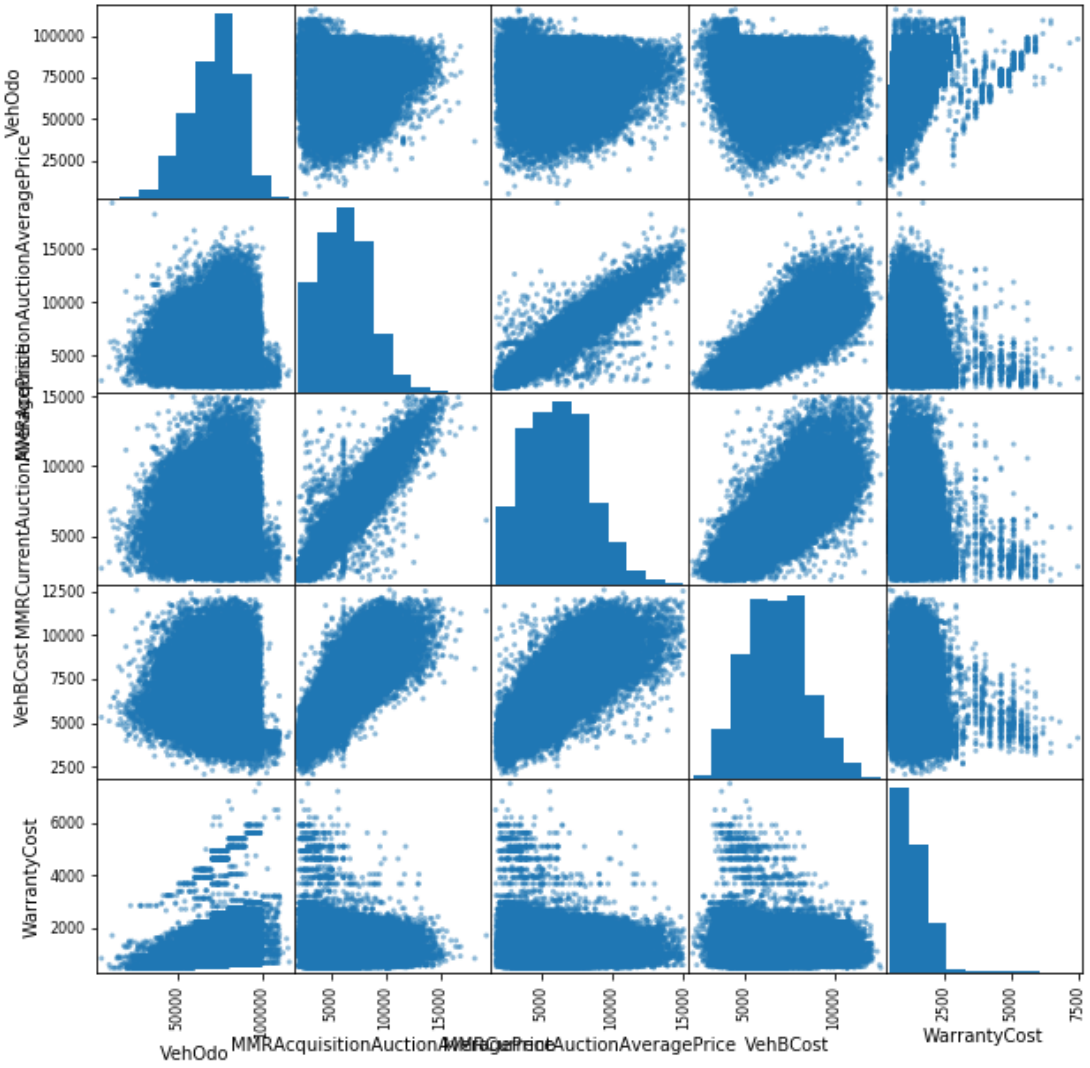


Figure 2.1 Scatter matrix of dataset

2.1. K-means

Since the purpose is dividing the data in k groups on the basis of the similarity/closeness of each element from the centroids/medoids of that distribution, the first issue is to select K points as the initial centroids and choose a K number of initial clusters.

Parameters that were examined in the analysis are the following: **n_clusters** (number of clusters); **n_init** (number of iterations K-means will be run) and **max_iter** (maximum number of iterations K-means will be run in a single run).

- n_clusters = 3;
- n_init = 10;
- max_iter = 100.

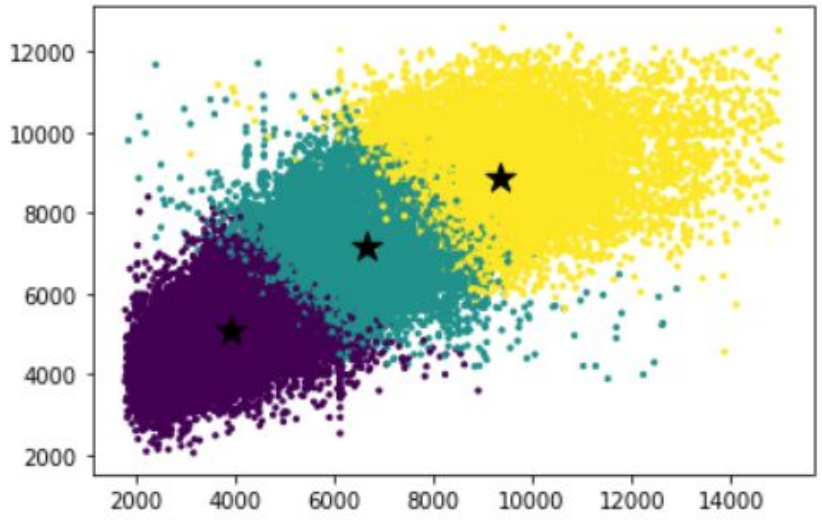


Figure 2.2 Clusters using K-Means

In the figure 2.2 the clusters are obtained by considering the two attributes (MMRCurrentAuctionAveragePrice - VehBCost) as we discussed. By looking at the graph itself we can understand there are 3 clusters with a centroid represented by a star.

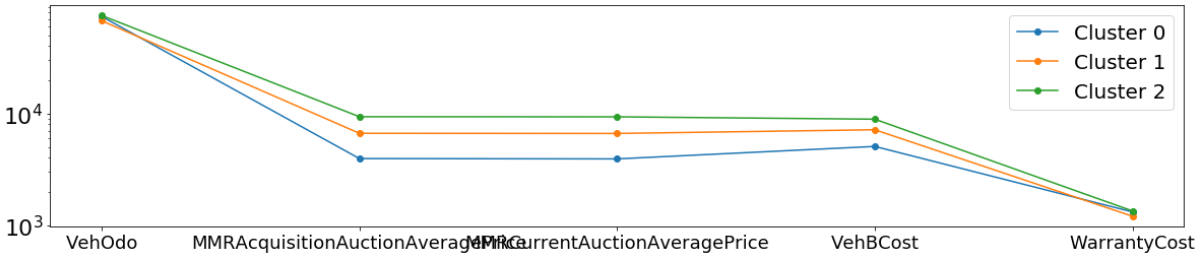


Figure 2.3 Visualization of clusters centers by means of parallel coordinates

In the figure 2.3 we visualize the centers, initially the values of centers start from 73000 (approximately) so it starts from the highest point in the graph. Since we used slicing of centers and applied yscale as “log” it the graph is self explanatory.

2.1.1. Best value of *k*

In order to find the best value of K we adopted Knee method. As shown in Figure 2.3, the point where the derivatives of SSE curve changes is around 5.

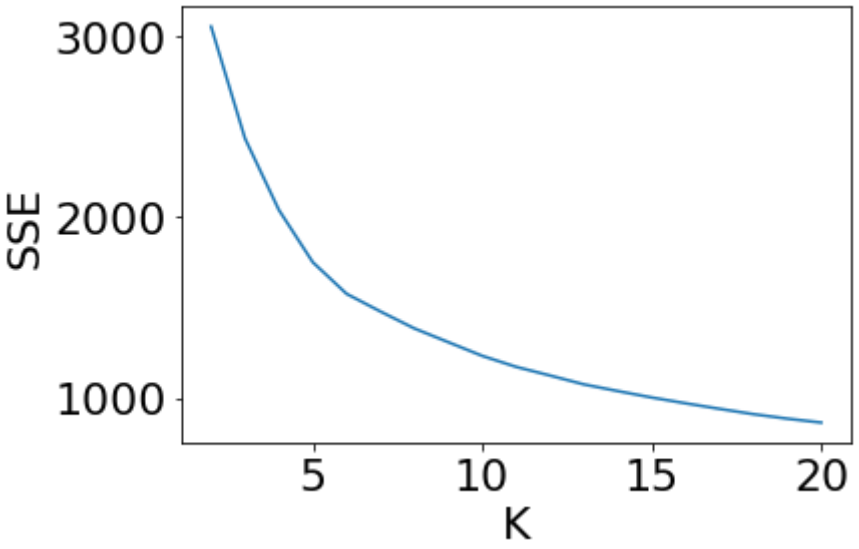


Figure 2.4 SSE curve for K-Means clustering

2.2. DBSCAN

Unlike K-Means, the DBSCAN uses another method to find the clusters. Parameters that were examined in the analysis are the following: **eps** (maximum distance between two points in order to be considered as in the same neighborhood) and **min_sup** (minimum number of points in the neighborhood for a point to be considered as a core point, that point included), and using euclidean distance as default distance function.

Different values of eps and min_sup has been tested but the following ultimately resulted to be the best with a Silhouette value of 0.314:

- eps = 0.7 (the ideal value)
- min_sup = 500

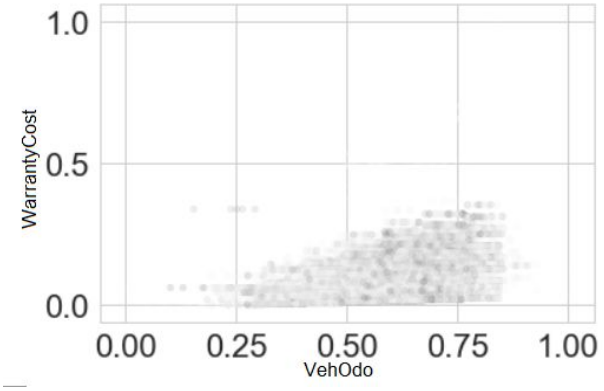
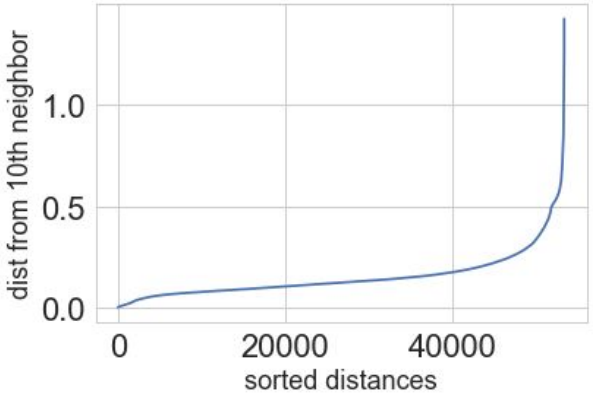


Fig 2.4.1: DBSCAN between WarrentyCost and VehOdo



2.3. Hierarchical

Hierarchical clustering did not give any important result. The threshold for finding interesting subsets is very low and it doesn't seem to be useful for the analysis. As from the results in single link (which takes nearest point of two clusters) is not efficient and made a whole cluster. On the other side, with complete link(takes longest distance of two clusters) we are able to see some clusters but most of them are merged at certain points.

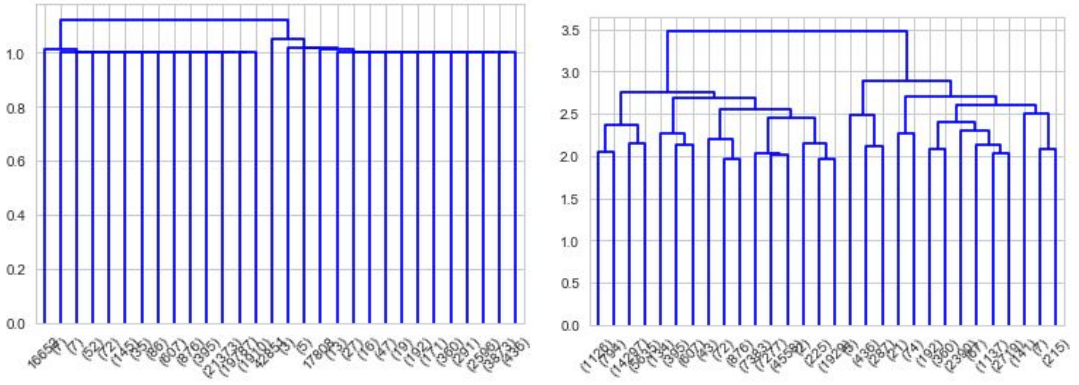


Fig. 2.5 Single link(Left) and complete link(Right)

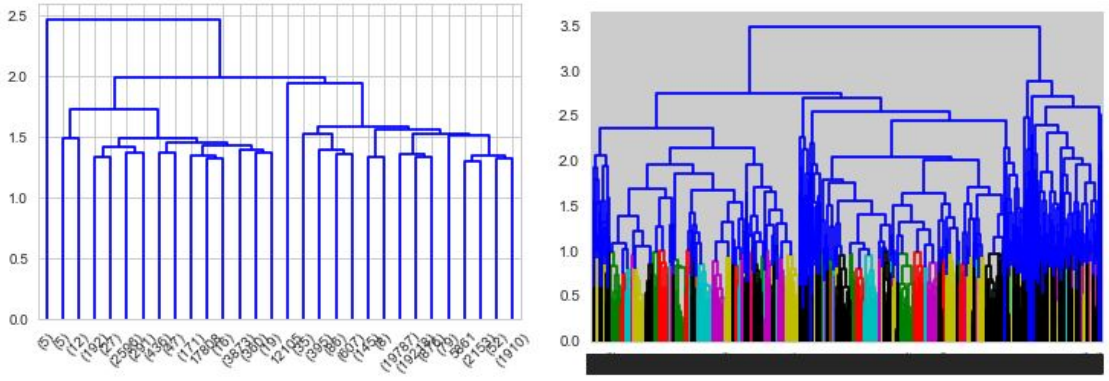


Fig. 2.6. Average link(Left) and Clusters using hierarchical.

Conclusion

The three clustering Techniques are unique and performed well depending on the considered constraints. Clusters are visible well in K-Means and easily understandable without any prior knowledge to anyone. We also tried by changing number of clusters, But K-means performed faster and efficient when compared with other algorithms.

3. Association Rules Mining

In order to move forward the analysis with the extraction of frequent patterns from the dataset some preprocessing operations were needed. So, Following steps are taken:

- Most of the features were dropped while those that we chose to keep are the predictive attribute *IsBadBuy and Make*, *VehOdo*, *VehicleAge, Size* and *MMRAcquisitionAuctionAveragePrice*.
- We have created 10 bins for *MMRAcquisitionAuctionAveragePrice, VehOdo*, *VehicleAge.* And attaches strings at the end
- *IsBadBuy* values have been map to string as follows:
    - 0: Good.
    - 1: Not Good.

Parameters that were examined in the analysis are the following: **supp** (frequency of an itemset with regard to the dataset)**; conf** (X→Y, how frequently the elements of Y appear in X)**; min_lift** (correlation measure).

The other attributes are left as they are; in addition all of the attributes have been considered in this task. After the discretization of the continuous attributes, an acronym or a symbolic abbreviation has been attached to the end of every value in the dataset, in order to obtain easily interpretable strings; a piece of the transformed dataset is depicted in the following figure (Fig. 3.1)

	IsBadBuy	Make	Size	MMRAcquisitionAuctionAveragePriceBin	VehOdoBin	VehicleAgeBin
0	Not Good	KIA	MEDIUM	[3572.2, 7144.4)_MMR	[60271.0, 71360.2)_Veh	[1.8, 2.7)_VAg
1	Good	DODGE	MEDIUM	[3572.2, 7144.4)_MMR	[82449.4, 93538.6)_Veh	[2.7, 3.6)_VAg
2	Good	DODGE	MEDIUM	[7144.4, 10716.6)_MMR	[38092.6, 49181.8)_Veh	[1.8, 2.7)_VAg
3	Good	FORD	VAN	[3572.2, 7144.4)_MMR	[71360.2, 82449.4)_Veh	[3.6, 4.5)_VAg
4	Good	CHRYSLER	VAN	[3572.2, 7144.4)_MMR	[71360.2, 82449.4)_Veh	[3.6, 4.5)_VAg

Figure 3.1 Preprocessed dataset

3.1 Frequent patterns extraction

Frequent pattern extraction task has been performed through the Apriori algorithm imported from the fim library in Python. As far as pattern mining is concerned, only frequent patterns were extracted. We tested with different support values and found 75%, 85% and 95% are to be most appropriate. Support threshold may have the value with respect to different types of itemset (frequent, closed and maximal).

The most interesting patterns are shown in the following tables (Table 3.1, Table 3.2), along with a brief discussion.

Table 3.1: Number of extracted itemsets by support and type

Support	Frequent I.	Closed I.	Maximal I.
75%	1019	1019	370
85%	888	888	319
95%	777	777	288

Table 3.2: Most interesting itemsets

Support	Frequent I.	Maximal I.
75%	1. ('MEDIUM', '[3572.2, 7144.4)_MMR', 'Good'), 11934); 2. ('[3.6, 4.5)_VAg', '[3572.2, 7144.4)_MMR', 'Good'), 8082); 3. ('[71360.2, 82449.4)_Veh', '[3572.2, 7144.4)_MMR', 'Good'), 7470); 4. ('CHEVROLET', '[3572.2, 7144.4)_MMR', 'Good'), 6601); 5. '[60271.0, 71360.2)_Veh', '[3572.2, 7144.4)_MMR', 'Good'), 6135).	1. ('[6.3, 7.2)_VAg', '[3572.2, 7144.4)_MMR', 'Good'), 1312); 2. ('[82449.4, 93538.6)_Veh', 'CHEVROLET', '[3572.2, 7144.4)_MMR', 'Good'), 1283); 3. ('[0.0, 3572.2)_MMR', 'DODGE', 'Good'), 1248); 4. ('FORD', '[71360.2, 82449.4)_Veh', '[3572.2, 7144.4)_MMR', 'Good'), 1243); 5. ('[4.5, 5.4)_VAg', '[60271.0, 71360.2)_Veh', '[3572.2, 7144.4)_MMR', 'Good'), 1230).
85%	1. ('MEDIUM', '[3572.2, 7144.4)_MMR', 'Good'), 11934);	1. ('[2.7, 3.6)_VAg', '[60271.0, 71360.2)_Veh', '[7144.4, 10716.6)_MMR', 'Good'),

	2. ('[3.6, 4.5)_VAg', '[3572.2, 7144.4)_MMR', 'Good'), 8082); 3. ('[71360.2, 82449.4)_Veh', '[3572.2, 7144.4)_MMR', 'Good'), 7470); 4. ('CHEVROLET', '[3572.2, 7144.4)_MMR', 'Good'), 6601); 5. ('[60271.0, 71360.2)_Veh', '[3572.2, 7144.4)_MMR', 'Good'), 6135)	1381); 2. ('[6.3, 7.2)_VAg', '[3572.2, 7144.4)_MMR', 'Good'), 1312); 3. ('[4.5, 5.4)_VAg', 'CHEVROLET', '[3572.2, 7144.4)_MMR', 'Good'), 1292); 4. ('[82449.4, 93538.6)_Veh', 'CHEVROLET', '[3572.2, 7144.4)_MMR', 'Good'), 1283); 5. ('[38092.6, 49181.8)_Veh', 'MEDIUM', '[3572.2, 7144.4)_MMR', 'Good'), 1263).
95%	1. ('MEDIUM', '[3572.2, 7144.4)_MMR', 'Good'), 11934); 2. ('[3.6, 4.5)_VAg', '[3572.2, 7144.4)_MMR', 'Good'), 8082); 3. ('[71360.2, 82449.4)_Veh', '[3572.2, 7144.4)_MMR', 'Good'), 7470); 4. ('CHEVROLET', '[3572.2, 7144.4)_MMR', 'Good'), 6601); 5. ('[60271.0, 71360.2)_Veh', '[3572.2, 7144.4)_MMR', 'Good'), 6135).	1. ('COMPACT', '[0.0, 3572.2)_MMR', 'Good'), 1797); 2. ('[2.7, 3.6)_VAg', '[7144.4, 10716.6)_MMR', 'MEDIUM', 'Good'), 1532); 3. ('Good', 'MEDIUM', '[3572.2, 7144.4)_MMR'), 1508); 4. ('[2.7, 3.6)_VAg', '[60271.0, 71360.2)_Veh', '[7144.4, 10716.6)_MMR', 'Good'), 1381); 5. ('COMPACT', 'CHEVROLET', '[3572.2, 7144.4)_MMR', 'Good'), 1353).

As it is possible to read from the table, the most common traits of the typical not good car is 'MEDIUM' and followed by Price is between [3572.2, 7144.4] and VehOdo [i.e number of km car has traveled] is also less then 71360. Meaning that a car which is 'MEDIUM' in size and is between [3572.2, 7144.4] in average price and in miles run lower then 71360 it is likely to be a good buy.

3.2 Association Rules extraction

For the Association Rules extraction, we need to know the best values to choose. So we tested on various parameters and finally ended up with minimum support of 75%

and 85%, Minimum confidence of 60%, 50% and 40%. Only the rules having a lift value greater than 1.11 have been analyzed, since they are the most valuable ones. The same reason applies to the choice of confidence values, which are sufficiently high.

Number of Good entries are significantly more than the number of bad buy cars in the dataset that is why we are getting more rules for Good. So our thinking was that we can make the model which at the place of telling which is a bad car tells, which good car to buy because we don't have enough data for making model for the other case.

Table 3.3: Extracted rules with 75% support

Support = 75 %				
Min. conf.	Total no. of rules	No. of rules (lift >= 1.11)	No. of rules (Bad buy)	No. of rules (Good buy)
60%	1305	576	3	266
50%	1684	868	10	438
40%	2382	1339	16	749

As explained earlier the dataset is biased towards good so you can observe more number in good buy then bad buy.

Table 3.4: Extracted rules with 85% support

Support = 85 %				
Min. conf.	Total no. of rules	No. of rules (lift >= 1.11)	No. of rules (Bad buy)	No. of rules (Good buy)
60%	1145	496	2	227
50%	1483	755	8	386
40%	2117	1187	14	667

Table 3.5: Extracted rules with Bad Buy

Rules with Bad Buy		
Rule	Confidence	Lift
{[0.0, 3572.2)_MMR', ('[7.2, 8.1)_VAg} -- 'Not Good'	71.45%	4.513
{[7144.4, 10716.6)_MMR', ('[1.8, 2.7)_VAg} -- Not 'Good'	61.76%	2.003

{'MEDIUM', ('CHRYSLER')} -- 'Not Good'	74.94%	1.775

Table 3.5: Extracted rules with Good Buy

Rules with Good Buy		
Rule	Confidence	Lift
{'LARGE', ('[1.8, 2.7)_VAg', 'CHEVROLET', '[71360.2, 82449.4)_Veh', '[7144.4, 10716.6)_MMR'} -- 'Good'	83.26%	6.828
{'LARGE', ('[1.8, 2.7)_VAg', 'CHEVROLET', '[7144.4, 10716.6)_MMR'} -- 'Good'	77.65%	6.367
{'CHRYSLER', ('CROSSOVER', '[3572.2, 7144.4)_MMR'} -- 'Good'	77.31%	6.363
{'MEDIUM SUV', ('[3.6, 4.5)_VAg', 'CHEVROLET', '[7144.4, 10716.6)_MMR'} -- 'Good')	61.26%	5.590

3.3 Predict the target variable and evaluate the accuracy

The most meaning full rule we found in case of being a good car are that VehicleAge is between [0.9 -1.8] and VehOdo between [38092.6 - 49181.8] if we use this two test set provided we can predict with 97.52% accuracy when it's a good buy.

It's not possible to predict missing values in this case because there are no missing values to replace by means of association rules in this dataset. Because rules determined by algorithm are general and missing value is specific.

4. Classification

In the end, learning a model that could predict as accurately as possible whether a car is a good or a bad purchase was the main goal. Hence, the predictive attribute was *IsBadBuy*.

4.1 Features selection

It was considered that the relevant features the analysis should focus on were:

- *VehicleAge;*

- *Make*;
- *Transmission*;
- *VehOdo*;
- *Size*;
- *MMRAcquisitionAuctionAveragePrice*;
- *MMRCurrentAuctionAveragePrice*;
- *VehBCost;*
- *WarrantyCost*.

Make, *Transmission* and *Size* categorical attributes were map from string to integer.

4.2 Decision Trees Learning

Decision trees was implemented as a classification technique. The dataset was split into train (70%) and test (30%). Furthermore, pre-pruning was essential to avoid a potential overfitting issue. Different combinations of parameters were tested with the main objective of maximizing the model performances. Entropy and Gini coefficient were considered as gain formulas.
After trying with a normal Decision Tree Classifier and getting a maximum accuracy of 82%, it was adopted a Random Forest Classifier that was tested with a set of possible parameters. The one that got the best validation was selected.

The table (Table 4.1.) below summarizes the range of parameters adopted:

Table 4.1. Parameters

Criterion	Gini, Entropy
Max_depth	None, [2, …, 50]
Min_Samples_Split	2, 5, 10, 20, 30, 50, 100
Min_Samples_Leaf	1, 5, 10, 20, 30, 50, 100

4.3. Decision Tree Interpretation

Features Importance

VehicleAge 0.4725589745866476
Make 0.050835530463015095
Transmission 0.0

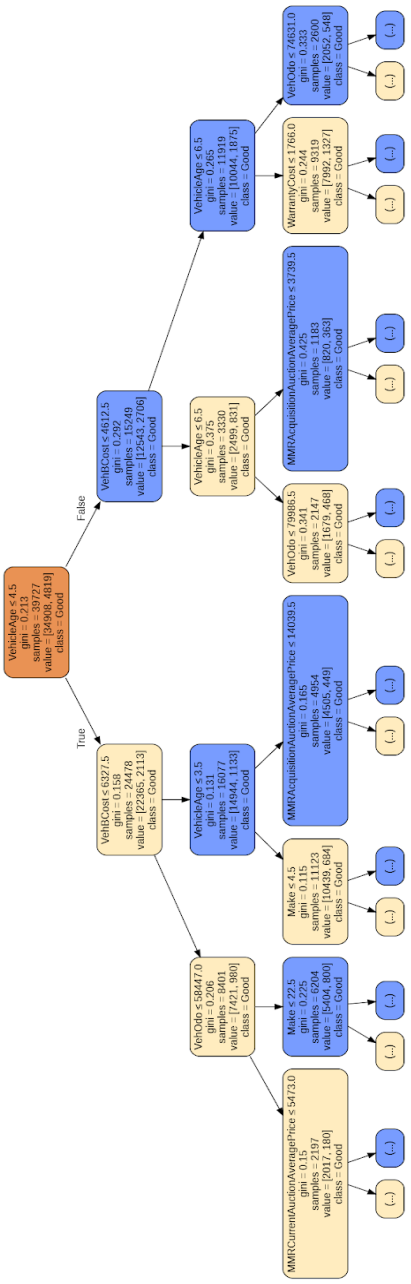
VehOdo 0.09111590036074954
Size 0.003651394837360154
MMRAcqAuctAveragePrice 0.05580809881525737
MMRCurrAuctAveragePrice 0.05701424068699304
VehBCost 0.2288500566167199
WarrantyCost 0.040165803633257384

As we can observe that the most important feature according to decision tree algorithm is VehicleAge and second most important one is VehBCost.

Observing the plot of the tree emerges that there are few parameters determining a good buy. In that *VehBCost* and *VehOdo* occur in many discriminative nodes because that is determined as important feature.

At the root of the tree it was found the cost of the vehicle at the time of its first buy (*VehBCost*). If the price was higher than 5482.5$, the vehicle has higher probability to be a good buy. Another discriminative attribute is the distance traveled by the vehicle (*VehOdo*): if it's over 77000 miles the car will be rarely good. The last remarkable difference is made by the current auction average price (*MMRCurrentAuctionAveragePrice*), that can be a signal of a overstimed car or of a scrap.

Figure 4.1 Decision Tree



4.4. Decision trees validation with test and training set

The validation of the classifier gave a train accuracy of 0.87874 and a train F1-score of 0.93545. The model was considered valid because the test accuracy has been 0.87883 and the test F1-score 0.935496.

4.5. Discussion of the best prediction model

The winner model was the one with the following parameters : **'min_samples_split'**: 15, **'min_samples_leaf'**: 15, **'max_depth'**: 33, **'criterion'**: 'entropy', that gain a mean validation score of 0.879 (std: 0.000).

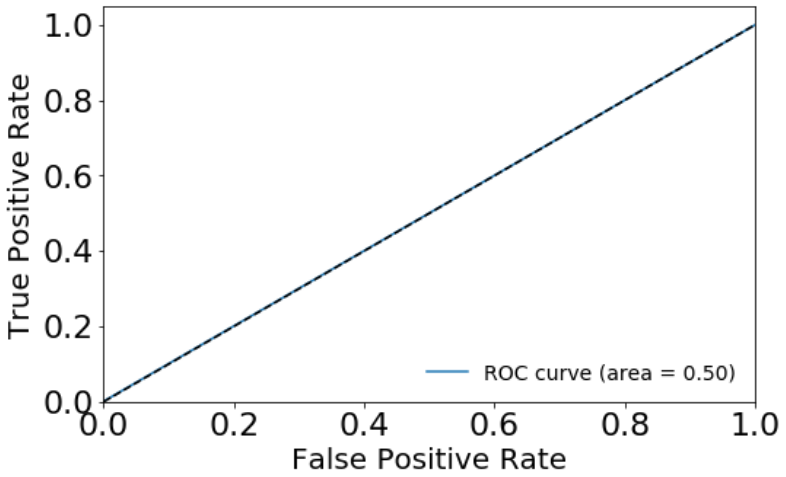


Figure 4.2 ROC curve of the DT

	precision	recall	f1-score	support
0	0.88	1.00	0.94	14960
1	0.24	0.00	0.00	2066
accuracy			0.88	17026
macro avg	0.56	0.50	0.47	17026
weighted avg	0.80	0.88	0.82	17026

Figure 4.3 Performance evaluation measures

Given the very different type of attributes in the dataset, the classification tree has been the best solution to build a model that predicted the goodness of a buy. The low result of the ROC curve seem to be caused by the fact that the data are unbalanced (6993 bad buys over 50479 good ones).

Conclusion

By doing all the above analysis we have determined that clustering algorithms are not able to predict in this situation but decision tree and association rules are doing very well. As you can see there predictive capacity above.