1/21/2022

Data Mining Group Project

Automobile Sales Data Group

Compiled By: Syed Shahzaib Raza (EP20101055)

Course: CS-626 (Data Warehousing & Data Mining)

Instructor: Dr. Tahseen Ahmed Jilani

Degree: MCS

Batch: 2020

Department: DCS-UBIT (UoK)

Shahzaib
DCS-UBIT

Table of Contents

Abstract

The world has seen gradual development in the mode of transportation The Every person in the world wants the best automobile he can have however to achieve that he must have to find the desired vehicle among a lot of data , that is really hard since currently there are hundreds of thousands of data presents on the internet. To achieve that purpose we are extracting automobiles data from one of the most popular website and performing different data mining tools and techniques on it to classify whether an automobile is affordable , mid-range or luxurious. The classification and clustering algorithms used to achieve the goal are Decision tree, K-means, DBSCAN. Python language was used to get the job done while PyCharm (IDE) software was used in order to use the following techniques. Among the algorithms K-means shows the most distinguishable clusters. The result has shown that on the basis of characteristics a car can be judged and its price can be predicted. Among the cars budget ones are the most available car on the market.

Introduction

The world has seen the gradual development of mode of transportation from riding animals like horse, oxen, bull, camels, donkey etc. To man-powered or animal powered vehicles. In 1800's, Watt invented the steam engine and applied this technology to the mode of transportation and with this mankind move towards the age of steam motive power. In 1885 the man known as Karl Benz who is widely known as the one who brought us the era of modern auto mobiles (I.e. cars powered by internal combustion engine). While around the same time Daimler and Maybach fitted 'grandfather clock' (named because of its shape) which was small gas engine powered by kerosene/ paraffin to a two wheeler known as 'riding car'. This engine was smaller and more powerful than any other engine before. In 1908, Ford introduced their model T which was the first car to be ever produced by the assembly line. This made a great development in technology and made automobile much cheaper and more widely available after which the automobiles were made more comfortable and easier to handle. Now a days different industries like Toyota, Honda, Nissan, ford , BMW, Mercedes etc. are mass producing automobiles. Right now there are hundreds of thousands of automobiles models in the market. Some of these models are cheaper than others while some are not ,some are more comfortable than the other while some are straight forward too expensive and luxurious. Every person in the world wants the best of the best, however that is not possible for everyone, therefore they try achieve the next best thing i.e. to get the best in their price range. However to find that best automobile affordable to person, one has to work a lot.

In this project we are using automobiles data and performing different data mining tools and techniques on it for classification of automobiles to find whether the an automobile is affordable, mid-range or luxury. We are trying to study data processing and data mining.

Literature Review

Background:

Checking automobile pricing through machine learning is directly related to information gathering process for technical system. The selling and buying of automobile is greatly dependent on internet these days.
Whether it is a new car or used (resell) car the automobiles industries are the fastest growing industries in Pakistan people. We as a human can classify objects on the basis of their features. We can approach similar ability of computer through data mining which can recognize anomalies, pattern and correlation within data. In this project we would discuss the methods of methodology for forecasting automobile data. One can classify a car by looking at its price, model year or similar other features. So we will be implementing data mining models to perform the classification, recognize anomalies, patterns and correlation. With the help of processing power we can do this on a much larger scale (let say classifying hundreds of cars at a time).

Data Source:

We extracted data from Pakistan's one of the best automobile resell e-commerce site, PakWheels, by our own data extracting script. PakWheels.com gets over 25,000,000 viewer yearly who viewed more than 250,000,000 pages on the site. In last year alone, close to fifty percent of Pakistan's internet population visited PakWheels.com to buy and sell over 400,000 vehicles Since we are only trying to implement the data mining models in a quick way, we are only extracting the data for automobile listed for sale in a specific region of country (taking Karachi as a region).

API: https://www.pakwheels.com/used-cars/search/-/ct_karachi/.json?

The API link is inoperable without client_id and client_secret which vary from user to user. The link returns a JSON object including each in-listed ad of a car as a node which further caries the details of the car posted in ad including price. We scrape the data of first 15 pages of search result of the link (contains almost 500 ads or we can say 500 used cars in Karachi). The features we extracted of each car are as follows:

- Make (name of the company that made that car)
- Model Year(the year in which the car was made in)
- Engine Capacity(power of the engine)
- Air Bags (availability of air bags)
- Air conditioning(whether the air conditioner is in working state or not)
- Power Windows(whether the windows used in car are power windows or not)
- Power Steering (whether the car has power steering or not)
- Sun Roof (availability of sun roof)
- Alloy Rims(availability of alloy rims)
- Price (price of the car)

We added another column in our scraped data as a category column and categorize each vehicle on the basis of its price.

You can visit our data scraping script from below link:

https://github.com/shahzaib-raza/DataMining/blob/main/data_extractor.py

Decision Tree:

Decision tree uses a tree like structure and their possible combinations to solve a particular problem. It lies in the class of supervised learning algorithms and it can be used for both regression and classification purposes.

A decision tree is a tree data structure which have a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, and each of the branch denotes the outcome of a test, finally each leaf node holds a class label which predicts the class. The first and the topmost node of the tree is the root node.

We have to make some assumptions while implementing Decision-Tree algorithm. These are listed below:-

- The whole training set is considered as the root at initial stage.

- Feature values need to be categorical. If the feature columns have continuous values then they must discretized before building the model.
- Tree node splits on the basis of test results.
- Order of attributes to be placed as nodes is done by statistical approaches, mainly Gini and Entropy.

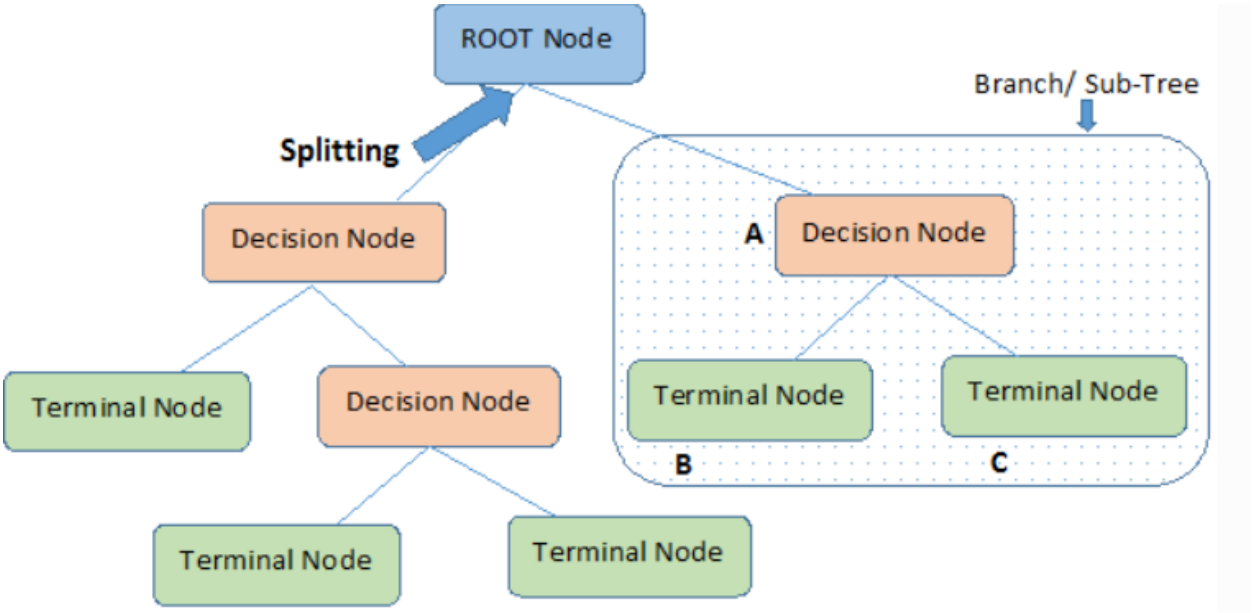
Some advantages of decision tree are:

- A decision tree does not require normalization and scaling of data as well.
- Missing values in the data also do not affect the decision tree.
- A Decision tree model is very easy to explain to a non-technical person.

While some of the disadvantages are:

- It sometimes take huge amount of time to train a decision tree model.
- Calculations can get sometimes very complex and thus it can be expensive on training.
- It can only be used for classification purpose.

Some of the terminologies of decision tree are:



There are two main criterion in decision tree on the basis of which we split a node of the tree optimally:

- Gini Index
- Entropy

Gini Index:

$$\text{GiniIndex} = 1 - \sum_{i=1}^{c}(P_i^2)$$

Here, i: i_{th} class

c: no. of classes

P_i: Probability of i_{th} class

Entropy:

$$\text{Entropy} = -\sum_{i=1}^{c}(P_i - \log_2(P_i))$$

Here, i: i_{th} class

c: no. of classes

P_i: Probability of i_{th} class

References

The guide is provided by our instructor

K-Means:

K-Means clustering is one of the hierarchical clustering method which identifies clusters on basis of centroids or we can say it is a distance based algorithm. The clustering is done by minimizing sum of squares of distance between data and the corresponding cluster centroid.

$$\text{SSE} = \sum_{j=1}^{k}\sum_{i=1}^{n}|x_i - c_j|^2$$

Here, k: No. of centers (no. of clusters)

n: No. data points

Hierarchical clustering determines cluster assignments by building a hierarchy. This is implemented by either a bottom-up or a top-down approach. This methods produce a tree-based hierarchy of points called a Dendrogram. Similar to partition clustering, in hierarchical clustering the number of clusters (k) is often predetermined by the user. Clusters are assigned by cutting the Dendrogram at a specified depth that results in k groups of smaller Dendrograms.

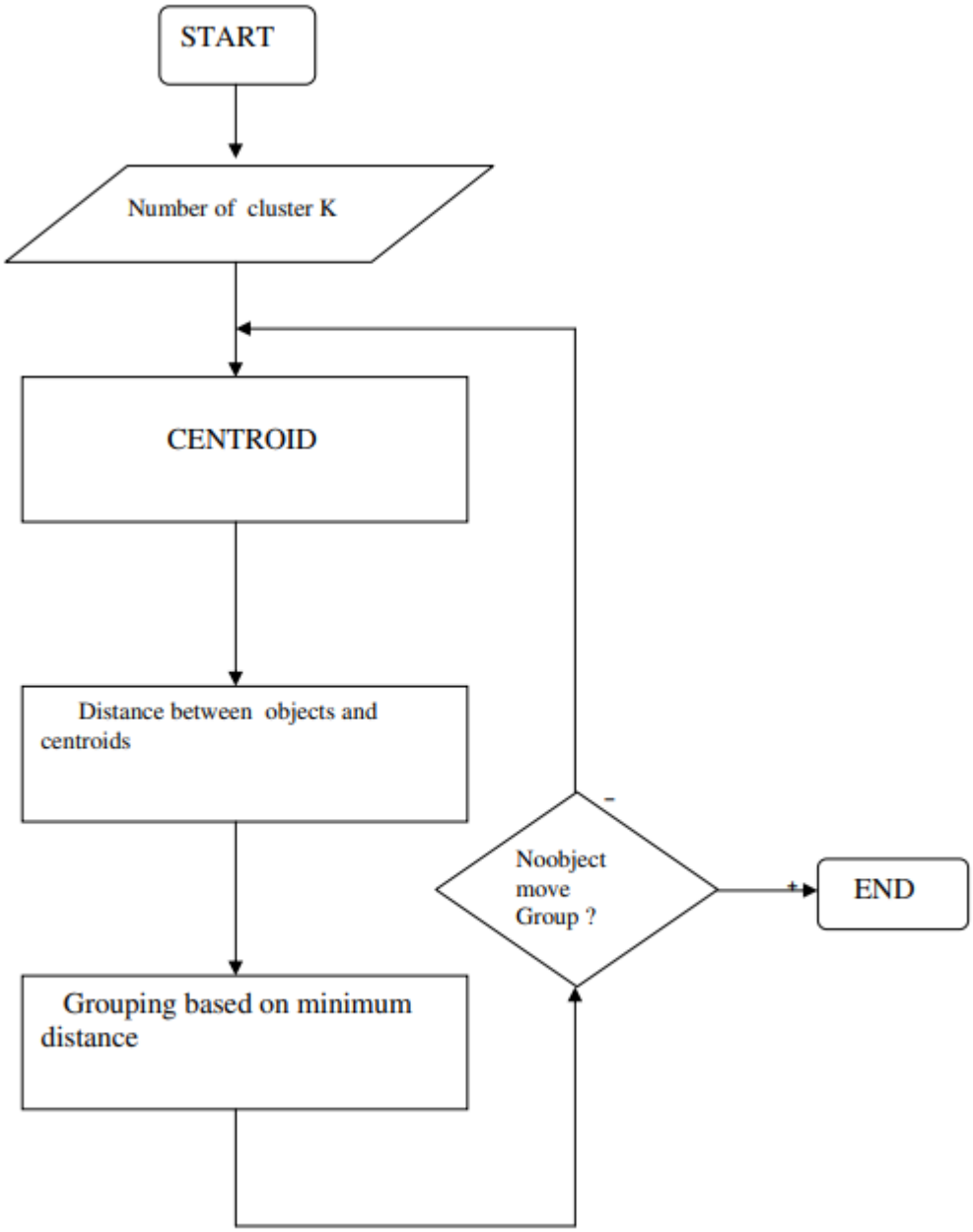
The strengths of hierarchical clustering methods include the following:

- They often reveal the finer details about the relationships between data objects.
- They provide an interpretable Dendrogram.

The weaknesses of hierarchical clustering methods include the following:

- They're computationally expensive with respect to algorithm complexity.
- They're sensitive to noise and outliers.

K-Means algorithm flowchart:



References

Arvai, K., 2019. *K-Means Clustering in Python: A Practical Guide.* [Online]
Available at: https://realpython.com/k-means-clustering-python/

Karim, M. E., Yun, F. & Krishna Madani, S. P. V. S., 2010. *Fuzzy Clustering Analysis.* [Online]
Available at: https://www.diva-portal.org/smash/get/diva2:829433/FULLTEXT01.pdf

DBSCAN:

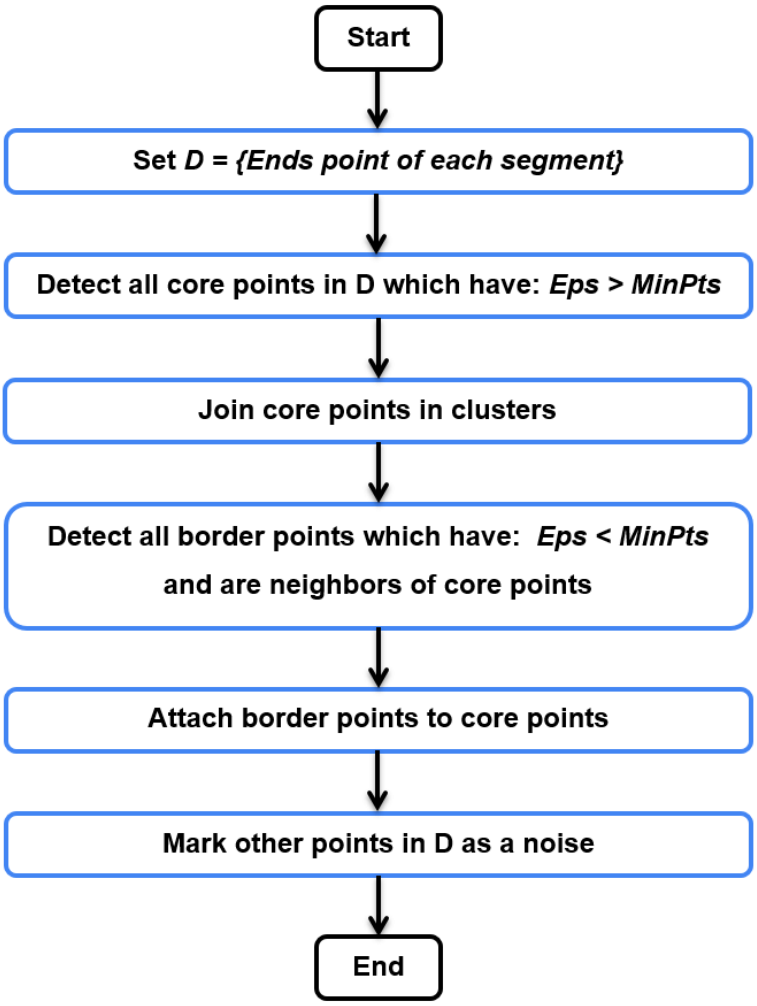
Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a unsupervised model, which means it automatically select number of clusters unlike the K-means where we have to specify the number of clusters (k). There are various advantages of DBSCAN over K-means which are as follows:

- Capable of finding arbitrarily shaped clusters.
- Clusters are defined as dense regions surrounded by low-density regions.
- Automatically select the number of clusters.
- Needs only one scan through the original data set.

Some disadvantages of DBSCAN algorithms are:

- Cannot work with high dimensional data.
- Fails in making clusters with variable density.
- It is very sensitive to its cluster parameters, i.e. eps and min_points.

DBSCAN algorithm:



The two most important parameters used in DBSCAN are: ε and N_{min}

- ε: Max. distance between two neighbors
- N_{min}: Min. data points must be neighbors to a single point in order to make it a center.

References

The guide is provided by our instructor

Automobile Sales Data

Our extracted data includes the following feature columns:

Feature Name	Data Type
make	String
model_year	Integer
engine_capacity	Integer
air_bags	Boolean
air_conditioning	Boolean
power_windows	Boolean
power_steering	Boolean
sun_roof	Boolean
alloy_rims	Boolean
amount	Integer

And the data set have a predictor column: category (String)

We divided automobiles into 3 categories:

- Affordable (price under 12 lacs)
- Mid_range (price > 12 lacs and < 22 lacs)
- Luxury (price > 22 lacs)

	make	model_year	engine_capacity	air_bags	air_conditioning	power_windows	power_steering	sun_roof	alloy_rims	category	price
2	KIA	2021	2000	True	True	True	True	False	True	luxury	5100000
3	Honda	2019	1500	True	True	True	True	False	True	luxury	6800000
4	Honda	2018	1800	True	True	True	True	True	True	luxury	3695000
5	Honda	2009	1800	True	True	True	True	True	True	mid_range	2195000
6	Honda	2017	1300	False	True	True	True	False	True	luxury	2245000
7	Mercedes Benz	2013	1800	True	True	True	True	True	True	luxury	6200000
8	Audi	2018	1800	True	True	True	True	True	True	luxury	14000000
9	Proton	2021	1500	True	True	True	True	True	True	luxury	6500000
10	Hyundai	2021	2500	True	True	True	True	True	True	luxury	8000000
11	Audi	2020	1400	True	True	True	True	True	True	luxury	14500000
12	Honda	2017	1800	True	True	True	True	True	True	luxury	3425000
13	Honda	2017	1500	True	True	True	True	False	True	luxury	2885000
14	Honda	2017	1800	True	True	True	True	True	True	luxury	3400000
15	Honda	2004	1500	False	True	True	True	False	False	mid_range	1340000
16	Honda	2021	1800	True	True	True	True	True	True	luxury	4400000
17	Honda	2014	660	True	True	True	True	False	True	mid_range	1900000
18	Suzuki	2011	1000	False	True	False	False	False	False	affordable	725000
19	Changan	2021	1000	False	True	True	True	False	False	mid_range	1850000
20	Suzuki	2010	1000	False	True	False	False	False	False	affordable	690000
21	Toyota	2013	1800	True	True	True	True	False	False	luxury	3150000

You can visit our data scraping script from below link:

https://github.com/shahzaib-raza/DataMining/blob/main/data_extractor.py

Data Mining

Decision Tree

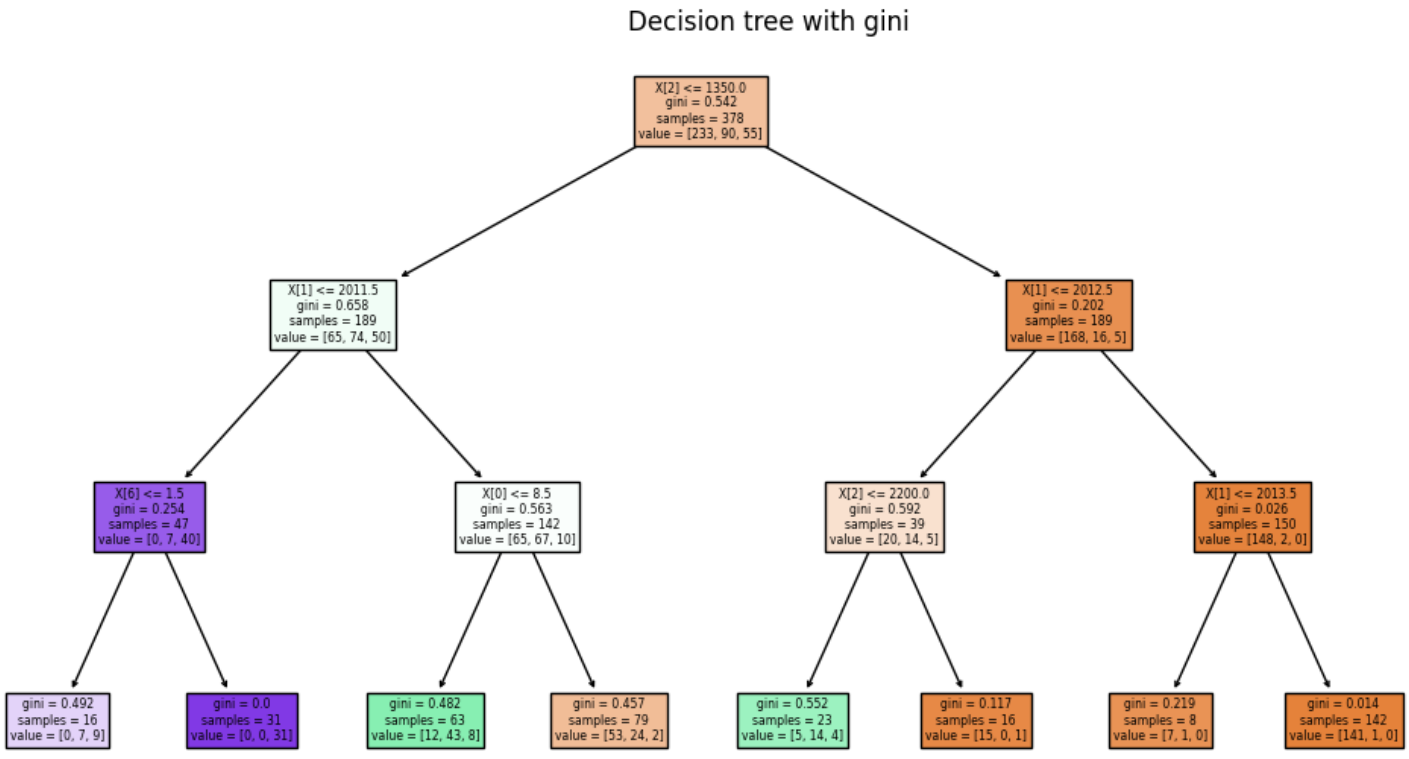
We applied decision tree algorithm using python's Scikit-Learn module. Since it is an algorithm of supervised learning we split our data into train and test sets. Secondly we have to convert all Boolean columns into numeric columns by category encoder. Also we removed the price column from our data set as we are trying to categorize the car without knowing its price, thus our data looks like:

```
Run:  dec_tree
C:\Users\Shahzaib\AppData\Local\Programs\Python\Python39\python.exe C:/Users/Shahzaib/
     make  model_year  engine_capacity  ...  power_steering  sun_roof  alloy_rims
190     2        2016             1800  ...               1         2           1
128     9        2019             1300  ...               1         1           2
183     9        2011             3000  ...               1         1           1
401    13        2015              660  ...               1         1           2
324     9        2012             1300  ...               1         1           2

[5 rows x 9 columns]
Run  TODO  Problems  Terminal  Database Changes  Python Packages  Python Console
```

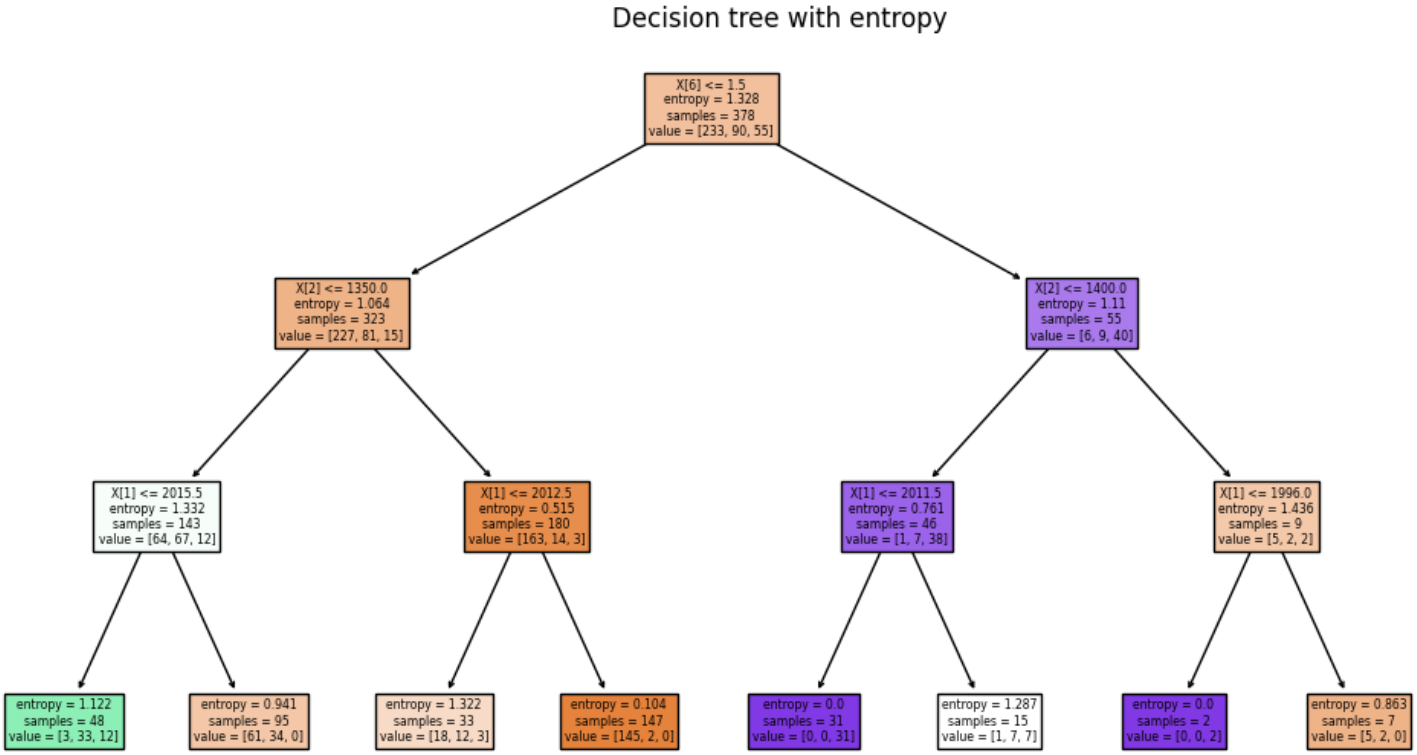
Output for Gini Criterion:

```
Accuracy score for gini criterion:  0.7301587301587301
Confusion matrix for gini criterion:
[[68  3  0]
 [14 19  0]
 [ 3 14  5]]
```



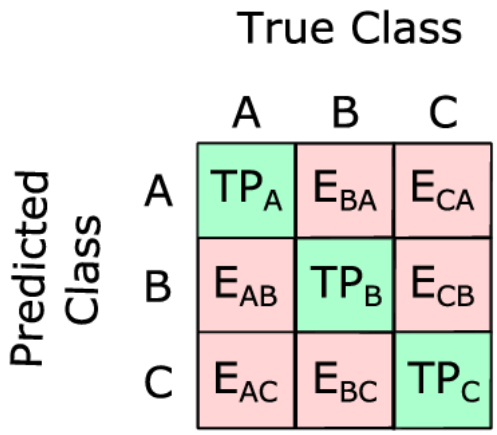
Output for Entropy Criterion:

```
----------------------------------------------------------
Accuracy score for entropy criterion:  0.7380952380952381
Confusion matrix for entropy criterion:
[[71  2  0]
 [17 12  0]
 [ 5  9 10]]
```



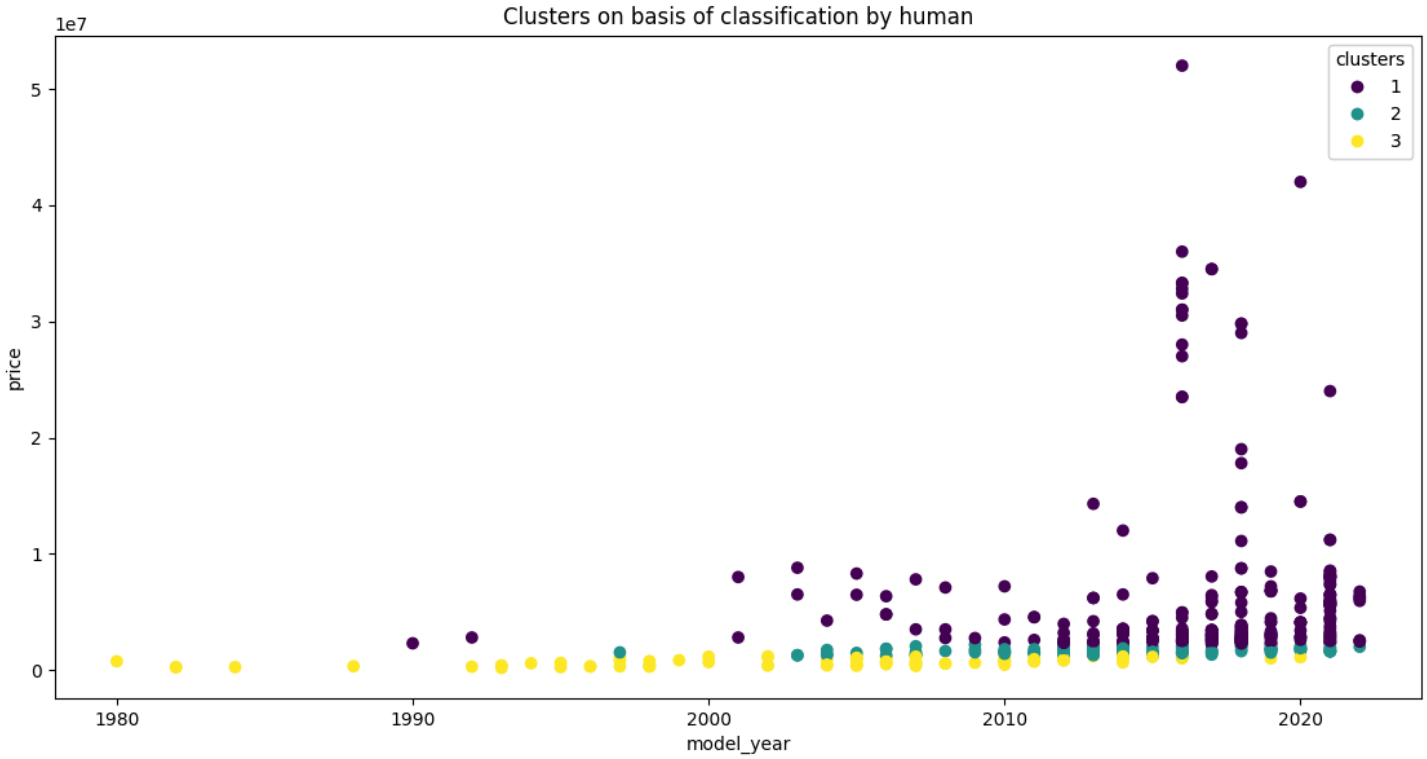
Result:

As we can see the accuracy score is almost same for both gini and entropy criterion i.e. 0.73 or we can say 73%. The multiclass confusion matrix tells us about the counts after fitting and predicting the data as follows:

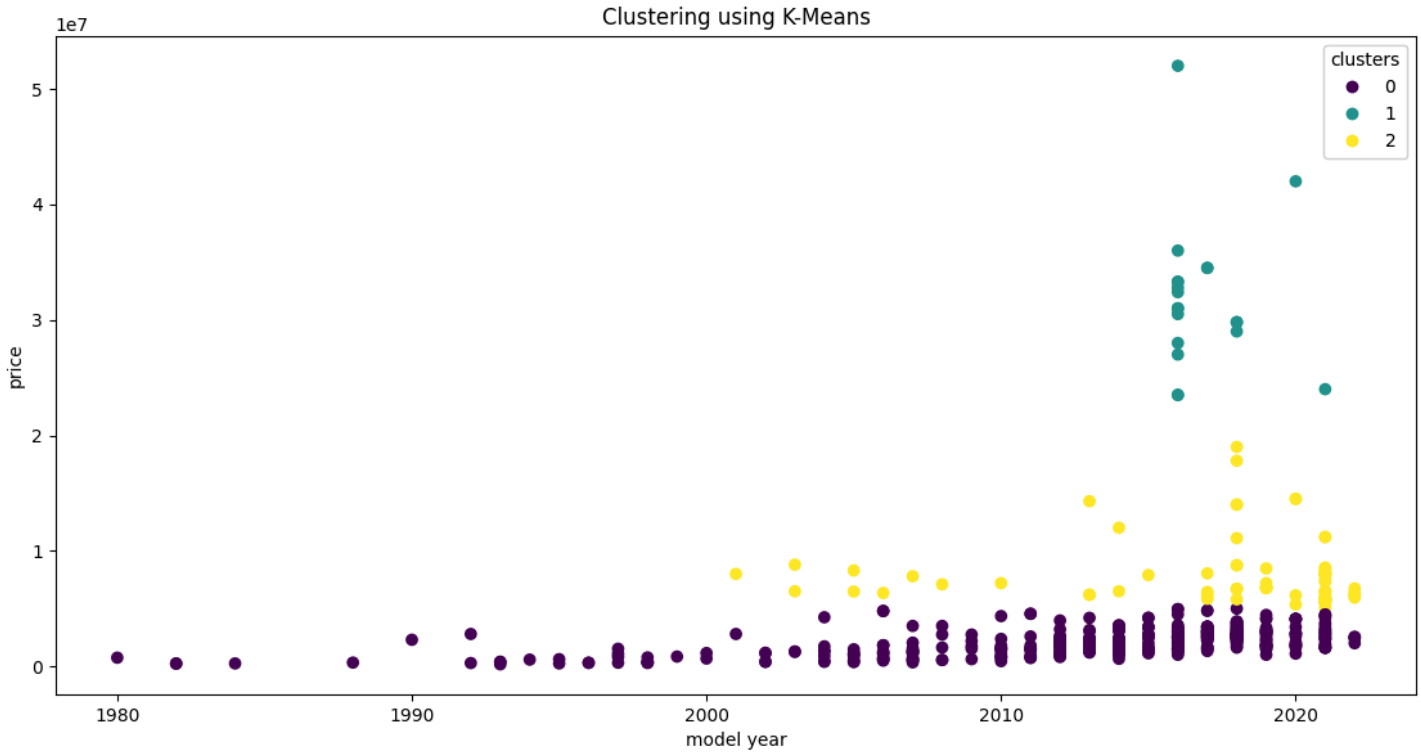


K-Means

In K-Means we first assign K-Means classifier from Scikit-Learn module with number of clusters equal to 3. We assign 3 clusters as we are classifying the data into three categories. First we select the two columns from data in order to observe the clusters on the basis of category column and the results are as follows:



And after fitting the data into K-Means classifier the clusters formed by K-Means can be visualize as follows:



Result:

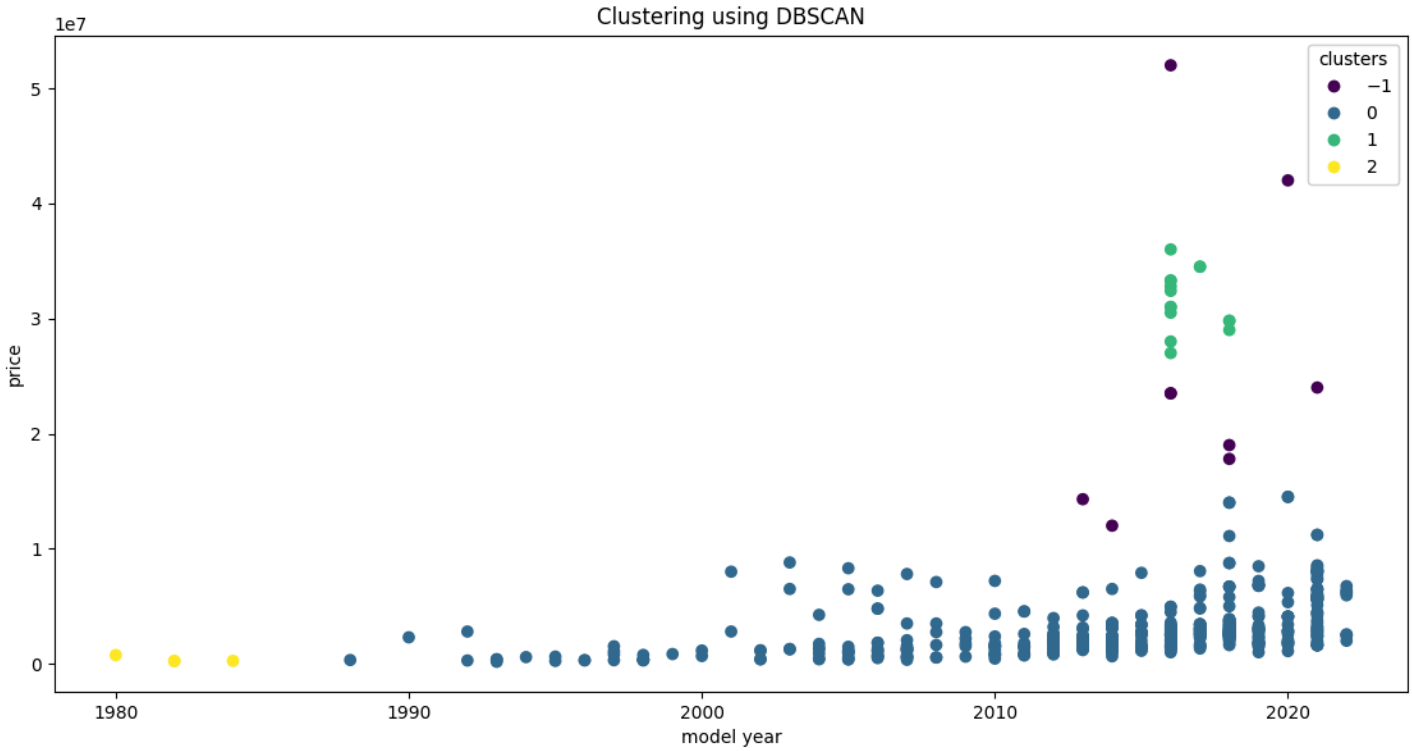
As we can see from visualizations the classification of cars on basis of clusters formed by K-Means algorithm is way more distinguishable than the classification of cars done by humans.

DBSCAN

While dealing with DBSCAN algorithm we must have to consider couple of things. First we have to preprocess the data first and make feature values on a same scale. For the sake of the scaling the data we used Scikit-Learn's preprocessor class and specifically used StandardScaler method to scale the data. Secondly the DBSCAN algorithm works on spatial data means we cannot apply DBSCAN on multidimensional data. Now keeping all this in mind first we select columns from our data set which are most obvious ones i.e. "Model_year" and "Price". Then we preprocessed the data using our StandardScaler method. After all this preprocessing our input matrix which we will be fitting in our DBSCAN class looks like:

	model_year	price
0	0.991582	0.133348
1	0.704283	0.403840
2	0.560633	-0.090205
3	-0.732215	-0.328874
4	0.416983	-0.320918

Now all we have to fit this preprocessed data into our DBSCAN class and the clusters our DBSCAN algorithm formed can be visualized as:



The cluster 1, 2 and 3 are the respective clusters of classes and the cluster -1 is used to identify the outliers, i.e. those points which are not classified as part of any cluster in our data. The parameters which we set for clustering in DBSCAN are:

- ε: 0.5
- N_{min}: 3

Conclusion

From the above results of classification and clustering we can conclude that we can predict the class of a car without having any idea about its price by just giving its features and model year to our data mining models. It is very useful for predicting the category of upcoming cars too. We can use these models on industry level for automating the human behavior of categorizing cars while enlisting them for sale. Also in some cases we can clearly see that the clusters formed by our data mining models are better classifying the data as compared to as we humans do, as in our experiment K-Means done.

E-Code Book:

Github Repository for complete project: https://github.com/shahzaib-raza/DataMining

Data Scrapper: https://github.com/shahzaib-raza/DataMining/blob/main/data_extractor.py

Data Set: https://github.com/shahzaib-raza/DataMining/blob/main/data.csv

Decision Tree: https://github.com/shahzaib-raza/DataMining/blob/main/dec_tree.py

K-Means: https://github.com/shahzaib-raza/DataMining/blob/main/k_means.py

DBSCAN: https://github.com/shahzaib-raza/DataMining/blob/main/db_scan.py