DATA SCIENCE AND MACHINE LEARNING

Automobile Customer Segmentation

Submitted By –

Subham Sarangi

TABLE OF CONTENT

INTRODUCTION.....3
PROBLEM STATEMENT.....3
DATA UNDERSTANDING.....4
DATA PREPARATION.....5
MODELING.....8
CONCLUSION.....17

Source File- https://archive.ics.uci.edu/dataset/10/automobile

INTRODUCTION

Businesses must figure out how to customize their goods and services to satisfy various requirements of their consumers in the rapidly evolving globe of motor vehicles, where buyer tastes, demands, and buying behaviors are highly variable. We present a novel solution—a thorough model of automobile customer segmentation—within this intricate and competitive environment. The foundation of our strategy is the well-known CRISP-DM structure, which is strengthened by the effective use of clustering methods.

The CRISP-DM technique offers an organized and methodical structure for deriving insightful information from data. It includes several distinct phases, such as modeling, assessment, execution, and comprehension of business. We make certain that our buyer division attempts are demanding and in line with business goals by following this approach.

The real innovation, though, lies in our decision to make clustering the primary approach part of the CRISP-DM. Using methods of clustering, we can separate the vast amount of data in the automobile industry and classify customers into useful sections according to shared traits and behaviors. Businesses can precisely tailor their goods, marketing plans, and interactions thanks to division, which reveals undetected trends and unique buyer personas.

since client division is the primary driver of personalized and aimed experiences, companies can move past generalised advertising and provide goods. By matching their approaches to the particular requirements of their various consumer categories, this strategy promises to enable automobile manufacturers to improve customer happiness, foster customer devotion, and eventually excel in a highly competitive marketplace. We are leading the way in changing the customer-focused environment of the auto sector via this innovative initiative.

PROBLEM STATEMENT

Our goal is to recognize and categorize various customer groups beneath the auto industry using grouping as a potent data analysis technique. The division will depend on several factors, such as behavioral patterns, personal demographics, past purchases, and tastes for vehicles, among others. In doing this, we hope to:

Improved Customization, Efficient Focus (segment), Production Optimization, Enhance Client Satisfaction, Improve the Competitiveness

DATA UNDERSTANDING

It can be observed that there is missing data in the given data set.

- Data Set, as mentioned below, includes 27 columns with “205” data entries.

Next, we have numeric and categorical data in the given data set.

- Categorical data: 10

  Make, Fuel Type, Num-of-Doors, Body Style, Drive wheels, Engine Location, Engine-type, Num-of-cylinders, Fuel-system.

- Numerical columns: 17

  ID, Symbolling, Normalized Losses, Wheelbase, Length, Width, Height, Curb Weight, Engine-size, Bore, Stroke, Compression-ratio, Horsepower, Peak-rpm, City-mpg, Highway-mpg, Price.

ID	For indexing
Symbolling	Risk factors associated with the price (-3 to +3)
Normalized Losses	Annual avg. loss of money/ insured automobile
Make	Automobile name
Fuel Type	Diesel or petrol
Num-of-Doors	Number of doors – 2 or 4
Body-Style	Open air, leg space, convertible
Drive-wheels	Wheels - Forward
Engine-location	Back or front end
Wheelbase	Type of size of base
Length	Length of model
Width	Width of model
Height	Height of model
Curb-weight	Weight of automobile
Engine-type	Type of engine used
Num-of-cylinders	Number of cylinders present
Engine-size	Size of engine
Fuel-system	Type of fuel system
Bore	The diameter of each wheel of the automobile
Stroke	Phases in the engine’s cycle
Compression-ratio	Vol. of cylinder and chamber in the engine
Horsepower	Power of automobile
Peak-rpm	Highest revolutions per min
City-mpg	Score of a car in an average city
Highway-mpg	Score of a car on the highway
Price	Amount of the automobile

DATA PREPARATION

Since null values are present in the data, data cleaning is necessary to remove redundancy.

(i) Firstly, dropping the "ID" column and looking for missing values in each attribute

```
#Dropping ID which is not required for the analysis
df.drop('ID', axis=1, inplace=True)
```

Missing values:

normalized-losses: 41, num-of-doors: 2, bore: 4, stroke: 4, horsepower: 2, peak-rpm: 2, price: 4

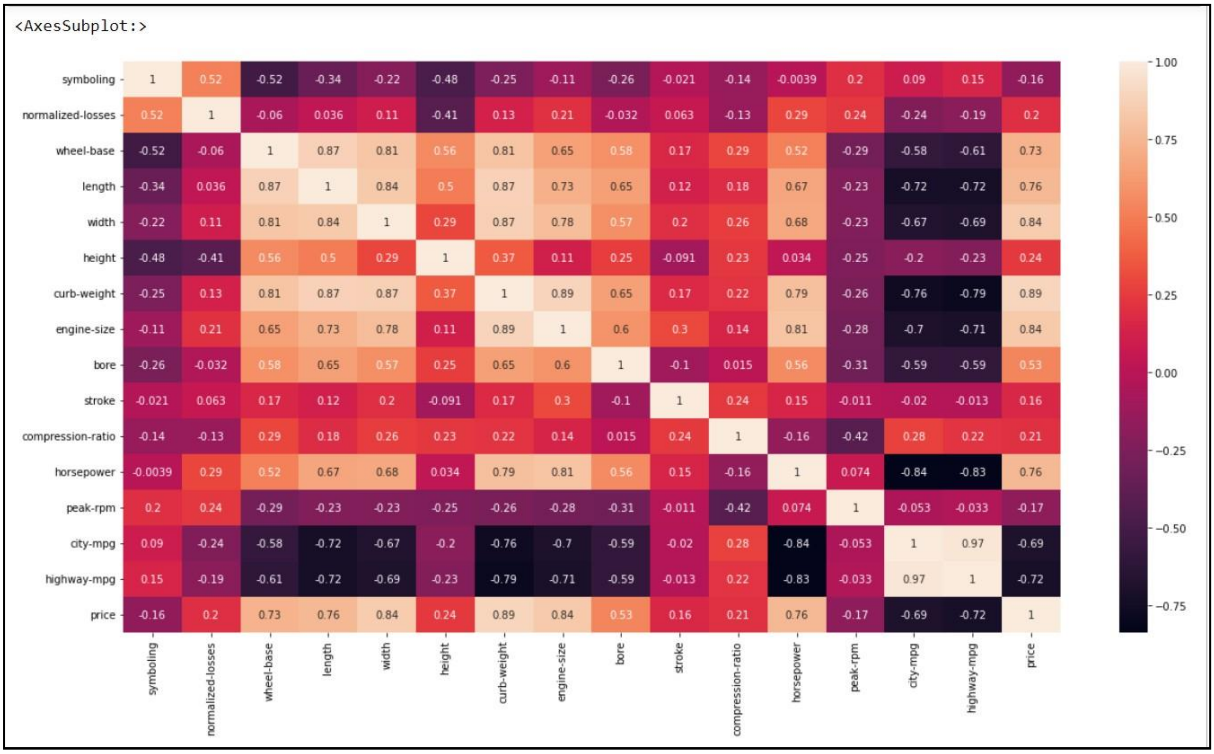
(ii) Secondly, finding the proportion of missing values in the data set

Overall missing values is < 10%

(iii) Now, carrying out missing value analysis with a "**complete case approach**"

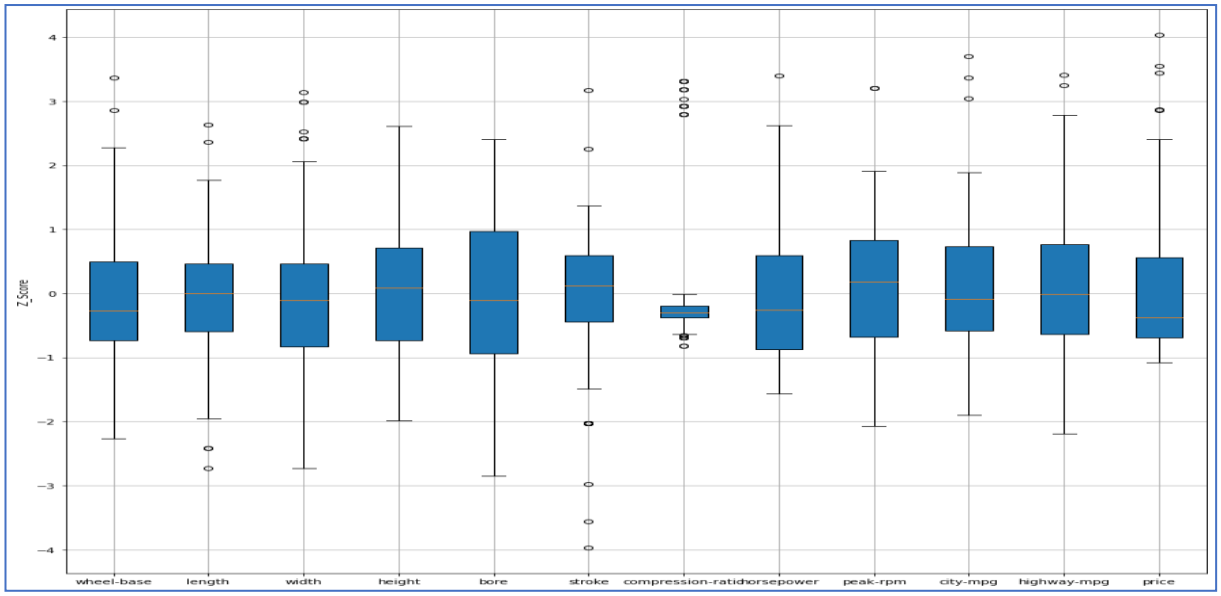
```
Original data: (204, 26)
After removing cases with missing values: (159, 26)
```

(iv) Visualizing the reduced data set using a "**heat map**"



(v) Now detecting the outliers and removing them using a "**box plot**"

From the box plot, we can notice that fewer variables 'wheelbase', 'length', 'width', 'stroke', and 'compression-ratio' have several outliers.



Size of data set after removal of outliers

```
df_cc = df_cc.drop([46,69,71])
df_cc = df_cc.drop(['z_score_tc'],axis=1)
df_cc.shape
```

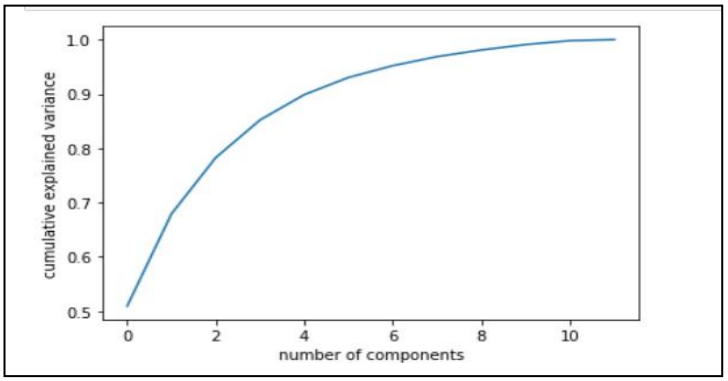
```
(156, 26)
```

(vi) The data has been cleaned, but the range of variables varies, which can again cause inappropriate results. So, all the variables are **standardized** (scaled down to the same level)

	wheel-base	length	width	height	bore	stroke	compression-ratio	horsepower	peak-rpm	
count	1.590000e+02	1.590000e+02	1.590000e+02	1.590000e+02	1.590000e+02	1.590000e+02	1.590000e+02	1.590000e+02	1.590000e+02	1
mean	3.910219e-17	1.592018e-16	-9.216946e-17	7.680788e-17	1.578053e-16	-1.431420e-17	-1.024687e-16	-7.820439e-17	7.122185e-17	-6
std	1.003160e+00	1.003160e+00	1.003160e+00	1.003160e+00	1.003160e+00	1.003160e+00	1.003160e+00	1.003160e+00	1.003160e+00	1
min	-2.264382e+00	-2.726052e+00	-2.733387e+00	-1.989450e+00	-2.852323e+00	-3.967739e+00	-8.153080e-01	-1.562169e+00	-2.075946e+00	-1
25%	-7.307413e-01	-5.888313e-01	-8.278868e-01	-7.292889e-01	-9.385807e-01	-4.468386e-01	-3.768500e-01	-8.763838e-01	-6.759523e-01	-5
50%	-2.648250e-01	-1.204546e-03	-1.068868e-01	8.871045e-02	-1.130451e-01	1.144643e-01	-2.994751e-01	-2.559115e-01	1.855821e-01	-8
75%	4.922888e-01	4.688969e-01	4.596132e-01	7.077369e-01	9.751611e-01	5.907213e-01	-1.963085e-01	5.931560e-01	8.317329e-01	7
max	3.365439e+00	2.627881e+00	3.137613e+00	2.609033e+00	2.401086e+00	3.176117e+00	3.311356e+00	3.401610e+00	3.200952e+00	3

K-means Clustering, Hierarchical clustering uses the Euclidean distance, which gets affected as the number of dimensions increases. So, before using these methods, we must reduce the number of dimensions. Hence, we are using Principal Component Analysis which is by far the most popular dimensionality reduction algorithm.

(vii) **PCA** has been executed so that the actual dimension gets reduced and features carrying "maximum variance" can be sorted. The find the number of components and subsequent variance- the **Scree Plot** is used.



N_components = 7, we can capture 95% of the variance in the data

Understanding the principal components derived-

```
df_scaled_PCA.describe()
```

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
count	1.590000e+02	1.590000e+02	1.590000e+02	1.590000e+02	1.590000e+02	1.590000e+02	1.590000e+02
mean	-6.703233e-17	3.281791e-17	2.548625e-17	-3.980045e-17	1.605983e-17	-3.491267e-18	-2.374062e-17
std	2.480103e+00	1.432203e+00	1.117914e+00	9.150836e-01	7.506872e-01	6.180477e-01	5.122136e-01
min	-6.419292e+00	-2.946859e+00	-3.027363e+00	-2.133965e+00	-1.774734e+00	-1.409866e+00	-1.234456e+00
25%	-2.023323e+00	-8.097650e-01	-7.632839e-01	-6.371174e-01	-3.998593e-01	-3.064365e-01	-3.904681e-01
50%	-2.524995e-01	-2.202330e-01	-1.108700e-01	2.133919e-02	-1.054212e-01	-2.551090e-02	4.156915e-03
75%	1.801396e+00	2.519642e-01	6.867493e-01	5.576630e-01	5.209028e-01	3.333498e-01	3.394876e-01
max	6.628805e+00	4.018251e+00	3.435482e+00	2.902558e+00	2.765473e+00	2.975714e+00	1.510192e+00

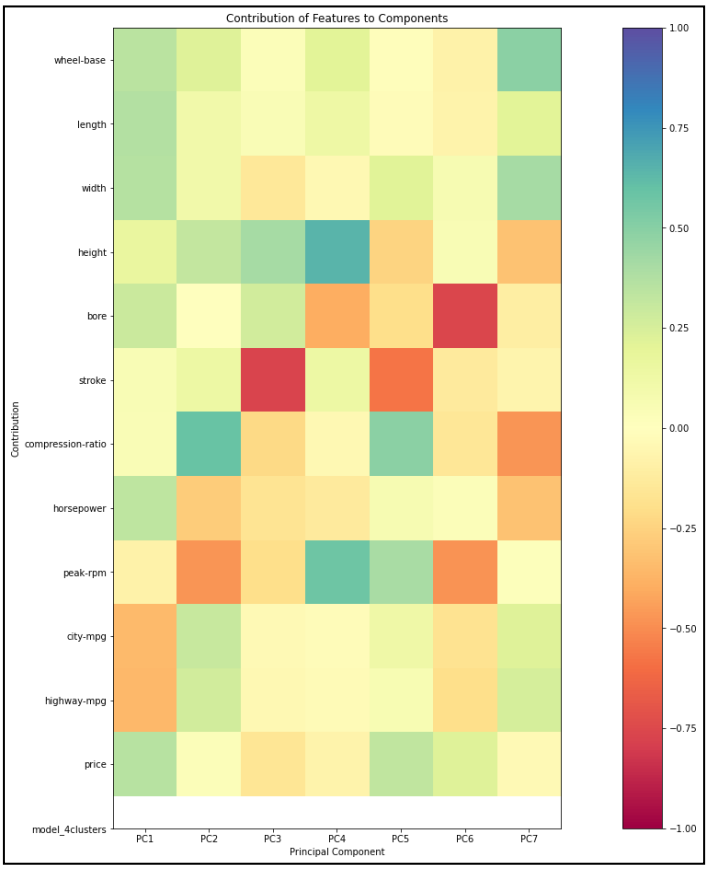
Above mentioned are the seven (07) principal components derived. Further, features like "mean", "std dev", "min", "max", "quartile" is described. From this variance captured by each component can be derived.

Subsequently finding the features of our dataset that are **contributing to PCA/** Principal Components using a heat map: -

Principal components

vs

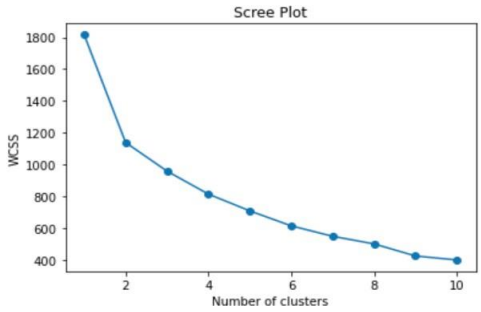
Contribution



MODELING

(i) Building K-Means Cluster Model

```
Counter({0: 159})
Counter({1: 86, 0: 73})
Counter({1: 62, 2: 58, 0: 39})
Counter({2: 57, 1: 53, 3: 40, 0: 9})
Counter({4: 52, 2: 51, 0: 40, 3: 9, 1: 7})
Counter({3: 56, 2: 38, 4: 35, 1: 14, 5: 9, 0: 7})
Counter({0: 50, 2: 43, 4: 27, 1: 15, 6: 9, 5: 8, 3: 7})
Counter({4: 41, 5: 34, 7: 21, 3: 19, 0: 16, 1: 13, 6: 8, 2: 7})
Counter({6: 27, 7: 26, 4: 24, 1: 22, 2: 20, 8: 13, 3: 12, 5: 8, 0: 7})
Counter({2: 31, 9: 26, 0: 23, 7: 20, 1: 19, 6: 13, 3: 8, 8: 7, 5: 7, 4: 5})
cluster_errors: [1816.542520172245, 1139.6459091783045, 959.7125550517167, 815.1460899726046, 709.9943260722448, 615.7755570
505301, 550.4112198297769, 502.4802804332869, 426.64900832966225, 401.5449734180206]
```



Grouped data points into a predefined number of clusters. After looping through all the values of the number of clusters, **WCSS vs the number of clusters is plotted**, i.e the scree plot, we can see as the WCSS decreases, the number of clusters increases.

The plot also suggests that there is a point where adding more clusters does not significantly improve the clustering.

Size-wise = cluster solutions containing 4 and 5 clusters look good

Measuring the performance-

```
Silhouette Coefficient for 3 clusters: 0.236
Silhouette Coefficient for 4 clusters: 0.267
```

From the performance metrics, we can notice that 3 cluster solution is performing better and is also the below point.

```
Calinski-Harabasz index of 3 clusters: 69.633
Calinski-Harabasz index of 4 clusters: 63.248
```

Examining Characteristics with 3 cluster Solution

There are 63 data points in Cluster 0, 44 data points in Cluster 1, and 52 data points in Cluster 2.

```
model_3clusters
0    63
1    44
2    52
dtype: int64
```

s

model_3clusters	length	wheel-base	width	cluster_size
0	-0.02	-0.16	-0.16	63
1	1.18	1.17	1.23	44
2	-0.98	-0.80	-0.85	52

Classification based on cluster size, cluster 0 is the larger cluster containing 63 data points followed by cluster 2 with 52 data points and the last is cluster 1 with 44 data points.

We can notice that cluster 0 and 2 has the lowest length, width & and wheelbase of the automobile model whereas cluster 1 has all three features higher than the other two.

model_3clusters	bore	height	stroke	cluster_size
0	0.16	-0.11	0.02	63
1	0.86	0.57	0.08	44
2	-0.92	-0.34	-0.09	52

Cluster 0 has models with less height but medium diameter of each wheel and number of phas es in engine's cycle of the model.

Cluster 1 has models with highest diameter of each wheel in the automobile (bore), with the highest height and highest number of phases in the engine's cycle(stroke).

Cluster 2 has models with the lowest diameter of each wheel(bore), lowest height of the model, and low number of phases in the engine's cycle(stroke).

model_3clusters	compression-ratio	horsepower	peak-rpm	cluster_size
0	-0.31	-0.02	-0.04	63
1	0.24	1.15	-0.15	44
2	0.17	-0.95	0.17	52

Cluster 1 has models with highest volume of cylinder and chamber (compression-ratio) in the engine with the highest power automobile (horsepower) but low revolutions per minute (peak-rpm)

Cluster 0 has models with the lowest volume of cylinder and chamber in the engine (compression-ratio), low power automobile (horsepower), and low revolutions per minute ((peak-rpm)

Cluster 2 has models with the lowest power automobile (horsepower) but the medium volume of cylinder and chamber of the engine (compression-ratio) with medium revolutions per minute (peak-rpm)

model_3clusters	city-mpg	highway-mpg	price	cluster_size
0	-0.20	-0.18	-0.25	97
1	-1.02	-1.04	1.29	39
2	1.10	1.09	-0.79	23

Cluster 2 has cheapest automobile models (price) with highest scoring car in an average city (city-mpg) & highway (highway-mpg)

Cluster 1 are the most expensive automobile models with the lowest scoring car in an average city or highway as it is obvious that in an average city or highway, the number of expensive cars would be less due to the affordability of the citizens and road infrastructure.

Cluster 0 has an average price automobile model with a medium score in city and highway and the highest number of data points.

(ii) Building Hierarchical Clustering Model

```
clusterid3
0    97
1    39
2    23
dtype: int64

clusterid4
0    59
1    39
2    23
3    38
dtype: int64

clusterid5
0    39
1    38
2    23
3    29
4    30
dtype: int64

clusterid6
0    30
1    38
2    23
3    29
4    30
5     9
dtype: int64

clusterid7
0    23
1    38
2     9
3    29
4    30
5    18
6    12
dtype: int64
clusterid8
0    38
1    29
2     9
3    17
4    30
5    18
6    12
7     6
dtype: int64
```

Performed Hierarchical clustering with 3,4,5,6,7,8,9,10 clusters.

Measuring the performance-

```
Silhouette Coefficient of 3 clusters: 0.301
Silhouette Coefficient of 4 clusters: 0.483
Silhouette Coefficient of 5 clusters: 0.569
Silhouette Coefficient of 6 clusters: 0.633
Silhouette Coefficient of 7 clusters: 0.622
Silhouette Coefficient of 8 clusters: 0.660
Silhouette Coefficient of 9 clusters: 0.622
Silhouette Coefficient of 10 clusters: 0.598
```

```
Calinski-Harabasz index of 3 clusters: 35.890
Calinski-Harabasz index of 4 clusters: 85.549
Calinski-Harabasz index of 5 clusters: 117.445
Calinski-Harabasz index of 6 clusters: 188.320
Calinski-Harabasz index of 7 clusters: 180.083
Calinski-Harabasz index of 8 clusters: 226.160
Calinski-Harabasz index of 9 clusters: 240.381
Calinski-Harabasz index of 10 clusters: 233.249
```

From the above metrics, we can observe that 9 cluster solution is good.

```
clusterid9
0    29
1    10
2     9
3    28
4    30
5    18
6    12
7     6
8    17
dtype: int64
```

Here, we calculated the cluster size, there are 9 clusters, to print the list of customers for each cluster.

Examining Characteristics

clusterid9	length	wheel-base	width	cluster_size
0	-0.57	-0.57	-0.83	29
1	0.84	0.40	1.76	10
2	1.40	2.01	1.87	9
3	1.15	1.04	0.78	28
4	-0.45	-0.50	-0.42	30
5	0.35	0.25	0.20	18
6	-0.31	-0.41	-0.34	12
7	-0.31	-0.38	-0.39	6
8	-1.40	-1.01	-0.99	17

It can be observed that clusters with highest length have highest type of wheelbase as well (clusters 2 and 3), clusters 1 and 5 have medium wheelbase model whereas clusters 0,4,6,7,8 have a low type of wheelbase.

Clusters 1 and 2 have the highest width of the automobile model while clusters 3 & 5 have medium-width models whereas clusters 0,4,6,7,8 have the lowest width automobile models.

Clusters 2 and 3 have automobile models of the highest length while Cluster 1 and 5 have medium-length models whereas Clusters 0,4,6,7,8 have lowest length automobile models.

clusterid9	bore	height	stroke	cluster_size
0	-0.63	0.36	-0.26	29
1	0.44	-0.91	0.45	10
2	0.83	1.07	1.04	9
3	0.89	0.86	-0.59	28
4	-0.58	-0.41	0.67	30
5	0.57	-0.17	0.78	18
6	1.20	-0.07	-2.11	12
7	-0.78	0.30	0.54	6
8	-1.25	-1.22	-0.11	17

Clusters 6 and 3 have the models with highest diameter of each wheel, whereas cluster 8 and 7 have models with lowest diameter of each wheel.

Clusters 2 and 3 have models with the highest height, whereas clusters 8 and 1 have the lowest height models.

Clusters 2 and 5 have models with the greatest number of phases in the engine's cycle and clusters 6 & 3 have models with the least number of phases in the engine's cycle.

clusterid9	compression-ratio	horsepower	peak-rpm	cluster_size
0	-0.28	-0.88	-0.30	29
1	-0.37	1.98	0.19	10
2	2.95	0.34	-1.65	9
3	-0.35	1.00	0.44	28
4	-0.37	-0.07	1.07	30
5	-0.35	0.26	-1.07	18
6	-0.35	-0.31	-0.73	12
7	3.22	-1.28	-1.00	6
8	-0.17	-0.97	0.67	17

It can be observed that Cluster 1 & 3 have models with highest power of engine despite having low volume of chamber of the engine, also cluster 7 having the highest volume of the cylinder, and the chamber in the engine has the lowest power engine.

Cluster 3 & 2 have models with the highest volume of cylinders and chambers in the engine whereas clusters 1 & 4 have models with the lowest volume of cylinders and chambers in the engine.

Cluster 4 has the lowest volume of the chamber in the engine but has the highest revolutions per minute.

clusterid9	city-mpg	highway-mpg	price	cluster_size
0	0.60	0.64	-0.70	29
1	-1.40	-1.41	1.78	10
2	-0.20	-0.51	1.74	9
3	-1.07	-1.01	0.83	28
4	-0.06	-0.01	-0.44	30
5	-0.30	-0.24	0.08	18
6	-0.03	-0.21	-0.50	12
7	1.89	1.93	-0.59	6
8	1.44	1.41	-0.88	17

The more expensive car is the least is the number of cars in an average city or on a highway.

Cluster 7 & 8 have lowest price and highest score in number of cars in an average city and on a highway whereas clusters 1 and 2 have the models with the most expensive cars and the lowest number in an average city and on a highway.

(iii) Executing Clustering

Prepare initial centres using k means++ method

```
initial_centers = kmeans_plusplus_initializer(df_scaled_PCA, 4).initialize()
```

Created Gower metric that will be used for clustering

```
gower_metric = distance_metric(type_metric.GOWER,data=df_scaled_PCA)
```

Created instance of k-means using a specific distance metric

```
kmeans_instance = kmeans(df_scaled_PCA, initial_centers, metric=gower_metric)
```

Run cluster analysis to obtain the results

```
kmeans_instance.process()
clusters = kmeans_instance.get_clusters()
```

Printing the allocated clusters-

```
print(clusters)
```

```
[[0, 6, 7, 13, 18, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 38, 39, 40, 41, 42, 52, 53, 54, 55, 56, 57, 58, 69, 70, 85, 9
7, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 124, 125, 126, 127, 128, 129, 130, 131, 132, 134, 135, 136, 141, 14
3, 144, 146, 147], [1, 2, 3, 4, 5, 32, 43, 44, 45, 46, 47, 48, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 90, 91, 9
2, 93, 94, 95, 96, 133, 137, 138, 139, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158], [8, 9, 10, 11, 12, 14, 15, 1
6, 19, 21, 49, 50, 51, 60, 84, 86, 87, 88, 117, 118, 140, 142, 145], [17, 33, 34, 35, 36, 37, 59, 61, 62, 63, 64, 65, 66, 6
7, 68, 89, 109, 110, 111, 112, 113, 114, 115, 116, 119, 120, 121, 122, 123]]
```

Adding the cluster labels to data frame df for analysis: -

```
cluster_size = df.groupby(['clusterid']).size()
print(cluster_size)
```

```
clusterid
0    60
1    47
2    23
3    29
dtype: int64
```

Performance Measure

```
print("Silhouette Coefficient: %0.3f"% metrics.silhouette_score(df_scaled_PCA, df['clusterid']))
# Silhouette score between -1 and 1
```

```
Silhouette Coefficient: 0.525
```

```
print("Calinski-Harabasz index: %0.3f"% metrics.calinski_harabasz_score(df_scaled_PCA, df['clusterid']))
```

```
Calinski-Harabasz index: 112.981
```

Examining Characteristics

clusterid	length	wheel-base	width	cluster_size
0	-0.18	-0.25	-0.22	60
1	1.13	1.09	1.20	47
2	-1.11	-0.85	-0.83	23
3	-0.57	-0.57	-0.83	29

Classification based on cluster size, cluster 0 is the larger cluster containing 60 data points followed by Cluster 1 with 47 data points Cluster 3 with 29 data points and last is Cluster 2 with 23 data points.

We can notice that cluster 2 and 3 has the lowest length, width, and wheelbase of the automobile model whereas cluster 1 has all three features higher than the other two.

clusterid	bore	height	stroke	cluster_size
0	0.12	-0.27	0.15	60
1	0.78	0.52	-0.06	47
2	-1.13	-0.82	0.06	23
3	-0.63	0.36	-0.26	29

Cluster 1 has models with the highest diameter of each wheel in the automobile (bore), with the highest height but a low number of phases in the engine's cycle(stroke).

Cluster 2 has models with the lowest diameter of each wheel(bore), the lowest height of the model but a moderate number of phases in the engine's cycle(stroke).

Cluster 0 has models with less height but a moderate diameter of each wheel and the highest number of phases in the engine's cycle of the model.

Cluster 3 has models with moderate height but the lowest diameter of each wheel in the automobile.

clusterid	compression-ratio	horsepower	peak-rpm	cluster_size
0	-0.36	-0.02	0.07	60
1	0.28	1.08	-0.01	47
2	0.71	-1.06	0.23	23
3	-0.28	-0.88	-0.30	29

Cluster 2 has models with highest volume of cylinder and chamber (compression-ratio) in the engine and low revolutions per minute (peak-rpm) but the lowest power automobile (horsepower).

Cluster 1 has models with a moderate volume of cylinder and chamber (compression-ratio) in the engine and low revolutions per minute (peak-rpm) but has the highest power automobile (horsepower).

Cluster 0 has models with the lowest volume of cylinder and chamber in the engine (compression-ratio), low power automobile (horsepower) but moderate revolutions per minute (peak-rpm).

Cluster 3 has models with low-power automobiles (horsepower), low volume of cylinder, and chamber of the engine (compression-ratio) with lowest revolutions per minute (peak-rpm).

clusterid	city-mpg	highway-mpg	price	cluster_size
0	-0.12	-0.12	-0.3	60
1	-0.98	-1.00	1.2	47
2	1.56	1.55	-0.8	23
3	0.60	0.64	-0.7	29

Cluster 2 has the cheapest automobile models (price) with the highest scoring car in an average city (city-mpg) and highway (highway-mpg).

Cluster 1 is the most expensive automobile model with the lowest scoring car in an average city or highway as it is obvious that in an average city or highway, the number of expensive cars would be less due to the affordability of the citizens and road infrastructure.

Cluster 3 has a moderate number of cars in the average city and highway with moderate pricing.

Cluster 0 has a low-price automobile model with a low score in city and highway and the highest number of data points.

CONCLUSION

The **K-Means and Hierarchical Clustering** methods yielded similar results in terms of the number of optimal clusters.

Cluster 1 consistently appeared to represent higher-end automobile models with larger wheels, higher-power engines, and higher prices.

Cluster 2 tended to represent more economical models with smaller dimensions and lower power.

Cluster 0 (K-Means) and Cluster 3 (Hierarchical) had somewhat intermediate characteristics.