CLUSTERING THE AUTOMOTIVE

Runkang Yang[*], **Panxin Tao**[†], **Zhuoyang Bu**[†]
{yangrk2022,taopx2022,buzy2022}@shanghaitech.edu.cn

ABSTRACT

Clustering analysis plays a pivotal role in identifying patterns within complex datasets and can be particularly valuable for competitor analysis in automotive industry, This study addresses the challenge of identifying competing vehicles for Volkswagen. We first conduct Principal Component Analysis (PCA) and AutoEncoder for dimensionality reduction, followed by K-means and hierarchical clustering for product segmentation. Through qualitative analysis and quantitative comparison, we empirically find that *the combination of AE for feature extraction and HC for clustering appears to be the most effective approach*. It not only achieves the best clustering evaluation metrics but also results in a logical grouping of vehicles that reflects market segmentation. This research not only advances the application of clustering methods in automotive sales industry but also illustrates the potential of deep learning techniques used in high-dimensional data spaces.

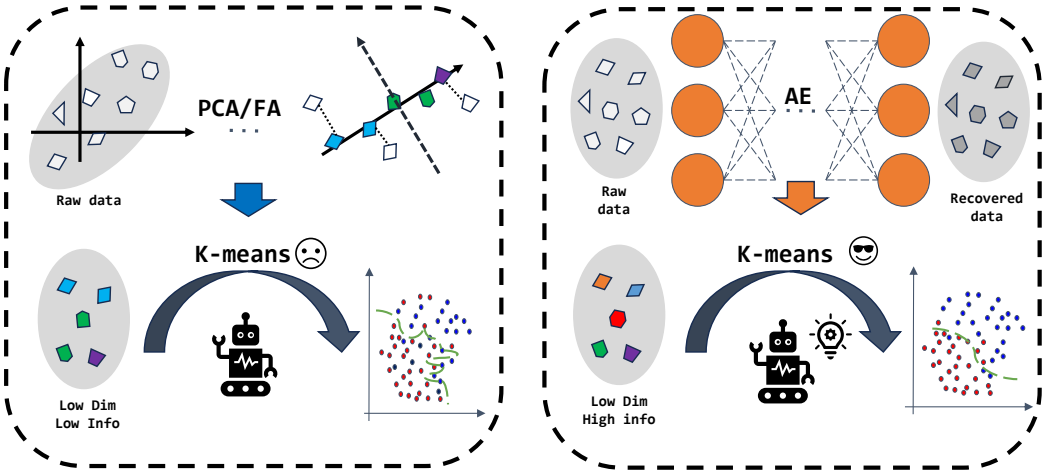


Figure 1: **Traditional PCA-based dimensionality reduction (left) and neural network-based autoencoder dimensionality reduction (right).** Traditional PCA/FA relies on linear transformations, which inherently possess limitations in capturing nonlinear patterns in data. In contrast, neural network-based autoencoders leverage nonlinearity and hierarchical feature extraction, enabling them to handle complex data distributions effectively, thus contributes to downstream clustering tasks.

1 INTRODUCTION

Clustering analysis is a widely used technique for identifying patterns in large and complex datasets, making it particularly useful in fields such as market segmentation, product grouping, and competitor analysis. In the automotive industry, clustering can be an invaluable tool for identifying groups of similar vehicles, which can inform product positioning, pricing strategies, and competitive analysis.

To perform effective clustering, it is crucial to reduce the dimensionality of the data while preserving its key structure. Traditional methods, such as Principal Component Analysis (PCA), have long been

[*]Project Leader.
[†]Equal contribution.

used for this purpose by transforming the original feature space into a set of uncorrelated components that capture the most significant variance in the data (Pearson, 1901; Hotelling, 1933). However, PCA often struggles to capture non-linear relationships, which are prevalent in real-world datasets like automotive data in this problem.

To address these limitations, more recent methods have leveraged deep learning techniques, particularly AutoEncoders (AE), for dimensionality reduction. AutoEncoders are neural network-based models that learn a compressed representation of the input data in a lower-dimensional space, capable of preserving complex, non-linear patterns (Hinton & Salakhutdinov, 2006; Goodfellow et al., 2016). By reducing the dimensionality of the data using AutoEncoders, we can extract more meaningful representations that can improve the performance of clustering algorithms.

After dimensionality reduction, clustering algorithms such as K-means (MacQueen, 1967; Jain, 2010) and hierarchical clustering (Lance & Williams, 1967; Murtagh & Contreras, 2012) are commonly used to group vehicles based on their feature similarities. K-means is a well-known clustering algorithm that works efficiently on large datasets and produces a predefined number of clusters based on the Euclidean distance between points. However, K-means requires the number of clusters to be specified in advance and may struggle to detect clusters with non-spherical shapes. Hierarchical clustering, on the other hand, builds a tree of clusters and provides a more flexible approach to identifying relationships between data points. After that, we quantitatively evaluate the clustering quality through internal metrics like the Silhouette Score(SC) (Rousseeuw, 1987), Calinski-Harabasz Index(CH) (Caliński & Harabasz, 1974), Davies-Bouldin Index(DB) (Davies & Bouldin, 1979) and Dunn Index(DI) (Dunn, 1974).

This paper aims to investigate the effectiveness of combining AutoEncoder-based dimensionality reduction with clustering methods for automotive competitor analysis. Specifically, We detailed our methodology in section 2, mainly through comparing PCA and AutoEncoder for dimensionality reduction, followed by clustering with K-means and hierarchical methods. And then present the numerical result in section 3, the conclusion in section 4.

2 METHODOLOGY

This section details the methodology employed for the analysis of the automotive dataset. The approach includes data preprocessing, feature selection, and training of clustering models using two dimensionality reduction techniques: Principal Component Analysis (PCA) and AutoEncoders (AE). Two clustering algorithms, K-means and hierarchical clustering, are then applied to identify competitor vehicles for Volkswagen, as shown in Figure 1, 3.

2.1 FEATURE ENGINEERING

The first step involves preprocessing the dataset to ensure that it is ready for dimensionality reduction and clustering. This includes handling missing values, detecting and treating outliers, normalizing numerical data, and converting categorical variables into a format suitable for machine learning models.

The dataset has no missing value and except for the Car_ID attribute which is only an index, we will divide the remaining attributes into two kind: categorical attributes and numerical attributes and handle them respectively.

2.1.1 DATA CLEANING

This step involves correcting any errors present in the dataset, such as incorrect entries in categorical fields (e.g., misspelled car names or incorrect labels). These issues were manually inspected and corrected by cross-referencing with external sources. And in this problem, there is a typo in line 184 where *vokswagen* refers to *volkswagen*, and *vw* is short for *volkswagen*, we have manually identified and corrected them.

2.1.2 ONE-HOT ENCODING OF CATEGORICAL VARIABLES

In this section, we will handle the 10 categorical attributes to fit them into the subsequent models.

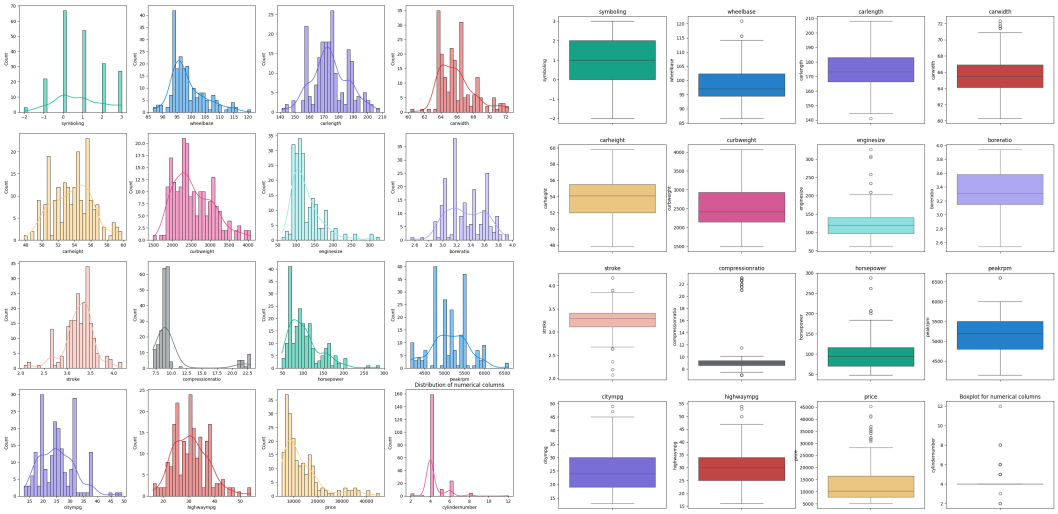


Figure 2: Data distribution and boxplot.

First of all, we notice that the attributes doornumber and cylindernumber are expressed as number. We argue that the cylindernumber attribute has numerical significance and can be treated as a numerical attribute. On the other hand, the value of doornumber barely matters the choice of competitors, which should be treated as a categorical attribute.

We also notice that the CarName attribute has 147 different categories which is quite large and contains little useful information. Therefore, it will be extracted and handled as categorical attribute CarBrand subsequently. We observed misspelled or abbreviated categories in CarName, which should be corrected when extracting the CarBrand attribute.

After the data cleaning, one-hot encoding is applied to all the categorical attributes. We use one-hot encoding instead of label encoding because all the attributes have no ordinal relation. For a categorical variable C with N possible categories, each category was encoded as a binary vector. For instance, if $C = \{\text{convertible}, \text{hatchback}, \text{sedan}\}$, the one-hot encoding would convert these categories into the following binary vectors:

$$\text{convertible} = [1, 0, 0], \quad \text{hatchback} = [0, 1, 0], \quad \text{sedan} = [0, 0, 1]$$

2.1.3 Outlier Detection and Handling

The remaining 15 numerical attributes and the cylindernumber attribute mentioned before are treated as numerical features. Outliers in numerical features can significantly distort the analysis and modeling results. To detect and handle outliers, we utilized the z-score method, as shown in Figure 2. For a given numerical feature x_i, the z-score is defined as: $z_i = \frac{x_i - \mu}{\sigma}$ where μ is the mean and σ is the standard deviation of the feature. Any data point with a z-score greater than 3 or less than -3 was considered an outlier and replaced by the median value of the corresponding feature $\tilde{x}$: $x_i \leftarrow \tilde{x}$

2.1.4 Normalization of Numerical Features

To eliminate the influence of different scales in numerical features, all numerical attributes were standardized. Since we notice quite a number of the numerical attributes in this dataset barely follow a normal distribution, MinMaxScalar is favored instead of StandardScalar. Actually, a much better performance with minmaxscalar is observed in subsequent result. This transformation helps reserve the distribution and relative distance of the origin data and ensures that the numerical attributes are within [0,1]. The MinMaxScaler transformation is defined as:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}},$$

where $x_{\min}$ and $x_{\max}$ are the minimum and maximum values of the attribute, respectively. This transformation maps all values of x to the range $[0, 1]$.

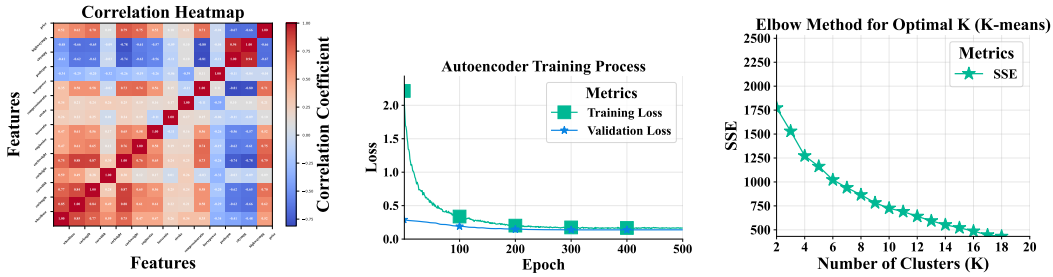


Figure 3: **Our propused *Data Selection - Model Training - Application of Clustering Algorithm* Pipeline.** The left figure shows the heatmap of correlations between different data points in the raw data. Middle figure shows the change in the autoencoder loss across training epochs, while the rightmost figure illustrates the process of selecting the optimal parameter K for clustering. And in this problem, we choose K=8.

2.2 Feature Selection

After preprocessing, the next step is to identify the most relevant features for clustering. We begin by visualizing the correlation between numerical features using a heatmap, where the correlation coefficient r between two features x_i and x_j is computed as:

$$r(x_i, x_j) = \frac{\text{cov}(x_i, x_j)}{\sigma_{x_i}\sigma_{x_j}}$$

where $\text{cov}(x_i, x_j)$ is the covariance of the features and σ_{x_i} and σ_{x_j} are their respective standard deviations. Features with high correlation ($|r| > 0.85$) were considered redundant and removed.

2.2.1 Dimensionality Reduction

We applied two methods for dimensionality reduction: Principal Component Analysis (PCA) and AutoEncoders (AE). Both methods aim to reduce the number of features while retaining as much variance (PCA) or meaningful information (AE) as possible.

2.2.2 Principal Component Analysis (PCA)

PCA (Pearson, 1901; Hotelling, 1933) is a linear dimensionality reduction technique that projects the data into a new space such that the first principal component captures the maximum variance in the data, the second principal component captures the second largest variance, and so on. Mathematically, PCA involves solving the eigenvalue problem for the covariance matrix $\mathbf{C}$ of the dataset:

$$\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

where λ_i are the eigenvalues and $\mathbf{v}_i$ are the eigenvectors. We selected the number of principal components such that the cumulative explained variance was 95%. This resulted in approximately 18 principal components for our dataset, which were used for further analysis.

2.2.3 AutoEncoder (AE)

AutoEncoders (Hinton & Salakhutdinov, 2006; Goodfellow et al., 2016) are a class of artificial neural networks that aim to learn a compressed, low-dimensional representation of the input data. The AE consists of an encoder network that maps the input data x to a lower-dimensional space z, and a decoder network that reconstructs the original data from this lower-dimensional representation. The objective of training an AE is to minimize the reconstruction error:

$$\mathcal{L}(\hat{x}, x) = \|\hat{x} - x\|^2$$

For our dataset, we empirically chose the hidden space dimension to match the number of components from PCA (i.e., 20). The architecture and hyperparameters of the AE, including the number of layers and neurons, are summarized in Table 1.

Hyperparameter	Value
Active Function	ReLU
Encoding Dimension	20
Number of Layers	7
Batch Size	128
Epochs	500
Learning Rate	1e-3
Loss Function	MSE
Optimizer	Adam
Dropout Rate	0.2

Table 1: Autoencoder Architecture and Hyperparameters

2.3 Clustering Algorithms

After reducing the feature space to approximately 20 dimensions, we applied two clustering methods to identify groups of similar vehicles: K-means and hierarchical clustering.

2.3.1 K-means Clustering

K-means clustering (MacQueen, 1967; Jain, 2010) is a centroid-based clustering algorithm that partitions the dataset into K clusters, where each data point belongs to the cluster whose centroid is nearest. The objective is to minimize the within-cluster sum of squared errors (SSE):

$$SSE(K) = \sum_{i=1}^{N} \sum_{k=1}^{K} \mathbf{1}(c_i = k) \|x_i - \mu_k\|^2$$

where c_i is the cluster assignment for data point x_i, μ_k is the centroid of cluster k, and $\mathbf{1}(\cdot)$ is the indicator function. To determine the optimal number of clusters K, we used the elbow method, plotting the SSE as a function of K and selecting the K where the rate of decrease in SSE starts to slow, as shown in figure 3.

2.3.2 Hierarchical Clustering

Hierarchical clustering (Lance & Williams, 1967; Murtagh & Contreras, 2012) builds a tree of clusters by either iteratively merging smaller clusters (agglomerative) or splitting larger clusters (divisive). The agglomerative approach was used for this problem, where at each step, the two clusters that are closest according to a chosen distance metric are merged. The proximity between clusters can be measured using various linkage criteria, such as single linkage, complete linkage, or average linkage. The distance between two clusters A and B is given by:

$$d(A, B) = \min_{x \in A, y \in B} \|x - y\|$$

We used a dendrogram to visually inspect the hierarchical relationships between the clusters, as shown in Figure 4, then we selected the appropriate number of clusters based on the structure of the tree. For this problem, we finally choose 8 clusters (Distance around 10).

3 Numerical Results

3.1 Model Evaluation

To evaluate the performance of the clustering algorithms, we used internal validation metrics in Table 2 in Appendix A[1]. The table compares the performance of clustering methods (Kmeans and Hierarchical Clustering, HC) on features extracted using PCA and Autoencoder (AE) across four evaluation metrics: Silhouette Score(SC) (Rousseeuw, 1987), Calinski-Harabasz Index(CH) (Caliński & Harabasz, 1974), Davies-Bouldin Index(DB) (Davies & Bouldin, 1979) and Dunn Index(DI) (Dunn, 1974).

[1]Due to space limitations and for the sake of aesthetic formatting.

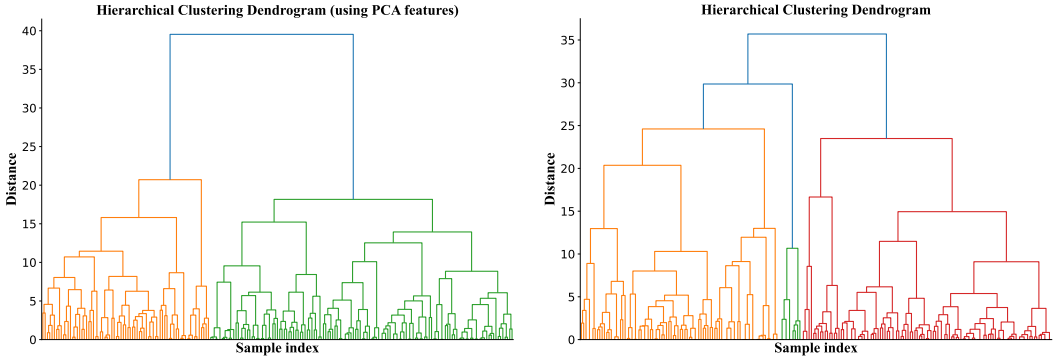


Figure 4: **Dendrogram of the Bottom-Up Hierarchical Clustering Process on the Low-Dimensional Features Generated by *PCA* and *AE*.** From the figure, it is evident that a reasonable choice for the number of clusters is around 10, corresponding to a distance of approximately 10.

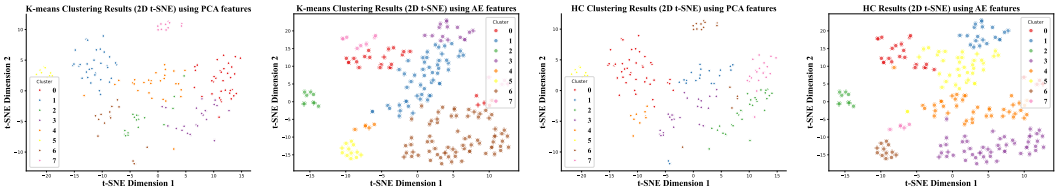


Figure 5: **2D Visualization of Features Generated by *Kmeans* and *Hierarchical Clustering* after Dimensionality Reduction Using *PCA* and *AE*.** The clustering results are visually clearer when using *AE* for dimensionality reduction, as samples within the same cluster are more densely packed.

3.2 Clustering Results

For the clustering results, see Figure 5, Tabel 3,4,5 and 6 in Appendix A. Based on the results of the five tables, we find that for feature extraction, Autoencoder (AE) outperforms Principal Component Analysis (PCA) in terms of clustering quality. For example, AE combined with hierarchical clustering (HC) yields the best SC (0.3864) and DB (1.0059), suggesting that this method effectively separates data points.

In the clustering methods, K-means slightly outperforms HC when using AE features, achieving a higher CH score. However, HC with PCA features achieves the highest Dunn Index (DI) score (0.3674), suggesting that it can identify well-separated clusters despite its lower SC and CH.

Examining the cluster composition, AE-based features result in a more meaningful distribution of vehicles across clusters, with distinct median prices and a better grouping of VW vehicles. For example, clusters generated with AE features using K-means place most VW vehicles in clusters with moderate median prices (e.g., $9,717.14$ and $10,445$), aligning with VW's market position. In contrast, PCA-based clusters show less consistent grouping, with VW vehicles distributed across multiple clusters without clear price patterns.

Overall, the combination of AE for feature extraction and HC for clustering not only achieves the best clustering evaluation metrics but also results in a logical grouping of vehicles that reflects market segmentation. And we use this result (i.e. AE+HC) for next sections's analysis.

4 Conclusion

We selected cars that *belong to the same cluster as Volkswagen vehicles and have prices within 10% of the average price of Volkswagen vehicles* as the final competitor models and provide the competitors in the .csv files in our `Supplementary materials`, please refer to cars_within_10_percent.csv. And we further analysis the attributes of the cars after clustering, as shown in Figure 6 in Appendix A.

According to these figures. The competitors of Volkswagen share several characteristics that align with the positioning of affordable and practical urban vehicles. These models typically fall into

the sedan or hatchback categories, emphasizing compactness and maneuverability, making them well-suited for city driving. Furthermore, these vehicles are generally offered at a lower price point, appealing to cost-sensitive consumers who prioritize economic value over premium features. Their fuel efficiency, reflected in high city and highway mileage, further solidifies their appeal to urban drivers.

However, there are notable differences that set Volkswagen apart from its competitors. Many competing models prioritize lightweight construction, which, while beneficial for fuel efficiency, may compromise perceived safety and structural integrity. In contrast, Volkswagen is often recognized for its robust build and focus on safety, offering a significant edge in terms of quality assurance. Moreover, the competitors' emphasis on fuel efficiency often results in lower engine power, with many models delivering modest horsepower. Volkswagen, on the other hand, strives to strike a balance between performance and efficiency, catering to drivers who value a more dynamic driving experience. Another area of differentiation lies in interior design and comfort. Competitors frequently offer smaller cabin dimensions, which may suffice for short urban commutes but could reduce comfort on longer journeys.

REFERENCES

Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27, 1974. doi: 10.1080/03610927408827101.

David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979. doi: 10.1109/TPAMI.1979.4766909.

J. C. Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4(1): 95–104, 1974. doi: 10.1080/01969727408546059.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. Available at https://www.deeplearningbook.org/.

Geoffrey E. Hinton and Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. doi: 10.1126/science.1127647.

Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933. doi: 10.1037/h0071325.

Anil K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010. doi: 10.1016/j.patrec.2009.09.011.

G. N. Lance and W. T. Williams. A general theory of classificatory sorting strategies: I. hierarchical systems. *The Computer Journal*, 9(4):373–380, 1967. doi: 10.1093/comjnl/9.4.373.

J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pp. 281–297. University of California Press, 1967.

Fionn Murtagh and Pedro Contreras. Algorithms for hierarchical clustering: An overview. *WIREs Data Mining and Knowledge Discovery*, 2(1):86–97, 2012. doi: 10.1002/widm.53.

Karl Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(11):559–572, 1901. doi: 10.1080/14786440109462720.

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. doi: 10.1016/0377-0427(87)90125-7.

A APPENDIX

Feature	Cluster	Evaluation Metric			
		SC	CH	DB	DI
PCA	Kmeans	0.2093	29.9797	1.6294	0.2707
	HC	0.2010	28.0231	1.6160	**0.3674** ↑
AE	Kmeans	0.3746	**78.2700** ↑	1.0583	0.0867
	HC	**0.3864** ↑	75.5768	**1.0059** ↓	0.1957

Table 2: Clustering Methods and Evaluation Metrics.

	K-means Cluster (PCA feature)							
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Number of cars	21	46	35	21	27	25	18	12
Median price	11541	8279	6817	14670	22163	12175	22326	20602
Number of VW	2	1	0	0	0	8	0	1
VW Median	10788	12290	-	-	-	9153	-	13845

Table 3: K-means clusters using PCA features

	K-means Cluster (AE feature)							
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Number of cars	27	44	17	39	50	14	11	3
Median price	21624	15348	7428	7862	9843	17659	23079	12145
Number of VW	0	2	0	0	7	3	0	0
VW Median	-	10788	-	-	9717	10445	-	-

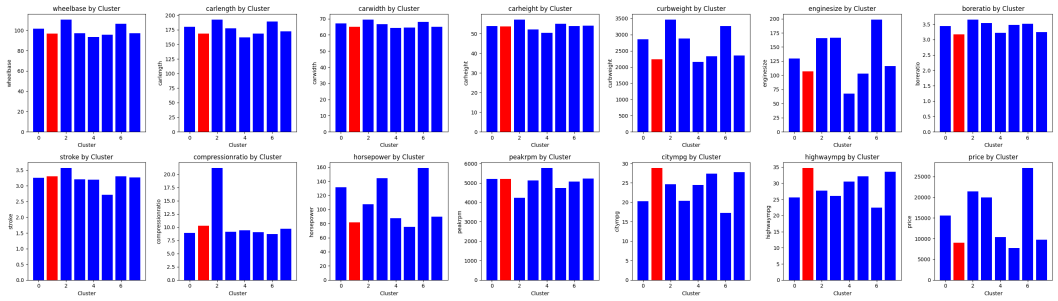
Table 4: K-means clusters using AE features

	Hierarchical Cluster (PCA feature)							
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Number of cars	47	25	21	38	19	20	16	19
Median price	7004	15092	15934	8816	13251	18938	11608	27859
Number of VW	0	0	4	1	5	0	2	0
VW Median	-	-	9778	12290	9591	-	10788	-

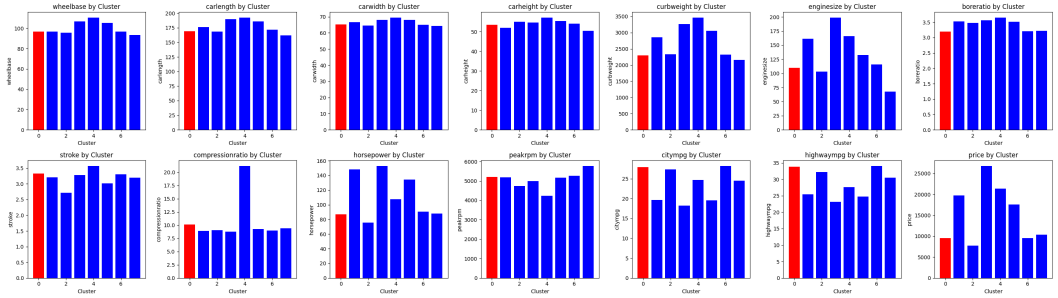
Table 5: Hierarchical clusters using PCA features

	Hierarchical Cluster (AE feature)							
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Number of cars	61	13	52	17	14	28	3	17
Median price	9081	19505	10559	18530	20958	21216	12145	7428
Number of VW	2	1	9	0	0	0	0	0
VW Median	10788	9495	9984	-	-	-	-	-

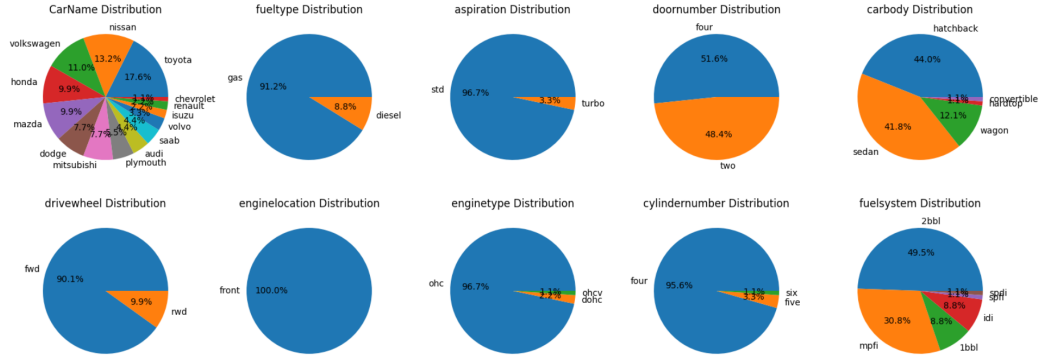
Table 6: Hierarchical clusters using AE features



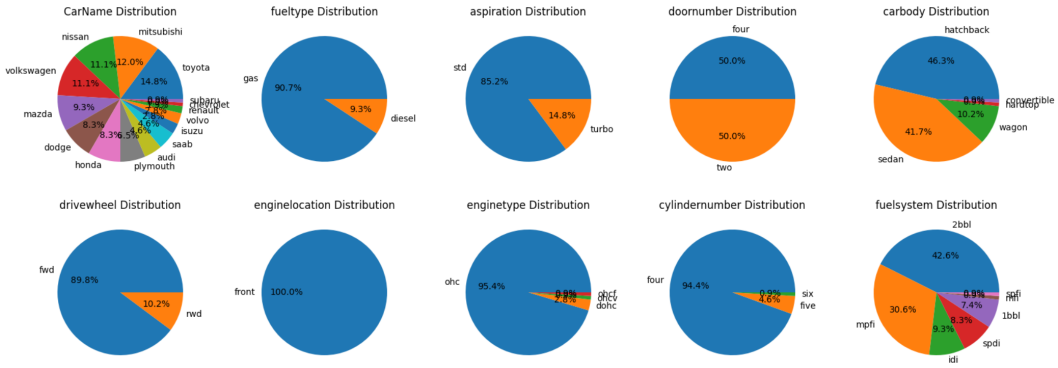
(a) Bar chart analysis using kmeans clustering.



(b) Bar chart analysis using hierarchical clustering.



(c) Pie chart analysis using kmeans clustering.



(d) Pie chart analysis using hierarchical clustering.

Figure 6