Clustering the automotive

Zhengtao Han,*Zeping Ruan, Yiyang Tan
{hanzht2022,ruanzp2023,tanyy2023}@shanghaitech.edu.cn

Abstract

This project investigates the identification of competing vehicle models for Volkswagen using unsupervised machine learning. Dimensionality reduction techniques—PCA, AutoEncoders (AE), and Stacked AutoEncoders (SAE)—are combined with K-means and hierarchical clustering to segment 205 car models described by 26 features. Results indicate that AE-based features with hierarchical clustering yield the most coherent groupings and highest evaluation metrics, demonstrating the effectiveness of deep learning-based clustering for automotive market analysis.

1 Introduction

Identifying competitors in the automotive market has traditionally relied on qualitative analysis. This project uses unsupervised learning to cluster car models and identify Volkswagen's competitors based on data. Our main contributions are:

- A complete pipeline for competitor identification using unsupervised learning on automotive data.
- Systematic data preprocessing: data cleaning, outlier handling, encoding, and normalization.
- Implementation of PCA, AE, and SAE for dimensionality reduction, followed by K-means and hierarchical clustering.
- Comprehensive evaluation with various metrics to select optimal cluster configurations.

2 Methodology

This section outlines the methodology for analyzing the automotive dataset, involving data preprocessing, dimensionality reduction using PCA, AutoEncoders (AE), and Stacked AutoEncoders (SAE), followed by K-means and hierarchical clustering to identify vehicles competing with Volkswagen.

2.1 Data Preprocessing

The dataset was preprocessed to ensure suitability for dimensionality reduction and clustering, including outlier handling, normalization, and encoding categorical variables.

2.1.1 Data Cleaning

We inspected the dataset, correcting spelling errors in categorical fields like *CarName* to ensure data consistency for analysis.

2.1.2 One-hot Encoding on Categorical Variables

The *CarName* field was split to extract *CarBrand*, with misspellings corrected. One-hot encoding was applied to categorical features (*CarBrand*, *fueltype*, *aspiration*, *doornumber*, *carbody*, *drivewheel*, *enginelocation*, *enginetype*, *cylindernumber*, *fuelsystem*) to avoid ordinal assumptions.

*Project Leader.

2.1.3 Outlier Handling

Outliers were detected in numerical features (*wheelbase*, *enginesize*, *stroke*, *compressionratio*, *horsepower*, *peakrpm*, *citympg*, *highwaympg*, *price*) using z-scores:

$$z_i = \frac{x_i - \mu}{\sigma}$$

Data points with $|z_i| > 3$ were replaced with the feature's median.

2.1.4 Normalization on Numerical Features

Numerical features (*symboling*, *wheelbase*, *carlength*, *carwidth* ...) were normalized to [0, 1] using *MinMaxScaler* (Detailed distribution is in Appendix A.0.1)

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

2.2 Dimensionality Reduction

2.2.1 Principal Component Analysis (PCA)

PCA [6, 3] is a linear dimensionality reduction method that projects data into a new space, maximizing variance along principal components. It solves the eigenvalue problem for the covariance matrix $\mathbf{C}$: $\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i$, where λ_i and $\mathbf{v}_i$ are eigenvalues and eigenvectors.

2.2.2 AutoEncoder (AE)

AutoEncoders [2, 1] are neural networks that learn compact data representations unsupervised. An encoder maps input x to latent representation z, and a decoder reconstructs $\hat{x}$. The model minimizes reconstruction error: $\mathcal{L}(\hat{x}, x) = \|\hat{x} - x\|^2$. Our AE uses a multi-layer encoder (input, 256, 128, 64, 16) with a symmetric decoder. Details are in Appendix A.0.2.

2.2.3 Stacked AutoEncoder (SAE)

Stacked AutoEncoders (SAEs) [7] extend AutoEncoders by training multiple AEs hierarchically, with each layer pretrained to compress data into lower dimensions (256, 64, 16), followed by end-to-end fine-tuning. Details are in Appendix A.0.2. Our ablation study shows that training without pretraining fails to converge, underscoring the pretraining-fine-tuning approach's effectiveness.

2.3 Metrics for Selecting the Optimal Cluster Number

The evaluation was conducted by calculating the Silhouette Score, Davies-Bouldin Score, and Inertia for k values ranging from 2 to 14. To facilitate comparison, each metric was normalized to a [0, 1] range. The details of metrics and result can be found in Appendix A.0.4.

2.3.1 Combined Score

A combined score was computed as a weighted average of the normalized metrics to balance their contributions:

$$\texttt{combined\_scores} = 0.4 \cdot \texttt{norm\_sil} + 0.3 \cdot \texttt{norm\_db} + 0.3 \cdot \texttt{norm\_inertia}$$

The optimal k was selected as the value that maximizes the combined score:

$$\text{optimal\_k} = \arg \max_{k \in \{2,\ldots,14\}} (\texttt{combined\_scores}(k))$$

This approach ensures a comprehensive evaluation of clustering quality, leveraging multiple metrics to identify the most suitable number of clusters for the dataset.

2.4 Clustering Algorithms

After reducing features to 16 dimensions, we applied K-means and hierarchical clustering to group similar vehicles. Detailed algorithms can be found in Appendix A.0.3.

2.4.1 K-means Clustering

K-means [5, 4] partitions data into K clusters by initializing centroids, assigning points to the nearest centroid, and updating centroids iteratively until convergence. The objective is to minimize the within-cluster sum of squared errors (SSE):

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

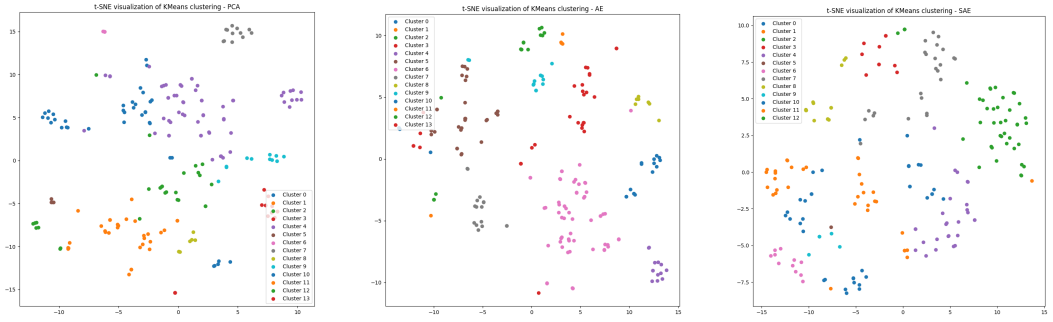


Figure 1: 2D Visualization of Features Clustered by *Kmeans* after Dimensionality Reduction Using *PCA*, *AE* and *SAE*

2.4.2 Hierarchical Clustering

Hierarchical clustering [8] builds a cluster hierarchy using agglomerative Ward's linkage. Clusters merge to minimize:

$$\Delta(C_i, C_j) = \frac{|C_i| \cdot |C_j|}{|C_i| + |C_j|} \cdot \|\text{centroid}(C_i) - \text{centroid}(C_j)\|^2$$

A dendrogram visualizes the hierarchy is shown in Appendix A.0.3.

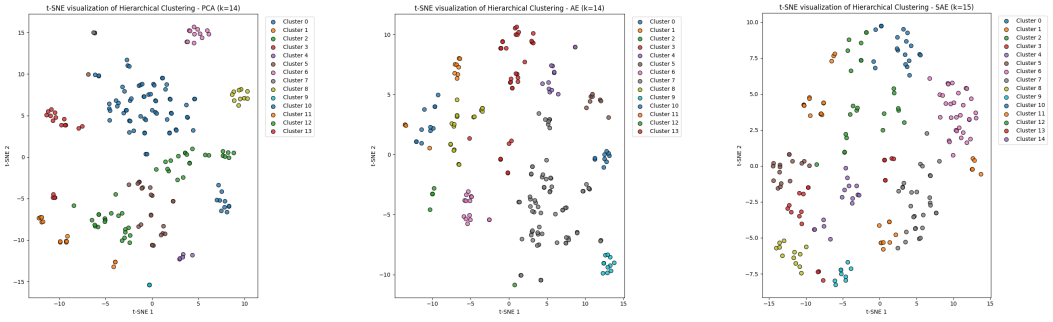


Figure 2: 2D Visualization of Features Clustered by *Hierarchical clustering* after Dimensionality Reduction Using *PCA*, *AE* and *SAE*

3 Results

3.1 Clustering results

In this section, we display the results obtained after applying K-means and hierarchical clustering. We used t-SNE (2D) to project the high-dimensional data into a two-dimensional space for visualization. The resulting cluster assignments and their spatial distributions are shown in Figure 1 and Figure 2, respectively. From these plots we observe:

- **K-means (Figure 1):**

  - PCA-reduced data produces loosely defined, overlapping groups around the origin.
  - AE-reduced data yields compact, well-separated "islands," revealing meaningful nonlinear structure.
  - SAE-reduced data (with $k = 15$) shows many small, sparsely populated clusters, indicating potential over-fragmentation.
- **Hierarchical clustering (Figure 2):**
  - PCA features give some cohesive regions but retain central overlap.
  - AE features produce dense, clearly separated clusters under Ward's linkage.
  - SAE features again create numerous tiny peripheral clusters, reducing interpretability.

3.2 Numerical analysis

We complement the visual assessment with quantitative counts and price alignment for Volkswagen (VW).

Table 1: Volkswagen counts in each cluster for both clustering schemes.

Clustering	Embedding	Cluster: VW count
K-means	PCA	2: **2**, 4: **9**, 9: **1**
	AE	13: **12**
	SAE	4: **1**, 6: **1**, 7: **1**, 11: **9**
Hierarchical	PCA	2: **8**, 8: **4**
	AE	4: **4**, 7: **7**, 13: **1**
	SAE	2: **1**, 7: **1**, 8: **1**, 11: **5**, 14: **4**

Price-median alignment under hierarchical clustering (Detailed data can be found in Appendix A.0.5):

- **PCA+HC:** VW medians in Cluster 2 (9,987.5) and Cluster 8 (8,745.0) match cluster medians (13,622.5 and 8,745.0) within acceptable range.
- **AE+HC:** Small deviations in Cluster 4 (VW 8,745.0 vs. 9,395.0) and Cluster 13 (VW 11,595.0 vs. 11,374.0); larger mismatches in Clusters 7.
- **SAE+HC:** VW in Cluster 2 (11,595.0 vs. 11,722.5), Cluster 14 (8,745.0 vs. 8,745.0) and Cluster 11 (8,495.0 vs. 9,991.5) differ by <10%, demonstrating the tightest price-based grouping.

We found that *AE+HC* strikes the best balance between clear cluster structure and sound metrics, while *SAE+HC* gives the tightest price alignment ($< 10\%$). Based on this, we recommand use *AE+HC* for broad segmentation and *SAE+HC* for price-focused competitor selection.

4 Conclusion

We introduced a two-stage pipeline that first use PCA, Autoencoder (AE) or Stacked Autoencoder (SAE) for dimension reduction, and then clusters these embeddings with K-means or hierarchical clustering (HC).

Key findings

- *AE* form the most compact, well-separated clusters; PCA is adequate but linear, while SAE can over-fragment when k is large.
- *AE+HC* delivers the clearest overall structure; *SAE+HC* gives the tightest price alignment for Volkswagen (median errors $< 10\%$).

References

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

[2] Geoffrey E. Hinton and Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[3] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

[4] Anil K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[5] 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, pages 281–297. University of California Press, 1967.

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

[7] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre Lamblin. Stacked denoising autoencoders: Learning useful representations in a hierarchical fashion. *Journal of Machine Learning Research*, 11:1529–1560, 2010.

[8] Joe H. Jr. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963. Introduced Ward's minimum-variance agglomerative clustering.

A Appendix

A.0.1 Data Distribution

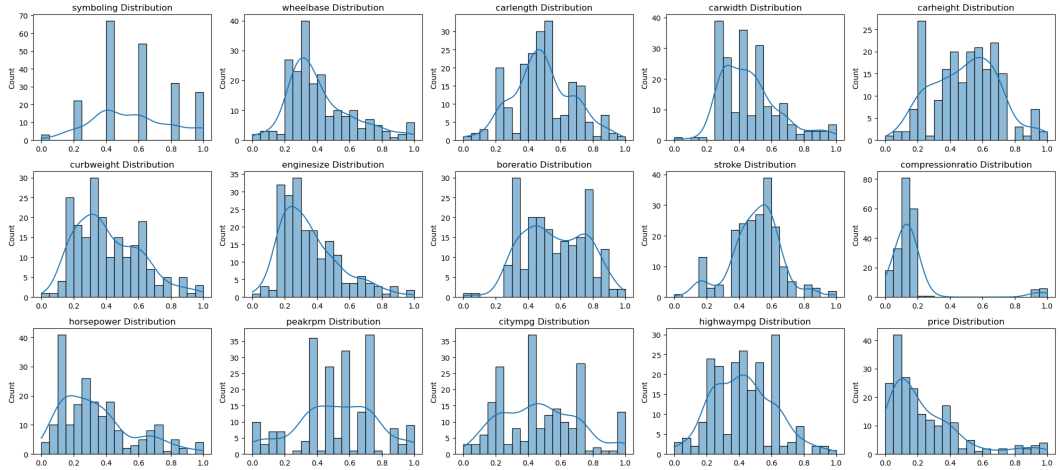


Figure 3: Data distribution on preprocessed Data

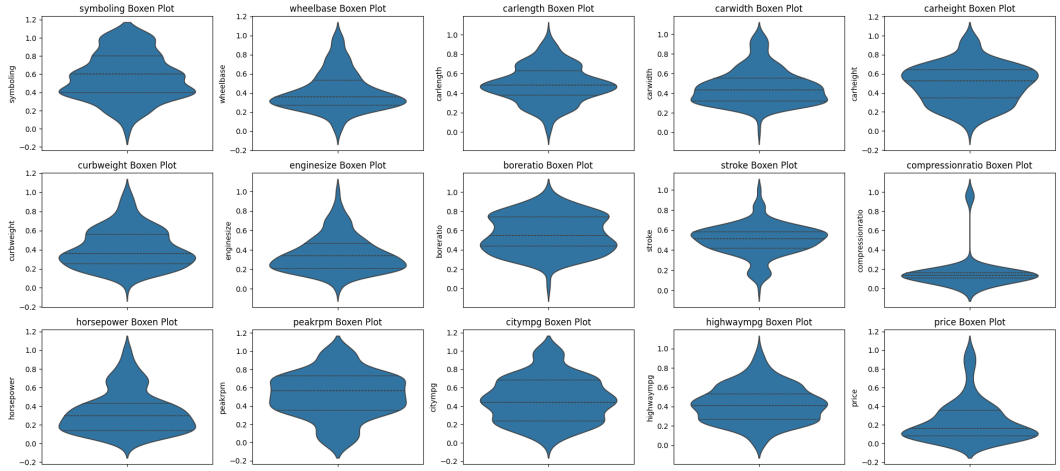
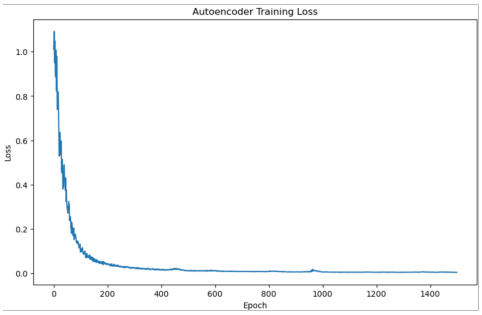
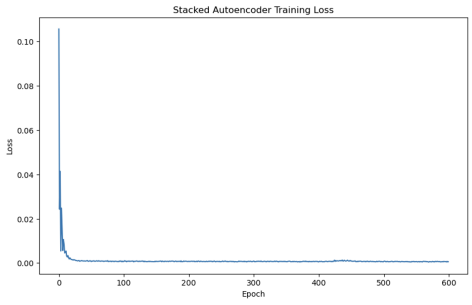


Figure 4: Data distribution on preprocessed Data

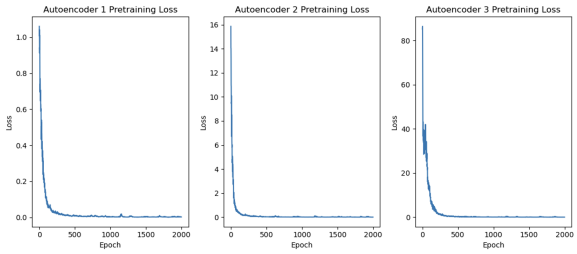
A.0.2 Training Details of AE and SAE



Loss of AE



Loss of SAE



Loss of AEs in pretraining stage

Loss of SAE without pretraining

Parameter	**Autoencoder (AE)**	**Stacked Autoencoder (SAE)**
Architecture Depth	4 Encoder + 3 Decoder layers	4 layers per AE $\times$ 3 stacked
Hidden Units (Encoder)	[256, 128, 64, 16]	[512, 256, 128, 64] per AE
Encoding Dimension	16	$256 \rightarrow 64 \rightarrow 16$
Activation Function	ReLU	ReLU
Loss Function	MSE	MSE
Optimizer	Adam	Adam
Learning Rate	1e-3	1e-3 (pretrain), 1e-4 (fine-tune)
Batch Size	128	128
Epochs (Total)	1500	2000 per AE + 600 fine-tuning
Dropout	None	None
Output Activation	None	None

Table 2: Comparison of Autoencoder (AE) and Stacked Autoencoder (SAE) Architectures and Training Settings

A.0.3 Clustering Algorithms

Algorithm 1 K-means Clustering

Require: Data points $X = \{x_1, x_2, \ldots, x_N\}$, number of clusters K
Ensure: Cluster assignments c_i and centroids μ_k

Initialize K cluster centroids $\{\mu_1, \mu_2, \ldots, \mu_K\}$ (e.g., randomly or with K-means++)
repeat
  for each point x_i **do**
    Assign to closest centroid:

$$c_i \leftarrow \arg\min_k \|x_i - \mu_k\|^2$$

  end for
  for each cluster $k = 1$ to K **do**
    Update centroid:

$$\mu_k \leftarrow \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$$

  end for
until convergence (e.g., assignments do not change)
Compute SSE:

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

Algorithm 2 Hierarchical Clustering with Ward's Linkage

1: Initialize each data point as a single cluster, resulting in n clusters where n is the number of data points.
2: Compute the proximity matrix (initially containing pairwise squared Euclidean distances between all data points).
3: **while** number of clusters > 1 **do**
4: Find the pair of clusters C_i and C_j that minimizes the Ward's linkage criterion:

$$\Delta(C_i, C_j) = \frac{|C_i| \cdot |C_j|}{|C_i| + |C_j|} \cdot \|\text{centroid}(C_i) - \text{centroid}(C_j)\|^2$$

5: where $|C_i|$ and $|C_j|$ are the sizes (number of points) of clusters C_i and C_j, and centroid is the mean vector of the cluster.
6: Merge C_i and C_j into a new cluster C_{new}.
7: Update the proximity matrix by computing distances between C_{new} and all other clusters using Ward's criterion.
8: Remove C_i and C_j from the list of clusters and add C_{new}.
9: **end while**
10: Return the hierarchy of clusters (e.g., as a dendrogram) or extract clusters at a desired level.

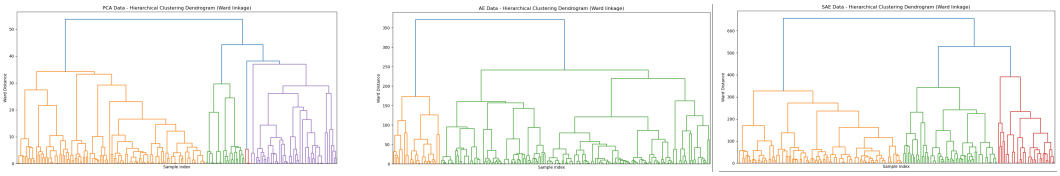


Figure 7: Dendrogram of the Bottom-Up Hierarchical Clustering Process on the Features Generated by *PCA* , *AE* and *SAE*.

A.0.4 Detailed Metrics and Results

Silhouette Score

Silhouette Score measures the cohesion within clusters and separation between clusters, defined as:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

where $a(i)$ is the average intra-cluster distance for point i, and $b(i)$ is the smallest average distance to points in any other cluster. The overall score is the mean over all points:

$$\text{Silhouette Score} = \frac{1}{N} \sum_{i=1}^{N} s(i)$$

Higher values (closer to 1) indicate better-defined clusters. The normalized Silhouette Score (`norm_sil`) is used directly as higher is better.

Davies-Bouldin Score

Davies-Bouldin Score quantifies the average similarity ratio between each cluster and its most similar cluster:

$$\text{DB} = \frac{1}{K} \sum_{i=1}^{K} \max_{j \neq i} \left(\frac{s_i + s_j}{d(c_i, c_j)} \right)$$

where s_i is the average distance from points in cluster i to its centroid, and $d(c_i, c_j)$ is the distance between centroids of clusters i and j. Lower values indicate better clustering. The normalized Davies-Bouldin Score (`norm_db`) is inverted (i.e., $1 -$ normalized DB) to align with the preference for higher values.

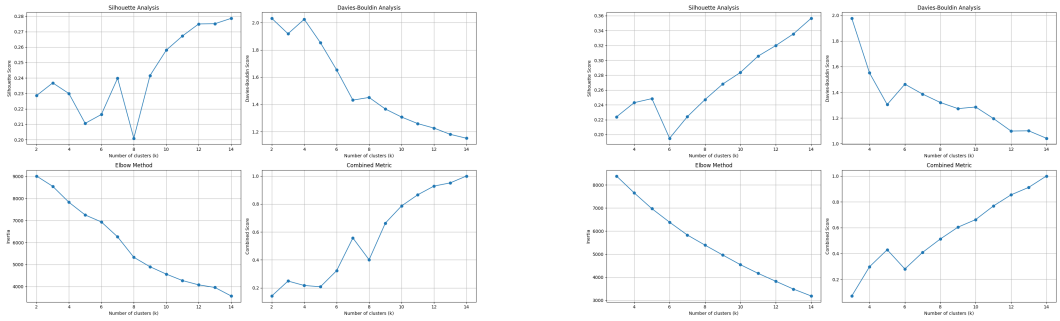
Inertia

Inertia Measures the compactness of clusters as the sum of squared distances from each point to its cluster centroid:

$$\text{Inertia} = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - c_i\|^2$$

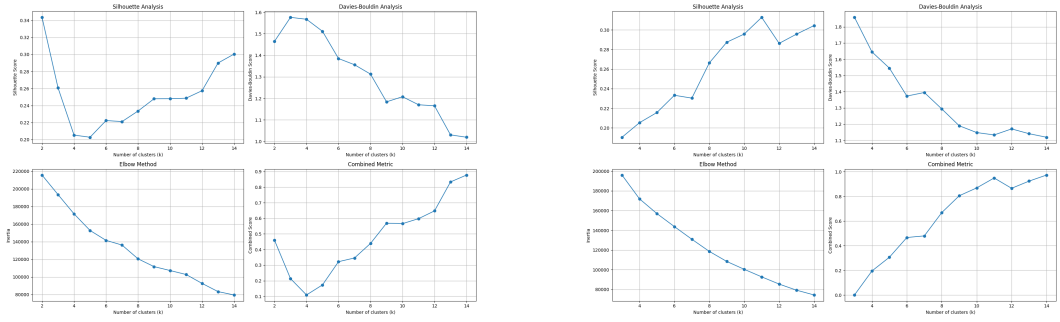
where C_i is the set of points in cluster i, and c_i is the centroid. Lower values indicate more compact clusters. The normalized Inertia (`norm_inertia`) is inverted similarly to the Davies-Bouldin Score.

Numerical Results of the Metrics



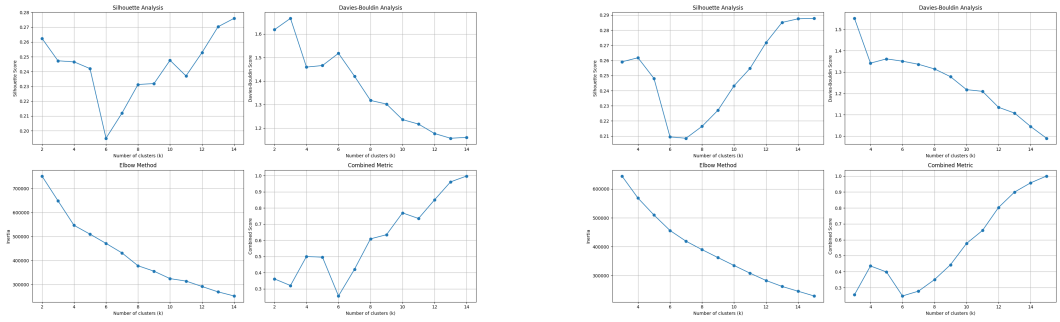
K-means hierarchical clustering

Figure 8: Metrics and combined score for PCA



K-means hierarchical clustering

Figure 9: Metrics and combined score for AE



K-means hierarchical clustering

Figure 10: Metrics and combined score for SAE

A.0.5 Numerical Results

	Cluster price medians and VW price medians for PCA													
Cluster	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Cluster Median	6599.0	16845.0	11374.0	34028.0	8189.0	28212.0	12745.0	7894.0	16500.0	17450.0	35056.0	23025.0	16695.0	11294.0
VW Median	—	—	10787.5	—	8495.0	—	—	—	—	13295.0	—	—	—	—

Table 3: Cluster price medians and VW price medians for PCA-reduced data (KMeans)

	Cluster price medians and VW price medians for AE													
Cluster	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Cluster Median	8499.0	28212.0	16566.5	16500.0	7295.0	18620.0	7673.5	8395.5	17710.0	16885.0	33278.0	11048.0	12745.0	9767.0
VW Median	—	—	—	—	—	—	—	—	—	—	—	—	—	9737.5

Table 4: Cluster price medians and VW price medians for AE-reduced data (KMeans)

	Cluster price medians and VW price medians for SAE												
Cluster	0	1	2	3	4	5	6	7	8	9	10	11	12
Cluster Median	9588.5	16845.0	7899.0	23950.25	7295.0	36000.0	22470.0	8916.5	16797.5	17859.167	19699.0	9995.0	12745.0
VW Median	—	—	—	—	9980.0	—	13295.0	11595.0	—	—	—	8495.0	—

Table 5: Cluster price medians and VW price medians for SAE-reduced data (KMeans)

	Cluster price medians and VW price medians for PCA														
Cluster	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Cluster Median	7447.0	16630.0	13622.5	7295.0	35056.0	14399.0	7894.0	12745.0	8745.0	34028.0	9619.0	35550.0	18950.0	28212.0	—
VW Median	—	—	9987.5	—	—	—	—	—	8745.0	—	—	—	—	—	—

Table 6: Cluster price medians and VW price medians for PCA-reduced data (HC)

	Cluster price medians and VW price medians for AE													
Cluster	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Cluster Median	34184.0	33278.0	11845.0	16900.0	9395.0	17580.0	7955.5	7878.0	15690.0	7295.0	9959.0	22835.0	5151.0	11374.0
VW Median	—	—	—	—	8745.0	—	—	9980.0	—	—	—	—	—	11595.0

Table 7: Cluster price medians and VW price medians for AE-reduced data (HC)

	Cluster price medians and VW price medians for SAE														
Cluster	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Cluster Median	11446.5	16797.5	11722.5	16925.0	32250.0	17022.0	7447.0	7129.0	20695.0	9959.0	7955.5	9991.5	31964.25	9639.0	8745.0
VW Median	—	—	11595.0	—	—	—	—	9980.0	13295.0	—	—	8495.0	—	—	8745.0

Table 8: Cluster price medians and VW price medians for SAE-reduced data (HC)