K-Means Clustering Analysis on CarDekho Vehicle Dataset

Avinash Bunga

Master of Science in Information Systems and Business Analytics

Park University

CIS625HOS2P2025 Machine Learning for Business

Professor: Abdelmonaem Jornaz

April 18, 2025

K-Means Clustering Analysis on CarDekho Vehicle Dataset

This study applies the K-Means clustering algorithm to the CarDekho vehicle dataset to uncover hidden patterns and group vehicles with similar characteristics. The dataset includes various numeric and categorical variables such as price, fuel type, transmission, mileage, and car age. The primary objective is to segment vehicles into meaningful groups using machine learning (see Appendix A; Birla, n.d.).

Methodology

1. Running K-Means Clustering on the Data

The K-Means clustering algorithm from sklearn.cluster was used to segment the dataset into logical groups based on feature similarity (see Appendix F; Arvai, 2024).

2. Preparing the Data for Clustering

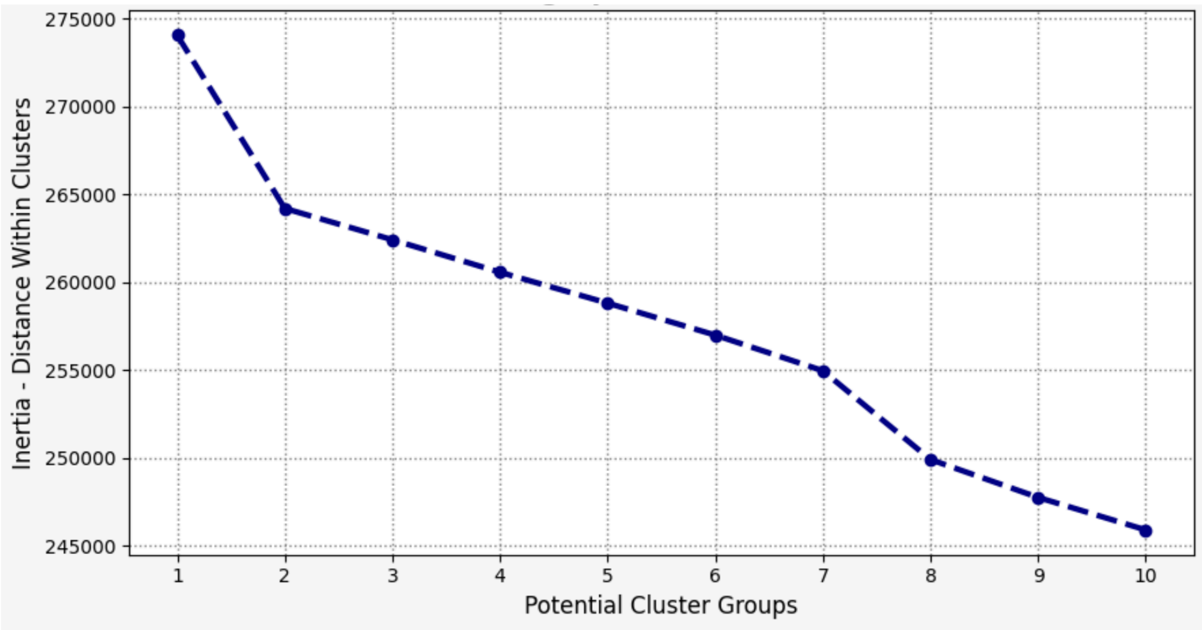
To prepare the data for clustering, text-heavy columns such as 'Model', 'Engine', 'Max Power', and 'Max Torque' were removed. Categorical variables were converted into numeric form using one-hot encoding, and all features were standardized using StandardScaler to ensure uniformity across variables (see Appendix B through D; Mulani, 2022).

3. Determining Number of Clusters

The Elbow Method was applied to choose the optimal value of k. As shown below, the inertia dropped sharply until k = 2, after which the gains diminished (see Appendix E; Tomar & Whitfield, 2025).

Figure 1

Elbow Curve to Determine Optimal Clusters



This graph shows that k = 2 is the optimal number of clusters, balancing accuracy and simplicity.

4. Final K-Means Clustering (k = 2)

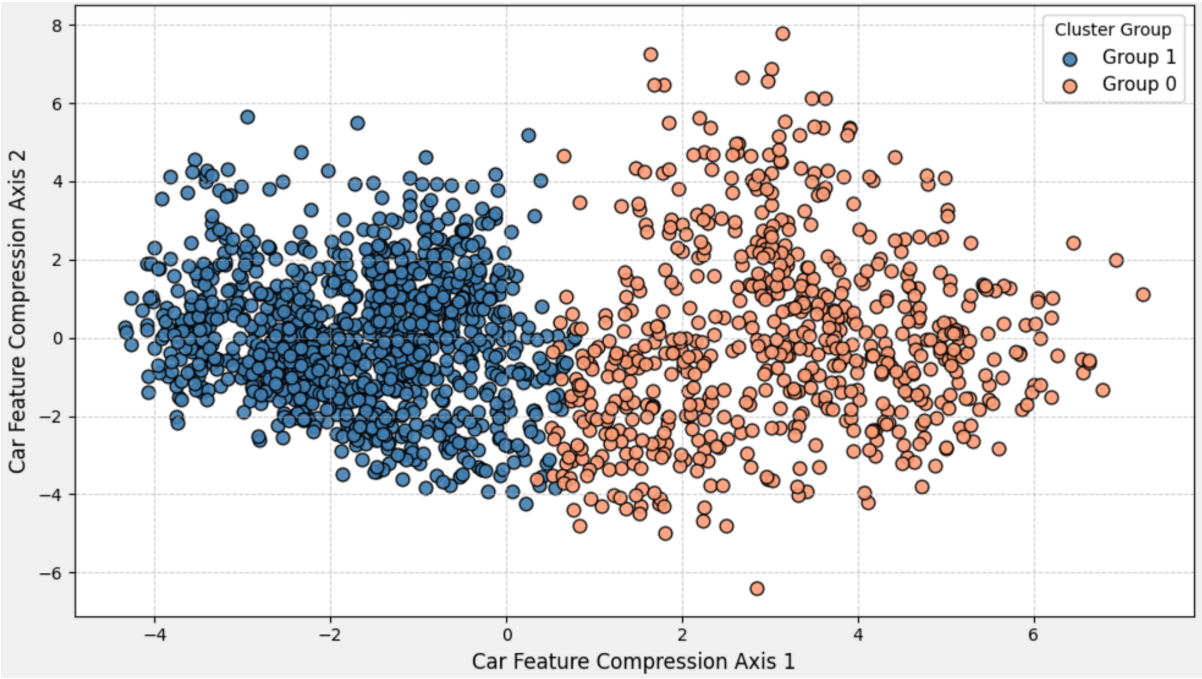
The K-Means algorithm was run with n_clusters=2, assigning each vehicle a cluster label (0 or 1) (see Appendix F).

5. Visualizing the Clusters

Principal Component Analysis (PCA) was used to reduce the dataset into two dimensions, and a scatter plot was created to visualize the clustering outcome (see Appendix G; Plotly, n.d.).

Figure 2

Visual Representation of Car Clusters



6. Statistical Summary by Cluster

Cluster-wise averages of key numeric variables were calculated for comparison (see Appendix H).

Table 1

Average Feature Values by Cluster

Cluster	Price	Kilometer	Car Age	Fuel Tank Capacity	Seating Capacity
0	2,207,536	58,048	8.38	64.38	5.77
1	627,503	51,223	8.75	42.43	5.06

Table 2

Final Outcome Summary Table

Aspect	Cluster 0	Cluster 1
Type	Premium, larger vehicles (SUVs/MPVs)	Compact or entry-level vehicles
Price	Higher-priced (~2.2M INR avg)	Lower-priced (~627K INR avg)
Kilometers Driven	More (avg ~58K km)	Slightly less (avg ~51K km)
Fuel Tank Capacity	Larger (avg ~64.38L)	Smaller (avg ~42.43L)
Seating Capacity	More seats (avg ~5.77)	Fewer seats (avg ~5.06)

Conclusion

This clustering project successfully segmented the CarDekho vehicle dataset into two distinct groups using the K-Means clustering algorithm. The number of clusters ($k = 2$) was selected based on the Elbow Method, with clear justification shown in Figure 1. The clustering model was applied on a cleaned and standardized dataset, and PCA was used for effective 2D visualization. As shown in Figure 2, the two clusters are clearly separated and align well with real-world interpretations of vehicle categories. Cluster 0 contains higher-priced, larger, utility-focused vehicles, while Cluster 1 includes lower-priced, compact vehicles. The statistical summaries further support these interpretations. Overall, this analysis demonstrates the value of unsupervised machine learning in revealing hidden structures in commercial automotive data.

References

Arvai, K. (2024). *K-means clustering in Python: A practical guide*. Real Python. Retrieved April 18, 2025, from https://realpython.com/k-means-clustering-python/

Birla, N. (n.d.). *Vehicle dataset: Used cars data from websites*. Kaggle. Retrieved April 5, 2025, from https://www.kaggle.com/datasets/nehalbirla/vehicle-dataset-from-cardekho

Mulani, S. (2022, August 3). *Using StandardScaler() function to standardize Python data*. DigitalOcean. Retrieved April 18, 2025, from https://www.digitalocean.com/community/tutorials/standardscaler-function-in-python

Plotly. (n.d.). *PCA visualization in Python*. Retrieved April 18, 2025, from https://plotly.com/python/pca-visualization/

Tomar, A., & Whitfield, B. (2025, March 13). *Elbow method: Definition, drawbacks, vs. silhouette score*. Built In. Retrieved April 18, 2025, from https://builtin.com/data-science/elbow-method

Appendix A

```
#Load and preview the dataset
import pandas as pd
data = pd.read_csv("cleaned_v4_filtered.csv")
data.head()
```

Appendix B

```
#Drop text-heavy columns like Model and Engine
columns_to_remove = ['Model', 'Engine', 'Max Power', 'Max Torque']
data = data.drop(columns=columns_to_remove)
data.head()
```

Appendix C

```
#One-hot encode categorical features
data_encoded = pd.get_dummies(data)
data_encoded.head()
```

Appendix D

```
#Standardize features using StandardScaler
from sklearn.preprocessing import StandardScaler
data_encoded = data_encoded.dropna()
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data_encoded)
scaled_df = pd.DataFrame(scaled_data, columns=data_encoded.columns)
scaled_df.head()
```

Appendix E

```
#Elbow method to choose optimal k
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
inertia = []
k_values = range(1, 11)
for k in k_values:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(scaled_df)
    inertia.append(kmeans.inertia_)
#Styled Elbow Plot
plt.figure(figsize=(9, 5), facecolor='#f7f7f7')
plt.plot(k_values, inertia, marker='o', linestyle='--', linewidth=3, color='darkblue')
plt.title("Choosing Optimal Cluster Count", fontsize=14, fontweight='bold')
plt.xlabel("Potential Cluster Groups", fontsize=12)
plt.ylabel("Inertia - Distance Within Clusters", fontsize=12)
plt.grid(color='gray', linestyle=':', linewidth=1)
plt.xticks(k_values)
plt.tight_layout()
plt.show()
```

Appendix F

```
#Final KMeans model (k=2) and cluster assignment
kmeans = KMeans(n_clusters=2, random_state=42)
```

```
labels = kmeans.fit_predict(scaled_df)
data_encoded['Cluster'] = labels
data_encoded[['Price', 'Kilometer', 'Car_Age', 'Cluster']].head()
```

Appendix G

```
#PCA dimensionality reduction and cluster visualization
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
pca_result = pca.fit_transform(scaled_df)
pca_df = pd.DataFrame(pca_result, columns=['Component_1', 'Component_2'])
pca_df['Cluster'] = labels
#Color palette
colors = ['#ffa07a', '#4682b4']
#Styled PCA Plot
plt.figure(figsize=(10, 6), facecolor='#f0f0f0')
for cluster_id in pca_df['Cluster'].unique():
    cluster_slice = pca_df[pca_df['Cluster'] == cluster_id]
    plt.scatter(cluster_slice['Component_1'], cluster_slice['Component_2'],
                label=f'Group {cluster_id}', s=60, alpha=0.9,
                edgecolor='black', color=colors[cluster_id])
plt.title("Visual Representation of Car Clusters", fontsize=15, fontweight='bold')
plt.xlabel("Car Feature Compression Axis 1", fontsize=12)
plt.ylabel("Car Feature Compression Axis 2", fontsize=12)
plt.legend(title="Cluster Group", fontsize=11)
plt.grid(True, linestyle='--', linewidth=0.7, alpha=0.6)
plt.tight_layout()
plt.show()
```

Appendix H

```
#Summary stats calculation and grouping by cluster
summary_columns = ['Price', 'Kilometer', 'Car_Age', 'Fuel Tank Capacity', 'Seating Capacity']
cluster_summary = data_encoded.groupby('Cluster')[summary_columns].mean().round(2)
cluster_summary.head()
```