Feynn Labs: Project 2

Electric Vehicle Market Analysis Using K-Means Clustering

Name: Mathias Prajwal Dsouza

GitHub Link:

Date: 13/10/2024

Problem Statement

To analyse the Electric Vehicle market in India using Segmentation analysis and come up with a feasible strategy to enter the market, targeting the segments most likely to use Electric vehicles.

Dataset Collections

1. Materials

- **Dataset**: The dataset, named `Extended_EV_Cars_Data.csv`, was uploaded for analysis. It contains key features related to Electric Vehicles (EVs) in the Indian market, such as price, range, efficiency, and the presence of charging infrastructure. The columns include:
    - `PriceINR`: Price of the vehicle in Indian Rupees.
    - `Range_Km`: Maximum range (in kilometers) that the EV can travel on a full charge.
    - `Efficiency_WhKm`: Efficiency of the vehicle in Watt-hours per kilometer.
    - `Charging_Stations`: Number of charging stations in the vicinity.
    - `EV_Market_Share_Percent`: Market share of EVs in the region.
    - `Average_Daily_Travel_Km`: Average daily distance traveled by the vehicle.
    - `Avg_Frequency_of_Charging_Per_Week`: How often the vehicle is charged per week.
    - Other features like vehicle body style and cluster labels (for post-clustering analysis).
- **Data Source**: The dataset could be collected from various industry reports, manufacturer-provided data, or government databases related to EV sales and infrastructure.

1. Kaggle

Kaggle is a popular platform for sharing and discovering datasets. You can explore similar EV-related datasets, participate in discussions, and access resources for data science.

Link: https://www.kaggle.com/datasets/geoffnel/evs-one-electric-vehicle-dataset

2. EV-Volumes

EV-Volumes offers data and analysis regarding the electric vehicle market worldwide. It provides market share, sales, and production data for EVs.

Link: https://ev-volumes.com/

3. Government Open Data Portals

Many governments share open datasets related to vehicle registration, transportation, and energy consumption, which could be useful for market analysis of electric vehicles.

Link: https://www.data.gov.in/keywords/Electric

3. Explanation of the Process

- **Feature Selection**:
  - A set of features, including `PriceINR`, `Range_Km`, `Efficiency_WhKm`, and others, were selected to cluster the EV data.
  - These features were chosen because they represent essential metrics for evaluating the performance and market potential of EVs.
- **Data Preprocessing**:
  - The data was standardized using `StandardScaler()` to ensure that all features are on a similar scale. This is essential for algorithms like K-Means, which are sensitive to the magnitude of feature values.
- **Elbow Method**:
  - To determine the optimal number of clusters, the Elbow Method was applied. This involved fitting K-Means models with varying numbers of clusters (from 1 to 10) and plotting the SSE to find the "elbow" point, which indicated the best number of clusters.
- **K-Means Clustering**:
  - Once the optimal number of clusters was determined, the K-Means algorithm was applied to segment the vehicles into distinct clusters.
  - Each cluster represents a **unique group of EVs** based on shared characteristics like price, range, and efficiency.
- **Visualization**:
  - Several visualizations were used to understand the data better and interpret the clustering results:
    - **Scatter Plot**: Displayed price vs. range, highlighting how different clusters compare in terms of cost and range.
    - **Box Plot**: Compared price distribution across different body styles and vehicle types.
    - **Correlation Heatmap**: Showed the relationships between key features in the dataset.
    - **3D Plot**: Helped visualize clustering based on price, range, and efficiency.

Step 1: Importing Required Libraries

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
import seaborn as sns
```

- pandas is used for loading and manipulating the dataset.
- **matplotlib.pyplot** and **seaborn** are used for creating visualizations.
- **StandardScaler** is imported from `sklearn` to normalize the dataset (important for algorithms like K-Means that are sensitive to data scales).
- **KMeans** is imported from `sklearn.cluster` to perform K-Means clustering.

- **Axes3D** is used for creating 3D visualizations in Matplotlib.

Step 2: Loading the Dataset

```
ev_data = pd.read_csv('/content/Extended_EV_Cars_Data.csv')
display(ev_data)
```

- The dataset is loaded using `pd.read_csv()`, which reads the CSV file and stores it in the DataFrame `ev_data`.
- The `display()` function is used to show the content of the dataset.

Step 3: Feature Selection

```
features = ['PriceINR', 'Range_Km', 'Efficiency_WhKm', 'Charging_Stations',
            'EV_Market_Share_Percent', 'Average_Daily_Travel_Km', 'Avg_Frequency_of_Charging_Per_Week']
```

- A list of features (columns) from the dataset is selected for clustering.

- These features represent essential metrics for understanding the EV market: price, range, efficiency, market share, charging infrastructure, etc.

Step 4: Data Preprocessing (Standardization)

```
scaler = StandardScaler()
scaled_features = scaler.fit_transform(ev_data[features])
```

- **StandardScaler** is initialized to standardize the features.
- `fit_transform()` is applied to the selected features, transforming them to have a mean of 0 and a standard deviation of 1, ensuring all features are on the same scale. This is crucial because K-Means is sensitive to the scale of data.

Step 5: Determining Optimal Number of Clusters (Elbow Method)

```
sse = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(scaled_features)
    sse.append(kmeans.inertia_)
```

- An empty list `sse` is initialized to store the **Sum of Squared Errors (SSE)** for different values of clusters `k`.
- A loop runs from 1 to 10 clusters, where the K-Means model is fitted on the scaled features for each `k`.
- `kmeans.inertia_` stores the SSE (how close data points in a cluster are to the centroid), which is appended to the `sse` list.
- The Elbow Method is used to plot SSE against the number of clusters to find the "elbow," where the SSE stops decreasing significantly. This indicates the optimal number of clusters.

Step 6: Plotting the Elbow Method.

```
plt.figure(figsize=(6, 4))
plt.plot(range(1, 11), sse, marker='o')
plt.title('Elbow Method for Optimal Clusters')
plt.xlabel('Number of clusters')
plt.ylabel('SSE (Sum of squared errors)')
plt.grid(True)
plt.show()
```

Step 7: Applying K-Means Clustering

```
optimal_clusters = 4
kmeans = KMeans(n_clusters=optimal_clusters, random_state=42)
ev_data['Cluster'] = kmeans.fit_predict(scaled_features)
```

- The optimal number of clusters (in this case, 4) is chosen based on the Elbow Method.
- A K-Means model is created with 4 clusters and fit on the scaled features.
- `fit_predict()` assigns each vehicle in the dataset to a cluster, and the result is added as a new column `Cluster` in the `ev_data` DataFrame.

Step 8: Visualization (Scatter Plot)

```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='PriceINR', y='Range_Km', hue='Cluster', data=ev_data, palette='viridis')
plt.title('Scatter Plot of Price vs Range by Cluster')
plt.xlabel('Price (INR)')
plt.ylabel('Range (Km)')
plt.grid(True)
plt.show()
```

Step 9: Visualization (Bar Plot - EV Count by Body Style)

```
plt.figure(figsize=(10, 6))
sns.boxplot(x='BodyStyle', y='PriceINR', data=ev_data, palette='Set3')
plt.title('Price Distribution by Body Style')
plt.xlabel('Body Style')
plt.ylabel('Price (INR)')
plt.grid(True)
plt.show()
```

Step 10: Visualization (Box Plot - Price Distribution by Body Style)

```
plt.figure(figsize=(10, 6))
sns.boxplot(x='BodyStyle', y='PriceINR', data=ev_data, palette='Set3')
plt.title('Price Distribution by Body Style')
plt.xlabel('Body Style')
plt.ylabel('Price (INR)')
plt.grid(True)
plt.show()
```

Step 11: Visualization (Heatmap - Correlation Matrix)

```
plt.figure(figsize=(10, 6))
correlation_matrix = ev_data[features].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Heatmap between Features')
plt.show()
```

Step 12: 3D Scatter Plot (Price, Range, Efficiency)

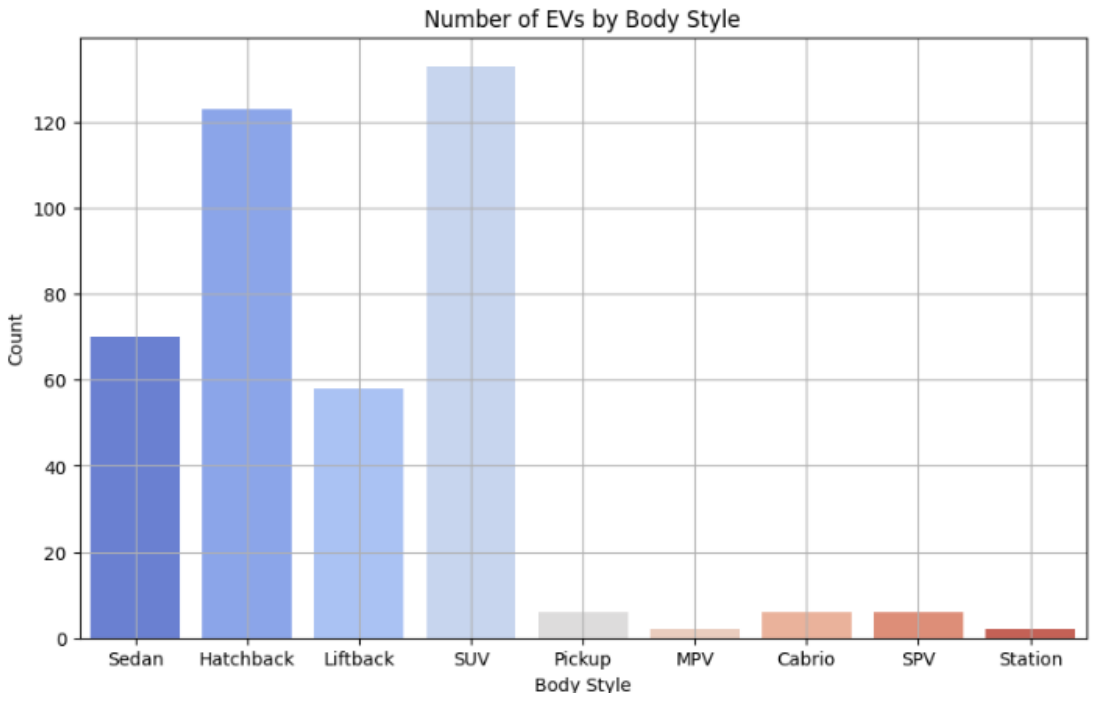
```
# Create a 3D plot for clustering based on Price, Range, and Efficiency
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')
# Plotting the clusters with Price, Range, and Efficiency
scatter = ax.scatter(ev_data['PriceINR'], ev_data['Range_Km'], ev_data['Efficiency_WhKm'],
                     c=ev_data['Cluster'], cmap='viridis', s=50)
# Adding labels and title
ax.set_xlabel('Price (INR)')
ax.set_ylabel('Range (Km)')
ax.set_zlabel('Efficiency (Wh/Km)')
ax.set_title('3D Plot of Clusters by Price, Range, and Efficiency')
# Add a color bar to indicate clusters
plt.colorbar(scatter)
plt.show()
```

Step 13: Identifying the Best Cluster

```
# Analyze the clusters by calculating the mean of key metrics for each cluster
cluster_means = ev_data.groupby('Cluster')[['PriceINR', 'Range_Km', 'Efficiency_WhKm']].mean()
print("Cluster-wise Averages:\n", cluster_means)

# Set relaxed criteria for finding the best cluster
# Example: Looking for clusters with a range above the average and a reasonable efficiency
avg_price = cluster_means['PriceINR'].mean()
avg_range = cluster_means['Range_Km'].mean()
avg_efficiency = cluster_means['Efficiency_WhKm'].mean()

# Modify the criteria to check if any cluster meets one or more conditions
best_cluster = cluster_means[(cluster_means['Range_Km'] > avg_range) &
                             (cluster_means['Efficiency_WhKm'] < avg_efficiency)]

print("\nBest Cluster Characteristics:\n", best_cluster)

# Check if best_cluster is empty before filtering vehicles
if not best_cluster.empty:
    # Filter the dataset to show vehicles in the best cluster
    best_vehicle_index = best_cluster.index[0]  # Get the first index of the best cluster
    best_vehicles = ev_data[ev_data['Cluster'] == best_vehicle_index]
    print("\nRecommended Vehicles for Purchase:\n", best_vehicles[['Brand', 'Model', 'PriceINR', 'Range_Km', 'Efficiency_WhKm']])

    # Conclusion based on the analysis
    print("\nBased on the clustering analysis, the vehicles in the following cluster provide the best balance between price, range, and efficiency.")
    print("These vehicles are recommended for purchase.")
else:
    print("\nNo optimal cluster found based on the relaxed criteria. Consider adjusting the criteria further.")
```

4. Expiations of Graphical and Visual Prestation

1. Bar Plot

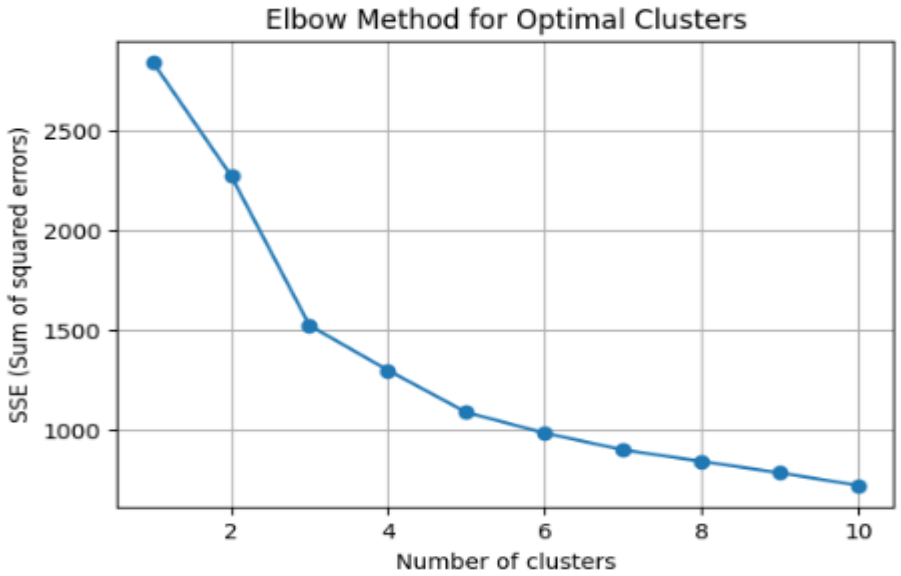


(Bar Plot: Number of EVs by Body Style) visualizes the distribution of electric vehicles (EVs) across different body styles. In the bar chart, SUVs are the most common, followed by Hatchbacks, Sedans, and Lift backs. The other body styles such as Pickup, MPV, Cabrio, SPV, and Station show significantly fewer options.

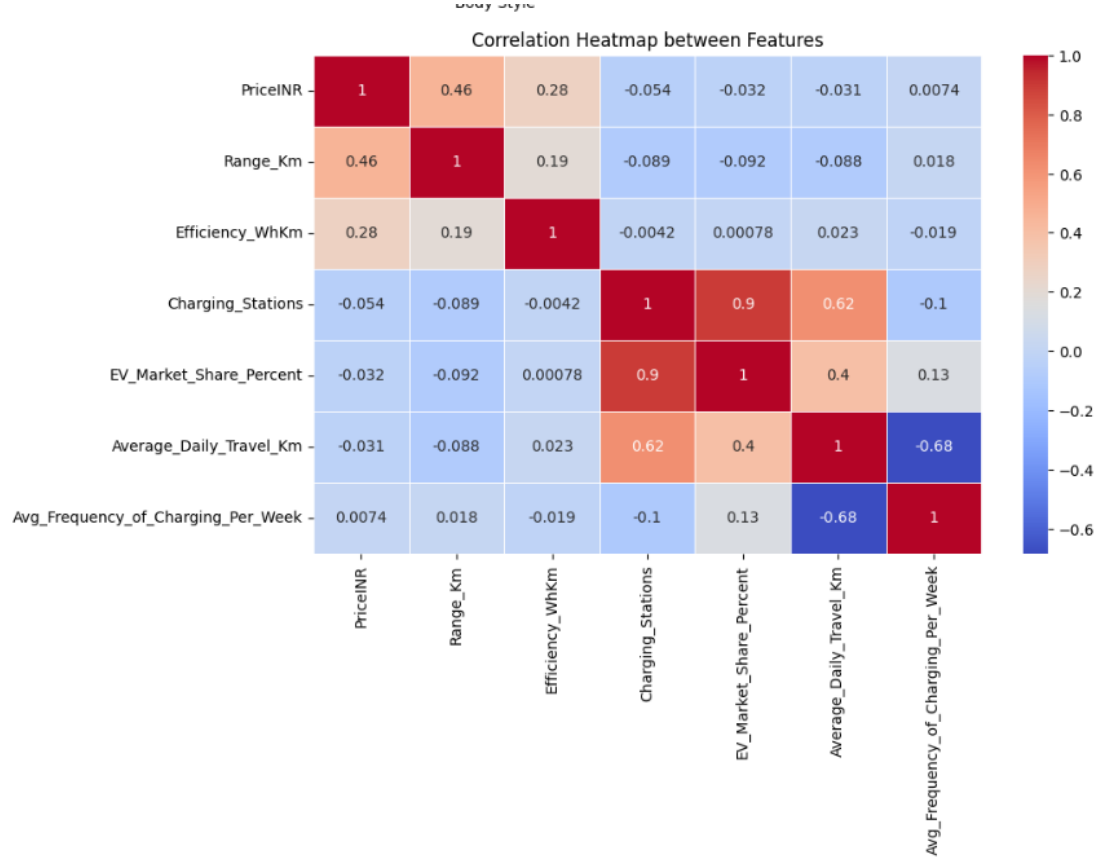
2. Scatter Plot of Price vs Range by Cluster

- **Description:** This scatter plot displays the relationship between the price (in INR) and the range (in km) of EVs, color-coded by the cluster each vehicle belongs to.

3. Elbow Method for Optimal Clusters



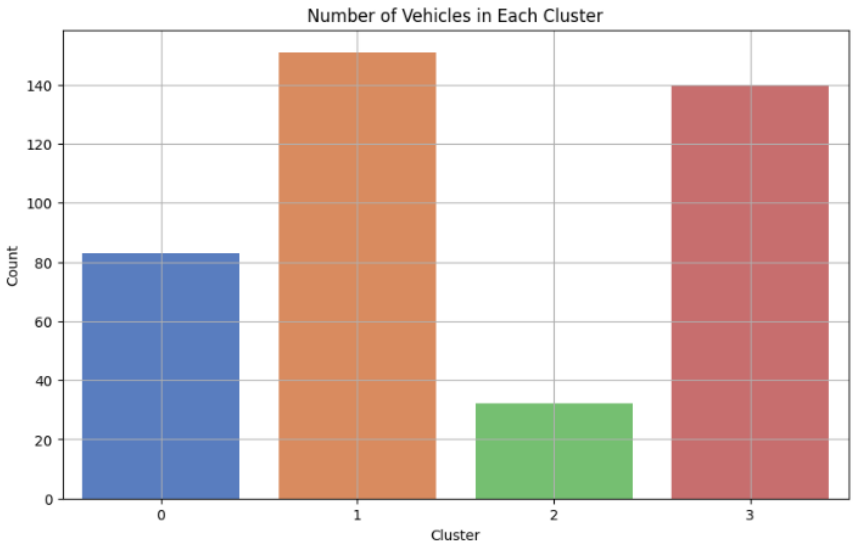
4. Heatmap Correlation



<ipython-input-8-9ce61e1d97d6>:80: FutureWarning:

In the correlation heatmap, each cell represents the strength of relationships between features. The highest correlation (0.9) is between "Charging Stations" and "EV Market Share Percent," suggesting that more charging stations are associated with higher market shares. Conversely, "Average Daily Travel" and "Avg. Frequency of Charging Per Week" show a strong negative correlation (-0.68), meaning vehicles that travel more frequently charge less often.

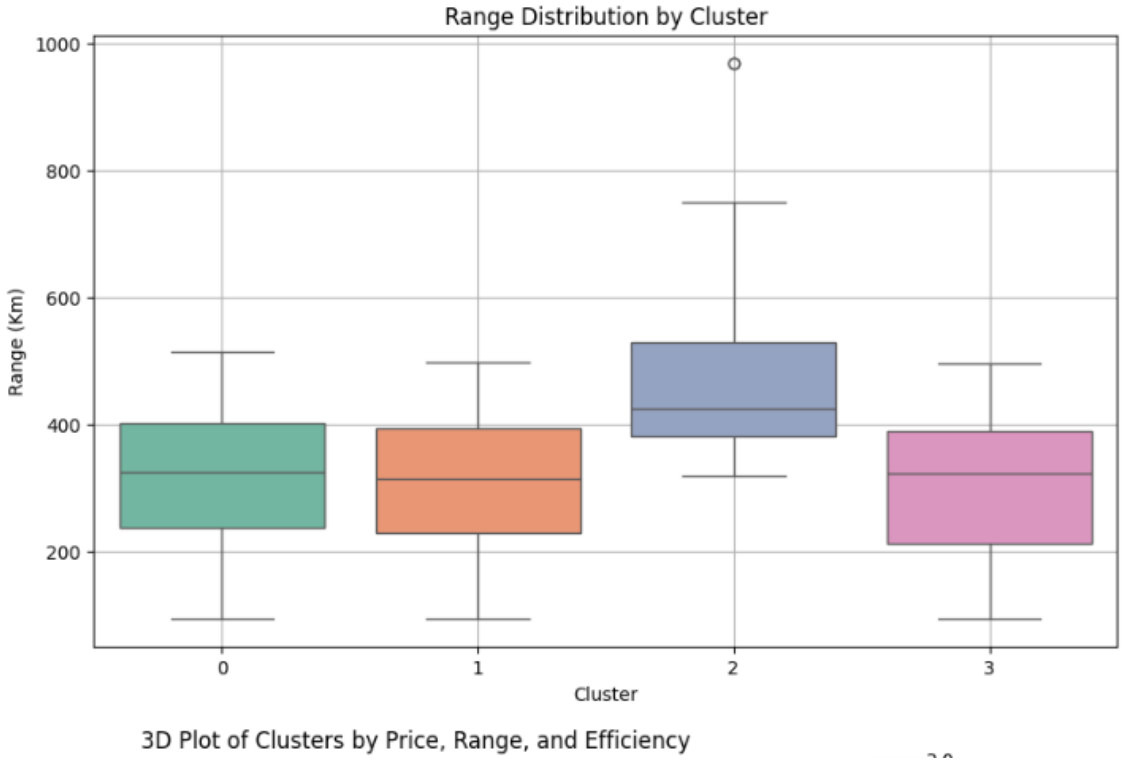
5. Bar Chart



<ipython-input-8-9ce61e1d97d6>:89: FutureWarning:

The bar chart for the number of vehicles in each cluster shows the distribution of data points across four clusters. Cluster 1 has the highest count of vehicles, indicating a dominant group based on the applied clustering algorithm, while Cluster 2 has the fewest.

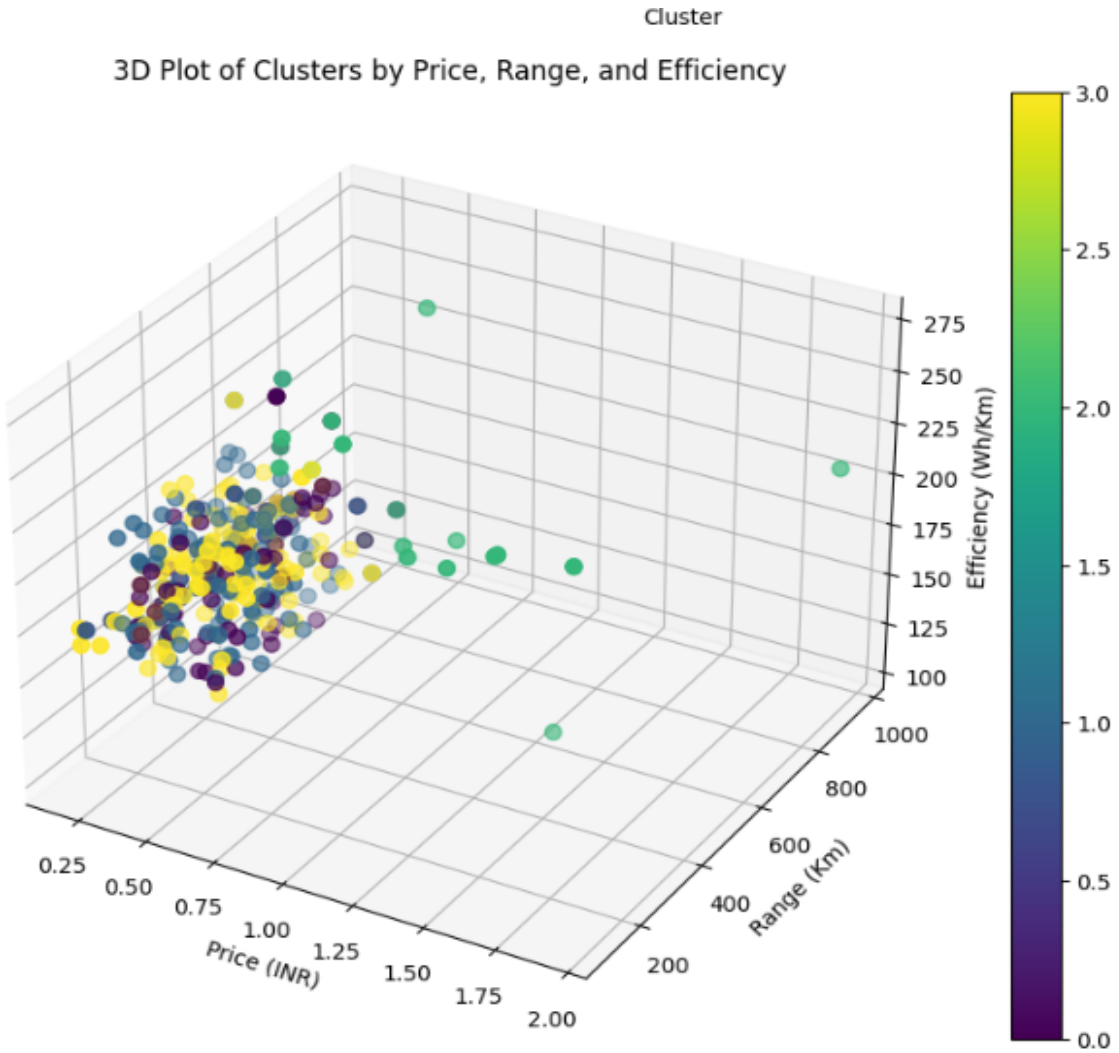
6. Boxplot



In the boxplot for range distribution by cluster, the vehicles in Cluster 2 stand out with the highest range, including an outlier reaching close to 1000 km. Clusters 0, 1, and 3 have lower, more consistent ranges, indicating that vehicles in Cluster 2 are designed for longer distances.

7. 3D Scatter Plot

The 3D scatter plot visualizes clusters by price, range, and efficiency. Each cluster occupies a distinct region in this 3D space. Cluster 2 has higher vehicle ranges and prices, while the remaining clusters show more moderate levels, with some overlap in efficiency and range but clearer distinctions in price.



Cluster-wise Averages:

Cluster	PriceINR	Range_Km	Efficiency_WhKm
0	4.764259e+06	318.204819	186.710843
1	4.384404e+06	314.682119	184.509934
2	1.021521e+07	485.000000	214.437500
3	4.468136e+06	307.578571	186.264286

5. Problem Can Be Faced with Relevant Solution

1. Companies need to ensure that their product offerings align with the preferences and affordability of each segment. Misalignment in pricing or features could reduce market acceptance.

Sol: Based on the K-Means clustering, companies can identify specific customer segments with distinct characteristics (price, range, charging infrastructure). For example, Cluster 2 (vehicles with a high range and price) might target premium customers, while Clusters with lower price ranges can cater to the mass market

2. Expanding charging infrastructure requires significant investment and collaboration with local governments or third parties. Delays in infrastructure development can slow down market entry or growth.

Sol: The correlation analysis shows a strong relationship between the availability of charging stations and market share. Companies can increase EV adoption by partnering with infrastructure providers to expand charging stations, making EVs more convenient for users.

3. Manufacturing diverse body styles requires a flexible production process, which may be costly. Additionally, companies may face challenges adapting existing production lines for electric platforms.

Sol: The analysis indicates that SUVs dominate the EV market, followed by hatchbacks and sedans. Companies should prioritize developing EV models in these body styles, as they are most popular among consumers.

4. Improving vehicle range while keeping costs low can be technologically challenging. Battery technology is a major factor, and advancements may require time and significant R&D investment.

Sol: The 3D scatter plot shows distinct clusters based on vehicle range and efficiency. Companies can differentiate their products by developing EVs with either high efficiency (for cost-conscious customers) or extended range (for long-distance travellers).