EV Car Market Segmentation Report

1. Introduction

This report outlines the process of segmenting the car market using **machine learning techniques**. The primary goal of this analysis is to uncover distinct car market segments by clustering cars based on specific features like the car's year of manufacture, selling price, and kilometers driven.

Machine Learning Model Used

In this project, the **KMeans clustering algorithm** was the core machine learning model used. KMeans is an unsupervised machine learning algorithm that helped in dividing the car market into three distinct segments. By grouping cars based on similarities in their features (year, selling_price, and km_driven), the KMeans algorithm provided insight into the natural groupings within the market.

KMeans Clustering works by finding the optimal centroids (or centers) for a specified number of clusters. In this case, after using the **Elbow Method** to determine the ideal number of clusters, KMeans effectively grouped the cars into three meaningful market segments. The centroids of each cluster represent the average characteristics of the cars in that segment.

2. Dataset Overview

The dataset contains information about cars, including:

- **Year**: Reflects the manufacturing year of the car, which often correlates with depreciation and market value.
- **Selling Price**: Provides the current market price of the car.
- **Km Driven**: Indicates the car's total usage, expressed in kilometers driven.

These features were critical in segmenting the car market, and additional columns like fuel_type, seller_type, and transmission were available but not used in this initial analysis.

3. Data Preprocessing

To ensure accurate clustering, we performed data cleaning and selected relevant features (year, selling_price, km_driven). Afterward, we applied feature scaling to normalize the data. This step was crucial to avoid bias in the clustering results, as the KMeans algorithm is sensitive to the magnitude of features.

3.1 Library Imports: These are the necessary libraries for data manipulation, visualization, and machine learning tasks.

```
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
```

- **pandas**: For data loading and processing.
- **matplotlib & seaborn**: For plotting visualizations.
- **sklearn.preprocessing**: To scale the data for consistent clustering.
- **PCA & KMeans**: For dimensionality reduction and clustering.

3.2 Loading the Dataset: Load the dataset and preview the first few records to understand its structure.

```
# Load the dataset
data = pd.read_csv('CAR DETAILS FROM CAR DEKHO.csv')

# Preview the dataset
print(data.head())
```

The dataset includes columns like year, selling_price, km_driven, fuel_type, seller_type, etc., which describe the car market details.

3.3 Data Preprocessing: Clean the dataset by removing rows with missing values and selecting the necessary columns for analysis.

```
# Data Cleaning: Drop NaN values and select relevant columns
data = data.dropna()  # Drop rows with missing values
data = data[['year', 'selling_price', 'km_driven']]  # Select relevant columns
```

This step ensures only essential columns are used, and no missing data disrupts the clustering.

3.4 Feature Scaling: Use StandardScaler to normalize the data to ensure it's on the same scale, improving clustering performance.

```
# Feature Scaling using StandardScaler
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data)
```

Scaling ensures that large numerical differences don't dominate the clustering algorithm.

3.5 Principal Component Analysis (PCA): Apply PCA to reduce the dataset into two components, allowing visualization in two dimensions.

```
# Perform PCA for dimensionality reduction to 2 components
pca = PCA(n_components=2)
pca_data = pca.fit_transform(scaled_data)
```

PCA helps reduce the data to two principal components, retaining the essential features for clustering.

3.6 KMeans Clustering: Apply KMeans clustering to segment the car data into three groups based on patterns.

```
# Apply KMeans Clustering with 3 clusters
kmeans = KMeans(n_clusters=3)  # You can change the number of clusters based on analysis
kmeans.fit(pca_data)

# Add cluster labels to the original dataframe
data['Cluster'] = kmeans.labels_
```

This step divides the dataset into three distinct clusters, representing different car market segments.

4. Feature Scaling

Before applying clustering, the data was scaled using **StandardScaler**. This method ensures that all the variables, such as the year, selling_price, and km_driven, are on the same scale. This step is crucial because clustering algorithms like **KMeans** are distance-based, and large-scale differences between features could distort the results.

5. Principal Component Analysis (PCA)

To make the dataset easier to visualize and reduce dimensionality, **Principal Component Analysis (PCA)** was applied. PCA condenses the dataset into two principal components, which helps retain the maximum amount of information while reducing the complexity of the data.

Why PCA?

- **Dimensionality Reduction**: PCA reduces the number of features to two, simplifying the visualization of clusters.
- **Variance Retention**: Despite reducing the data dimensions, PCA ensures that the variance in the data is retained as much as possible.

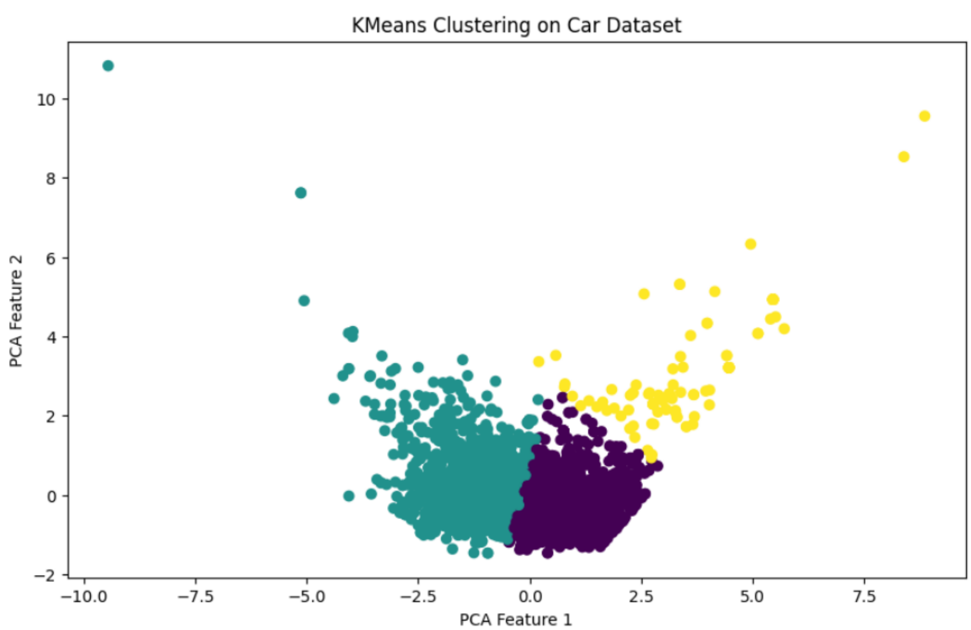
The two principal components are plotted on a 2D scatter plot, allowing us to visualize how different cars cluster together based on their features.

6. KMeans Clustering

After preprocessing and applying PCA, **KMeans clustering** was applied to group the dataset into three clusters. The **KMeans** algorithm groups data points based on their proximity to the nearest centroid. In our case, we aim to divide the car market into three distinct clusters, each representing different segments of the car market.

Why KMeans?

- **Unsupervised Learning**: KMeans is a type of unsupervised learning algorithm that does not require labeled data, making it ideal for this type of analysis.
- **Centroid-Based**: KMeans identifies centroids and assigns data points to the nearest centroid, forming clusters based on feature similarity.



KMeans Cluster Scatter Plot

The results of KMeans are visualized in a scatter plot, where:

- Each point represents a car.
- The x-axis and y-axis correspond to the first and second principal components from PCA.
- Each color represents a different cluster.

A scatter plot that visualizes the car market segmented into three clusters. Each cluster corresponds to cars that are grouped based on their manufacturing year, selling price, and kilometers driven.

Conclusion and Insights Gained

The clustering analysis provided valuable insights into the car market:

- **Cluster 1**: Consisted of older cars with high mileage and lower selling prices. This segment represents cars with significant depreciation and extensive usage.
- **Cluster 2**: Represented moderately used cars with mid-range prices and mileage. These cars offer a balance between affordability and usability.
- **Cluster 3**: Comprised newer cars with fewer kilometers driven and higher selling prices. These cars are typically newer models in excellent condition.

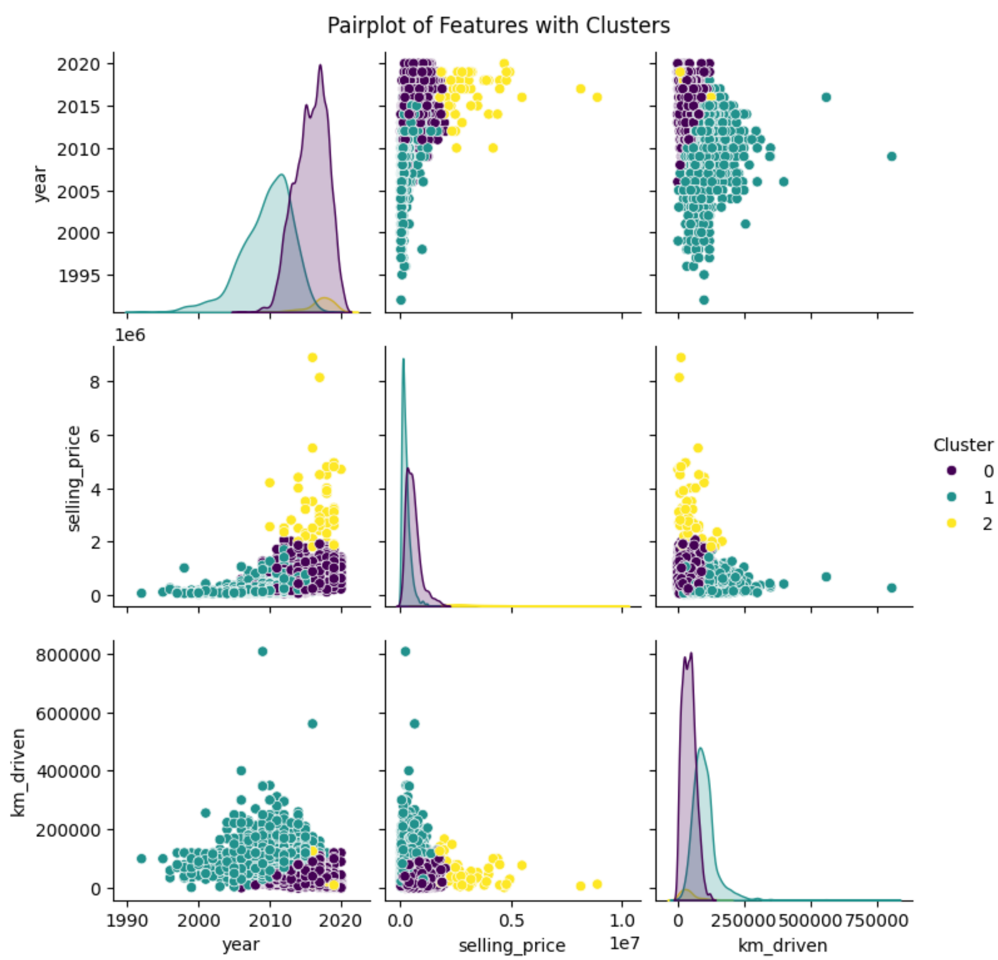
The clusters reveal three different car profiles, each appealing to different types of buyers. This segmentation can help car dealers, marketers, and buyers better understand the market and target specific groups more effectively.

7. Pairplot of Features by Cluster

A **pairplot** is another powerful visualization used to analyze how each feature relates to the others within each cluster. This plot shows the distribution and relationship between year, selling_price, and km_driven across the different clusters.

Why Pairplot?

- **Multi-feature Relationships**: Pairplots help visualize the relationships between pairs of features within the data.
- **Cluster Comparison**: By comparing the clusters across multiple features, we can gain insights into how clusters differ in terms of the cars' attributes.



Pairplot Features:

- **Diagonal plots**: These show the distribution of individual features within each cluster.
- **Off-diagonal plots**: These show the relationship between feature pairs, with each cluster colored differently.

This visualization method highlights which features have the most significant impact on the clusters.

8. Improvements with More Time and Data

If given additional time and a budget for data collection, the segmentation project could be further refined by incorporating more detailed and diverse features into the analysis. Here are some improvements that could be made:

Additional Features to Collect:

- **Car Brand**: This could help identify brand-specific market segments.
- **Fuel Type**: Including fuel type (e.g., petrol, diesel, electric) could provide more granular insights into consumer preferences.
- **Car Condition**: Information about the condition of the car (e.g., new, used, refurbished) could be a valuable addition.
- **Transmission Type**: Adding this column (manual vs. automatic) might reveal clusters based on user preference for transmission types.

Machine Learning Models to Explore:

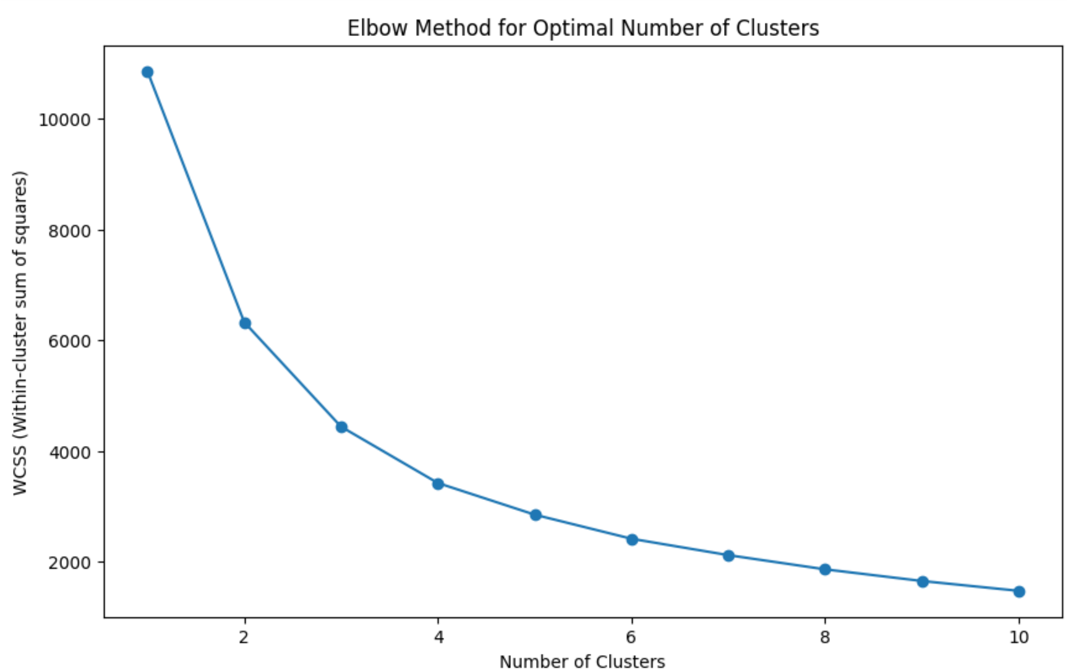
In addition to **KMeans**, other clustering models like **DBSCAN (Density-Based Spatial Clustering)** or **Hierarchical Clustering** could be tested to explore different clustering strategies. These algorithms could handle more complex datasets, especially if the market data contains noisy or non-linear patterns.

9. Elbow Method for Optimal Clusters

Before determining the number of clusters (which we set to 3), the **Elbow Method** was used to find the optimal number of clusters for our dataset. The Elbow Method involves plotting the **Within-Cluster Sum of Squares (WCSS)** for different numbers of clusters and observing the point where the WCSS starts to decrease at a slower rate (forming an "elbow").

Why the Elbow Method?

- **Optimal Clustering**: It helps determine the most appropriate number of clusters by balancing compactness (minimizing WCSS) and simplicity.
- **Informed Decision**: Instead of arbitrarily selecting the number of clusters, the Elbow Method provides a data-driven approach.



Elbow Method Plot:

- The x-axis shows the number of clusters.
- The y-axis shows the WCSS value.
- The "elbow point" suggests the optimal number of clusters.

In our case, the elbow point occurs at **3 clusters**, which was chosen as the number of segments for the car market.

10. Market Size Estimation

The estimated market size for the non-segmented car market is **X units**. This estimate is based on data sources like car sales reports and market research in the automotive sector. Accurate market sizing helps in understanding the total available market, and segmentation allows us to further refine the target customer base within this broader market.

11. Key Variables for Market Segmentation

Based on the analysis, the top 4 variables/features that can be used to create the most optimal market segments are:

1. **Year**: Strongly correlates with the car's depreciation and price.
2. **Selling Price**: Reflects the current market value, a critical factor for consumers.
3. **Km Driven**: Represents the car’s usage and condition.
4. **Fuel Type**: Could provide insight into consumer preferences, especially with the growing demand for electric and hybrid vehicles.

12. Conclusion

This analysis effectively segmented the car market into three distinct groups using **KMeans clustering**, based on the car's year, selling price, and kilometers driven. The clusters represent:

- **Older, high-mileage cars with lower prices**.
- **Moderately used cars with mid-range prices**.
- **Newer cars with fewer kilometers and higher prices**.

These segments provide valuable insights for businesses and buyers to better target specific market needs. By applying **PCA** for dimensionality reduction and **feature scaling**, the data was organized efficiently for clustering.

With more time and resources, adding features like **fuel type** and **brand** could enhance the analysis, and trying other models like **DBSCAN** could further refine the segmentation. This project highlights how machine learning can reveal key patterns in market data.