```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

```

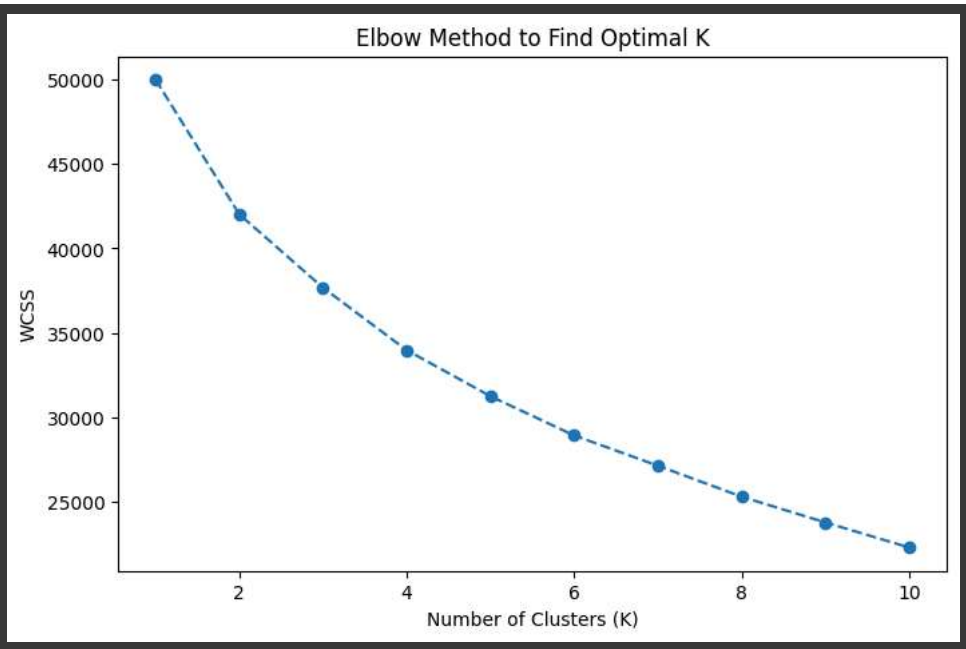
```
# Load dataset
df = pd.read_csv("/content/car_price_dataset.csv")

# Select numerical features for clustering
X = df.select_dtypes(include=[np.number]).drop(columns=['Price'])  # Exclude target variable

# Scale features for better clustering performance
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

print("Data Preprocessing Complete!")
print("Shape of Processed Data:", X_scaled.shape)

```

```
Data Preprocessing Complete!
Shape of Processed Data: (10000, 5)
```

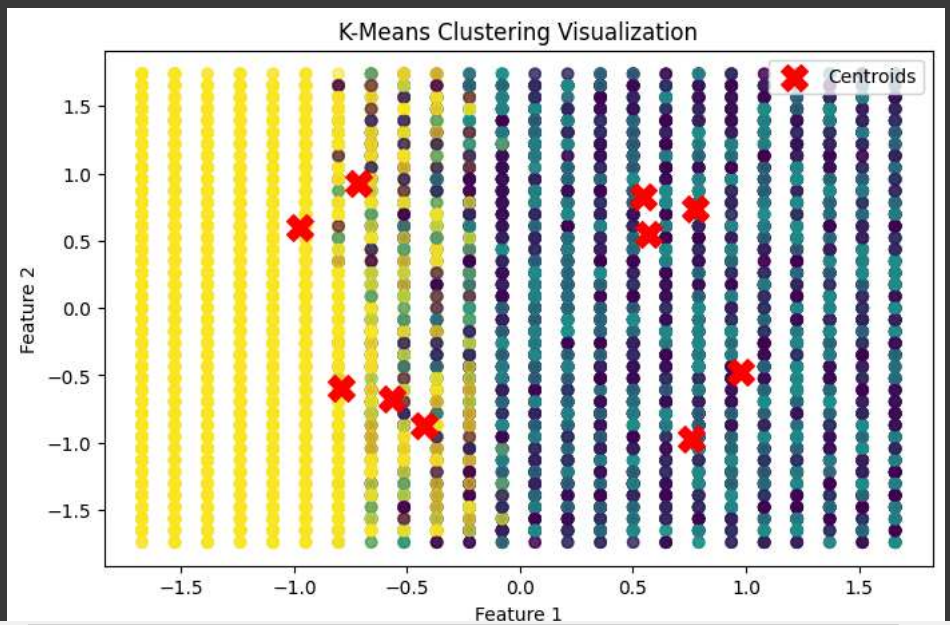
```
# Set the number of clusters (K)
k = 3  # You can vary this

# Apply K-Means
kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
clusters = kmeans.fit_predict(X_scaled)

# Add cluster labels to original dataset
df['Cluster'] = clusters

print("K-Means Clustering Applied Successfully!")
print(df[['Cluster', 'Price']].head(10))  # Display first 10 cluster assignments

```

```
K-Means Clustering Applied Successfully!
   Cluster  Price
0        0   8501
1        0  12092
2        1  11171
3        0  11780
4        2   2867
5        2   7242
6        0  11208
7        2   7950
8        0   9926
9        2   6545
```

```
wcss = []  # Within-cluster sum of squares

for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)

# Plot the elbow curve
plt.figure(figsize=(8, 5))
plt.plot(range(1, 11), wcss, marker='o', linestyle='--')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('WCSS')
plt.title('Elbow Method to Find Optimal K')
plt.show()

```




```
plt.figure(figsize=(8, 5))
plt.scatter(X_scaled[:, 0], X_scaled[:, 1], c=clusters, cmap='viridis', alpha=0.6)
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], marker='X', s=200, c='red', label='Centroids')
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.title("K-Means Clustering Visualization")
plt.legend()
plt.show()

```




```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```