

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
```

```
prd=pd.read_excel('/content/Dataset-SA-Numeric-With-Review-And-ProductName.xlsx')
```

```
prd.head()
```

	product_name	product_price	Rate	Review	Sentiment
0	Candes 12 L Room/Personal Air Cooler?????? (Whi...	3999	5	super!	1
1	Candes 12 L Room/Personal Air Cooler?????? (Whi...	3999	5	awesome	1
2	Candes 12 L Room/Personal Air Cooler?????? (Whi...	3999	3	fair	1
3	Candes 12 L Room/Personal Air Cooler?????? (Whi...	3999	1	useless product	-1
4	Candes 12 L Room/Personal Air Cooler?????? (Whi...	3999	3	fair	0

```
prd.isnull().sum()
```

```

0
product_name 0
product_price 0
Rate 0
Review 0
Sentiment 0
```

```
dtype: int64
```

```
prd.info ()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180385 entries, 0 to 180384
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  -
0   product_name    180385 non-null object
1   product_price   180385 non-null int64
2   Rate            180385 non-null int64
3   Review          180385 non-null object
4   Sentiment       180385 non-null int64
dtypes: int64(3), object(2)
memory usage: 6.9+ MB
```

```
prd.shape
```

```
(180385, 5)
```

```
x = prd.iloc[:, [2, 4]].values
```

```
wcss = []
```

```
for i in range(1, 12):  
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state=45)  
    kmeans.fit(x)  
    wcss.append(kmeans.inertia_)
```

```
sns.set()
```

```
plt.figure(figsize=(8, 5))
```

```
plt.plot(range(1, 12), wcss, marker='o')
```

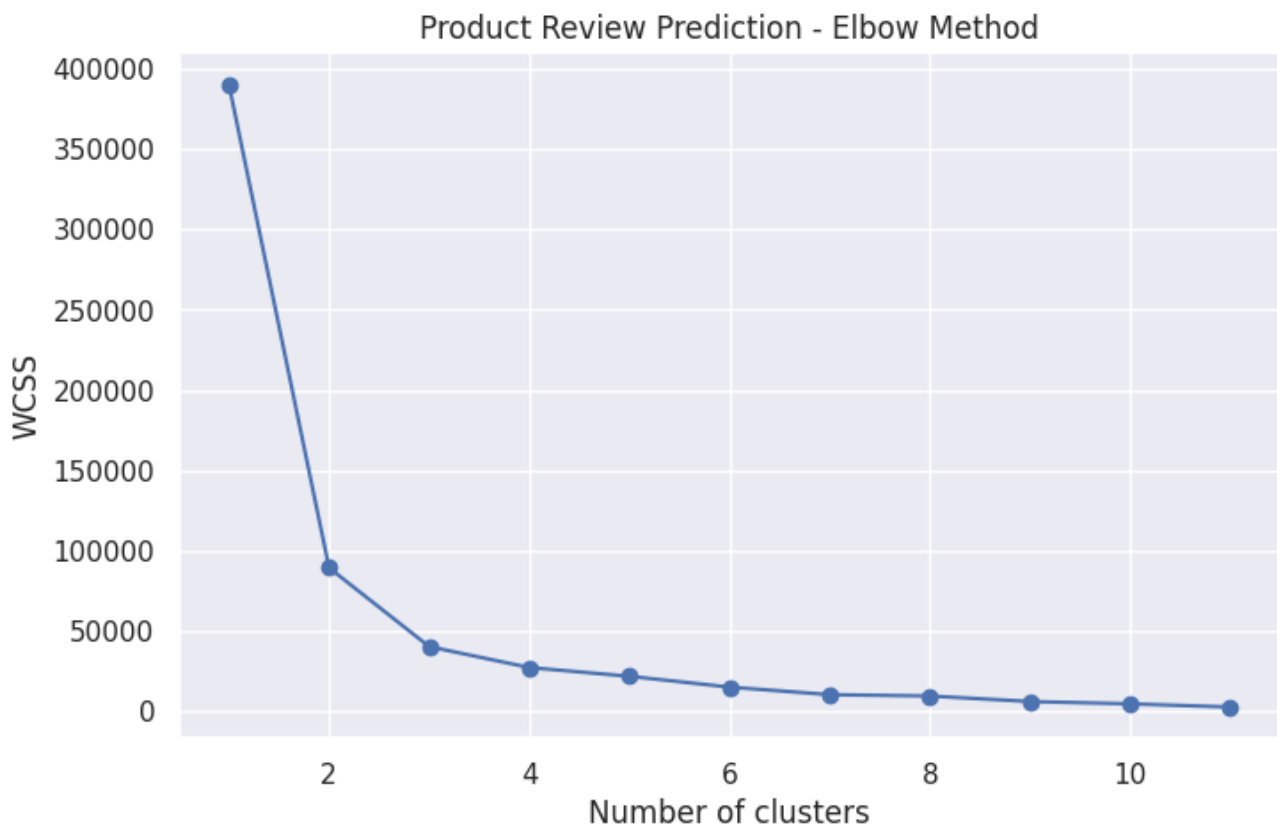
```
plt.title('Product Review Prediction - Elbow Method')
```

```
plt.xlabel('Number of clusters')
```

```
plt.ylabel('WCSS')
```

```
plt.grid(True)
```

```
plt.show()
```



```
K = KMeans(n_clusters=8, init='k-means++', random_state=0)
```

```
y = K.fit_predict(x)
```

```
print(y)
```

```
[0 0 4 ... 4 0 6]
```

```

plt.figure(figsize=(8, 6))
colors = ['red', 'blue', 'green', 'orange', 'purple', 'brown', 'pink', 'gray']
for i in range(8):
    plt.scatter(x[y == i, 0], x[y == i, 1], s=50, color=colors[i], label=f'Cluster {i+1}')
plt.scatter(K.cluster_centers[:, 0], K.cluster_centers[:, 1], s=300, c='cyan', label='Centroids')
plt.title('Clusters of Product Reviews')
plt.xlabel('Rate')
plt.ylabel('Sentiment')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

```

<ipython-input-13-1808984679>:11: UserWarning: Creating legend with loc="best" can be slow with large amounts of data.

```
plt.tight_layout()
```

