**FACEBOOK DATASET**

* **Sharon Blessy J V**

1. **How does the time of upload [‘status published’] affect the 'num reaction’?**

**Ans.**

***Code Snippet:***

**from datetime import time**

**fd['status\_published'] = pd.to\_datetime(fd['status\_published'])**

**fd['date'] = fd['status\_published'].dt.date**

**fd['time'] = fd['status\_published'].dt.time**

**def categorize\_time(t):**

**if time(0, 0) <= t < time(12, 0):**

**return 'Morning'**

**elif time(12, 0) <= t < time(15, 0):**

**return 'Afternoon'**

**elif time(15, 0) <= t <= time(23, 59):**

**return 'Evening'**

**else:**

**return 'Unknown'**

**fd['Time\_Category'] = fd['time'].apply(categorize\_time)**

**relation = fd.groupby(['Time\_Category','date'])['num\_reactions'].sum().unstack()**

**relation\_reset = relation.reset\_index()**

**relation\_melted = relation\_reset.melt(id\_vars='Time\_Category', var\_name='date', value\_name='num\_reactions')**

**sns.barplot(x='Time\_Category', y='num\_reactions', data=relation\_melted, palette="viridis")**

**plt.xlabel('Time of Upload')**

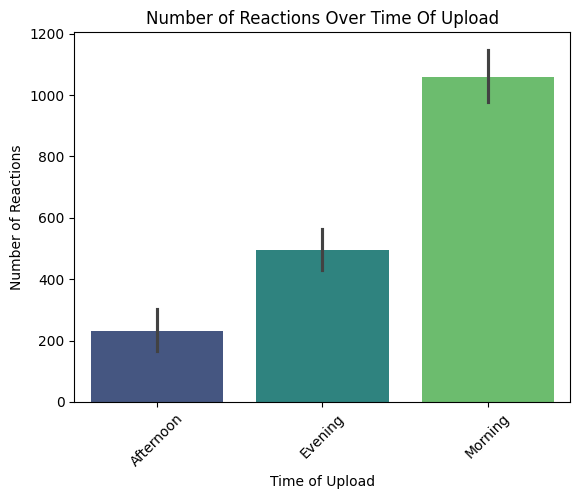
**plt.ylabel('Number of Reactions')**

**plt.title('Number of Reactions Over Time')**

**plt.xticks(rotation=45)**

**plt.show()**

***Output:***

****

Posts uploaded in the morning have a greater no, of reactions when compared to posts uploaded in the afternoon and evening.

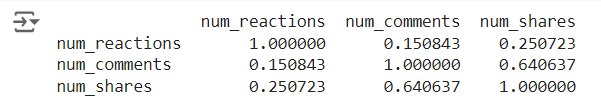
1. **Is there a correlation between the number of reactions (num\_reactions) and other engagement metrics such as comments (num\_comments) and shares (num\_shares)? If so, what is the strength and direction of this correlation?**

**Ans.**

***Code Snippet:***

**correlation\_matrix = fd[['num\_reactions', 'num\_comments', 'num\_shares']].corr()**

**print(correlation\_matrix)**

***Output:***

1. **Use the columns status\_type, num\_reactions, num\_comments, num\_shares, num\_likes, num\_loves, num\_wows, num\_hahas, num\_sads, and num\_angrys to train a K-Means clustering model on the Facebook Live Sellers dataset.**

**Ans.**

***Code Snippet:***

**columns = ['status\_type', 'num\_reactions', 'num\_comments', 'num\_shares',**

**'num\_likes', 'num\_loves', 'num\_wows', 'num\_hahas', 'num\_sads', 'num\_angrys']**

**data = fd[columns]**

**data = pd.get\_dummies(data, columns=['status\_type'], drop\_first=True)**

**scaler = StandardScaler()**

**data\_scaled = scaler.fit\_transform(data)**

**wcss = []**

**for i in range(1, 11):**

**kmeans = KMeans(n\_clusters=i, init='k-means++', random\_state=42)**

**kmeans.fit(data\_scaled)**

**wcss.append(kmeans.inertia\_)**

**plt.plot(range(1, 11), wcss, marker = 'o')**

**plt.title('Elbow Method')**

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

**plt.ylabel('WCSS')**

**plt.show()**

**# Train the K-Means model with 3 clusters**

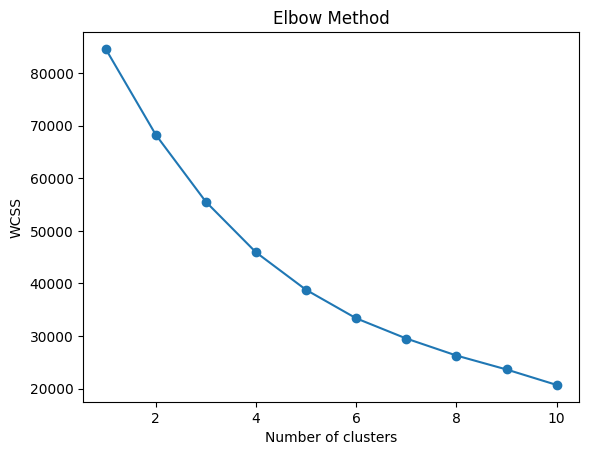
**kmeans = KMeans(n\_clusters=3, init='k-means++', random\_state=42)**

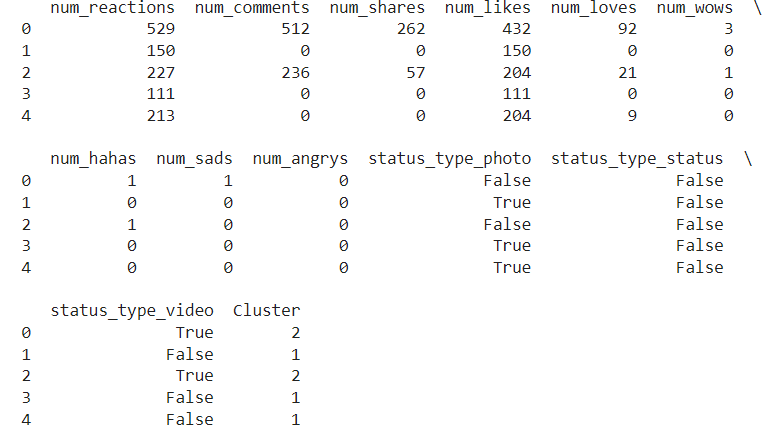
**clusters = kmeans.fit\_predict(data\_scaled)**

**# Add cluster labels to the original dataset**

**data['Cluster'] = clusters**

**print(data.head())**

***Output:***

****

1. **Use the elbow method to find the optimum number of clusters.**

**Ans.**

***Code Snippet:***

**columns = ['status\_type', 'num\_reactions', 'num\_comments', 'num\_shares',**

**'num\_likes', 'num\_loves', 'num\_wows', 'num\_hahas', 'num\_sads', 'num\_angrys']**

**data = fd[columns]**

**data = pd.get\_dummies(data, columns=['status\_type'], drop\_first=True)**

**scaler = StandardScaler()**

**data\_scaled = scaler.fit\_transform(data)**

**# Elbow method to find the optimal number of clusters**

**wcss = []**

**for i in range(1, 11):**

**kmeans = KMeans(n\_clusters=i, init='k-means++', max\_iter=300, n\_init=10, random\_state=42)**

**kmeans.fit(data\_scaled)**

**wcss.append(kmeans.inertia\_)**

**# Plot the Elbow graph**

**plt.figure(figsize=(8, 6))**

**plt.plot(range(1, 11), wcss, marker='o', linestyle='--')**

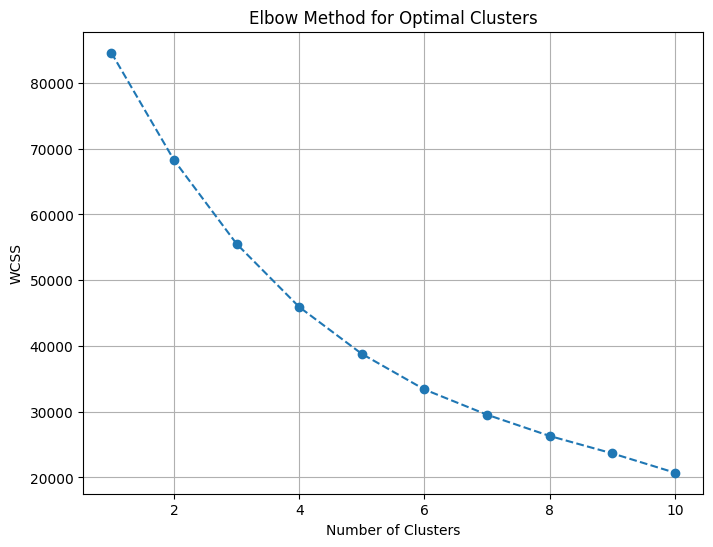
**plt.title('Elbow Method for Optimal Clusters')**

**plt.xlabel('Number of Clusters')**

**plt.ylabel('WCSS')**

**plt.grid(True)**

**plt.show()**

***Output:***

1. **What is the count of different types of posts in the dataset?**

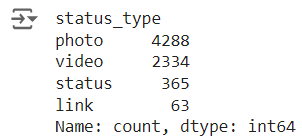
**Ans.**

***Code Snippet:***

**post\_type\_counts = fd['status\_type'].value\_counts()**

**print(post\_type\_counts)**

***Output:***

****

Most posts are photos while a very few are links.

1. **What is the average value of num\_reaction, num\_comments, num\_shares for each post type?**

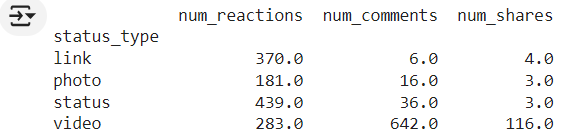
**Ans.**

***Code Snippet:***

**avg\_values = fd.groupby('status\_type')[['num\_reactions', 'num\_comments', 'num\_shares']].mean().round()**

**print(avg\_values)**

***Output:***

****

The above table shows the average no. of reactions, comments and shares for each post type.