REPORT ON FACEBOOK DATASET

1. **PROBLEM STATEMENT**

To train a K-Means clustering model on the Facebook Live Sellers dataset..

1. **DATASET**

I was given “The Facebook Live Sellers in Thailand” dataset that contains information about the Facebook pages of 10 Thai fashion and cosmetics retail sellers. Below is a description of the dataset:

1. Title: Facebook Live Sellers in Thailand Dataset
2. Source: The dataset is sourced from the UCI Machine Learning Repository.
3. Data Type: The dataset is in a tabular format, typically stored in a CSV (Comma Separated Values) file.
4. Number of Instances: There are a total of 7050 instances (rows) in the dataset.
5. Number of Attributes: The dataset initially consists of 16 attributes (columns). After removing redundant columns, there are 14 attributes remaining.
6. Attribute Information:

- status\_id: Unique identifier for each status post.

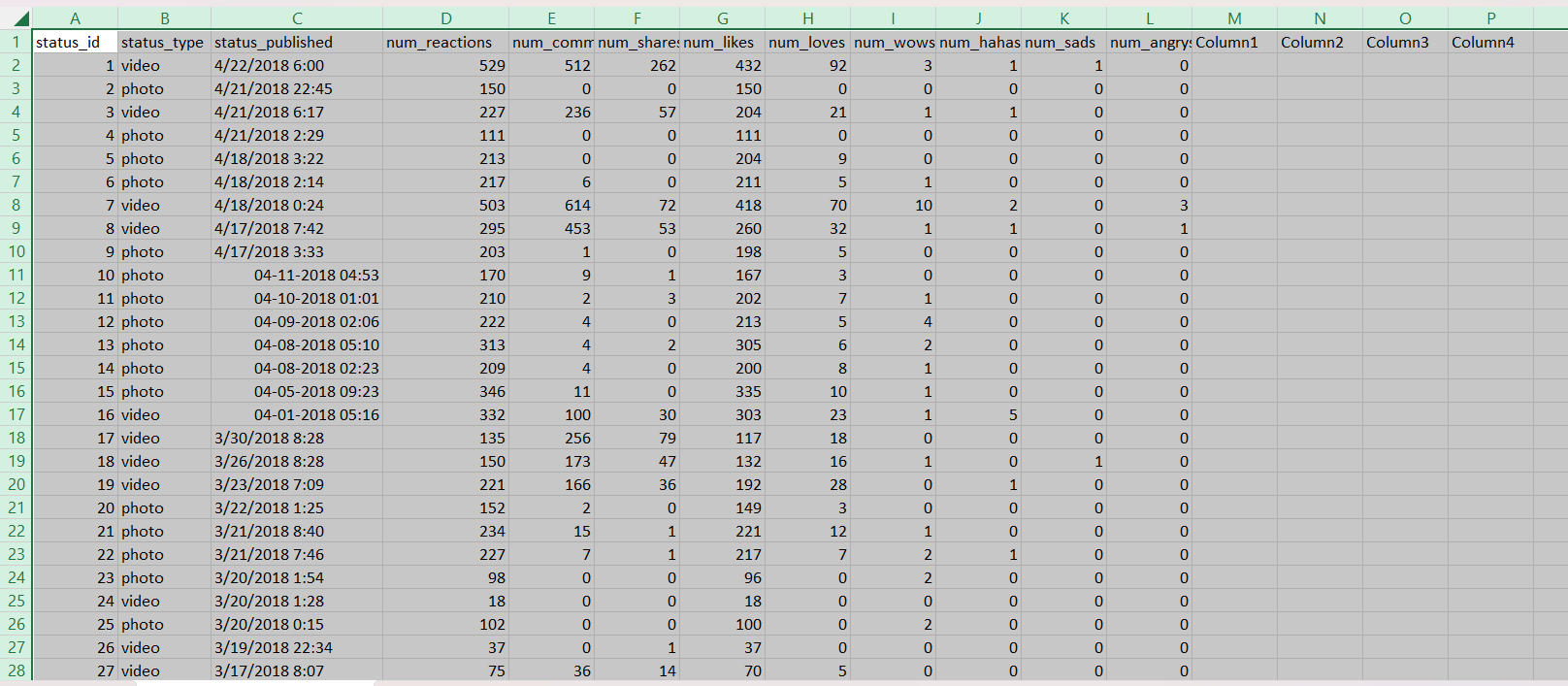
- status\_published: Date and time when the status post was published.

- status\_type: Nature of the status post (e.g., video, photo, status, link).

- num\_reactions: Number of reactions (e.g., likes, loves, wow, haha, sad, angry) received on the status post.

- num\_comments: Number of comments received on the status post.

- num\_shares: Number of shares received on the status post.



1. **METHODOLOGY**

STEP 1: Preprocessing the Data.

* This is the first and crucial step on receiving any dataset. You try and understand the data, look for missing values, drop unnecessary columns, check out the distribution and stretch of the data, learn out the outliers for each feature, transform and scale it, encoding categorical data to make them fit for training so that the model can be trained on the best possible dataset.
* This step is really crucial as it sets the tone for model performance and better accuracy. While training my model, I left some of these steps at first which posed a lot of issues while training the model.
* I will highlight it as we go so that you know what not to do and avoid these silly, basic but important pitfalls.
  + Null values



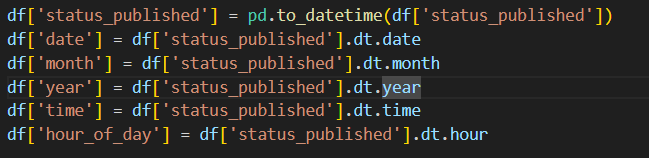
* + Statistical Analysis

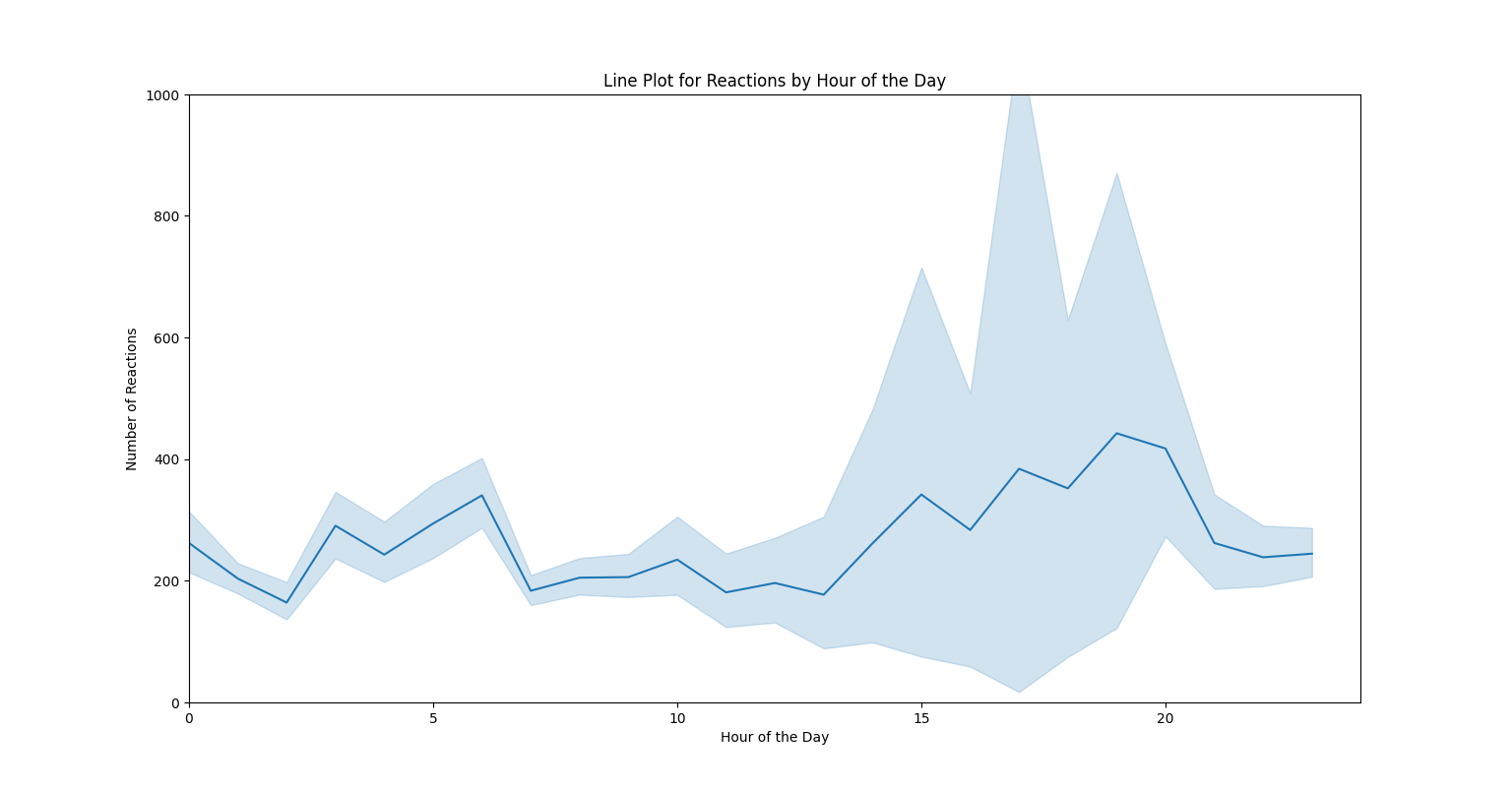
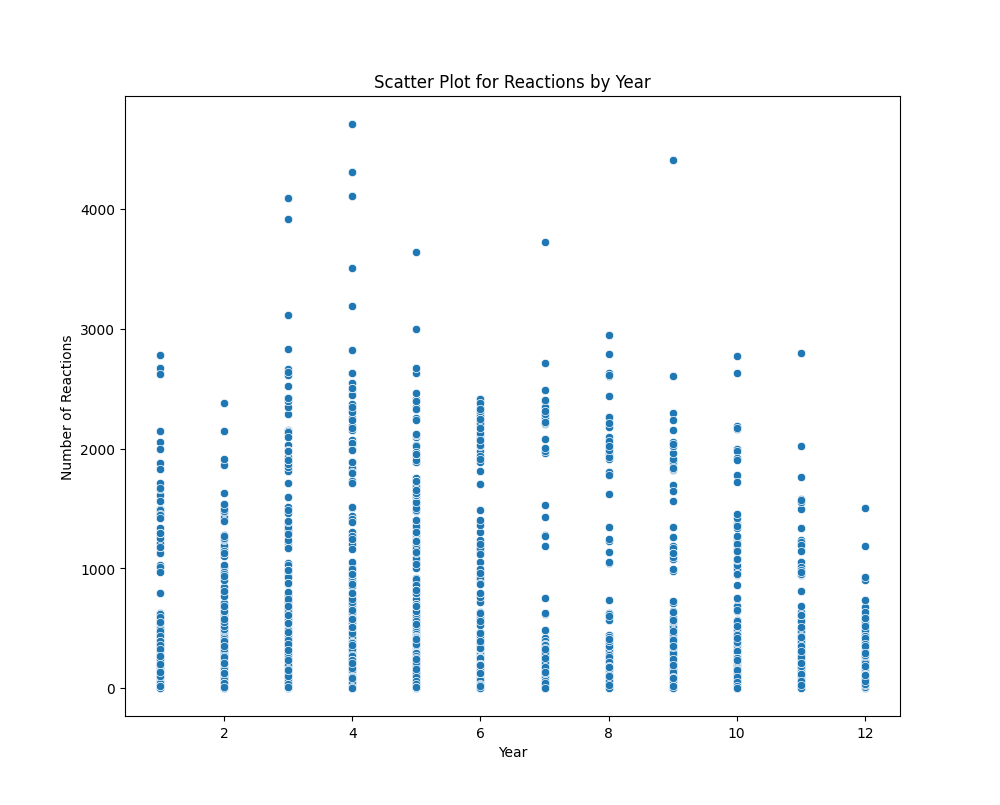
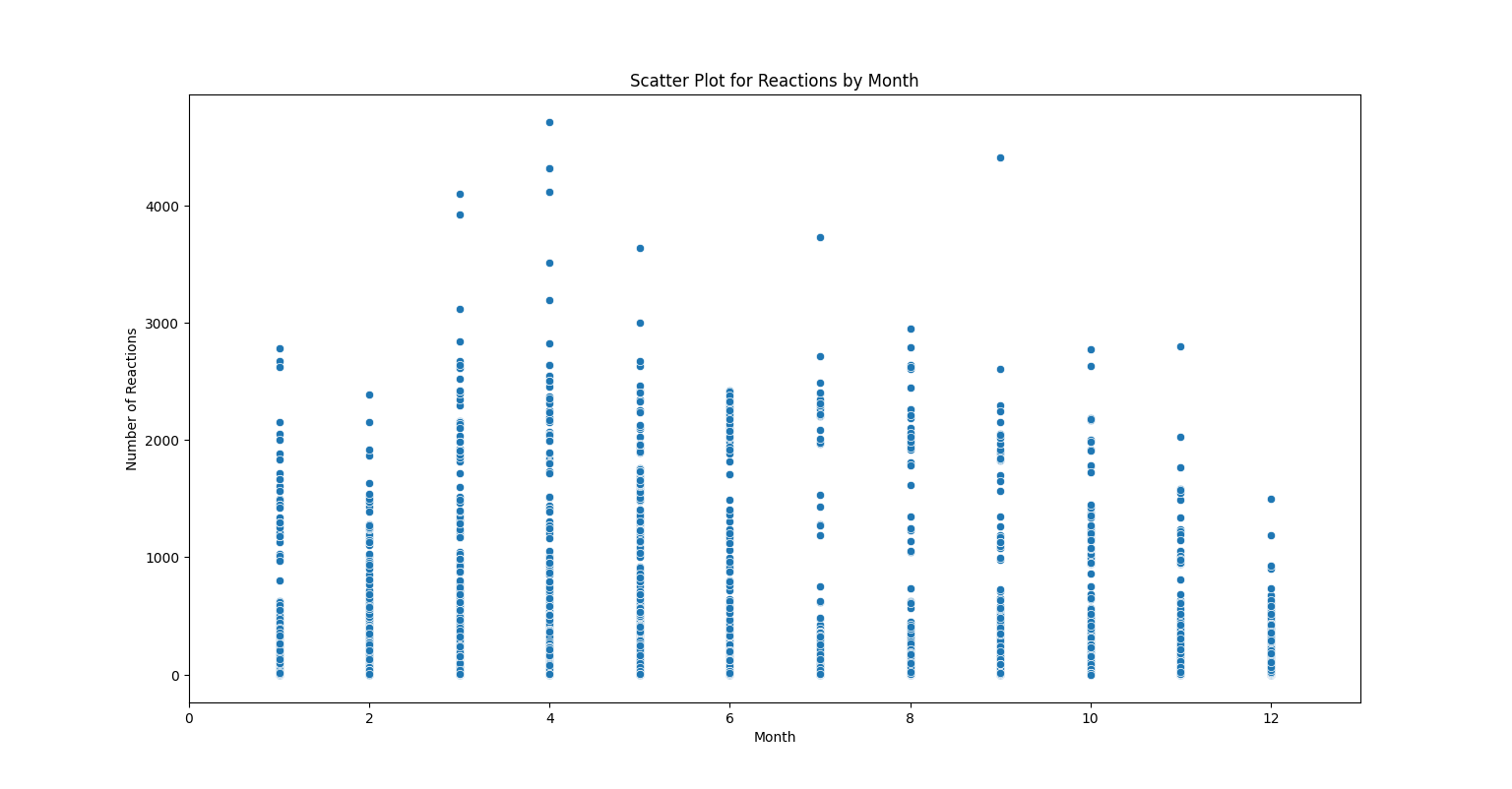
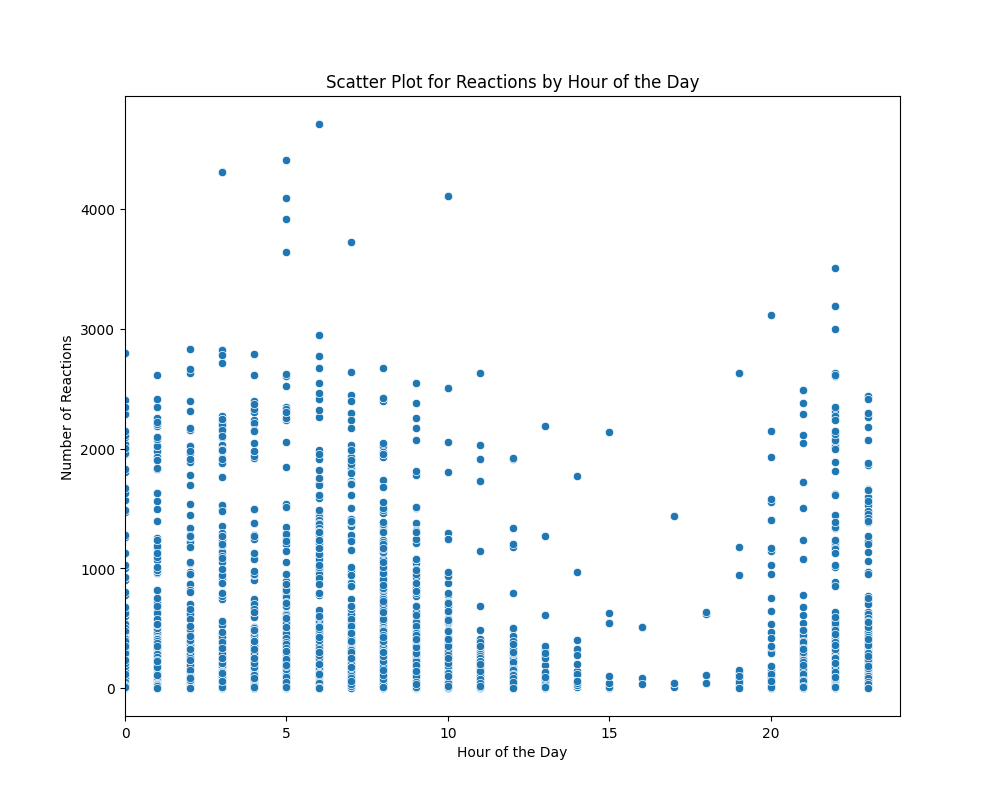


* + Dropping Unnecessary Columns

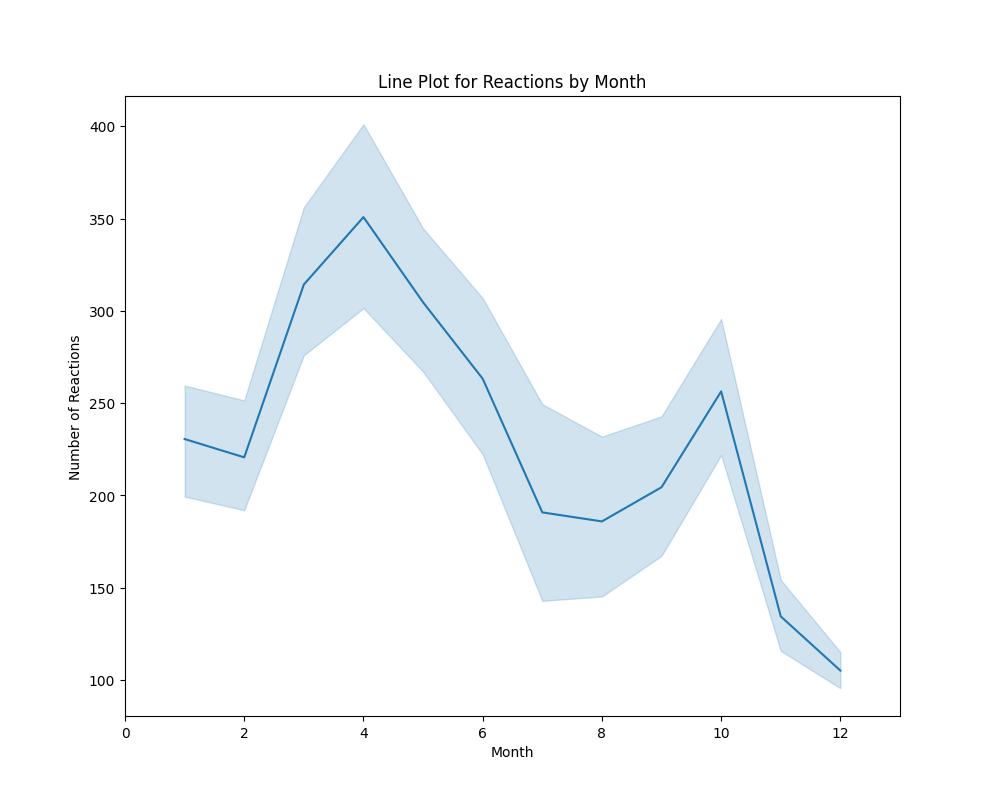


1.2. Finding how does ‘status\_published’ affect the feature ‘num\_reaction’

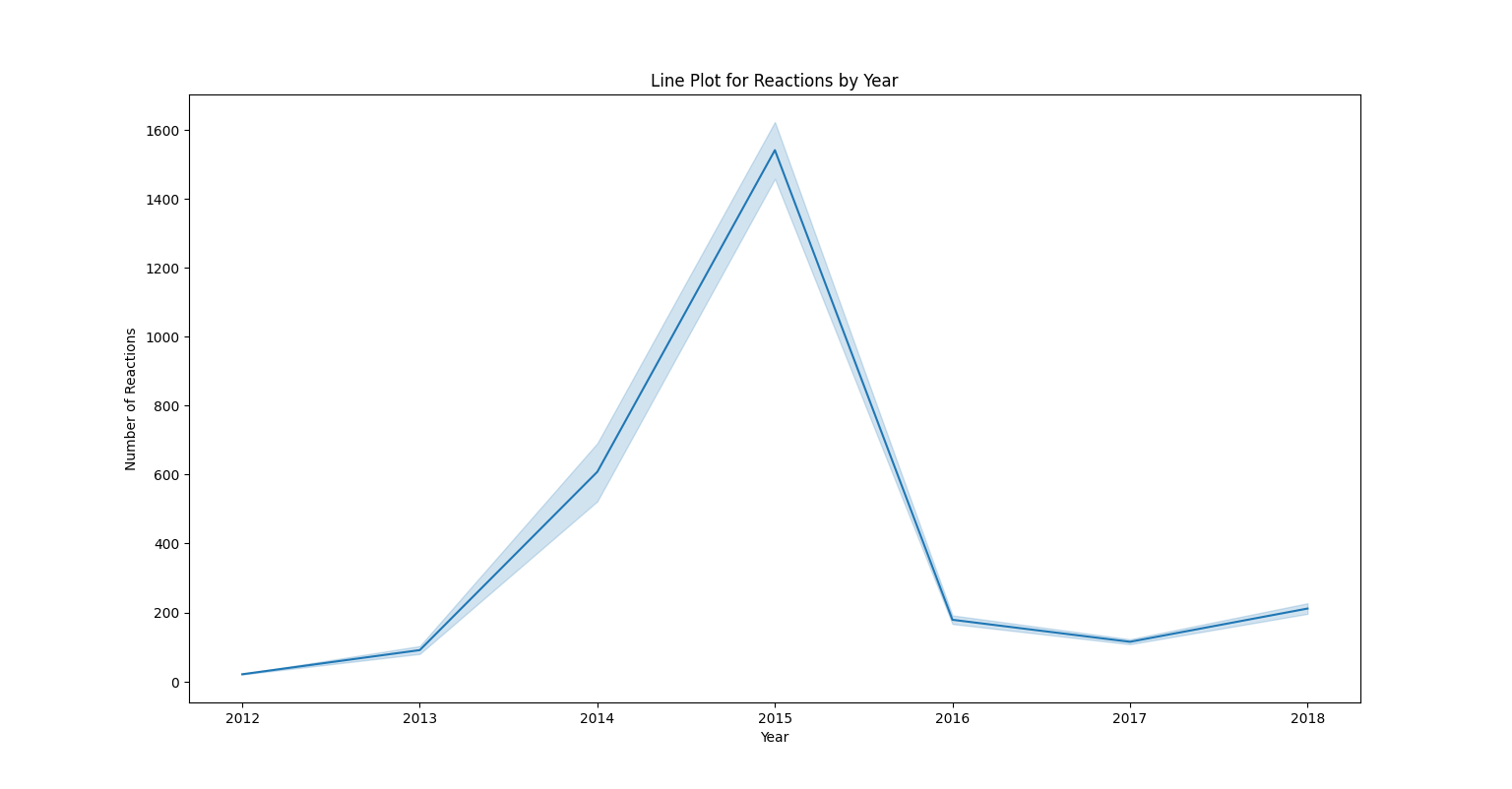
* The primary and basic way to find how one feature affects the other is by tracing out the realationship between them visually..
* But, ‘status\_published’ was of datetime type and therefore, finding a relationship to visualize between them is quite difficult.
* Therefore, I split the status\_published into date and time and then found a relation between the hour of the day and the number of reactions in that hour. 
* After, the columns were separated I was able to make six Graphs to clearly define how one affects the other.



There is a noticeable fluctuation in the number of reactions throughout the day. Early hours show relatively low activity, while there is a significant spike in reactions during certain hours (e.g., late afternoon or evening). This suggests that user activity might peak during specific times, likely when users are more active online.



There appears to be a seasonal pattern in reactions, with higher activity during certain months (e.g., spring) and lower activity toward the end of the year. This could reflect seasonal behaviour or engagement patterns tied to holidays, events, or platform usage cycles.



The number of reactions varies significantly across years, with a sharp peak in 2015 followed by a steep decline. This could indicate a trend tied to external factors such as platform popularity, changes in user behaviour, or specific events influencing reactions during certain years.

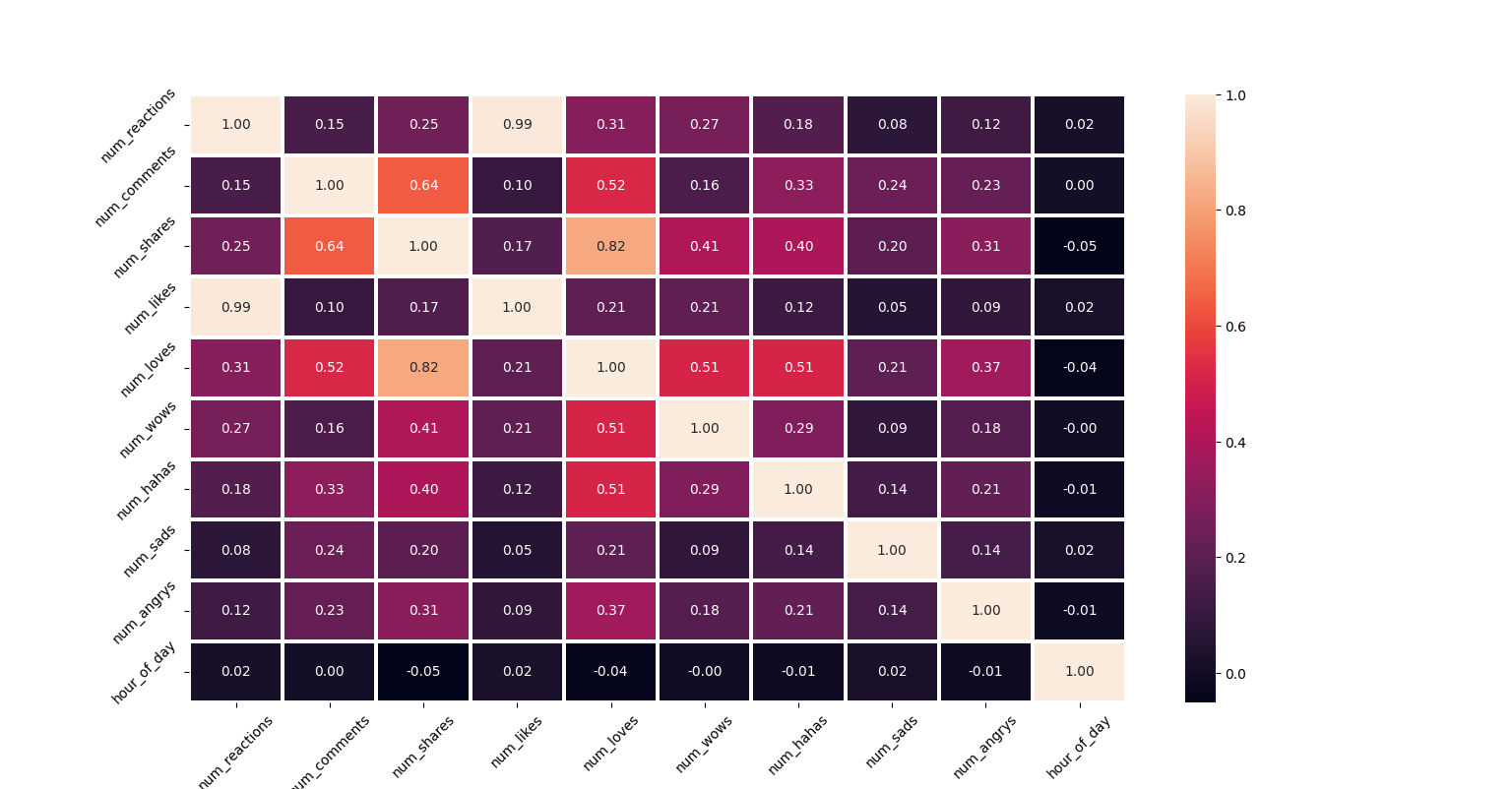
1.3. Correlation Matrix

* **Correlation coefficient** is a statistical measure that shows how strongly two variables are related to each other. It ranges from -1 to 1:
  + 1: Perfect positive correlation. When one variable increases, the other also increases.
  + -1: Perfect negative correlation. When one variable increases, the other decreases.
  + 0: No correlation. Changes in one variable do not predict changes in the other.

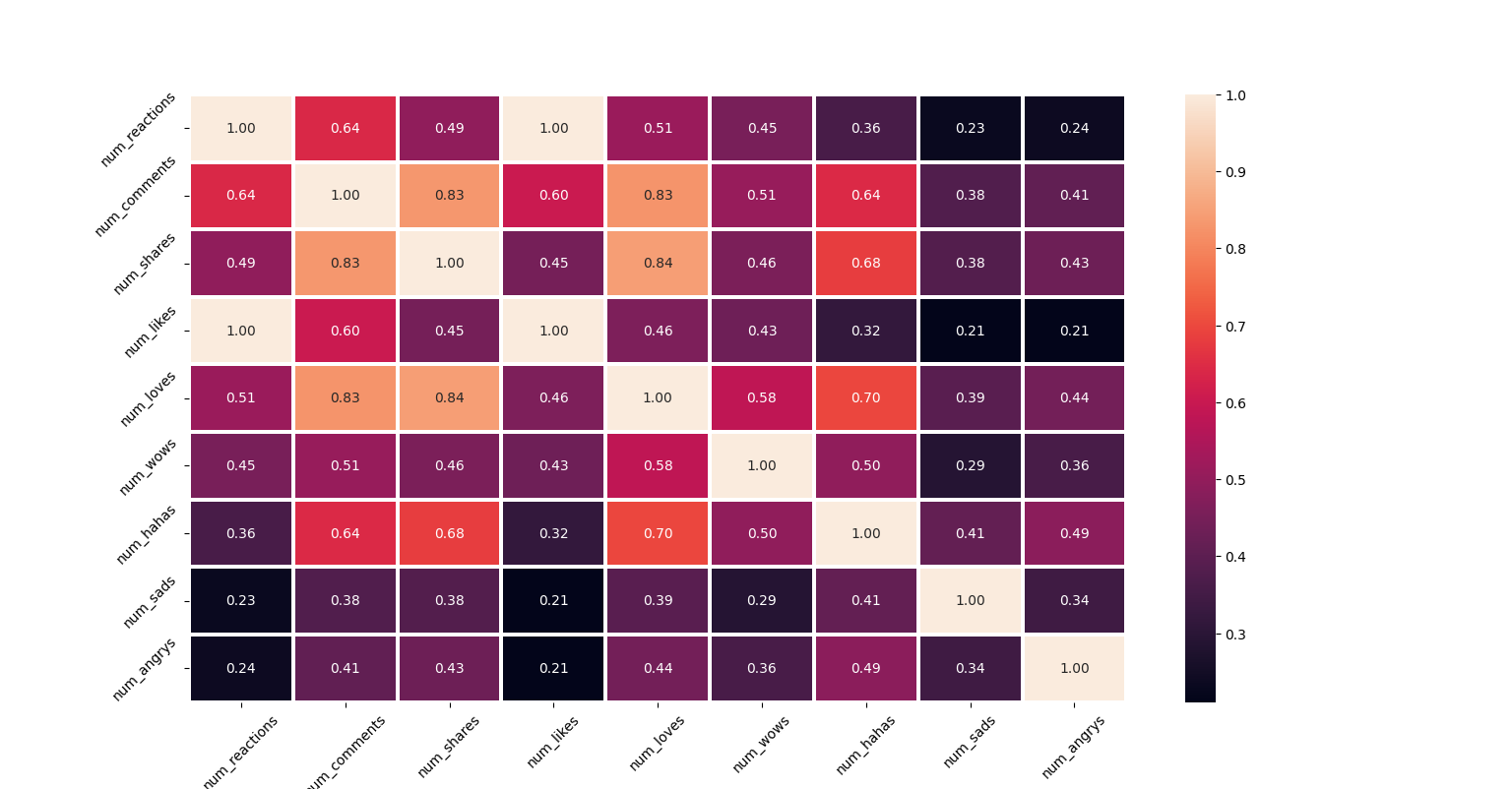
Think of it like a scale that measures how well two things move together. If they move together in the same direction, the correlation is positive. If they move in opposite directions, it's negative. If they don't move together at all, it's zero.

* **Scaling** does not affect the correlation matrix, though **non-linear transformations** do.

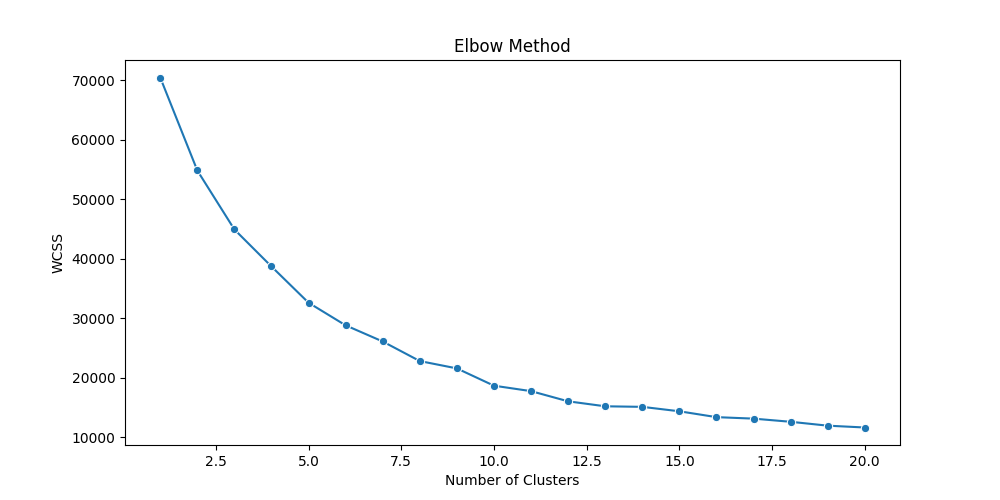
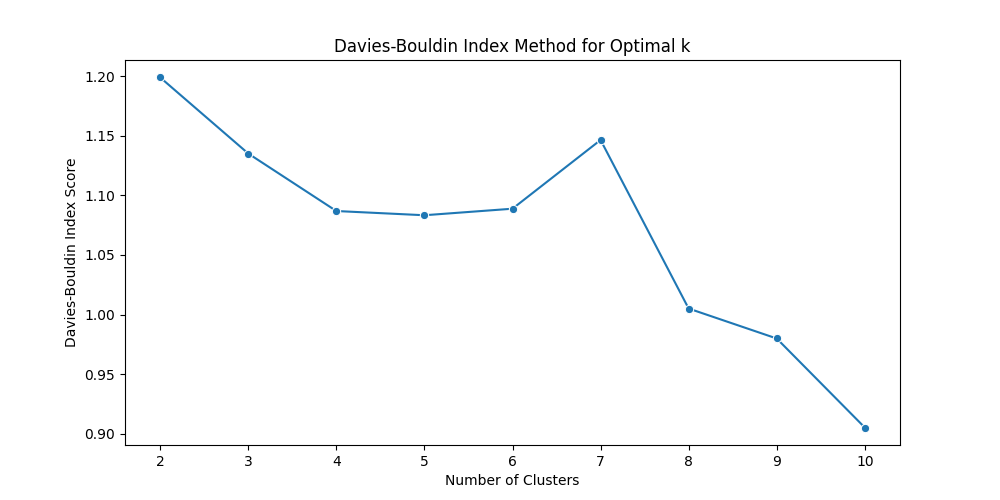
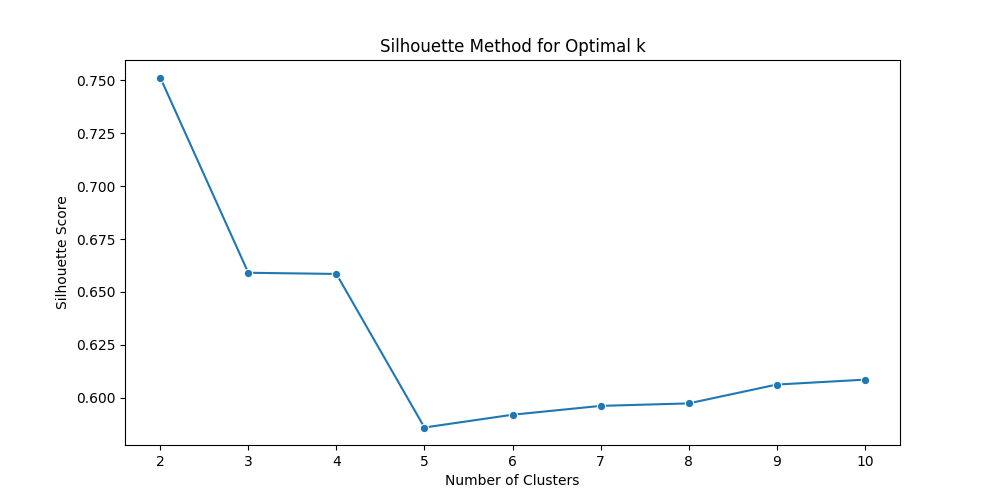
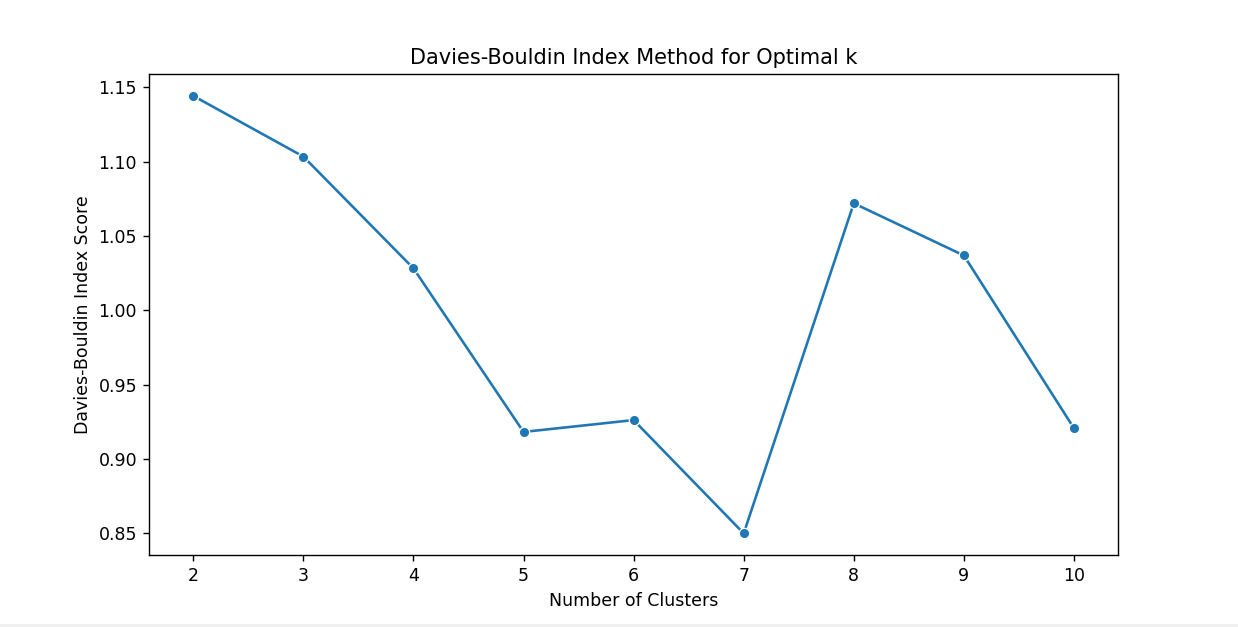
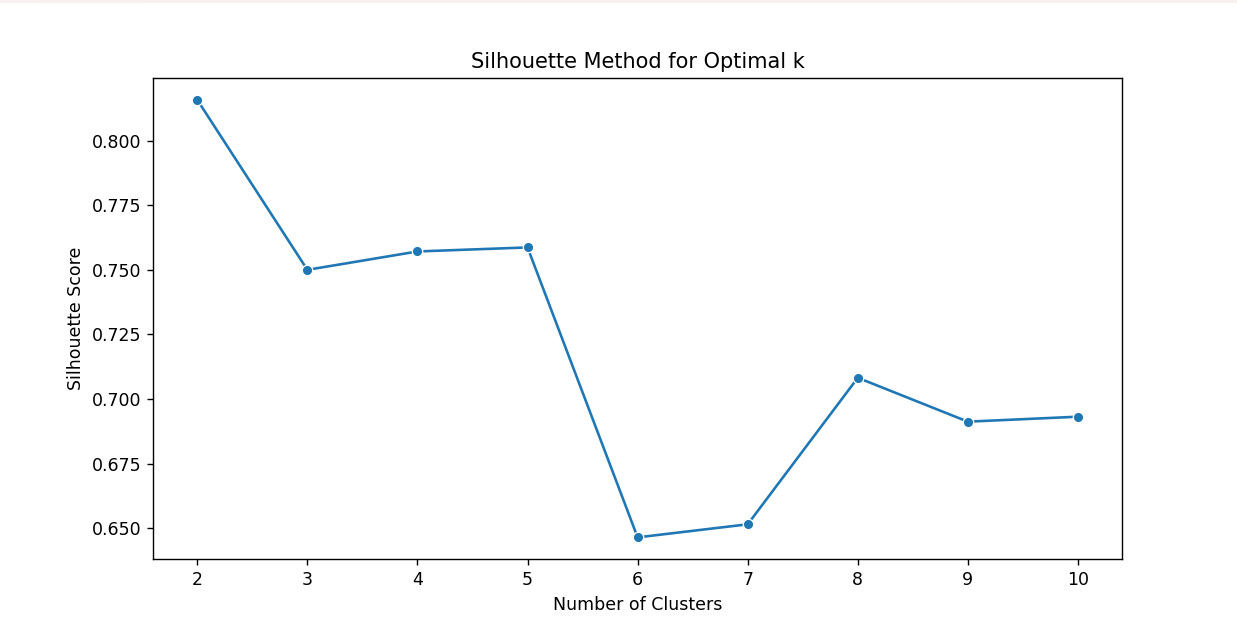
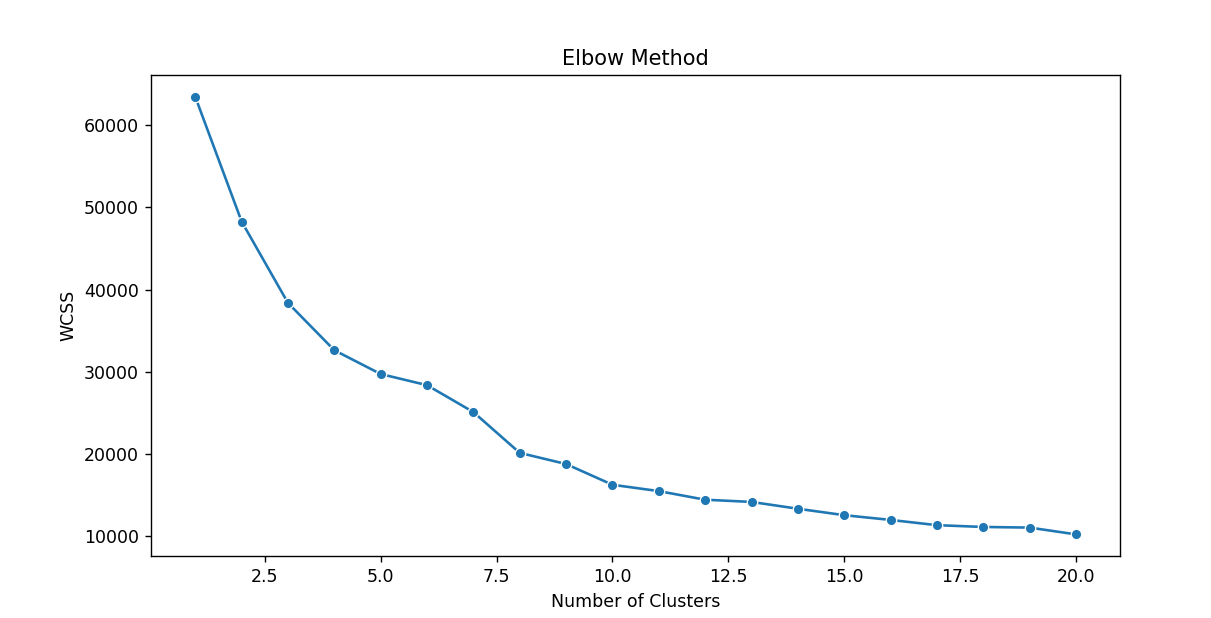
**(BEFORE SCALING AND TRANSFORMATION)**



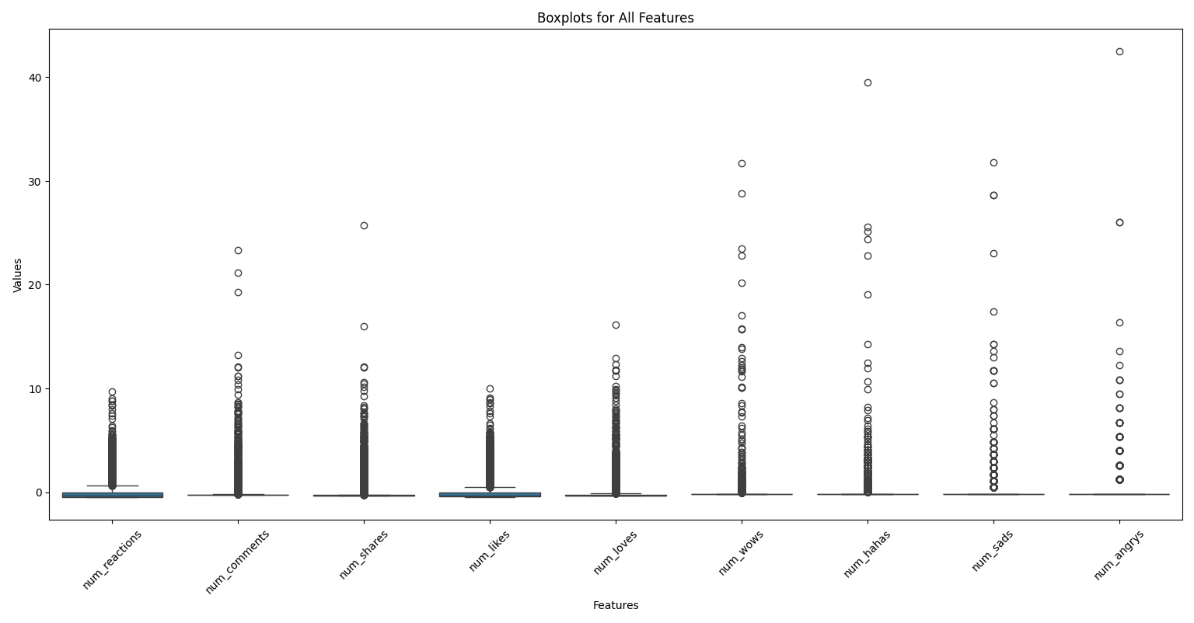
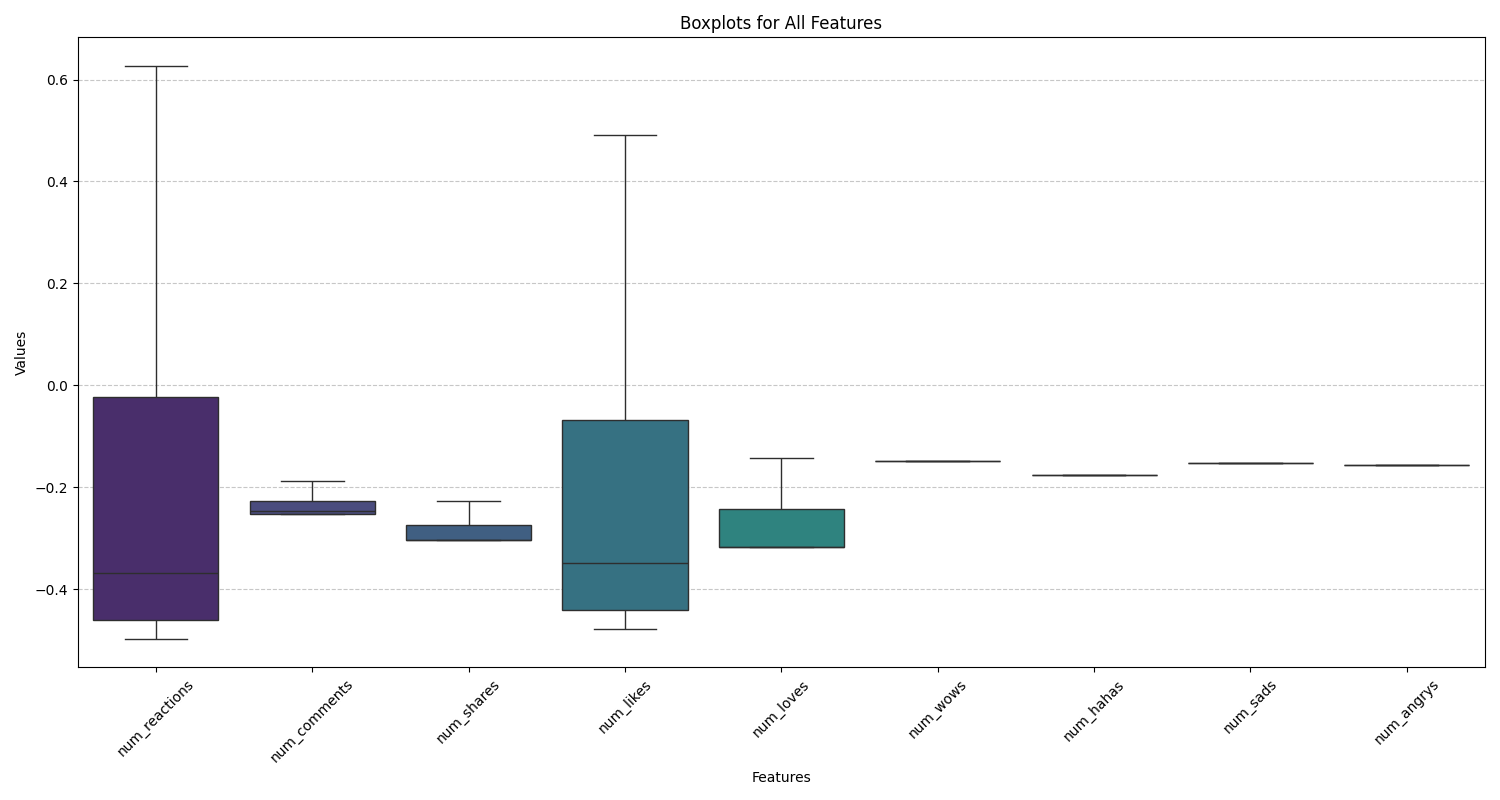
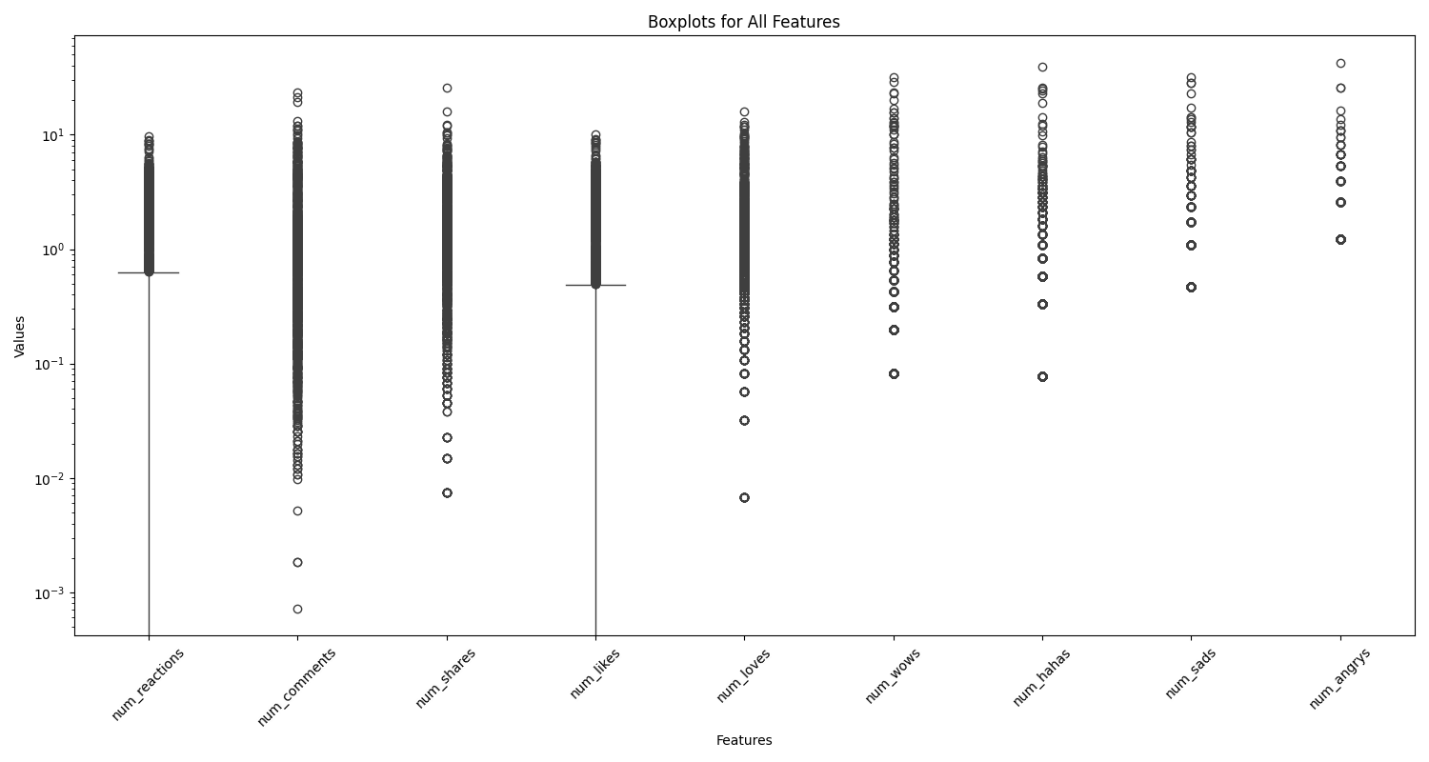
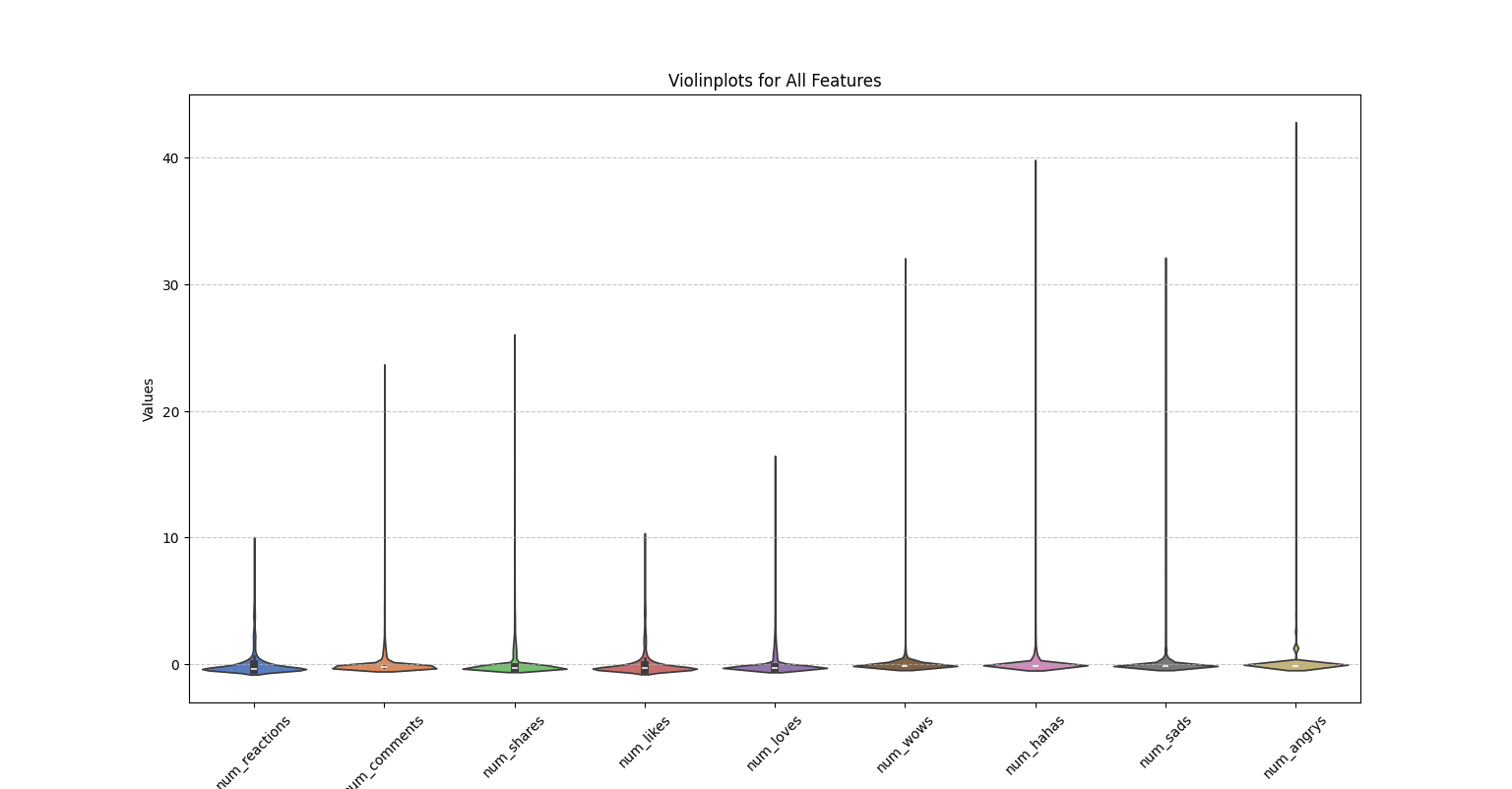
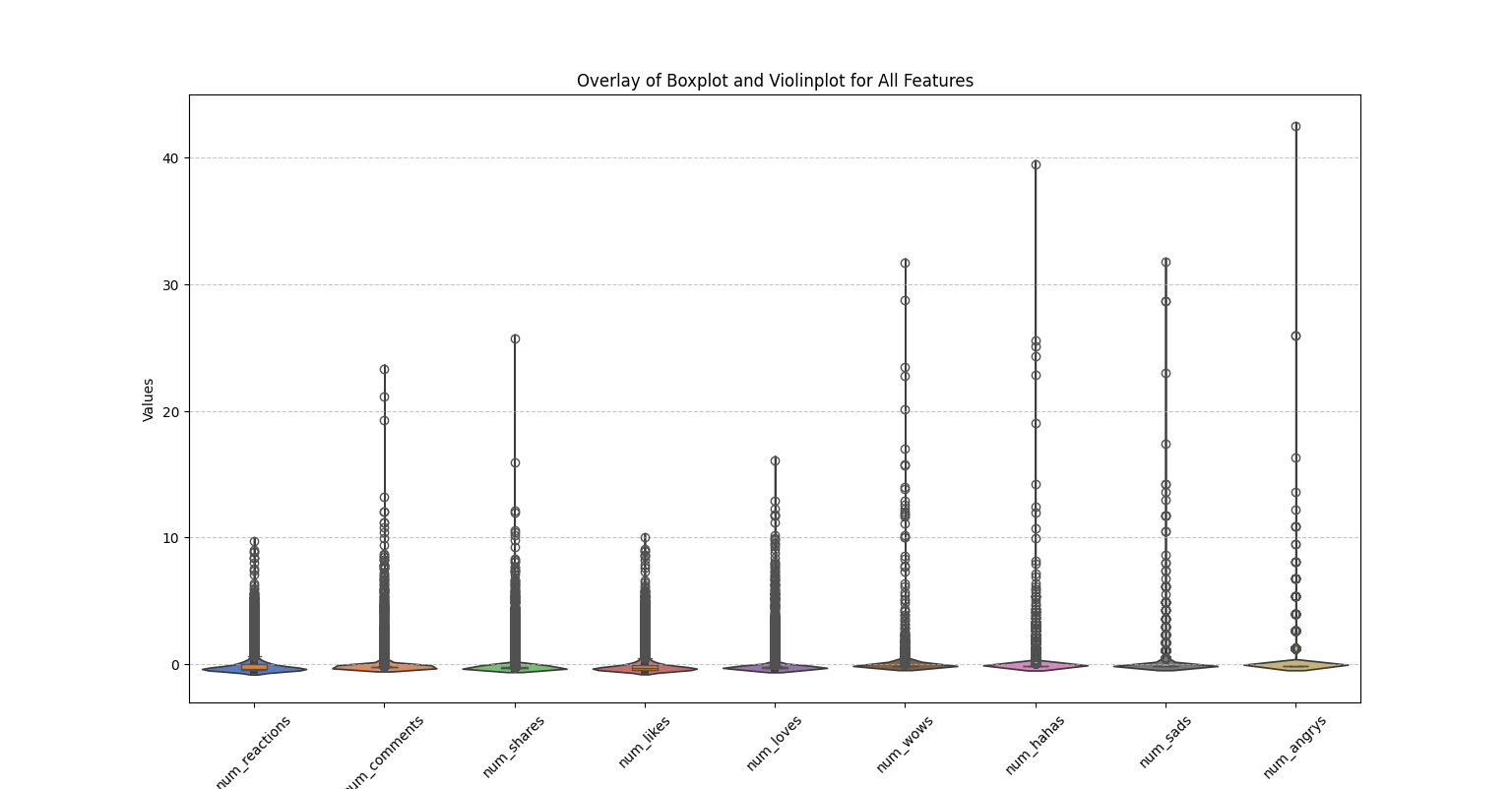
**(AFTER SCALING AND TRANSFORMATION)**



STEP 2: Finding the value of K

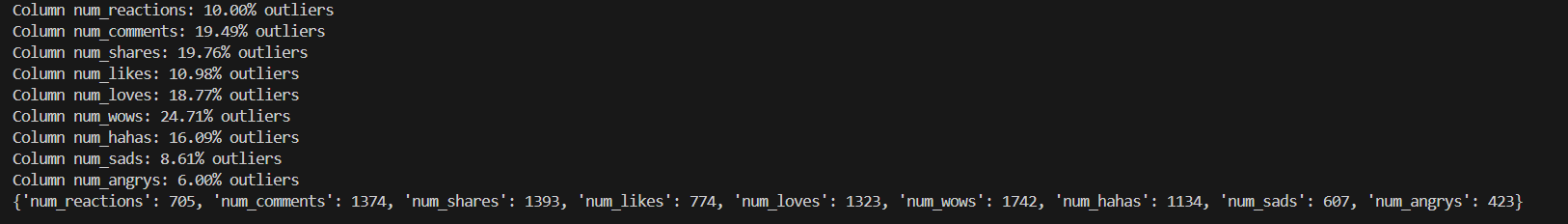
* The **Elbow Method** is a way to figure out the best number of clusters for your data when using clustering algorithms like K-Means. It's called "elbow" because it involves finding a point on a graph that looks like the bend in an elbow. The Elbow Method plots the number of clusters on one axis and the "within cluster sum of squares" (WCSS) on the other. WCSS measures how spread-out points are within each cluster.
* The **Silhouette Score** is a measure used to evaluate how well each data point fits into its assigned cluster compared to other clusters. Imagine you're at a party with different groups of people. The Silhouette Score tells you how well you fit into your group versus how close you are to another group. It helps determine if the clusters are well-separated and cohesive, which is crucial for choosing the right number of clusters in algorithms like K-Means.
  + High Score (Close to 1): You're clearly part of your group and far from others.
  + Low Score (Close to 0): You're right on the edge between groups.
  + Negative Score: You might actually belong to a different group.
* The **Davies-Bouldin Index** is another measure used to evaluate the quality of clusters. It assesses how similar clusters are to each other. Think of it like comparing different teams in a competition. DBI looks at how similar each team is to its nearest rival. It helps identify the optimal number of clusters by ensuring they are distinct and not too similar.
  + Low DBI Score: Teams are very different from each other (good clustering).
  + High DBI Score: Teams are very similar (poor clustering).
* PITFALL 1
  + At first, I only scaled the data. After scaling the data, I tried finding the k value using the basic **Elbow-method.** But that resulted in no absolute value for k as it lacked the clear, sharp elbow. 
  + Therefore, I applied other methods like Silhouette Score and Davies-Bouldin Index to find the value of k. 
  + But I still couldn’t close in on a particular value of k. According to silhouette method the value of k should be 2 and according to DBI the value of k should be 10
  + Later, I realized that it was because I was considering the categorical value in training the model. This was throwing off the model’s accuracy and misleading it since categorical value column when converted to numeric does not depict a continuous relationship. Scaling them introduces an artificial relationship and throws off the model. 
  + As You can see, there is still no concrete value of k where all the threee graphs intersect.
  + I decided to perform two more steps in preprocessing that earlier I did not consider
    - I analyzed the outliers in each features. This included checking the number of outliers, range of outliers to see whether I can remove some of them to improve the model performance.
    - Looking at the outliers, I concluded that applying Transformation on the Data would be best course of action. The reason for this is given further in the report.

STEP 3: Further Preprocessing

* **Analyzing Outliers** 
  + **This step is important since k-means clustering is a distance-based model. This means that outliers significantly affect the performance of the model.**
  + This is the boxplot on the raw scaled data without and transformation
  + I hid the outliers to better see the boxplot next. This is done using (showFliers=False) as argument in the boxplot code
  + Also, converted the yscale into log scale to better adjust the data. 
  + Made a violinplot to look at the distribution of data as well.
  + Finally, overlaying both the plots. 
* Strategy for Handling Outliers

Given you have 7050 data points, here's a balanced approach to consider:

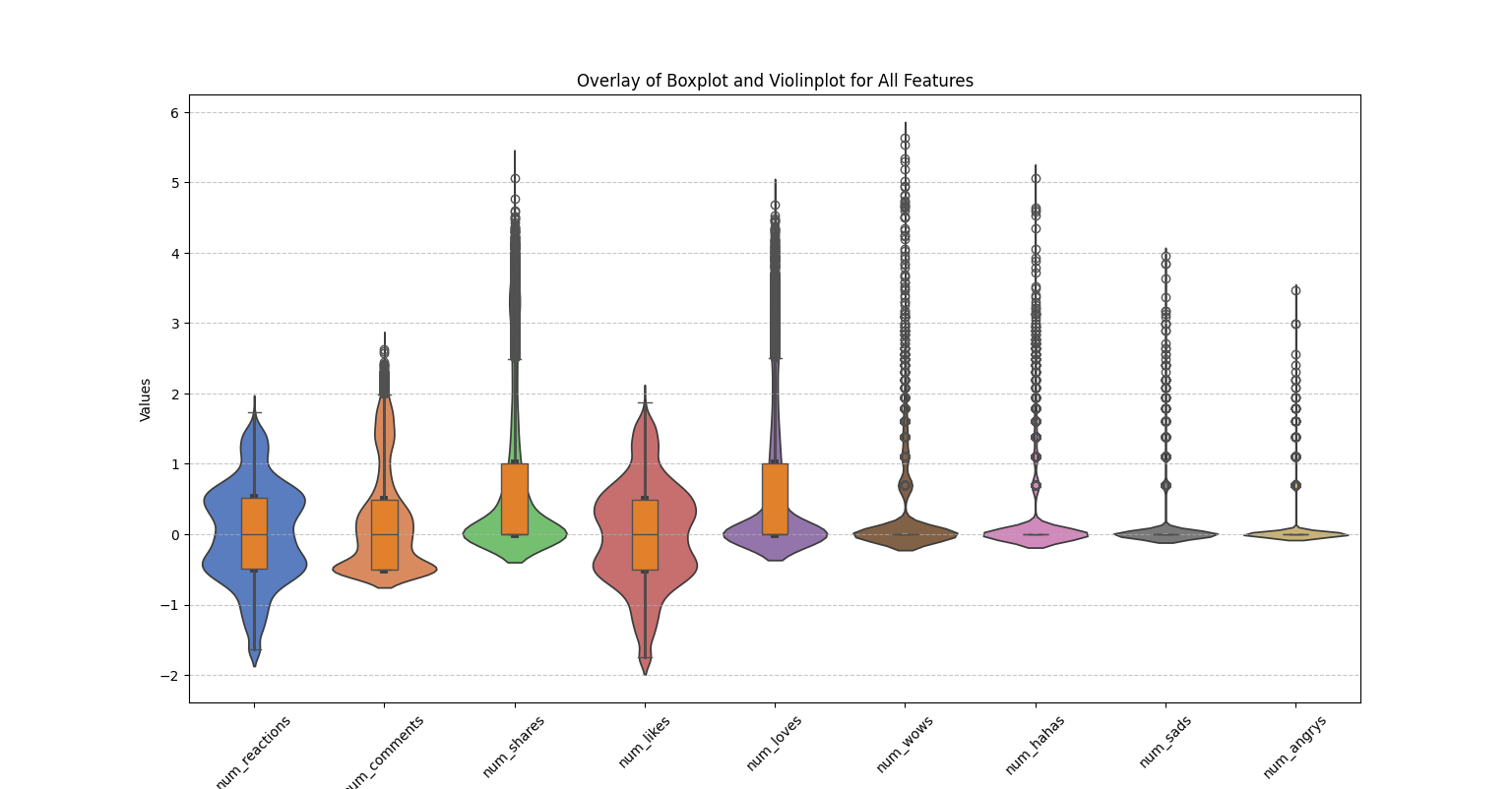
* + Quantify Outliers: The percentage of quantifiers is more than 5-10%. Therefore, removing outliers will not improve the performance of the model.

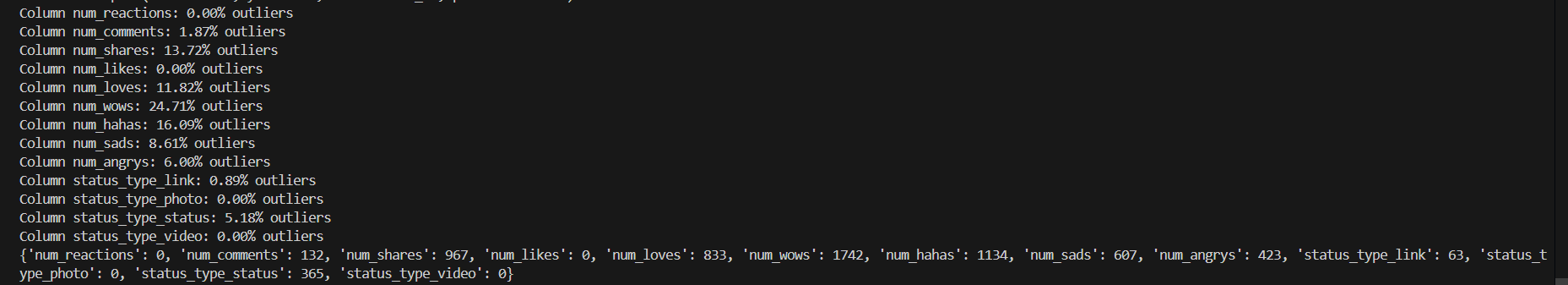


* + Decide on a Course of Action: Also, through domain knowledge we know that outliers here just mean viral posts and keeping them would be a rather good decision.

Therefore, the **best course of action would be transforming the data**, since I have already performed scaling.

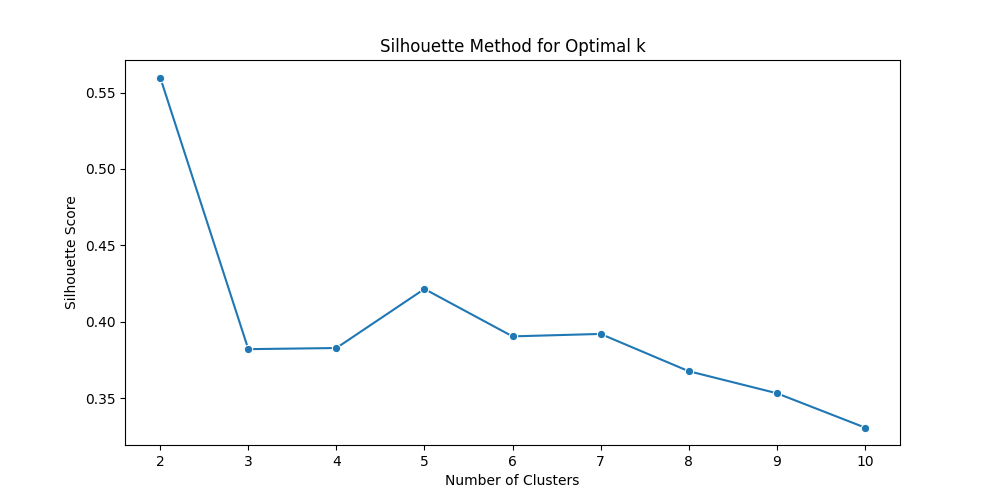
* **Transforming the data**

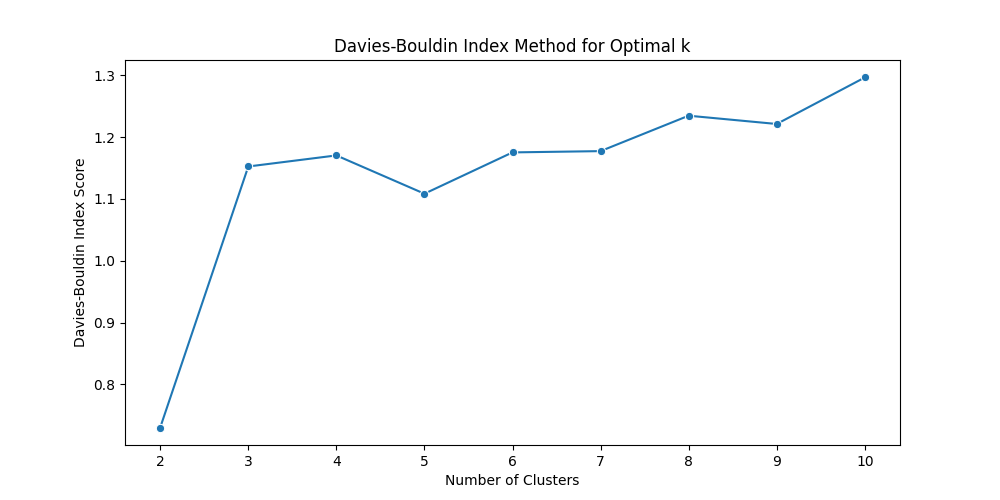


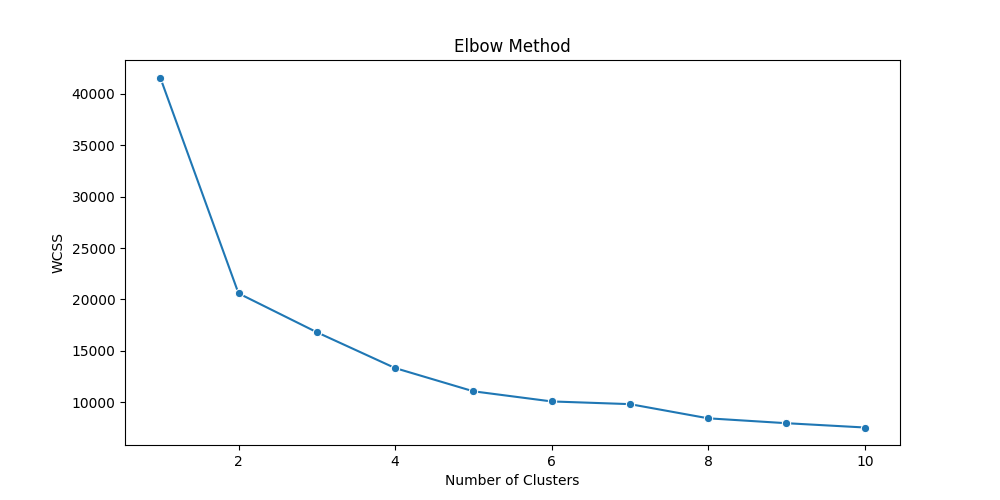


STEP 4: Finding the k value

* After the scaling, transforming and encoding of categorical values, I again used the three methods namely elbow method, silhouette method and DBI Index to find the optimal value of k
* Looking at the graph we can see a clear overlap of all the three graph for k value equal to 2.
  + Elbow is at 2
  + Silhouette score is highest at 2
  + DBI index is lowest at 2





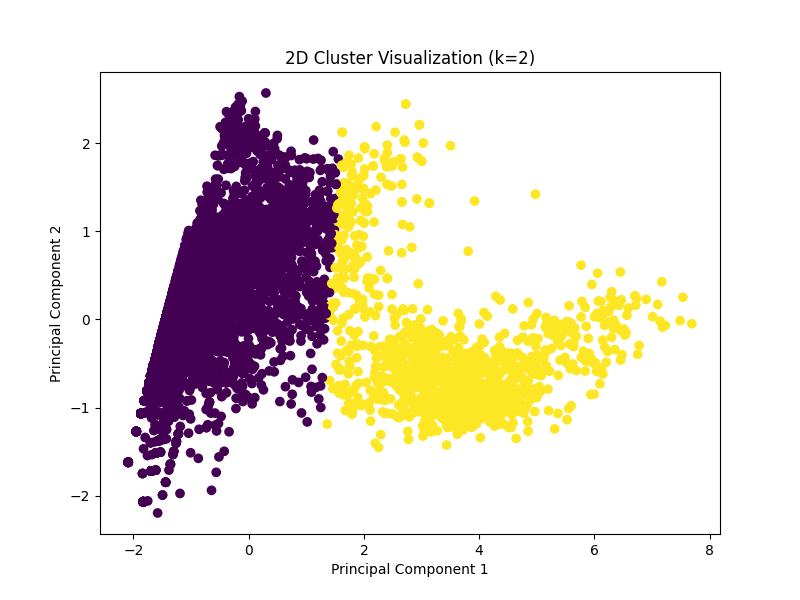


STEP 5: Training the K-Means Clustering Model on K=2

* From Scikit-learn library we will make an instance of KMeans and train the model.
* After Training,
* **Principal Component Analysis (PCA) is** a technique used to simplify complex data by reducing the number of variables while keeping most of the information.

Imagine a Big Table. You have a large dataset with many columns (features) and rows (data points). PCA identifies the directions in which the data varies the most. These directions are called principal components. The first principal component explains the most variation, the second explains the next most, and so on. By focusing on the top principal components, you can reduce the number of features without losing much information.

PCA is used to visualize high-dimensional data by reducing the dimensions to something like 2D or 3D.

* Since, my dataset has 11 features I will apply PCA before forming the scatter plot of the data.
  + IN 2D, the scatter plot looks something like
  + In 3D, the scatter plot looks like something



1. **CONCLUSION**

This report details the process of training a K-Means clustering model on the Facebook Live Sellers dataset, aiming to uncover inherent patterns within the data of Thai fashion and cosmetic retail sellers. The journey involved several critical steps, from initial data preprocessing to final model training and visualization.

**1. Key Takeaways**

* **Data Preprocessing is Paramount:** The importance of thorough data preprocessing cannot be overstated. Early challenges in finding a suitable K-value highlighted the necessity of addressing outliers and appropriately handling categorical variables. Skipping these steps initially led to significant roadblocks in model training.
* **Transformation Over Removal:** Given the nature of the dataset, particularly the potential for "viral posts" represented as outliers, data transformation proved more effective than outlier removal. This approach allowed us to retain valuable information while mitigating the influence of extreme values. Log transformation and then robust scaling
* **Interpreting Evaluation Metrics:** Achieving consensus among different evaluation metrics (Elbow Method, Silhouette Score, and Davies-Bouldin Index) is crucial for determining the optimal number of clusters. A clear agreement across these methods significantly strengthens the robustness of the chosen K-value, like K=2, that was determined after the data was properly preprocessed.

**2. Pitfalls to Avoid**

* **Treating Categorical Values as Continuous:** Directly scaling categorical variables can introduce artificial relationships and mislead the model. Ensure proper encoding techniques, such as one-hot encoding, are applied to categorical features before incorporating them into the model.
* **Ignoring Outliers:** Outliers can significantly distort distance-based models like K-Means. However, blindly removing them can lead to data loss. Consider transformation techniques to mitigate their impact while retaining valuable information.
* **Relying on a Single Evaluation Metric:** Relying solely on one metric to determine the optimal K-value can be misleading. Utilize multiple evaluation methods and look for consensus to ensure a more robust choice.

**3. Real-World Applications**

The insights derived from this K-Means clustering model can be applied in several practical scenarios:

* **Targeted Marketing Strategies:** By identifying distinct clusters of users (e.g., high-engagement vs. low-engagement), retailers can tailor marketing strategies to better meet the needs and preferences of each group. For instance, high-engagement users might receive exclusive offers or early access to new products, while low-engagement users might be targeted with personalized content designed to boost their interest.
* **Content Optimization:** Understanding which types of posts (e.g., videos, photos, text) resonate most with different clusters can inform content creation and scheduling decisions. Retailers can focus on producing content that aligns with the preferences of each segment, maximizing engagement and reach.
* **Resource Allocation:** Insights into peak activity times can help retailers optimize their staffing and advertising efforts. For example, they can increase their social media presence during hours when user activity is highest, ensuring maximum visibility and responsiveness.
* **Customer Segmentation:** By separating the user base with data clustering, more personalized and detailed marketing strategies can be created.

**4. Future Directions**

* **Exploring Alternative Clustering Algorithms:** Consider experimenting with other clustering algorithms, such as hierarchical clustering or DBSCAN, to compare results and potentially uncover different patterns in the data.
* **Incorporating More Features:** Investigate the inclusion of additional features, such as sentiment analysis of comments or text analysis of status updates, to enrich the dataset and potentially improve clustering accuracy.
* **Fine-Tuning Model Parameters:** Conduct a more detailed grid search to fine-tune the parameters of the K-Means model and PCA, potentially leading to enhanced performance.
* **Analyze time series models:** Time series data could be made to understand the number of user reactions according to different times.