



# **BAHRIA UNIVERSITY, (Karachi Campus)**

## *Department of Software Engineering*

### **Assignment 1 - Spring 2024**

---

COURSE TITLE: **Data Mining**  
Class: **BSE 6 (A)**  
Course Instructor: **Engr. Misbah Perveen**

---

COURSE CODE: **CSC-452**  
Shift: **Morning**

#### **Group members:**

Rimsha Zahid (02-131212-011)  
Areeba Kabir (02-131212-025)  
Ariba Azam (02-131212-035)

---

**Apply advanced data mining algorithms, including clustering and association rule mining, to analyze and detect patterns in bullying statement datasets, demonstrating their practical applications and effectiveness in real-world text analysis scenarios.**

### **Report**

#### **Objective:**

The objective of this assignment is to apply clustering and association rule mining techniques on a dataset related to bullying statement detection. The students will gain hands-on experience in data preprocessing, applying clustering algorithms, and extracting association rules to identify patterns in bullying statements.

#### **Dataset:**

Use a dataset containing text data with labeled instances of bullying statements. Suitable datasets include the “Cyberbullying Data” from Kaggle or any other similar dataset that contains text data and bullying labels.

#### **Dataset URL:**

- Cyberbullying Classification Dataset:  
<https://www.kaggle.com/andrewmvd/cyberbullying-classification>

## Preprocessing:

Outputs and implementation are in the attached .ipynb file

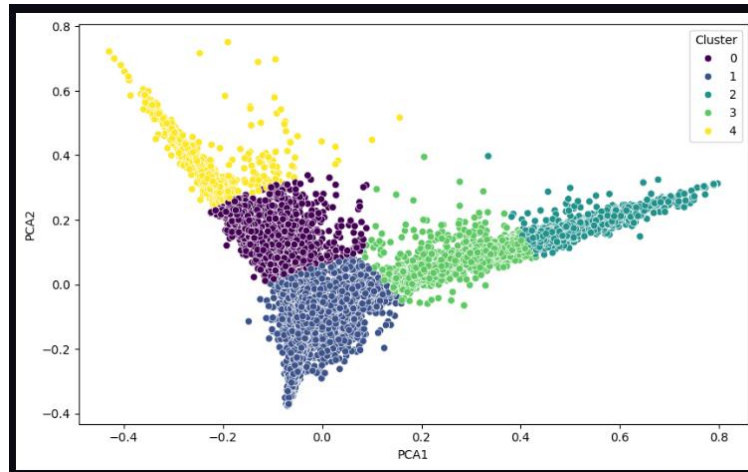
- **def reducedataset(500):**  
# This function reduces the dataset to only the first 500 rows.  
# It helps to speed up processing by working with a smaller subset of data.
- **def remove\_punct(text):**  
# This function removes punctuation from the given text.  
# It simplifies the text and helps in standardizing input for further processing.
- **def decontract(text):**  
# This function converts contractions in the text to their expanded forms.  
# For example, "don't" becomes "do not", improving clarity and processing.
- **def lower(text):**  
# This function converts all characters in the text to lowercase.  
# It ensures uniformity, making text comparison and processing easier.
- **def remove\_stopwords(text):**  
# This function removes common stop words like "the", "is", "am", "are".  
# Removing these words reduces noise and focuses on more meaningful words.
- **def smile\_handle(word\_list):**  
# This function removes emojis and special characters from the word list.  
# It helps in cleaning the text for more accurate text analysis and processing.
- **def lemmatize(words):**  
# This function converts words to their base or root form using lemmatization.  
# It reduces inflected words to a common base form, aiding in consistent text analysis.

## Clustering:

K-Means Clustering is used because of the following features.

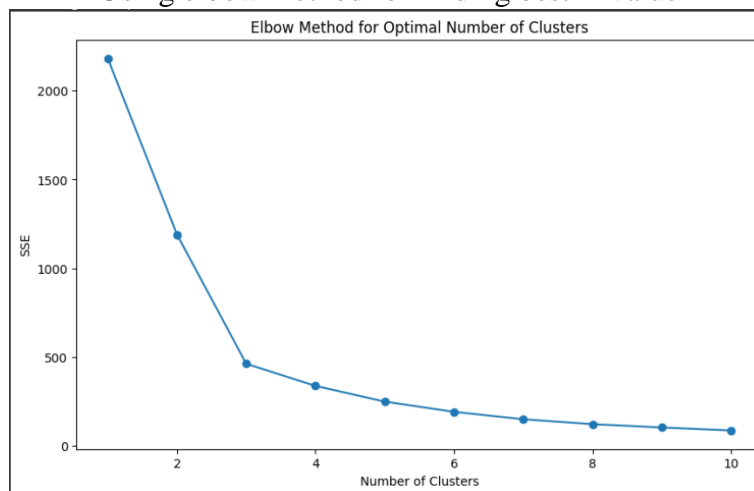
- Used for large data sets
- Efficient when we have to make equal no of clusters

Insights :



- **Cluster Separation:** The PCA-transformed space shows a good separation between the clusters, indicating that the features used for clustering are successful in differentiating the various groups.
- **Cluster Characteristics:** Each cluster has a different distribution and spread; for example, clusters 0 and 3 appear more densely packed, whereas clusters 1 and 4 are more widely distributed. This implies that each cluster has unique characteristics and levels of variability.
- **Influence of PCA Components:** The first principal component (PCA1) and second principal component (PCA2) have respective ranges of roughly -0.4 and 0.8. The major directions of variance in the data are depicted by the spread along these axes, which also highlights the most important patterns that set the clusters apart.

Using elbow method for finding best k value



### Elbow Method:

The elbow method plot you provided is used to determine the optimal number of clusters for a k-means clustering algorithm. The plot shows the Sum of Squared Errors (SSE) on the y-axis against the number of clusters on the x-axis.

- To find the optimal number of clusters, you look for the "elbow" point in the plot, where the SSE starts to decrease more slowly. This point indicates a balance between minimizing the SSE and avoiding overfitting with too many clusters.
- From the plot, the elbow appears to be around the point where the number of clusters is 3. After 3 clusters, the decrease in SSE becomes more gradual. Thus, the optimal number of clusters for this dataset is likely 3.

### Association:

```
# Step 2: Apply FP-Growth Algorithm
frequent_itemsets_fp = fpgrowth(oht_df_bool, min_support=0.01, use_colnames=True)

# Step 3: Generate Association Rules
rules = association_rules(frequent_itemsets_fp, metric='confidence', min_threshold=0.5)

# Step 4: Set thresholds and filter meaningful rules
filtered_rules = rules[(rules['lift'] > 1) & (rules['confidence'] > 0.5)]
```

### Here are predicted associations

```
print(filtered_rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']])
```

	antecedents	consequents	support	confidence	lift
0	(school, not)	(bulli)	0.017487	0.923588	4.478219
1	(not, bulli)	(school)	0.017487	0.818449	4.819545
2	(school, but)	(bulli)	0.025036	0.950637	4.609371
3	(bulli, but)	(school)	0.025036	0.863965	5.087570
4	(school, like)	(bulli)	0.034870	0.962384	4.666331
...	...	...	...	...	...
1740	(rt, obama)	(tayyoung, nigger, fuck, dumb, as)	0.010211	0.868093	46.258186
1741	(dumb, rt)	(tayyoung, nigger, fuck, obama, as)	0.010211	0.563006	30.000974
1742	(rt, as)	(tayyoung, nigger, fuck, obama, dumb)	0.010211	0.762128	40.611647
1743	(obama, as)	(tayyoung, nigger, rt, fuck, dumb)	0.010211	0.503099	49.268595
1744	(tayyoung)	(nigger, rt, fuck, obama, dumb, as)	0.010211	0.523656	50.452925

[1745 rows x 5 columns]

### Explanation:

- **Antecedents:** Items or itemsets on the left-hand side of the association rule, e.g., (school, not).
- **Consequents:** Items or itemsets on the right-hand side of the association rule, e.g., (bulli).
- **Support:** Frequency of the itemset in the dataset, calculated as the number of transactions containing the itemset divided by total transactions, e.g., 0.017487 means 1.75% of transactions.
- **Confidence:** Reliability of the rule, calculated as the number of transactions with both antecedent and consequent divided by transactions with the antecedent, e.g., 0.923588 means 92.36% reliability.
- **Lift:** Measure of how much more likely the consequent is to occur with the antecedent than without, where a value greater than 1 indicates positive correlation, e.g., 4.478219 means the antecedent increases the likelihood of the consequent by 4.48 times.

#### Results:

- High confidence and lift values indicate strong associations between the antecedents and consequents. For example, (school, not) => (bulli) with a confidence of 0.923588 and a lift of 4.478219 shows a strong association between these terms.
- The rules with antecedents containing terms like (rt, obama), (dumb, rt), and (tayyoung) have very high lift values (ranging from 30 to 50), indicating exceptionally strong associations in those contexts.

#### PKL File for saving model:

For saving the model this is converted into .pkl file.

```

▶ with open('pca_model.pkl', 'wb') as file:
    pickle.dump(pca, file)

    with open('association_rules.pkl', 'wb') as file:
        pickle.dump(filtered_rules, file)

[ ] #saving model




with open('/content/pca_model.pkl', 'rb') as file:
    pca_loaded = pickle.load(file)

with open('/content/kmeans_clustering_model.pkl', 'rb') as file:
    kmeans_loaded = pickle.load(file)

# Verify the loaded objects
print(kmeans_loaded)
print(pca_loaded)

➞ KMeans(n_clusters=5, random_state=42)
   PCA(n_components=2)

```

	association_rules.pkl Type: PKL File
	kmeans_clustering_model.pkl Type: PKL File
	pca_model.pkl Type: PKL File

### Recommendations:



- **Cluster-Based Association Rule Mining:** First, use clustering (e.g., K-Means or DBSCAN) to group similar instances. Then, apply association mining techniques (e.g., Apriori or FP-Growth) within each cluster to discover rules specific to each cluster.
- To enhance the robustness of detection models, gather a diverse and representative range of datasets from various platforms, languages, and demographics. Utilize **multimodal data**, including images and videos, to capture a broader spectrum of bullying behaviors.
- **Advanced Text Analysis Techniques:** Use of sophisticated natural language processing (NLP) techniques, such as transformer-based models like **BERT** and **GPT**, for improved contextual understanding and feature extraction. Develop methods for continuous learning to adapt to new bullying patterns and linguistic changes.
- **Contextual and Behavioral Analysis:** having more contextual information, such as users' past behaviors and engagement patterns, to enhance the accuracy of bullying detection. Enable models to understand and interpret subtle expressions, such as humor and sarcasm, which are often present in bullying language.

### Prototype:

This is a simple prototype of real time system for the clustering and association use.

It can be used for large social media platforms.

Files for prototype system:

	app Type: Python Source File
	requirements

