Lab Report: **05**

Subject: CSE - 477

Section: 1

Submitted By:

Syed Musayedul Hussain

ID: 2021-2-60-015

Submitted to:

Amit Mandal
Lecturer

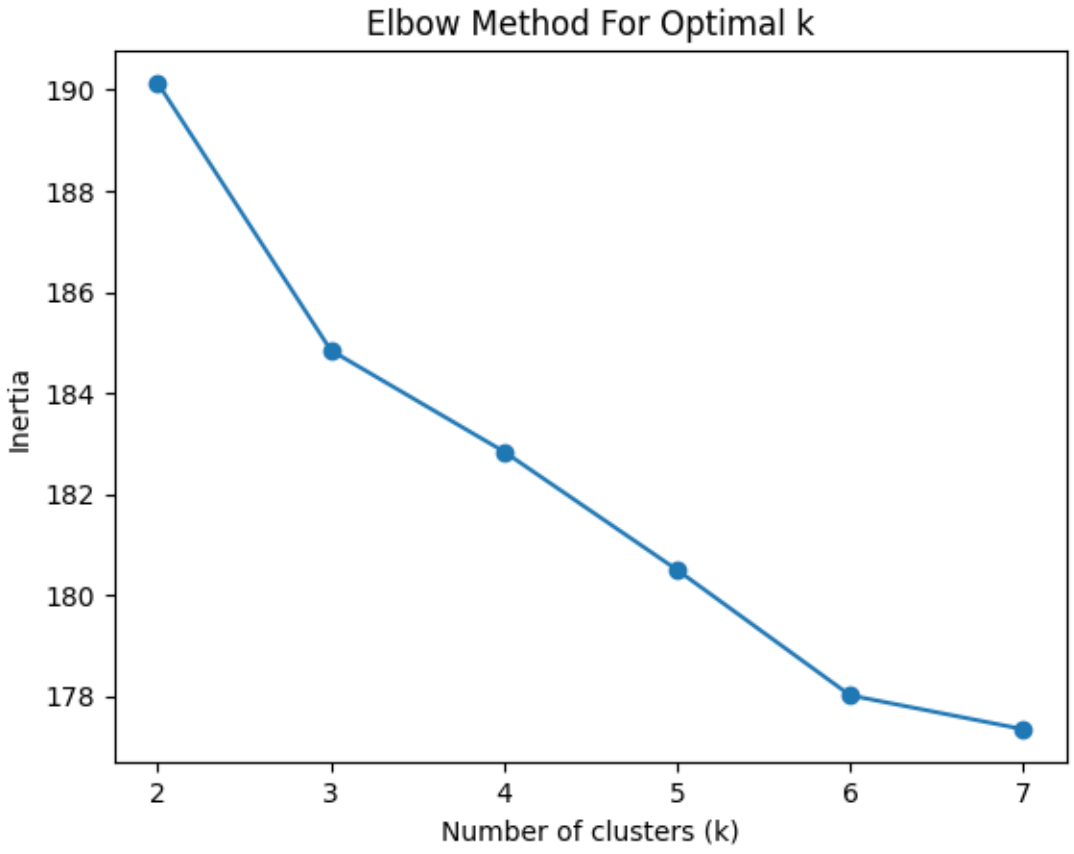
Department of Computer Science and Engineering ,East West University

In this lab, I applied two popular clustering algorithms—K-Means and DBSCAN—to my own cleaned YouTube comment dataset from Lab 2. The goal was to explore how unsupervised learning can uncover hidden patterns and groupings within real-world text data. Using TF-IDF vectorization, the comments were transformed into numerical form for analysis. K-Means was used with the Elbow Method to determine an optimal number of clusters, while DBSCAN was tuned to identify meaningful clusters and noise. By comparing the results, I was able to evaluate which method provided more insightful interpretations of my dataset and reflect on how dataset size, topic, and language influenced the outcomes.

Critical Prompt 1 — Initial Hypothesis

With 200 comments, there's enough data to potentially form some coherent clusters, but the dataset is still relatively small for robust topic modeling. While this size may produce identifiable themes, results can be sensitive to noise and less diverse compared to larger datasets.

Critical Prompt 2 — Choice of k and Evaluation

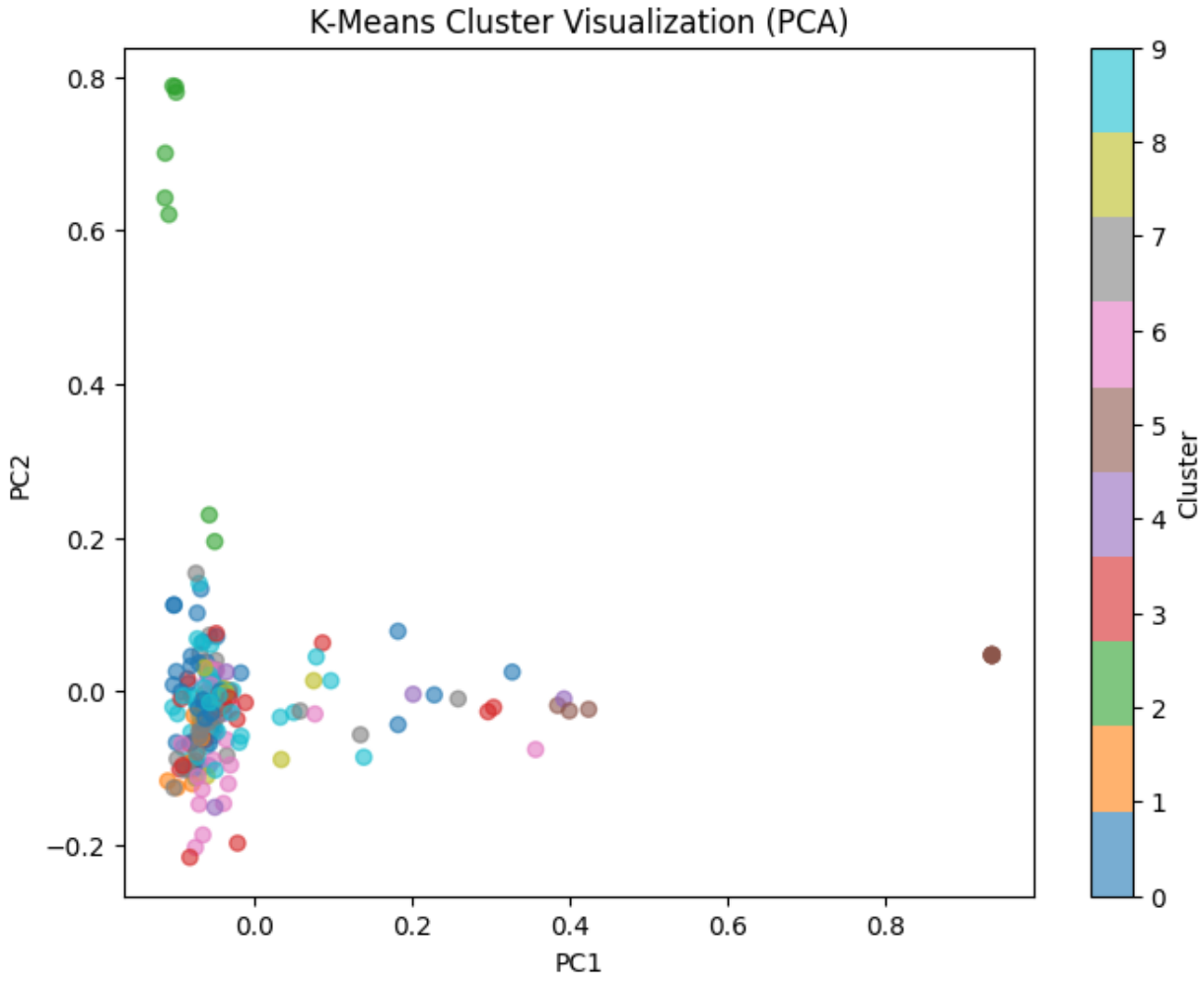


Elbow Method Observation: The elbow point was somewhat ambiguous, but a value of k=10 was selected because it provided a balance between low inertia and distinct topic separation without over-clustering.

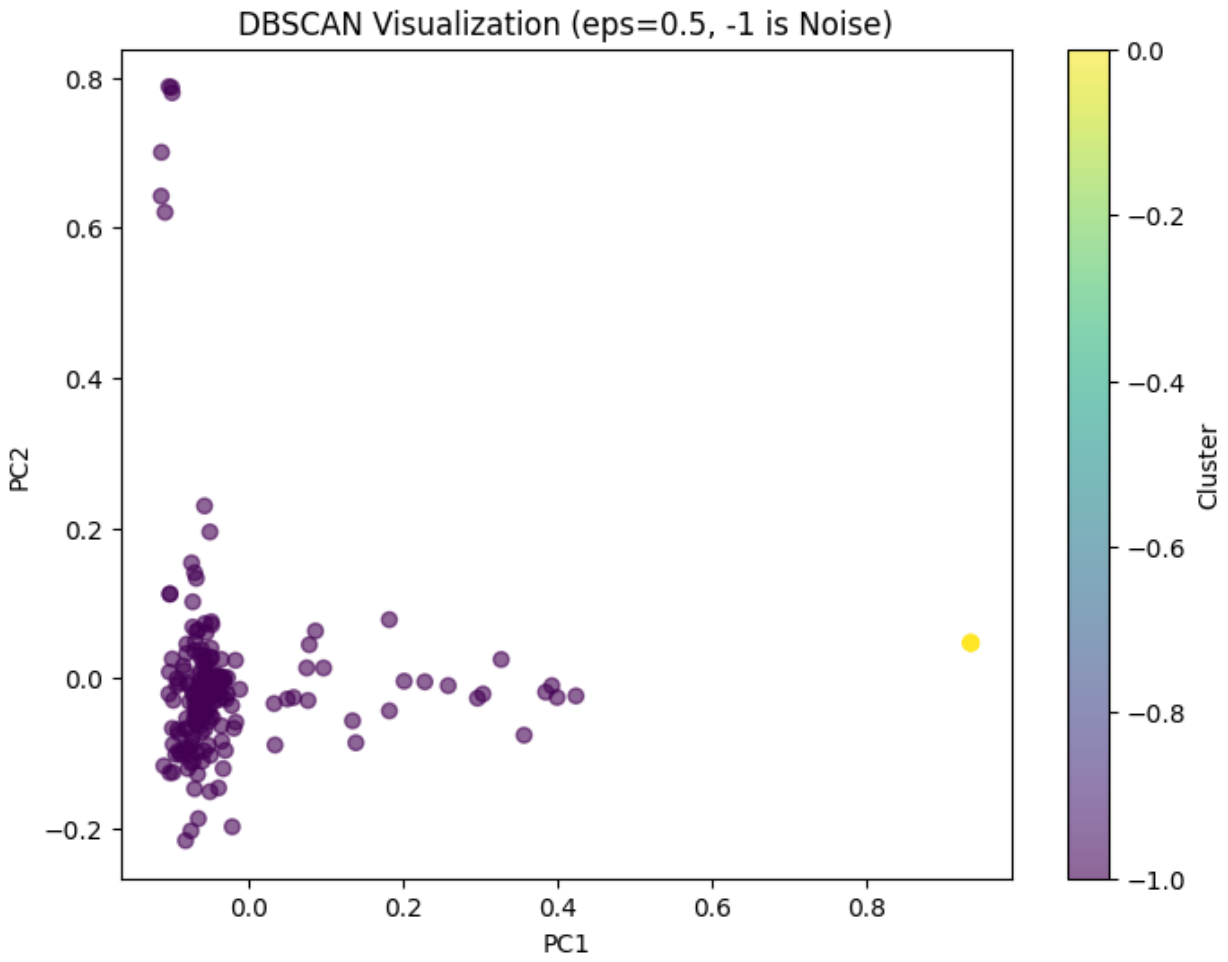
Good Cluster Example: Cluster 2 contains terms like “regression, implementation, using, linear, model, data, tensorflow, training, neural, bayes,” which clearly points to a machine learning regression tutorial theme.

Bad/Confusing Cluster Example: Cluster 7 has terms like “reply, woke, day, code, falling, lol, list, error, else, asleep,” which mixes casual chatting (“lol, asleep”) with technical terms (“code, error”), making it hard to assign a single coherent topic.

Critical Prompt 3 — K-Means vs. DBSCAN



K-Means Results: Produced 10 clusters with identifiable topics such as ML tutorials, personal gratitude, casual/funny comments, and learning experiences.



DBSCAN Results: For eps values 0.3, 0.5, and 0.7, DBSCAN produced only 1 cluster and marked 194 out of 200 points as noise — indicating it failed to detect meaningful groupings.

More Useful Algorithm: K-Means clearly provided more useful insights because it separated the comments into interpretable groups.

Scenario where DBSCAN would be better: DBSCAN excels when dealing with datasets that have non-spherical clusters or when we expect outliers/noise (e.g., detecting spam or abusive comments among mostly normal conversation).

Critical Prompt 4 — Reflection

```
Cluster 0: video, best, learning, machine, youtube, watched, lot, great, brilliant, one
Cluster 1: asleep, fell, woke, watching, playing, youtube, video, dashie, sooooo, reply
Cluster 2: regression, implementation, using, linear, model, data, tensorflow, training, neural, bayes
Cluster 3: tutorial, excellent, nice, time, much, python, thank, first, found, awesome
Cluster 4: sharing, wow, beautiful, immense, thank, without, helpful, thanks, effort, make
Cluster 5: thanks, pakistan, excellent, creating, sharing, content, amazing, kylie, course, fleshed
Cluster 6: kylie, thank, ying, really, professor, much, like, life, morocco, course
Cluster 7: reply, woke, day, code, falling, lol, list, error, else, asleep
Cluster 8: explained, nothing, perfectly, honest, yikes, thank, much, everything, lesson, dog
Cluster 9: course, good, understand, beginner, thank, learning, dont, great, know, watch
```

The dataset's size (200 comments), topic focus (YouTube tech/learning content), and mixed language style (formal tutorial feedback and casual chatting) influenced clustering quality. K-Means handled this moderately well, identifying distinct educational and conversational topics. DBSCAN struggled due to sparse high-dimensional TF-IDF vectors and the lack of dense clusters. Key Lesson for Businesses: When applying clustering to customer feedback, algorithm choice must match data characteristics—structured, spherical-topic data suits K-Means, while noisy, irregularly shaped data benefits from DBSCAN.

Final Summary

Overall, K-Means outperformed DBSCAN for this dataset, producing multiple meaningful clusters and interpretable keyword themes. DBSCAN's inability to form clusters under reasonable parameters suggests that density-based methods are less effective on small, sparse, high-dimensional comment datasets. K-Means remains the better choice here for uncovering patterns in structured feedback-like YouTube comments.