Logo

Description automatically generated **EAST WEST UNIVERSITY**

**CSE477**

**Section: 02**

**Lab: 05 Report**

**Topic: Clustering YouTube Comments — A Case Study Approach**

**Submitted By:**

**Name: Md Sifatullah Sheikh**

**ID: 2022-1-60-029**

**Submitted To:**

**Amit Mandal**

**Lecturer**

**Department of Computer Science & Engineering**

**Date: 9 August 2025**

# Abstract

This report presents an end-to-end unsupervised analysis of YouTube comments using TF–IDF features with two clustering algorithms: K-Means and DBSCAN. I evaluate cluster quality via elbow and internal metrics, visualize cluster structure, inspect top terms and representative comments per cluster, and compare the two methods. The study concludes with reflections on dataset characteristics and practical takeaways for real-world feedback analysis.

# 1. Introduction

The objective of this lab is to discover meaningful discussion themes in YouTube comments without relying on labels. K-Means partitions the data into a fixed number of clusters (k), while DBSCAN groups points by density and identifies noise. These complementary views help assess both dominant topics and outliers such as spam or off-topic remarks.

# 2. Data and Environment Setup

• Data file: /content/drive/MyDrive/CSE477/cleaned\_comments.csv

• Expected columns: cleaned\_tokens (list-like) or cleaned\_text (string).

• Notebook environment: Google Colab (Drive mounted).

**Insert Screenshot 1 — Proof of Drive mounted and file path exists** (e.g., output of the setup cell showing the path and a true assertion).

Code installs (pandas, numpy, matplotlib, seaborn, scikit-learn).

# 3. Step 1 — Data Validation and Initial Hypothesis

I loaded the dataset and reported its shape. A quick preview verified the presence of the expected text column. Dataset size (number of comments) impacts cluster stability and interpretability.

**Insert Screenshot 2 — Data shape and head() preview** (first 5 rows).

Critical Prompt #1 (1–2 sentences):

Based on the number of comments, my initial hypothesis about cluster quality is: [Write your hypothesis here. Smaller datasets can run but may yield less stable, less generalizable topics.]

# 4. Step 2 — Text to Vector (TF–IDF)

I constructed a text corpus from cleaned\_tokens or cleaned\_text and computed a TF–IDF matrix with unigrams and bigrams. This representation captures important words and short phrases while limiting feature size.

**Insert Screenshot 3 — TF–IDF matrix shape and Top TF–IDF Terms plot** (bar chart of global top terms).

# 5. Step 3 — K-Means Clustering and Interpretation

## 5.1 Elbow and Internal Metrics

To choose k, I plotted inertia (elbow method) and computed internal validation metrics: Silhouette (higher is better), Calinski–Harabasz (higher is better), and Davies–Bouldin (lower is better). When the elbow is ambiguous, these metrics help triangulate a sensible k.

**Insert Screenshot 4 — Elbow plot (inertia vs k)** and **Insert Screenshot 5 — Metrics table/plots (Silhouette, CH, DB vs k).**

## 5.2 Final k, Clusters, and Top Terms

I selected k = [fill value] based on the elbow and supporting metrics. After fitting K-Means, I extracted the top TF–IDF terms for each cluster and reviewed representative comments.

**Insert Screenshot 6 — Top terms per cluster (bar charts)** for all clusters.

**Insert Screenshot 7 — Example representative comments per cluster** (the printed examples).

## 5.3 2D Visualization

To visualize structure, I projected TF–IDF to 2D (TruncatedSVD). The scatter plot shows how comments group under K-Means labels.

**Insert Screenshot 8 — K-Means 2D scatter plot** (SVD 2D).

Critical Prompt #2:

• Chosen k and justification: [Explain with reference to the elbow and metrics].  
• One ‘good’ cluster: [Describe why the top terms and sample comments form a clear, cohesive theme].  
• One ‘confusing’ cluster: [Explain overlaps, generic tokens, or mixed topics].

# 6. Step 4 — DBSCAN (Comparative Analysis)

## 6.1 Scaling and Parameter Tuning

I scaled the sparse TF–IDF matrix using StandardScaler(with\_mean=False) and swept multiple eps values with min\_samples=5. I recorded the number of clusters, noise points, and (when applicable) the silhouette score to pick a reasonable eps.

**Insert Screenshot 9 — DBSCAN tuning results table** (eps, clusters, noise, silhouette).

## 6.2 Final DBSCAN Result and Visualization

With the chosen eps, I re-ran DBSCAN and visualized cluster assignments on the same 2D projection used for K-Means. Label -1 indicates noise points.

**Insert Screenshot 10 — DBSCAN 2D scatter plot** (with -1 as noise).

Cluster size comparison:  
• K-Means counts: [paste printed counts]  
• DBSCAN counts: [paste printed counts]

Critical Prompt #3:

Compare K-Means versus DBSCAN:  
• Which provided more useful insight for this dataset and why?  
• Scenario where DBSCAN is superior: [e.g., irregular cluster shapes, presence of many outliers/spam].

# 7. Final Reflection and Application

I reflect on how dataset characteristics (size, topic, language style, code-mixed Bangla/English, slang, emojis) influenced the results. I identify limitations (e.g., small sample size, short comments, sarcasm) and suggest directions (domain-specific stopwords, phrase modeling).

# 8. Exported Artifacts

I exported a CSV containing the original text and cluster labels for both K-Means and DBSCAN. This artifact supports downstream analysis (e.g., manual labeling, topic summaries).

**Insert Screenshot 11 — Export confirmation** (path and filename).

# 9. Optional: UMAP 2D Visualization

If UMAP was available, I would have generated an alternative 2D projection that sometimes separates clusters more cleanly. This is presented as a supplementary view.

**Insert Screenshot 12 — UMAP 2D colored by K-Means**

# Appendix A — Reproducibility Notes

• Random seeds fixed where applicable (random\_state=42). • Ensure the same TF–IDF parameters and data path when re-running. • If columns differ from expectations, adjust corpus construction accordingly.