Comparison and Discussion:

Your Name	Dataset used	Preprocessing Steps Applied	Text Representation	Analysis Method	Number of clusters/topic components	Best Number of Clusters/Topics
SRINATHI K	SMS Spam Collection Dataset	Tokenization Lowercasing Removal of Punctuation Removal of Stopwords Lemmatization	BoW	Clustering 1.KMeans 2.Agglomerative 3.Gaussian Mixture Model (GMM) 4.Spectral Clustering 5.DBSCAN	n_clusters = 3 n_clusters = 5 n_clusters = 7	Best Number of clustering: 5
			Tf Idf + Embedding model 1 (MiniLM-L6)			
		-	Embedding model 2 (MiniLM-L12)	Topic Modeling Latent Dirichlet Allocation (LDA)	n_topics = 5 n_topics = 7 n_topics = 10	Best Number of Topics: 7
			Embedding model 3 (DistilRoBERTa)			

Observation/Intuition Questions:

Impact of Preprocessing: For the dataset you worked on, how did the preprocessing steps (like stopword removal or lemmatization) seem to affect the raw text, and do you think these changes would make it "easier" or "harder" for a human to understand the text, versus a computer?

It is easy for humans to understand these strangers because Preprocessing had a pretty noticeable impact on the text. For example, removing stopwords like “the”, “is”, “at” cleaned out a lot of noise that doesn't add much meaning for machines. Lemmatization also helped by converting words like “running” and “ran” into their base form “run”, so similar ideas weren’t split up by minor differences in wording.

TF-IDF vs. Sentence Embeddings (Similarity/Clustering): When comparing documents using TF-IDF versus any of the Sentence Embedding models, did you notice a qualitative difference in *why* documents were considered "similar"? Which approach seemed to capture semantic meaning more effectively, and can you give a simple example from your results?

I definitely noticed a difference in how similarity was judged between TF-IDF and sentence embeddings. With TF-IDF, similarity depends heavily on exact word overlap—so two messages like “Call now to claim your prize” and “You won a free gift, claim it!” may not be seen as similar if the exact words differ.

On the other hand, when I used sentence embeddings like all-MiniLM-L6-v2, the model did a better job at capturing the semantic meaning of messages. These embeddings turned entire sentences into vectors based on meaning—not just words.

Comparing Embedding Models: You used at least three different Sentence Embedding models. Did the clusters or the similarity results produced by these different models look the same, or were there subtle (or even obvious) differences? What might account for any differences you observed between them?

When I compared clustering results across different sentence embedding models—using algorithms like K-Means, Agglomerative Clustering, and DBSCAN—I noticed that the clusters were quite different and more distinct from each other. These embeddings captured deeper semantic relationships between texts, leading to more nuanced groupings. On the other

hand, when I used Bag of Words (BoW) and TF-IDF representations with K-Means, the clusters appeared more similar to each other. This suggests that BoW and TF-IDF, being more surface-level and frequency-based, tend to capture similar patterns, whereas sentence embeddings provide richer context, resulting in more varied and informative clusters.

Challenges in Interpretation: What was the most challenging part of interpreting your clusters or topics? Were there any clusters/topics that seemed unclear or didn't make intuitive sense, and what might be the reason for that?

The toughest part was interpreting clusters that didn’t have a clear or consistent theme. In some cases, a cluster contained a mix of seemingly unrelated messages—like different types of spam SMS—and it wasn’t immediately obvious what connected them. This ambiguity made it challenging to assign meaningful labels or draw clear insights from those clusters.

Real-world Value: Imagine you're explaining your findings to someone who isn't an NLP expert. How would you briefly describe one valuable insight you gained from your experiment (either from clustering or topic modelling) for your chosen dataset, and how could it be used in a real-world application?

I’d begin with a real-world example that directly relates to the findings I obtained. This helps ground the audience in a context they can easily understand. Once they grasp the practical scenario, I’d present a high-level overview of what I’m trying to convey—essentially outlining the key findings and why they matter. After that, I’d walk them through the details of the results, connecting each finding back to the real-world example to ensure clarity and relevance throughout.