Data Science

**Task 4: Critical Analysis**

**1. TF-IDF vs One-Hot Encoding – How Did They Perform?**

In this project, I explored two different ways to represent textual features of diseases: **TF-IDF** and **One-Hot Encoding**. These encodings were used to train and evaluate both **K-Nearest Neighbors (KNN)** and **Logistic Regression** classifiers.

From the results, I observed some clear patterns:

* **TF-IDF performed better with Logistic Regression**, likely because it provides a weighted view of feature importance.
* **One-hot encoding was more effective for KNN**, especially when using **Manhattan** or **Cosine** distances, probably because KNN benefits from the binary and consistent nature of one-hot vectors.

**Why does this happen?**

TF-IDF assigns higher weights to rare terms, which helps in distinguishing between diseases that share common symptoms but differ on key, less frequent indicators. On the other hand, one-hot encoding treats every feature equally — this may not carry much semantic weight but works well for models that rely on exact feature matching, like KNN.

**2. Clinical Relevance – Do These Patterns Make Sense in the Real World?**

To better understand how these encodings behave, I visualized the feature space using **PCA and SVD**.

What stood out was that **TF-IDF** generated clusters that aligned with real-world disease categories. For instance, diseases like *Stroke*, *Alzheimer’s*, and *Parkinson’s* grouped together, reflecting the **neurological category**. This suggests that TF-IDF can capture meaningful clinical patterns that go beyond simple keyword matching.

This could be incredibly useful for building medical decision support systems — the ability to semantically group diseases based on symptoms and risk factors could improve early diagnosis or case prioritization.

**3. Limitations – No Approach is Perfect**

**TF-IDF:**

* Doesn’t understand the *order* or *context* of words. So "chest pain" and "pain in chest" are treated as different.
* It's sensitive to inconsistent vocabulary — for example, “high blood pressure” vs. “hypertension”.
* It may also give too much weight to rare but clinically irrelevant words.

**One-Hot Encoding:**

* It’s extremely sparse and doesn’t reflect feature importance.
* Every symptom or sign contributes equally — which isn’t realistic in medical diagnosis.
* It ignores frequency completely — whether a symptom appears once or five times doesn’t matter to the model.

**Final Thoughts**

There’s no single best encoding — it depends on the model and the task.

* **Use TF-IDF** if you care about the deeper, semantic relationships in textual data — it works well for models like Logistic Regression.
* **Use One-Hot Encoding** when you need a straightforward representation — it's well-suited for models like KNN that rely on direct feature comparison.