Reflective Journal

This lab came with a lot of challenges, especially when it came to loading the data and making sure it was in the right format. At first, I ran into a FileNotFoundError, and then a ParserError, which showed me how important it is to double-check file paths and understand the structure of the raw data. Problems like inconsistent delimiters or messed-up rows made it even harder. Later, I kept getting a KeyError: 'text' after splitting and renaming columns, which reminded me to always carefully check DataFrame column names and make sure the changes actually carry over between notebook cells. On top of that, I ran into a LinAlgError during visualization and saw nan values in the analysis, which meant the data I was feeding into those functions wasn't right. These problems came from the way the sample was made and from the incomplete analysis functions. Overall, the experience showed me how even small mistakes early on can cause much bigger issues later in the pipeline.

One of the biggest things I learned is that data preprocessing is not just a "setup" step—it's one of the most important parts of data science and machine learning, especially with natural language processing. If the input data isn't clean and reliable, the models that use it won't perform well. I realized that data loading and cleaning are the foundation of everything, because even small errors here can stop the whole process. Cleaning text, like changing everything to lowercase, removing special characters, and fixing whitespace, is also super important to reduce noise. Tokenization and normalization, like splitting text into tokens and lemmatizing, are key transformations that prepare text for analysis. Even though my analysis wasn't fully finished, I saw the difference between static embeddings like GloVe and contextual embeddings like BERT, which is huge for choosing the right approach. I also learned that preprocessing is usually an iterative process—you often have to go back and adjust earlier steps once you see problems in later stages.

These same preprocessing techniques are also really important for generative AI. Models like large language models depend on massive amounts of text data, and preprocessing makes sure that data is clean, consistent, and ready to train on. This involves tokenization, handling special tokens like [CLS] and [SEP], and breaking text into smaller sequences that the model can actually process. Contextual embeddings, like the ones from BERT, are essential because they help the model understand the meaning of words in different situations, which makes the generated text sound natural and relevant. Tokenization also decides the model's vocabulary and gives initial embeddings that the model improves as it trains. The way you preprocess data also changes depending on the type of text—like code, creative writing, or dialogue—so the model can learn the right patterns. Finally, there's the issue of bias. Since both datasets and embeddings can contain biases, preprocessing and debiasing techniques are necessary to make sure generative AI doesn't produce unfair or harmful outputs.