This assignment was very fun. We implemented 6 embedding techniques that we learned in class and got to see how each one performs.

The Bag of Words embedding did moderately okay with an accuracy ranging from 0.5915 to 0.5930. The bag of words embedding did a good job at capturing the occurrence of words, but it ignores the context, which results in the embedding only performing well on simple sentences. Since the Bag of Words embedding doesn't account for context, word order, or semantics, it treats sentences like 'feel bad" and "bad feel" as identical, which hinders the classification. In addition, as the vocabulary grows the feature space becomes sparse which reduces the performance of the embedding when working with more complex or larger datasets. For example, the embedding classified the following sentences, "im feeling rather rotten im ambitious right"(True: 0, Predicted: 1), and "feel beautifully emotional knowing women knew handful holding baba journey" (True: 0, Predicted: 1), thus showing the embedding not being able to capture the sentiment and miss classify complex sentences.

The TF-IDF embedding performed slightly better than the Bag of Words. Though the accuracy range is from 0.5730 to 0.5990, the embedding performed better at capturing significance. By weighing the importance of rare words more heavily, TF-IDF is better at capturing the significance of specific terms across documents. It also helps to reduce the influence of stop words, which can improve performance when the dataset is rich in vocabulary. Like the bag of words embedding, tf-idf doesn't' consider semantic relationships between words which results on missing contextual meanings and the term-frequency matrix can become sparse, especially with a. lard corpus. Both the TF-IDF and Bag of Words embeddings miss classified similar sentences like "feel particularly agitated" (True: 4, Predicted: 3) and "feel like role would tear sails pessimism discontent"(True: 0, Predicted: 1). The TF-IDF embedding performed better than the Bag of Words embedding but still struggled with similar sentences and issues.

The Glove embedding underperformed to BoW and TF-IDF embeddings, with the accuracies ranging from 0.5140 to 0.5425. While the Glove embedding captures semantic relationships, it still doesn't completely capture the context such as negativity or sarcasm. And the embedding struggles with sentences like "im feeling good" and "im feeling bad" as it could still be challenging to differentiate without more context. The embedding often misclassifies emotional or contextual cues, for example, "feel particularly agitated"(True: 4, Predicted: 3) and "feel like role would tear sails pessimism discontent" (True: 0, Predicted: 3). The Glove embedding is useful for understanding semantics and would perform better with some more fine-tuning.

The Universal Sentence Encoder embedding proves a significant improvement in performance compared to earlier embeddings, with the accuracy ranging from 0.6115 to 0.6630. The USE model captures the overall meaning of the sentence, considering word order and semantic context. The USE embedding is pre-trained on large datasets and is designed to handle sentence-level representations, making them ideal for capturing the meaning of more complex text. Though USE can capture deeper semantic meanings, the size of the embedding can be large which means it may need more hardware capacity like memory and computational power to perform better. The USE embedding does a good job for many general tasks but it's not able to fully capture domain-specific knowledge unless fine-tuned on task-specific data. The embedding still misclassified phrases like "feel particularly agitated" (True: 4, Predicted: 3) and "feel like role would tear sails pessimism discontent" (True: 0, Predicted: 3).

The Word2Vec embedding outperformed all the embeddings with the accuracies ranging from 0.5800 to 0.7180. The Word2Vec embedding better captured the word relationships in context, especially for phrases. It is better suited for capturing complex relationships between words, and improving classification performance compared to Glove. Although it performs better than Glove, it can still struggle with complex sentences that contain languages like sarcasm, negations, and idiomatic expressions. For example, "felt anger end telephone call" (True: 3, Predicted: 0), and "cant walk shop anywhere feel comfortable"(True: 4, Predicted: 0). The Word2Vec embedding does an amazing job but also shows that there are still ways to go.

In conclusion, if I were picking a model purely from the performance I saw today, I would pick the Word2Vec embedding as it had the highest accuracy. If I were to go back and be allowed to fine-tune the embeddings better, then I would pick the USE embedding can capture deeper semantic meaning and I would increase the hardware allocated to allow the embedding to have more memory and computational power.