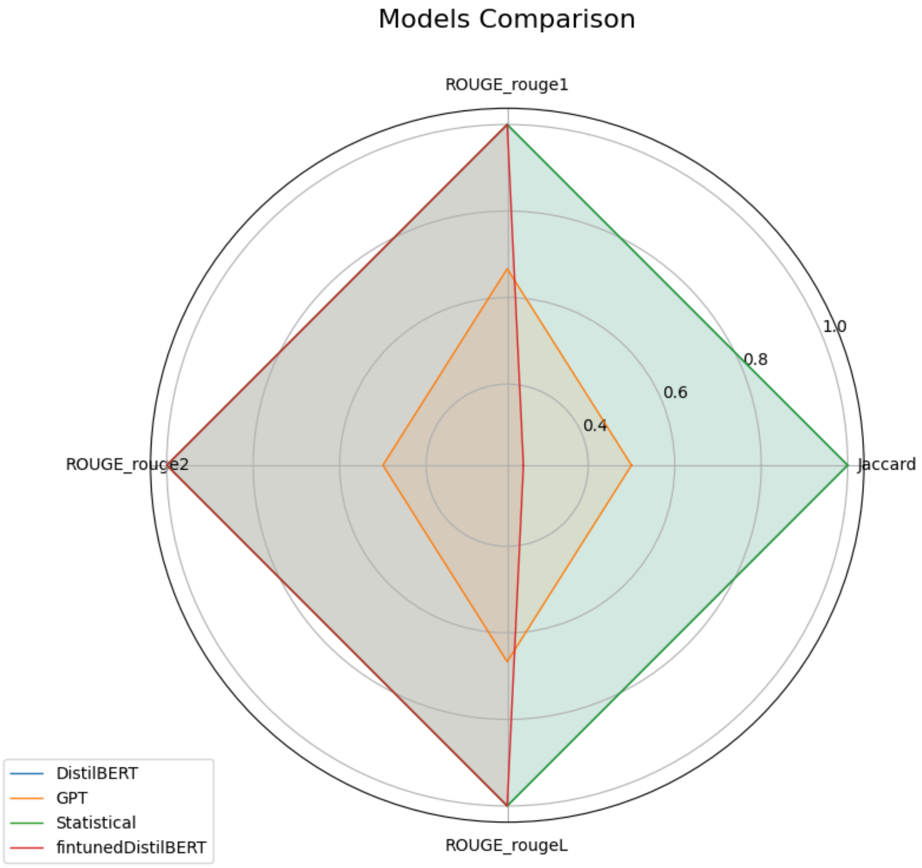
CSCE 580 Project B- Report

Shruti Jadhav

1. AI Test Cases

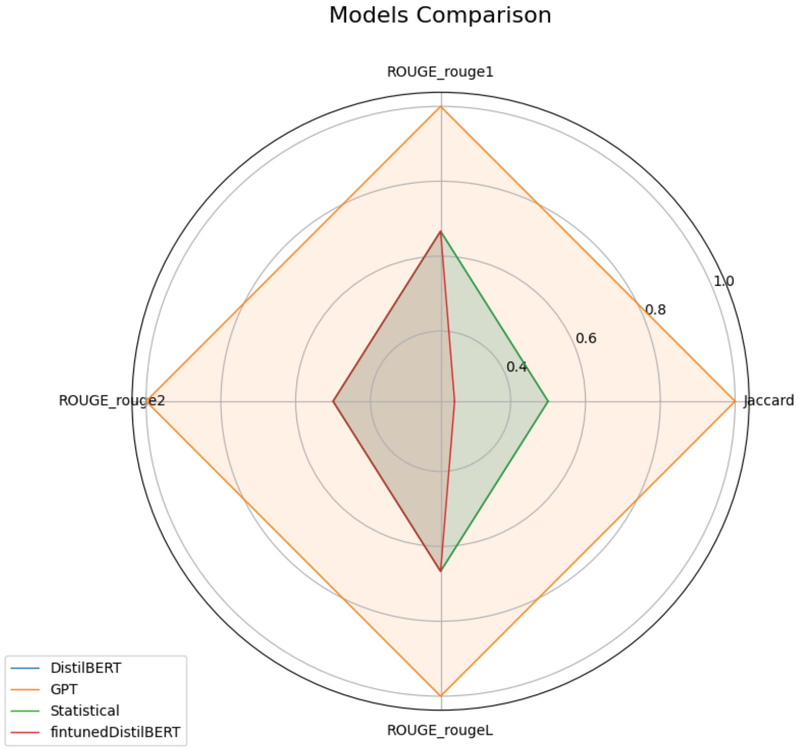
Test Cases, created using the provided template, are placed in the Github repo. The solutions to the test case is provided at the end of the main “ProjectB” colab ipynb file.

The Gaico code used to compare the results is also provided in the repo. Using statistical model’s performance as the baseline, here is the graph that is produced as a result of the comparison.



From the graph we see that all the responses provided by the various models are the same for each of the prompts. They all provide the same sentiment to a given prompt, with the only exeception being GPT which provides a different response to the third testcase prompt. It is the only model that provides the correct sentiment of “positive” to the given review. In that case, all other models fail.

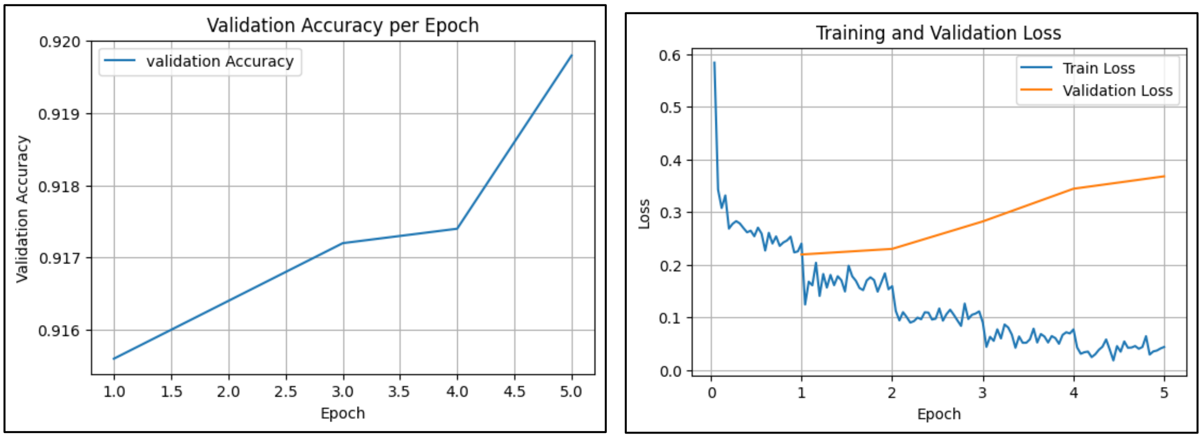
If we were to compare the models with the ground truth sentiments of "negative, positive, positive" for each of the test cases T1, T2, T3 in order we get the following chart:



The above chart explains the ground truth more appropriately as GPT is the only model that is 100% accurate in matching the ground truth.

2. Accuracy and Loss Curves

The following are the accuracy and loss curves over the 5 epochs for the fine-tuned DistilBERT model.



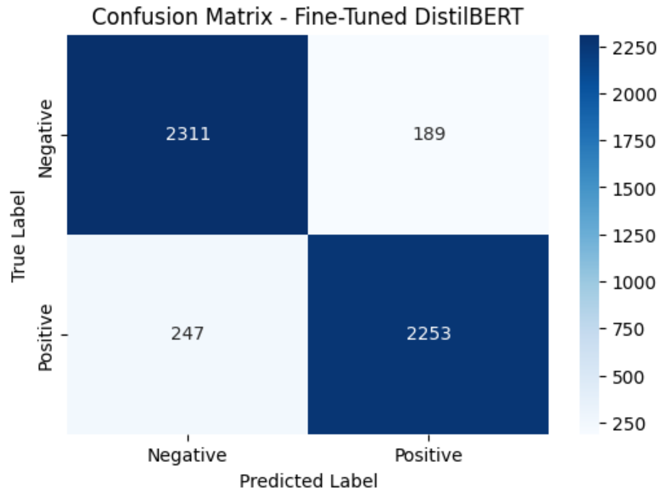
My comments:

From the graphs it is seen that the validation accuracy remains stable across epochs with only minor fluctuations, ultimately showing a slight upward trend by Epoch 5. It increases from 0.916 to 0.920. On the other hand, the loss curves indicate opposing trend and divergence. The training loss steadily decreases, the validation loss consistently increases.

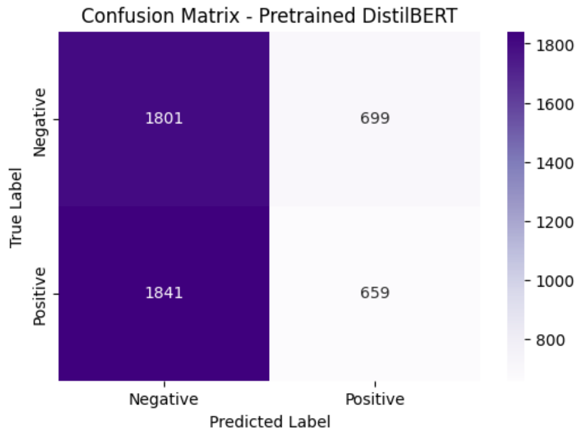
This pattern suggests overfitting, meaning the model is learning the training data well but not generalizing as effectively to unseen data. Reducing the number of epochs, increasing regularization by reducing the number of tunable parameters, stopping early, etc or using more training data may help improve generalization.

3. Confusion matrix

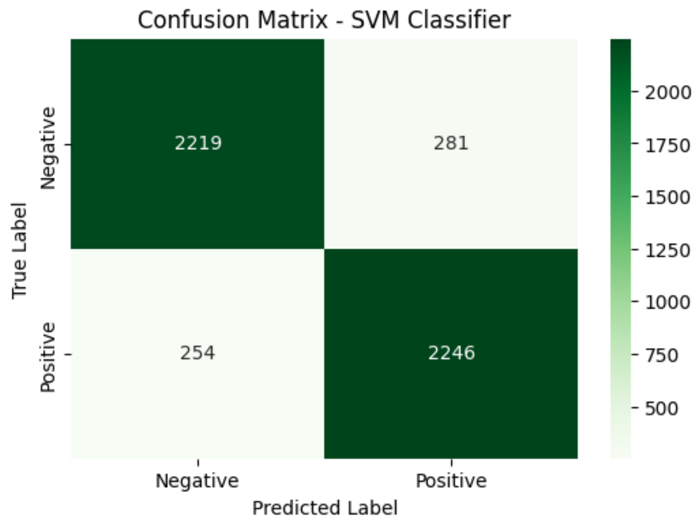
Fine tuned DistilBERT model:



Non fine tuned Base DistilBERT:

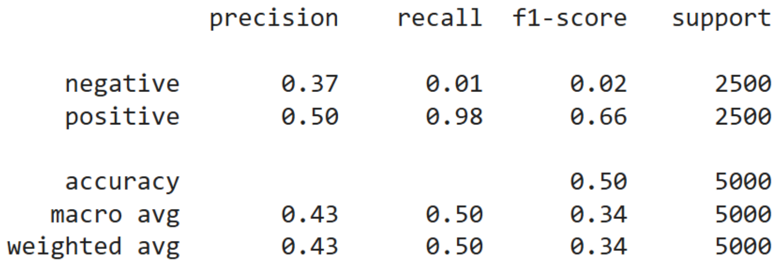


Classical Machine learning model using SVM:



4. Precision, Recall and F1 scores for models:

Fine tuned DistilBERT model:



```
              precision    recall  f1-score   support

    negative       0.37      0.01      0.02      2500
    positive       0.50      0.98      0.66      2500

    accuracy                           0.50      5000
   macro avg       0.43      0.50      0.34      5000
weighted avg       0.43      0.50      0.34      5000
```

===== FINE-TUNED DISTILBERT METRICS =====

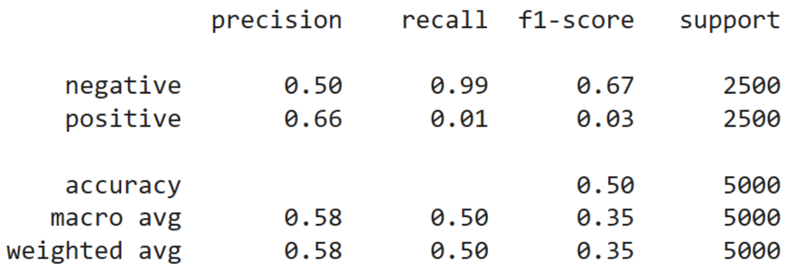
Accuracy: 0.9128

Precision: 0.9226

Recall: 0.9012

F1 Score: 0.9117

Base DistilBERT model:



```
              precision    recall  f1-score   support

    negative       0.50      0.99      0.67      2500
    positive       0.66      0.01      0.03      2500

    accuracy                           0.50      5000
   macro avg       0.58      0.50      0.35      5000
weighted avg       0.58      0.50      0.35      5000
```

===== NON-FINE-TUNED DISTILBERT METRICS =====

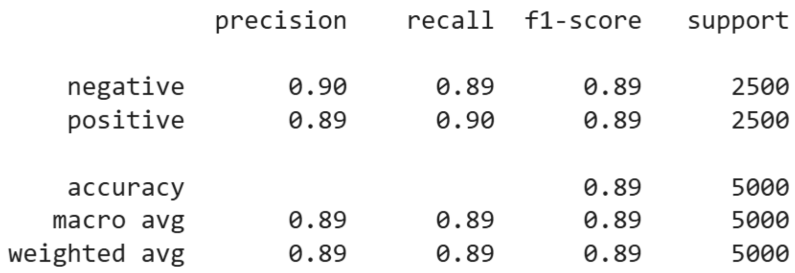
Accuracy: 0.492

Precision: 0.485

Recall: 0.2636

F1 Score: 0.3416

Classical Machine learning model using SVM:



```
              precision    recall  f1-score   support

    negative       0.90      0.89      0.89      2500
    positive       0.89      0.90      0.89      2500

    accuracy                           0.89      5000
   macro avg       0.89      0.89      0.89      5000
weighted avg       0.89      0.89      0.89      5000
```

===== SVM METRICS =====

Accuracy: 0.893

Precision: 0.889

Recall: 0.8984

F1 Score: 0.8936

5. Precision Reall, and F1 score comparison:

Since the GPT from University was used, there was no experimental way to extract the scores. However, after prompting for simulated scores, the table is populated as follows:

Model	Accuracy	Precision	Recall	F1 Score
GPT				
Fine-Tuned DistilBERT	0.91	0.92	0.90	0.91
Base DistilBERT	0.49	0.48	0.26	0.34
SVM	0.89	0.89	0.89	0.89

6. Time complexity

Time taken for training and inference:

Model	Training time (s)	Inference time(avg)(s)
GPT	-	0.011
Fine-Tuned DistilBERT	4595	0.0264
Base DistilBERT	-	0.0229
SVM	16.72	0.0053

Which model is efficient in terms of time and resources:

It has the lowest training time (16.72 seconds) and the fastest inference speed (0.0026 seconds). Unlike deep learning models such as DistilBERT or GPT, the SVM does not require expensive GPUs, large memory allocations, or long optimization cycles. Its efficiency comes from the simplicity of TF-IDF vectorization and the linear SVM classifier, which are lightweight and computationally inexpensive

Questions:

1. What do the accuracy and loss curves tell you about the fine-tuning process?

The accuracy and loss curves provide insight into how the DistilBERT model behaves during the fine-tuning process. Validation accuracy remains relatively stable across all five epochs, fluctuating only slightly within a narrow range from 0.916 to 0.920. This suggests that the model maintains strong performance on unseen data and that fine-tuning provides only modest improvements. The slight upward trend toward the final epoch indicates that the model does benefit from continued training. The overall stability implies that it was already well-aligned with the task from the start.

In contrast, the loss curves reveals a contradictory pattern. While the training loss steadily decreases, the validation loss consistently rises with each epoch. This divergence indicates overfitting: the model becomes increasingly specialized to the training data at the expense of generalization. The validation accuracy does not drop sharply, however the growing gap between training and validation loss shows the model is learning patterns that may not be helpful for additional data. This suggests that fewer training epochs or stronger regularization strategies such as stopping early or having less of the tunable parameters can assist in improving performance.

2. How does the fine-tuned DistilBERT model compare to the classical ML model? What advantages or limitations do transformers present over classical algorithms?

The fine-tuned DistilBERT model performs worse than the classical ML model. There is a clear advantage for the classical approach. While the fine-tuned transformer achieves an accuracy of only 0.50 and suffers from extreme imbalance in class predictions, the SVM achieves a stable and much higher accuracy of 0.89, with balanced precision, recall, and F1-scores for both classes. In this task, the SVM displays far greater reliability and consistency, especially in distinguishing between positive and negative samples. From this it is evident that while transformer models although can be fine tuned for a specific data set, can still lack in accuracy and underperform when the model overfits.

Transformers such as DistilBERT offer advantages such as deep contextual understanding, strong performance on NLP tasks, and the ability to transfer knowledge from large-scale pretraining. However, this comes with limitations as transformers are computationally expensive, requires extensive time for training, sensitive to hyperparameters, and prone to overfitting. On the other hand classical algorithms like SVMs or Logistic Regression are lightweight, efficient, and often times can be more robust in low-data scenarios. In this case, these properties allow the SVM to outperform the transformer-based model substantially.

3. What insights can you draw from the confusion matrix? Are there any patterns in the misclassifications?

The confusion matrices reveal clear differences in performance across the three classification models. The fine-tuned DistilBERT achieves the best results, with very low rates of both false positives and false negatives. The errors are minimal and evenly distributed, showcasing that fine-tuning allows the model to understand complex cues. In contrast, the pretrained DistilBERT without any fine-tuning performs poorly and shows a strong negative bias. It misclassifies the majority of positive reviews as negative, indicating that the model has not yet learned patterns specific to the current dataset. Since it has not been fine-tuned, the model relies solely on the general patterns it acquired during pretraining, which may not be the same labeling scheme or distribution of this task. As a result, it defaults to one label, leading to uniform and uninformative predictions.

In contrast, the SVM performs noticeably better. Its confusion matrix shows balanced error rates across both classes, with only slightly more misclassifications than the fine-tuned DistilBERT. This suggests that while modern transformer models can show superior performance when fine-tuned, classical machine-learning approaches can still perform well on large datasets specific to various linguistic preferences in conveying information.

4. Why might the fine-tuned model outperform the base model?

Despite its limitations, the fine-tuned DistilBERT model still performs better than the base model because fine-tuning provides objective-specific adjustments that the pretrained base model lacks. The base model has no exposure to the dataset's labeling scheme and therefore makes predictions that fail almost entirely for one of the classes. Fine-tuning exposes the model to examples of both positive and negative samples from the training set. This allows for capturing of accurate internal representations and decision boundaries. The improvement in F1-score for the positive class shows that the model has at least learned to recognize some task-relevant patterns.

5. Which model would you recommend for deployment in a real-world scenario, and why? Consider both performance and efficiency in your answer.

For real-world deployment, I would recommend the SVM. SVM is a strong and the most practical choice because it provides significantly higher accuracy, balanced precision and recall, and reliable performance across both classes. Its confusion matrix demonstrates consistent generalization, making it better suited for applications where both classes must be detected accurately. The SVM also offers important efficiency advantages: it requires minimal memory resources, fast training and inference times, and does not require specialized hardware such as GPUs. These properties make it easy to integrate into real-world systems and maintain over time.

Sources:

https://zilliz.com/learn/distilbert-distilled-version-of-bert

Fine-Tuning Sentiment Analysis Model | by Okan Yenigün | Artificial Intelligence in Plain English

IMDB sentiment analysis - EDA, ML, LSTM, BERT