Project B Report Luke Peter

1. AI test cases: [30 points]

TC-identifier: Test Case 1

TC-name: Short positive

TC-objective: Evaluate basic positive-word detection on a short input.

TC-input: Loved it! Great acting and a heartwarming story.

TC-reference-output: positive

TC-harm-risk-info: HC5, none

TC-identifier: Test Case 2

TC-name: longer more complex negative

TC-objective: test ability to handle more text and more complex review

TC-input: Although the cinematography is sometimes beautiful, the plot drags endlessly and the characters, who should have been compelling, feel hollow and unconvincing — a disappointing waste of time.

TC-reference-output: negative

TC-harm-risk-info: HC5, none

TC-identifier: Test Case 3

TC-name: ambiguous

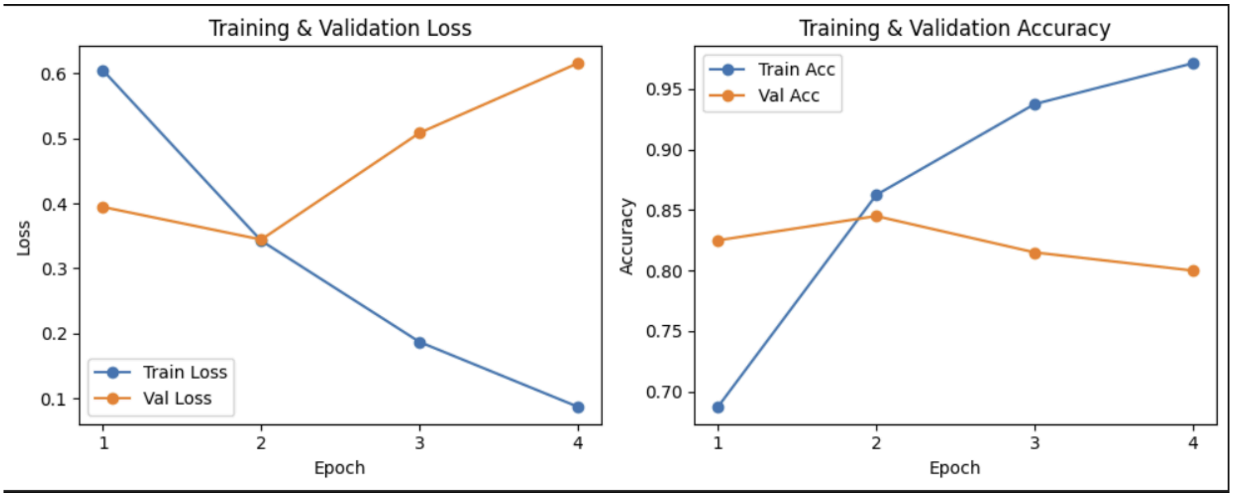
TC-objective: test ability to detect sarcasm and double-meaning

TC-input: Oh, fantastic another sequel that adds nothing new. Absolutely thrilled.

TC-reference-output: negative

TC-harm-risk-info: HC2, opinion manipulation

2. Accuracy and Loss Curves: [30 points]



The training loss decreases linearly as the number of epochs increases. The training accuracy also increases steadily as the number of epochs increases. The validation loss/accuracy improves from epoch 1 to 2, but worsens with 3 and 4 epochs. This means that the model may have begun overfitting after 2 epochs.

3. Confusion Matrices: [30 points]

Fine-tuned DistilBERT(4 epochs):

```
 [[100   0]
 [100   0]]
```

This matrix is consistent with the above curve, because with 4 epochs the model was overfitted, and eventually only predicted negatives.

Fine-tuned DistilBERT(2 epochs):

```
[[87 13]
 [14 86]]
```

With 2 epochs the confusion matrix shows the model is much more accurate.

Base DistilBERT:

```
[[ 2 98]
 [ 1 99]]
```

The base model with no training nearly always predicted positive. This shows why fine-tuning is essential to some degree, but too much will cause overfitting.

Classical ML model (Logistic regression):

```
[[4315  625]
 [ 465 4512]]
```

This has a much more balanced matrix,having about the same amount of misclassifications in both

4. Precision, Recall, and F1-Score: [30 points]

5. Performance Comparison: [30 points]

Model/scores	Accuracy	Precision	Recall	F1
GPT	0.73	0.713	0.770	0.740
Fine-tuned Dist(2 epoch)	0.865	0.8687	0.86	0.8643
Base Distilbert	0.49	0.494	0.82	0.6165
Logistic regression	0.8901	0.8783	0.9066	0.8922

6. Time Complexity: [30 points]

The Distilbert model with training took longer than all the other models. With only 1 epoch it took 2 minutes, 2 epochs took 5 minutes, and 4 epochs took about 9 minutes. The pretrained Distilbert and the logistic regression model only took about 7 seconds.

Questions: [50 points]

1. The accuracy and loss curves show that fine-tuning does help a model produce better results, have less losses and more accuracy. However, too much fine-tuning will overfit a model and the quality of the results will decrease. There is a “sweet spot” for fine-tuning that will in theory produce the best results.

2. In this case the fine-tuned Distilbert performed slightly worse than the Logistic regression model in the terms of all 4 evaluation metrics. The limitation of the transformer is that it takes a long time to execute without having a powerful GPU or other strong processing power. Because of this, I had to input only 200 reviews into the transformer in order to execute in a reasonable amount of time, while the Logistic regression model easily

processed 10,000 reviews in a fraction of the time. The advantage of the Distilbert model is that it is much more accurate than the classical model once it is sufficiently trained. I can infer this because despite having 2% as many reviews to train off of, it performed nearly as well as Logistic regression.

3. Some patterns I noticed is that the base distilbert model essentially predicts positive nearly every time, while the over-trained model predicts negative every time. You can clearly see that the fine-tuned DistilBERT model and the Logistic Regression model both had similar performance from the confusion matrices, however the number of samples in the DistilBERT model is only 200, while the logistic regression is 10,000. This further shows that DistilBERT is likely the best predictor given equal sample sizes.

4. The fine-tuned model outperforms the base model because the base model isn't designed specifically for sentiments of reviews. On the other hand, the fine-tuned model processes the sample of reviews and their sentiments, so it is then able to make more accurate predictions.

5. When looking at the efficiency and accuracy of this case, the Logistic Regression model would be the best model. In this case it was able to sample 9,800 more reviews, and have a slightly better accuracy, while only taking 7 seconds compared to 5 minutes. However, in the real-world, access to much faster computational power than my laptop is available. Even though it didn't perform as well in this project, with the ability to train it on as many samples as the Logistic Regression, DistilBERT would almost definitely take the lead in performance.

Challenges faced (for presentation):

- The first time I attempted to run DIstilBERT with just a 20% train/test split, but it was going to take nearly a whole day to run. I had to use a small sample size of 200 in order to complete the training in a reasonable time. As a result my fine-tuned model actually performed slightly worse than logistic regression, although I think it should have performed better.
- At first I tried using tensorflow but found it has been “deprecated” in transformers and was causing issues in my environment. I had to switch to Pytorch.
- More environment issues and performance/crashing issues.