Professor Review Transformer

By: Ayden Barrios

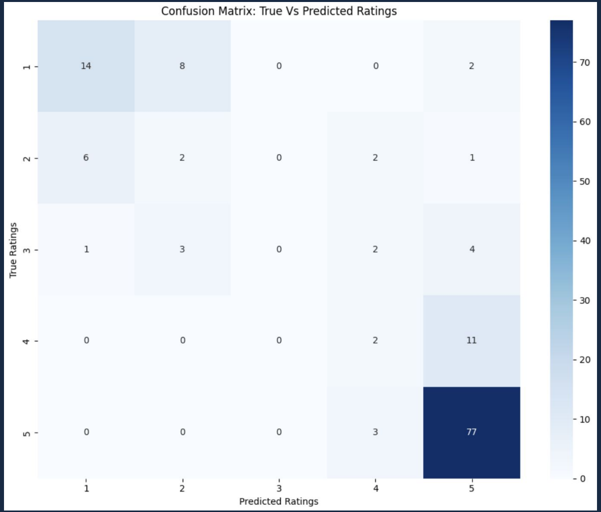
Code

Goal

This project involved using the PlanetTerp API to extract the reviews and ratings of 5 professors at the University of Maryland. A transformer was trained on the data and classified student reviews into ratings ranging from 1–5 stars. By adjusting parameters and comparing various models, I attempted to achieve the best accuracy that I could. The overall goal was to evaluate how well a fine-tuned language model could interpret qualitative feedback and correctly map it to a numerical rating. This allowed me to explore the strengths and limitations of transformer based approaches when working with small, real world datasets.

First Attempt

This was a solid start resulting in an accuracy of 68.7% and a mean absolute error of .54 I noticed that the main pitfalls were that the model was wrongly classifying 2 star ratings as 1 star and 4 star ratings as 5 star. This is likely due to the fact that 1 and 5 were the most common reviews in the data set.



Model: distilbert-base-uncased

Epochs = 3

Train/Val/Test split = 70/15/15

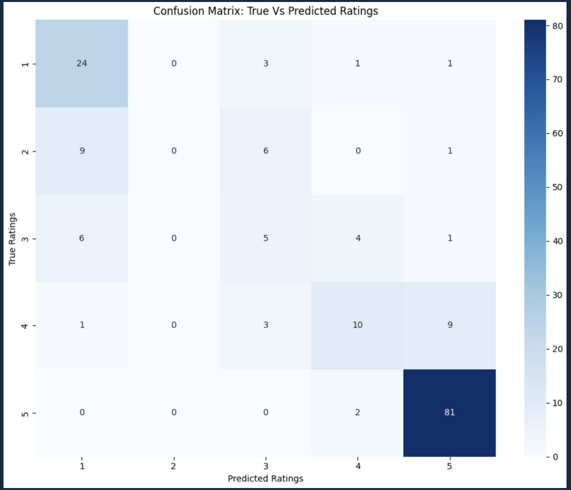
Batch size: (Train = 8, Eval = 16)

Total Reviews: 921

Learning rate: Default

Continued Tuning

Here I found professors with more reviews giving the model more data as well as increasing the epochs by 1. Along with introducing a dropout rate, I was able to get a 70% accuracy and a .44 mean squared error. The batch size here was decreased to 4 to see if it would be thrown off by noise, but it still did a solid job.



Model: bert-base-uncased

Epochs = 4

Train/Val/Test split = 80/8/12

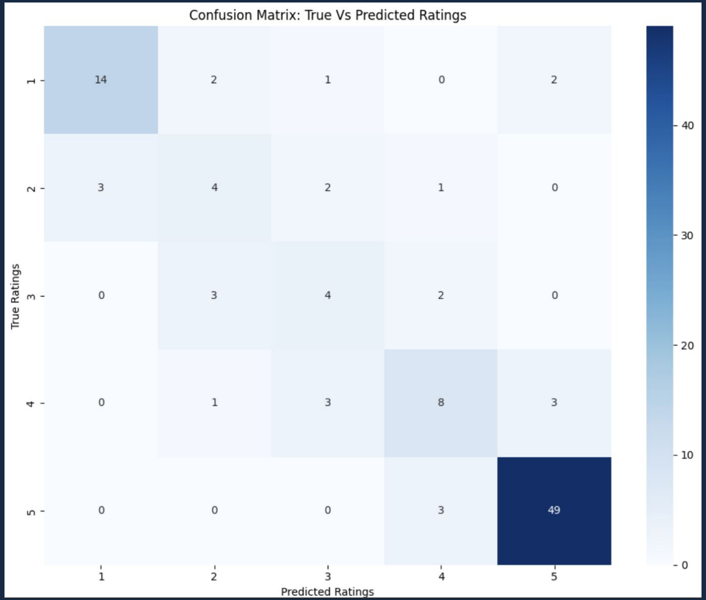
Batch size: (Train = 4, Eval = 16)

Total Reviews: 1390

Dropout Probability: 20%

Final Attempt

These were the best results I could obtain achieving an accuracy of 75.2%, a mean absolute error of .33, and a root mean squared error of .787



	1	2	3	4	5
1	14	2	1	0	2
2	3	4	2	1	0
3	0	3	4	2	0
4	0	1	3	8	3
5	0	0	0	3	49

Model: bert-base-uncased

Epochs = 5

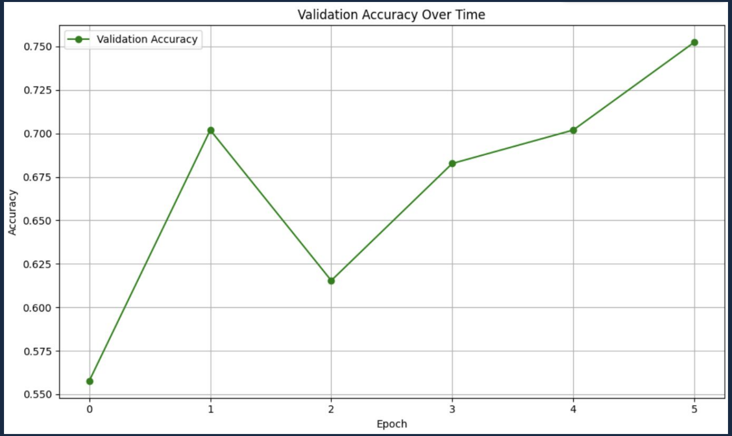
Train/Val/Test split = 85/7.5/7.5

Batch size: (Train = 8, Eval = 16)

Total Reviews: 1390

Learning rate: Default

Key Notes and Final model Accuracy Plot



Learning rate:

I mainly kept this at default because lowering it usually lowered accuracy

Dropout:

I used dropout for some of my run throughs and found that it helped reduce overfitting in certain cases

Token length:

512 was the sweet spot . It was computationally efficient while incorporating longer reviews

Review distribution:

This was a key pitfall of this project. The main failing classifications were within the 2-4 star ratings due to a lack of train data

Conclusions

Epochs

I found that increasing the number of epochs during training led to better learning and increased accuracy, but if I went too high for example over 5 then the model would start over fitting and the accuracy would start getting worse. Additionally, the eval loss would start increasing after 5-7 epochs so there was no point in using the extra GPU power

Train/Val/Test

More training data generally resulted in better accuracy and mae. However, more training data means less test data , which resulted in the model not having a good accuracy because of high penalization for a single mistake. I found the sweet spot to be 80-85% training data

Batch Size

The batch size determined how much data is ran through the transformer before updating weights lower numbers like 4 allowed the weights to be updated more frequently but made the model more sensitive to noise. I found that 8 or 16 was good because it made the model less noise sensitive while also updating the weights enough.

Model Choice

DistilBERT: This model was fast and less computationally heavy, but didn't learn the training data as well

BERT: This model was slower so it was more tedious to adjust parameters and compare, but it was generally more accurate

Total Reviews

More reviews helped the data during training because it had more data to learn from. One issue is that usually the professors with the most reviews on planetterp either have very good or very bad reviews. Thus, it was difficult to get an even split of 1/2/3/4/5 stars for the data set while sticking to a 5 professor constraint.