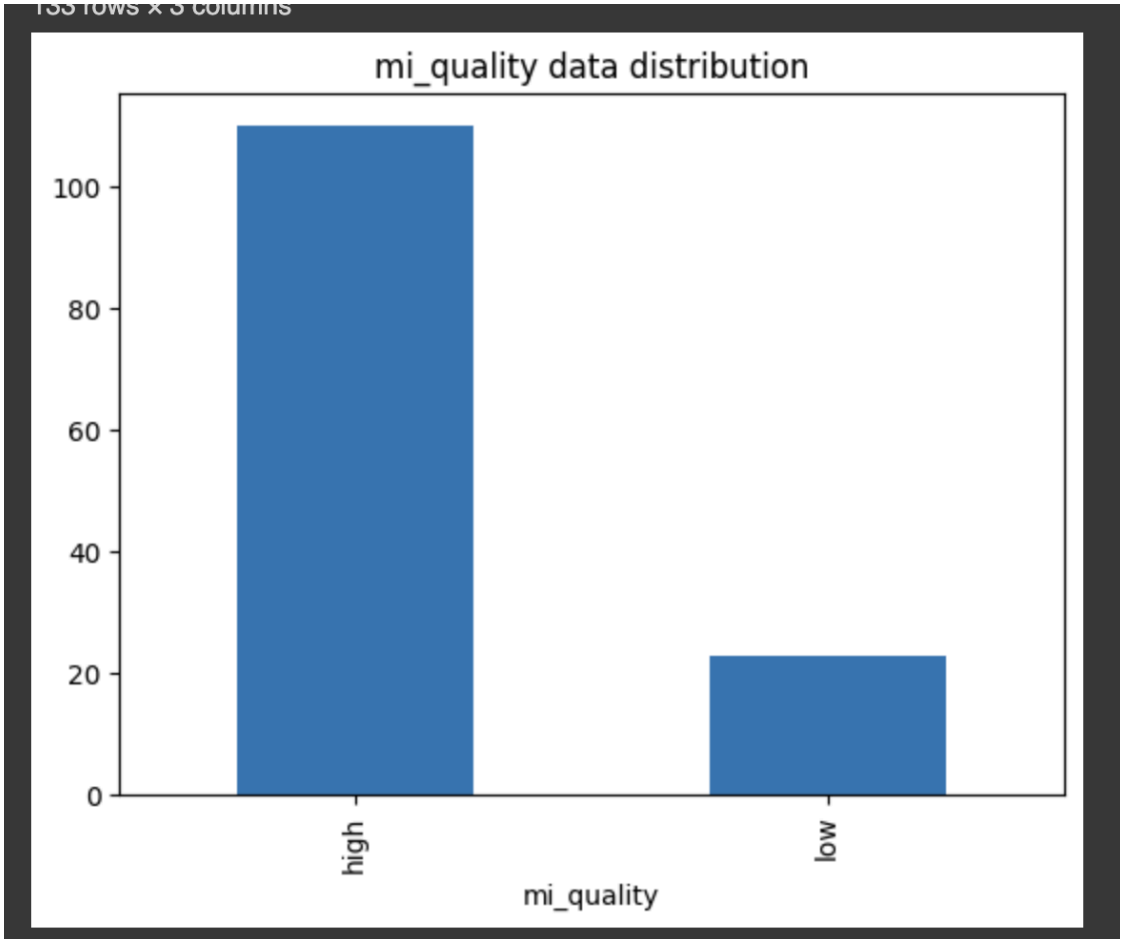
Reflex_ai take home assignment

Problem discussion

Download the AnnoMI dataset, the dataset includes transcripts from video demonstrations of high- and low-quality motivational interviewing. The dataset also includes annotations from experienced motivational interviewing practitioners.Perform analyses to explore 1 or 2 aspects of the dataset. Then, create a text classification model that predicts the main_therapist_behaviour label. Evaluate the model that you created. In this hypothetical example, the model will be used to characterize the behavior of MI practitioners during their clinical sessions.

Exploratory Data Analysis
The quality of the utterance (interaction between the client and therapist) can be categorized to two classes: high quality and low quality. I thought of just using the high quality data to avoid any problems in the analysis but since the numbers will only give us 110 high quality data and drop 23 low quality data, I decided to keep the data.

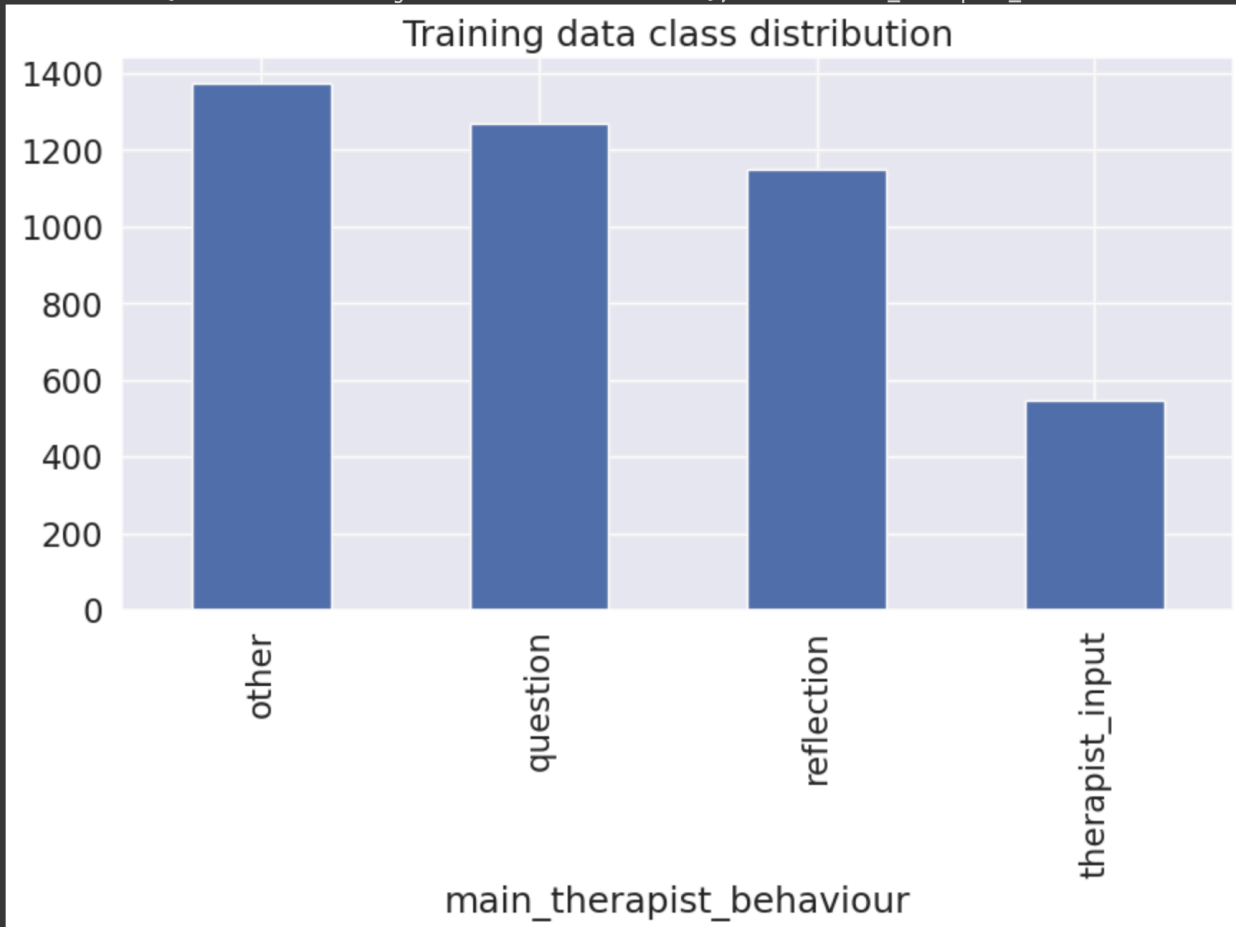


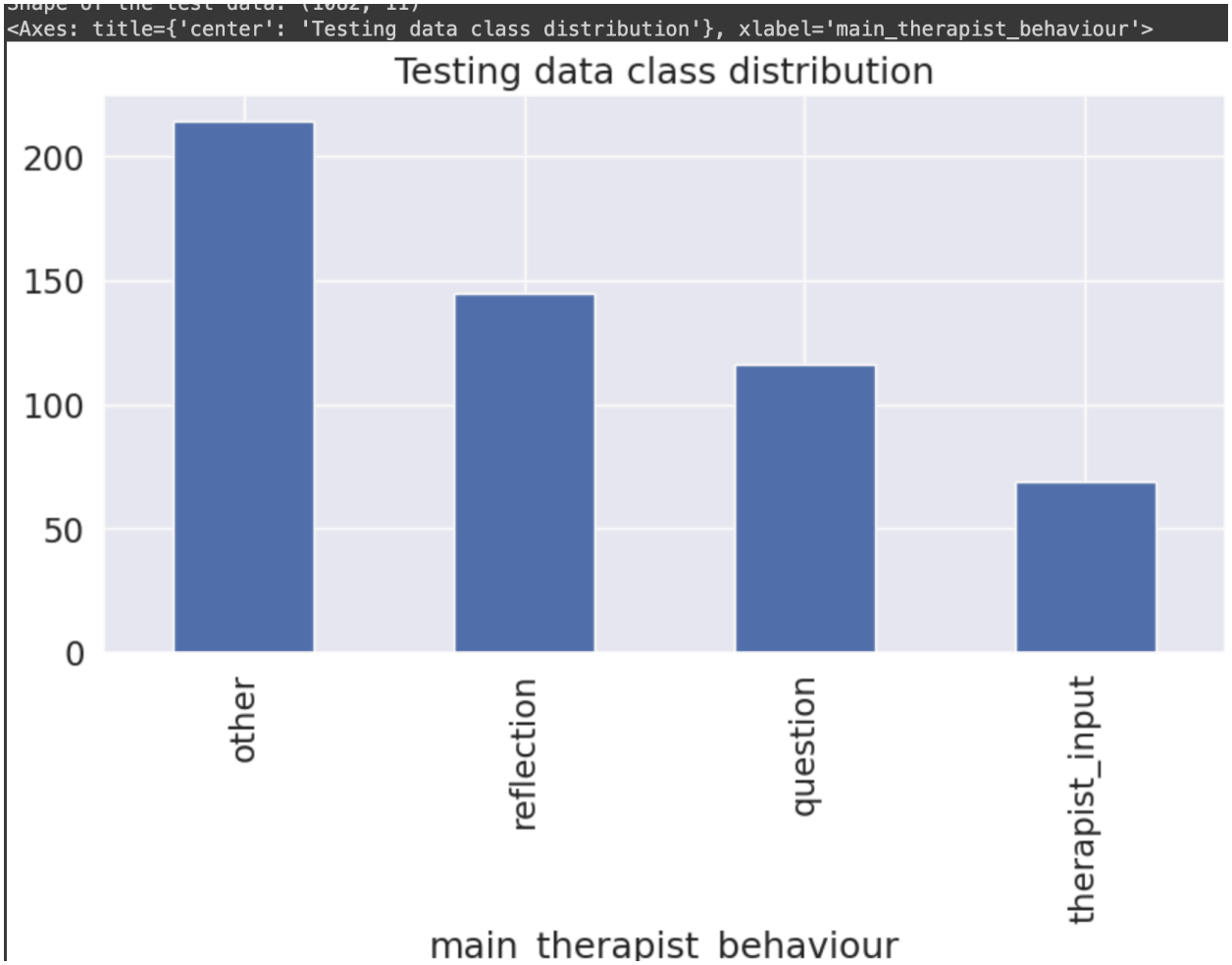
```
Result dictionary sorted by values ( in reversed order ) :
{'diet; reducing alcohol consumption; diabetes management': 40,
'birth control': 39,
'smoking cessation; reducing alcohol consumption': 38,
'managing life': 37,
'reducing drug use; following medical procedure': 36,
'taking steps towards getting help with day-care': 35,
'reducing alcohol consumption; safe sex': 34,
'engaging in community activities': 33,
'avoiding DOI': 32, 'reducing self-harm': 31,
'Being assertive with flatmate about moving out': 30,
'recognising success': 29,
'reducing alcohol consumption; compliance with rules': 28,
'providing information on medicines': 27,
'smoking cessation': 26,
'completion of community service': 25,
'increasing activity; taking medicine / following medical procedure': 24,
'not getting into a car with someone who is under the influence of drugs
or alcohol': 23,
'more exercise / increasing activity': 22,
'reducing coffee consumption': 21,
'reducing gambling': 20,
'opening up': 19,
'increasing self-confidence': 18,
'unidentifiable': 17,
'reducing drug use': 16,
'reducing violence': 15,
'changing approach to disease': 14,
'more exercise / increasing activity; weight loss': 13,
'charging battery': 12,
'asthma management': 11,
'compliance with rules': 10,
'anxiety management': 9,
'weight loss': 8,
'taking medicine / following medical procedure': 7,
'diabetes management': 6,
'smoking cessation ': 5,
'reducing recidivism': 4,
'better oral health': 3,
'problem recognition': 2,
'weight loss; diet': 1,
'reducing alcohol consumption': 0}
```

In this analysis I am using 'AnnoMI-simple.csv' as the AnnoMI-full.csv has duplicate value in the utterance_text column, diving deep into the data the dropping duplicate may affect the analysis so for the time being I decided to do analysis on the simple file.
The distribution of the classes are shown below in the bar plot.

The data was split to test and train at random with 10% for test and 90% for train.

```
Shape of the tarining data: (8617, 11)
<Axes: title={'center': 'Training data class distribution'}, xlabel='main_therapist_behaviour'>
```





Problem Solution - 1

Hypothesis:

Use a pre-trained BERT classification model to fine tune the classifier to make the prediction of the behavior of the therapist.

Pre-processing:

In the preprocessing step, I combined the client utterance text to the therapist utterance text based on the timestamp for context especially for the class Reflection and questions, I thought that context is important.

The input sequence length of the utterance text -

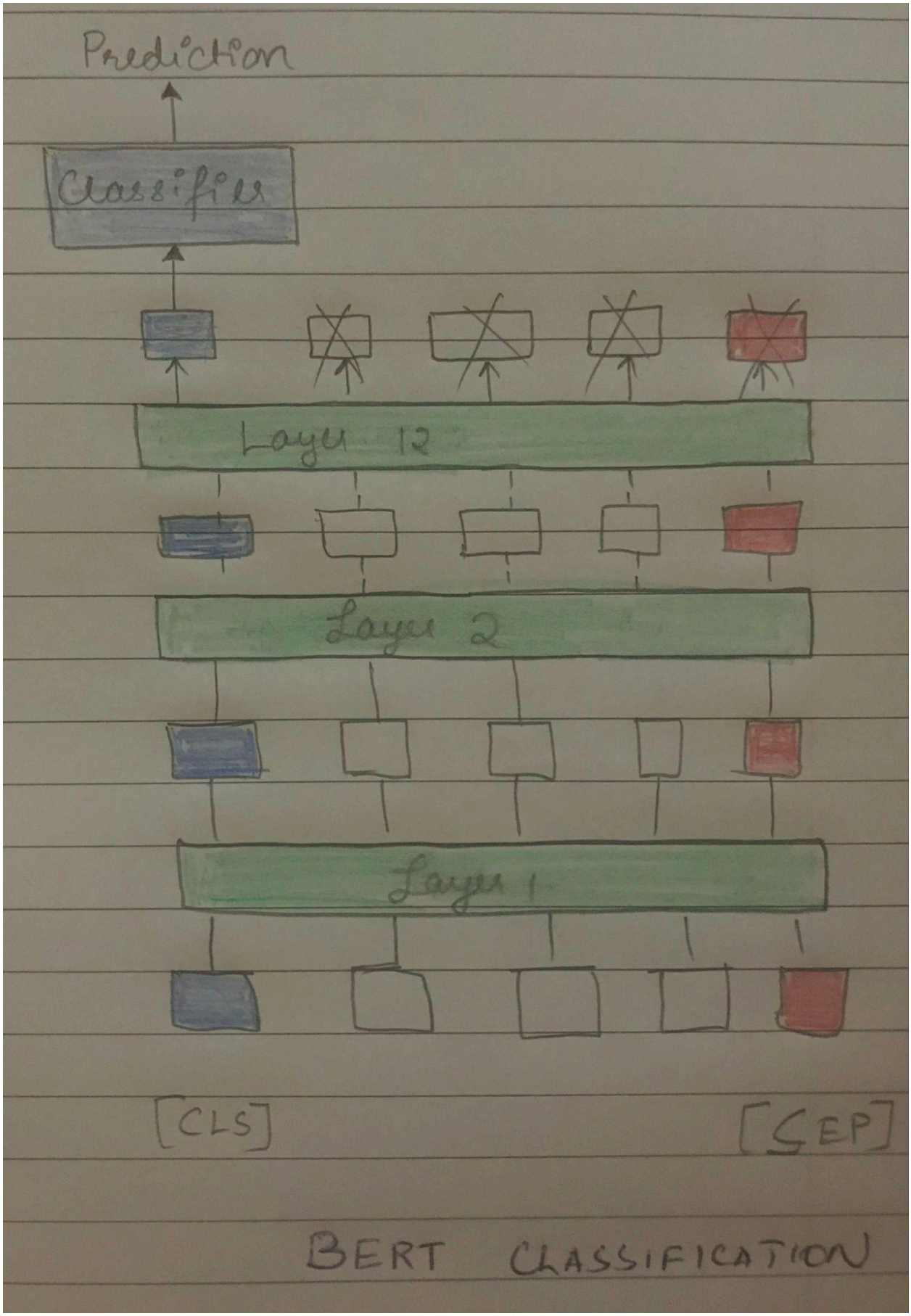
- Min length: 3 tokens
- Max length: 356 tokens
- Avg length: 20.0 tokens

3. The occurrence of a unique sequence in the utternace_text column for the therapist only, top 5 are shown below.

utterance_text	# count
Mm-hmm.	240
Yeah.	182
Okay.	153
Right.	75
Mm.	50

- Solution Architecture - Combine the client utterance text with the therapist utterance text and input to BERT base model, separated by SEP token. Take the embeddings from the CLS token from the last layer and apply Cross Entropy loss. Calculate the accuracy and loss for each epoch
Here is the model:

```
The BERT model has 199 different named parameters.

==== Embedding Layer ====

embeddings.word_embeddings.weight                       (30522, 768)
embeddings.position_embeddings.weight                     (512, 768)
embeddings.token_type_embeddings.weight                     (2, 768)
embeddings.LayerNorm.weight                                 (768,)
embeddings.LayerNorm.bias                                   (768,)

==== First Transformer ====

encoder.layer.0.attention.self.query.weight               (768, 768)
encoder.layer.0.attention.self.query.bias                   (768,)
encoder.layer.0.attention.self.key.weight                 (768, 768)
encoder.layer.0.attention.self.key.bias                     (768,)
encoder.layer.0.attention.self.value.weight               (768, 768)
encoder.layer.0.attention.self.value.bias                   (768,)
encoder.layer.0.attention.output.dense.weight             (768, 768)
encoder.layer.0.attention.output.dense.bias                 (768,)
encoder.layer.0.attention.output.LayerNorm.weight           (768,)
encoder.layer.0.attention.output.LayerNorm.bias             (768,)
encoder.layer.0.intermediate.dense.weight                (3072, 768)
encoder.layer.0.intermediate.dense.bias                     (3072,)
encoder.layer.0.output.dense.weight                      (768, 3072)
encoder.layer.0.output.dense.bias                           (768,)
encoder.layer.0.output.LayerNorm.weight                     (768,)
encoder.layer.0.output.LayerNorm.bias                       (768,)

==== Output Layer ====

encoder.layer.11.output.LayerNorm.weight                    (768,)
encoder.layer.11.output.LayerNorm.bias                      (768,)
pooler.dense.weight                                       (768, 768)
pooler.dense.bias                                           (768,)
```



BERT CLASSIFICATION

Used the following hyper parameters

- lr=0.0001
- epoch = 10

The training and test accuracy and loss were plotted and analyzed.
Figure 1: Shows the plot of training accuracy with weights for the class and without weights for the CE loss. The dark blue plot is for a higher learning rate

From the plot below, the model is overfitting the data, never converging.
Figure 2: Shows the loss is reducing with each epoch and reaches 0 after 8 epochs.

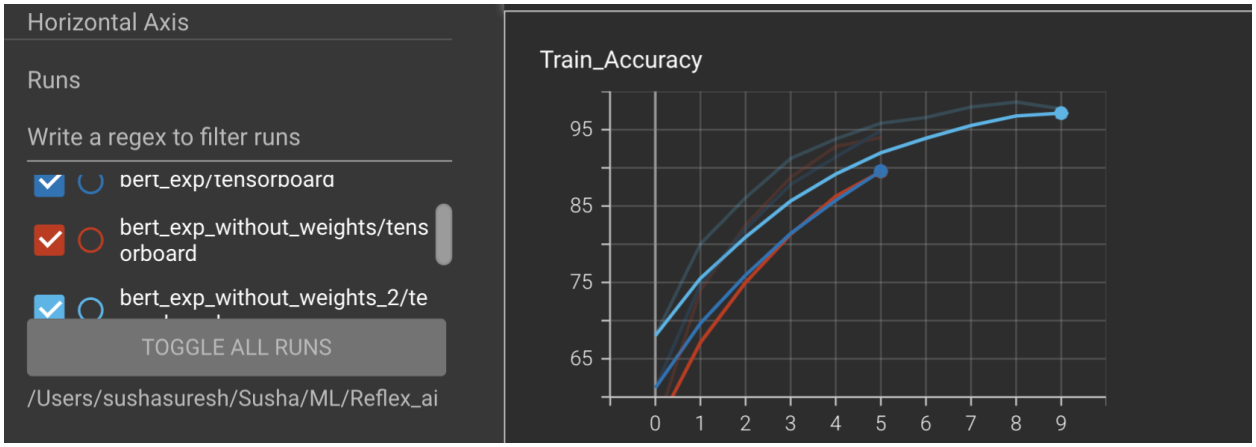


Fig :1 plot between the train accuracy and epoch

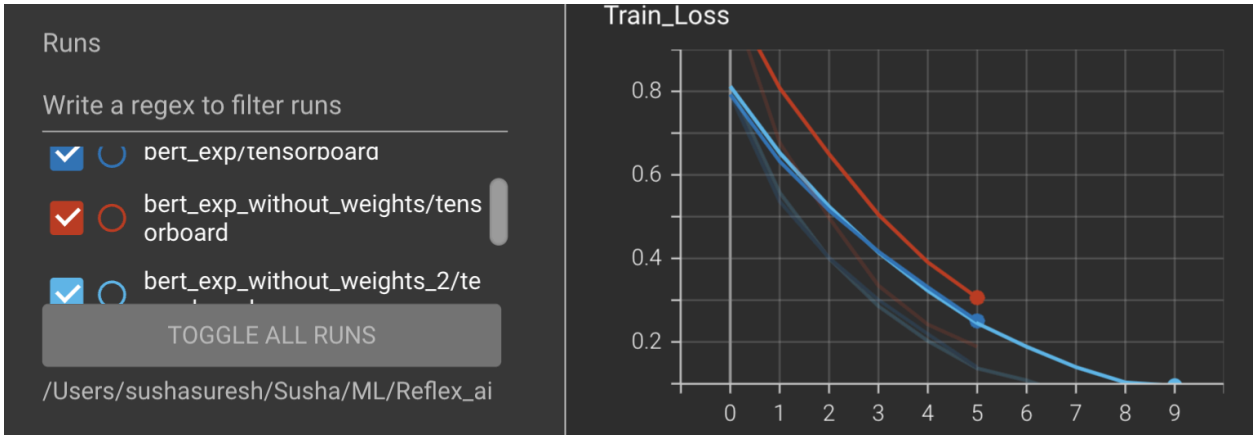


Fig :2 plot between the train loss and epoch

Fig :3 plot between the test accuracy and epoch

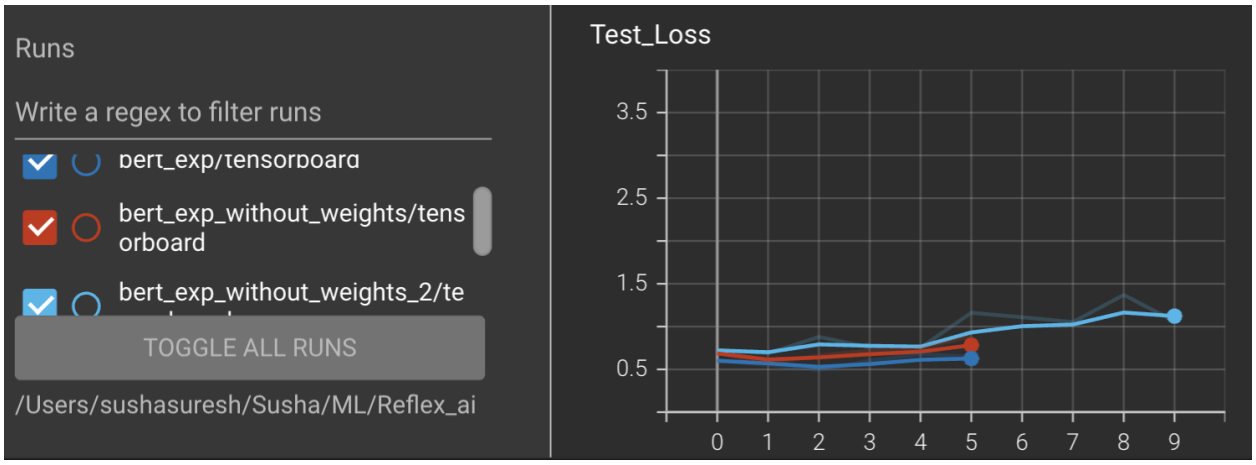


Fig :4 plot between the test loss and epoch

Conclusion:

1. with and without the weights to the loss function the model is overfitting, I need to look into the data more, to avoid the class imbalance by over or undersampling the data.
2. Better hyper parameter tuning will help.
3. Trying to take the avg embeddings of all the output layers rather than CLS token.
4. Try the SBERT model because the model is better with short sequences.
5. More data pre-processing to remove really short sequences from training and analysis.

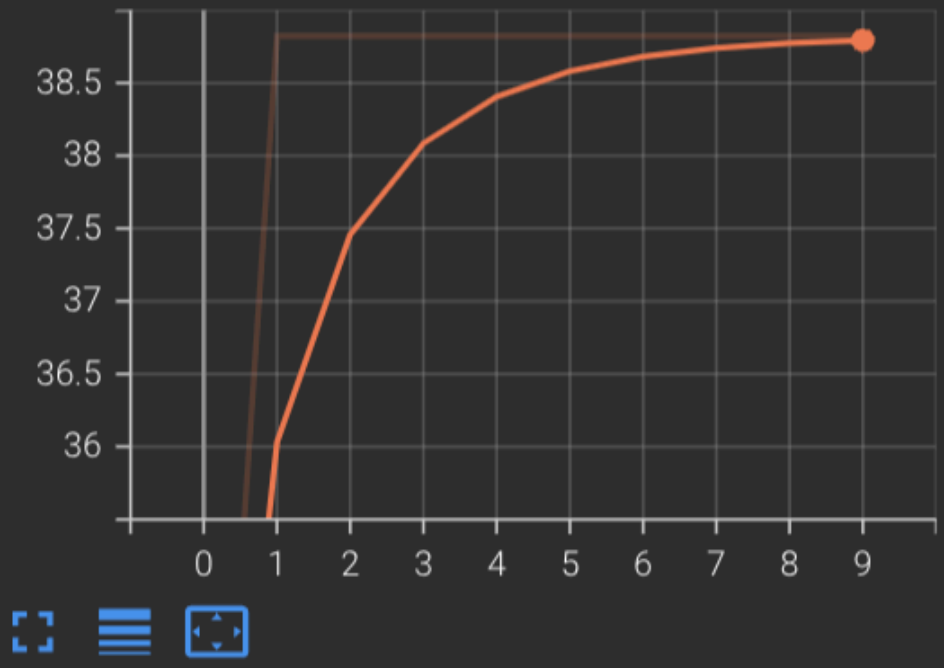
Problem Solution - 2

Hypothesis: It is important to know the complete context of the conversation rather than just the previous conversation based on the time-stamp for an effective analysis. LSTMs are really good at remembering the long term dependencies of a sequence.

Pre-processing: For each transaction id, I combined the utterances and gave flags to client and the therapist data. Used one layer of LSTM and it looked like the model was over-fitting. I added a dropout layer and other regularization techniques to help with over-fitting. It just caused underfitting or no training, needed more analysis and hyper-parameter tuning.

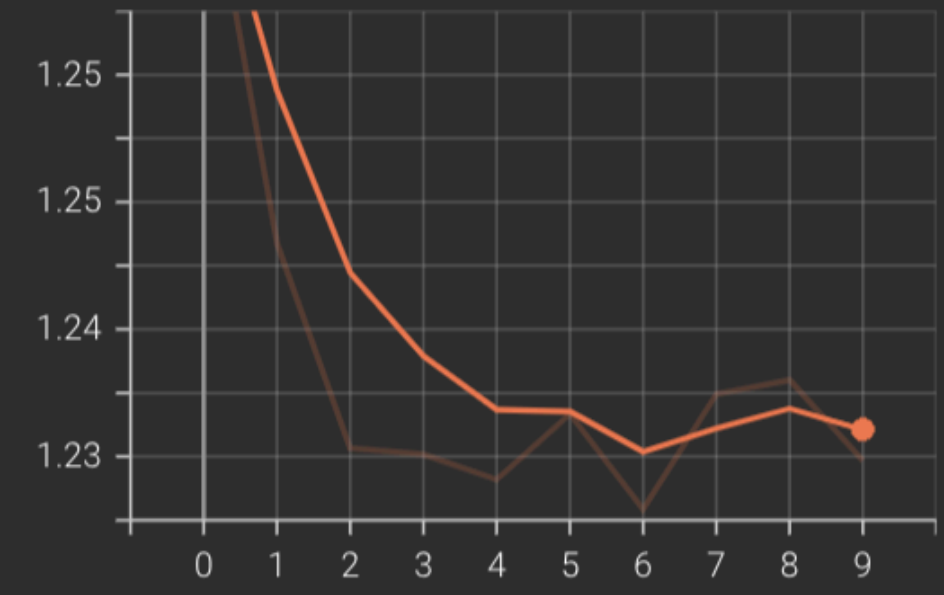
Test_Accuracy

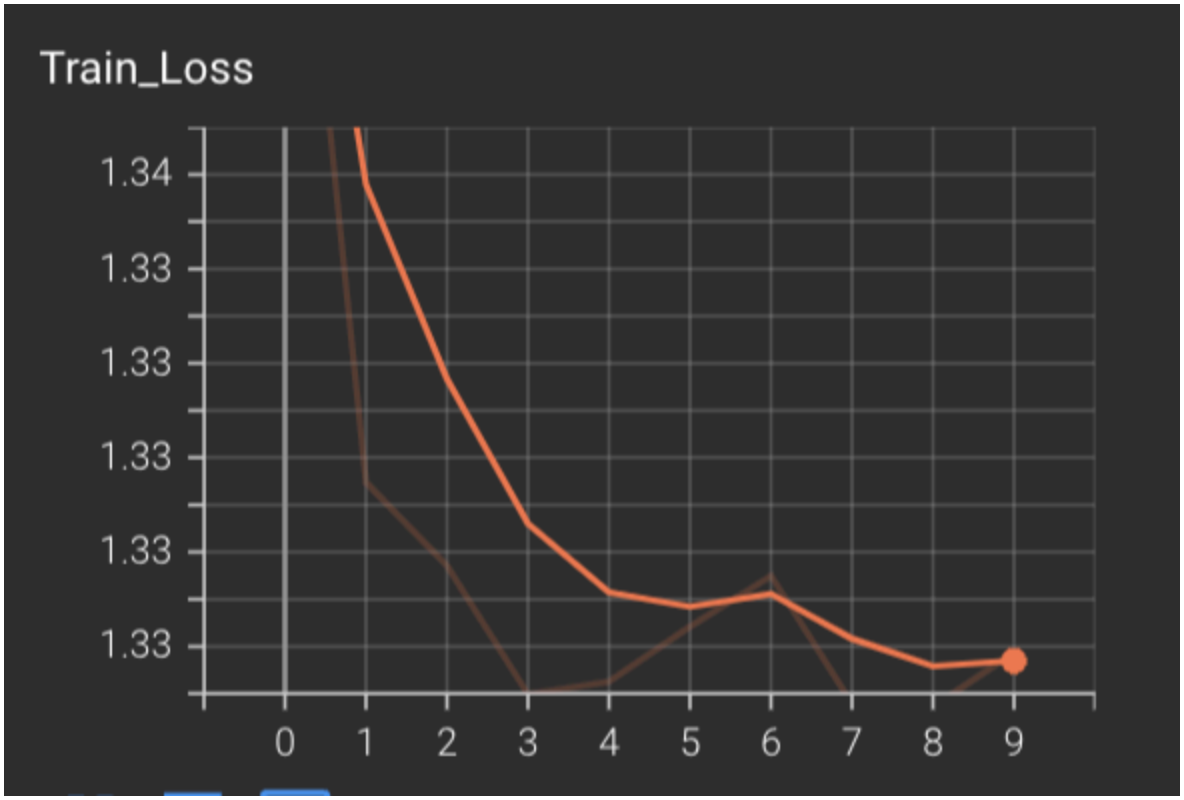
Test_Accuracy

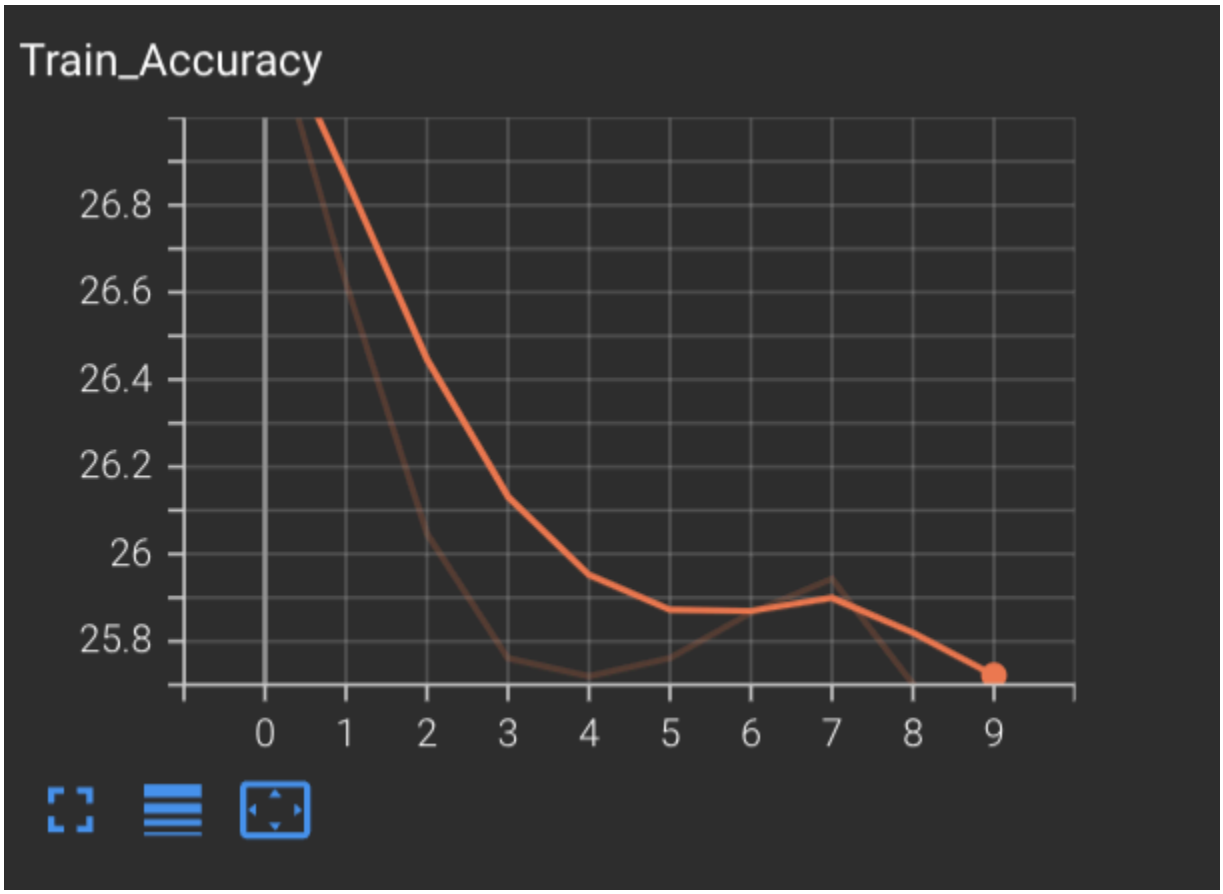


Test_Loss

Test_Loss




Train_Loss
1.34
1.33
1.33
1.33
1.33
1.33
0
1
2
3
4
5
6
7
8
9


Train_Accuracy
26.8
26.6
26.4
26.2
26
25.8
0
1
2
3
4
5
6
7
8
9