Report

Name: Bhavesh Sood
Roll No: 2019355
Name: Vishwajeet Kumar
Roll No: 2019128

1. Data Visualisation

```
array([[1,
        'predictive models are involved with predicting a value based on other values in the dataset. the process of training a predictive model is known as supervised learning.',
        'predict a value based on other values in the dataset. process of training a pred model is supervised learning.'],
       [1,
        'predict a value based on other values in the dataset. process of training a pred model is supervised learning.',
        'involved with predicting a value based on other values in the dataset; process of training this type of model is known as supervised learning'],
       [1,
        'predicting one value (the target variable) using other values',
        'predictive models are involved with predicting a value based on other values in the dataset.'],
       ...,
       [1,
        'calculates the shortest route from a source to a destination through a network and is restricted to follow a linear network',
        'network analysis is used in many different applications. four are highlighted here:  optimal routes closest facility service areas origin-destination cost matrix'],
       [0,
        'involves linear features or network of linear features that are topologically structured and is used to identify shortest path, time of travel, and service areas.',
        'network analysis is used in many different applications. four are highlighted here:  optimal routes closest facility service areas origin-destination cost matrix'],
       [0,
        'calculates the shortest route from a source to a destination through a network and is restricted to follow a linear network',
        'the study of the nodes (vertices) and edges (lines) in a network']],
      dtype=object)
```

Train data:

	Binary labels	Definition1	Definition2
0	1	predictive models are involved with predicting...	predict a value based on other values in the d...
1	1	predict a value based on other values in the d...	involved with predicting a value based on othe...
2	1	predicting one value (the target variable) usi...	predictive models are involved with predicting...
3	1	predictive models are involved with predicting...	predict value based on other values in data se...
4	0	predict a value based on other values in the d...	predict value based on other values in data se...
...	...	...	...
7545	0	- flow of distance of line of sight in an elec...	network analysis is used in many different app...
7546	0	involves linear features or network of linear ...	the study of the nodes (vertices) and edges (l...
7547	1	calculates the shortest route from a source to...	network analysis is used in many different app...
7548	0	involves linear features or network of linear ...	network analysis is used in many different app...
7549	0	calculates the shortest route from a source to...	the study of the nodes (vertices) and edges (l...

7550 rows × 3 columns

Test data:

	Binary labels	Definition1	Definition2
0	0	must be both relevant and accurate to achieve ...	a list of all external data needed for the use...
1	0	-any data that the program receives while it i...	the data values that are scanned by a program
2	1	vulnerability exists but wasn't detected by vu...	a security incident that isn't detected or rep...
3	0	vulnerability exists but wasn't detected by vu...	an error in which you are not alerted to a sit...
4	1	vulnerability exists but wasn't detected by vu...	term for when a scan fails to find real vulner...
...	...	...	...
1352	0	ensures that only authorized personnel can acc...	the generation, storage, distribution, deletio...
1353	0	- needs to be managed - should be a formal pro...	the generation, storage, distribution, deletio...
1354	0	- needs to be managed - should be a formal pro...	involves proper storing and management of keys...
1355	1	method of properly issuing, maintaining and or...	the methods for creating and managing cryptogr...
1356	0	method of properly issuing, maintaining and or...	ensures that only authorized personnel can acc...

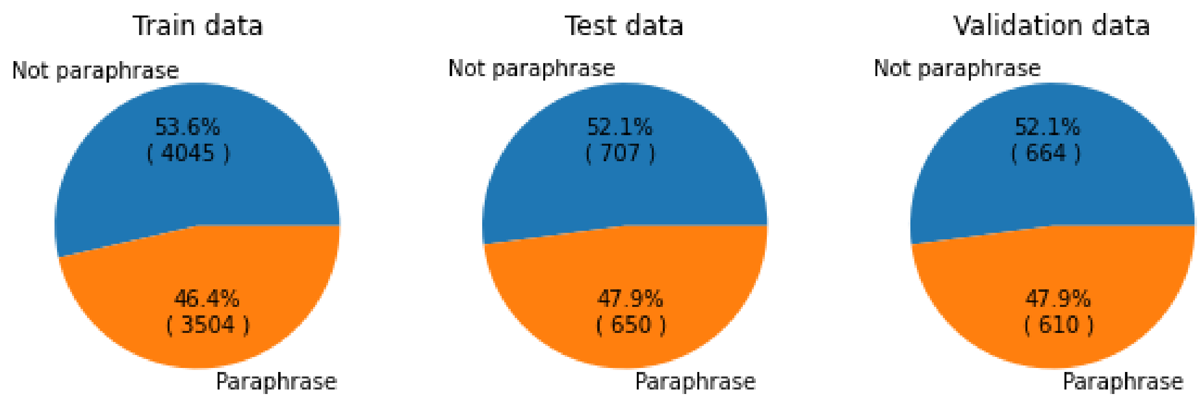
1357 rows × 3 columns

Validation data:

	Binary labels	Definition1	Definition2
0	0	time elapsed between clock readings	math symbol: Δt (delta t)
1	0	how much time it takes for something to happen...	elapsed time, or the time that has gone by
2	0	a knowledge based system that acts on a user's...	-special purpose knowledge based info system ...
3	0	these use built-in and learned knowledge to ma...	programs that work in the background without d...
4	1	software programs that use a built-in or learn...	programs that work in the background without d...
...	...	...	...
1270	1	concerned with the general structure of the so...	overall structure of the system, how system sh...
1271	0	how a software system should be organized and ...	once the interactions between the software sys...
1272	1	first set of activities in the up's design dis...	the process that selects and describes the exa...
1273	1	where you identify the overall structure of th...	where you identify the structure of the system...
1274	0	concerned with the overall structure of a syst...	concerned with the general structure of the so...

1275 rows × 3 columns

Number of examples of each type:



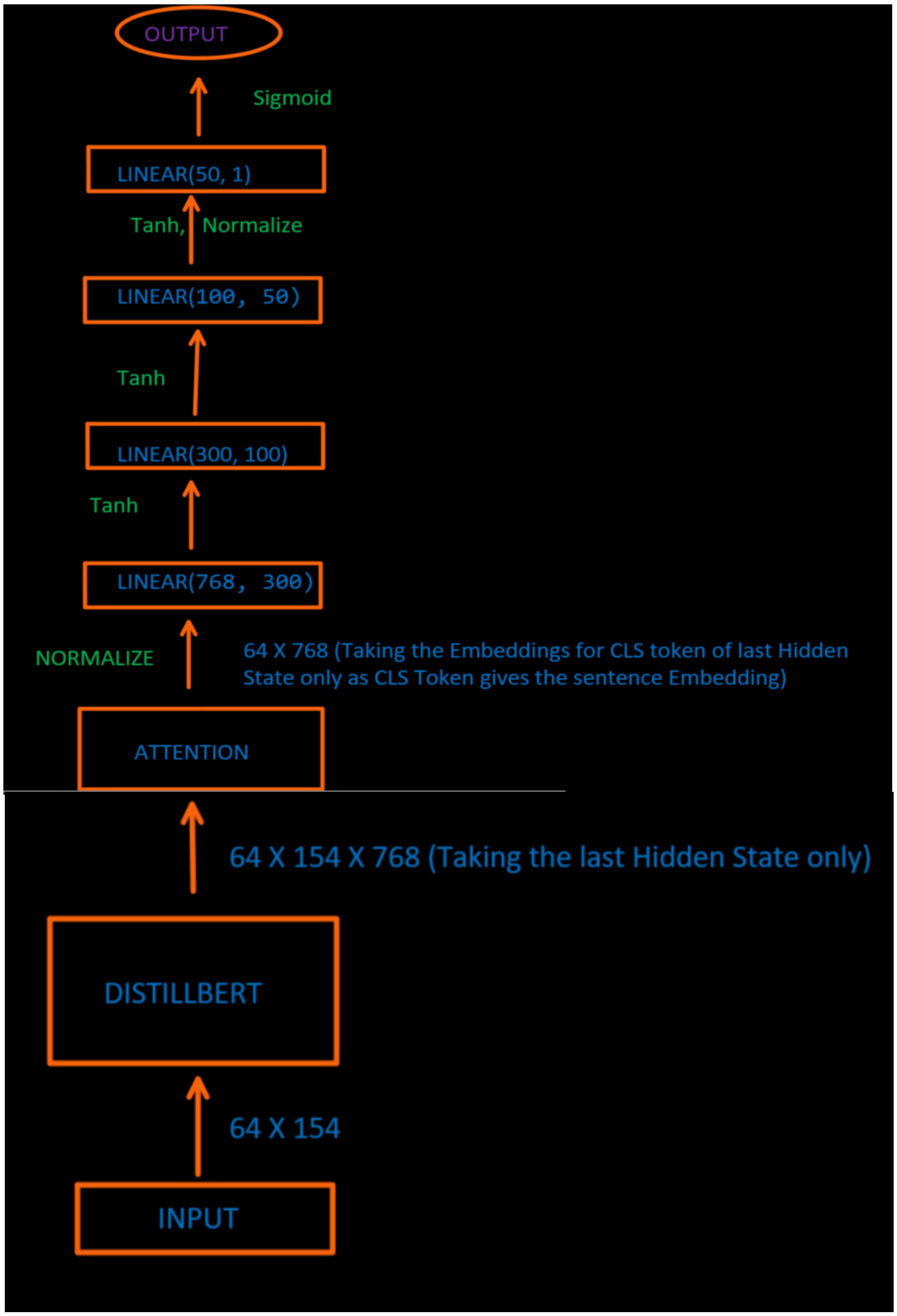
We notice that the data is almost equally balanced , specially in the training dataset.
Also we notice that in each dataset Not Paraphrase is more as compared to the Paraphrased examples. Thus simply outputting Not Paraphrase would give an accuracy of 50 % , however that would be wrong , and thus we should also look at F1 score for a better model .

2. Architecture

```
Net(
  (transformer): DistilBertModel(
    (embeddings): Embeddings(
      (word_embeddings): Embedding(30522, 768, padding_idx=0)
      (position_embeddings): Embedding(512, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (transformer): Transformer(
      (layer): ModuleList(
        (0): TransformerBlock(
          (attention): MultiHeadSelfAttention(
            (dropout): Dropout(p=0.1, inplace=False)
            (q_lin): Linear(in_features=768, out_features=768, bias=True)
            (k_lin): Linear(in_features=768, out_features=768, bias=True)
            (v_lin): Linear(in_features=768, out_features=768, bias=True)
            (out_lin): Linear(in_features=768, out_features=768, bias=True)
          )
          (sa_layer_norm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
          (ffn): FFN(
            (dropout): Dropout(p=0.1, inplace=False)
            (lin1): Linear(in_features=768, out_features=3072, bias=True)
            (lin2): Linear(in_features=3072, out_features=768, bias=True)
            (activation): GELUActivation()
          )
          (output_layer_norm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
        )
        (1): TransformerBlock(
          (attention): MultiHeadSelfAttention(
            (dropout): Dropout(p=0.1, inplace=False)
            (q_lin): Linear(in_features=768, out_features=768, bias=True)
            (k_lin): Linear(in_features=768, out_features=768, bias=True)
            (v_lin): Linear(in_features=768, out_features=768, bias=True)
            (out_lin): Linear(in_features=768, out_features=768, bias=True)
          )
          (sa_layer_norm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
          (ffn): FFN(
            (dropout): Dropout(p=0.1, inplace=False)
            (lin1): Linear(in_features=768, out_features=3072, bias=True)
            (lin2): Linear(in_features=3072, out_features=768, bias=True)
            (activation): GELUActivation()
          )
          (output_layer_norm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
        )
        (2): TransformerBlock(
          (attention): MultiHeadSelfAttention(
            (dropout): Dropout(p=0.1, inplace=False)
            (q_lin): Linear(in_features=768, out_features=768, bias=True)
            (k_lin): Linear(in_features=768, out_features=768, bias=True)
            (v_lin): Linear(in_features=768, out_features=768, bias=True)
            (out_lin): Linear(in_features=768, out_features=768, bias=True)
          )
          (sa_layer_norm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
          (ffn): FFN(
            (dropout): Dropout(p=0.1, inplace=False)
            (lin1): Linear(in_features=768, out_features=3072, bias=True)
            (lin2): Linear(in_features=3072, out_features=768, bias=True)
            (activation): GELUActivation()
          )
          (output_layer_norm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
        )
```

```
        (3): TransformerBlock(
          (attention): MultiHeadSelfAttention(
            (dropout): Dropout(p=0.1, inplace=False)
            (q_lin): Linear(in_features=768, out_features=768, bias=True)
            (k_lin): Linear(in_features=768, out_features=768, bias=True)
            (v_lin): Linear(in_features=768, out_features=768, bias=True)
            (out_lin): Linear(in_features=768, out_features=768, bias=True)
          )
          (sa_layer_norm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
          (ffn): FFN(
            (dropout): Dropout(p=0.1, inplace=False)
            (lin1): Linear(in_features=768, out_features=3072, bias=True)
            (lin2): Linear(in_features=3072, out_features=768, bias=True)
            (activation): GELUActivation()
          )
          (output_layer_norm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
        )
        (4): TransformerBlock(
          (attention): MultiHeadSelfAttention(
            (dropout): Dropout(p=0.1, inplace=False)
            (q_lin): Linear(in_features=768, out_features=768, bias=True)
            (k_lin): Linear(in_features=768, out_features=768, bias=True)
            (v_lin): Linear(in_features=768, out_features=768, bias=True)
            (out_lin): Linear(in_features=768, out_features=768, bias=True)
          )
          (sa_layer_norm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
          (ffn): FFN(
            (dropout): Dropout(p=0.1, inplace=False)
            (lin1): Linear(in_features=768, out_features=3072, bias=True)
            (lin2): Linear(in_features=3072, out_features=768, bias=True)
            (activation): GELUActivation()
          )
          (output_layer_norm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
        )
        (5): TransformerBlock(
          (attention): MultiHeadSelfAttention(
            (dropout): Dropout(p=0.1, inplace=False)
            (q_lin): Linear(in_features=768, out_features=768, bias=True)
            (k_lin): Linear(in_features=768, out_features=768, bias=True)
            (v_lin): Linear(in_features=768, out_features=768, bias=True)
            (out_lin): Linear(in_features=768, out_features=768, bias=True)
          )
          (sa_layer_norm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
          (ffn): FFN(
            (dropout): Dropout(p=0.1, inplace=False)
            (lin1): Linear(in_features=768, out_features=3072, bias=True)
            (lin2): Linear(in_features=3072, out_features=768, bias=True)
            (activation): GELUActivation()
          )
          (output_layer_norm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
        )
      )
    )
  )
  (attention): MultiheadAttention(
    (out_proj): NonDynamicallyQuantizableLinear(in_features=768, out_features=768, bias=True)
  )
  (fc1): Linear(in_features=768, out_features=300, bias=True)
  (fc2): Linear(in_features=300, out_features=100, bias=True)
  (fc3): Linear(in_features=100, out_features=50, bias=True)
  (fc4): Linear(in_features=50, out_features=1, bias=True)
)
```

My Model:



So after the output of the DistilBert Transformer we get a 768 embedding of each word of input, we then pass it through a self-attention layer(dropout of 0.2) where we get again an 768 word embedding of each token. Then we take the <cls> token embedding and use it as a sentence embedding and pass it to the Feed Forward Net as shown above using Tanh as activation and a normalise layer.

3. Freeze All Layers of Distil Bert

```
model = DistilBertModel.from_pretrained('distilbert-base-uncased').to(device)
for param in model.parameters():
  param.requires_grad = False
```

We freeze all the layers of the transformer like this .

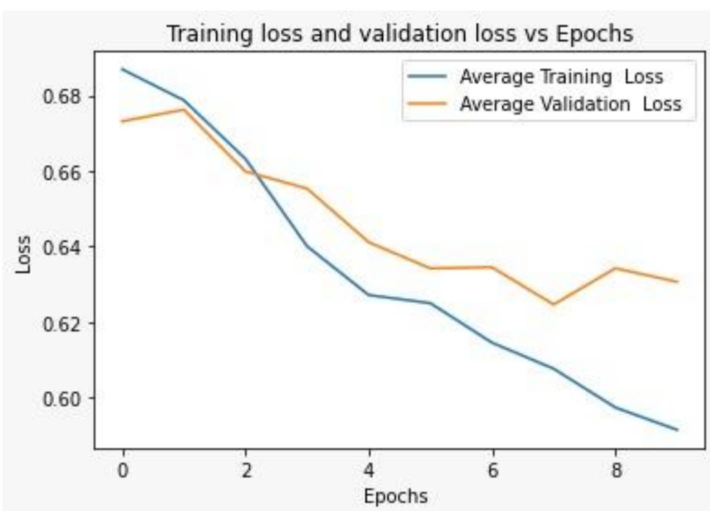
```
class Net (nn.Module) :
  def __init__ (self,transformer):
    super().__init__()
    # define the layers here
    self.transformer=transformer
```

Now even though the first layer of our model is the transformer , its weight would stay freezed and gradient would not flow back , thus only Our FF Network would fine tune itself to the data.

a. Accuracy and F1 Score are :

```
The accuracy score is : 0.6624907885040531
The weighted F1_score is : 0.6571899848528812
The classification report is :
              precision    recall  f1-score   support

           0       0.75      0.53      0.62       707
           1       0.61      0.80      0.69       650

    accuracy                           0.66      1357
   macro avg       0.68      0.67      0.66      1357
weighted avg       0.68      0.66      0.66      1357

(0.6624907885040531, 0.6571899848528812)
```

b. Train and Validation Loss Plots are:



c. 2 Misclassified Examples are :

```
[0
 '-any data that the program receives while it is running -from file or user input -is typically &"un-meaningful&" until &"processed&" -data that is typed on a keyboard'
 'the data values that are scanned by a program']
[0 "vulnerability exists but wasn't detected by vulnerability scanner"
 'an error in which you are not alerted to a situation when you should be alerted due to which, you miss crucial things.']
```

Both sentences have some identical words in same context and also the second example is a false negative type of sentence, Both the definition are saying something and then denyng it using "was not " type of words. So the model is not able to capture their context completely.

4. <u>Freeze Some Layers of Distil Bert</u>

```python
model_some_freeze_2 = DistilBertModel.from_pretrained('distilbert-base-uncased').to(device)
    # We freeze here the embeddings of the model
for param in model_some_freeze_2.embeddings.parameters():
      param.requires_grad = False
cnt=0
for i in model_some_freeze_2.transformer.layer:
  cnt+=1
  for j in i.parameters():
    j.requires_grad = False
  if(cnt==4) : break

net_some_freez_2=Net(model_some_freeze_2).to(device)
```

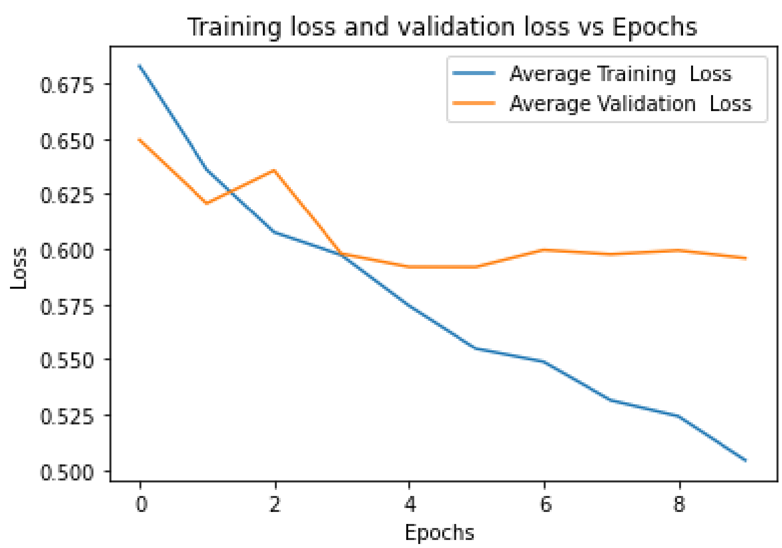
Here we Freeze the embedding layers and the first 4 transformer blocks of the DistilBert Transformer, the rest of the transformer is open to be fine tuned to the dataset.

a. Accuracy and F1 Score are :

```
The accuracy score is : 0.7030213706705969
The weighted F1_score is : 0.7029004337197505
The classification report is :
              precision    recall  f1-score   support

           0       0.71      0.72      0.72       707
           1       0.69      0.68      0.69       650

    accuracy                           0.70      1357
   macro avg       0.70      0.70      0.70      1357
weighted avg       0.70      0.70      0.70      1357

(0.7030213706705969, 0.7029004337197505)
```

We get an F1 score of above 70 % (weighted avg) , otherwise for 0 class it is of 72%.

b. Train and Validation Loss Plots are:



c. 2 Misclassified Examples are :

```
[0
 '-any data that the program receives while it is running -from file or user input -is typically &"un-meaningful&" until &"processed&" -data that is typed on a keyboard'
 'the data values that are scanned by a program']
[0 "vulnerability exists but wasn't detected by vulnerability scanner"
 'an error in which you are not alerted to a situation when you should be alerted due to which, you miss crucial things.']
```

Both sentences have some identical words in same context and also the second example is a false negative type of sentence, Both the definition are saying something and then denyng it using "was not " type of words. So the model is not able to capture their context completely. We notice same examples like in q3 , this is because these two examples are containing almost same words in their context.

5. Comparison

We notice that freezing only some layers of the transformed provided a better result in terms of both accuracy and F1 score. This could be due to the fact that keeping the embedding layers and initial transformer layers of the DistilBert Transformer freezed we use it as it is from the pretrained model of hugging face library which has been trained on the basis of general knowledge of English language. After that the last few layers that ultimately connect to our own FF network are left unfrozen so that they also learn from the data and update the weights accordingly. This allowed the final output of the DistilBert to be a little more customized to this particular task and thus we achieve slightly higher accuracy and F1 score.

Key differences:

Model	Accuracy	F1	Precision	Recall
All frozen	66.24	65.7	68	66
Some Frozen	70.3	70.29	70	70

We notice that also Precision and Recall increase slightly in the Fine Tuned DisitilBert model with only some Frozen Layers.

6. Various Hyperparameters Testing

Model with learning rate : 0.0003 and optimizer : SGD

```
The accuracy score is : 0.5210022107590273
The weighted F1_score is : 0.3569269021382096
The classification report is :
              precision    recall  f1-score   support

           0       0.52      1.00      0.69       707
           1       0.00      0.00      0.00       650

    accuracy                           0.52      1357
   macro avg       0.26      0.50      0.34      1357
weighted avg       0.27      0.52      0.36      1357

---------------------------------------------------------------------
```

Model with learning rate : 0.001 and optimizer : SGD

```
The accuracy score is : 0.5210022107590273
The weighted F1_score is : 0.3569269021382096
The classification report is :
              precision    recall  f1-score   support

           0       0.52      1.00      0.69       707
           1       0.00      0.00      0.00       650

    accuracy                           0.52      1357
   macro avg       0.26      0.50      0.34      1357
weighted avg       0.27      0.52      0.36      1357

-------------------------------------------------------------------
```

Model with learning rate : 0.01 and optimizer : SGD

```
The accuracy score is : 0.5210022107590273
The weighted F1_score is : 0.3569269021382096
The classification report is :
              precision    recall  f1-score   support

           0       0.52      1.00      0.69       707
           1       0.00      0.00      0.00       650

    accuracy                           0.52      1357
   macro avg       0.26      0.50      0.34      1357
weighted avg       0.27      0.52      0.36      1357

-------------------------------------------------------------------
```

Model with learning rate : 0.1 and optimizer : SGD

```
The accuracy score is : 0.63448784082535
The weighted F1_score is : 0.6161114280462696
The classification report is :
              precision    recall  f1-score   support

           0       0.78      0.42      0.54       707
           1       0.58      0.87      0.70       650

    accuracy                           0.63      1357
   macro avg       0.68      0.64      0.62      1357
weighted avg       0.68      0.63      0.62      1357
```

Model with learning rate : 0.0003 and optimizer : AdamW

```
The accuracy score is : 0.6750184229918939
The weighted F1_score is : 0.6746925646078971
The classification report is :
              precision    recall  f1-score   support

           0       0.68      0.70      0.69       707
           1       0.67      0.64      0.65       650

    accuracy                           0.68      1357
   macro avg       0.67      0.67      0.67      1357
weighted avg       0.67      0.68      0.67      1357

-----------------------------------------------------------------------
```

Model with learning rate : 0.001 and optimizer : AdamW

```
The accuracy score is : 0.5210022107590273
The weighted F1_score is : 0.3569269021382096
The classification report is :
              precision    recall  f1-score   support

           0       0.52      1.00      0.69       707
           1       0.00      0.00      0.00       650

    accuracy                           0.52      1357
   macro avg       0.26      0.50      0.34      1357
weighted avg       0.27      0.52      0.36      1357

-----------------------------------------------------------------------
```

Model with learning rate : 0.01 and optimizer : AdamW

```
The accuracy score is : 0.5210022107590273
The weighted F1_score is : 0.3569269021382096
The classification report is :
              precision    recall  f1-score   support

           0       0.52      1.00      0.69       707
           1       0.00      0.00      0.00       650

    accuracy                           0.52      1357
   macro avg       0.26      0.50      0.34      1357
weighted avg       0.27      0.52      0.36      1357

-----------------------------------------------------------------------
```

Model with learning rate : 0.1 and optimizer : AdamW

```
The accuracy score is : 0.5210022107590273
The weighted F1_score is : 0.3569269021382096
The classification report is :
              precision    recall  f1-score   support

           0       0.52      1.00      0.69       707
           1       0.00      0.00      0.00       650

    accuracy                           0.52      1357
   macro avg       0.26      0.50      0.34      1357
weighted avg       0.27      0.52      0.36      1357
```

In General We observe that for some of the learning rates, we observe that accuracy and F1 come out to be same , as it might be jumping very big in the gradient curve and thus not able to move down the hill .

In general AdamW provided a better accuracy and F1 score.

And a smaller learning rate worked better for us.

BONUS:

Our best model as shown in Q4 has higher F1 than 70%.