UNIVERSITY OF ATHENS
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

Deep Learning for NLP

Student name: <Αγγελος Κόντος>
sdi: <sdi2000089>

Course: *Artificial Intelligence II (M138, M226, M262, M325)*
Semester: *Fall Semester 2023*

Contents

1 Abstract 2

2 Data processing and analysis 2
2.1 Pre-processing 2
2.2 Analysis 2
2.3 Data partitioning for train, test and validation 2
2.4 Vectorization 2

3 Algorithms and Experiments 3
3.1 Experiments 3
3.1.1 Table of trials 3
3.2 Hyper-parameter tuning 3
3.3 Optimization techniques 4
3.4 Evaluation 4
3.4.1 F1 SCORE 4
3.4.2 ROC curve 4
3.4.3 Learning Curve 7
3.4.4 Confusion matrix 10

4 Results and Overall Analysis 13
4.1 Results Analysis 13
4.1.1 Best trial 13
4.1.2 F1 SCORE 13
4.1.3 ROC curve 14
4.1.4 Learning Curve 14
4.1.5 Confusion matrix 15
4.2 Comparison with the first project 16
4.3 Comparison with the second project 16
4.4 Comparison with the third project 16

5 Bibliography 16

1. Abstract

The job I have for this activity is to get 3 (dataframes) lists of elements where they have a small text (tweets) in which political party they are and what type they are (positive negative neutral), and after modifying them in various ways to output the ones I think are useless for the search and then learn in my system with the help of vectorizer and logisticregrasion so that it can receive suggestions and then see what kind they are (positive negative neutral).

2. Data processing and analysis

2.1. Pre-processing

For this step at the beginning I took out all the # and @ along with the sentences that follow also making all the chapters small to make it easier to group them I did the same by taking out the accents and with some library for some further grouping with lemmatization I also have try deleting extra spaces if there are any and of course my own stopword

2.2. Analysis

I had tried and seen words from him, nd, kk, etc., etc., various words that are either in general use by citizens or parties due to the type of tweets, also connections with which party he is from, etc., but I did not find a good solution to the connection with the parties where it's in, but it deletes someone from common simple words, another pattern I saw but didn't find a way to use is that when he was at the political party, I hissed and they said nd, mitsotakis, etc. were almost always negative and on the contrary if he was at the political party nd and general if they mentioned their party and names of their party leaders then it is positive and if they were opponents then negative, also the world coud is:

2.3. Data partitioning for train, test and validation

The way I split the dataset is by experimenting, after first playing with all the data and seeing what better combination exists (e.g. from stopwords, bring chapters, extract tons, use some libraries, etc., etc.) decision to keep 100 % I also tried 90 % 80% 70% and I could see how it only dropped to 90 to have 39% (and only when I passed randomstate=1000 in all but it was smaller) it was like 100% is 38-39-41 the other was 38-39-40 (100%) and since there is a 41% I decided to keep 90% of the dataset whatever I put in TfidfVectorizer does not change the result

2.4. Vectorization

I use tf-idf because it contains not only if a word exists or not but also the probabilities which are general and in a fast way without putting the ones that don't exist with 0 but only those that are in but less time is spent searching and its creation

3. Algorithms and Experiments

3.1. Experiments

Without any change in my data except tokenazation and conversion to numbers the text and sentiment got 40-38-40 then I tried in probabilities and the best was 90% with 40-39-40, I directly made almost all the basic reductions I thought need like removing @, # and the words after these symbols all links, all capitals and all tones and no stopwords and I got 37-39-40 the negative fell and the neutral rose with 100% elements, then I played with the stopwords one that the spacy library had and one of mine with corresponding data 39-38-41, 36-39-41 then I tried to do lemmetazition with spacy library respectively with stopwords of the library 38-39-39, and finally I played with the percentages again to see if there was any change with the spacy stopwords plus the changes in the dataset and I found 90% with 38-39-41 has dropped negative but the possitive has gone up, one last thing was to combine the parties and the sentiments in order to get better results but it came out 31.8-34.3-30.3, general I noticed that I have an underfill in almost all the examples I had because it does not have enough examples to be able to make a decision on whether they are positive negative and neutral or the tables below and the plots will be in the order I did the experiments ((1)brute force 100%,(2)brute force 90%,(3)dataset customization no stop words ,(4)dataset customization spacy stopwords,(5)dataset customization custom stopwords,(6)dataset customization spacy library , (7) 90% spacy stopwords dataset customization,(8)party+sentiment) Comments [initially I think it dropped when I put restrictions because many tweets only had useless information whenever the number dropped and since we have a general underfill it hits its efficiency this was also done with the custom stopwords because I was producing so many that it reduced its number, the spacy library respectively plus had another disadvantage that it took much longer, I think that if I had more to test my algorithms would have had a better result.]

Trial	NEGATIVE	NEUTRAL	POSITIVE	Final Score
1	40%	38%	40%	39%
2	40%	39%	40%	39%
3	37%	39%	40%	39%
4	39%	38%	41%	39%
5	36%	39%	41%	38%
6	38%	38%	39%	38%
7	38%	39%	41%	39%
8	31.8%	34.3%	30.3%	34%

Table 1: Trials

3.1.1. Table of trials.

3.2. Hyper-parameter tuning

I saw in general that whatever strategy I tried I had a general an underfill because I think we had a small dataset for the work we have to do

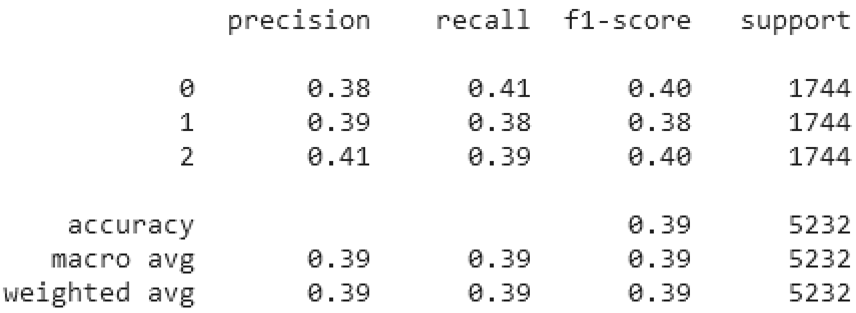
3.3. Optimization techniques

To find the best prices for her I did it manually by hand and didn't use any ready-made algorithm, but I should because it would help me save time

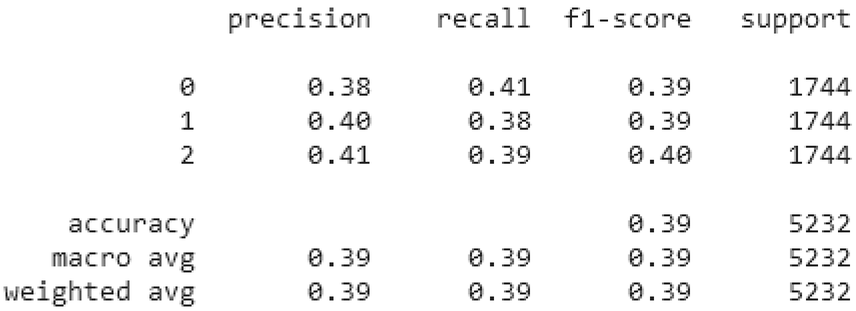
3.4. Evaluation

General enough, I believe it is in a reasonable object with the possibility of tools that we have and the number of tweets that it is a little higher than the randomness of 33. 3% but if I found better solutions I could take it up to 41-42 but I believe it will not happen with the techniques she knows and dataset number more than 50% (but I don't know, I guess because of the underfill)
((1)brute force 100%,(2)brute force 90%,(3) dataset customization spacy stopwords 100%)
(the rest are in the folder with the photo, I did this so that the pdf is not too big)

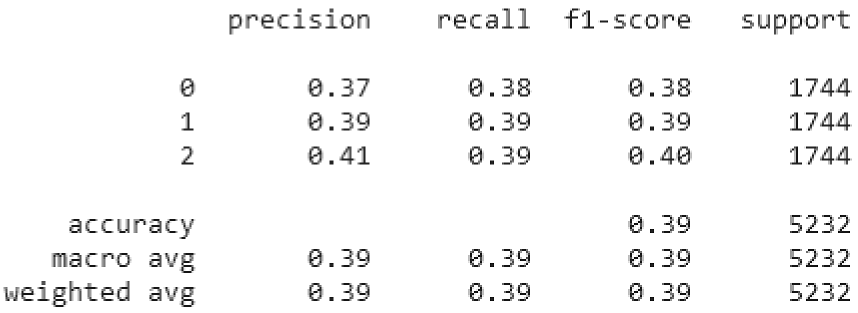
3.4.1. F1 SCORE.



```
              precision    recall  f1-score   support

           0       0.38      0.41      0.40      1744
           1       0.39      0.38      0.38      1744
           2       0.41      0.39      0.40      1744

    accuracy                           0.39      5232
   macro avg       0.39      0.39      0.39      5232
weighted avg       0.39      0.39      0.39      5232
```

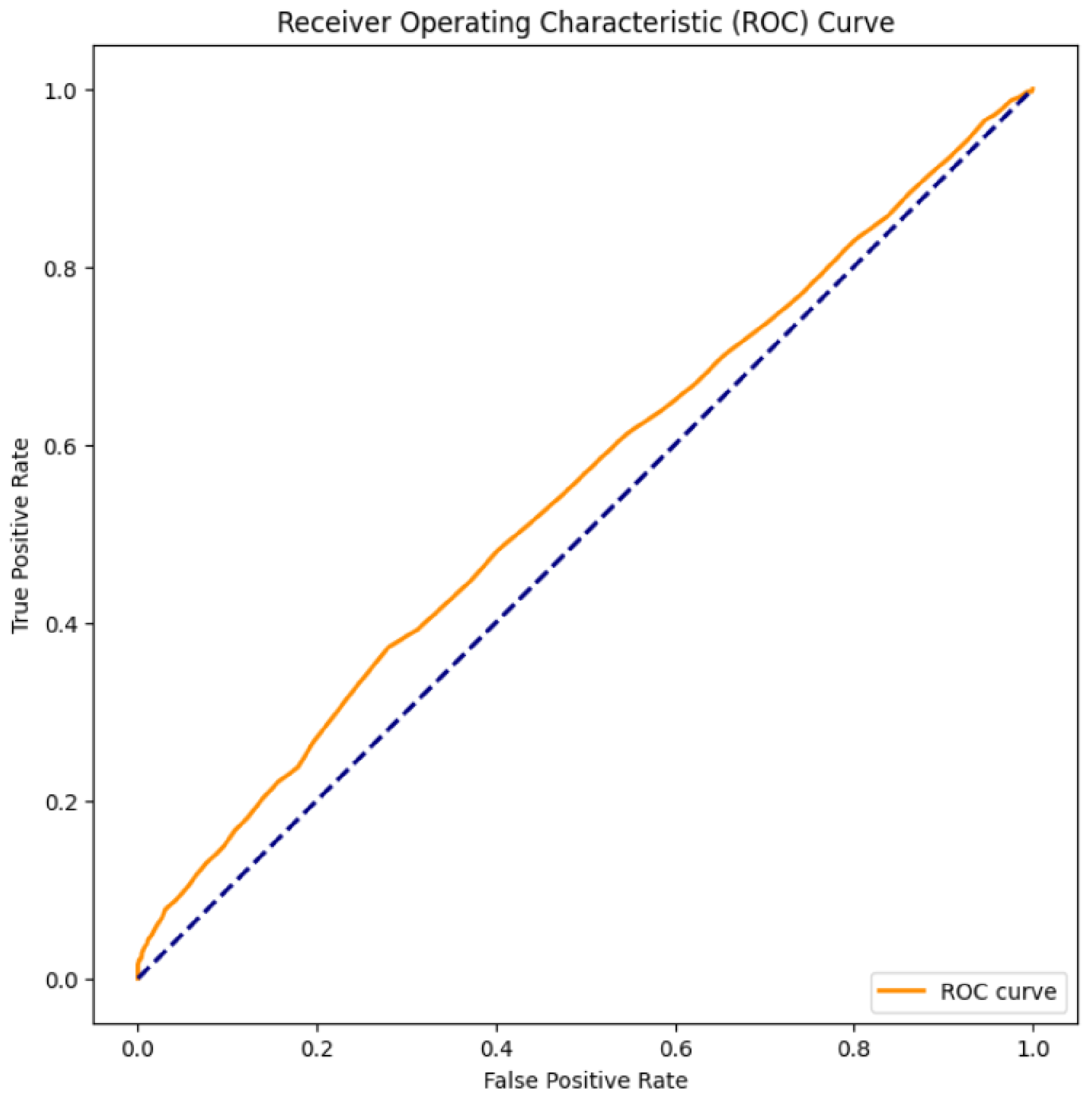


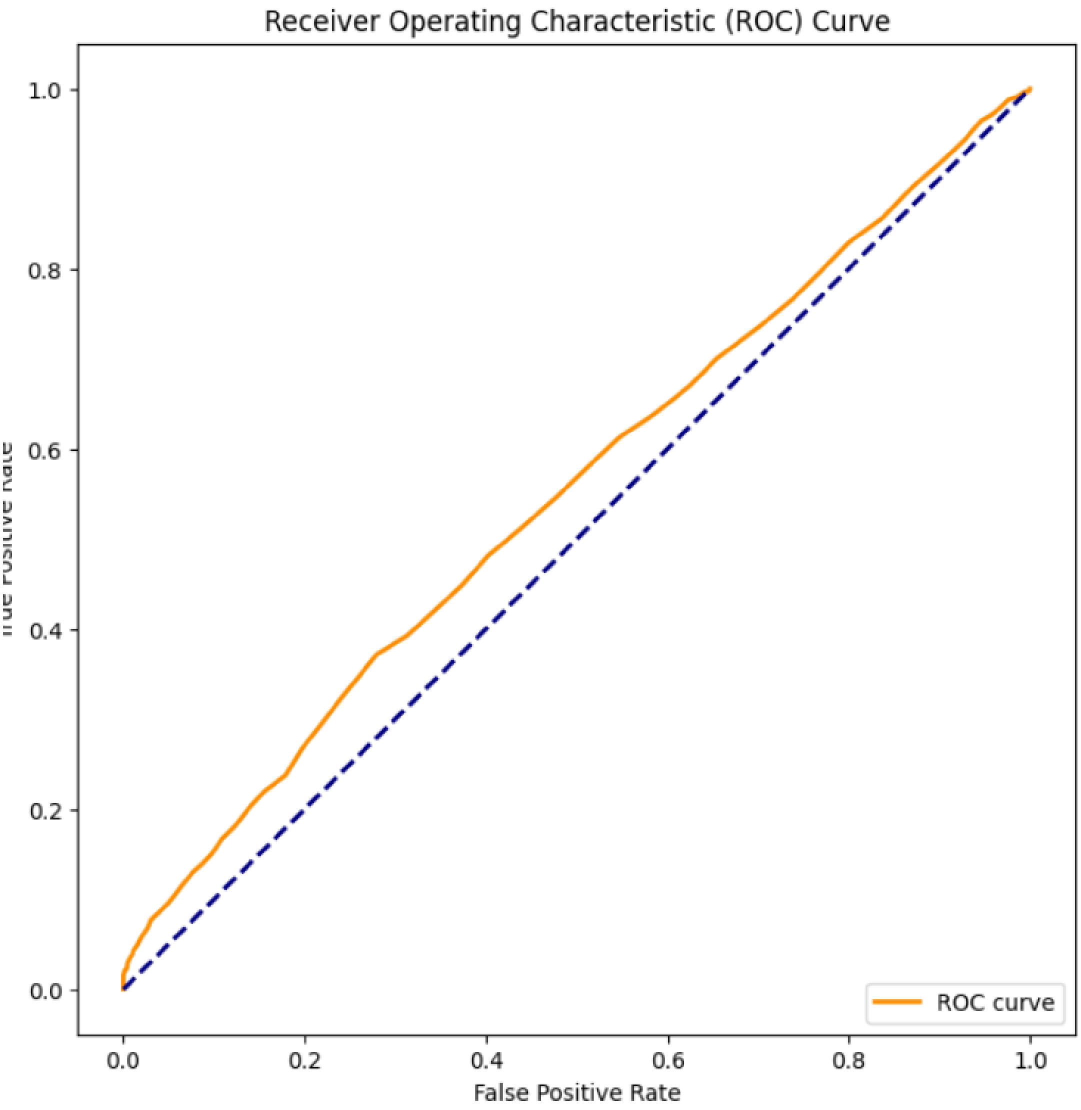
```
              precision    recall  f1-score   support

           0       0.38      0.41      0.39      1744
           1       0.40      0.38      0.39      1744
           2       0.41      0.39      0.40      1744

    accuracy                           0.39      5232
   macro avg       0.39      0.39      0.39      5232
weighted avg       0.39      0.39      0.39      5232
```

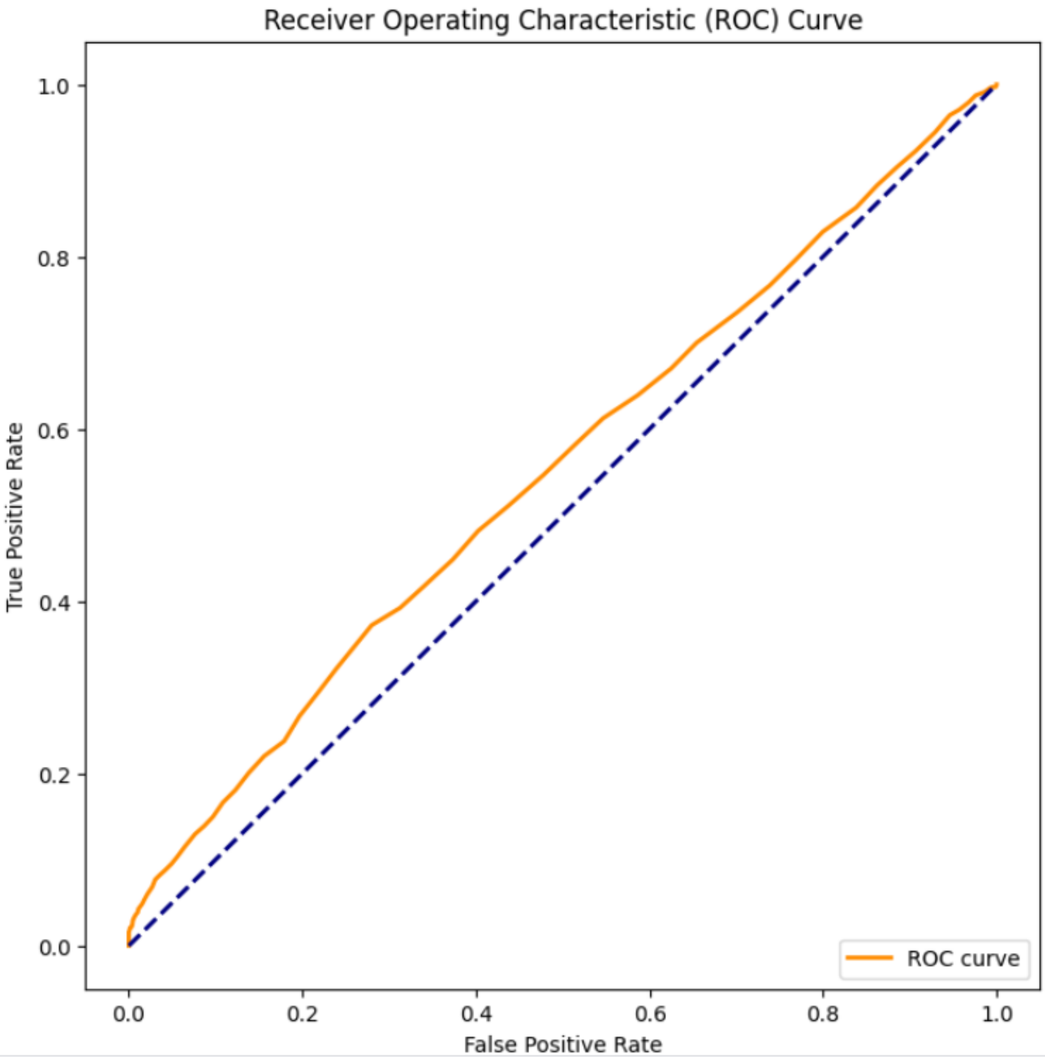


```
              precision    recall  f1-score   support

           0       0.37      0.38      0.38      1744
           1       0.39      0.39      0.39      1744
           2       0.41      0.39      0.40      1744

    accuracy                           0.39      5232
   macro avg       0.39      0.39      0.39      5232
weighted avg       0.39      0.39      0.39      5232
```

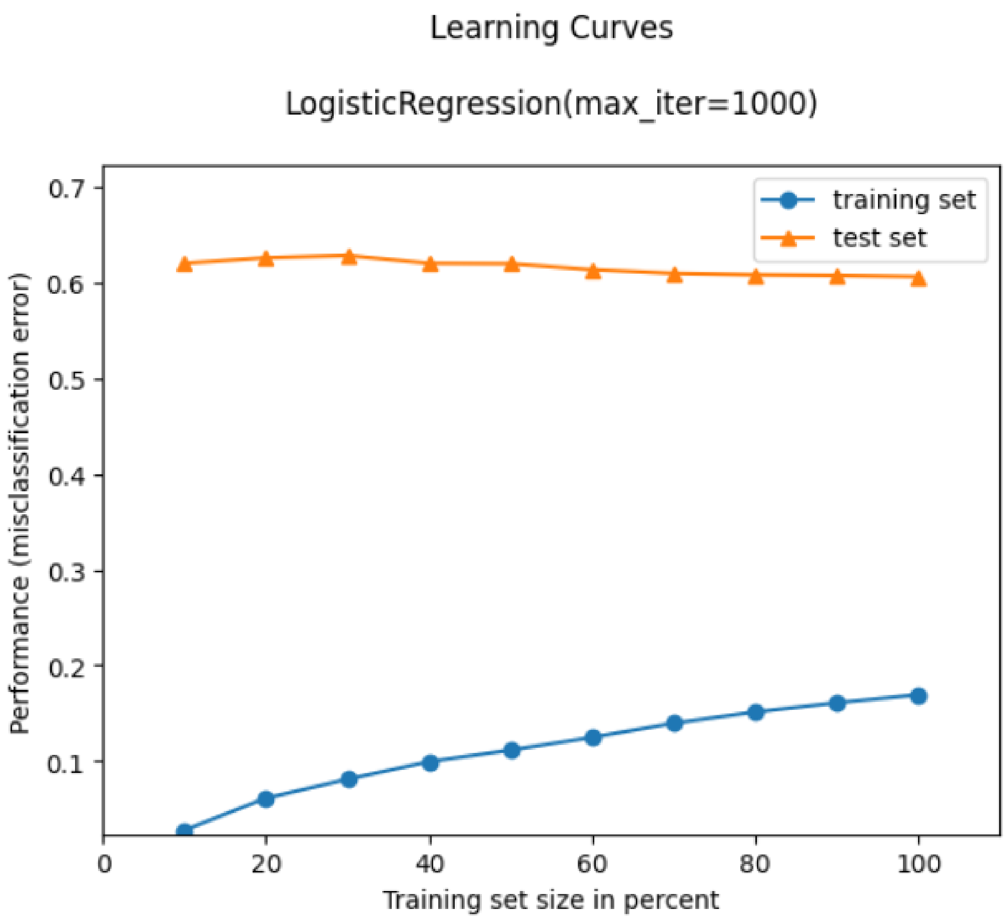
3.4.2. ROC curve.

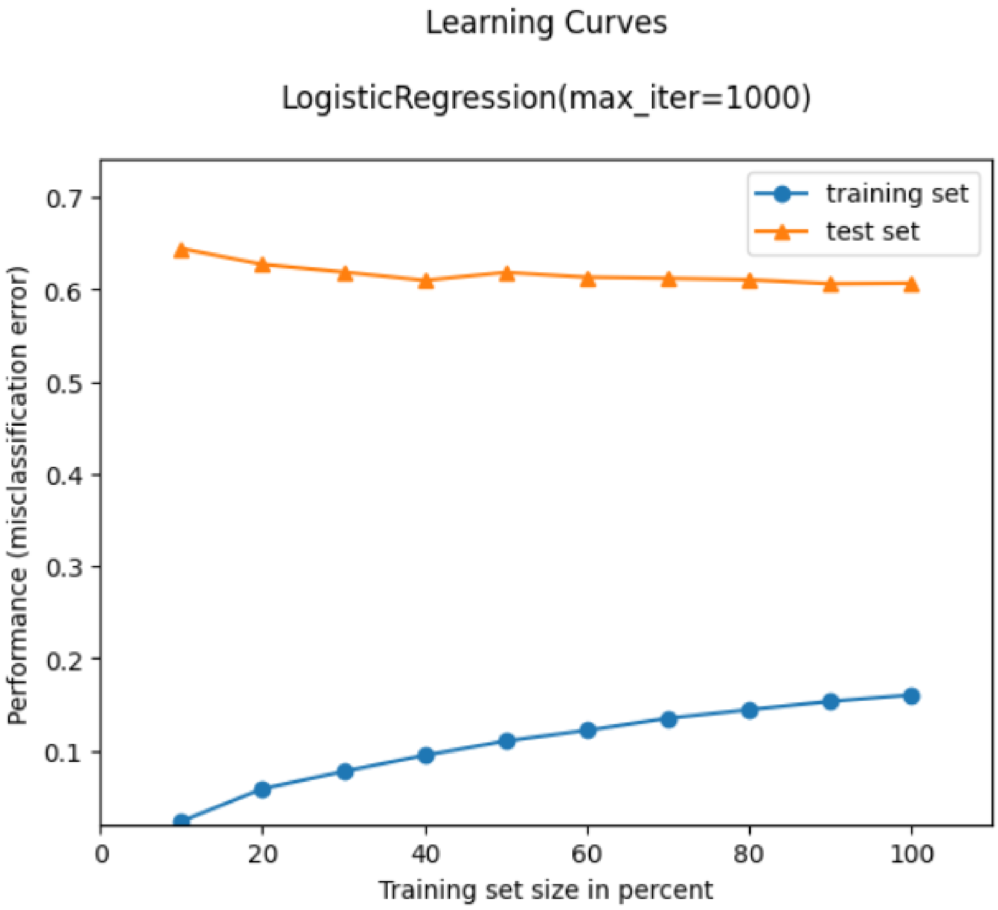

Receiver Operating Characteristic (ROC) Curve
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve

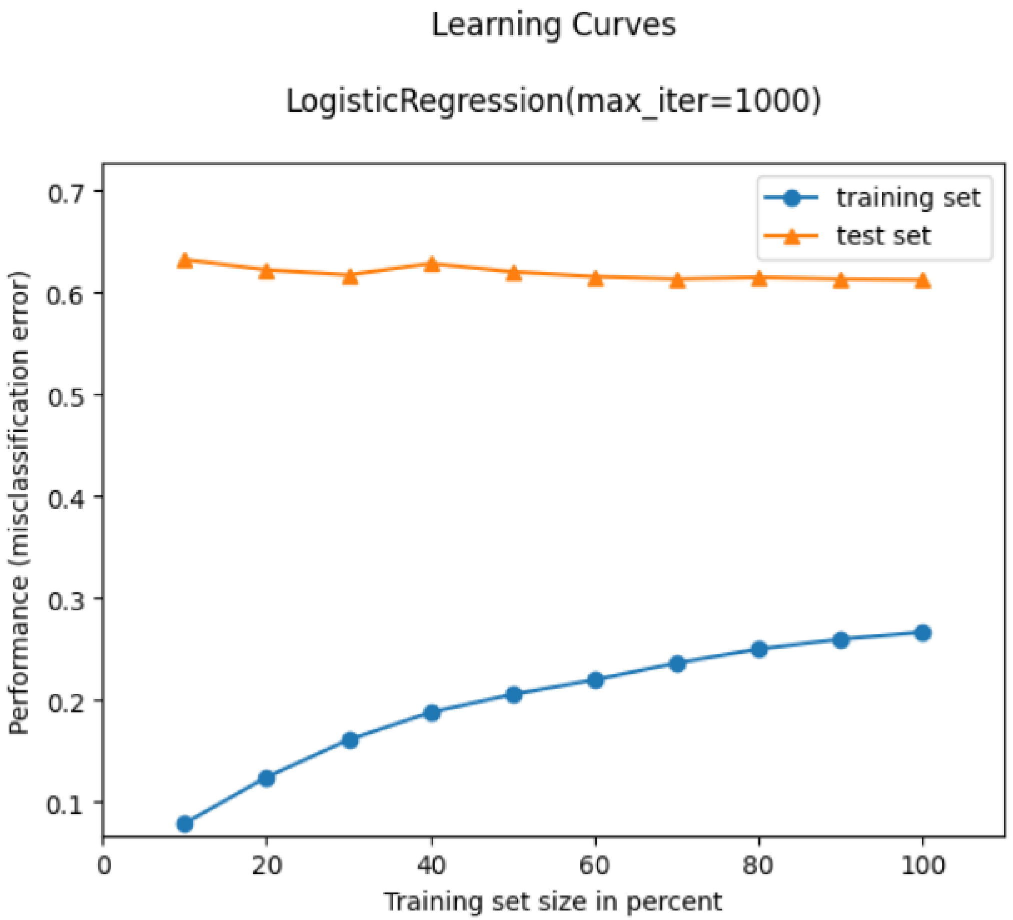

Receiver Operating Characteristic (ROC) Curve
True Positive Rate
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
ROC curve



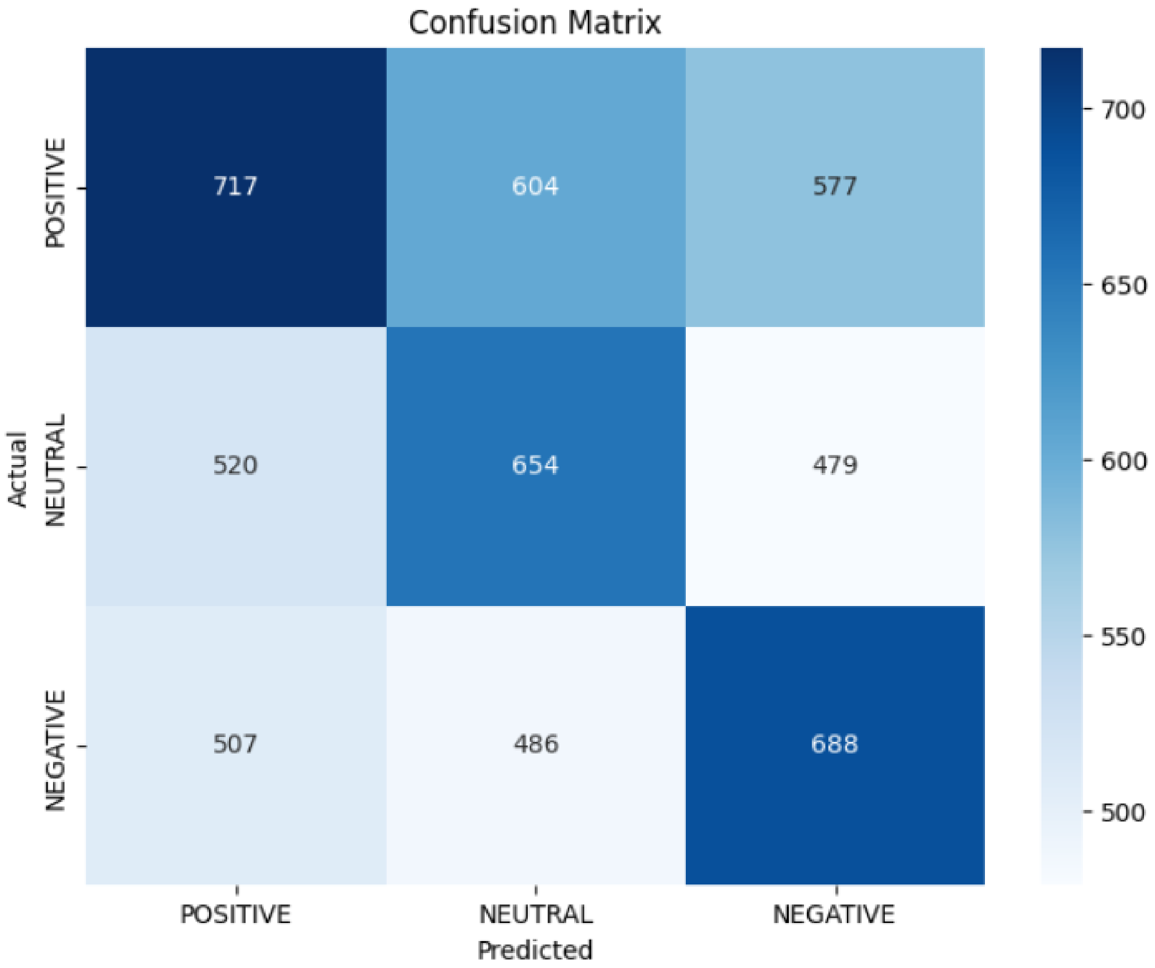
3.4.3. Learning Curve.

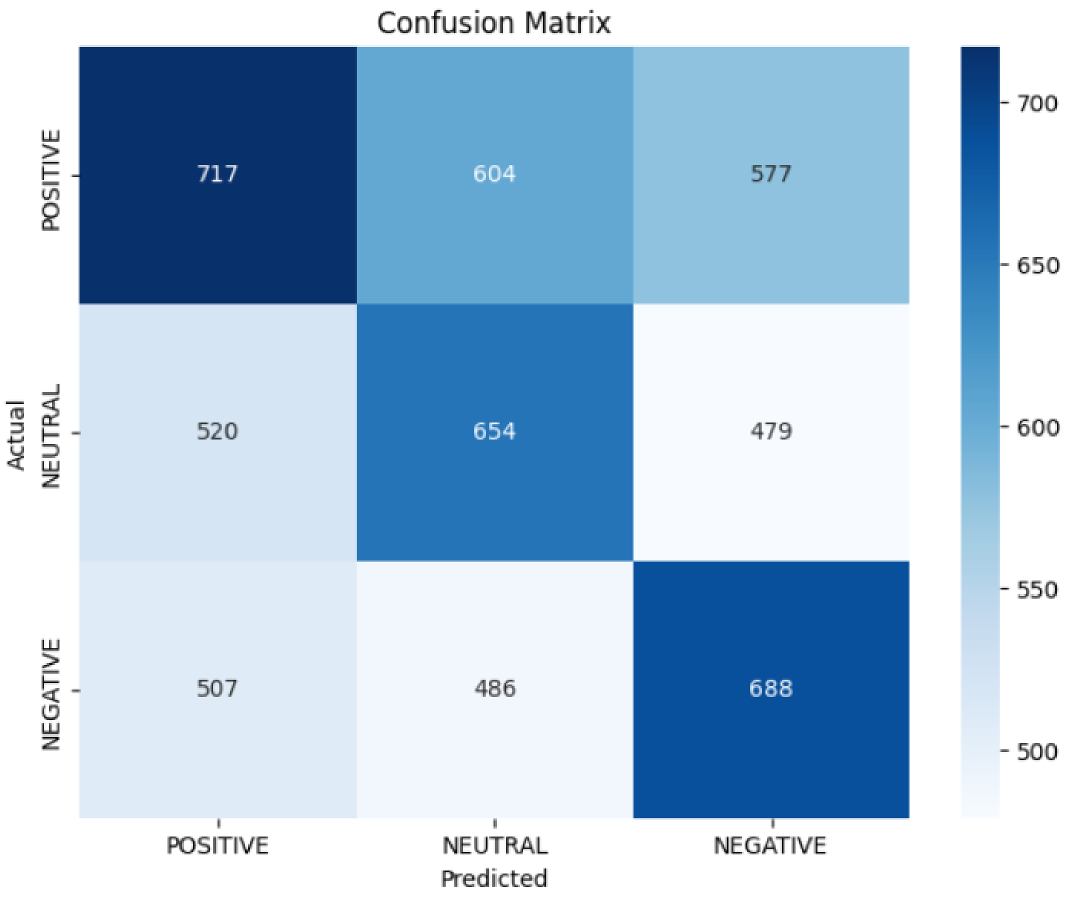

Learning Curves
LogisticRegression(max_iter=1000)
Performance (misclassification error)
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
20
40
60
80
100
Training set size in percent
training set
test set

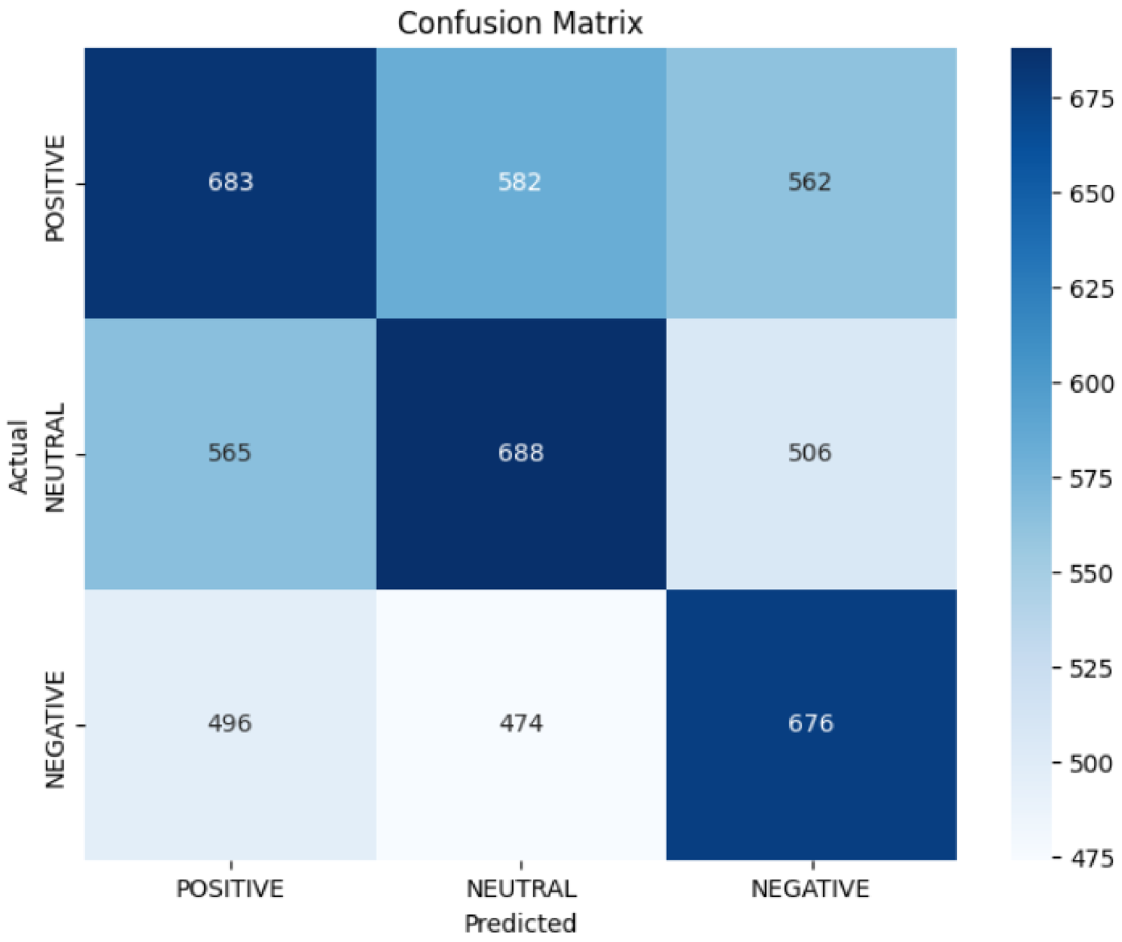

Learning Curves
LogisticRegression(max_iter=1000)
training set
test set
Performance (misclassification error)
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
20
40
60
80
100
Training set size in percent



3.4.4. Confusion matrix.


Confusion Matrix
Actual
POSITIVE
NEUTRAL
NEGATIVE
717
604
577
520
654
479
507
486
688
POSITIVE
NEUTRAL
NEGATIVE
Predicted
700
650
600
550
500


Confusion Matrix
Actual
POSITIVE
NEUTRAL
NEGATIVE
717
604
577
520
654
479
507
486
688
POSITIVE
NEUTRAL
NEGATIVE
Predicted
700
650
600
550
500



4. Results and Overall Analysis

4.1. Results Analysis

The results I have are predicted of course they can go up a bit to 42-43% type but I don't know what I can do because if other words are mentioned or modified I don't see a change, plus the small amount of data we have it is logical that it is not satisfactory the percentage but expected, the only thing I would like to try but I didn't find a good way to combine sentiment with the parties I believe would increase her chances but I didn't find a good way that gives good results

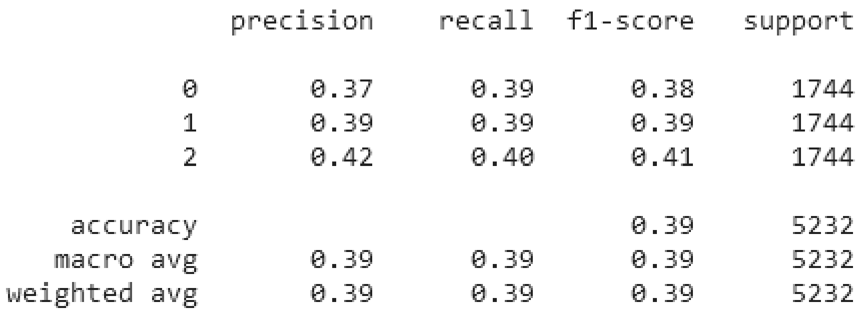
4.1.1. Best trial. (7) 90% spacy stopwords dataset customization

Trial				Score
7	38%	39%	41%	39%

Table 2: Trials

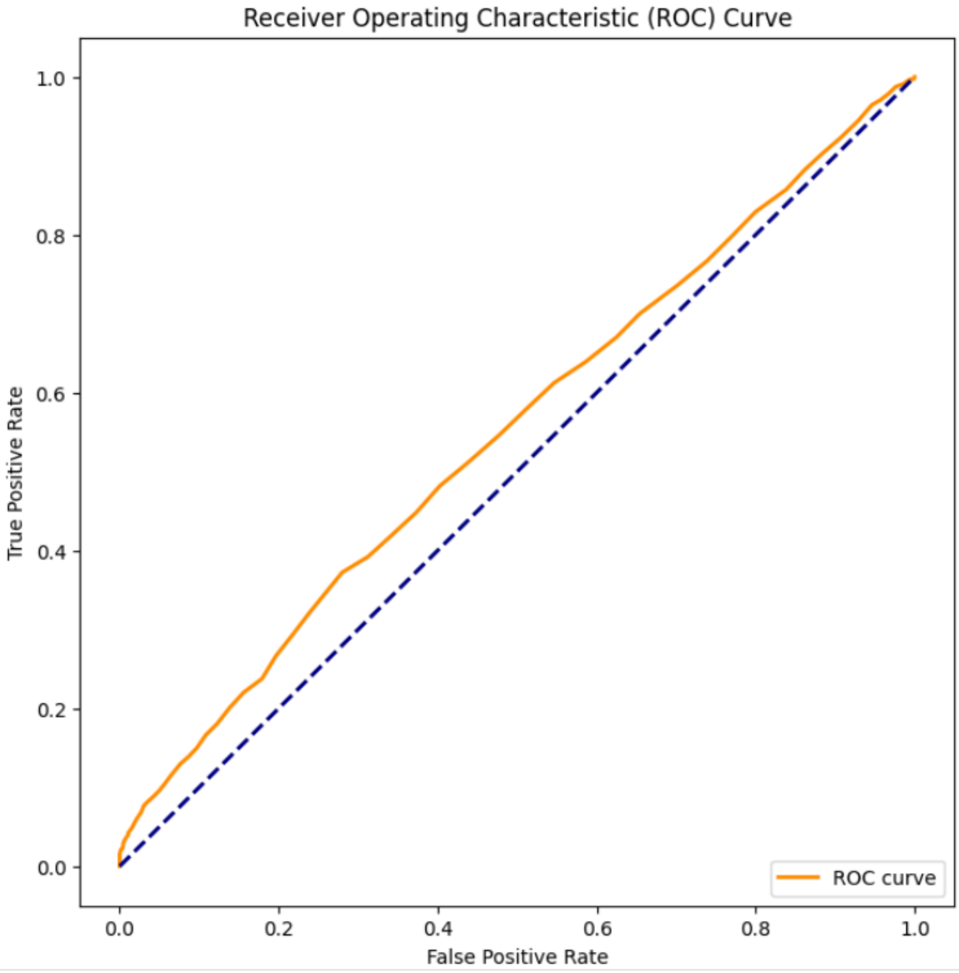
The plots are:

4.1.2. F1 SCORE.

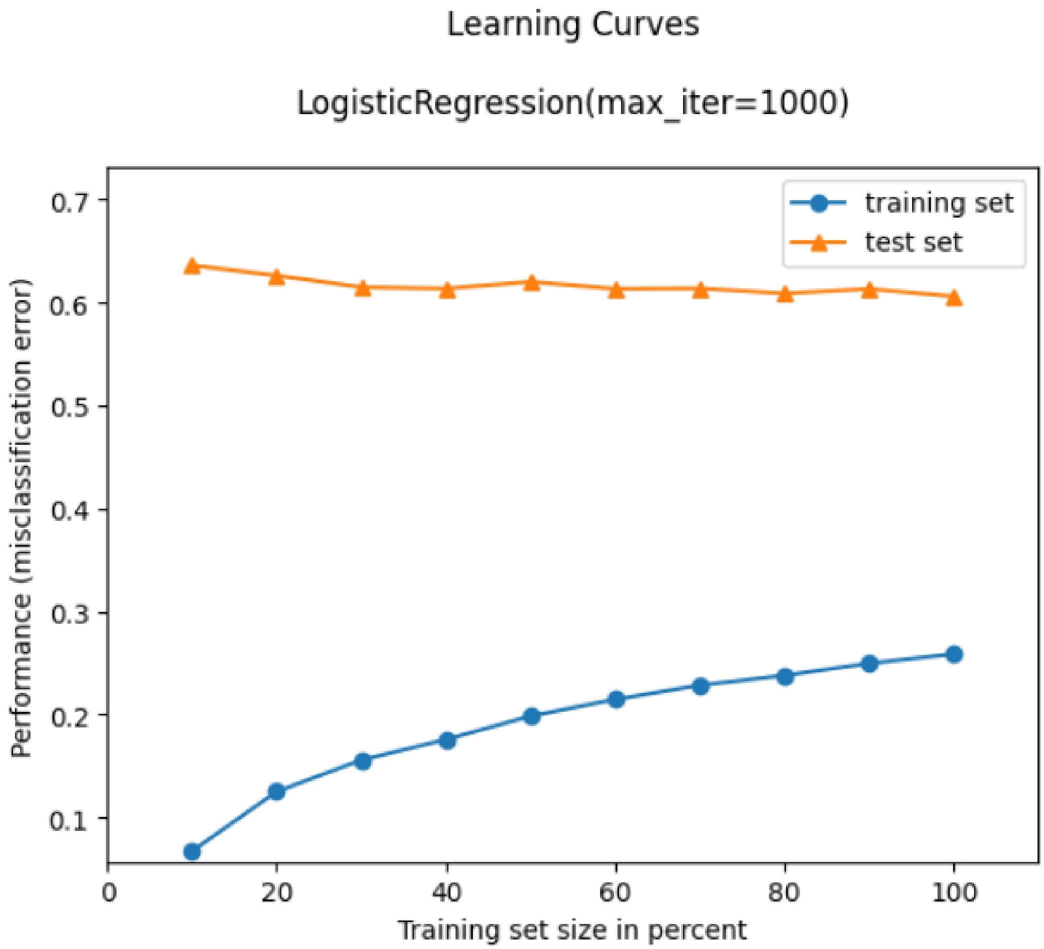


```
              precision    recall  f1-score   support

           0       0.37      0.39      0.38      1744
           1       0.39      0.39      0.39      1744
           2       0.42      0.40      0.41      1744

    accuracy                           0.39      5232
   macro avg       0.39      0.39      0.39      5232
weighted avg       0.39      0.39      0.39      5232
```

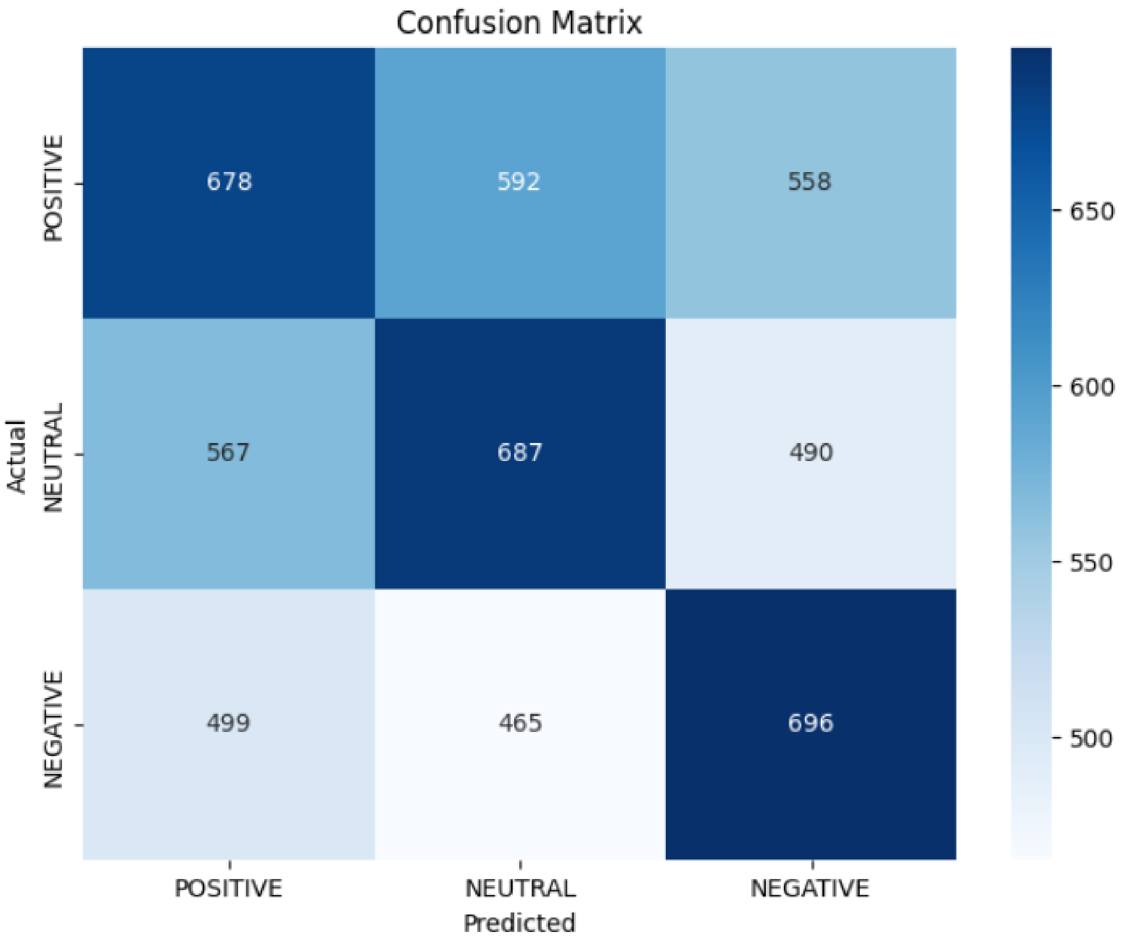
4.1.3. ROC curve.



4.1.4. Learning Curve.



4.1.5. Confusion matrix.



4.2. Comparison with the first project

<Use only for projects 2,3,4>
<Comment the results. Why the results are better/worse/the same?>

4.3. Comparison with the second project

<Use only for projects 3,4>
<Comment the results. Why the results are better/worse/the same?>

4.4. Comparison with the third project

<Use only for project 4>
<Comment the results. Why the results are better/worse/the same?>

5. Bibliography

References

ROC Curve Example , Confusion Matrix Example , Tokenization in Spacy, Language Processing Pipeline in Spacy , Text Representation Using TF-IDF