UNIVERSITY OF ATHENS
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

Deep Learning for NLP

Student name: *Kleopatra Karapanagiotou*
sdi: lt12200010

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 **2**
3.1 Experiments 2
3.1.1 Table of trials 2
3.2 Hyper-parameter tuning 3
3.3 Optimization techniques 3
3.4 Evaluation 3
3.4.1 ROC curve 3
3.4.2 Learning Curve 3
3.4.3 Confusion matrix 3

4 Results and Overall Analysis **3**
4.1 Results Analysis 3
4.1.1 Best trial 3
4.2 Comparison with the first project 3
4.3 Comparison with the second project 3
4.4 Comparison with the third project 4

5 Bibliography **4**

1. Abstract

The purpose of the assignment is to develop a sentiment classifier for a dataset containing Greek tweets about the Greek general elections. In the following paragraphs, I explain how I approached this task, in terms of preprocessing, feature engineering and choosing the hyperparameters for this multiclass classification task.

2. Data processing and analysis

2.1. Pre-processing

During cleaning the following were removed:

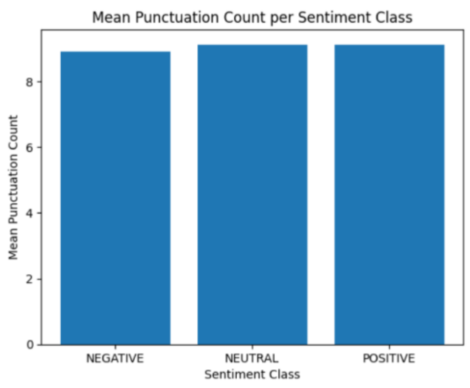
- Duplicates
- Null-values

During preprocessing the following were removed:

- Step 1: Common characters found only in tweet content: RT-tags (retweets), @unsernames, urls , html and http links.
- After step 1: the count of character length per tweet was calculated in order to be added as a feature in a new column (the column 'n_chars'). I considered this as the right part to capture the character length, in order to capture information from punctuation and hashtag comments.
- Step 2: Punctuation separation and removal: separation in order to remove the underscores from hashtags (συριζα_τελος.. etc), latin characters and number removal
- After step 2, I used 2 lists of negative and positive words in Greek, transformed them into a dictionary, by assigning them their polarity and used them in order to extract the count of positive and count of negative words found in each tweet. Images of the results per class will be showcased later in the report.
- Last steps: Removal of words with character length <1, due some single charcters found during visualisations (like: "κ" (common abbreviation of 'και' in Greek social media text)
- Removal of stopwords apart from those having negative commutation: "όχι, μην, δεν, πολύ, πολύ" and removal of emojis.

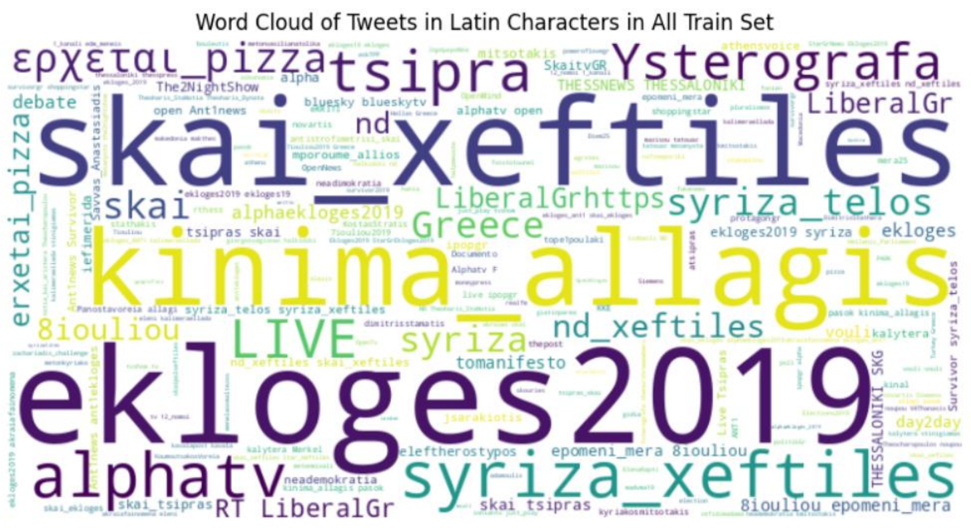
2.2. Analysis

In his section I elaborate on why I excluded and kept specific information from the tweets through visualizations and statistics

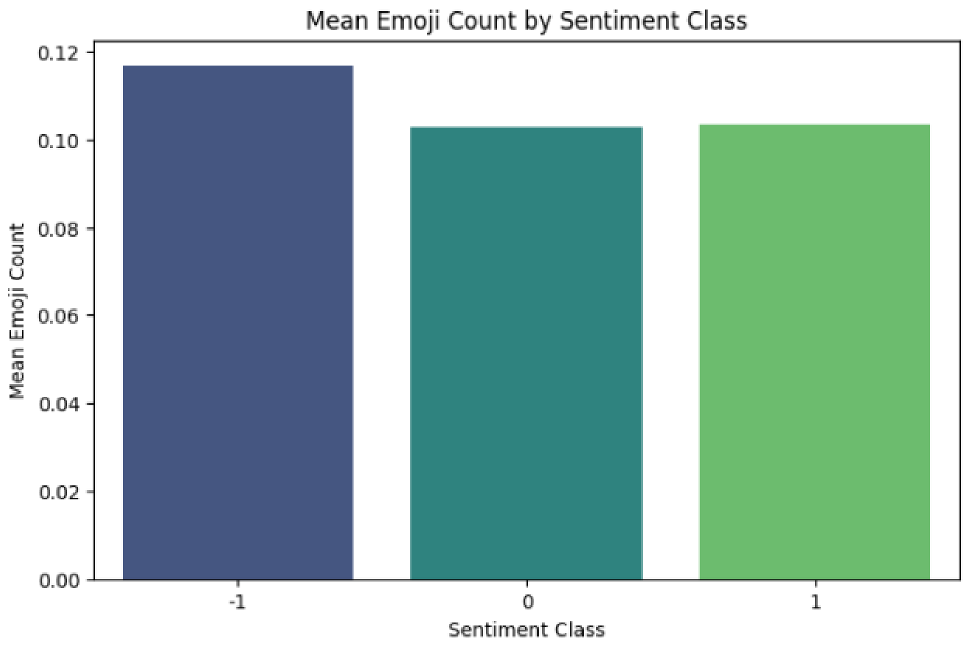


Count of Punctuation characters was one of my initial concerns in the corpus. But after seeing there is no significant difference in the mean punctuation count per sentiment class, I decided to remove them. Maybe it was a bad choice, because the test set will contain a lot of punctuation too, but hopefully the model will focus more on other features.

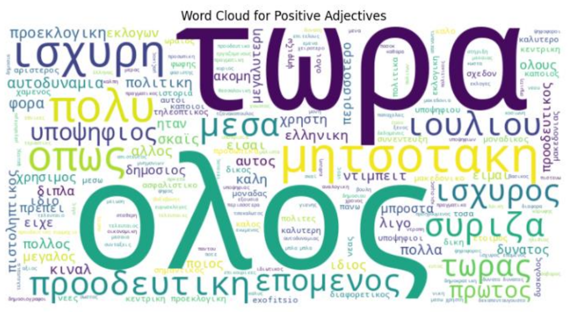
In the following wordcloud we see the content of hashtags with Latin characters only. This was also an import concern. As you can see hastags with the suffix "_xeftiles" are very common in our tweets. It would be a good idea to keep those tweets with such suffix. I did not do it this time. I wanted to focus only in Greek content. Maybe this was a bad idea too.

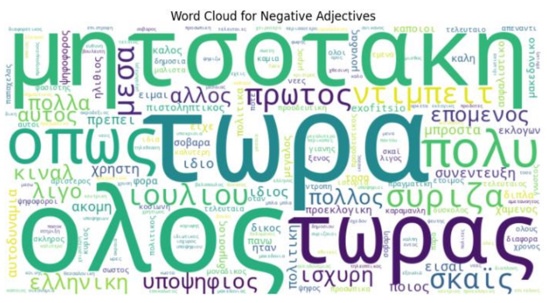


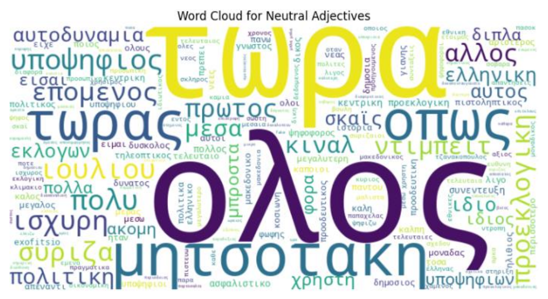
For elaborating the removal of emojis, as you can see there is no significant difference in the mean count of tweets per class. It was not assessed whether this difference is statistically significant or not. This was my personal assumption.

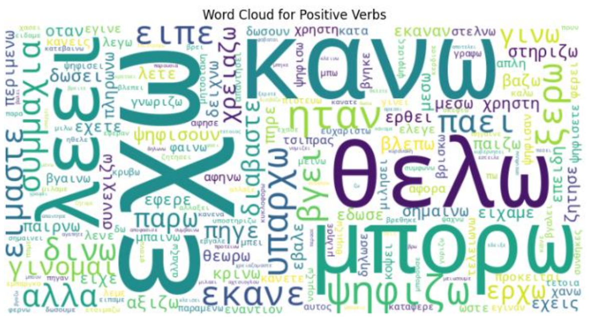


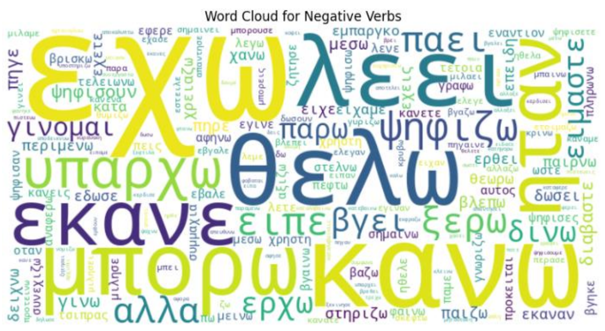
Inspecting the power of POS-tagging with the Greek spacy (large model). After assigning POS tags with the help of spacy library, it was a safe step to visualize the results for each tag I was interested in: Adjectives, Verbs and Nouns

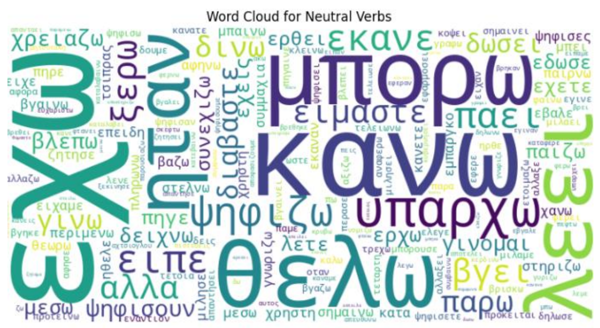


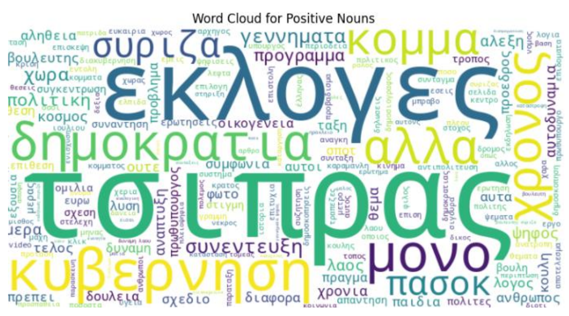


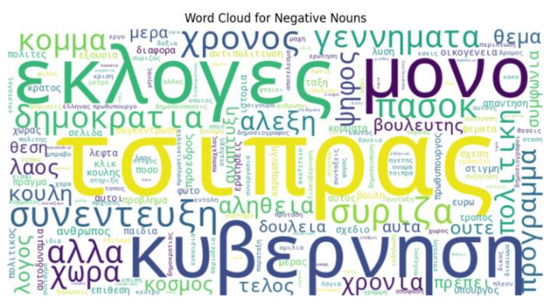


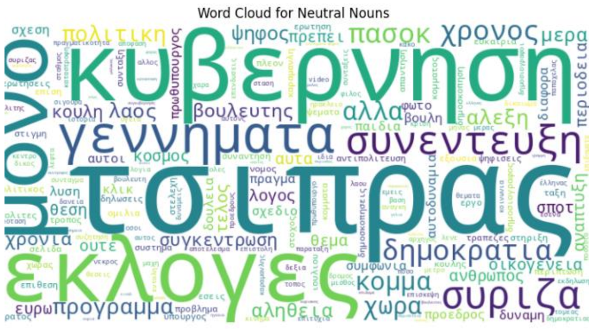




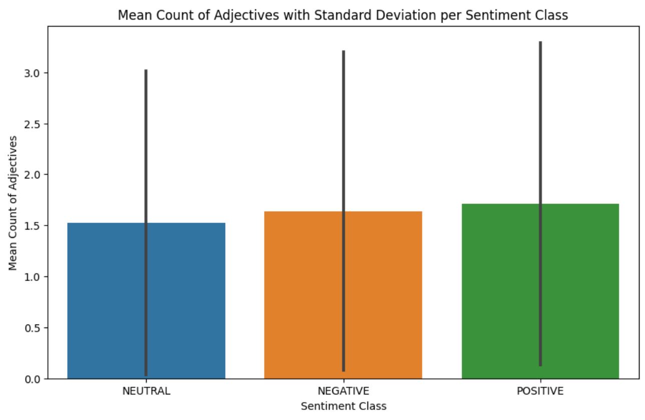




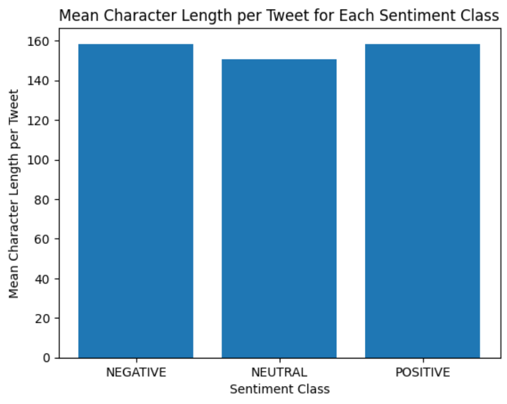


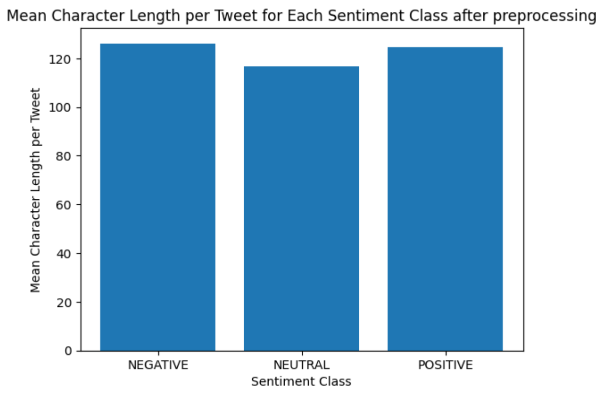


I decided not to include information about POS tags in my features, since I observe inconsistencies and a lot of false tagging. Maybe the corpus needs more cleaning first, before POS tagging. Also Named Entity Recognition would maybe be helpful here, to exclude named entities from the corpus. The following plot indicates the mean and standard deviation count of adjectives per class. We see that all three classes have a high variance, meaning that the datapoints of ADJ tags are very spread out from the mean, meaning that the mean count of adjectives per class is not a reliable and representative metric to include in the features.



Mean character length per tweet (after username, rt-tag and links removal)





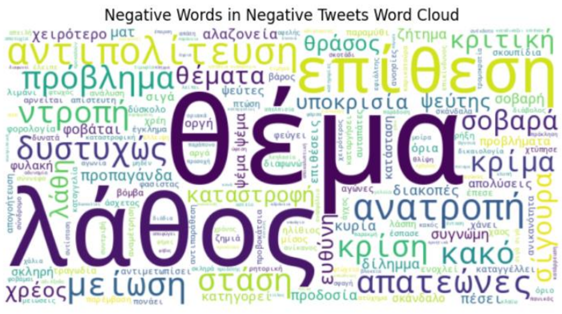
Performed Tukey test for multiple comparisons of means and got the following results (after preprocessing and outlier removal)

group1	group2	meandiff	p-adj	lower	upper	upper
NEGATIVE	NEUTRAL	-9.5799	0.0	-11.7271	-7.4326	True
NEGATIVE	POSITIVE	-1.4327	0.2614	-3.58	0.7146	False
NEUTRAL	POSITIVE	8.1471	0.0	5.9999	10.2944	True

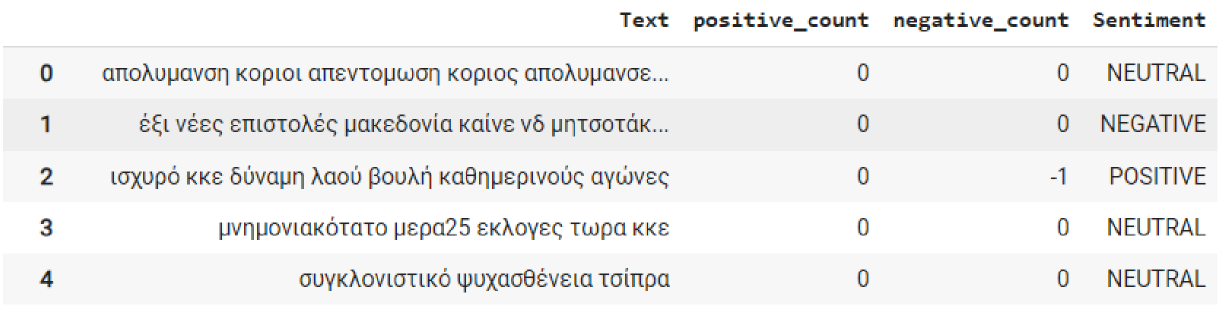
Based on the results we see that at least one pair of groups differ significantly, so we can keep the mean character length per tweet as a feature.

Positive and Negative lexicon words (further info here: https://github.com/NKryst/Greek-Sentiment-Analysis/tree/master/Files/Greek%20Sentiment%20Lexicon)

Positive Words in Positive Tweets Word Cloud
πρωθυπουργός
πολύ
συμφωνία
μπροστά
δουλειά
καλύτερα
νίκη
μεγαλύτερη
όνομα
επιτυχία



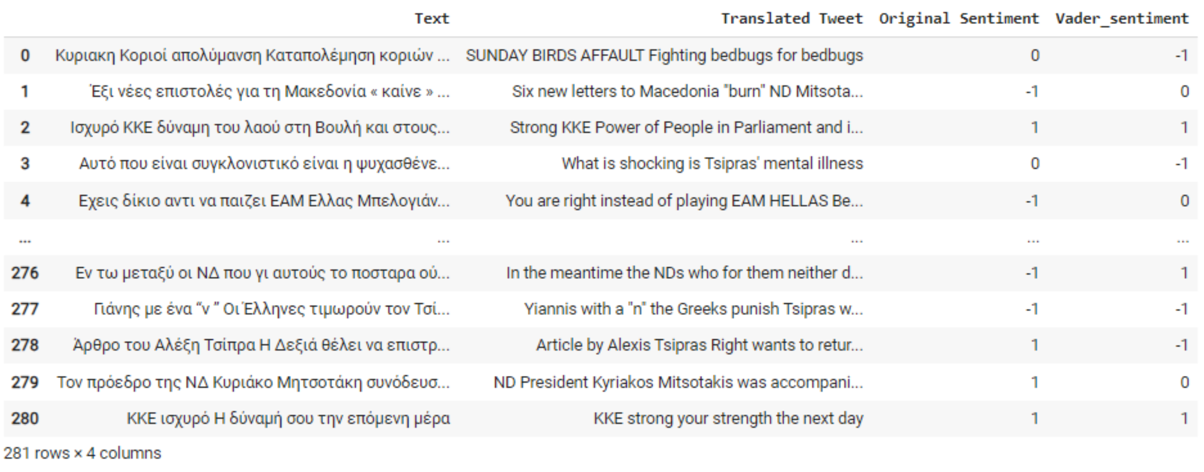
One limitation to note after adding the counts of positive and negative words per tweet as a feature is that a lot of words of the two dictionaries were lemmatized, but my corpus wasn't. Looking at some examples on the corpus:



	Text	positive_count	negative_count	Sentiment
0	απολυμανση κοριοι απεντομωση κοριος απολυμανσε...	0	0	NEUTRAL
1	έξι νέες επιστολές μακεδονία καίνε νδ μητσοτάκ...	0	0	NEGATIVE
2	ισχυρό κκε δύναμη λαού βουλή καθημερινούς αγώνες	0	-1	POSITIVE
3	μνημονιακότατο μερα25 εκλογες τωρα κκε	0	0	NEUTRAL
4	συγκλονιστικό ψυχασθένεια τσίπρα	0	0	NEUTRAL

On the 3rd row, we see the tweet considering only the word «αγώνες» included in the negative lexicon, but not the word «ισχυρό» (=strong). That happened because in the positive word dictionary the word "strong" is included with another form «ισχυρός», which is the lemmatized form for the Adjective. It is interesting though that the word «αγώνες» is included in the lexicon with both the singular and the plural form («αγώνας», «αγώνες»). By these small findings I think it would be reasonable to first lemmatize the corpus and then count the positive and negative words of the dictionaries. Also in the tweets we see a lot of use of metaphoric speech, like the verb "burn" «καίνε».

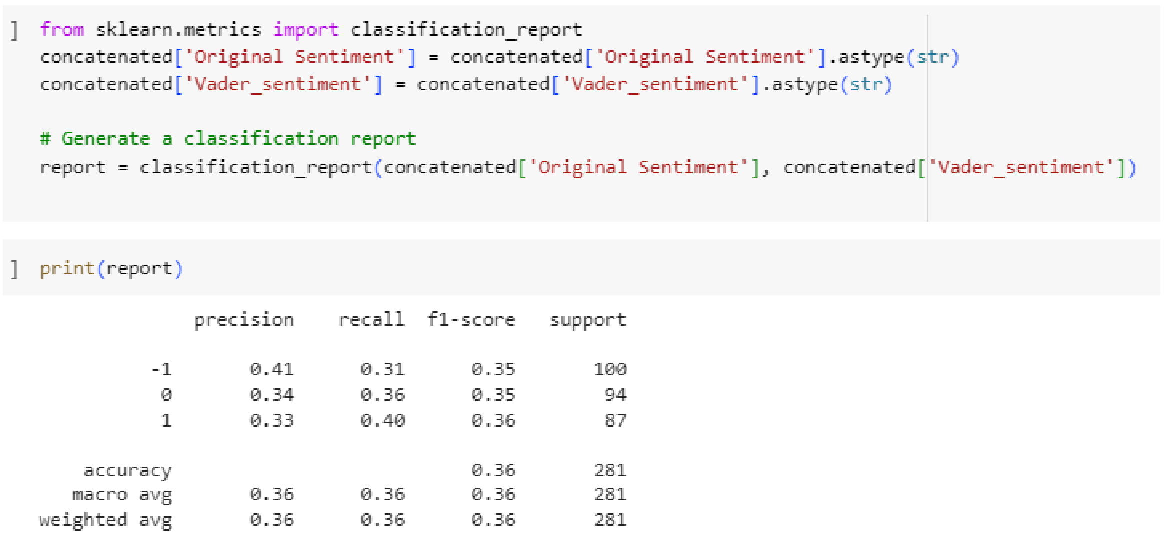
Another aspect which was considered but not thoroughly investigated (due to lack of computational power and RAM space), was the use of VaderSentiment logits after translating the tweets in English with the help of googletrans library (https://pypi.org/project/googletrans/).
In the following dataframe we see some samples from this effort.



	Text	Translated Tweet	Original Sentiment	Vader_sentiment
0	Κυριακη Κοριοί απολύμανση Καταπολέμηση κοριών ...	SUNDAY BIRDS AFFAULT Fighting bedbugs for bedbugs	0	-1
1	Έξι νέες επιστολές για τη Μακεδονία « καίνε » ...	Six new letters to Macedonia "burn" ND Mitsota...	-1	0
2	Ισχυρό ΚΚΕ δύναμη του λαού στη Βουλή και στους...	Strong KKE Power of People in Parliament and i...	1	1
3	Αυτό που είναι συγκλονιστικό είναι η ψυχασθένε...	What is shocking is Tsipras' mental illness	0	-1
4	Εχεις δίκιο αντι να παιζει ΕΑΜ Ελλας Μπελογιάν...	You are right instead of playing EAM HELLAS Be...	-1	0
...	...	...	...	...
276	Εν τω μεταξύ οι ΝΔ που γι αυτούς το ποσταρα ού...	In the meantime the NDs who for them neither d...	-1	1
277	Γιάνης με ένα "ν " Οι Έλληνες τιμωρούν τον Τσί...	Yiannis with a "n" the Greeks punish Tsipras w...	-1	-1
278	Άρθρο του Αλέξη Τσίπρα Η Δεξιά θέλει να επιστρ...	Article by Alexis Tsipras Right wants to retur...	1	-1
279	Τον πρόεδρο της ΝΔ Κυριάκο Μητσοτάκη συνόδευσ...	ND President Kyriakos Mitsotakis was accompani...	1	0
280	ΚΚΕ ισχυρό Η δύναμή σου την επόμενη μέρα	KKE strong your strength the next day	1	1

281 rows × 4 columns

It is worth noting though that through this process a lot of inconsistencies not only in the predicted vader logits, but also in the dataset itself were observed. But out of pure curiosity I wanted to see an evaluation on these 281 first rows. (not bad compared to what's coming)



```
] from sklearn.metrics import classification_report
  concatenated['Original Sentiment'] = concatenated['Original Sentiment'].astype(str)
  concatenated['Vader_sentiment'] = concatenated['Vader_sentiment'].astype(str)

  # Generate a classification report
  report = classification_report(concatenated['Original Sentiment'], concatenated['Vader_sentiment'])
```

```
] print(report)
```

```
              precision    recall  f1-score   support

          -1       0.41      0.31      0.35       100
           0       0.34      0.36      0.35        94
           1       0.33      0.40      0.36        87

    accuracy                           0.36       281
   macro avg       0.36      0.36      0.36       281
weighted avg       0.36      0.36      0.36       281
```

Long story short the features kept as predictors are the following:

1. the vectorised and preprocessed text (tfidf/countvectorizer)
2. Scaled count of positive words (multiplied by the sign of the polarity : so for the count of 3 positive words the count will be (+)3 and for the count of negative words in a negative sentiment sentence the count will be -3
3. Scaled count of negative words
4. Scaled count of characters per tweet.

Textual and numerical values were combined with np.hstack()

2.3. Data partitioning for train, test and validation

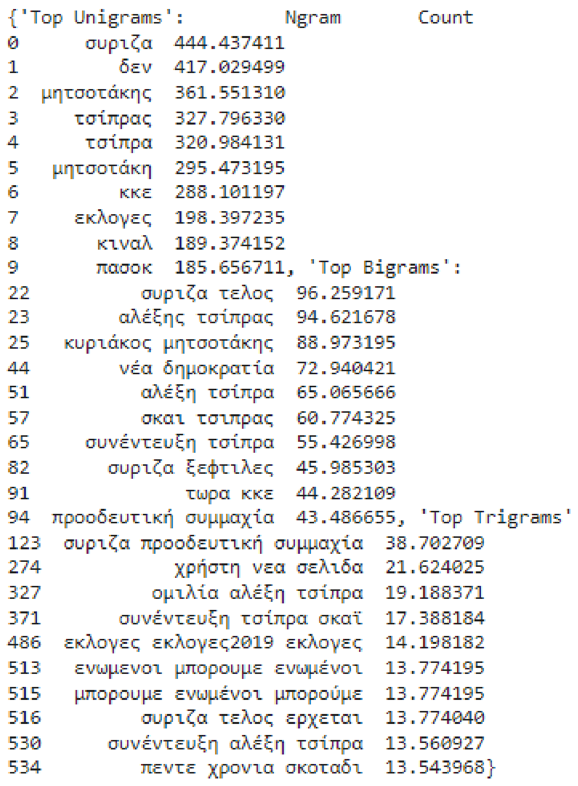
For data partitioning the common ratios of 80/20 were applied. Train and validation data were both splitted into (train/test and dev/test accordingly)
Train set: training ,hyperparameter tuning with GridSearchCV, one first evaluation with the common metrics on the test set of the train set (accuracy,precision, recall ,f1)
Validation set: served as the test set in my experiments. (fit the final trained model and showcased evaluation results: metrics, learning curves, confusion matrices, classification reports)

Test set: Used to predict the missing polarity column 'Predicted'.

2.4. Vectorization

TfIdf Vectorizer (word importance): vectorizes a word based on how important that word is within a tweet relative to our dataset. A high score in tf–idf means high word frequency (in the given tweet) and a low document frequency of the word in the whole dataset.

Here are the top uni, bi and trigrams with tfidf



```
{'Top Unigrams':         Ngram      Count
0      συριζα  444.437411
1         δεν  417.029499
2  μητσοτάκης  361.551310
3     τσίπρας  327.796330
4      τσίπρα  320.984131
5   μητσοτάκη  295.473195
6         κκε  288.101197
7     εκλογες  198.397235
8       κιναλ  189.374152
9       πασοκ  185.656711, 'Top Bigrams':
22          συριζα τελος  96.259171
23        αλέξης τσίπρας  94.621678
25   κυριάκος μητσοτάκης  88.973195
44        νέα δημοκρατία  72.940421
51          αλέξη τσίπρα  65.065666
57          σκαι τσιπρας  60.774325
65     συνέντευξη τσίπρα  55.426998
82       συριζα ξεφτιλες  45.985303
91              τωρα κκε  44.282109
94  προοδευτική συμμαχία  43.486655, 'Top Trigrams'
123  συριζα προοδευτική συμμαχία  38.702709
274             χρήστη νεα σελιδα  21.624025
327            ομιλία αλέξη τσίπρα  19.188371
371        συνέντευξη τσίπρα σκαϊ  17.388184
486  εκλογες εκλογες2019 εκλογες  14.198182
513   ενωμενοι μπορουμε ενωμένοι  13.774195
515   μπορουμε ενωμένοι μπορούμε  13.774195
516          συριζα τελος ερχεται  13.774040
530      συνέντευξη αλέξη τσίπρα  13.560927
534          πεντε χρονια σκοταδι  13.543968}
```

Count Vectorizer (BOW-approach): vectorizes a word based on the count of occurrences of this word in the whole dataset. A similar table of uni , bi ,and trigrams was generated for the Countvectorizer (the results were almost identical).

For the vectorizers' hyperparameters, I chose to work with defaults, except for the following: max_feautures=1000, set the limit of text features vectors to 1000 due to computational load.
For tokenization, no tokenizer was specified, because I saw that the default of the analyzer is 'word', so n-grams were generated based on words.

3. Algorithms and Experiments

3.1. Experiments

My initial approach was to experiment and compare my results with the 2 popular vectorizers, tfidf and countvectorizer. This idea came after the long preprocessing of the Text column which left me with a lot of tweets containing meaningful unigrams, bigrams and trigrams. Especially some of the most frequent bigrams and trigrams in the vectorization part were: 'σύριζα ξεφτίλες, σύριζα τέλος, τωρα κκε, προοδευτική συμμαχία, πέντε χρόνια σκοτάδι, ενωμενοι μπορουμε ενωμένοι'.

So, with this in mind I wanted to see If the use of specific n-grams could be a more useful feature for the classification. I started with uni-, bi- and tri-grams on the 2 vectorizers separately. Due to computational complexity, I reduced the maximum vocabulary to 1000 to each experiment.

3.1.1. Table of trials (metrics tested on the test set of the train set after train test split)

Vectorized Text + Feature engineering	Model	Precison	Recall	F1	Accuracy
TfIdf (uni)- max features=1000 (Sum: 1003 features)	LogReg	37.48	37.48	37.48	37.48
Tfidf (bi)-max features 1000 (Sum: 1003 features)	LogReg	37.50	37.50	37.50	37.50
TfIdf (tri) max features 1000 (Sum: 1003 features)	LogReg	36.47	36.47	36.47	36.47
Count (uni)- max features 1000 (Sum: 1003 features)	LogReg	37.41	37.41	37.41	37.41
Count (bi) – max features 1000 (Sum: 1003 features)	LogReg	37.64	37.64	37.64	37.64
Count (tri) – max features 1000 (Sum: 1003 features)	LogReg	36.47	36.47	36.47	36.47
GridSearcgCV (TFidf (uni)) 1000 C:1.0, saga, L1, cv=5	**LogReg**	**37.67**	**37.67**	**37.67**	**37.67**
GridSearcgCV (Count (bi)) 1000 C:1.0, saga, L1, cv=5	LogReg			37.5 (F1 micro)	
Count (bi)-max feat 1000 (Select 100 best: with mutual info classif)	LogReg	36.48	36.48	36.48	36.48
Count (bi)-max feat 1000 (Select 500 best: with mutual info classif)	LogReg	36.36	36.36	36.36	36.36
Count (uni_bi)-max feat 1000 (Select 500 best: with mutual info classif)	LogReg	35.84	35.84	35.84	35.84

3.2. Hyper-parameter tuning

It was observed from the table above that TFIdf unigrams with max features 1000 where the best scoring features. Selecting the parameter max_features 1000 was a boundary that I set to myself on my own (due to lack of RAM). Starting from these 1000 features, I also wanted to apply another feature selection technique more suitable for my data. I chose mutual info classification because it estimates mutual information for a discrete target variable (like the polarity) and uses the K-nearest neighbors distance to find the higher scores. The experiments were performed on the Countvectorier in bigrams and bigrams+ unigrams. Based on this technique I experimented with with 2 values of K Best: select 100-best and select 500 best, which is technically select the top 10% of my features and select the top 50% of my features. The results of the 3 last experiments showed that there is no significant difference when liniting to 100 best features. Compared to 500 best all results were worse compared to the simple max_features 1000 approach.

3.3. Optimization techniques

For the hyperparameter tuning, I chose the framework of GridSeachCV in which (again due to computational complexity) I restricted myself to searching only for the best C regularization strength.
This was my first brute force approach in choosing the hyperparameters.
The rest parameters on the Grid were chosen by me after personal research for the multiclass classification task:

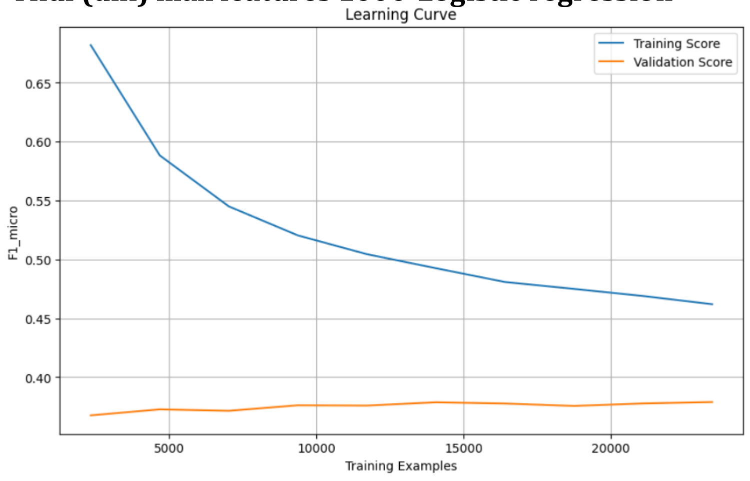
- L1 regularization (Lasso regression): I chose Lasso because my features were less that my observations and because I wanted to do feature selection on my already limited features (1003) by setting the weight of none informative features close to zero.
- C= 1.0
- Gradient based algorithm: saga : I chose saga because of its speed with large datasets and because it works with L1 penalty.
- 5-fold cross validation
- Scoring: F1-micro, which is a metric more appropriate for multiclass classification tasks. Also my dataset is balanced so there is no need for a macro F1.

3.4. Evaluation

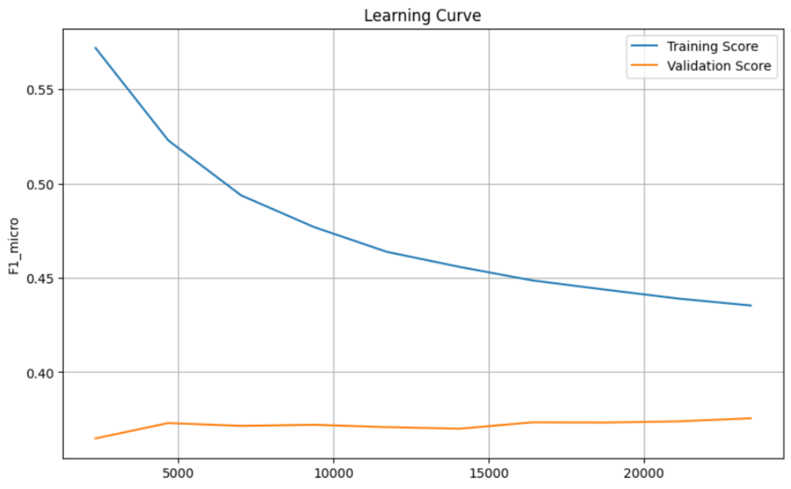
3.4.1. Learning Curve.

Learing curves were plotted with scoring: F1_micro
Micro F1 score: Calculates the metrics (precision, recall, accuracy) globally by counting the total true positives, false negatives and false positives.

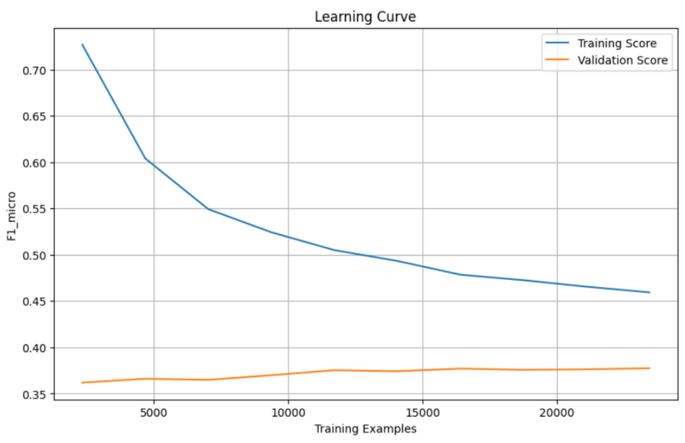
Tfidf (uni) max features 1000-Logistic regression



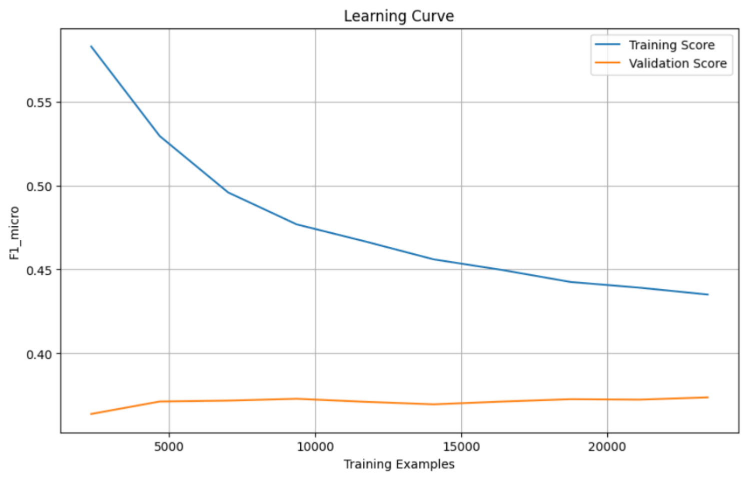
Tfidf (bi)



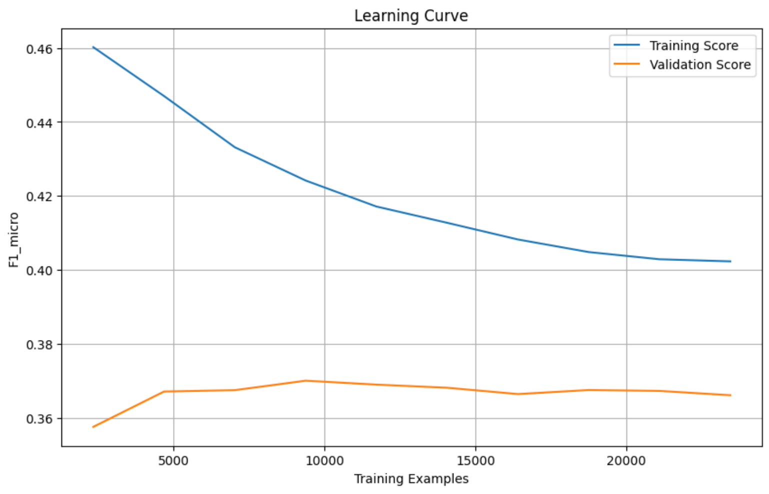
Count (uni)



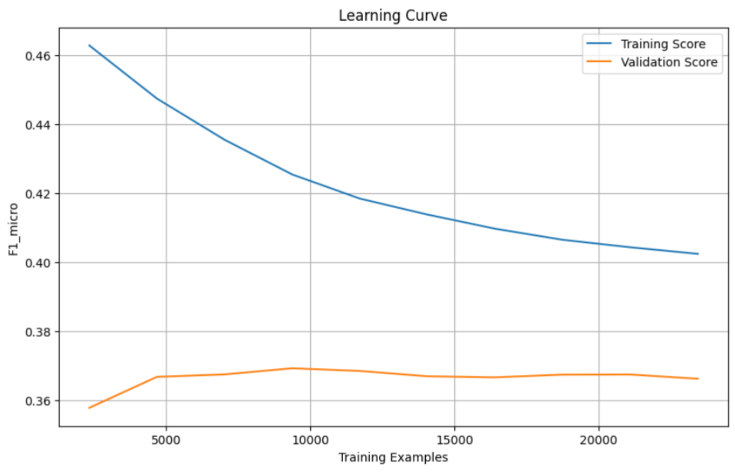
Count (bi)



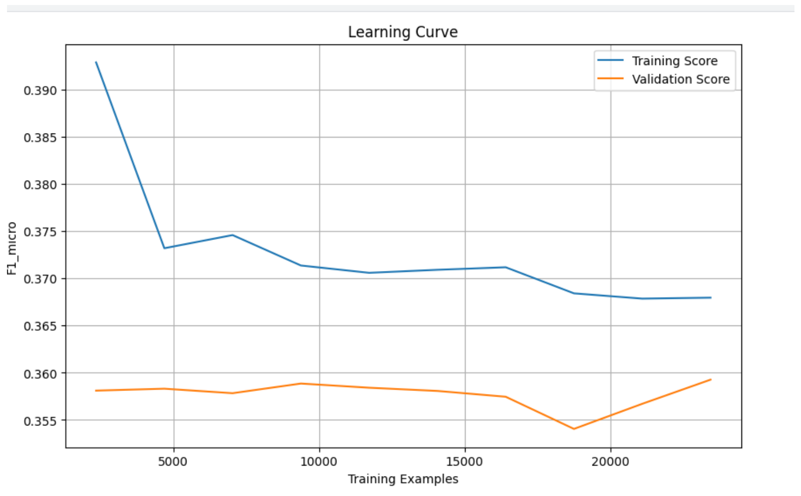
Tfidf (tri)



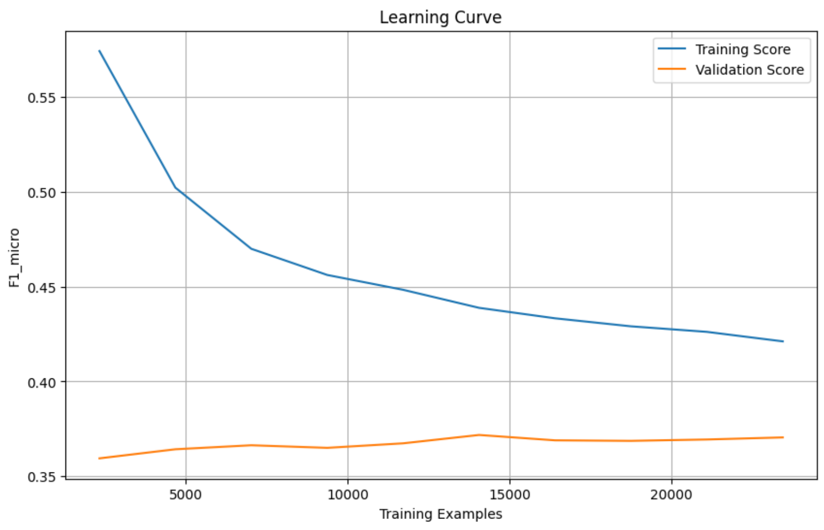
Count (tri)



Count (bi) 1000 (select 100 Best)



Count (uni_bi)-max feat 1000 (Select 500 best: with mutual info classif)



It can be deducted from all the above curves, that all our trials led **to underfitting** of the model. Typical features of an underfit model that we see above are:

1. High train score at the beginning, which gradually decreases upon adding training examples.
2. Train score and validation score are (kind- of) close to each other at the end, and they both flatten, indicating that addition of more training examples cannot improve the model performance on unseen data.

4. Results and Overall Analysis

4.1. Results Analysis

From the results above we can see that not any of our experiments improved the results in overfitting and underfitting. My assumption is that we should look for non-linear relationships between our data because the linear ones did not seem to help. I think it is a model issue here. We could try an SVM with Radial Basis Kernel, that takes the datapoints to a higher dimension so that they are linearly separable.

Based on my findings, some other experiments I could try:

- Use as a feature only the Vader logits we mentioned earlier.
- Use feature engineering to extract features like the "_xeftiles" we commented on earlier.
- Try dimensionality Reduction with PCA and keep the number of features that cover +/- 90% of the data variance.
- Lemmatize corpus (with Greek spacy) before counting the positive and negative words per tweet

4.1.1. Best trial.

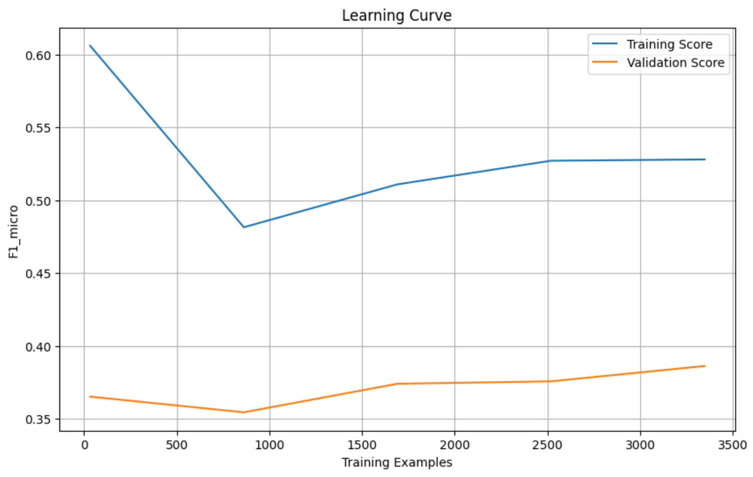
TfIdf (uni) max_features 1000 +3 numeric , L1 regularisation, saga, C: 1.0

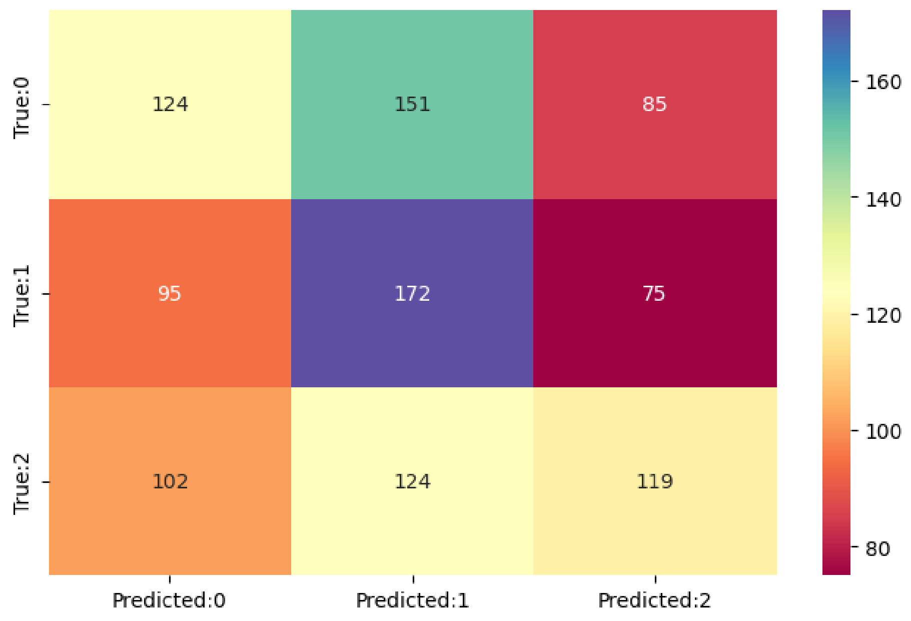
Performance of the best trial on the validation data

```
Classification report:
              precision    recall  f1-score   support

          -1       0.39      0.34      0.36       360
           0       0.38      0.50      0.44       342
           1       0.43      0.34      0.38       345

    accuracy                           0.40      1047
   macro avg       0.40      0.40      0.39      1047
weighted avg       0.40      0.40      0.39      1047
```

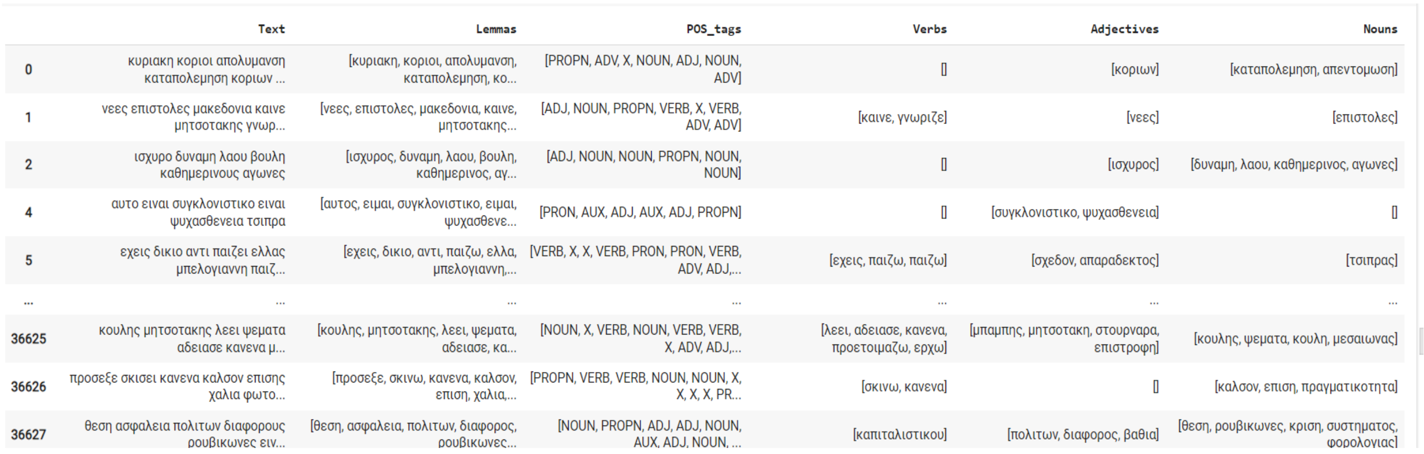




It is obvious from the results above that I could not avoid the model underfitting. As per theory, in order to deal with underfitting I should use a more complex model or come up with other features.
Features: We say above in the feature engineering part, that there were a lot of inconsistencies in the counts of positive and negative words. So maybe the counts of positive and negative words added noise to our data rather than helping them. Also, selecting the top 1000 tfidf unigrams may have been an arbitrary feature selection technique that did not help our model. Coming up with other features like the frequency of a pattern like (_xeftiles, _ξεφτίλες, _τέλος, τώρα_), lemmatizing the words before counting positive and negative word counts, the demojisation of emojies. Specially this last idea of demojisation of emojis could be implemented as long as we could create/find a lexicon similar to the sentiment dictionaries, so that a polarity sign is also available and not only the translation of emojis into natural language.

More complex model:
We could try other regularization techniques: L2 or elasticnet, and another solver like lbgfs, but still, I believe it is because the data are not linearly separable. We could plot a decision boundary to validate that assumption. Bringing the problem into higher dimensions, I think, it will improve the underfitting issue and the performance. As per theory, the model's generalization error can be expressed as the sum of three different errors:

1. Bias: we assume in our task that our data is linear.
2. Variance: We have small variance in our data, since we have done a lot of preprocessing and have left for training only words of Greek and no punctuation, emojis, latin characters and other text characters which are for sure to be expected in unseen data. Also we work with a very simple first degree polynomial model, the simplest version of logistic regression.
3. Irreducible error: Here as mentioned before, I suspect it is the count of positive and negative words that are not correctly labeled, nevertheless, I have to say that after trying Greek spacy I did not stay satisfied with the POS tagging and lemmatization annotations, which is why I avoided using it.



	Text	Lemmas	POS_tags	Verbs	Adjectives	Nouns
0	κυριακη κοριοι απολυμανση καταπολεμηση κοριων ...	[κυριακη, κοριοι, απολυμανση, καταπολεμηση, κο...	[PROPN, ADV, X, NOUN, ADJ, NOUN, ADV]	[]	[κοριων]	[καταπολεμηση, απεντομωση]
1	νεες επιστολες μακεδονια καινε μητσοτακης γνωρ...	[νεες, επιστολες, μακεδονια, καινε, μητσοτακης...	[ADJ, NOUN, PROPN, VERB, X, VERB, ADV, ADV]	[καινε, γνωριζε]	[νεες]	[επιστολες]
2	ισχυρο δυναμη λαου βουλη καθημερινους αγωνες	[ισχυρος, δυναμη, λαου, βουλη, καθημερινος, αγ...	[ADJ, NOUN, NOUN, PROPN, NOUN, NOUN]	[]	[ισχυρος]	[δυναμη, λαου, καθημερινος, αγωνες]
4	αυτο ειναι συγκλονιστικο ειναι ψυχασθενεια τσιπρα	[αυτος, ειμαι, συγκλονιστικο, ειμαι, ψυχασθενε...	[PRON, AUX, ADJ, AUX, ADJ, PROPN]	[]	[συγκλονιστικο, ψυχασθενεια]	[]
5	εχεις δικιο αντι παιζει ελλας μπελογιαννη παιζ...	[εχεις, δικιο, αντι, παιζω, ελλα, μπελογιαννη,...	[VERB, X, X, VERB, PRON, PRON, VERB, ADV, ADJ,...	[εχεις, παιζω, παιζω]	[σχεδον, απαραδεκτος]	[τσιπρας]
...	...	...	...	...	...	...
36625	κουλης μητσοτακης λεει ψεματα αδειασε κανενα μ...	[κουλης, μητσοτακης, λεει, ψεματα, αδειασε, κα...	[NOUN, X, VERB, NOUN, VERB, VERB, X, ADV, ADJ,...	[λεει, αδειασε, κανενα, προετοιμαζω, ερχω]	[μπαμπης, μητσοτακη, στουρναρα, επιστροφη]	[κουλης, ψεματα, κουλη, μεσαιωνας]
36626	προσεξε σκισει κανενα καλσον επισης χαλια φωτο...	[προσεξε, σκινω, κανενα, καλσον, επιση, χαλια,...	[PROPN, VERB, VERB, NOUN, NOUN, X, X, X, X, PR...	[σκινω, κανενα]	[]	[καλσον, επιση, πραγματικοτητα]
36627	θεση ασφαλεια πολιτων διαφορους ρουβικωνες ειν...	[θεση, ασφαλεια, πολιτων, διαφορος, ρουβικωνες...	[NOUN, PROPN, ADJ, ADJ, NOUN, AUX, ADJ, NOUN, ...	[καπιταλιστικου]	[πολιτων, διαφορος, βαθια]	[θεση, ρουβικωνες, κριση, συστηματος, φορολογιας]

Also another process that could help the model by focusing only on words that denote sentiment is to remove named entities from the corpus with NER of Greek spacy. It was seen from the Tfidf vectorizer results that the top unigrams and bigrams of the corpus contained single names or full names of the Greek parties and their representatives (κκε, νδ, συριζα,πασοκ,κιναλ., τσιπρας, μητσοτακης, κυριακος μητσοτακης, αλεξης τσιπρας, νεα δημοκρατια, συριζα προοδευτική συμμαχια.....) So I can imagine that selecting the top 1000 tfidf feature vectors from a corpus from which named entities are not removed, is a noise that has not really helped the model learn to associate patterns with polarities, but rather only names with polarities, which does not make sense, for our task, because there will be both negative and positive tweets for all parties.

5. Bibliography

References

https://towardsdatascience.com/tuning-the-hyperparameters-of-your-machine-learning-model-using-gridsearchcv-7fc2bb76ff27

https://medium.com/@rithpansanga/logistic-regression-and-regularization-avoiding-overfitting-and-improving-generalization-e9afdcddd09d

https://towardsdatascience.com/dont-sweat-the-solver-stuff-aea7cddc3451

https://guhanesvar.medium.com/feature-selection-based-on-mutual-information-gain-for-classification-and-regression-d0f86ea5262a

https://towardsdatascience.com/learning-curve-to-identify-overfitting-underfitting-problems-133177f38df5

https://github.com/NKryst/Greek-Sentiment-Analysis/tree/master