Module 4:

Logistic Regression and Naive Bayes on Sentiment Analysis

Input Data

The data used for the sentiment analysis is from Kaggle 'Amazon Fine Food Reviews': https://www.kaggle.com/snap/amazon-fine-food-reviews?select=Reviews.csv

The original size of the data was 568,454 reviews. The original csv file brings 10 fields of information: 'id', 'Productid', 'Userid', 'ProfileName', 'HelpfulnessNumerator', 'HelpfulDenominator', 'Score', 'Time', 'Summary', 'Text'.

From these fields we only need Score and Text. Score stands for a 1-5 star while Text is the review itself.

In order to know if the review is positive or negative, reviews with a score of 1 or 2 are consider negative while score 4 or 5 are consider positives. Score 3 are drop.

After doing this modification we get a dataframe with 525,814 reviews, from which 443,777 are consider positive and 82,037 are consider negative reviews. So, in order to keep both classes with the same number of elements, only 82,037 positives reviews are randomly selected.

At the end we have 164,074 reviews, with only 2 fields. One is Text, consisting of the review and the other is Sentiment, consisting in a 0 if negative or 1 if positive review.

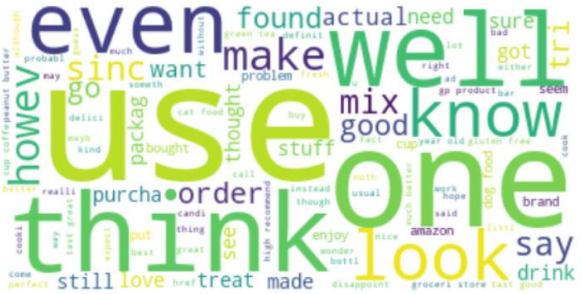
The data is then treated to do the clean-up, tokenization, remove stop words and stemming.

Alejandro C Parra Garcia

Word Clouds

We can do 3 different word clouds, one for all the reviews, another one for positive reviews and the third one for negative reviews.

All reviews



Positive reviews

Negative reviews

As we can see the words ‘one’ and ‘use’ are upon the most used words. We can also see that the word ‘good’ is used more frequently on the positive reviews than on the negatives, as well as the word ‘make’, ‘love’, ‘well’. While words such as ‘think’, ‘order’, ‘even’ are used at a higher rate on the negative reviews.

Algorithms

I have made use of 2 algorithms Multinomial Naïve Bayes and Logistic Regression. Each of them has been done with bag of words and Term frequency–inverse document frequency (TF-IDF).

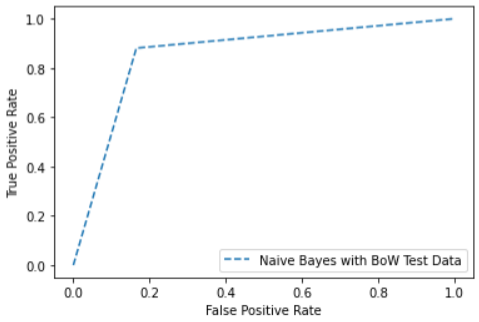
When using the combinations, I got the next results on the test data:

Naive Bayes

Bag of Words

	precision	recall	f1-score	support
0	0.88	0.83	0.85	24640
1	0.84	0.88	0.86	24583
accuracy			0.86	49223
macro avg	0.86	0.86	0.86	49223
weighted avg	0.86	0.86	0.86	49223

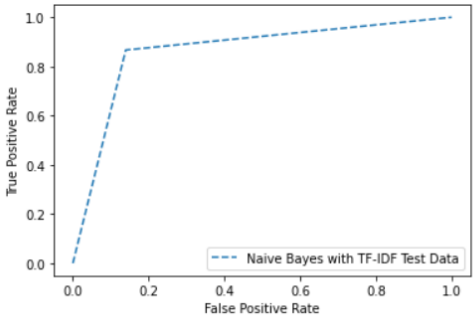
```
Confusion matrix:
[[20565  4075]
 [ 2931 21652]]
Accuracy:
0.857668163257014
ROC Curve:
```



TF-IDF

	precision	recall	f1-score	support
0	0.87	0.86	0.86	24640
1	0.86	0.87	0.86	24583
accuracy			0.86	49223
macro avg	0.86	0.86	0.86	49223
weighted avg	0.86	0.86	0.86	49223

```
Confusion matrix:
[[21199  3441]
 [ 3273 21310]]
Accuracy:
0.8636003494301444
ROC Curve:
```

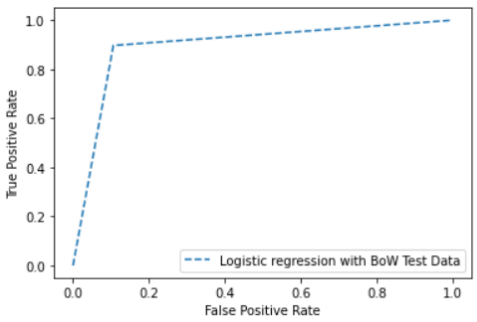


Logistic regression

Bag of Words

	precision	recall	f1-score	support
0	0.90	0.89	0.90	24640
1	0.89	0.90	0.90	24583
accuracy			0.90	49223
macro avg	0.90	0.90	0.90	49223
weighted avg	0.90	0.90	0.90	49223

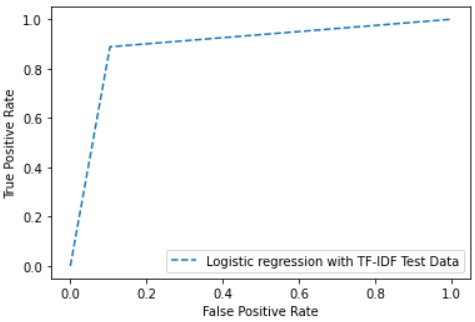
```
Confusion matrix:
[[22021  2619]
 [ 2536 22047]]
Accuracy:
0.8952725351969608
ROC Curve:
```



TF-IDF

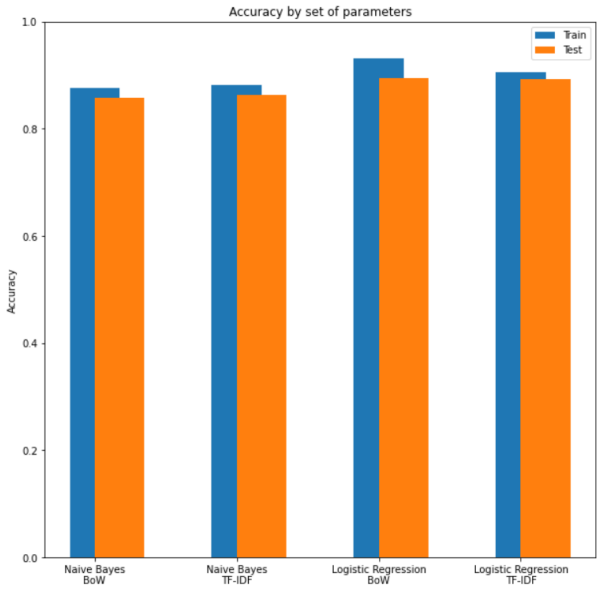
	precision	recall	f1-score	support
0	0.89	0.90	0.89	24640
1	0.89	0.89	0.89	24583
accuracy			0.89	49223
macro avg	0.89	0.89	0.89	49223
weighted avg	0.89	0.89	0.89	49223

```
Confusion matrix:
[[22076  2564]
 [ 2742 21841]]
Accuracy:
0.8922048635800337
ROC Curve:
```



In order to compere the data its easier to do it visually:

Accuracy:



Confusion matrix

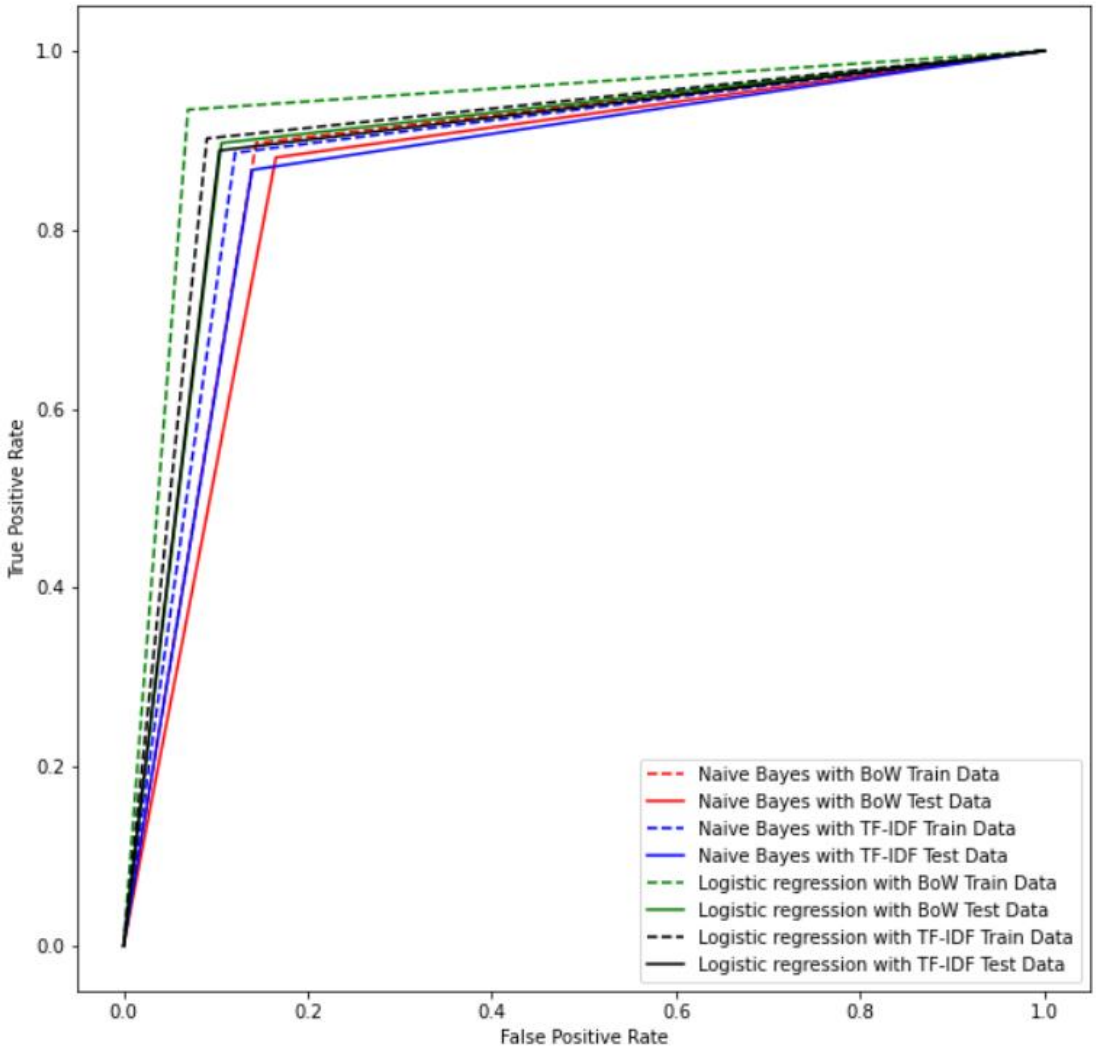
```
Naive Bayes with Bag of Words
  Train:
 [[49136  8261]
 [ 5929 51525]]
  Test:
 [[20565  4075]
 [ 2931 21652]]

Naive Bayes with TF-IDF
  Train:
 [[50444  6953]
 [ 6553 50901]]
  Test:
 [[21199  3441]
 [ 3273 21310]]

Logistic regression with Bag of Words
  Train:
 [[53383  4014]
 [ 3783 53671]]
  Test:
 [[22021  2619]
 [ 2536 22047]]

Logistic regression with TF-IDF
  Train:
 [[52202  5195]
 [ 5633 51821]]
  Test:
 [[22076  2564]
 [ 2742 21841]]
```

ROC Curves:



From all the metrics we can conclude that Logistic regression is better than Naïve Bayes, as we can see from the ROC curves, the black and green lines represent Logistic Regression, while the blue and red are Naive Bayes.

As well from the accuracy, as Logistic regression have an accuracy of 89.52% (BoW) and 89.22% (TF-IDF) compere with Naïve Bayes 85.76% (BoW) and 86.36% (TF-IDF).

Between BoW and TF-IDF, we can conclude that in the case of Naïve Bayes, TF-IDF is better, as it achieves a higher Accuracy and a better ROC curve. While in the case of logistic regression BoW is marginally better than TF-IDF, as it gets just a 0.3% better accuracy. And if we check the ROC Curves, both curves are closely mach. So, in return we can say that TF-IDF is better.

Next steps will be to try N-grams in place of tokens of one word, and see the effects it has on the algorithm, as well as to try new algorithms.