Report Title Goes Here

Dino Fejzulovic - 000817879

# Data Description

Data set name, source, description of features, description of classification task, plus some statistics – number of features, number of items in each class, range of each feature, etc.

This report describes the Naive Bayes function in relation to Covid Tweets created in the last year. We will be comparing Covid tweets and the overall sentiment the tweet is in. I have used the Bag of words expression in relation to both CNB and MNB , I have additionally experimented with a tfid vectorizer with a linear SVC. Some experimentation with stems and n-grams were also conducted, though in very very limited scope.

# Description of the Text Representation

The experiments that I conducted were based on a the Multinomial Naïve Bayes with a bag of words classifier, a Component Naïve Bayes with a bag of words classifier, and a tfid classifier with a svc linear output. These were done with a variety of entries to fully test each method, these could include n grams and simple words. The main purpose of this program is to of course run covid test to test them for sentiments with a new test set. I want my results to be as follows:

Sentiments when converted to integers ( They were originally

1. Extremely Negative – 0
2. Negative – 1
3. Neutral – 2
4. Positive – 3
5. Extremely Positive - 4

# Results

## Part 1: MNB with bag of words classifier

Overview goes here. Describe the three parameters you varied and the values you chose for the parameters (K parameter, voting, distance, normalization, etc.). Describe how you computed the testing/training split for each run

This was conducted with random sentences and words that I came up with, as well as some from the test set.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Average | Run 1 | Run 2 | Run 3 | Run 4 | Run 5 | sentiment |
| text = ['Not because Im paranoid, but because my food stock is litteraly empty. The coronavirus is a serious thing, but please, dont panic. It causes shortage...'] | 1 | 1 | 1 | 1 | 1 | 1 | Negative |
| text = ['covid-19 big lies '] | 1 | 1 | 1 | 1 | 1 | 1 | Negative |
| text = ['I think covid will be over soon, we will then be very happy!'] | 4 | 4 | 4 | 4 | 4 | 4 | Extremely Positive |
| text = ['Covid bad'] | 0 | 0 | 0 | 0 | 0 | 0 | Extremely Negative |
| text = ['What is the deal with airline food? I mean it really sucks!'] | 1 | 1 | 1 | 1 | 1 | 1 | Negative |
| text = ['Joe Biden has to be responsible for Covid-19 if he only gives me $400 for a stimulus check! '] | 3 | 3 | 3 | 3 | 3 | 3 | Positive |

## Part 2: CNB with bag of words classifier

Overview goes here. Describe the three parameters you varied (criterion, splitter, max\_depth, max\_leaf\_nodes, etc.) and the values you chose for the parameters (K parameter, voting, distance, normalization, etc.). Describe how you computed the testing/training split for each run

This was conducted with random sentences and words that I came up with, as well as some from the test set, CNB delivered a little different of results.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Average | Run 1 | Run 2 | Run 3 | Run 4 | Run 5 | sentiment |
| text = ['Not because Im paranoid, but because my food stock is litteraly empty. The coronavirus is a serious thing, but please, dont panic. It causes shortage...'] | 0 | 0 | 0 | 0 | 0 | 0 | Extremely Negative |
| text = ['covid-19 big lies '] | 1 | 1 | 1 | 1 | 1 | 1 | Negative |
| text = ['I think covid will be over soon, we will then be very happy!'] | 4 | 4 | 4 | 4 | 4 | 4 | Extremely Positive |
| text = ['Covid bad'] | 0 | 0 | 0 | 0 | 0 | 0 | Extremely Negative |
| text = ['Joe Biden has to be responsible for Covid-19 if he only gives me $400 for a stimulus check! '] | 0 | 0 | 0 | 0 | 0 | 0 | Extremely Negative |
| text = ['What is the deal with airline food? I mean it really sucks!'] | 0 | 0 | 0 | 0 | 0 | 0 | Extremely Negative |

## Part 3: TFID vectorizer with SVM linear

Tried to find one or two new examples to test with a TFID vectorizer to see how it runs, got a very precise reading surprisingly. I was hoping for more neutral solutions as my previous set ups have never given me a neutral sentiment towards the tweets. Thankfully I got one , but it is somewhat skewed in my opinion as it would fail to detect some tests that the other algorithms got correct. This was more so for testing purposes to create a third alternative.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Average | Run 1 | Run 2 | Run 3 | Run 4 | Run 5 | sentiment |
| text = ['Joe Biden has to be responsible for Covid-19 if he only gives me $400 for a stimulus check! '] | 2 | 2 | 2 | 2 | 2 | 2 | Neutral |
| text = ['covid-19 big lies '] | 1 | 1 | 1 | 1 | 1 | 1 | Negative |
| text = ['I think covid will be over soon, we will then be very happy!'] | 4 | 4 | 4 | 4 | 4 | 4 | Extremely Positive |
| text = ['housing rentals have become so damn expensive because of the stupid property owners! I hate them all extremely! '] | 0 | 0 | 0 | 0 | 0 | 0 | Extremely Negative |

# Discussion

**Part 1: Are there clear winners or losers for any of your algorithms? Give some solid ideas for why some versions might be better or not better than others.** **Are there clear winners or losers? Be as specific as you can and reference the properties of your data set. Which text classifier would you recommend and why?**

I *think each algorithm has its pro’s and cons. The CNB tended to perform better than the MNB when it came to certain properties while the TFID completely blew both of them in the areas that they lacked. The issue that I had was that my accuracies were unfortunately below 0.50 so I could tell that my three classifiers were having difficulty determining it. While it was for the most part correct on simple sentences and key words, it struggled in complex sentences .. though the tfid excelled in that part but lacked in the fundamental examples. I would say that I do not have a winner or a loser but that the TFID probably has the most potential to get polished out be more successful of the two for this data set*

# Future Work

If you had more time, where would you go next? What other variations of text representation would you like to explore? What other algorithms or data sets would you like to use? What other tests would you like to do? Etc.

If I had more time I would simply try to get my accuracy score much much higher as the Naïve Bayes I feel can be much much more accurate. Though it did not struggle it most definitely did not get every prediction that I wanted it to make ( I may be too naïve about the naïve bayes after all ! ) . With that settled I would like to work more on the TFID SVM configuration. I didn’t have a chance to play around with the Bernoulli Naïve bayes configuration but I believe that it may give a similar result ( I might have to play around with it as my dataset is not the ideal environment for it) . In terms of other tests I believe I covered whatever I could based on what I wanted out of the script, I tried to of course confuse it as best as I can but I feel a true test would take almost 200 tests with bag of words, bag of stems, n grams ( briefly used, etc. I truly do believe that with enough iterations that my algorithms can work well.