Roche

Interview Exercise 2

Dawid Sitnik

1.Objective

The aim of this exercise is to analyze a dataset of headlines and classify them as sarcastic or not using binary classification model. I decided to solve the whole problem using only python.

2.Data preprocessing

In this particular case, data was already acquired, so the first thing which I did was the preparation of a dataset. After my analysis I found out, that it was already well-formatted – it was a JSON file consisted of two fields: 'headline' and 'is_sarcastic' which is 0 for not sarcastic headline and 1 for sarcastic one. To ensure that there aren't any inconsistencies in the dataset I got basic information about it.

```
RangeIndex: 26709 entries, 0 to 26708
Data columns (total 2 columns):
headline        26709 non-null object
is_sarcastic    26709 non-null int64
dtypes: int64(1), object(1)
memory usage: 417.4+ KB
None
```

As we see there are no null values in the dataset, so the data can be processed. I was also looking for any overlapping data but I didn't find. The data was already in the lowercase form, so I just needed to clean the data, tokenize it, and then do lemmatization process.

Text Cleansing

The purpose of text cleansing is to remove unnecessary character from our texts. Here we gonna remove digits and punctuations since they are not needed in our sarcasm detection.

```
Original texts :
'moana' sails straight to the top of the box office with massive $81.1 million opening

After cleansed :
moana sails straight to the top of the box office with massive  million opening
```

Tokenization

Tokenization is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens. It can help with words checking or conversion process.

```
Before tokenization :
moana sails straight to the top of the box office with massive  million opening

After tokenization :
['moana', 'sails', 'straight', 'to', 'the', 'top', 'of', 'the', 'box', 'office', 'with', 'massive', 'million', 'opening']
```

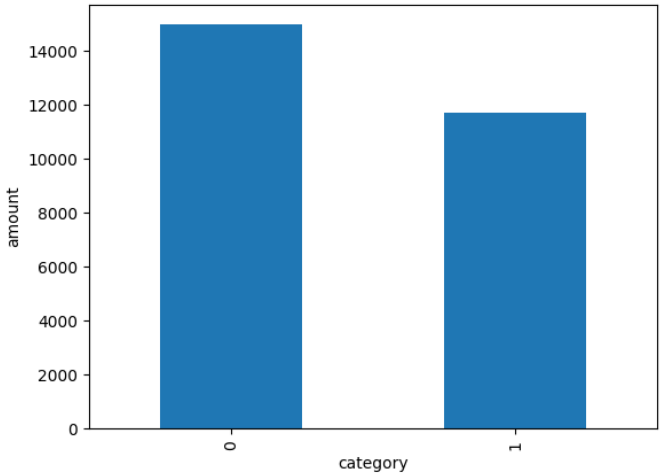
Lemmatization

Lemmatization is a process of converting the words of a sentence to its dictionary form, which is known as the lemma. Unlike stemming, lemmatization depends on correctly identifying the intended part of speech and meaning of a word in a sentence, as well as within the larger context surrounding that sentence, such as neighboring sentences or even an entire document.

```
Before lemmatization :   ['skyrim', 'dragons', 'are', 'having', 'parties']
After lemmatization :    ['skyrim', 'dragon', 'be', 'have', 'party']
```

3. Data Visualization

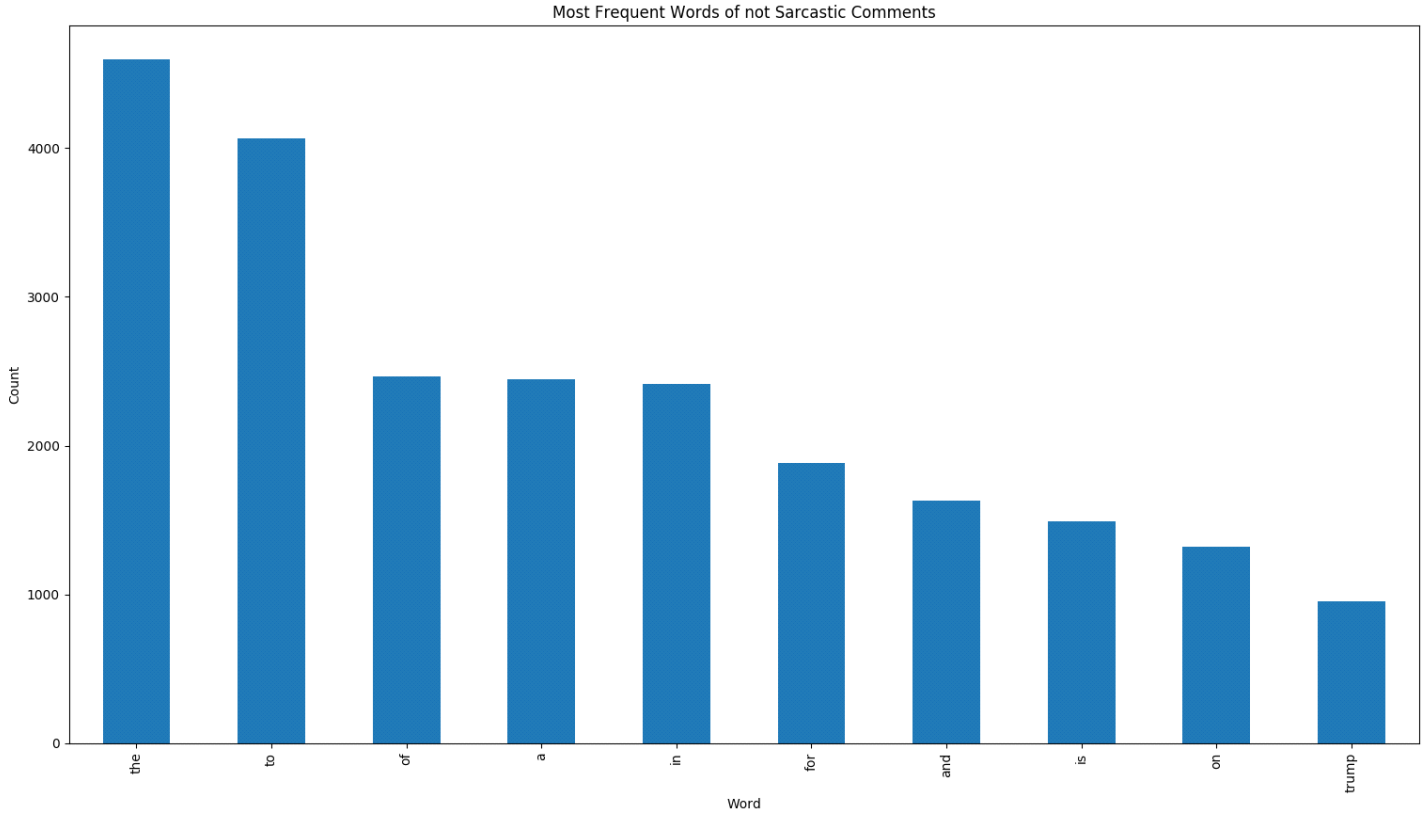
The aspect with which I want to start my analysis is the amount of sarcastic and not sarcastic headlines.

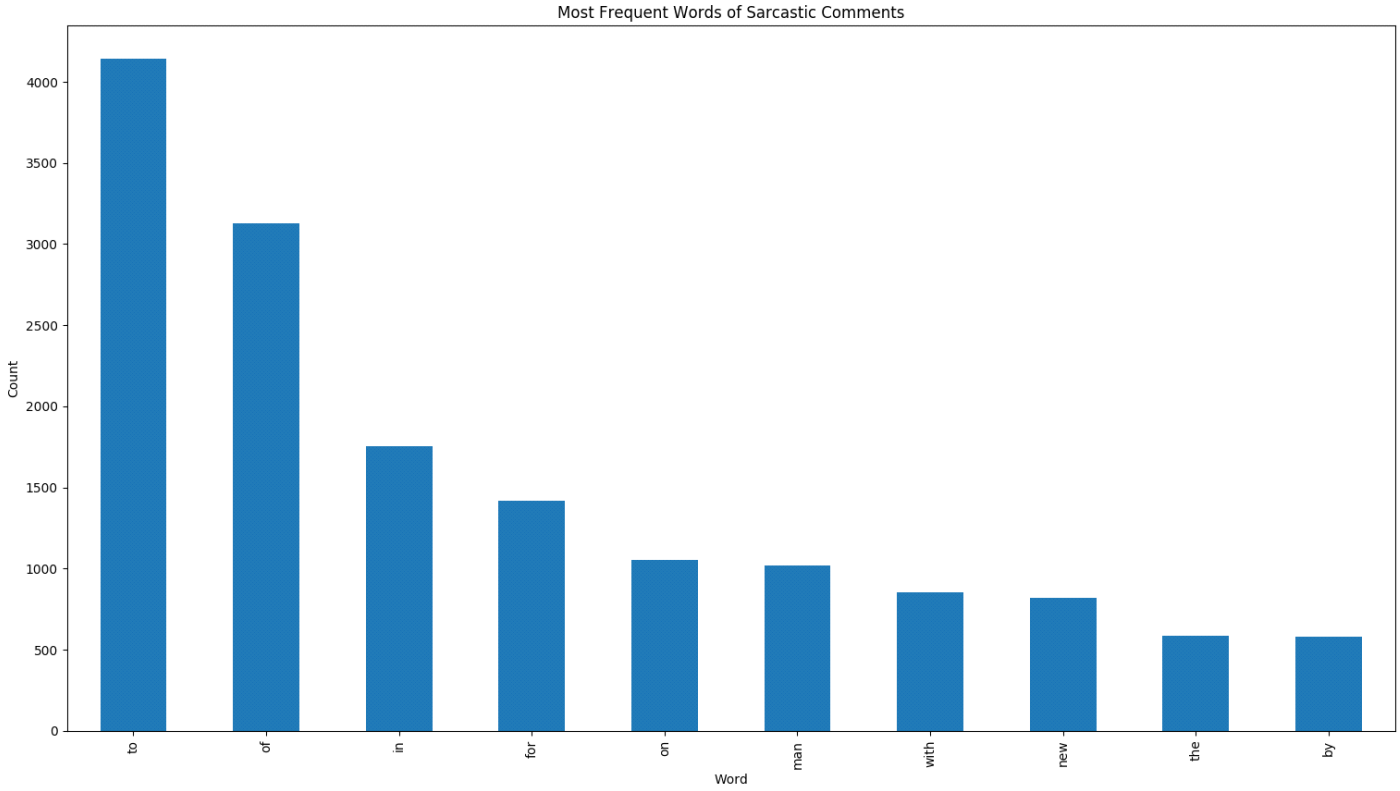


Amount of Sarcastic (1) and Non-Sarcastic Words (0)

According to the picture there is the advantage of non-sarcastic headlines. The difference is about 3000 words.

Then I count and display the most popular words for two groups

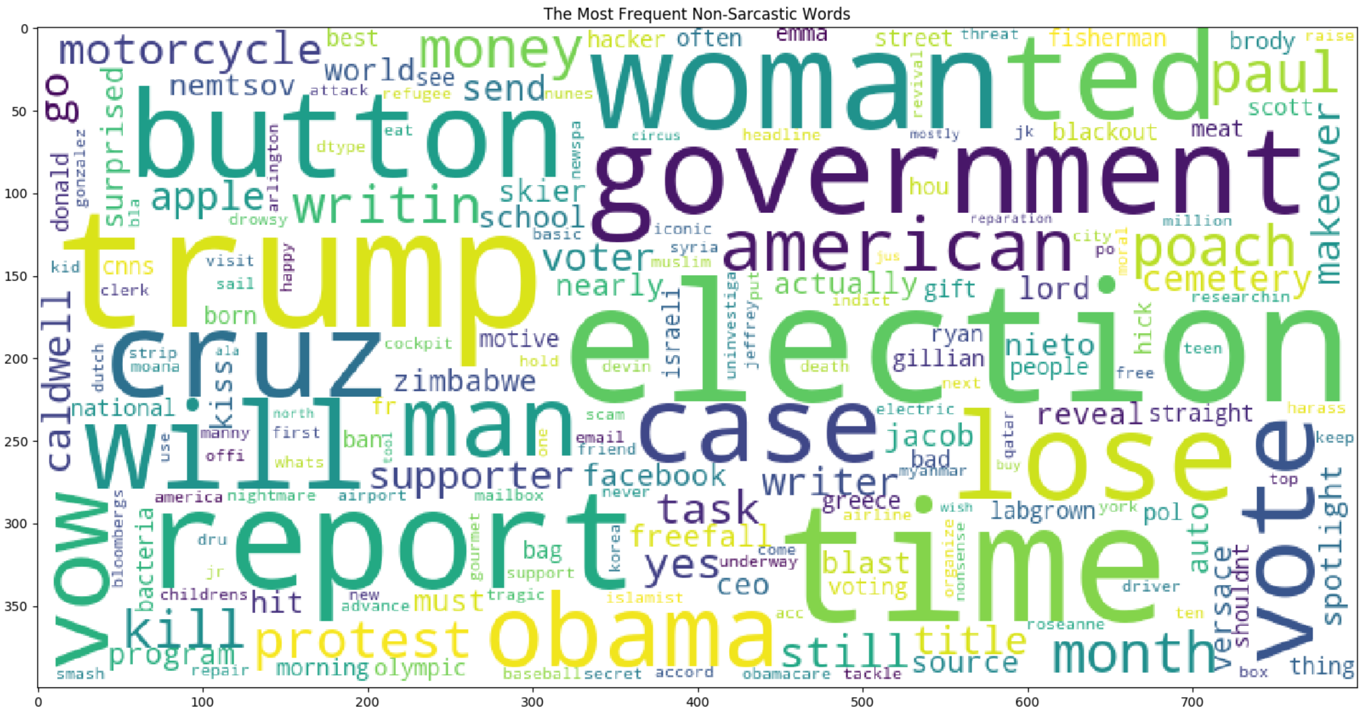




As we can see, the most popular words are mostly prepositions like to, the, of, etc.

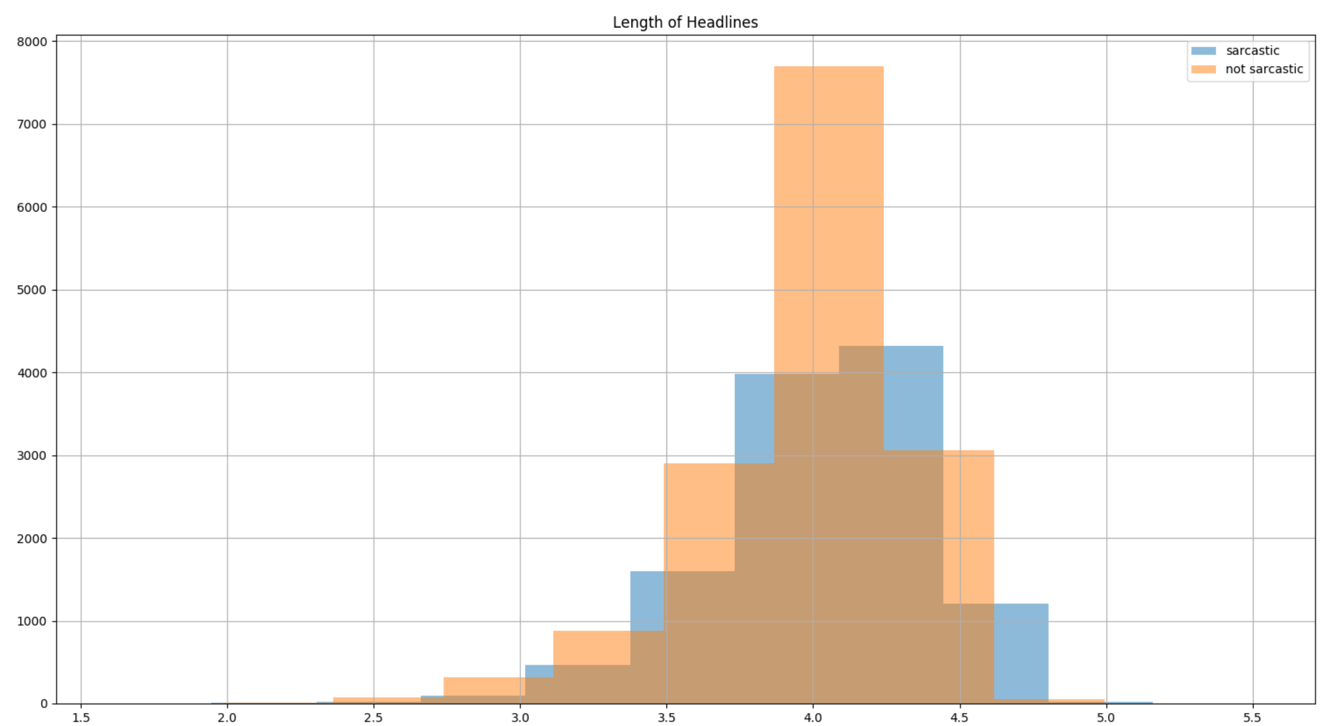
To visualize the popularity of longer words I used word map and get those results:

For sarcastic headlines, words which definitely out stands are a report, without, man, woman, area, break, etc.



For non-sarcastic the most popular ones are: button, time, trump, election, Cruz, new, government, etc.

In the end, I wanted to show the distribution of lengths for sarcastic and neutral headlines. Non-sarcastic words seems to be a little bit longer in average. The biggest difference is between headlines which contain about 4 words.



4. Modeling

4.1 Steps To Be followed When Applying an Algorithm

1. Split the dataset into training and testing dataset.
2. Select classification algorithm
3. Train algorithm on dataset
4. Pass the testing data to the trained algorithm to predict the outcome
5. Check the accuracy by passing the predicted outcome and the actual output to the model.

4.2 Split Dataset into a Training Set and a Testing Set

At the beginning of modeling process I started with splitting data into training and testing sets.

Advantages

- By splitting the dataset we can train using one set and test using another.
- This ensures that we won't use the same observations in both sets.
- More flexible and faster than creating a model using all of the dataset for training.

Disadvantages

- We can get different results depending on how the set was spitted
- We can solve this problem using k-fold-cross-validation. In have already used this method in previous exercise, so now I decided to skip this part.

4.3 Select classification algorithm

I decided to use different algorithms and see which one will give the best result. I used:

- Logistic Regression
- SVC
- Linear Discriminant Analysis
- GBoost
- Mnb

As there are many of them I am going to describe one.

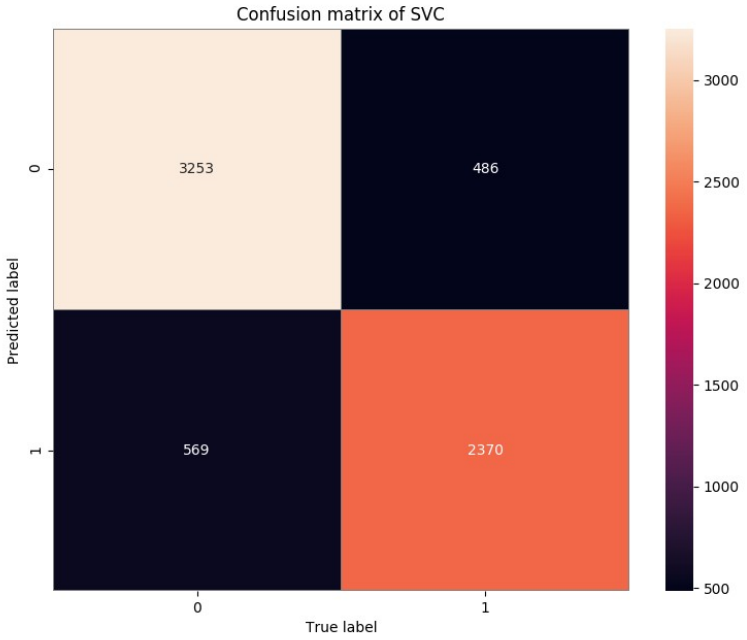
Before I was able to teach my models I needed to exchange my headlines into numeric values, because algorithms can’t process string data.

Accuracy of my models is shown bellow:

```
Accuracy of Logistic Regression classifier on test set: 0.838
Accuracy of Random Forest Classifier classifier on test set: 0.800
Accuracy of SVC classifier on test set: 0.842
Accuracy of GBoost classifier on test set: 0.818
Accuracy of Mnb classifier on test set: 0.839
```

As we can see all the results are quite similar, but the best algorithm for that case was SVC with accuracy 0.842. The worst one was Random Forest Classifier with 0.800.

To visualize my result I have also printed its confusion matrix.



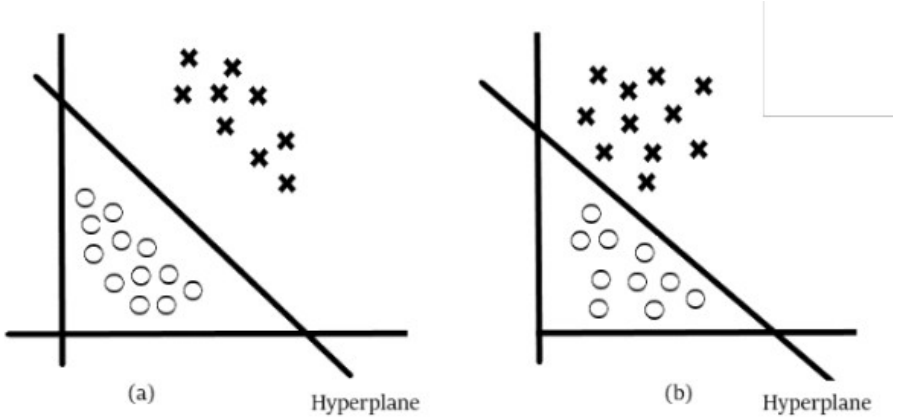
As we can see, the total size of the testing data is 6678, which is more less 25% of the whole data. Looking at the matrix, we can see how many mistakes our algorithm did and in which situations. The number of proper classifications for testing set was 3253 + 2370 and the number of mistakes is the sum from black squares which is 486 + 569.

4.4. Best algorithm explanation:

What is SVM?
Support vector machines is a supervised algorithm used for classification and regression problems. It is used for the smaller dataset as it takes too long to process

The ideology behind SVM:
SVM is based on the idea of finding a hyperplane that best separates the features into different domains.

Intuition development:
Let's consider our situation:
We have headlines that are sarcastic or not and we want to design a function (hyperplane) which will clearly differentiate those two groups.



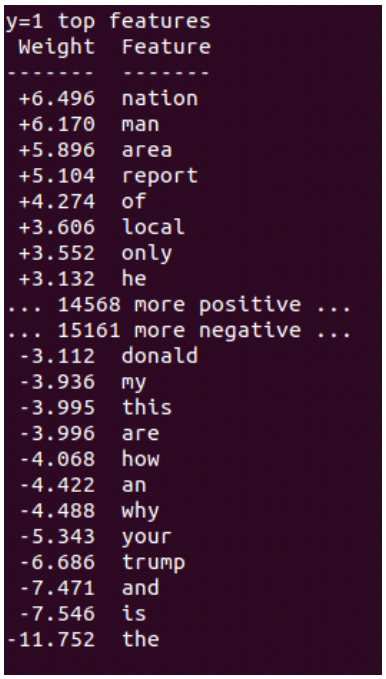
Consider those two figures with drawn hyperplane. As we can see the first one makes celearer classification so we would pick this one.

Basically, SVM bases on choosing an optimal hyperplane which will clearly classify the different classes(binary in this case).

4.5. Words weights

There are two main ways to look at a classification model:
- inspect model parameters and try to figure out how the model works globally
- inspect an individual prediction of a model, try to figure out why the model makes the decision it makes.

I decided to follow the first proposition and used show_weights() function from the ELI5 library. This function prints us weights of each word, starting from the most important ones. The biggest score word got the biggest impact on the classification it has.



Words Weights

As we can see at the picture, the words which have the biggest impact on our classification were: nation, man and area, the least important ones were short words like the, and, is (the one which we could see on the plot as the most frequent)