Capstone project

Data Science

Mohammad Saleh

### Sentiment Analysis of Text Data (COVID-19 Tweets)

# Abstract

This project addresses the problem of analyzing feelings on Twitter. The goal of this project was to predict the emotions of this specific Twitter dataset using Python. Sentiment analysis can predict many different emotions associated with the text, but in this report only 3 disciplines were considered: positive, negative and neutral. The training data set was large and the data in it were very good, which greatly affected the ease of building a good classifier. After creating a lot of custom features, using word bag representations and applying some classification algorithms, the classification accuracy was achieved at the level of 89%. The custom features include: Analyzing general sentiments as companies try to know the response of their products to the market, predict political elections and forecast social and economic phenomena such as stock market. This report will describe my project, explain the process used to complete my project, analyze the steps I took in completing my project, and provide two improvements for future projects.

# Introduction

This project addresses the problem of analyzing feelings on Twitter. The goal of this project was to predict the emotions of this specific Twitter dataset using Python. Sentiment analysis can predict many different emotions associated with the text, but in this report only 3 disciplines were considered: positive, negative and neutral. The training data set was large and the data in it were very good, which greatly affected the ease of building a good classifier. After creating a lot of custom features, using word bag representations and applying some classification algorithms, the classification accuracy was achieved at the level of 89%. The custom features include: Analyzing general sentiments as companies try to know the response of their products to the market, predict political elections and forecast social and economic phenomena such as stock market. This report will describe my project, explain the process used to complete my project, analyze the steps I took in completing my project, and provide two improvements for future projects.

**Statement of the Problem**

Text analysis to see if the person's feelings are positive, negative or neutral

**Significance of the project**

The biggest benefit of this analysis is knowing people's opinions about the virus and analyzing texts to find out their reactions and classify them as positive, negative or neutral

# Background

In this project, I had the opportunity to work in the sentiment analysis field. While dealing with the reviews of people, wit is important to interpret what the user tends to portray so that we can give him the best results. Apart from this, sentiment analysis has been an interesting field of study. This is still an evolving subject, it has functions that are too complicated to understand by the machines such as sarcasm, negative emotions, hyperbole etc. Because I am part of the industry, I know the potential in sentiment analysis. It adds a lot of value to the industry. Sentiment analysis bases its results on factors that are so inherently humane, it is bound to become one the major drivers of many business decisions in future.

# Methodology

Sentiment analysis is a wonderful field in data science. There are several steps that must be taken to analyze sentiment in tweets. First, you must have data about the tweets that you should work on, and it is preferable to have big data because the bigger the better the analysis. You display the data, read it and understand its characteristics, and begin to understand the relationships between the columns. Then we start by cleaning the data from punctuation marks, links and special symbols, and many other things that are not necessary for us in the analysis and will consume from RAM only. After we clean the data, we start building models based on clean data. There are several classifiers that can be used such as Logistic Regression, Stochastic Gradient Decent, Naive Beyers and XGboost. Then, you compare the result for each Classifier you used and take the Classifier which has the highest accuracy.

# Conclusions and recommendations

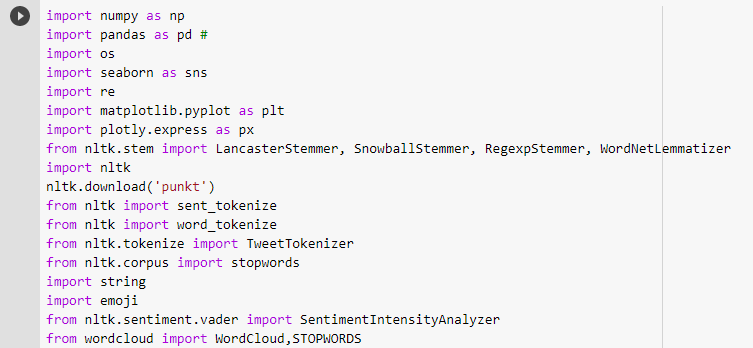
The increase in microblogging sites like Twitter provides an unparalleled opportunity to shape and use methods and technologies that seek and mine emotions. The work presented in this paper outlines an approach to analyzing sentiment on Twitter data. To reveal feelings, we extracted relevant data from Tweets and added features. The overall sentiment of the tweets was then calculated using a model presented in this report. This work is exploratory in nature and the prototype that has been evaluated is a prototype. The models showed that predicting text sentiment is a non-trivial task of machine learning. A lot of pre-processing is needed just to be able to run an algorithm. The main problem with sentiment analysis is the formulation of an automated representation of a text. The simple word set was definitely not enough to get satisfactory results; thus, a lot of additional features were created based on common sense (number of symbols, exclamation marks, number of question marks, etc.). I believe a slight improvement in the classification accuracy could be developed for the given training data set, but since it included highly skewed data (a small number of negative cases), the difference is likely to be in the order of a few percentages. The thing that could boost the classification results is adding a lot of additional examples (increasing the training data set), moreover - a lot of information expressing feelings is definitely missing.

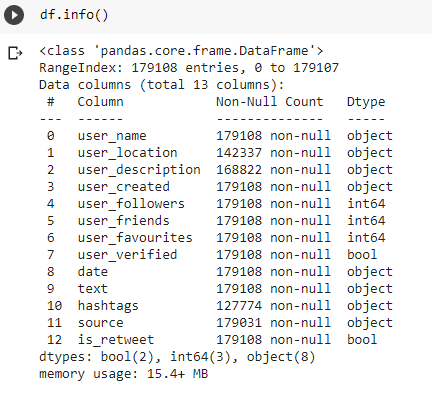
# References

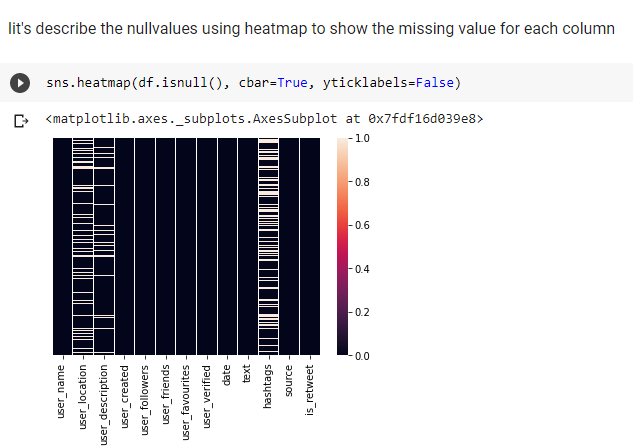
[1] [Covid 19 tweets Analysis | Kaggle](https://www.kaggle.com/renjithrrkj/covid-19-tweets-analysis)

# use the Natural Language Toolkits such as NLTK.

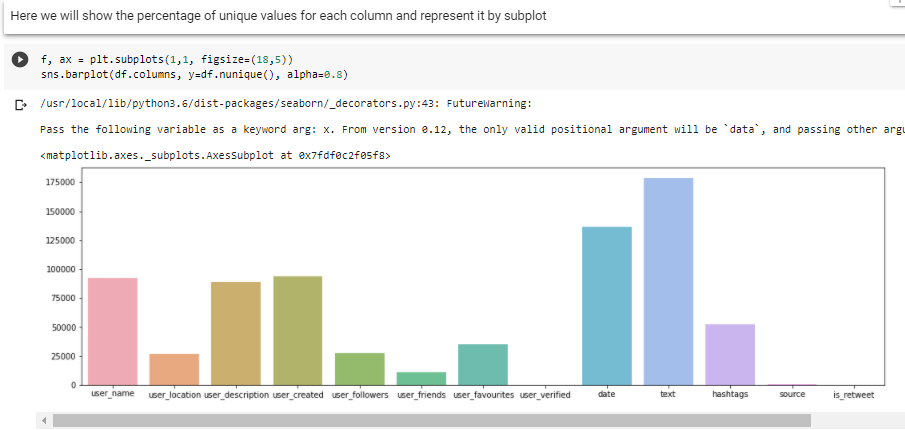
**Import the important libraries**



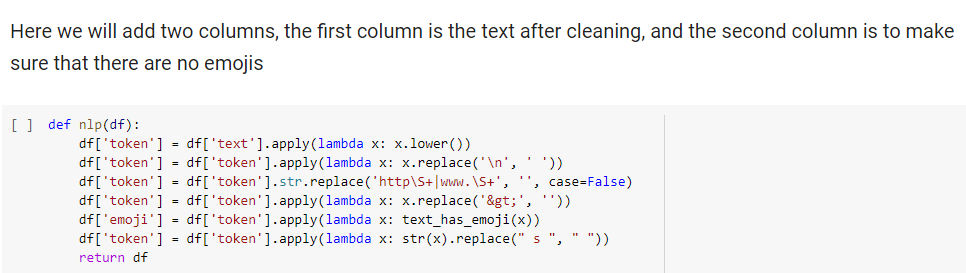
* **A basic process on text-based content text data/tweet cleaning**

****

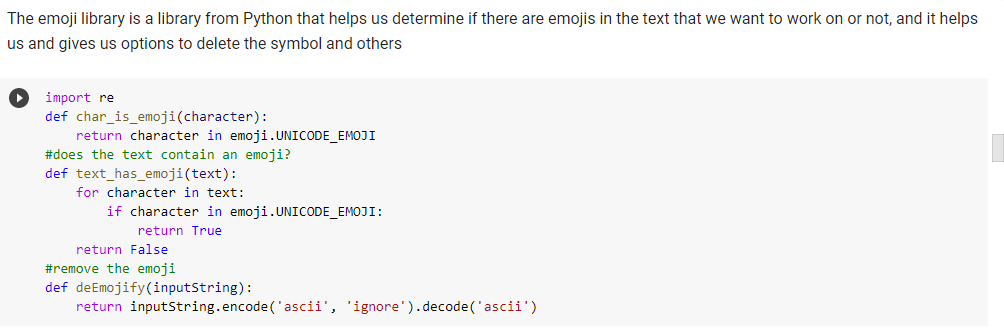
* **Descibe the percentage of unique values**

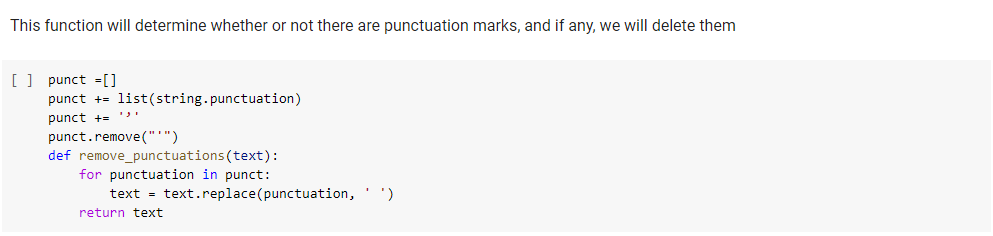
****

# Use Regular expression to clean text

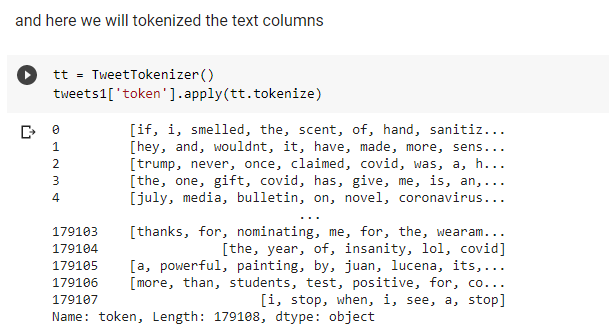
****

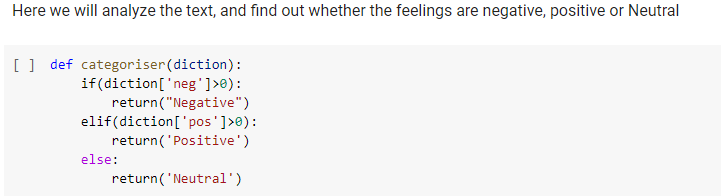
* **Remove emoji and Punctuation**

****

****

* **Tokenization**

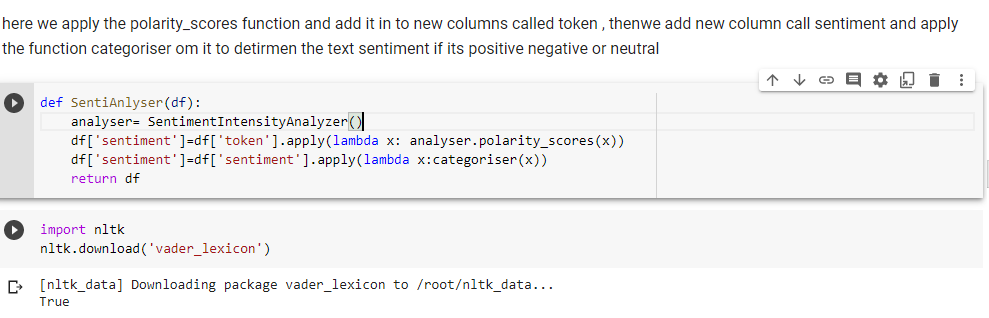


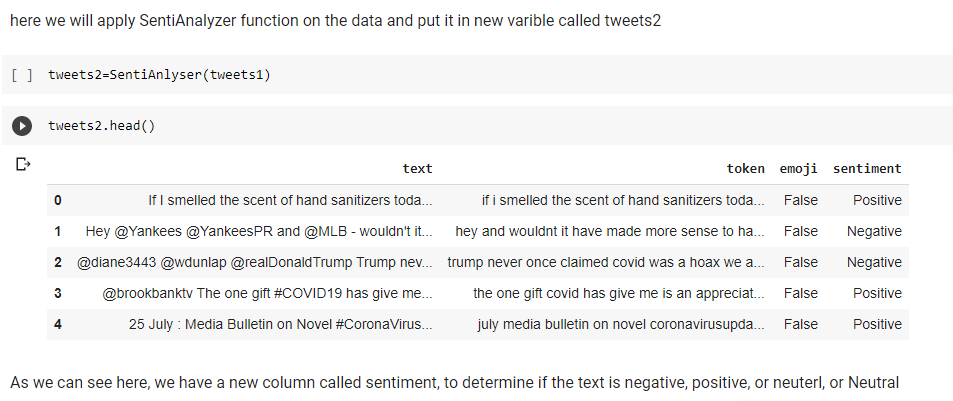


Sentiment Analysis is the process of ‘computationally’ determining whether a piece of writing is positive, negative or neutral. It’s also known as opinion mining, deriving the opinion or attitude of a speak. VADER (Valence Aware Dictionary and sentiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. VADER uses a combination of A sentiment lexicon is a list of lexical features (e.g., words) which are generally labeled according to their semantic orientation as either positive or negative. VADER not only talks about the Positivity and Negativity score but also tells us about how positive or negative a sentiment is.

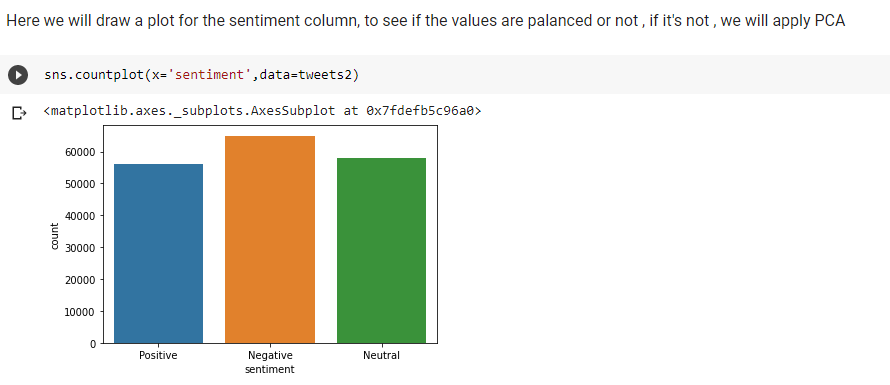
Lambda functions are useful when working on a collection of elements through higher-order functions, which can take one or more functions as arguments. This function is then invoked on the collection. When combined with Pandas functions such as.map (). apply (), or. applymap (), a Lambda function can be a powerful tool to derive new values.

* **polarity scores function**

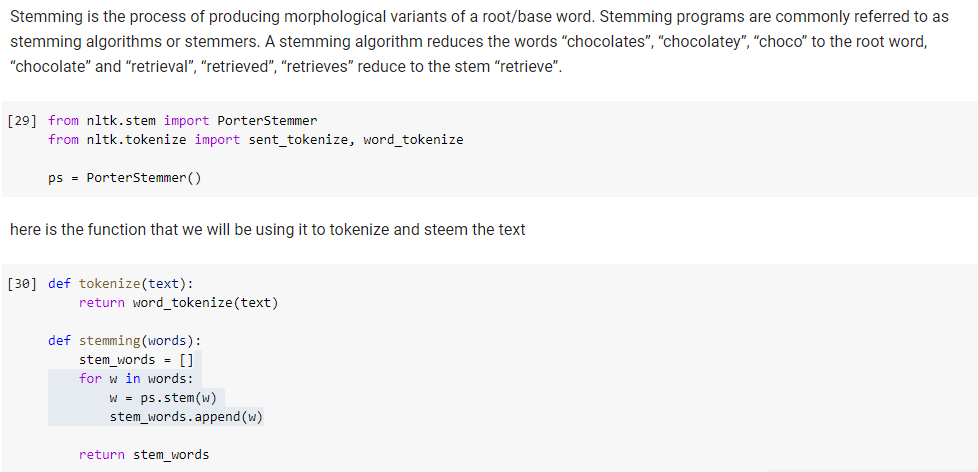
****

****

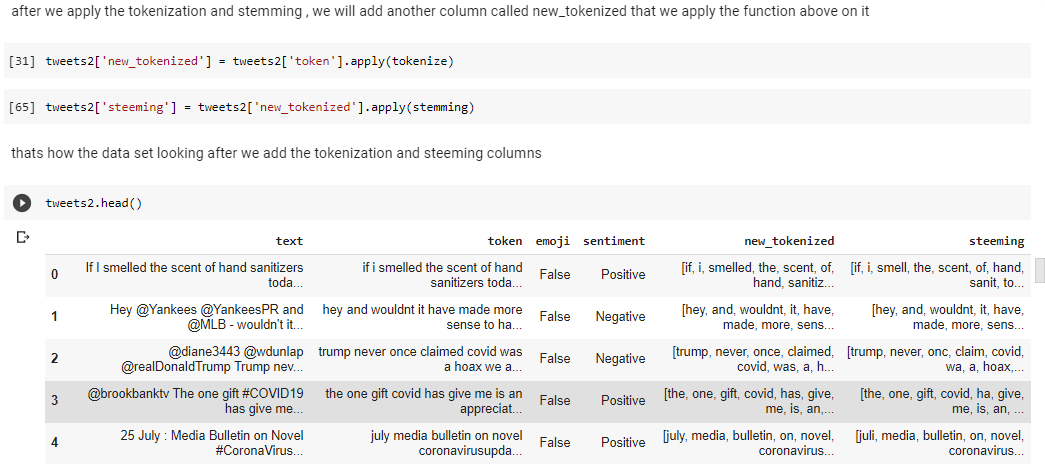
* **plot for the sentiment column**

****

* **Tokenize and Steeming**

****

**add the tokenization and stemming columns**

****

# Wordcloud

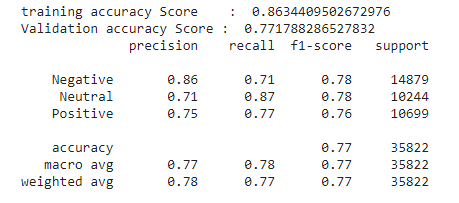
****

# Classification

**Experiment 1: Naive Bayes classifier**

The multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts. However, in practice, fractional counts such as tf-idf may also work.

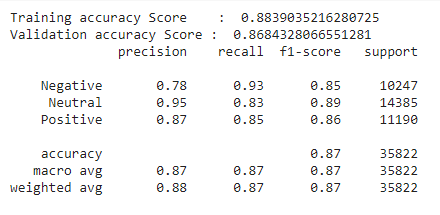
After we apply multinomial Naive Bayes classifier, we will see the confusion matrix and the accuracy score for this classifier.



**Experiment 2: stochastic gradient descent (SGD) classifier**

This estimator implements regularized linear models with stochastic gradient descent (SGD) learning: the gradient of the loss is estimated each sample at a time and the model is updated along the way with a decreasing strength schedule (aka learning rate). SGD allows minibatch (online/out-of-core) learning, see the partial fit method.

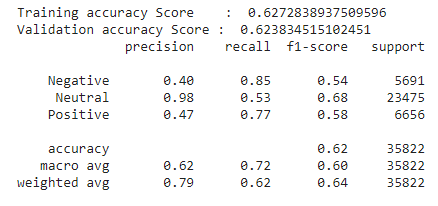
After we apply stochastic gradient descent (SGD) classifier, we will see the confusion matrix and the accuracy score for this classifier.

****

**Experiment 3: XGboost classifier**

XGBoost stands for “Extreme Gradient Boosting”. XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements Machine Learning algorithms under the Gradient Boosting framework. It provides a parallel tree boosting to solve many data science problems in a fast and accurate way.

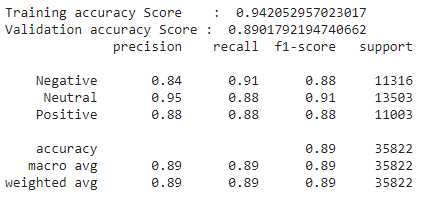
After we apply XGBoost classifier, we will see the confusion matrix and the accuracy score for this classifier.

****

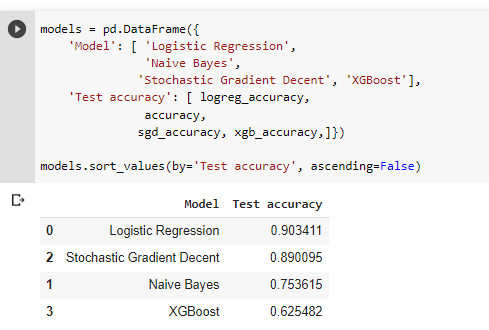
**Experiment 4: Logistic Regression classifier**

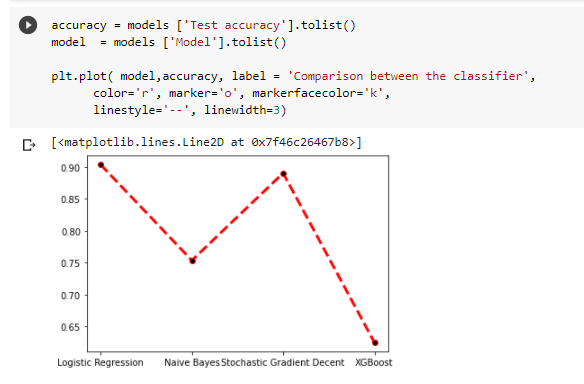
**Logistic** **regression** is a statistical model that in its basic form uses a **logistic** function to model a binary dependent variable, although many more complex extensions exist. In **regression** analysis, **logistic** **regression** (logit **regression**) is estimating the parameters of a **logistic** model (a form of binary **regression**).

After we apply Logistic Regression classifier, we will see the confusion matrix and the accuracy score for this classifier.



# Compare between the classifier

****

****