EGE UNIVERSİTY

COMPUTER ENGINEERING

COMPUTATIONAL INTELLIGENCE and DEEP LEARNING

COVID-19 VACCINE SENTIMENT ANALYSIS

PREPARED BY:

Emre ESER

Fatih GÜRKAN

Contents

1. Introduction

The development of information technology is increasing. The development of information technology is characterized by the emergence of increasingly social media such as Youtube, Facebook, Instagram, Line, Twitter and others. One of the most widely used social media is Twitter. Twitter is one of the social networking services where users can send and read text-based messages with a limit of 140 characters, known as tweets. Tweet data on Twitter can be a form of perception and opinion. The opinions posted on Twitter can contain many things, one of which is about Covid-19 pandemic. In this article we wanted to develop a model for opinions of people about covid -19 vaccines.

2. Problem and Used Methods

Sentiment analysis can be defined as a process that automates mining of attitudes, opinions, views and emotions from text, speech, tweets and database sources through Natural Language Processing (NLP). Sentiment analysis involves classifying opinions in text into categories like "positive" or "negative" or "neutral". It's also referred as subjectivity analysis, opinion mining, and appraisal extraction. The words opinion, sentiment, view and belief are used interchangeably but there are differences between them.

- Opinion: A conclusion open to dispute (because different experts have different opinions)
- View: subjective opinion
- Belief: deliberate acceptance and intellectual assent
- Sentiment: opinion representing one„s feelings An example for terminologies for Sentiment Analysis is as given below,

<SENTENCE> = The story of the movie was weak and boring

<OPINION HOLDER> =<author>

<OBJECT> = <movie>

<FEATURE> = <story>

<OPINION >= <weak><boring>

<POLARITY> = <negative>

Sentiment Analysis is a term that include many tasks such as sentiment extraction, sentiment classification, subjectivity classification, summarization of opinions or opinion spam detection, among others. It aims to analyze people's sentiments, , attitudes, opinions emotions, etc. towards elements such as, products, individuals, topics ,organizations, and services.

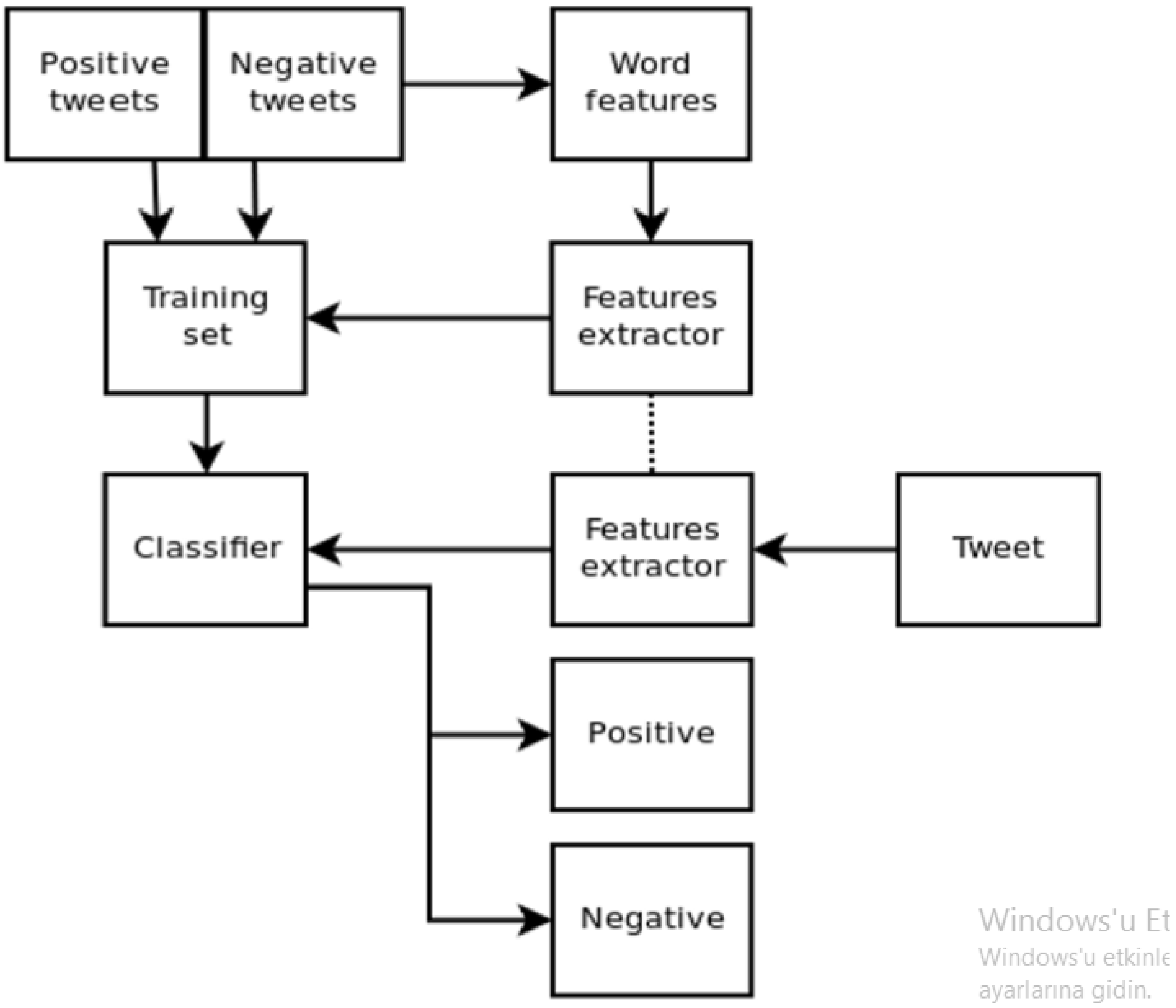


Figure 1 : Sentiment Analysis Architecture

We are developing a twitter sentiment analysis project. In this project we are analyzing tweets related to Covid-19 vaccines. There are a lot of covid-19 vaccines in the industry and people are using twitter for sharing their ideas about this vaccines. These tweets are very important for vaccine companies that manufacture these vaccines. Carefully listening to the voice of the customer on Twitter using sentiment analysis allows companies to understand their audience, keep on top of what's being said about their brand – and their competitors – and discover new trends in the industry. These companies can use these tweets about their vaccines and their competitors' vaccines and they can make decisions according to reactions of people. On the other hand these tweets are also important for other people who will choose a vaccine for avoiding from covid. People can choose the vaccine with the most positive reaction according to these tweets. We are using covid 19 related tweets for training our sentiment analyse model. We are classifying the tweets into three groups that are positive, negatife and neutral.

We did some searching and found various machine learning models that are using to classifying tweets. We are assessing different machine learning model for classifying tweets in this article :

Naive Bayes: 2.4.1 Naive Bayes: It is a probabilistic classifier and can learn the pattern of examining a set of documents that has been categorized . It compares the contents with the list of words to classify the documents to their right category or class. Let d be the tweet and c* be a class that is assigned to d, where

$$C^* = \arg mac_c P_{NB}(c \mid d)$$

$$P_{NB}(c \mid d) = \frac{(P(c))\sum_{i=1}^{m} p(f \mid c)^{n_{i(d)}}}{P(d)}$$

Figure 2 : Bayes Theorem

From the above equation, „f" is a „feature", count of feature (fi) is denoted with ni(d) and is present in d which represents a tweet. Here, m denotes no. of features. Parameters P(c) and P(f|c) are computed through maximum likelihood estimates, and smoothing is utilized for unseen features. To train and classify using Naïve Bayes Machine Learning technique ,we can use the Python NLTK library .

Support Vector Machine: Like Naive Bayes classifiers, support vector classifiers also work well for text classification and require relative few training examples. Support vector machine analyzes the data, define the decision boundaries and uses the kernels for computation which are performed in input space. The input data are two sets of vectors of size m each. Then every data which represented as a vector is classified into a class. Nextly we find a margin between the two classes that is far from any document. The distance defines the margin of the classifier, maximizing the margin reduces indecisive decisions. SVM also supports classification and regression which are useful for statistical learning theory and it also helps recognizing the factors precisely, that needs to be taken into account, to understand it successfully.

Decision Tree: Decisions tress (DTs) are the most powerful non-parametric supervised learning method. They can be used for the classification and regression tasks. The main goal of DTs is to create a model predicting target variable value by learning simple decision rules deduced from the data features. Decision trees have two main entities; one is root node, where the data splits, and other is decision nodes or leaves, where we got final output.

k-Nearest Neighbors: Neighbor based learning method are of both types namely supervised and unsupervised. Supervised neighbors-based learning can be used for both classification as well as regression predictive problems but, it is mainly used for classification predictive problems in industry.

Neighbors based learning methods do not have a specialised training phase and uses all the data for training while classification. It also does not assume anything about the underlying data. That's the reason they are lazy and non-parametric in nature.

The main principle behind nearest neighbor methods is −

- To find a predefined number of training samples closet in distance to the new data point
- Predict the label from these number of training samples.

Here, the number of samples can be a user-defined constant like in K-nearest neighbor learning or vary based on the local density of point like in radius-based neighbor learning.

Logistic Regression : Logistic regression, despite its name, is a classification algorithm rather than regression algorithm. Based on a given set of independent variables, it is used to estimate discrete value (0 or 1, yes/no, true/false). It is also called logit or MaxEnt Classifier.

Basically, it measures the relationship between the categorical dependent variable and one or more independent variables by estimating the probability of occurrence of an event using its logistics function.

In our first experiment we used Naive Bayes classifier that has 80% accuracy. All of other classifiers achieved lower accuracies.Then we decided our first experiment has low accuracy result and we decided to change our solution.

In our second experiment we tested accuracies of these ML algorytms with our dataset. The highest accuracy for these ML classifier is Support Vector Machine that has 88% accuracy and therefore we used SVM, Decision Tree has 76% accuracy, Random Forest has 80% accuracy, K-nearest neighbors chieved 71% accuracy and Logistic Regression achieved %86 accuracy. Therefore we decided to use SVM.

3. Previous Studies

Pak and Paroubek proposed a model to classify the tweets as objective, positive and negative. They created a twitter corpus by collecting tweets using Twitter API and automatically annotating those tweets using emoticons. Using that corpus, theydeveloped a sentiment classifier based on the multinomial Naive Bayes method that uses features like Ngram and POS-tags. The training set they used was less efficient since it contains only tweets having emoticons [1].

Parikh and Movassate implemented two models, a Naive Bayes bigram model and a Maximum Entropy model to classify tweets. They found that the Naive Bayes classifiers worked much better than the Maximum Entropy model [2].

Go and L.Huang proposed a solution for sentiment analysis for twitter data by using distant supervision, in which their training data consisted of tweets with emoticons which served as noisy labels. They build models using Naive Bayes, MaxEnt and Support Vector Machines (SVM). Their feature space consisted of unigrams, bigrams and POS. They concluded that SVM outperformed other models and that unigram were more effective as features [3].

Barbosa et al. designed a two phase automatic sentiment analysis method for classifying tweets. They classified tweets as objective or subjective and then in second phase, the subjective tweets were classified as positive or negative. The feature space used included retweets, hashtags, link, punctuation and exclamation marks in conjunction with features like prior polarity of words and POS [4].

In a previous study with titled Twitter Sentiment Analysis of Movie Reviews Using Information Gain and Naïve Bayes Classifier, this research aims to do the rating of a film using document analysis on user ratings and comments on the website. In this study, Naïve Bayes used the classification model and assisted by the method of Information Gain by generating an accuracy score of 82.19% from 317 data tests. From the overall class that is used both positive and negative, it can be seen that much data is neutral so that the response data is less so maximal produced [5].

Another study titled Public Services Satisfaction based on Sentiment Analysis. In this research score was achieved significantly with a lower neutral response value of the overall crawling data collected in Table I. With a value of 162 negatives, 12 positives, and 31 neutral. The resulting score appears that the data can be processed well and differ in previous research resulting in less useful data due to differences in the preprocessing phase [6].

4. Proposed Method

In this project we used SVM classifier. Like Naive Bayes classifiers, support vector classifiers also work well for text classification and require relative few training examples. Support vector machine analyzes the data, define the decision boundaries and uses the kernels for computation which are performed in input space. The input data are two sets of vectors of size m each. Then every data which represented as a vector is classified into a class. Nextly we find a margin between the two classes that is far from any document. The distance defines the margin of the classifier, maximizing the margin reduces indecisive decisions. SVM also supports classification and regression which are useful for statistical learning theory and it also helps recognizing the factors precisely, that needs to be taken into account, to understand it successfully.

We used TF-IDF method for vectorization of words. A Term frequency (TF) is a simple measurement in the weighing method. This method, every term assumed to have a proportion of interests according to the number of occurrences (emergence) in the text (document). Term frequency can improve the recall value of retrieval information, but not always fix the precision value. The Inverse document frequency, commonly abbreviated IDF, is a term for a more focused method of paying attention to the term occurrence of the whole text set. On IDF, a rare term that appears in the entire collection of text is judged to be more valuable. The value of each term interest is assumed inversely with the amount of text containing the term .

As shown in the figure 3 below we used a prepared tweet dataset from kaggle. First of all we drop out columns except tweet_text and sentiment. We converted the tweet text to lower case and then we removed unnecessary tweet words such as username, retweets, hashtags, links, punctuations, digits etc.And then we fixed contractions. (e.g we converted "i'm" to "i am") Then we removed stop words and single characters such as 'i' and 'a'. Because any of these don't help us to identifying sentiment of tweet. After all of this we completed our cleaning stage. For testing our model we pulled tweet data from twitter and we drop out columns except tweet_text. We implemented cleaning process on this tweet data and we send this data to our model for prediction and we recieved realistic results.

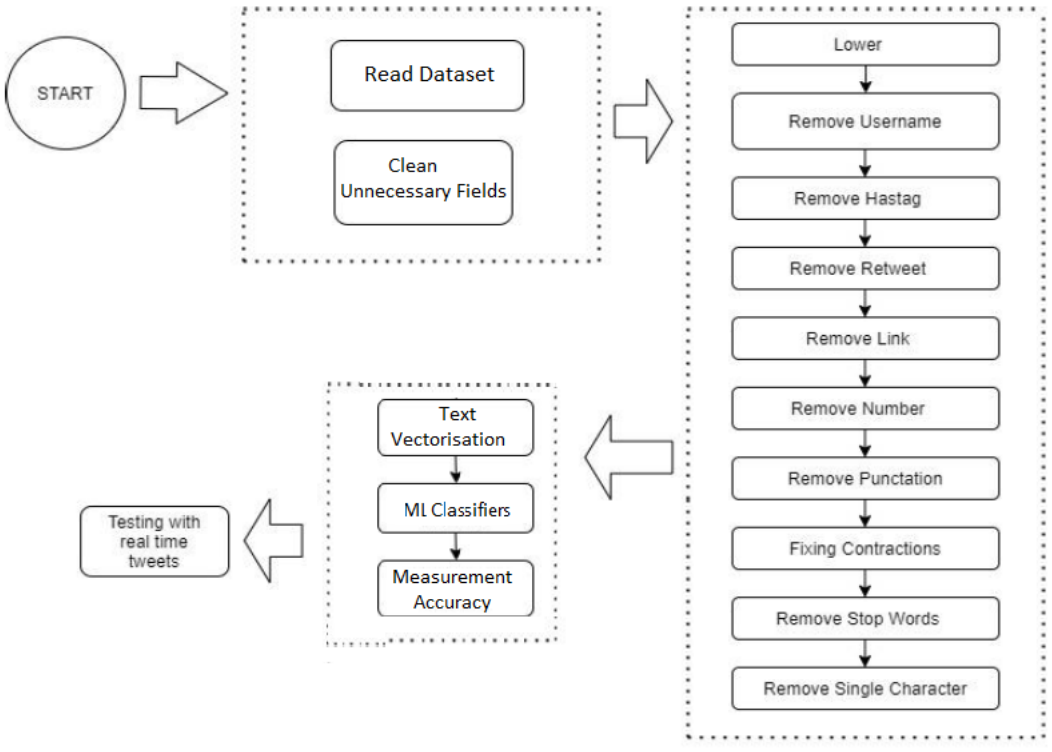


Figure 3 : The architecture of our system.

5. Experimental Studies

In our first experiment we prepared the dataset ourselves.We pulled tweets from Twitter through Tweepy library and then we labeled our tweet text with the help of TextBlob library. Firstly we extract the polarity of tweets and then according to polarity we labeled our tweets as negative , positive or neutral. Then we used TF-IDF method and we vectorized (i.e. we extracted features of) our data. And then our data become ready for training process.We splited the data as %80 training and %20 test data. We choose the Naive Bayes Classifier as classifier and then we trained the classifier with our cleaned and vectorized data. We tested the classifier real time Twitter data through twitter API and tweepy library. We saw the classifier has achieved %80 accuracy rate. But we decided to try another dataset because the accuracy rate didn't satisfy us. The Diagram of our first experiment (Figure 3) below.

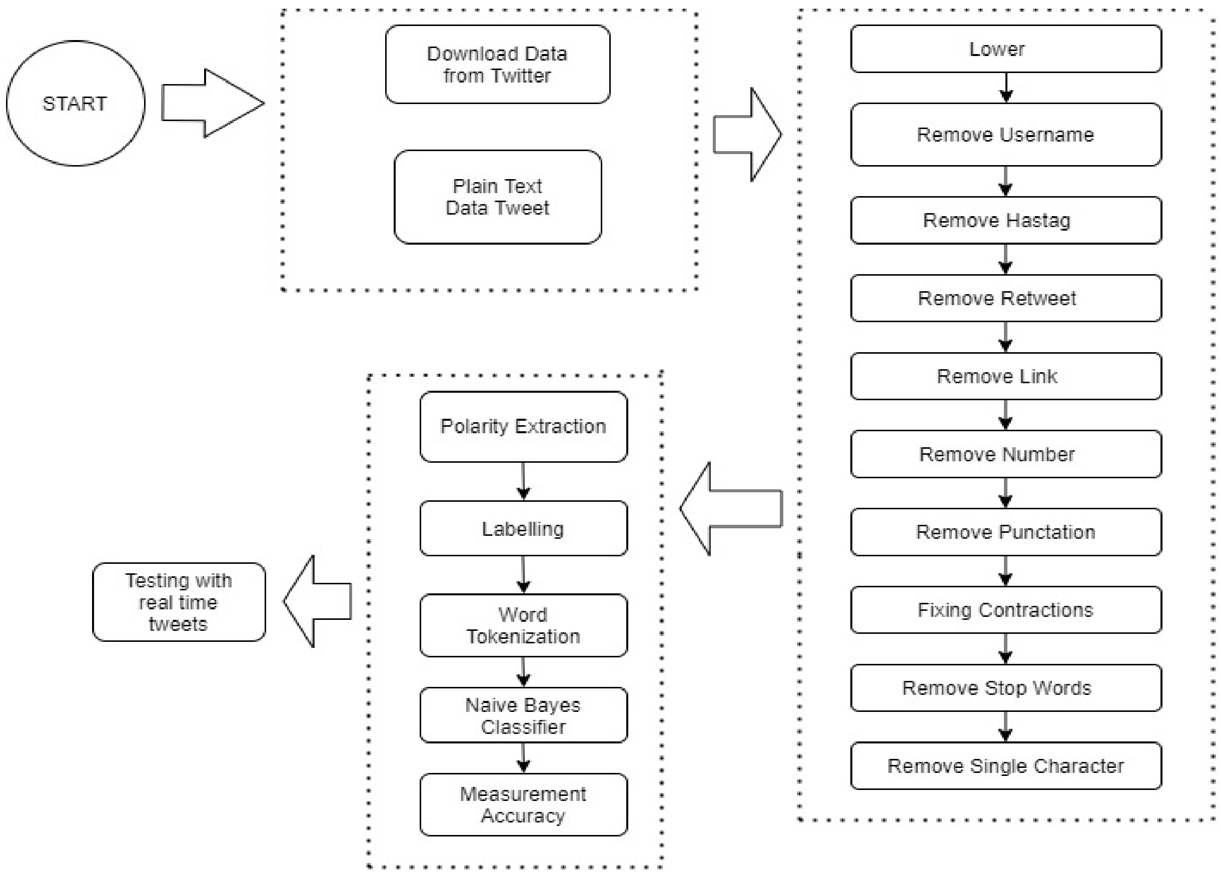


Figure 4 : The Diagram of our first experiment

In our second experiment we found Tweet Sentiment Analysis dataset from Kaggle. Process the training data consisting of various recent themes centered around current events. There are a series of sentiments given in the training data. Predict the sentiment of the test data.
https://www.kaggle.com/c/tweet-sentiment-analysis-ssn/data?select=train.csv
Dataset has 41157 tweet rows and 6 columns. Columns' values are UserName, ScreenName, TweetAt, Tweet and Sentiment. We used Tweet and Sentiment columns. But we used only sentiment and tweet columns for train our model. Sentiment column has 2 different class in the cleaned dataset : positive and negative.

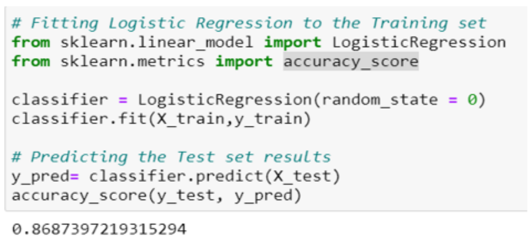
The Libraries Used :

• Python	3.8.0
• pandas	1.2.4
• numpy	1.19.5
• matplotlib	3.3.4
• re	2.2.1
• nltk	3.5
• tweepy	3.10.0
• wordcloud	1.8.1
• sklearn	0.24.1

Firstly, we have cleaned dataset for our needs. These processes are: deleting columns "UserName", "ScreenName", "Location" and "TweetAt", replacing sentiment column's values "Extremely Negative" to "Negative" and "Extremely Positive" to "Positive". We have deleted the rows whose sentiment colon had "Neutral" value.

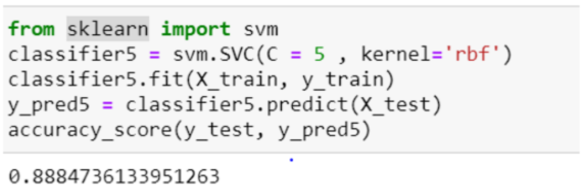
We converted the tweet text to lower case and then we removed unnecessary tweet words such as username, retweets, hashtags, links, punctuations, digits etc.And then we fixed contractions. (e.g we converted "i'm" to "i am") Then we removed stop words and single characters such as 'i' and 'a'. Because any of these don't help us to identifying sentiment of tweet. After all of this we completed our cleaning stage.

Then we used TF-IDF method and we vectorized (i.e. we extracted features of) our data. And then our data become ready for training process.We splited the data as %80 training and %20 test data. We trained data with different Machine Learning classifiers which are Logistic Regression, Random Forest, Support Vector Machine(SVM), K-NN and Decision Tree.



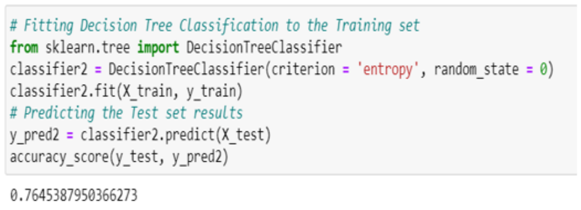
```
# Fitting Logistic Regression to the Training set
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train,y_train)

# Predicting the Test set results
y_pred= classifier.predict(X_test)
accuracy_score(y_test, y_pred)
```

```
0.8687397219315294
```



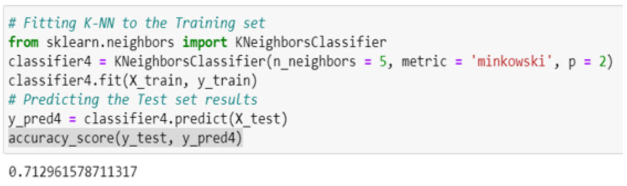
```
from sklearn import svm
classifier5 = svm.SVC(C = 5 , kernel='rbf')
classifier5.fit(X_train, y_train)
y_pred5 = classifier5.predict(X_test)
accuracy_score(y_test, y_pred5)
```

```
0.8884736133951263
```



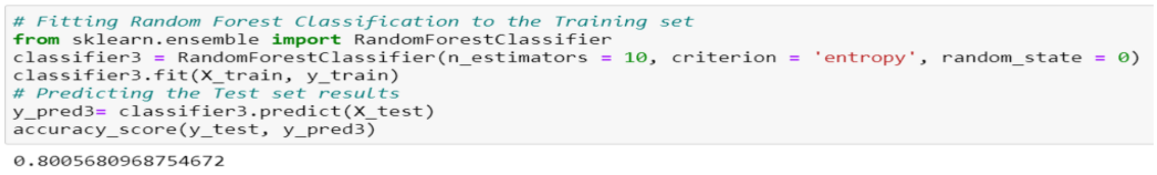
```
# Fitting Decision Tree Classification to the Training set
from sklearn.tree import DecisionTreeClassifier
classifier2 = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
classifier2.fit(X_train, y_train)
# Predicting the Test set results
y_pred2 = classifier2.predict(X_test)
accuracy_score(y_test, y_pred2)
```

```
0.7645387950366273
```



```
# Fitting K-NN to the Training set
from sklearn.neighbors import KNeighborsClassifier
classifier4 = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
classifier4.fit(X_train, y_train)
# Predicting the Test set results
y_pred4 = classifier4.predict(X_test)
accuracy_score(y_test, y_pred4)
```

```
0.712961578711317
```



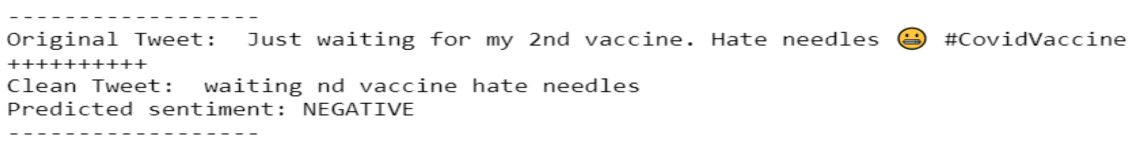
```
# Fitting Random Forest Classification to the Training set
from sklearn.ensemble import RandomForestClassifier
classifier3 = RandomForestClassifier(n_estimators = 10, criterion = 'entropy', random_state = 0)
classifier3.fit(X_train, y_train)
# Predicting the Test set results
y_pred3= classifier3.predict(X_test)
accuracy_score(y_test, y_pred3)
```

```
0.8005680968754672
```

6. Result

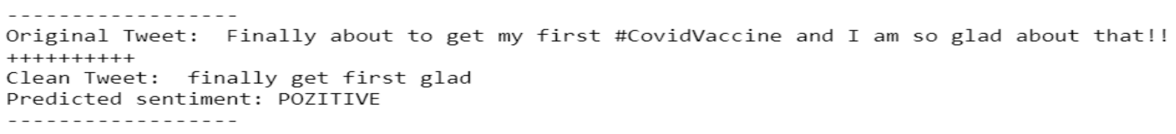
We choose the Support Vector Machine Classifier as classifier and then we trained the classifier with our cleaned and vectorized data. We saw the classifier has achieved %88 accuracy rate with dataset's test data. Also, we tested the classifier real time Twitter data through twitter API and tweepy library. Examples are below.

Negative tweet prediction:



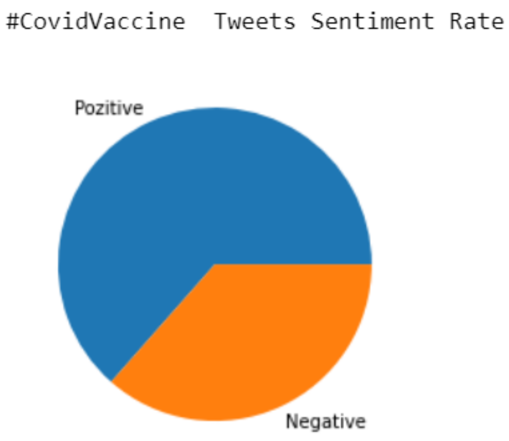
```
-------------------
Original Tweet:  Just waiting for my 2nd vaccine. Hate needles 😬 #CovidVaccine
++++++++++
Clean Tweet:  waiting nd vaccine hate needles
Predicted sentiment: NEGATIVE
-------------------
```

Positive tweet prediction:



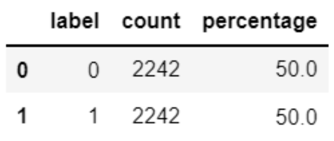
```
------------------
Original Tweet:  Finally about to get my first #CovidVaccine and I am so glad about that!!
++++++++++
Clean Tweet:  finally get first glad
Predicted sentiment: POZITIVE
------------------
```

After prediction with real time tweets(1000 tweets), we had realistic result about "#CovidVaccine" topic. These sentiment rates are below.



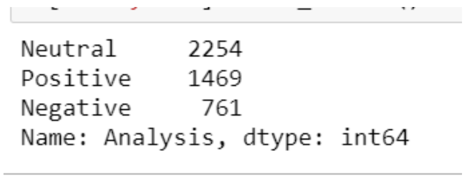
Appendix 1

Firstly, We used real time tweets from Twitter for training model. Then We used TextBlob library for labeling tweets. Finally, We tested our model with new real time tweets. Then, we compared TextBlob library with prepared tweet dataset, we decided TextBlob library is not good accuracies about our problem. Prepared tweet dataset has 2242 negative sentiments and 2242 positive sentiments(**Figure 4**) with our process. But TextBlob couldn't find same sentiment counts(**Figure 5**) from this dataset.



	label	count	percentage
0	0	2242	50.0
1	1	2242	50.0

Figure 4



```
Neutral     2254
Positive    1469
Negative     761
Name: Analysis, dtype: int64
```

Figure 5

When we found low accuracy with TextBlob, we decided about working prepared dataset for training model. We found 80% accuracy with TextBlob and Naive Bayes Classifier our first experiment. Finally we had 88% accuracy with Tweet Sentiment Analysis Dataset and Support Vector Machine Classifier our last experiment.

Appendix 2

Previous studies trained real time tweets and these accuracies are low. Firstly, we trained model with real time tweets and TextBlob library. TextBlob library is labelizer for dataset about sentiment by polarise method. Then we measured TextBlob's accuracy with prepared dataset, and result was low for us. Then we used prepared dataset for training model. Finally, we had 88% accuracy with Support Vector Machine Classifier.

In Twitter Sentiment Analysis towards COVID-19 Vaccines in the Philippines Using Naïve Bayes, the sentiments were annotated and trained using the Naïve Bayes model to classify English and Filipino language tweets into positive, neutral, and negative polarities through the RapidMiner data science software. The results yielded an 81.77% accuracy, which outweighs the accuracy of recent sentiment analysis studies using Twitter data from the Philippines [7].

In Twitter Sentiment Analysis on Coronavirus: Machine Learning Approach, the tweets were classified as positive or negative by applying the Logistic Regression algorithm, using this method we got a classification accuracy of 78.5% [8].

Appendix 3

We used prepared codes which are get tweets from Twitter(https://fairyonice.github.io/extract-someones-tweet-using-tweepy.html), English word contractions for sentences(https://stackoverflow.com/questions/19790188/expanding-english-language-contractions-in-python), regexes for cleaning tweet sentences and Machine Learning libraries to use. We used these because these are necessary for our project.

Appendix 4

Here are 10 of the most important things we learned:

- Fuzzy Logic : Fuzzy logic is an approach to computing based on "degrees of truth" rather than the usual "true or false" (1 or 0) Boolean logic on which the modern computer is based.
- Evolutionary computation : In computer science, evolutionary computation is a family of algorithms for global optimization inspired by biological evolution, and the subfield of artificial intelligence and soft computing studying these algorithms. In technical terms, they are a family of population-based trial and error problem solvers with a metaheuristic or stochastic optimization character.
- Swarm intelligence (SI) : is the collective behavior of decentralized, self-organized systems, natural or artificial. The concept is employed in work on artificial intelligence.
- Gradient Descent : Gradient descent is an optimization algorithm for minimizing the cost. To think of it intuitively, while climbing down a hill you should take small steps and walk down instead of just jumping down at once. Therefore, what we do is, if we start from a point x, we move down a little i.e. delta h, and update our position to x-delta h and we keep doing the same till we reach the bottom. Consider bottom to be the minimum cost point.
- Dropout : Dropout is a regularization technique which prevents over-fitting of the network. As the name suggests, during training a certain number of neurons in the hidden layer is randomly dropped. This means that the training happens on several architectures of the neural network on different combinations of the neurons. You can think of drop out as an ensemble technique, where the output of multiple networks is then used to produce the final output.
- RNN(Recurrent Neural Network) : Recurrent neural networks are used especially for sequential data where the previous output is used to predict the next one. In this case the networks have loops within them. The loops within the hidden neuron gives them the capability to store information about the previous words for some time to be able to predict the output. The output of the hidden layer is sent again to the hidden layer for t time stamps.The output of the recurrent neuron goes to the next layer only after completing all the time stamps. The output sent is more generalized and the previous information is retained for a longer period.
- Batch Normalization : As a concept, batch normalization can be considered as a dam we have set as specific checkpoints in a river. This is done to ensure that distribution of data is the same as the next layer hoped to get. When we are training the neural network, the weights are changed after each step of gradient descent. This changes the how the shape of data is sent to the next layer.
- Backpropagation : When we define a neural network, we assign random weights and bias values to our nodes. Once we have received the output for a single iteration, we can calculate the error of the network. This error is then fed back to the network along with the gradient of the cost function to update the weights of the network. These weights are then

updated so that the errors in the subsequent iterations is reduced. This updating of weights using the gradient of the cost function is known as back-propagation.

- LSTM : Long short-term memory is an artificial recurrent neural network architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points, but also entire sequences of data.

Appendix 5

We learned cleaning dataframe fields, cleaning tweet's sentences, to vectorise text(TF-IDF), using different machine learning classification techniques(K-NN,Decision Tree,SVM,Random Forest,Logistic Regression,Naive Bayes) for Sentiment Analsis training. We learned using Tweepy library for get real time tweets from Twitter.

References

[1] A.Pak and P. Paroubek. „Twitter as a Corpus for Sentiment Analysis and Opinion Mining". In Proceedings of the Seventh Conference on International Language Resources and Evaluation, 2010, pp.1320-1326
[2] R. Parikh and M. Movassate, "Sentiment Analysis of User- GeneratedTwitter Updates using Various Classi_cation Techniques",CS224N Final Report, 2009
[3] Go, R. Bhayani, L.Huang. "Twitter Sentiment ClassificationUsing Distant Supervision". Stanford University, Technical Paper,2009
[4] L. Barbosa, J. Feng. "Robust Sentiment Detection on Twitterfrom Biased and Noisy Data". COLING 2010: Poster Volume,pp. 36-44.
[5] S. Widya Sihwi, I. Prasetya Jati, and R. Anggrainingsih, "Twitter Sentiment Analysis of Movie Reviews Using Information Gain and Naïve Bayes Classifier," Proc. - 2018 Int. Semin. Appl.
[6] E. Susilawati, "Public services satisfaction based on sentiment analysis: Case study: Electrical services in Indonesia," 2016 Int. Conf. Inf. Technol. Syst. Innov. ICITSI 2016 - Proc., 2017.
[7] Charlyn Villavicencio, Julio Jerison Macrohon, X. Alphonse Inbaraj ,Jyh-Horng Jeng and Jer-Guang Hsieh "Twitter Sentiment Analysis towards COVID-19 Vaccines in the Philippines Using Naïve Bayes" 2021
[8] Cristian R. Machuca, Cristian Gallardo and Renato M. Toasa "Twitter Sentiment Analysis on Coronavirus: Machine Learning Approach", 2020

* https://scikit-learn.org/stable/