# Programming Assignment 3

# Stance Classification

**Dipti Aswal**a**, Atefeh Aminmoghaddam**b

a Department of Computer Science and Engineering, University of Gothenburg, Sweden, gusaswdi@student.gu.se  
b Department of Computer Science and Engineering, University of Gothenburg, Sweden, gusamiat@student.gu.se

**Abstract**

Classification is a process of [categorization](https://en.wikipedia.org/wiki/Categorization), in which ideas or objects are recognized, differentiated and classified. The ability of text classification on a tagged dataset or without it makes it powerful in many applications. In this report, we trained three different classifiers on a dataset containing comments on Covid-19 vaccination which were labeled as pro vaccination or anti vaccination. According to the results, the Support Vector classifier has a higher cross validation score. The accuracy of this model on the test dataset is 90% that shows the model performed quite well in identifying the right sentiments of the users in support or against covid-19 vaccination.

**Introduction**

Text classification problems have become quite popular over the last few years from spam filtering, topic labelling to sentiment analysis. Different kinds of methods are suggested for classifying text into categories dependent on the problem definition. Below are some of text classification applications:[[Gupta](https://dzone.com/users/3130803/shanky238.html), 2018]

* Tagging content or products using categories as a way to improve browsing or to identify related content on your website
* Text classification can also be used to automate CRM tasks
* Text classification of content on the website using tags helps Google crawl your website easily, which ultimately helps in SEO
* A faster emergency response system can be made by classifying panic conversation on social media
* As marketing is becoming more targeted every day, automated classification of users into cohorts can make the marketer’s life simple
* Academia, law practitioners, social researchers, government, and non-profit organizations can also make use of text classification technology

In the problem of this report binary classifiers are desired and we used three different prediction models to classify the comments.

**Methodology**

In this section, we describe the data used in this paper, explain preprocessing techniques used to prepare the data for analysis, and provide the statistical prediction techniques used in this paper. Fig. 1 shows the framework that this report used.



Figure 1: Report Framework

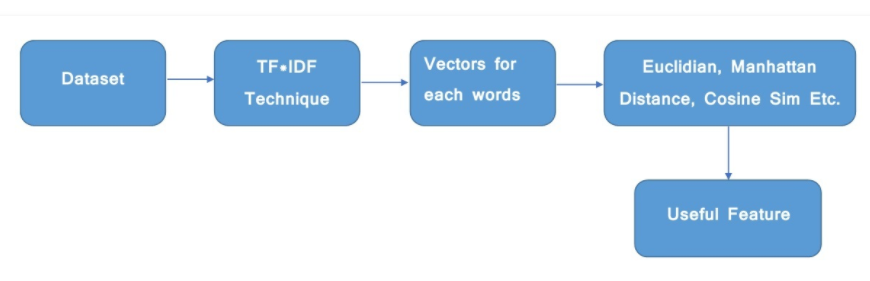
**Data**

The data used in our report is obtained from a University assignment for which each of the students collected at least 100 comments in English relating to COVID-19 vaccination from social media or the comment fields from online articles. The students were supposed to annotate the comments whether it is pro-vaccination (represented as 1 in the spreadsheet) or anti-vaccination (0 in the spreadsheet). Then the comments were sent to other students to annotate them as pro-vaccination(1), anti-vaccination(0) or in a case that they don't know (-1).

Finally, a data set of comments about the COVID-19 vaccination was formed with pro or anti-annotations.

**Feature Extraction**

It’s a well known fact that a machine learning model is as good or bad as the data fed to it so it's important to select only the relevant features that contribute to the quality of the model by reducing computation time and improving model performance. In this problem, we leverage TF-IDF ( Term Frequency, Inverse Document Frequency) technique for information retrieval. It converts raw data to vectors such that each word has its own vector and then performs certain techniques to retrieve the most useful features from them.

Figure 2: TF-IDF workflow for feature extraction.[[Dahiya](https://auth.geeksforgeeks.org/user/PulkitDahiya/articles), 2019]

The Tf-idf value increases as the frequency of a word increases in a sentence/instance but it reduces if the same word appears frequently in the entire document. This is to emphasise on the fact that few words in general appear frequently in a document. Based on the **tf-idf score** the most useful features are calculated. Below are few important features from one of the comments:

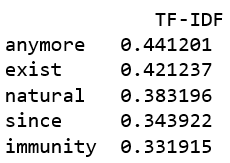


Figure 3: TF-IDF score of a Text

**Reliability Measure of Human Judgement**

In Crowdsourcing, the data annotated by a single annotator may be prone to error and hence is unreliable, so many annotators are required to label a document. Due to which the need for consensus comes into play. Consensus or overlap, measures the consistency and agreement amongst a group and is calculated by dividing the sum of agreeable data labels by the total number of labels. The idea is to arrive at the consensual decision for each data item.

In order to find consensus between annotators ***Inter-annotator agreement* (IAA)** is calculated by calculating precision, recall, f-measure and Cohen’s kappa between two annotators. A small value of IAA would indicate that the labels are very different. Cohen’s kappa coefficient(κ) calculates the chance of two annotators giving the same label to a text. It is defined using observed agreement (Ao) and agreement expected by chance (Ae). The observed agreement Ao (or percentage of agreement) is the number of instances the annotators agree on divided by the total number of instances.

Another measurement is ***Intra-annotator agreement***where the same annotator is made to perform the labelling activity twice on the same text after a certain time interval and observe the difference.[Deleger, Li, Lingren, Kaiser, Molnar, Stoutenborough, Kouril, Marsolo & Solti, 2012]

In this project, the comments are labeled by two or more students independently so we used Cohen's kappa coefficient(κ) to calculate the agreement of the labels. The value that is calculated by the sklearn library in python is **0.88**. As the value is relatively high, we can conclude that the labels are not very different and the prediction modeling is quite **reliable** based on the labels.

**Preprocessing**

In Natural language Processing tasks usually the data set is not perfectly clean and we need to clean the data before starting the prediction models construction. The importance of this step is increasing the performance of the algorithms. We convert the data from its raw form to a matrix of TF-IDF features using TfidfVectorizer and before feeding data to TFDFVectorizer preprocessing is done.

Below are the steps we followed to clean the data:

* Tokenize into alphabetic tokens
* Discard numbers and punctuations
* Lemmatize a sentence with the appropriate Part Of Speech (POS) tag that maps a word to an adjective, noun, verb or adverb.
* Remove Stop words
* Remove tokens that are too rare. Assume 20 for the current dataset.
* Remove tokens that occur too frequently, Assume 80% in this case.
* Derive truth labels from multiple labels. In this report, we consider the most commonly occuring annotation for each instance/row.

**Data Balancing**

Prediction results can be biased toward the majority class when there is a significant difference between the two classes of the target variables (James, Witten, Hastie, & Tibshirani, 2013). This bias can be mitigated using oversampling or undersampling techniques to balance the data. The balanced data is then used to train the prediction model. Note that an imbalanced data reduces the effectiveness of classification techniques, and results in a low accuracy of the prediction model (James et al., 2013).

So as a main step before starting the main part of the analysing data, we investigate whether the data is balanced. The result shows that in both train dataset and test data set, the data is balanced in two categories (Pro and Anti-vaccination sentiments).

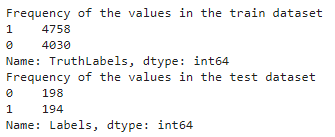


Figure 4: Number of rows/instances for each label

**Prediction Models**

In this section, as the final step in the text classification, we train a classifier using the features created in the previous step. There are many different choices of machine learning models which can be used to train a final model. We will implement following different classifiers for this purpose:

* Multinomial Naive Bayes Classifier

The Naïve Bayes classifier is a probabilistic machine learning model based on Bayes theorem which assumes that the features are not interdependent. Multinomial Naive Bayes is a form of NB classifier which is more suited for classification with discrete features (e.g., word counts for text classification). In this classifier all features follow multinomial distribution including TF-IDF. In this report we consider the multinomial naive bayes classifier as the trivial baseline model with an accuracy score of **~78%**.

* Random forest

Random forest performance is better than most of the classification algorithms due to the fact that it is based on ensemble and bagging technique. It is also stable because if a new data point is added only one tree is impacted. In this paper we aim to achieve highest accuracy by way of tuning hyperparameters namely max\_depth. Even after modifying the hyperparameters we did not get a decent score here.

* Support Vector Machines

Support Vector classifier is a non-parametric model which does not make any assumption about the shape of the data. This is quite versatile in nature as it uses kernelization internally to maximise the margin from the closest class points by creating hyperplanes. This supports linear as well as nonlinear boundaries. The results in this case were promising **~ 79% (Training set).** This score was comparable to the accuracy of the trivial baseline.

**Results**

**Cross validation scores** for 3 models:





Figure 5: Cross validation scores for all 3 models

Evaluated **Test set** on SVC to achieve **~90% accuracy**:

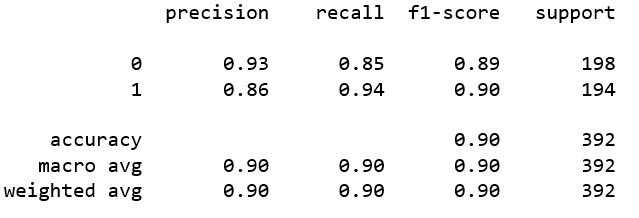


Figure 6: Accuracy score of SVC on Test data

As seen in the confusion matrix, the “*True positives*” and “*True negatives*” counts indicate that 90% of the data is correctly classified. However, the remaining ~10% constituting of “*False Positives*” and “*False Negatives*” are mis-classified as pro-vaccination(1) labels were wrongly classified as anti-vaccination(0) and vice-versa.

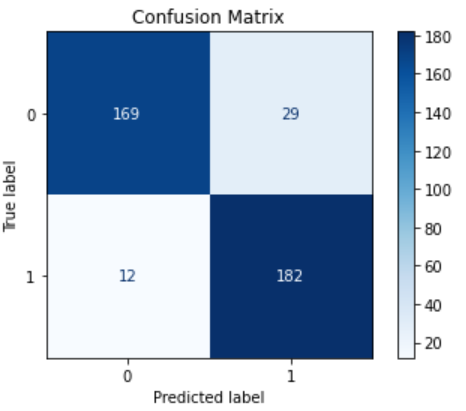


Figure 7: Confusion Matrix for True Vs Predicted labels

**Conclusion**

With a high score it can be concluded that the model performed quite well in identifying the right sentiments of the users in support or against covid-19 vaccination. In order to improve on the score we could try to improve the reliability of the annotations by setting clear guidelines for all the annotators so that their labels are in sync. Also, in the absence of truth labels we assumed most frequently occurring output values as the gold standard for labels so there could be some deviations due to that.

**References**

Deleger, Louise & Li, Qi & Lingren, Todd & Kaiser, Megan & Molnar, Katalin & Stoutenborough, Laura & Kouril, Michal & Marsolo, Keith & Solti, Imre. (2012). Building Gold Standard Corpora for Medical Natural Language Processing Tasks. AMIA . Annual Symposium proceedings / AMIA Symposium. AMIA Symposium. 2012. 144-153.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning, Vol. 112. Springer.

[Dahiya](https://auth.geeksforgeeks.org/user/PulkitDahiya/articles), [P](https://auth.geeksforgeeks.org/user/PulkitDahiya/articles) 2019,Sklearn | Feature Extraction with TF-IDF, <<https://www.geeksforgeeks.org/sklearn-feature-extraction-with-tf-idf/>>

[Gupta](https://dzone.com/users/3130803/shanky238.html), Sh 2018,Text Classification: Applications and Use Cases, <[Text Classification: Applications and Use Cases - DZone AI](https://dzone.com/articles/text-classification-applications-and-use-cases)>