Trade-off Between Accuracy and Performance For Text Classification Using Ensemble Models in Machine Learning

AXEL GORIS RAFAT KHAN

goris | rafatk @kth.se

October 24, 2022

Abstract

Machine learning algorithms have their limitations and producing a model with high accuracy is challenging. In many cases, one model is not enough for producing a reliable prediction because of the high variance of the input training data which might result in low accuracy as the model relies heavily on too few features when making a prediction. Such a situation arises in the case of text classification. Instead, if we build and combine multiple models, we have a chance to boost the overall accuracy. The combination of models will reduce the overall ensemble model's error and maintain the generalization of the model by aggregating the output from each model. In an ensemble learning architecture, the weak learners receive inputs as well as predictions from all other weak learners. A final ensemble model can be built using the combined predictions. The final ensemble models and the single models will be evaluated on text sentiment analysis which is one area where the input data variance is high. We used k-Nearest Neighbors (kNN), Naive Bayes and XGBoost for the models and compared the results with both Stacking and Max-Voting. The final ensemble model using Stacking improves the accuracy by 4% but the time costs increases by more than 50% and the accuracy result was not significantly different between kNN and Stacking. Overall this result contributes to the comparison of different machine learning models.

Contents

1 Introduction **3**
1.1 Background and Rationale 3
1.2 Ethics and Sustainability 3
1.3 Outline 4

2 Theoretical framework/literature study **4**
2.1 Models 4
2.1.1 K-Nearest Neighbors 4
2.1.2 Naive Bayes 4
2.1.3 Logistic Regression 4
2.1.4 XGBoost 5
2.2 Ensemble models 5
2.2.1 Max-voting 5
2.2.2 Stacking 5
2.3 Metrics 5
2.3.1 Confusion Matrix 5
2.3.2 Macro Average 6
2.4 Related Work 6

3 Research questions, hypotheses 7
3.1 Hypotheses 7
3.2 Research Questions 7

4 Research Methodology 7
4.1 Methods 7
4.2 Approach 7
4.3 Strategy 8
4.4 Data Collection and Analysis 8

5 Data Set And Models 8
5.1 Analysis of our data set 8
5.2 Pre-Processing 9
5.3 Models 11

6 Results and Analysis 11
6.1 Results 11
6.2 Analysis 12

7 Discussion and Conclusion 12
7.1 Discussion 12
7.1.1 Validity 12
7.1.2 Variance 12
7.1.3 Hyper-parameter tuning 13
7.1.4 Speed-Accuracy Trade-Off 13
7.1.5 Ensemble methods 13
7.1.6 Time and resource consumption on training 13
7.1.7 Using combination of weak and strong learners as base model 13
7.2 Conclusion 14

A Data Analysis 15

List of Acronyms and Abbreviations

kNN k-Nearest Neighbors

NB Naive Bayes

SDG Sustainable Development Goals

UN United Nations

XGBOOST eXtreme Gradient Boosting

1 Introduction

Texts are extremely rich sources of information. But the unstructured nature of the text makes it difficult and time-consuming to extract insights from it. Text classification techniques have made the task more convenient. Text classification is a machine learning technique for identifying predefined categories in open-ended text.

1.1 Background and Rationale

Machine learning algorithms have been used to classify texts for a long time[1]. Although, Support Vector Machines[2], Naive Bayes (NB)[3], eXtreme Gradient Boosting (XGBOOST)[4] and kNN[5] algorithms can classify texts, none of these produce high accuracy for text classification. To solve this and many other problems (e.g. variance, bias), the concept of model ensembling was introduced. In machine learning, ensemble models combine multiple other models for prediction. By using ensemble models, we can overcome the technical challenges of building a single estimator. An ensemble model can be built by stacking, blending, boosting, and bagging [6].

The goal of this research is to determine if ensemble model can help with accuracy without degrading performance in the case of text sentiment classification. Text sentiment classification is the process of analysing a text and labeling it as positive, neutral, or negative. In our case, we will also consider extremely positive and extremely negative sentiment [7].

1.2 Ethics and Sustainability

Ethically, we have multiple issues to take into account. The issues are the following:

- Data source : We need to make sure that there is no bias in the kind of data we are selecting. If we only have one type of text, our classifier might perform well but it will have a strong bias and will not be suitable for anything different (e.g. field, data set). Our conclusions will be biased and we will not be able to help science progress because our results will not be realistic.

- Transparency : We need to carefully explain and develop each step so that anyone can reproduce our results. Moreover, the data set we are going to use must be public so that everyone has access to it.

- Uncertainty : This is linked to the data source. We need to make sure that what we have done is correct and not too specific.

For Sustainability, we will refer directly to the United Nations Sustainability Goals [8]. More specifically:

- Quality Education - Sustainable Development Goals (SDG) 4 : By having access to better classification of documents, it gets easier for everyone to quickly find what they are looking for e.g. students in a library for example.

- Industry, Innovation & Infrastructure - SDG 9 : Our project is part of the innovation process, we are trying to build on what has been done before so that the machine learning algorithms get better and better. This way, industry will be able to use them in more diverse situations.

- Responsible Consumption & Production - SDG 12 : By having access to better classification, we can find informations quicker reducing time wasted. At the same time, if there is a decrease in performance, that might make our project unsuitable regarding the United Nations (UN)'s Goals; hence we need to carefully examine the trade-off between accuracy and performance.

1.3 Outline

The following is the outline for this report:

- Section 2 deals with the theoretical background, including key concepts and related work.
- Section 3 deals with research questions and the hypotheses that we have.
- Section 4 deals with our research methodology, our data collection, choice, and pre-processing of our data set.
- Section 5 deals with the results and the analysis of the project.
- Section 6 discusses the results and provides a conclusion to the study.

2 Theoretical framework/literature study

Accuracy and reliability can be more important than cost in different domain of text mining. To ensure maximum accuracy, we can use ensemble models. By combining multiple models instead of using just one, ensemble methods aim to improve the accuracy of results in models. The idea is that combining the models significantly increases the accuracy of the results. As a result, ensemble methods have gained popularity in machine learning.

2.1 Models

A machine learning model is an algorithm that has been trained to recognize patterns in a type of data. Once it has been trained, it can be used to classify or evaluate new data of the same type. It is very useful for large data set where the definition of strict rules is too complicated to lead to good predictions. However, you need a large amount of data if they are labelled (supervised learning [9]) and even more if they are not labelled (unsupervised learning [9]).

2.1.1 K-Nearest Neighbors

kNN is a supervised non-parametric learning model. To make a prediction, the kNN algorithm will base itself on the entire data set [5]. Indeed, for an observation, which is not part of the data set, that we want to predict, the algorithm will look for the K instances closest to our observation. Then for these K neighbors, the algorithm will use their output variables y to calculate the value of the variable y of the observation we want to predict.

A kNN model depends a lot on the value of K: K is the number of neighboors to consider for classification. A low value of K means that our model will have a low Bias and a High Variance (risk of overfitting). A high value of K means that our model will have a high Bias and a low Variance.

2.1.2 Naive Bayes

The Naive Bayesian classification method is a supervised machine learning algorithm that classifies a set of observations according to rules determined by the algorithm itself [3]. It is a probabilistic classifier, using Bayes' theorem [10], which is highly scalable and can achieve high level of accuracy. However, the features need to be independent (Naive assumption).

2.1.3 Logistic Regression

Logistic Regression is a supervised machine learning model [11]. It is used to predict a binary outcome thanks to prior observations of a data set of independent variables. The outcome is a probability between 0 and 1.

2.1.4 XGBoost

XGBOOST stands for Extreme Gradient Boosting and is a scalable, distributed gradient-boosted decision tree (GBDT). It is using parallel tree boosting to achieve high accuracy and high performance [4]. It is also an ensemble model of multiple decision tree but this will not be used to ensemble our data.

2.2 Ensemble models

An ensemble model is a method to train two or more models [12]. These models can be different or trained on different data, and the ensemble model aggregates the result and turn it into a final score, in order to increase accuracy or performance. One of the main downside is that the output of ensemble models are harder to explain and variations or impact of parameters is harder to calculate. Moreover, the performance is often worse than simple models.

Models can be ensembled in many different ways. We can implement these techniques using simple ensemble techniques, where we can ensemble the result produced by different models using Max-voting, Averaging, or Weighted Averaging or we can use Advanced Ensemble techniques that uses predictions from multiple models to build a new model by Stacking, Blending, Bagging, or Boosting [13].

Ensemble learning algorithms such as stacking are meta-learning algorithms or meta-learners that learn how to combine predictions from ensemble members. A meta-learning algorithm also learns how to learn across multiple related prediction tasks, sometimes referred to as multi-task learning.

2.2.1 Max-voting

Max-voting is an ensemble model and is one of the easiest ways to combine predictions from multiple machine learning algorithms for classification problems [14]. Max-voting involves each base model making a prediction and voting on each sample. The final predictive class includes only the sample class with the highest number of votes.

2.2.2 Stacking

In stacking, an ensemble model, a meta-learning algorithm is used to learn how to combine predictions from two or more machine learning algorithms known as base models [14]. Base models are fitted on the full dataset while meta model is trained using cross-validated predictions of the base estimators using the cross-validation prediction. Stacking offers the advantage of combining the capabilities of a range of well-performing models on a classification or regression task and generating predictions that are better than any single model. In stacking, the models are typically different and fit on the same data set instead of samples of the training data set. As a meta-learner, a single model (Meta-learner Model) is used to learn how to best combine the predictions from the contributing models (Base Models).

2.3 Metrics

This subsection presents the metrics we will use to quantify our results.

2.3.1 Confusion Matrix

A confusion matrix is a table that is used to define the performance of a classification algorithm. We can visualize and summarize the performance of the algorithm by seeing this confusion matrix.

Considering we have a multi-class problem we will consider each class as a binary classification problem. It is whether right or wrong for each class.

In order to calculate some properties, we will use the concept of True Negative, True Positive, False Negative and False Positive that are illustrated in Table 1.

Table 1: Confusion Matrix Explanation

		Real Value	
		True	False
Predicted	True	True Positive (TP)	False Positive (FP)
	False	False Negative (FN)	True Negative (TN)

2.3.2 Macro Average

We will be interested in the Macro average of each class to have more of a macro point of view for the 4 following properties.

Accuracy : Measure of how many times our model made a correct prediction. Good metric to measure the overall performance of our model.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

Precision : Measure of how many positive predictions were True Positive.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{2}$$

Recall : Measure of how many positives were correctly predicted, over all positive cases.

$$\text{Recall} = \frac{TP}{TP+FN} \tag{3}$$

F1-Score [15] : Harmonic mean of Precision and Recall, helping balancing the two previous metrics.

$$\text{F1-Score} = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

MacroAverage : Average on each class for each of the F1-scores. Example with $m = NumberOfClasses$:

$$\text{MacroAverageF1} = \frac{\sum_{n=1}^{m} F1_n}{m} \tag{5}$$

2.4 Related Work

S. Jayaprakash and P. Balamurugan published a paper proposing text classification based on features selection and pre-processing, and thereby reducing the dimensionality of the Feature vector and increasing the classification accuracy [16]. This work gave us the idea to use Multiple-Model Techniques [17] to further enhance the text classification performance. This concept of ensembling model is used for different purposes. The ensemble forecasting approach combines different data sources, models of different types, assumptions, and/or pattern recognition methods. A more accurate ensemble model can be achieved by combining information from multiple sources, analyzing it with different techniques, and considering different sources of uncertainty in the real world [18]. B. Singh, N. Kushwaha, and O. Vyas proposed a scalable hybrid ensemble model for text classification by combining the bagging and boosting approach of the models [19]. This model gains its performance from using a bagged ensemble of boosted trees. In this paper, they have used a model to build their own text classifier using existing or self-modified bagging and boosting techniques. Outside of text classification, C. Gupta and D. D. Virmani proposed a model for the classification of cybercrime using ensemble learning technique [20]. The proposed model is implemented with the help of model stacking, combining Support Vector Machine, Logistic Regression, Decision Tree, and Random Forest models to give better performance resulting in 96.55 % accuracy.

3 Research questions, hypotheses

This section presents the main question which serves as a guideline for this project. Additionally, a hypothesis is presented.

3.1 Hypotheses

Our hypothesis is that ensembling machine learning models will improve accuracy by at least 5% for classifying texts. The time cost will probably increase by at least 10%.

3.2 Research Questions

Building on the hypothesis of Ensemble Models this project uses stacking with kNN, Naive Bayes, and XGBOOST algorithms to build an ensemble model for text classifications, as shown in Figure 1. The resulting ensemble model is evaluated to determine if this approach enhances the accuracy or decreases performance. We will also conduct a comparison study of the accuracy and performance.

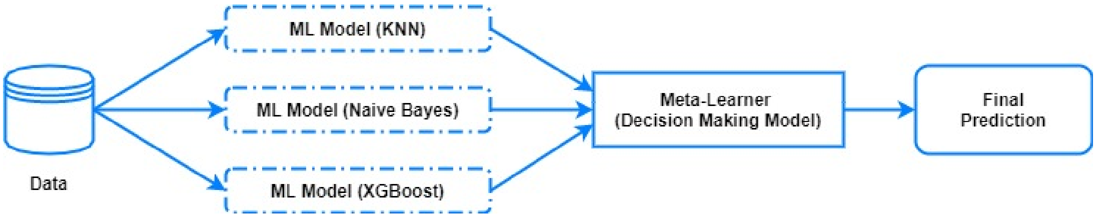


Figure 1: Architecture of proposed ensemble model.

The research questions can be defined as, whether or not, ensembling machine learning models improves accuracy for text classification and if it does, to what level and at what performance cost?

4 Research Methodology

In our project, we will use a quantitative methods since the project is about data classification and we have access to large data sets to try providing meaningful results with precise quantification. Moreover quantitative methods provide methods, approach, strategies, data collection and analysis in order to plan a good project [21].

4.1 Methods

For this project, we will specifically use an experimental research method because our project will try to study the impact of ensembling multiple models and adjusting parameters on the performance and accuracy of text classification. Therefore we will study causes and effects of using multiple models which is the main use case of Experimental research [21]. Alternative methods are descriptive, fundamental or applied but we are not searching to test theories or solve a practical problem so they are less suited for our project.

4.2 Approach

Considering the use of large data sets and the fact that our hypothesis is measurable, we will use a deductive approach and try to explain what is happening considering the real and expected outcome [21]. Another possible approach would have been an abductive approach which includes both deductive and inductive approach.

4.3 Strategy

As for the method, we will use Experimental research as our guideline considering we try to verify our hypotheses and have a large amount of raw data. We will not be doing case study or surveys so other research strategy are not as relevant for us.

4.4 Data Collection and Analysis

We need large data sets in order to establish a positive relation between ensembling models and the performance and accuracy of the model. Therefore, our project will use an experiments data collection method. In order to achieve this, we will utilize quantitative secondary data, available from Kaggle or Github. We will not be collecting data, instead we will use available data sets while still making sure that they are not biased and contain sufficient data to train, test and validate our models. Moreover, it allows for others to quickly replicate our process. Said data set must be a text data set because we are working with sentiment analysis on a text but it does not have to deal with a specific subject. The only requirements is that we use the same data set for all of our models. We have decided to use a COVID data set [22] because it has been used multiple times before and it is a manually tagged data set of Covid related tweets. The creator tagged every tweet with one of the five following tags: Extremely Negative - Negative - Neutral - Positive - Extremely Positive.

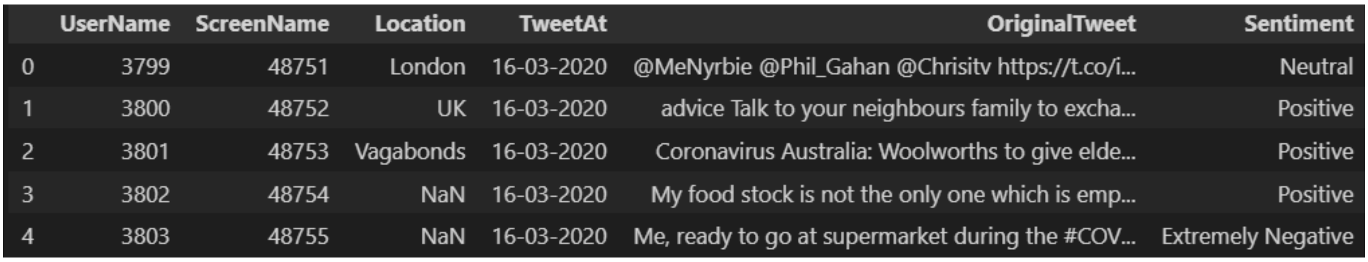
Our analysis will be conducted using a statistical method since we will not focus on the algorithm but rather their comparisons and the accuracy of the results.

5 Data Set And Models

Before starting the analysis in Python or R, we check for missing data, remove potential outliers, and considers the independence of observations. In the current form, the data set is too large for us to train our models on as we got a memory error so we will first pre-process the data. Before doing that, we will analyse our data set [23]. For the analysis and pre-processing of our data, we have decided to use Python and the library Scikit-Learn [24].

5.1 Analysis of our data set

Our data set is already split in two parts: a training data set containing 41 157 entries and a test data set containing 3 798 entries. The format of entries in the data set is shown in Figure 2 :



	UserName	ScreenName	Location	TweetAt	OriginalTweet	Sentiment
0	3799	48751	London	16-03-2020	@MeNyrbie @Phil_Gahan @Chrisitv https://t.co/i...	Neutral
1	3800	48752	UK	16-03-2020	advice Talk to your neighbours family to excha...	Positive
2	3801	48753	Vagabonds	16-03-2020	Coronavirus Australia: Woolworths to give elde...	Positive
3	3802	48754	NaN	16-03-2020	My food stock is not the only one which is emp...	Positive
4	3803	48755	NaN	16-03-2020	Me, ready to go at supermarket during the #COV...	Extremely Negative

Figure 2: Example of our data set

For the rest of the analysis, we will regroup ”Extremely Positive” and ”Positive” as ”Positive” and ”Extremely Negative” and Negative” as ”Negative”. First, we find out that the classes are not evenly distributed, with positive being bigger than the other two classes (”Negative” and ”Neutral”) (as shown in Figure 3, 4).

We found out that the classes in a tweet are related to the number of characters and words in a tweet. Moreover, the classes also depends on the average word length in a tweet A.

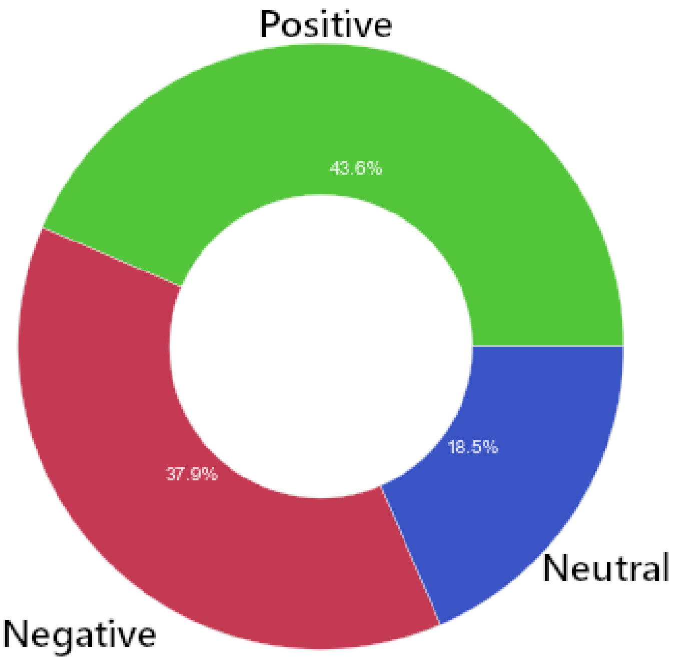


Figure 3: Classes Distribution

Some words are called ”Stopwords”: they are very present in our daily language but they do not convey useful meaning for sentiment classification. Similar to the stopwords, URLs, punctuations, mentions do not convey any valuable meaning and are very present in our data set. Thanks to the initial analysis, we know that there is a lot of data to clean up.

5.2 Pre-Processing

To try improving the sentiment analysis, we followed the steps of pre-processing of the flow 4 in [25] which provided the best accuracy for Naive Bayes sentiment analysis. The following steps have been executed to reduce the amount of data in our data set and make it easier to use:

- Change everything to lower case so that ”Covid” and ”COVID” are treated the same for example.
- Remove all of the URL and HTML characters from the tweets.
- Expand all contractions. ”Don’t” becomes ”Do not”. So on and so forth.
- Remove punctuation.
- Remove digits because they do not convey meaning for a sentiment analysis.
- Remove stopwords. They happens quite often in our data set and do not convey any valuable information.

By processing our data, we were able to use them with all of our models without running out of memory, which overcame our initial problem. Moreover, we can create some wordclouds for each classes to have an idea of what are the most used words after our pre-processing and if we can already see some differences between classes. Three of these wordclouds are shown in Figure 5.

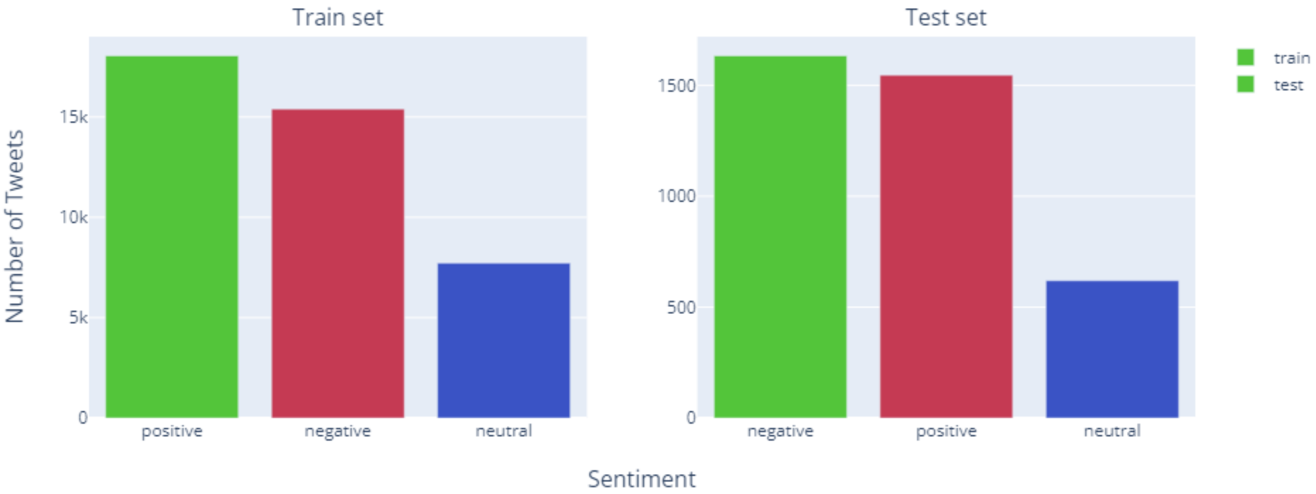


Figure 4: Classes Distribution Bar

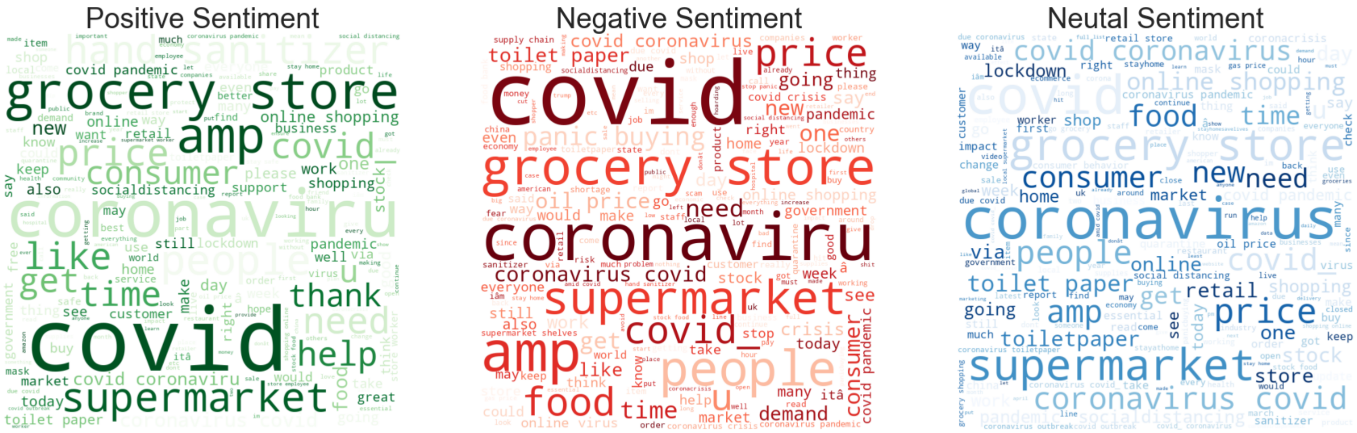


Figure 5: Words clouds depending on the tweets sentiment

5.3 Models

For the analysis, the 3 selected models are : kNN - Naive Bayes - XGBOOST. The 2 selected ensemble models are : Max-voting and Stacking. For stacking, we will use logistic regression as the meta-learner.

We will use already available implementation from Scikit-Learn [24] with the default hyper parameters except for kNN. We are using K=1 for the kNN Algorithm. On our data set, we have different result for each one of our three algorithms.

6 Results and Analysis

The analysis were conducted using F1-Score [15] to verify the validity of our tests as was done before [5].

6.1 Results

After training of our model, we tested them on the same test data set. The results are shown in Tables 2 and 3 and Figure 6. All of the training and testing have been conducted on the same machine e.g. Intel Core i7-12800H Processor (24M Cache, up to 4.80 GHz) with 16 GB RAM.

Table 2: Macro Average of our models for different measures

	Macro Average (in %)			
Model	Accuracy	Recall	Precision	F1-score
kNN	78	78	83	78
Naive Bayes	55	59	64	56
XGBoost	65	66	69	67
Max-voting	75	77	77	75
Stacking	82	82	84	83

Table 3: Training and Prediction time for our models

Model	Training Time (hh:mm:ss)	Prediction Time (hh:mm:ss)
kNN	00:02:51	00:00:48
Naive Bayes	00:06:07	00:00:40
XGBOOST	00:52:22	00:00:23
Max-voting	00:59:36	00:01:36
Stacking	06:50:16	00:01:21

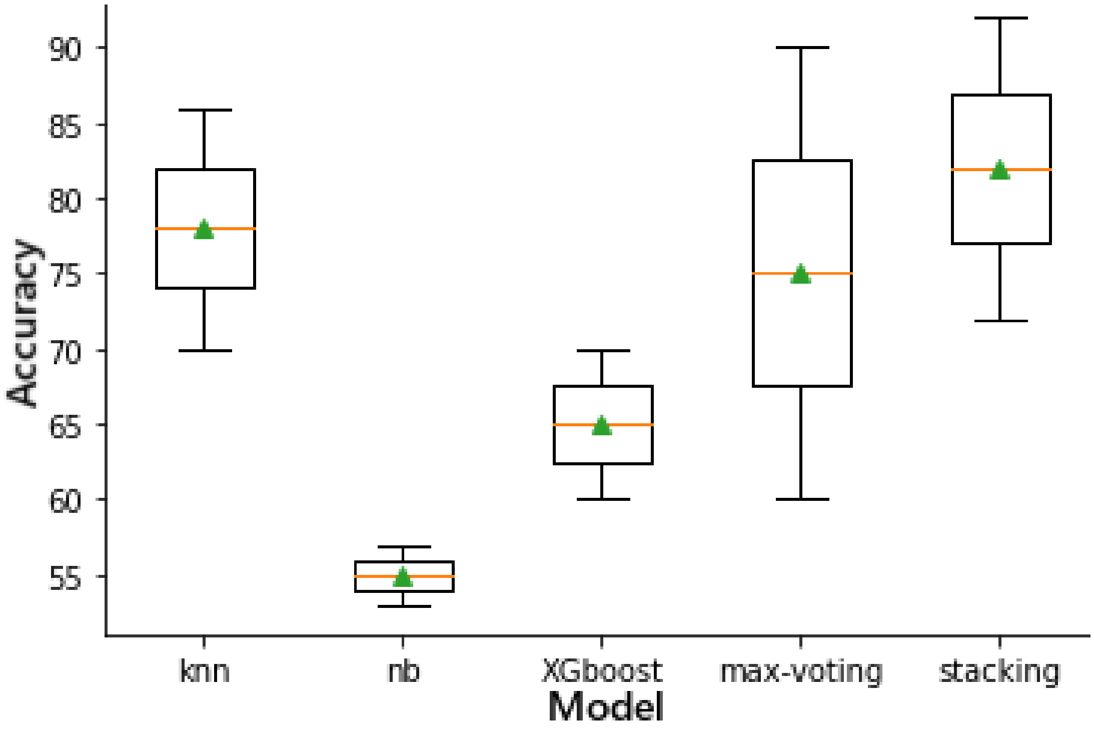


Figure 6: Box and Whisker Plot of Accuracy for the single models: kNN, Naive Bayes, XGBOOST and the ensembled models: Max-voting, Stacking.

6.2 Analysis

Our results do not follow our initial hypothese in the case of $K = 1$ for KNN. The accuracy only increased by 4% while the prediction time increased by 69% in the best accuracy scenario (Stacking vs kNN where $K = 1$). As we can see in Figure 6 in the Whisker plot, the upper and lower quartiles are further away from the average on the stacking compared to kNN meaning that the predictions are more spread in general.

7 Discussion and Conclusion

This section will discuss our results and analysis before establishing the conclusion of the project.

7.1 Discussion

The discussion will discuss the validity, the variance, hyper-parameter tuning, the speed-accuracy trade-off, the use of ensemble models and the time and resource consumption on training.

7.1.1 Validity

Considering kNN and Stacking have overlapping boxes and that the median line of each is in the other box, they do not have a statistically significant difference. The same can be said for Max-Voting. It is unlikely that there is a real difference.

7.1.2 Variance

In our case, kNN with K=1 provides the best results but a K value of 1 means that the model is very well fitted on training data (possible over fitting). Considering the test data are similar to the training data (extracted from the same data set), K=1 provides good results but they are not general. One of the question is then if the use of an ensemble model can help with the generalization of a kNN model.

7.1.3 Hyper-parameter tuning

An algorithm's learning process is controlled by hyper-parameters, which determine the model parameters it learns. As our work is more of a performance comparison study, we have used tuned and default hyper-parameter values for most of the models. Only in the case of KNN, we had to choose the number of neighbors to inspect. We have measured the accuracy of KNN using five different values as the number of neighbors, k (1,3,5,7,10) and we have kept the max-performing configuration (k=1). Naive Bayes requires almost no hyperparameters, but for other models, singletons or ensembles, it might be possible to improve accuracy by tuning the hyper-parameters.

7.1.4 Speed-Accuracy Trade-Off

There is also a trade-off between speed and accuracy. Some applications require the best possible accuracy at all time and do not need real time processing. In this specific application, a ”small” boost of 4% can be worth it even if it takes twice as much time. This is not a sustainable process but in some case, raising the accuracy from 50% to 55% can drastically change a process (e.g. predicting a head or toe with 55% accuracy makes it easier to win). Moreover, 01m:21s to make predictions on the test data set means that each tweet is analyzed in 21ms which is still a good performance. However, for text sentiment analysis in tweets, the small gain in accuracy does not justify the gap between the performance.

7.1.5 Ensemble methods

In this research, we have used Max-voting and Stacking to build the Ensemble model. Max-voting did not provide better result than all singleton models but Stacking did improve the performance. It is possible that other approaches could provide better performance in our use case. Additionally, Naive Bayes had a lower accuracy score, which might explain why the Max-voting Ensemble model showed lower accuracy scores. We may be able to improve the accuracy of the Max-voting Ensemble model if we remove Naive Bayes. But in that case, if there are only two base models, the votes can be evenly split resulting in an increment of random guessing.

7.1.6 Time and resource consumption on training

A classifier that takes a ”long time” to train may become incredibly painful if we consider cross validation i.e. the use of different parts of data to test and train a model on multiple iterations, model selection, etc., since we will need to train it several times and wait for the results before we can proceed. The computation cost can be another issue to consider. From our experiments, we have seen that the weak learners (kNN, Naive bayes) are fast and therefore requires less resources. But strong learners(XGBOOST) required much more. Max-voting ensemble model did not increase the time and resource consumption much, while in case of Stacking ensemble model, we can see a dramatic increase of time and resource consumption.

7.1.7 Using combination of weak and strong learners as base model

To create an ensemble model, the general approach is to use weak learners since they are trained faster compared to strong learners [26]. In our scenario, we have used a combination of weak and strong learners as base model. kNN and Naive Bayes are weak learners and XGBOOST is a strong learner built on the principles of ensemble modeling. In general, the XGBOOST algorithm creates multiple classifiers that are weak learners, which means a model that gives a bit better accuracy than just a random guess. We have seen this approach is not so efficient in terms of execution time and resources, still, a future scope of this research could be determine if the combination of weak and strong learners provide better accuracy.

7.2 Conclusion

In conclusion, we have seen that not every ensemble model will increase accuracy or performance. Furthermore, different models and different ensemble models can give very different results. In this paper, ensemble models failed to provide significant differences for the accuracy invalidating our hypothesis. Moreover, the average accuracy only increased by 4% invalidating our hypothesis as well. This research highlights that most of the work in the text sentiment field is very hard to generalize because different data set of different size, with a different amount of features will change the way you approach it. Meaning that even if this work is conclusive for text sentiment analysis and ensembling model in general, the whole of it might still be very hard to export into another field, unless the process is tailored specifically for it. Future work in this area could be the comparison of more models (both alone and as ensemble models), in different fields. We had a very specific data set that was a first step for the comparison of many models but only towards text sentiment analysis. A future work could be to see the impact of single models hyper parameters on the final ensemble model, such as the impact of finding the best base hyper parameters for each models before ensembling them. Another approach could be to try striking a balance between models with good performance but high variance and models with lower performance but low variance. Using this approach, would it be possible to achieve good performance on new data?

References

[1] F. Sebastiani, “Machine Learning in Automated Text Categorization,” *ACM Computing Surveys*, vol. 34, pp. 1–47, Apr. 2001. doi: 10.1145/505282.505283

[2] C. Shan, “Research of Support Vector Machine in Text Classification,” *Advances in Intelligent and Soft Computing*, vol. 119, pp. 567–573, Jan. 2011. doi: 10.1007/978-3-642-25538-0_79

[3] D. Li-guo, D. Peng, and L. Ai-ping, “A New Naive Bayes Text Classification Algorithm,” *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 12, Sep. 2014. doi: 10.11591/telkomnika.v12i2.4180

[4] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016. doi: 10.1145/2939672.2939785 pp. 785–794, arXiv:1603.02754 [cs]. [Online]. Available: http://arxiv.org/abs/1603.02754

[5] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, “Using kNN model for automatic text categorization,” *Soft Computing*, vol. 10, Mar. 2006. doi: 10.1007/s00500-005-0503-y

[6] A. Kumar and M. Jain, *Ensemble Learning for AI Developers: Learn Bagging, Stacking, and Boosting Methods with Use Cases*, 01 2020. ISBN 978-1-4842-5939-9

[7] K. Sailunaz and R. Alhajj, “Emotion and Sentiment Analysis from Twitter Text,” *Journal of Computational Science*, vol. 36, Jul. 2019. doi: 10.1016/j.jocs.2019.05.009

[8] “THE 17 GOALS | Sustainable Development,” last Accessed : 2022-10-13. [Online]. Available: https://sdgs.un.org/goals

[9] R. Sathya and A. Abraham, “Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification,” *International Journal of Advanced Research in Artificial Intelligence*, vol. 2, Feb. 2013. doi: 10.14569/IJARAI.2013.020206

[10] T. Donovan and R. Mickey, “Bayes’ Theorem,” May 2019, pp. 29–36. ISBN 978-0-19-884129-6

[11] A. Cucchiara, “Applied Logistic Regression,” *Technometrics*, vol. 34, pp. 358–359, Mar. 2012. doi: 10.1080/00401706.1992.10485291

[12] J. Obregon and J.-Y. Jung, "Explanation of ensemble models," Jun. 2022, pp. 51–72. ISBN 978-0-323-85648-5

[13] A. Singh, "Ensemble Learning | Ensemble Techniques," Jun. 2018. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/

[14] D. Sarkar, *Ensemble Machine Learning Cookbook*, 1st ed. Packt Publishing, 2019. ISBN 978-1-78913-660-9

[15] Y. Sasaki, "The truth of the F-measure," *Teach Tutor Mater*, Jan. 2007. [Online]. Available: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjQvY-Ux936AhXvS_EDHQPSBK4QFnoECBEQAQ&url=https%3A%2F%2Fwww.cs.odu.edu%2F~mukka%2Fcs795sum09dm%2FLecturenotes%2FDay3%2FF-measure-YS-26Oct07.pdf&usg=AOvVaw0nsg1kH_atrdnP_nEpvIJo

[16] S. Jayaprakash and P. Balamurugan, "An efficient text classification using kNN and naive bayesian," *International Journal on Computer Science and Engineering*, vol. 4, pp. 392–396, 03 2022.

[17] J. Brownlee, "A Gentle Introduction to Multiple-Model Machine Learning," https://www.kaggle.com/datasets/datatattle/covid-19-nlp-text-classification?resource=download, 2022, last Accessed : 2022-10-13.

[18] H. Wu and D. Levinson, "Ensemble models of for-hire vehicle trips," *Frontiers in Future Transportation*, vol. 3, p. 876880, 04 2022. doi: 10.3389/ffutr.2022.876880

[19] B. Singh, N. Kushwaha, and O. Vyas, "A scalable hybrid ensemble model for text classification," in *36th IEEE TENCON*, 11 2016. doi: 10.1109/TENCON.2016.7848630

[20] C. Gupta and D. D. Virmani, "Ensem_sldr: Classification of cybercrime using ensemble learning technique," *International Journal of Computer Network and Information Security*, vol. 14, pp. 81–90, 02 2022. doi: 10.5815/ijcnis.2022.01.07

[21] A. Håkansson, "Portal of Research Methods and Methodologies for Research Projects and Degree Projects." CSREA Press U.S.A, 2013, pp. 67–73. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-136960

[22] "Text Classification," last Accessed : 2022-10-13. [Online]. Available: https://kaggle.com/code/matleonard/text-classification

[23] "COVID 19 Tweets EDA & Viz ," last Accessed : 2022-10-13. [Online]. Available: https://kaggle.com/code/datatattle/covid-19-tweets-eda-viz

[24] "scikit-learn: machine learning in Python — scikit-learn 1.1.2 documentation," last Accessed : 2022-10-13. [Online]. Available: https://scikit-learn.org/stable/

[25] M. Palomino and F. Aider, "Evaluating the Effectiveness of Text Pre-Processing in Sentiment Analysis," *Applied Sciences*, vol. 12, p. 8765, Aug. 2022. doi: 10.3390/app12178765

[26] V. Vaghela, A. Ganatra, and A. Thakkar, "Boost a weak learner to a strong learner using ensemble system approach," 03 2009. doi: 10.1109/IADCC.2009.4809227

A Data Analysis

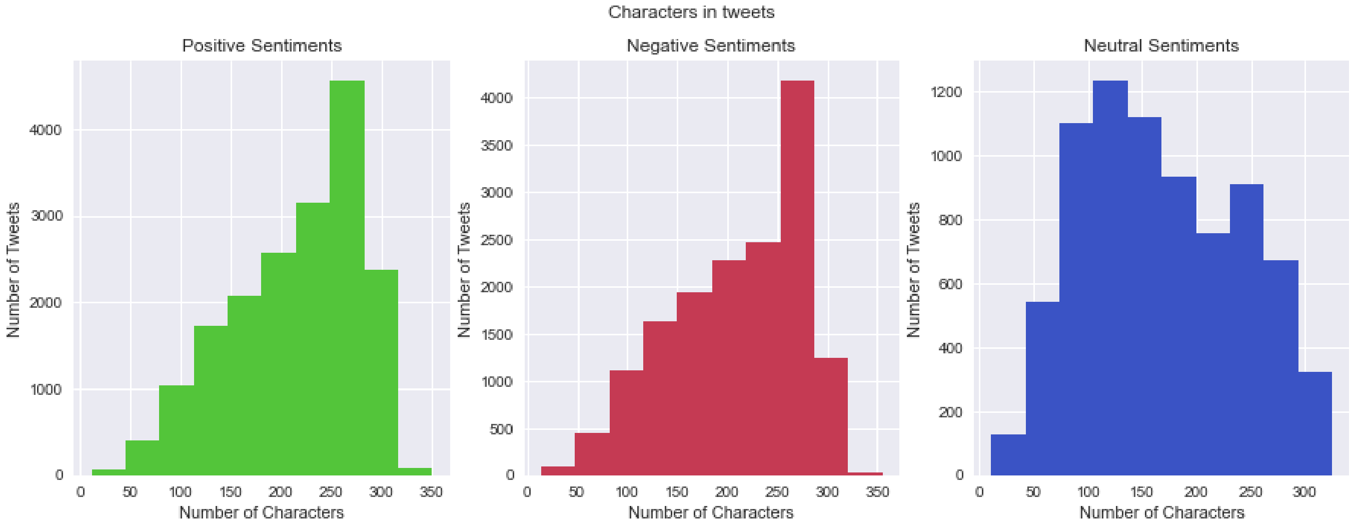


Figure 7: Numbers of characters in a tweet depending on the sentiment

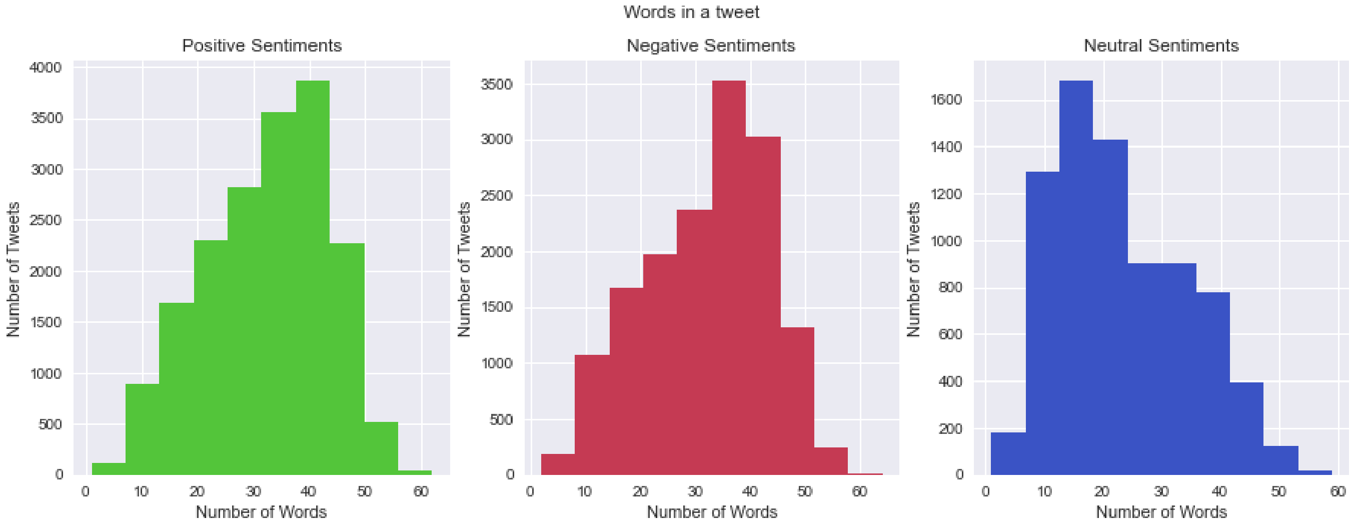


Figure 8: Number of words in a tweet depending on the sentiment

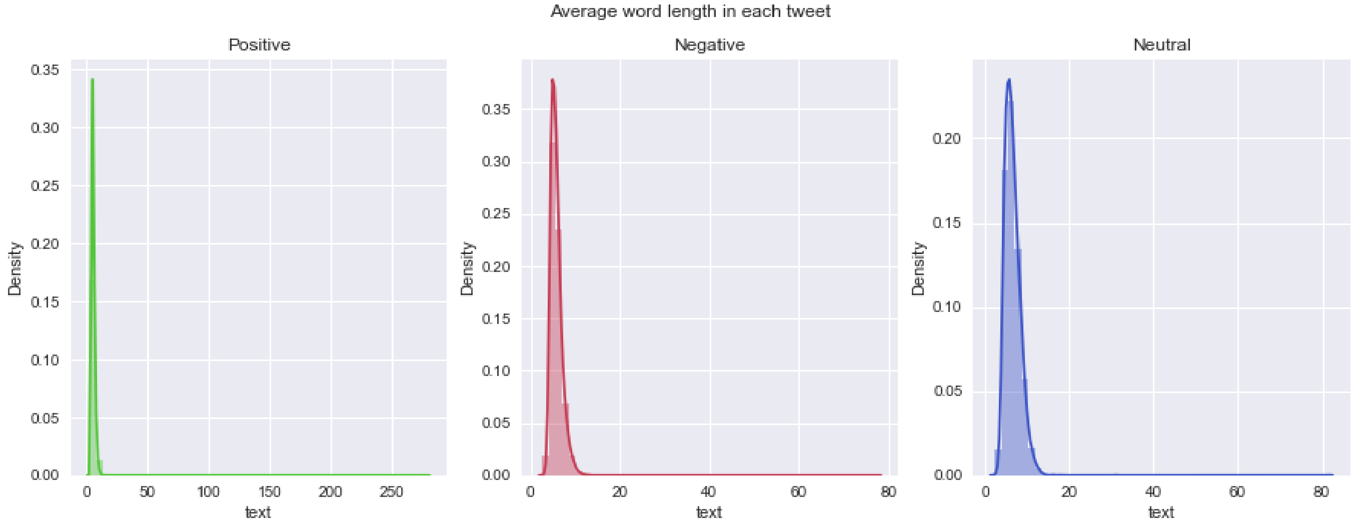


Figure 9: Length of words in a tweet depending the sentiment

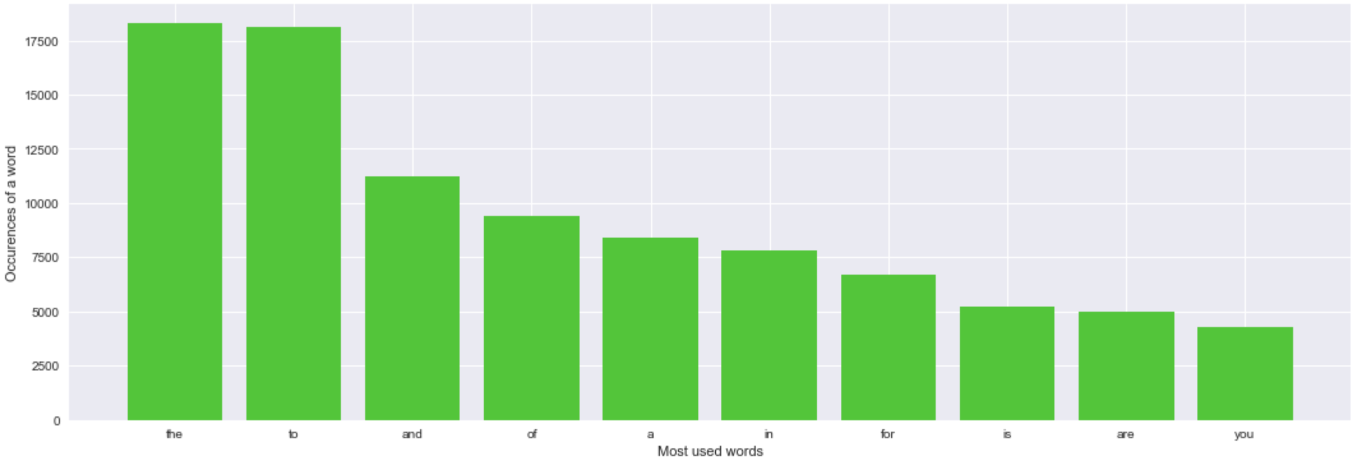


Figure 10: Length of words in a tweet depending the sentiment

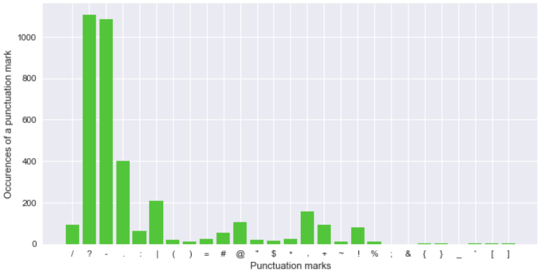


Figure 11: Punctuations occurrence in the positive class

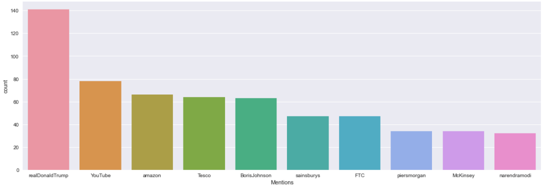


Figure 12: Mentions occurrences in the data set