CLASSIFYING TWEETS SENTIMENT ABOUT THE COVID-19 VACCINE IN ITALY

A MACHINE LEARNING MODEL COMPARISON IN TEXT CLASSIFICATION

CHIARA LOVATI

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

2069579

COMMITTEE

dr. Bruno Nicenboim
dr. Büşra Özgöde Yigin

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

July 9, 2022

WORD COUNT

7609

ACKNOWLEDGMENTS

Thanks to my supervisor, dr. Bruno Nicenboim, for his guidance and insightful suggestions throughout the completion of this thesis project. Thanks to my family and friends for their continuous encouragement and endless support.

CLASSIFYING TWEETS SENTIMENT ABOUT THE COVID-19 VACCINE IN ITALY

A MACHINE LEARNING MODEL COMPARISON IN TEXT CLASSIFICATION

CHIARA LOVATI

Abstract

The following thesis research project aims to explore the performance of different machine learning architectures in the task of sentiment analysis on COVID-19 vaccine-related tweets in Italy. The experimental setting is conceptualised to include: the evaluation of two vectorization strategies for textual features, Bag of Words (BoW) and Term-Frequency Inverse-Document-Frequency (Tf-IDF); a comparison of ensemble and individual classifiers on the task concerned, with the three supervised classifiers being Naïve Bayes (NB), Random Forest (RF), and Extra Trees Classifiers (ETC); and the assessment of the predictive power of two different subsets of variables, one related to the tweet itself and the other to the user posting the tweet. Results indicated little to no difference in accuracy between the two feature extraction strategies, however BoW had the advantage of being less computationally expensive than Tf-IDF. For classifiers, ensemble methods performed significantly better than their individual counterpart, with the RF model, in particular, achieving 0.91 accuracy together with BoW. Analysing the subsets of variables, it could ultimately be deduced that features about the tweet hold the most predictive power, out of the two subsets.

DATA SOURCE, CODE, AND ETHICS STATEMENT

Work on this thesis did not involve collecting data from human participants or animals. The author of this thesis acknowledges that they do not have any legal claim to this data. The code used in this thesis is publicly available: https://github.com/clovatx/Classifying_Tweets_Sentiment.git

1 PROBLEM STATEMENT & RESEARCH GOAL

Sentiment analysis is one of the most popular research topics in the field of natural language processing. Opinion mining, recommender systems, and event detection are just a few of the scientific and commercial applications linked to this field of study (Manguri et al., 2020).

As the Internet has grown in popularity, individuals have continued to resort to various online tools to express their thoughts, views and opinions. User-generated data can be used to monitor public opinion and improve decision-making in a wide range of areas, from the commercial to the public. As a result, sentiment analysis has grown in popularity in recent years among research communities (Wankhade et al., 2022). As social networking sites gained traction, an increasing number of researchers delved into studying these networks and their contents in order to extract relevant information. Predicting the sentiment of social media content gained critical importance.

On a separate front, in December 2019, cases of the novel human coronavirus disease (COVID-19) were first reported in Wuhan, China. The spread of COVID-19 rapidly raised global health concerns, and by March 2020, the World Health Organisation (WHO) had declared the outbreak a global pandemic. One year after the first cases, the first vaccine doses began being administrated.

What ties the relevance of sentiment analysis to the spread of COVID-19 and its vaccine is that, from the very beginning, social media platforms were extensively used to share news and opinions regarding the pandemic (Rustam et al., 2021).

The above considerations bring us to the scope of this thesis project: analysing sentiment towards COVID-19 vaccines across tweets, the building blocks of the social media platform Twitter.

From a societal point of view, understanding the significance of anti-COVID-19 vaccine sentiment is critical considering the recent and continuous vaccination program efforts. Successfully assessing the prevalence of negative versus positive sentiment toward COVID-19 vaccination will benefit governmental bodies in determining relevant policy decisions and communication strategies to address a community that remains skeptical of COVID-19 vaccines.

Furthermore, from a scientific point of view, machine learning methods have been used to categorise tweets and text sentiment for a long time, however, academics have recently been arguing that using ensemble methods is more efficient than using individual classifiers in the task of sentiment polarity detection (Rathi et al., 2018; Rustam et al., 2021; Yousaf et al., 2020). This research project will attempt to explore this assumption further.

This study will make use of three machine learning architectures, one individual classifier: Naïve Bayes (NB), and two ensemble classifiers: Random Forest (RF) and Extra Trees Classifier (ETC). Furthermore, it will evaluate a number of natural language processing techniques, including the Bag-of-Words (BoW) and Term Frequency-Inverse Document Frequency (Tf-IDF) feature extraction approaches. Because the experiment will compare supervised classifiers with balanced classes, accuracy, together with the Receiver Operator Characteristic (ROC) curve and Area Under the Curve (AUC) scores, will be utilised to evaluate their performance.

The data source chosen for this thesis consists of a sample of Italian-language tweets gathered between November 2020 and November 2021. According to studies in the field of sentiment analysis of the COVID-19 vaccine using Twitter as a data source, the vast majority of the experiments were conducted in English, with sentiment analysis about COVID-19 vaccine in other languages remaining relatively uncommon, hence the importance of conducting this study with data in Italian rather than English. (Agustiningsih et al., 2021).

As mentioned, the Twitter data sampled was collected during the course of a year, with a distribution as represented in Fig. 1.

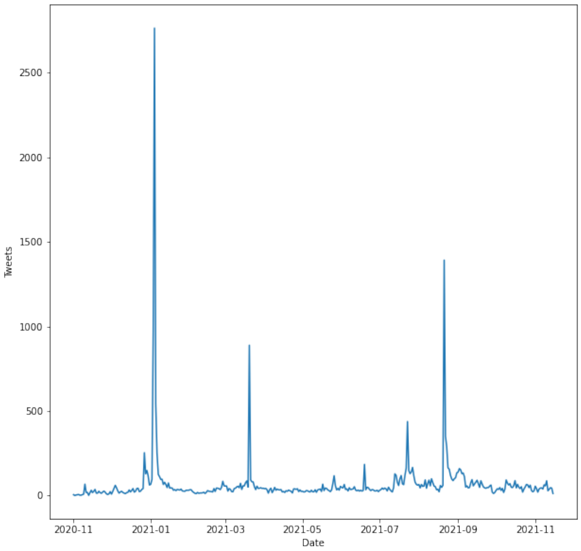

Figure 1: Distribution of tweets per date.

Tweets, also known as "status updates", from the Twitter social network, are commonly rendered in JSON format and are structured as follows:

- tweet - parent - object: "root-level" information, such as id, date of creation, and text of the tweet;
- user - child - object: user account metadata about author of the tweet, including username, number of followers and verified status;
- entity - child - object: further contextual information about the tweet, for example hashtags, user mentions and links. (Twitter.com, 2022)

The entity object will not be included in the scope of this study because the data at hand was collected on the basis of two lists of hashtags that denoted the tweet's position, positive or negative, on COVID-19 vaccination with high likelihood. More details will ensue in Sec. 3. From here on out, the two remaining JSON objects will be referred to as "user" and "tweet" subsets of variables.

In light of all that was mentioned above, the main research question of this project is the following:

> *How accurately can the polarity of the sentiment of Italian tweets on the COVID-19 vaccine be classified using feature extraction techniques combined with machine learning algorithms and trained on variables consisting of information about the tweet and its user?*

The deriving sub-questions can be listed as such:

RQ1 *To what degree does the NLP processing of the text variable with BoW rather than Tf-IDF affect classification accuracy?*

RQ2 *How do the ensemble machine learning models, Random Forest (RF) and Extra Trees Classifier (ETC), compare to the individual Naïve Bayes (NB) classifier in the task of detecting the polarity of the sentiment of Italian tweets on COVID-19 vaccine, and which algorithm performs best?*

RQ3 *Which subset of variables, "tweet" or "user" - when combined with the best performing feature extraction technique and model from the previous RQ - holds the highest predictive power in determining the polarity of the sentiment of Italian tweets on COVID-19?*

These will be evaluated as follows. As the first step, the data will be loaded into a DataFrame and cleaned. The resulting DataFrame will then be duplicated: in one of these, the BoW vectorization technique will be applied to the "text" variable; in the other one, the same will be done via

Tf-IDF. At this point, all three models will be trained and evaluated on both DataFrames. The preceding operation will allow for the comparison of both feature extraction techniques and model performances, thus providing an answer to RQ1 and RQ2. Following that, the DataFrame with the best performing vectorization algorithm will be separated into two partitions based "tweet" and "user" variables subset. The model from the prior phase with the greatest accuracy score will be trained and evaluated on both of these DataFrames individually, and the prediction outcomes will be compared to answer RQ3. The above operations will be detailed in-depth in Section 3.

In the following Section 2, relevant literature and previous works in this area of research will be summarised. Methods and experimental setup will be discussed in Section 3, while results will be presented and commented upon in Sections 4 and 5. Finally, some conclusive considerations will be found in Section 6.

2 LITERATURE REVIEW

In 2019, the World Health Organization named vaccine hesitancy as one of the top ten threats to global health (Akbar, 2019). While this is not a new phenomenon, the spread of anti-vaccination misinformation through social media has given it new urgency, especially in light of the now two-year-old coronavirus pandemic. Social networks, such as Twitter - a free microblogging social media platform with a daily active user base of 229 million people (Iqbal, 2022) and up to 500 million users vising without logging in each month (Rao, 2015) - can indeed serve as an effective tool for quickly informing, motivating, and even politicising citisens during public health crises. For example, in 2020, eight of the nine G7 world leaders had verified active Twitter accounts with a total of 85.7 million followers combined, demonstrating just how impressively influential social media can be (Rufai & Bunce, 2020).

The anti-vaccination sentiment is widespread, and many people get their vaccine information on social media platforms(Wilson & Wiysonge, 2020). A review on social media interventions aimed at influencing vaccine decision-making conducted by Rupali et al. (2021) showed evidence of social media content impacting vaccine knowledge, attitudes, intentions, and behaviours. In a 2021 study, exposure to misinformation was directly linked to vaccine hesitancy. Misinformation exposure with political affiliation showed to be strong predictors of vaccination even after accounting for other demographic predictors (Neely et al., 2022). It is critical to learn community's perceptions of vaccination policy implementation ahead of time and sentiment analysis is a valuable technique for this purpose.

Sentiment analysis can be achieved in a variety of ways: while the goal of this area of expertise can be summarised as that of categorising a document into one of several sentiment categories – or, more generally, assigning a predefined label to a given input text - most frameworks include not only computation techniques and algorithms, to extract and classify sentiments, but also a multitude of other procedures. These comprise Natural Language Processing (NLP), text analysis and dimensionality reduction, all of which get carried out using an assortment of combinations (Kowsari et al., 2019; Gasparetto et al., 2022; Agustiningsih et al., 2021).

Because the data from social media, such as Twitter, includes a multiplicity of writing and language styles, text preprocessing is a key step in conducting sentiment analysis. It also allows for efficient and smooth feature extraction. Text preprocessing normally contains tasks such as lowercase conversion, punctuation removal, tokenisation, stop words removal, and either stemming or lemmatisation. Depending on the domain and language, tokenisation types and stemming or lemmatisation techniques may differ. Extensive experimental analysis by Uysal & Gunal (2014) revealed that the right combination of preprocessing tasks can significantly improve classification accuracy, whereas the wrong one can potentially degrade it, making this a crucial procedure.

Many studies have used data from various social media platforms to analyse sentiment on the COVID-19 vaccine issue, with Twitter being one of the most commonly used social media data sources in sentiment analysis research (Agustiningsih et al., 2021). Agustiningsih et al. analysed twenty-one publications on sentiment analysis of tweets about the COVID-19 vaccine for different perspectives on data collecting, data processing, classification, and sentiment analysis outcomes. They observed that unnecessary characters, words, phrases, and even whole tweets are frequently removed from tweets during the text preprocessing stage. Several studies began by removing duplicate tweets, because so-called "retweets": posts written by one user and republished by another, are a very common occurrence on the Twitter platform. Furthermore, the removal of URLs, links, hashtags, mentions and punctuation was performed in the majority of the studies, along with the removal of stop words. Some further frequent procedures included lowercase conversion and lemmatisation or stemming of the words. Lemmatisation distinguishes a word's inflected forms and returns its base form; as such, a word like "better" would be lemmatised as "good", stemming, on the other hand, determines a word's common root form by deleting or substituting word suffixes; for example, "densely" would be stemmed as "dens" (Huang et al., 2019). According to Agustiningsih et al., the rationale for scholars to prefer one technique over the other may be that lemmatisation is more useful in languages with a large number of affixes, but another difference between the techniques may be influencing the choice: while lemmatisation has the advantage of producing true dictionary words, it is both more complicated and substantially longer to implement than stemming (Elia, 2020).

Overall, as mentioned, preprocessing techniques are crucial beacuse they enable a smooth implementation of the necessary feature extraction or vectorization techniques on text data, which would otherwise be very challenging to process. In fact, unstructured data cannot be handled directly by the classification models due to of the nature of algorithms, which often only accept input in the form of numbers. To extract features from text data some of the most frequently used methods include BoW and Tf-IDF, two techniques that can be used to convert text phrases into numerical vectors (Agustiningsih et al., 2021). Although a general consensus exists on the fact that Tf-IDF has the advantage of not ignoring semantic relationships between words, which should improve overall text classification performance, this does not appear to be definitively proven. For example, Pimpalkar and Raj (2020) found a greater improvement in F1-score with BoW rather than with Tf-IDF. Furthermore, because Tf-IDF computes document similarity directly in the word-count space, it has the disadvantage of becoming slow for large vocabularies. This will be an area of investigation for this thesis project.

BoW and Tf-IDF may have the disadvantage of working with collections of unigrams and storing text in tokens of size one word, which prevents them from capturing phrases and multi-word expressions and ignores word order dependencies. However, employing n-grams techniques, while addressing this problem, can quickly increase the dimensionality of the data, leading to less optimal models (scikit learn, 2022; Nair, 2021).

Following feature extraction, computational models enter the picture. In order to solve text classification problems, a majority of traditional machine learning methods have been adapted. The review of classification techniques used in classifying tweets regarding the COVID 19 vaccine by Agustiningsih et al. (2021) found a combination of machine and deep learning methods, and some examples were Linear Regression (LR), Naïve Bayes (NB), Random Forest (RF), Support Vector Machine (SVM), and Gradient Boosting (GB).

This experiment will use machine learning classifiers for the task at hand. More in depth, these can be seen as supervised learning methods, or machine learning techniques that use training data (i.e., pairs of input data points and the desired output) to learn a classifier or regression function that can be used to make predictions on new data that hasn't been seen before. (Aggarwal & Zhai, 2012).

According to research, a rather effective classifier utilised in sentiment analysis is NB (Agustiningsih et al., 2021). For example, in 2020, Samuel et al. compared the effectiveness of NB and LR, in classifying COVID-19 public sentiment and found that NB provided the best accuracy, with an accuracy score 0.91 against 0.74.

Among these supervised methods, a number of them have been combined and developed to maximise generalisability and resilience over typical individual estimators: ensemble machine learning models. An ensemble is a group of independent algorithms which are collectively trained on data (Rahman & Tasnim, 2014). They are designed to integrate predictions from various base classifiers (scikit learn, 2022) and have been found to perform incredibly well in a variety of studies and classification tasks (Rustam et al., 2021; Yousaf et al., 2020; Amasyali & Ersoy, 2011; Rathi et al., 2018).

Although not entirely related to Covid-19 sentiment analysis, comparisons between the two classes of algorithms were made. In a 2011 analysis of the classification accuracy and processing speed of 12 individual classifiers and 11 ensemble algorithms - including NB, SVM, DT, and RF - Amasyali & Ersoy evaluated the models on a variety of datasets and discovered that the top 6 algorithms were ensemble classifiers, with RF emerging as one of the top options when accuracy and execution time were considered together. In 2018, Rathi et al. experimented with SVM and DT in the classification of sentiment in text, including tweets, movie reviews, and sentences from different social media, intending to improve the efficiency and reliability of their approach; their results showed that a hybrid combination of the chosen models together achieved better classification results in terms of accuracy and f-1 score than when the two methods were used on an individual basis. Yousaf et al. (2020) compared seven machine learning algorithms, including four individual classifiers, LR, NB, DT, and SVM, and three ensemble classifiers, RF, Gradient Boosting model (GBM), and Voting Classifier (LR-SDG, or LR plus a Stochastic Gradient Descent classifier), in the classification of a dataset of diverse topics tweets as happy or unhappy. Their proposed voting classifier (LR-SGD) with Td-IDF produced the best result with 0.79 accuracy. In 2021, Rustam et al. compared a variety of machine learning models - Extra Trees Classifier (ETC), XGBoost classifier (XGB), Support Vector Classifier (SVC), Long-Short Term Memory (LSTM), DT and RF – in Covid-19 tweet sentiment analysis, and observed ETC outperforming all other models with a 0.93 accuracy score, with LSTM performing worst out of the remaining models. The outstanding performances of NB, RF and ETC motivated their inclusion in this research project.

3 METHODOLOGY & EXPERIMENTAL SETUP

3.1 *Data source*

The datasets used in this study were taken from the Harvard Dataverse's VaccineEU (Pierri, 2022) repository, created in 2022. Two lists of tweet IDs, one supporting the COVID-19 vaccine and the other opposing it, were specifically downloaded and used in this project. The author of the collection initially retrieved the tweets associated with the IDs from Twitter using two distinct sets of manually selected hashtags - surrounding COVID-19 vaccine - that indicated the stance ("pro" or "anti") of the tweets with a high degree of probability: these are known as Gold Hashtags (GH). It is assumed that tweets containing one or more GH from the same position express the same particular viewpoint on vaccines.

As per Twitter's privacy policy, tweet corpuses are only publicly stored as ID values, so in order to access the full data of the "status updates", a procedure known as "hydrating" must be implemented. This is the automated process of retrieving a complete Twitter Object from the site using its ID. The Hydrator software (the Now, 2020), which is available on GitHub, was used for this process. As a result, 49,275 negative sentiment tweet objects and 47,158 positive sentiment tweet objects were generated. The aforementioned datasets were given a sentiment variable with the value 1 for negative sentiment and 0 for positive sentiment. The features were then combined to yield a total of 96,167 instances and 36 variables. The appendix section contains a detailed list of variables, their specific subsets, and data types.

3.2 *Variables subsets*

Based on the position of the features in their JSON file of origin, tweet variables can be differentiated between those that relate to the main text object representing the Twitter "status update" and those that relate to the user behind the post. In this regard, all user-related variables have the prefix "user_" in their name, and as such, it is possible to easily identify them within the experiment and subset them to investigate their contribution to the sentiment analysis task.

3.3 *Data Preprocessing*

To begin Exploratory Data Analysis (EDA) and preprocessing, the data was inspected for missing values and incomplete variables. Table 1 illustrates the percentages of missing values per column that were found.

Table 1: Percentages of missing values per column.

Column	Missing values (%)
coordinates	99.92
created_at	0.00
favorite_count	0.00
id	0.00
in_reply_to_screen_name	93.04
in_reply_to_status_id	93.48
in_reply_to_user_id	93.04
lang	0.00
place	99.10
possibly_sensitive	80.04
quote_id	89.16
retweet_count	0.00
retweet_id	30.40
retweet_screen_name	30.40
source	0.00
text	0.00
tweet_url	0.00
user_created_at	0.00
user_id	0.00
user_default_profile_image	0.00
user_description	19.60
user_favourites_count	0.00
user_followers_count	0.00
user_friends_count	0.00
user_listed_count	0.00
user_location	47.13
user_name	0.04
user_screen_name	0.00
user_statuses_count	0.00
user_time_zone	100.00
user_urls	86.65
user_verified	0.00
sentiment	0.00

Following, as retweets are a very common occurrence on the Twitter social network, the "text" variable was inspected for duplicates. Indeed, only 31.56% of the samples reported an original text example, so the rest of the data was removed, such as in other studies reviewed by Agustiningsih et al. (2021). Furthermore, variables with over 50% of missing values were dropped from the DataFrame.

Out of the remaining variables, a further 11 features were dropped:

- columns such as "user_screen_name", "source", "tweet_url", "user_id", "user_name", "user_created_at", and "id", which acted as identifiers of each specific user or tweet;
- the "lang" column, containing the exact same value per every instance ("it");
- "user_location", presenting both a high percentage of missing values, but also several uninformative string instances such as "somewhere in EU" or the Italian translation of "mostly sitting";
- the "hashtags" variable, which contains only examples of hashtags that are strongly connected to the target variable.
- "user_description", a secondary text variable, in order to lower the computational cost of the experiment.

Some further cleaning operations included the transformation of the "created_at" variable from DateTime object to integer data type, the shuffling and resampling of the dataset to obtain balanced classes - in the measure of 12500 instances per each sentiment group - and the definition of the text cleaning and preprocessing functions based on the best practices detailed in Sec. 2.

Because the goal of text preprocessing or cleaning is to reduce the amount of data in a document to only the most necessary information while preserving its context and meaning, the "text" column was then preprocessed following these six main steps, in relation to their popularity in the Agustiningsih et al. (2021) review:

i letters lowercasing;

ii removal of hashtags, mentions, and links;

iii removal of punctuations and non-alphanumeric characters;

iv tokenisation;

v removal of stopwords;

vi lemmatisation.

The effect of letter lower-casing is that words in uppercase and lowercase are treated the same: even if a term may have been written differently by several users, the word would ultimately have the same meaning. Furthermore, elements like hashtags, mentions, and links which are common in tweets but are rarely needed for text processing, along with punctuation and non-alphanumeric characters, are removed. In the tokenisation step, the text is split into words - each of which distinct and unrelated to the others. At this point, stop-words can be removed; these are words that have little meaning due to their widespread use in language; examples include conjunctions such as "and, but, for, nor, or, so" and so on. Stop-word lists for 23 languages, including Italian, can be retrieved from the NLTK library (Bird et al., 2009). Finally, lemmatisation, or the process of removing inflectional endings from a word and returning it to its base or dictionary form, known as the lemma, was carried out. Due to the limited number of Python libraries available for this particular task, lemmatisation for languages other than British or American English can prove to be challenging. However, thanks to its compatibility with the Italian language, the SpaCy library (Honnibal & Montani, 2017) could be used for lemmatisation in this project (*spacy.io*, 2022; Vasiliev, 2020).

3.4 *Text Feature Extraction*

Once bodies of text are converted into lists of simplified and standardised tokens by preprocessing pipelines such as the one above - because textual data, unlike other types of data such as images or time series, lacks an inherent numerical representation - these must be projected into an appropriate feature space before being used as input to a classifier. In other words, they must be represented in a machine-digestible format, i.e. a vectorial form. At this point feature extraction techniques such as Bag-of-Words (BoW) and Term frequency-inverse document frequency (Tf-IDF) are introduced into the experimental setup. The BoW method is known for streamlining text bodies by treating their tokens as an unordered set of words, but it disregards sentence structure and semantic relationships between elements. Tf-IDF, on the other hand, counts the number of times a word appears in a body of text, or text frequency, and weights it with its inverse document frequency, decreasing the influence of common or less valuable terms by penalising their overall score (and raising the ones of rarer words) (Gasparetto et al., 2022; Vidhya, 2021; Roldós, 2021).

Feature extraction, or vectorized representation of textual features, commonly raises time complexity and memory consumption issues. As a result, dimension reduction techniques are required. Sub-setting the original features or transforming them into new ones are two options for reducing the size of the feature space. Non-negative matrix factorisation (NMF), Linear Discriminant Analysis (LDA), and Principal Component Analysis (PCA) are some of the most popular techniques for achieving dimensionality reduction (Kowsari et al., 2019).

To carry out feature extraction, the DataFrame was first duplicated, and for each of the duplicates, a different text vectorization method for the

$''text''$ variable was applied. The functions utilised were "scikit-learn" count vectorizer for BoW, and Tf-IDF vectorizer for the homonym representation. These were used with the $''max\_features''$ parameter set to 1000 samples, in ordee to reduce the complexity of the resultant matrices, which now only contained the top 1000 features based on term frequency across the corpus. The count vectorizer was also configured with a positive binary encoding parameter, which sets all non-zero counts to 1. This process generated sparse matrices of BoW and Tf-IDF representation as outputs for vectorization of the $''text''$ variable. However, to serve as input for the classifiers, these had to be converted to dense matrices and then to DataFrames, with the words as column names, and then concatenated back to the original DataFrames with the remaining features.

At this point, the proposed models could be used to be learn and predict the sentiment of the tweets.

Fig. 2 depicts the workflow of the process.

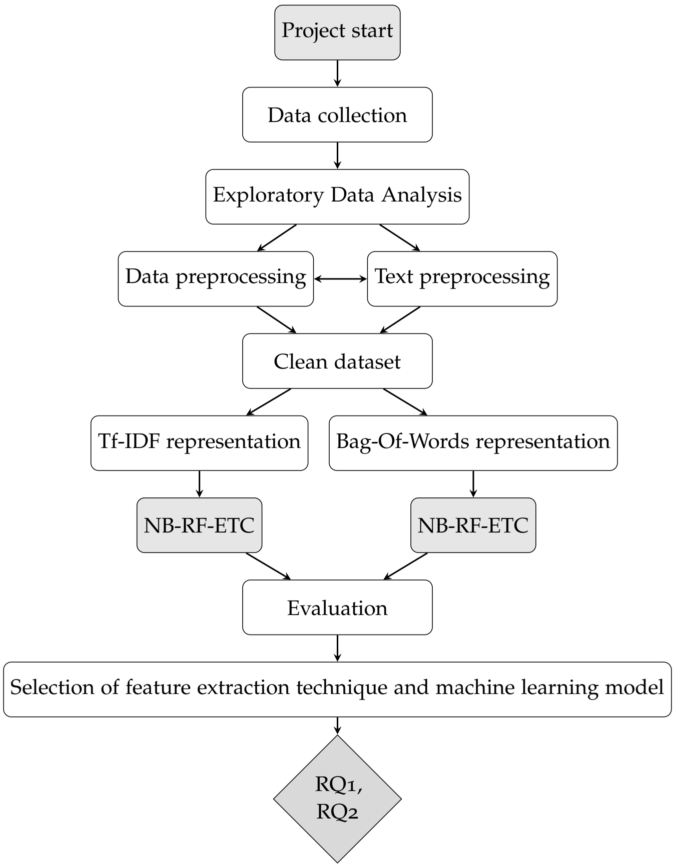

Figure 2: Method flowchart, first phase

3.5 *GridSearchCV*

To train and validate each one of models, parameters tuning via grid search and k-fold-cross-validation with k=10 was carried out. Grid search is a methodology implemented in order to find the optimal hyperparameters of a model, that works by trying multiple combinations of the provided parameter values and testing the classifier or estimator for each combination via cross-validation, comparing accuracy and loss for each hyper-parameter combination (scikit learn, 2022). Cross-validation is a re-sampling technique used to evaluate machine learning models on data samples. The process requires only one parameter, k, which specifies how many times a given sample should be divided into and folded. As a result, the method is also known as k-fold cross-validation, with the chosen value for k typically replacing k in the model's reference, so 10-fold cross-validation denotes a configuration in which k equals 10. Indeed, the choice of k is associated with a bias-variance trade-off; however, values of k=5 and k=10 have been empirically shown to produce test error rate estimates that are not excessively biased or with a very high variance and are widely used in the field of machine learning (James et al., 2013; Brownlee, 2020) .

The grid search function used in this project takes as inputs:

- the estimator (or classifier in the scope of this research);
- a dictionary of parameters, or parameter space;
- a scoring function, accuracy in this case;
- a cross-validation value, corresponding to 10;
- the amount of jobs to run in parallel, set to maximum.

The function mentioned above is fitted to the training data - from which it will independently extract validation partitions through cross-validation - searching for the optimal parameter combination. The model that results from this configuration is then used to predict the test data, and the prediction is evaluated against the true target test values. This process is repeated for every classifier in the experiment.

The three classifiers implemented for this experimental setup are Naïve Bayes (NB) (as the individual classifier), Random Forest (RF) and Extra Trees Classifier (ETC) (as the ensemble classifiers). Further details on the models' implementation and their hyperparameters tuning will follow.

3.6 *Models*

Before beginning sentiment analysis with the selected classifiers, a dummy classifier was trained and evaluated on the already preprocessed DataFrame - before the implementation of feature extraction techniques - in order to provide a very simple baseline for the model comparison that followed. It achieved an accuracy score of 0.49, uniformly generating predictions at random from the two sentiment classes.

3.6.1 *NB*

NB is a classifier based on Bayes' theorem and is characterised by an assumption of independence among predictors. It can predict the probability of an observation belonging to a class, assuming that the effects of different attributes are independent of one another for every class. This is also known as class conditional independence. It is deemed "naïve" because it is intended to make the computation simpler. (Sunil, 2021; Leung, 2007).

Bayes theorem provides a way of calculating posterior probability of a class c given a predictor x - $P(c|x)$ - from the prior probability of the class $P(c)$ and of the prediction $P(x)$ and the likelihood of the predictor given the class $P(x|c)$. It is also written as such:

$$P(c|x) = \frac{P(x|c) * P(c)}{P(x)}$$

There are three types of NB classifiers and they all have been built in such a way that they perform particularly well on a wide range of text-related tasks; however, each works best with different types of features and data:

- Gaussian NB, the preferred algorithm for normally distributed continuos features.
- Multinomial NB, working with discrete features that follow a multinomial distribution, and data as integer counts generated by term frequency.
- Bernoulli NB, which also works with discrete features, but more effective in dealing with binary data.

This project will utilise Multinomial NB (MNB), for the above reasons.

The MNB has one hyperparameter known as $alpha$, controlling additive smoothing for the algorithm and taking a float value as the input. The values evaluated with GridSearchCV were: $1, 0.1, 0.01, 0.001, 0.0001, 1e-05)$. Across analyses, the preferred parameter turned out to be an $alpha$

value of 1, corresponding to the default value. In predicting the data when combined with BoW and Tf-IDF, NB performed just slightly better than the dummy classifier . However, when classifying the DataFrames produced by the various subsets of features, it significantly improved performance when predicting the DataFrame of "tweet" features (0.54 to 0.70 accuracy), indicating that the "user" variables might be producing too much noise for this particular classifier.

3.6.2 *RF*

The RF algorithm is made up of a collection of decision trees with depths of one or more. The term random refers to the training set's random sampling, whereas forest refers to the model being a collection of trees (DT). In adding a node to a tree, each time, a random subset of features is chosen. The model searches for the optimal cutting point to determine the split for each of these by measuring their impurity with the Gini score. The feature from the randomly chosen subset that produces the purest split is then used to build the node in question, increasing the three to a depth of one. The process is repeated for the remaining nodes in the tree until the desired depth is reached. Its advantages stem from the fact that each tree is constructed independently using a different bootstrap, resulting in variation among the trees, and as the number of DT in the forest increases, the generalisation error reaches a limit. Furthermore, using a random selection of features to split each node results in noise-resistant error rates (Breiman, 2001; Zhu et al., 2021; Ceballos, 2020a, 2020b).

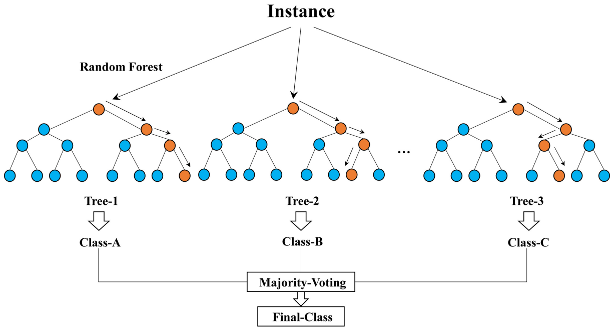

Figure 3: Diagram of the RF algorithm and prediction process (Zhu et al., 2021)

3.6.3 *ETC*

Extremely Randomized Trees Classifier, also known as Extra Trees Classifier (ETC), is a kind of ensemble learning method that combines the classification outcomes of several de-correlated DT gathered in a "forest" to obtain its classification result. The ETC Forest's DT are built from the original training sample. Despite being very similar to RF, ETC classifier introduces randomisation to the structure, where at each test node, each tree is given a random sample of k features from the feature-set out of which each decision tree must choose the best feature to split the data; this random sample of features naturally results in the formation of multiple de-correlated decision trees (Gupta, 2020). This usually allows to reduce the variance of the model a bit more in comparison to RF, at the expense of a slightly greater increase in bias (scikit learn, 2022).

With regards to parameter tuning, when using either the RF or ETC methods, the key parameters that can be adjusted are $n\_estimators$ and $max\_features$. The first is the number of trees in the forest; the greater the number, the longer the computing time. The second factor is the size of the random subsets of features to consider when splitting a node; decreasing this number reduces variance but it can increase bias. Some empirical good default values for regression are $max\_features$ equal to 1.0 or None, while for classification tasks, they are $max\_features$ equal to "sqrt", which means using a random subset of the size of the square root of the number of features in the data(scikit learn, 2022).

For GridSearchCV, a set of four parameters were tuned for both RF and ETC:

- firstly, $n\_estimators$ (default = 100) with values 10, 50,and 100;
- secondly, $max\_features$ (default = $"sqrt"$), with alternative $"sqrt"$ and $"log2"$;
- thirdly, max_depth (default = $None$) set to either $None$, 10, or 50;
- finally, $bootstrap$ (default RF = $True$, default ETC = $False$) with boolean $True$ and $False$ values.

The best grid search parameters were frequently the same as the default ones. This was the case for $n\_estimators$ equal to 100 and max_depth equal to $None$. The optimisation also consistently confirmed the $max\_features$ value as equal to $"sqrt"$ for both RF and ETC, in line with the empirical best practices (scikit learn, 2022), except in the case of the analysis of the "user" subset, where grid search found $"log2"$ as the best value for RF - possibly because the DataFrame in question contained only 7 predictor variables. While the default value for $bootstrap$ corresponds to $True$ for RF and $False$ for ETC, the optimal parameter was found to be $False$ for both models in predicting the Tf-IDF DataFrame, $False$ for RF and $True$ for ETC in predicting the BoW Dataframe, and $True$ for RF in both occasions of predicting the two variable subsets DataFrames.

3.7 *Evaluation Criteria*

The most commonly used metric in the scope of evaluating a supervised binary classifier with balanced classes is accuracy; hence it will be used to evaluate the performance of the classifiers employed in this investigation, along with ROC curve and AUC scores for further insight. Accuracy is part of the most well-known evaluation metrics for machine learning classifiers, together with precision and recall. These may be easily calculated from a confusion matrix, which is a type of contingency table arrangement that allows us to compare classifier predictions of observations to their true class. Accuracy is defined as the proportion of correctly identified data observations among all observations, while precision is defined as the fraction of correctly predicted positive observations among all expected positive observations, and recall is defined as the ratio of correctly predicted positive observations to all the observations in the true class.

Below, Fig. 4 illustrates the second phase of the experiment.

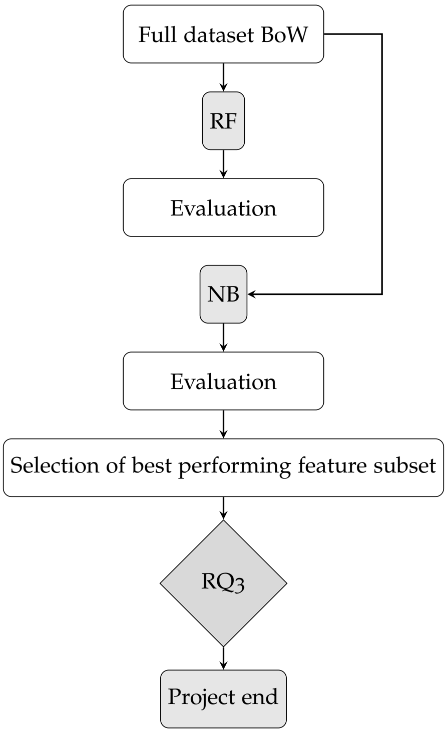

Figure 4: Method flowchart, second phase

4 RESULTS

The first set of analyses implemented allowed to search for answers to RQ1 and RQ2. In particular, it was possible to observe the influence of the two feature extraction techniques, Bow and Tf-IDF, on classification accuracy of the models. In regards to RQ1, with the exception of ETC, BoW and Tf-IDF achieved identical scores for all of the models utilised; thus, the results were largely overlapping and not clearly decisive, as depicted by Tab 2.

Table 2: Accuracy results per model, by vectorization technique

	BoW	Tf-IDF
NB	0.54	0.54
RF	**0.91**	**0.91**
ETC	0.85	0.87

One hypothesis as to why this happened is that the classification task has a binary output, rather than a multi-class target such as in a topic detection task, allowing for better accuracy on both vectorizers. Indeed, both methods represent state-of-the-art techniques to represent textual features. Complete classification reports and confusion matrices for each of the models run in combination with feature extraction algorithms can be found in Appendices A and B. As to gain further insight into these results, the models' training and execution times were recorded and then visualised. In this case, it was possible to observe the slight advantage of BoW from a time complexity point of view, achieving faster training times in three out of three model cases, as seen in Fig 5.

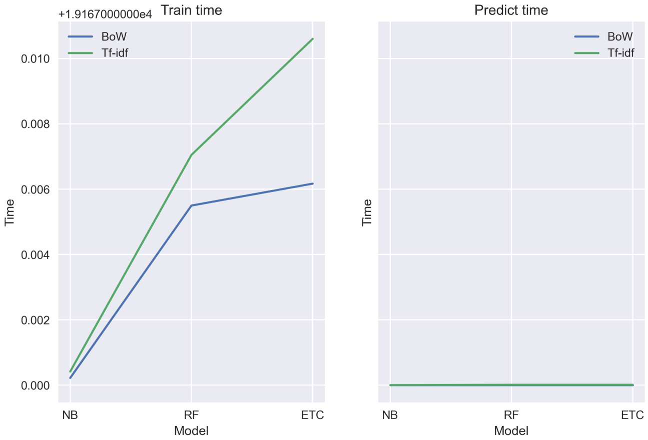

Figure 5: Execution time comparison per model, by vectorization technique

From a model perspective, in order to answer RQ2, some clear distinctions emerged starting from the accuracy scores (Tab 2). NB achieved 0.54 accuracy in predicting sentiment in each DataFrame regardless of the different feature extraction techniques; similarly, RF achieved 0.91 accuracy in both instances for the task at hand; ETC, on the other side, was the only model that showed differences, however subtle, in the classification of the two DataFrames: together with BoW features, ETC achieved 0.85 prediction accuracy, and with Tf-IDF features, accuracy improved to 0.87.

To further explore the performance of the model, their AUC (Area Under The Curve) scores and ROC (Receiver Operating Characteristics) curve were calculated and plotted, as seen in Tab. 3, Fig. 6 and 7.

Table 3: AUC scores per model, by vectorization technique

	BoW	Tf-IDF
NB	0.54	0.54
RF	**0.91**	**0.91**
ETC	0.85	0.87

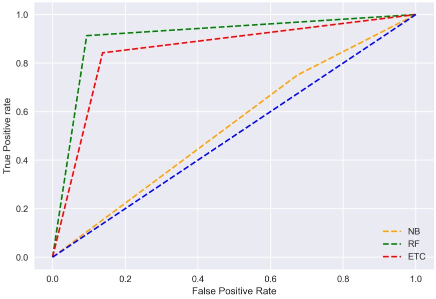

Figure 6: ROC curves per model, BoW

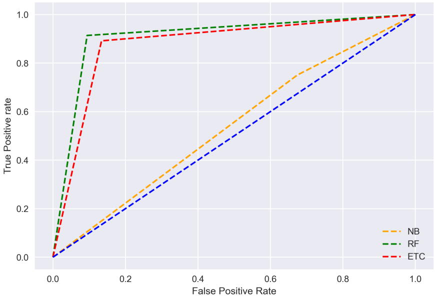

Figure 7: ROC curves per model, Tf-IDF

Consistently with the literature reviewed (Amasyali & Ersoy, 2011; Yousaf et al., 2020; Rustam et al., 2021; Rathi et al., 2018), ensemble classifiers distinctively outperformed the individual classifier proposed with a margin totaling to 0.31 between NB and ETC, and reaching 0.37 between NB and RF. The ensemble classifiers also showed better discrimination capacity to distinguish between positive class and negative class, while the individual classifier showed a substantial bias towards the negative sentiment category (Tab. 9 and 15). It is finally important to note, that one model indeed did perform better than all others in this first analyses: the RF classifier, achieving impressive accuracy in predicting sentiment on both BoW and Tf-IDF datasets.

To definitively answer RQ1 and RQ2 in order to move on to the investigation of RQ3, BoW proved to be the winning feature extraction strategy for the "text" variable, owing to its faster computational speed; whereas, in terms of models, ensemble classifiers, and particularly RF, proved to be the most successful. The second part of the analysis thus began by using BoW for feature extraction and RF for sentiment prediction, but the original BoW DataFrame was divided into "tweet" and "user" variables subsets, according to the features category. While one could suppose that the subset containing "user" metadata would not have the same predictive power as the one containing all information about the tweet, its date, and its text, the results indicated otherwise, as seen in Tab. 4 and Fig 8.

Table 4: Accuracy scores per RF, by variable subset

	Tweet	User
RF	0.86	**0.87**

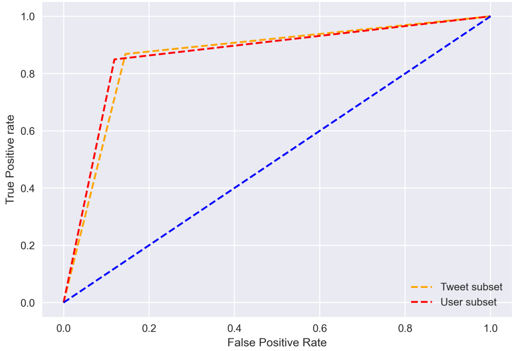

Figure 8: ROC curves per RF, by variable subset

The outcomes for both feature subsets were essentially the same, with the accuracy for predicting the "user" subset even being, although minutely, greater than the "tweet" one. Indeed, RF is a potent algorithm that relies on the fusion of various prediction outputs to reduce overfitting issues and variance, achieving high accuracy even in challenging circumstances, and this might be a reason for the above results. However, in order to further explore RQ3, the above analysis was repeated using NB, a simpler and (more naive) model, to see if the same feature behaviour would hold up.

Table 5: Accuracy scores per NB, by variable subset

	Tweet	User
NB	**0.70**	0.57

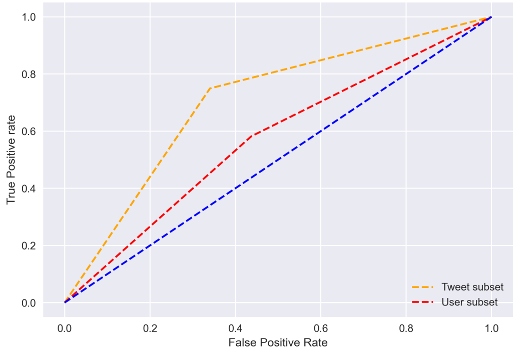

Figure 9: ROC curves per NB, by variable subset

In contrast to RF, NB showed clear differences in the performance of the subsets. This time, the "user" subset achieved a very lower accuracy than the "tweet" one, with a margin of 0.13. One last metric that was investigated to explain the difference in predictive power between the two variables subsets were correlation coefficient between predictor and target features. After calculating these scores for each of the subset DataFrames, the top correlation coefficients with the variable "sentiment" were plotted with a Seaborn heatmap. The following figures were the result, Fig. 10 and 11.

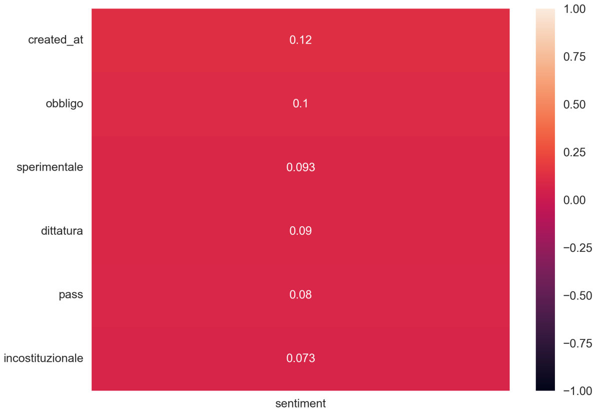

Figure 10: Correlation of tweet features with "sentiment"

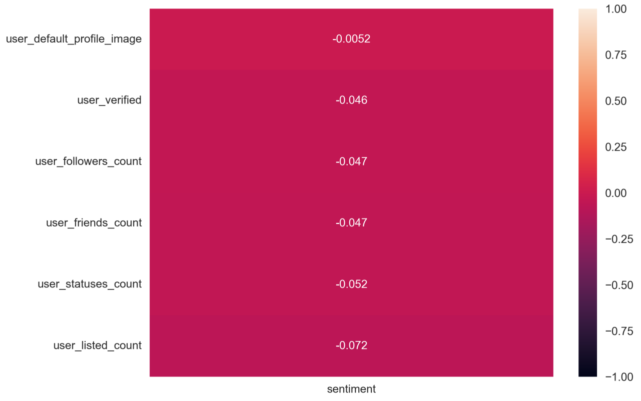

Figure 11: Correlation of user features with "sentiment"

Although statistical significance could not be determined, it was possible to observe that the highest Pearson's correlation coefficient regarded the "created at" variable, from the "tweet" subset, correlating the date of creation of the tweet with real world events and changes. Other close correlation coefficients included words (tokens encoded by BoW, from the "tweet" subset) such as "obbligo", "sperimentale", and "dittatura", which directly translate to "obligation", "experimental", and "dictatorship" - respectively. These visualizations supported the theory that variables in the "tweet" subset had a higher predictive power than those in the "user" subset.

5 DISCUSSION

The focus of this thesis research project was a sentiment analysis and text classification task. In light of the research goal, data preprocessing procedures, feature extraction techniques, and machine learning models were implemented and assessed to determine the combination yielding the highest accuracy.

RQ1 was designed to examine the influence of two distinct feature extraction strategies on the classification accuracy in predicting the sentiment of a tweet about the COVID-19 vaccination. As per the findings, these were, at first glance, overlapping and not decisive in terms of accuracy. One explanation for this could be that the classification task featured a binary sentiment output, rather than, for example, a topic detection task, which reduced the room for uncertainty and allowed for improved model accuracy on both vectorizers. Indeed, as stated in Sec 4, both methods continue to be amongst the most popular techniques for representing text features (Agustiningsih et al., 2021). Regardless, despite little to no variations in model accuracy between the two feature extraction methods, a difference in computing speed was observed, with BoW features allowing for faster model training and validation in all model implementations.

Through RQ2, ensemble classifiers (RF, ETC) were evaluated against an individual classifier (NB) in recognising the polarity of the sentiment of tweets about the COVID-19 vaccine. Results showed apparent differences, proving gains in terms of accuracy and sensitivity when using ensemble classifiers. Additionally, in the scope of this research, the RF algorithm consistently achieved better results than the other classifiers, with accuracy scores as high as 0.91.

The evaluation of RQ3 posed a bit of an enigma; in fact, once the analysis of the two DataFrames resulting from the split of the primary data according to the two variables subsets, "tweet" and "user", was conducted - together with BoW and RF as the classifier, because of their distinct performances in the previous phase - results obtained were inconclusive. With RF, both subsets reached an identical accuracy score. A different approach was thus taken, and the analysis was repeated using NB, a simpler algorithm, to see if the similarity persisted; indeed, this did not happen. NB showed a clear difference in prediction accuracy between the two subsets, with the "tweet" one reaching 0.70 accuracy, and the "user" one reaching 0.57. Finally, correlation coefficients of the variables were also examined in search of clarifications for RQ3. These, shown in Sec. 4, indicated higher correlations for variables in the "tweet" subset with the target "sentiment" variable. One reason for the different results with the two algorithms may be due to the particular robustness of the RF model.

Reflecting on the project's challenges, time complexity and memory consumption were two of the most prominent difficulties. Naturally, feature extraction procedures of text variables are known to cause problems in this regard (Kowsari et al., 2019). However, dimensionality reduction strategies such as decreasing the amount of features in text representation as detailed in Sec. 3, helped to overcome this issue.

On another note, a consideration worth making when dealing with "social" data such as tweets, relates to the quality of text from social media platforms. For example, Twitter allows users to upload their ideas in a maximum of 280 characters, which leads to people compressing their writing by employing slang, acronyms, abbreviated forms, and so on (Manguri et al., 2020). With learning from the literature that preprocessing techniques can "make or break" sentiment analysis, with combinations of these improving the outcome and others decreasing it (Uysal & Gunal, 2014), it is worth wondering what the impact of the text preprocessing techniques chosen in this project was. Despite implementing these techniques based on general popularity and consensus from previous studies in the field of COVID-19 vaccine sentiment analysis (Agustiningsih et al., 2021), it would have been interesting to examine other combinations, but time constraints connected to the high dimensionality of the feature extraction procedures did not allow for it. This is an area which would benefit from more experimentation in future research.

6 CONCLUSION

This research project took on the aspiration of predicting the sentiment of tweets regarding the COVID-19 vaccine in Italy. The work was inspired by the thriving fields of text classification and sentiment analysis, and it explored a variety of methodologies, investigating different algorithms, feature extraction techniques and variable subsets.

One of the biggest questions was whether ensemble classifiers did indeed achieve higher accuracy scores than individual classifiers. The algorithms selected were Naïve Bayes (NB), Random Forest (RF) and Extra Trees Classifier (ETC). This was found to be substantially true throughout the various analyses, in line with the literature examined (Amasyali & Ersoy, 2011; Yousaf et al., 2020; Rustam et al., 2021; Rathi et al., 2018). As was previously described in Sec. 2 and 4, ensemble models were created to combine the predictions of different individual classifiers in order to maximise generalisability and robustness, and they demonstrate to be successful in achieving this goal.

A further goal of this research was to investigate two different ways to extract and represent textual features, Bag-of-Words (BoW) and Term frequency-inverse document frequency (Tf-idf). Despite the scarse evidence of either of the two performing better than the other one in terms of accuracy, both methods allowed for the algorithms implemented to achieve very good accuracy scores (Tab. 2), and also, some interesting differences in computational speed emerged (Fig. 5), with BoW proving to allow for faster computations than Tf-IDF. Indeed, this demonstrates the importance of evaluating machine learning architectures from multiple perspectives; models are defined not only by prediction accuracy, but also by factors such as computational speed and memory consumption.

Finally, the last research question regarded feature importance and the predictive power of those variables concerning the tweet - such as its date of creation and its text - versus those variables regarding the user behind the post - such as its number of followers or its verification status. Despite some initial inconclusive results in combination with the RF algorithm, the analysis with the NB model clearly showed the higher predictive power of the tweet-related subset feature, achieving relatively higher accuracy than the user-related ones (Tab. 5.

Ultimately, the main research question was thoroughly studied and yielded a promising result: the sentiment of tweets on the COVID-19 vaccine in Italy can be predicted with significant accuracy. Through the study of the text classification research area, a wide range of innovative and state-of-the-art techniques for successfully processing text for machine learning algorithms were found, contributing significantly to the project's success. Research in this field is indeed essential in order to effectively extract relevant information from social media content. The implementation of the researched strategies on a larger scale, whether in Italy or the rest of the world, is a valuable asset in the hands of decision-makers during critical societal times, such as the recent health crisis caused by COVID-19 and the associated vaccination program.

REFERENCES

Aggarwal, C. C., & Zhai, C. (2012). A survey of text classification algorithms. In *Mining text data* (pp. 163–222). Springer.

Agustiningsih, K. K., Utami, E., & Al Fatta, H. (2021). Sentiment analysis of covid-19 vaccine on twitter social media: Systematic literature review. In *2021 ieee 5th international conference on information technology, information systems and electrical engineering (icitisee)* (p. 121-126). doi: 10.1109/ICITISEE53823.2021.9655960

Akbar, R. (2019). *Ten threats to global health in 2019.* World Health Organization. Retrieved from https://www.who.int/news-room/spotlight/ten-threats-to-global-health-in-2019

Amasyali, M., & Ersoy, O. (2011). Comparison of single and ensemble classifiers in terms of accuracy and execution time. In *2011 international symposium on innovations in intelligent systems and applications* (pp. 470–474).

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Brownlee, J. (2020, Aug). *A gentle introduction to k-fold cross-validation.* Retrieved from https://machinelearningmastery.com/k-fold-cross-validation/

Ceballos, F. (2020a, Apr). *An intuitive explanation of random forest and extra trees classifiers.* Towards Data Science. Retrieved from https://towardsdatascience.com/an-intuitive-explanation-of-random-forest-and-extra-trees-classifiers-8507ac21d54b

Ceballos, F. (2020b, Apr). *Scikit-learn decision trees explained.* Towards Data Science. Retrieved from https://towardsdatascience.com/scikit-learn-decision-trees-explained-803f3812290d

Elia, F. (2020, Jun). *Stemming vs lemmatization.* Retrieved from https://www.baeldung.com/cs/stemming-vs-lemmatization

Gasparetto, A., Marcuzzo, M., Zangari, A., & Albarelli, A. (2022). A survey on text classification algorithms: From text to predictions. *Information*, *13*(2), 83. doi: 10.3390/info13020083

Gupta, A. (2020, Jul). *Ml: Extra tree classifier for feature selection.* Retrieved from https://www.geeksforgeeks.org/ml-extra-tree-classifier-for-feature-selection/

Honnibal, M., & Montani, I. (2017). *spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.*

Huang, X., Li, Z., Wang, C., & Ning, H. (2019). Identifying disaster related social media for rapid response: a visual-textual fused cnn architecture. *International Journal of Digital Earth*.

Iqbal, M. (2022, May). *Twitter revenue and usage statistics (2022).* Retrieved from https://www.businessofapps.com/data/twitter-statistics/

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

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. *Information*, *10*(4). Retrieved from https://www.mdpi.com/2078-2489/10/4/150

Leung, K. M. (2007). Naive bayesian classifier. *Polytechnic University Department of Computer Science/Finance and Risk Engineering*, *2007*, 123–156.

Limaye, R. J., Holroyd, T. A., Blunt, M., Jamison, A. F., Sauer, M., Weeks, R., . . . Gellin, B. (2021). Social media strategies to affect vaccine acceptance: a systematic literature review. *Expert Review of Vaccines*, *20*(8), 959-973. Retrieved from https://doi.org/10.1080/14760584.2021.1949292 (PMID: 34192985) doi: 10.1080/14760584.2021.1949292

Manguri, K. H., Ramadhan, R. N., & Mohammed Amin, P. R. (2020). Twitter sentiment analysis on worldwide covid-19 outbreaks. *Kurdistan Journal of Applied Research*, 54–65. doi: 10.24017/covid.8

Nair, A. (2021, Nov). *Leveraging n-grams to extract context from text.* Towards Data Science. Retrieved from https://towardsdatascience.com/leveraging-n-grams-to-extract-context-from-text-bdc576b47049

Neely, S. R., Eldredge, C., Ersing, R., & Remington, C. (2022). Vaccine hesitancy and exposure to misinformation: a survey analysis. *Journal of general internal medicine*, *37*(1), 179–187.

Pierri, F. (2022). *Vaccineu: Covid-19 vaccine conversations on twitter in french, german and italian [replication data].*

Pimpalkar, A. P., & Raj, R. J. R. (2020). Influence of pre-processing strategies on the performance of ml classifiers exploiting tf-idf and bow features. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, *9*(2), 49.

Rahman, A., & Tasnim, S. (2014). Ensemble classifiers and their applications: A review. *arXiv preprint arXiv:1404.4088*.

Rao, D. (2015). *Testing promoted tweets on our logged-out experience.* Twitter. Retrieved from https://blog.twitter.com/en_us/a/2015/testing-promoted-tweets-on-our-logged-out-experience

Rathi, M., Malik, A., Varshney, D., Sharma, R., & Mendiratta, S. (2018). Sentiment analysis of tweets using machine learning approach. In *2018 eleventh international conference on contemporary computing (ic3)* (pp. 1–3).

Roldós, I. (2021, May). *Text cleaning for nlp: A tutorial.* Retrieved from https://monkeylearn.com/blog/text-cleaning/

Rufai, S. R., & Bunce, C. (2020). World leaders' usage of twitter in response to the covid-19 pandemic: a content analysis. *Journal of public health*, *42*(3), 510–516.

Rustam, F., Khalid, M., Aslam, W., Rupapara, V., Mehmood, A., & Choi, G. (2021). A performance comparison of supervised machine learning models for covid-19 tweets sentiment analysis. *PLoS ONE, 16(2)*.

Samuel, J., Ali, G., Rahman, M., Esawi, E., Samuel, Y., et al. (2020). Covid-19 public sentiment insights and machine learning for tweets classification. *Information*, *11*(6), 314.

scikit learn. (2022). *Ensemble methods.* Retrieved from https://scikit-learn.org/stable/modules/ensemble.html

spacy.io. (2022). Retrieved from https://spacy.io/

Sunil, R. (2021, Aug). *Learn naive bayes algorithm: Naive bayes classifier examples.* Retrieved from https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

the Now, D. (2020). *Hydrator.* Retrieved from https://github.com/docnow/hydrator

Twitter.com. (2022). *Tweet object | docs | twitter developer platform.* Twitter. Retrieved from https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/object-model/tweet

Uysal, A. K., & Gunal, S. (2014). The impact of preprocessing on text classification. *Information processing & management*, *50*(1), 104–112.

Vasiliev, Y. (2020). *Natural language processing with python and spacy a practical introduction.* No Starch Press, Inc.

Vidhya, A. (2021, Jun). *Text preprocessing nlp: Text preprocessing in nlp with python codes.* Retrieved from https://www.analyticsvidhya.com/blog/2021/06/text-preprocessing-in-nlp-with-python-codes/

Wankhade, M., Rao, A. C. S., & Kulkarni, C. (2022). A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, 1–50.

Wilson, S. L., & Wiysonge, C. (2020). Social media and vaccine hesitancy. *BMJ Global Health*, *5*(10), e004206.

Yousaf, A., Umer, M., Sadiq, S., Ullah, S., Mirjalili, S., Rupapara, V., & Nappi, M. (2020). Emotion recognition by textual tweets classification using voting classifier (lr-sgd). *IEEE Access*, *9*, 6286–6295.

Zhu, L., Zhou, X., & Zhang, C. (2021). Rapid identification of high-quality marine shale gas reservoirs based on the oversampling method and random forest algorithm. *Artificial Intelligence in Geosciences*, *2*, 76-81. Retrieved from https://www.sciencedirect.com/science/article/pii/S2666544121000307 doi: https://doi.org/10.1016/j.aiig.2021.12.001

APPENDIX

APPENDIX A, BOW CLASSIFICATION

Table 6: Confusion matrix for BoW features, with NB

	Predicted	
Actual	804	1666
	630	1900

Table 7: Confusion matrix for BoW features, with RF

	Predicted	
Actual	2233	237
	221	2309

Table 8: Confusion matrix for BoW features, with ETC

	Predicted	
Actual	2126	344
	395	2135

Table 9: Classification report for BoW features, with NB

	precision	recall	f1-score	support
positive	0.56	0.36	0.41	2470
negative	0.53	0.75	0.62	2530
accuracy			0.54	5000
macro avg	0.55	0.54	0.52	5000
weighted avg	0.55	0.54	0.52	5000

Table 10: Classification report for BoW features, with RF

	precision	recall	f1-score	support
positive	0.91	0.91	0.91	2470
negative	0.91	0.90	0.91	2530
accuracy			0.91	5000
macro avg	0.91	0.91	0.91	5000
weighted avg	0.91	0.91	0.91	5000

Table 11: Classification report for BoW features, with ETC

	precision	recall	f1-score	support
positive	0.84	0.86	0.85	2470
negative	0.86	0.84	0.85	2530
accuracy			0.85	5000
macro avg	0.85	0.85	0.85	5000
weighted avg	0.85	0.85	0.85	5000

APPENDIX B, TF-IDF CLASSIFICATION

Table 12: Confusion matrix for Tf-IDF features, with NB

	Predicted	
Actual	804	1666
	630	1900

Table 13: Confusion matrix for Tf-IDF features, with RF

	Predicted	
Actual	2237	233
	223	2307

Table 14: Confusion matrix for Tf-IDF features, with ETC

	Predicted	
Actual	2136	334
	274	2256

Table 15: Classification report for Tf-IDF features, with NB

	precision	recall	f1-score	support
positive	0.56	0.33	0.41	2470
negative	0.53	0.75	0.62	2530
accuracy			0.54	5000
macro avg	0.55	0.54	0.52	5000
weighted avg	0.55	0.54	0.52	5000

Table 16: Classification report for Tf-IDF features, with RF

	precision	recall	f1-score	support
positive	0.91	0.91	0.91	2470
negative	0.91	0.91	0.91	2530
accuracy			0.91	5000
macro avg	0.91	0.91	0.91	5000
weighted avg	0.91	0.91	0.91	5000

Table 17: Classification report for Tf-IDF features, with ETC

	precision	recall	f1-score	support
positive	0.89	0.86	0.88	2470
negative	0.87	0.89	0.88	2530
accuracy			0.88	5000
macro avg	0.88	0.88	0.88	5000
weighted avg	0.88	0.88	0.88	5000

appendix c, "tweet" variables classification

Table 18: Tweet related variables.

Variable	Data Type
"coordinates"	Object
"created_at"	Object
"favorite_count"	Integer
"id"	Integer
"in_reply_to_screen_name"	Object
"in_reply_to_status_id"	Float
"in_reply_to_user_id"	Float
"lang"	Object
"place"	Object
"possibly_sensitive"	Object
"quote_id"	Float
"retweet_count"	Integer
"retweet_id"	Float
"retweet_screen_name"	Object
"source"	Object
"text"	Object
"tweet_url"	Object

Table 19: Confusion matrix for "tweet" features, with NB

	Predicted	
Actual	1629	841
	634	1896

Table 20: Confusion matrix for "tweet" features, with RF

	Predicted	
Actual	2110	360
	325	2205

Table 21: Classification report for "tweet" features, with NB

	precision	recall	f1-score	support
positive	0.72	0.66	0.69	2470
negative	0.69	0.75	0.72	2530
accuracy			0.70	5000
macro avg	0.71	0.70	0.70	5000
weighted avg	0.71	0.70	0.70	5000

Table 22: Classification report for "tweet" features, with RF

	precision	recall	f1-score	support
positive	0.87	0.85	0.86	2470
negative	0.86	0.87	0.87	2530
accuracy			0.86	5000
macro avg	0.86	0.86	0.86	5000
weighted avg	0.86	0.86	0.86	5000

appendix d, "user" variables classification

Table 23: User related variables.

Variable	Data Type
'user_created_at'	Object
'user_id'	Integer
'user_default_profile_image'	Boolean
'user_description'	Object
'user_favourites_count'	Integer
'user_followers_count'	Integer
'user_friends_count'	Integer
'user_listed_count'	Integer
'user_location'	Object
'user_name'	Object
'user_screen_name'	Object
'user_statuses_count'	Integer
'user_time_zone'	Float
'user_urls'	Object
'user_verified'	Boolean

Table 24: Confusion matrix for "user" features, with NB

	Predicted	
Actual	1390	1080
	1058	1472

Table 25: Confusion matrix for "user" features, with RF

	Predicted	
Actual	2179	360
	325	2205

Table 26: Classification report for "user" features, with NB

	precision	recall	f1-score	support
positive	0.57	0.56	0.57	2470
negative	0.58	0.58	0.58	2530
accuracy			0.57	5000
macro avg	0.57	0.57	0.57	5000
weighted avg	0.57	0.57	0.57	5000

Table 27: Classification report for "user" features, with RF

	precision	recall	f1-score	support
positive	0.85	0.88	0.87	2470
negative	0.88	0.85	0.86	2530
accuracy			0.86	5000
macro avg	0.86	0.86	0.86	5000
weighted avg	0.86	0.86	0.86	5000