ERASMUS UNIVERSITY ROTTTERDAM

Erasmus School of Economics

Bachelor Thesis BSc^2 Econometrics / Economics

A horse race of eight machine learning methods to catch trumpery on Twitter

Jeroen Weijenborg

481763

Abstract

In this paper, I use eight different classification methods and examine their performance in identifying deceptive and truthful tweets. The models consist of a benchmark (logistic regression), classical methods (Naïve Bayes, Support Vector Machine and k-Nearest Neighbors), decision-tree based methods (Recursive Partitioning, Random Forests and XGBoost) and a deep neural network (Convolutional Neural Network). The models are trained on 4,599 tweets and tested on 3,066 tweets by the 45th president of the U.S., Donald Trump. I find 73 significant linguistic variables that help explain the difference between factually correct and incorrect tweets. The out-of-sample classification performance forms the basis for the comparison of the models: XGBoost attained the highest accuracy of all, with 74.69%. The difference between the performance of most other methods was relatively small, except for the worst performing Recursive Partitioning. Overall, no method or group of methods is the winner in every respect.

Supervisor: A. Baillon

Second assessor: R.L. Lumsdaine

July 4, 2021

Contents

1 Introduction

With the emergence of social media platforms, individuals have gotten access to different ways of acquiring information. Where the information used to be provided by television outlets, radio stations and newspapers, it now is available in countless formats. It is no longer necessary to be a professional journalist to spread information on a large scale. Every single person can create an account and share a message. While it is desirable to be free in choosing where to get information from, it also comes with a substantial threat: misleading information. Misleading information, or fake news, is defined as the deliberate presentation of fake or misleading claims as news, where these are misleading by design (Gelfert, 2018). The phrase 'by design' here refers to systemic features of the design of the sources and channels by which misinformation propagates and, thereby, possibly manipulates the audience. This manipulation poses a serious threat to the knowledge of the general public, thus also to democratic elections (Lee, 2019). Especially the two-party system in the USA is susceptible to distortion, as misinformation strengthens the partisan views, resulting in more polarisation.

One influential person that is considered to intentionally manipulate the masses via the internet is the 45th president of the United States: Donald Trump. During his presidency from 2017 to 2021, he has been active on different social media platforms, among which Twitter and Facebook. Of all different official communication channels, Trump has the most control of his Twitter account (Barbaro, 2015). In most cases, there is no team of writers that prepares his tweets, contrary to his official television speeches, for example. His tweets are therefore unfiltered, allowing him to spread false information whenever he wants. The fact that his claims of having won the 2020 elections formed the basis of the storming of the Capitol in January 2021 shows the impact his (false) tweets can have. Several journalism organisations, such as the Washington Post, have fact-checked all of his tweets in an attempt to prevent the false statements to manipulate the general public (Kessler et al., 2021). The social media platforms themselves are criticised for their lack of actions against misinformation. Twitter only started flagging tweets as potentially containing false information in May 2020. In order to reduce the threat of misleading information, it is important to adequately identify false claims in tweets.

Van Der Zee, Poppe, Havrileck & Baillon (forthcoming) introduce a deception detection model tailored to Donald Trump. They find that factually correct and incorrect tweets contain linguistic differences. Their model attains a classification accuracy of approximately 73% and performs better than other linguistic deception detection models in the literature. This paper builds on the research by Van Der Zee et al. and aims to improve the results in a few ways, i.e., by examining different machine learning classification methods and by increasing the data set.

This increases the model performance and thus makes the deception detection more useful in practice. For instance, platforms such as Twitter could build an automatic linguistic deception detection feature based on this research.

My research contributes to the existing literature in several ways. Firstly, I address a shortcoming in the forthcoming research by Van Der Zee et al. concerning the relatively small data set used, ranging from November 2017 until April 2018 and consisting of 953 observations. To obtain more robust results, I use a larger part of all tweets sent by the 45th president, @realDonaldTrump on Twitter, during his presidency. Furthermore, I investigate different classification methods and compare them to the 'benchmark' method, which is the logistic regression, as used in Van Der Zee et al. (forthcoming). For example, three classical methods (Support Vector Machines, k-Nearest Neighbors and Naïve Bayes) performed well in previous related work (Ott et al., 2011; Mihalcea and Strapparava, 2009). Additionally, I include a deep learning method (Convolutional Neural Network) and three different decision-tree based methods (Recursive Partitioning, Random Forests and XGBoost) that are often used in the literature (Kotsiantis et al., 2007). Finally, the implications of these different machine learning methods in combination with a larger data set could be interesting to social media platforms, such as Twitter, since it allows them to immediately place a preliminary true/false flag on tweets by influential individuals.

This paper is structured as follows. Section 2 provides a more elaborate literature review on the research topic. The data set is described in Section 3. Section 4 details the different methods used in the classification of the tweets. The results are presented in Section 5. Section 6 contains a discussion of the results and directions for future work are given. Finally, in Section 7 the research is concluded.

2 Literature review

Over the last few years, an increasing amount of research has been conducted on misleading information, or the connotative term 'fake news'. Some papers examine the influence of fake news on U.S. presidential elections, such as Allcott and Gentzkow (2017), while other papers pay attention to fake news as a threat to public health (Waszak et al., 2018) or to the impact fake news has on marketing (Di Domenico et al., 2021). Either way, the term 'fake news' has become popular from 2016 onward, which is likely correlated with Donald Trump and the 2016 U.S. presidential elections. His presence on Twitter allowed him to directly communicate with his followers, which contributed to his popularity.

Rubin (2017) writes that social media users usually hold a presumption of goodwill in the communication through social media. This is in line with the psychological findings by Evans and

Krueger (2009). They find that observed behavior reveals widespread trust and trustworthiness in strangers. It is thus not surprising that social media users are inclined to believe what they read on the platforms, where most of the content is created by strangers. A research by Morris et al. (2012) specifically states that Twitter users "are poor judges of truthfulness based on content alone, and instead are influenced by heuristics such as username when making credibility assessments". This implies that the social media users are vulnerable to intentional manipulation of news. Consequently, there is a clear role for the companies that own these platforms to reduce the vulnerability of its users.

The demand for regulation on the basis of effective deception detection models has increased, because it has been shown that statements by liars are linguistically different from truth tellers (Hauch et al., 2015). For example, liars tend to express more negative emotions, experience a greater cognitive load, use fewer first-person pronouns and provide fewer details. The paper by Bachenko et al. (2008) is one of the first researches in the literature to implement a system that automatically identifies deceptive and truthful statements. They examine civil and criminal narratives with a model that consists of several deception indicators, such as phrases that show the suspect's lack of commitment to make a statement (e.g., "later that day"). Their Classification and Regression Tree analysis attains an accuracy of nearly 75% and marked the beginning of automatic deception detection, which clearly outperforms the human's ability to distinguish truths from lies. A comparison of 39 studies examining the ability of humans to detect lies reports a mean accuracy of only 56.6% (Vrij, 2000).

The last decade, many existing classification techniques have been improved and new techniques have been introduced. Within the field of text classification, one of the simplest and most popular classification algorithms is the logistic regression. It is simple to implement yet proves to be efficient in many cases. The analysis in Van Der Zee et al. (forthcoming) is based on this approach. Therefore, the logistic regression functions as the benchmark in this research.

The other classification methods can be divided in different groups, primarily based on the similarity of their functioning. Figure 1 displays the division of the methods into the benchmark method, the classical methods, decision-tree based methods and a deep learning method. For simplicity, I grouped Support Vector Machine and k-Nearest Neighbors together with Naïve Bayes under the umbrella term 'classical methods', even though the former two methods differ from the latter one in the way they work. The reason for this grouping is that these classifiers are widely used in the literature and are among the oldest methods in machine learning. The second group consists of decision-tree based methods, which use decision trees in their classification. Thirdly, the group of deep learning methods consists of an artificial neural network, the only unsupervised method included in this research.

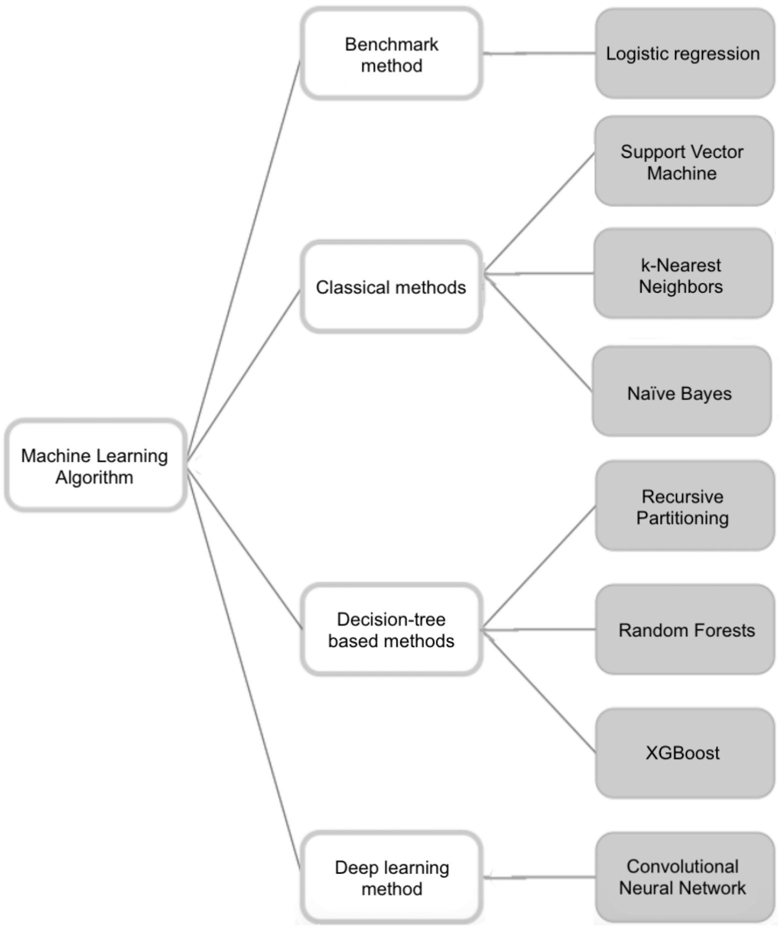


Figure 1: Overview of the different classification methods. The white shapes denote the groups, the grey shapes denote the classifiers.

The three classical methods are among the oldest machine learning methods that exist in the literature. Firstly, the Naïve Bayes classifier is computationally inexpensive but still effective, which made it a popular method when classification was relatively new. However, it has been shown that Naïve Bayes does not perform well in all automated text classification cases due to incorrect parameter estimation (Kim et al., 2006).

Support Vector Machines (SVM) are a more sophisticated method and emerged early 2000 due to their attractive features and the promising results. They are computationally somewhat expensive, but can handle high dimensions of data (Gunn et al., 1998). Since text classification usually includes a lot of linguistic aspects, SVMs are widely applied in the literature.

The third classical method, k-Nearest Neighbors (kNN), is a well-known algorithm in machine learning too, especially for pattern recognition. It is also used in text classification, but often has a poorer performance in this area, especially due to it being solely dependent on the training set and the calculation complexity (Suguna and Thanushkodi, 2010). However, with the correct specifications (e.g., a suited number of neighbors k), kNN can perform well.

The decision-tree based methods in this research are each other's successor, with Recursive Partitioning (rPart) being the oldest. Recursive Partitioning is a multivariate machine learning analysis that creates a single decision tree to classify objects. The method is easily implemented, with fast computations and intuitive results. Instead of calculating a direct probability (as in a regression), rPart more or less states a rule that if a text meets certain characteristics, it belongs to a specific category. The most prevalent use of rPart is in the medical discipline due to its clinical usefulness (James et al., 2005), though it is also applied in finance (Duttagupta and Cashin, 2008; Anil Kumar and Ravi, 2008) and text recognition (De Silva and Hull, 1994), among other fields.

An improvement and more state-of-the-art version of a decision tree is the method Random Forests (rf). This analysis technique consists of a multitude of trees simultaneously, where the output is determined by the majority of the trees. As a result, it is a more stable alternative to rPart. Additionally, Random Forests always converge, overcoming the problem of overfitting (Breiman, 2001). In a research by Fernández-Delgado et al. (2014), 179 classifiers are evaluated on 112 data sets of real world classification problems. Their results show that Random Forests are generally the best performing classifiers.

The successor of Random Forests is the recently introduced method XGBoost, a scalable machine learning system for tree boosting (Chen and Guestrin, 2016). XGBoost combines a multitude of decision trees with weight and loss function optimization and can be applied to all different scenarios, including regression and classification. The results in Chen and Guestrin (2016) demonstrate that XGBoost delivers great results and runs up to ten times faster than other popular machine learning methods.

Lastly, in the text classification algorithm survey paper by Kowsari et al. (2019), deep neural networks are extensively discussed as the researchers consider them to be of great value compared to alternative classifiers. These unsupervised learning approaches have proven to be successful in a wide range of tasks such as natural language processing and image recognition, as well as text classification. Different objectives require the neural network to be designed in a different manner, i.e., the number and type of layers matters. The major disadvantage of neural networks is that one cannot track down why the algorithm classifies objects the way it does.

Based on the literature, the most promising results are expected to come from the newer methods: Random Forests, XGBoost or a Convolutional Neural Network. The benchmark and the classical method SVM, though, could also perform well due to their simplicity and high dimensionality handling, respectively.

3 Data

In order to perform the classification analysis, I collected a data set of tweets by president Donald Trump, @realDonaldTrump on Twitter.[1] The tweets are extracted from the Trump Archive, which is a publicly available data base containing all tweets sent by the president. Trump's Twitter account has been permanently suspended as of January 8th, 2021, meaning that the number of retweets and likes have not increased since then. The time span of the collection is November 1, 2017 - April 3, 2020, covering a large part of his presidency. In total, this results in a data set of 14,714 tweets over 2.5 years. The choice to include two and a half year of data is mainly due to the labor intensity of the data collection and time constraints. However, the data set is large enough to obtain robust results and to be able to draw a reliable conclusion with respect to the performance of the different models. Namely, when looking at Figure 2, it is evident that some periods are characterized by relative tranquility, whereas other periods are more chaotic. During these tumultuous periods, the president might be less reluctant to lie since he faces more problems. It is therefore important to test the consistency of the model over a sample as large as possible, instead of only over a few months.

The Washington Post has created a publicly available database containing all false or misleading claims made by Donald Trump while in office. All tweets in the data set can thus be marked as either correct or incorrect, under the crucial assumption that The Washington Post has correctly identified misleading tweets. The correctness of an observation is stored in the binary variable veracity.

Not all tweets in the data set are useful, in a sense that it includes, for example, retweets and tweets sent by representatives of Trump. On March 29th, 2017, the White House director of social media Dan Scavino Jr. tweeted that Trump switched to an iPhone, thus confirming that the tweets coming from an iPhone are from the president himself (Scavino, 2017). As a result, I removed 277 tweets that were likely not sent by Trump himself. These include the tweets coming from Twitter Media Studio (212 tweets), Twitter Web Client (41 tweets) and Twitter for iPad (24 tweets). Additionally, I removed 5,006 retweets, 340 duplicate tweets and 1,426 tweets containing only links to a website or tweets containing quotes of more than six words, as these would contaminate the data set. The final data set consists of 7,665 tweets, with 26.9% of the tweets classified as incorrect.

[1]The collection is done in cooperation with fellow students Lucas Chau, Tommy Hu and Martijn Rigutto.

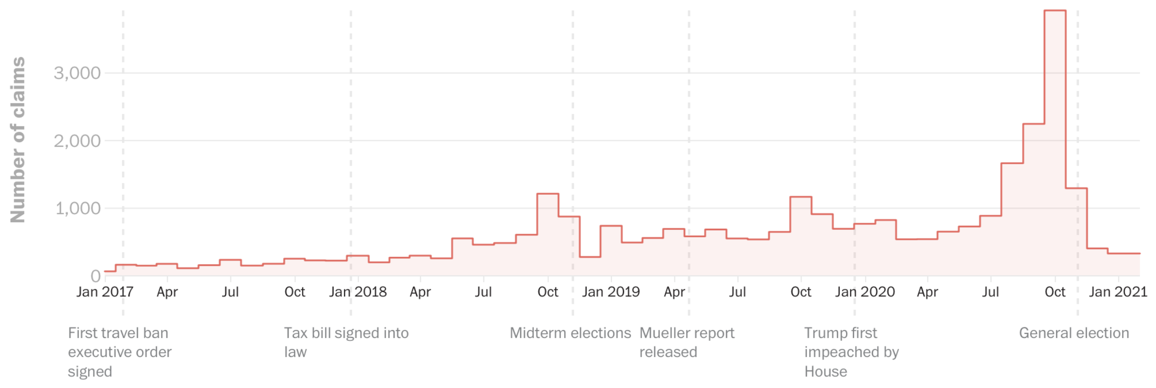


Figure 2: All false statement claims by Donald Trump, as flagged by The Washington Post. Monthly plotted, through all official communication channels. Source: The Washington Post.

3.1 Linguistic Data Processing

To obtain variables that are of assistance in the classification process, I use the text analysis program Linguistic Inquiry and Word Count (LIWC; Pennebaker et al., 2015). The program calculates the percentage of words in a tweet that belong to certain categories. Various social, cognitive and affective processes are recognized in the text and are subsequently assigned to different linguistic (i.e., adverbs, dictionary words, personal pronouns), psychological (i.e., swear words, sadness, positive emotions), topical (i.e., religion, money) and Twitter-specific categories (i.e., @, #). I follow the variable choice of Van Der Zee et al. (forthcoming) and therefore include 103 variables, consisting of 101 LIWC word categories and two Twitter-specific variables. Except for the variable word count, all variables are percentages, expressed as the relative occurrence of the words in one tweet corresponding to a particular word category.

In addition, I examine the significance of three other variables. The first two being the number of retweets and likes on a tweet, since these might be an indication of how controversial the message is, potentially due to false claims made. Additionally, a dummy variable indicating a period of 'crisis' might be of interest. In this case, a crisis would be defined as either a period in which the president is personally in trouble (such as the impeachment at the end of 2019) or a period in which there are external problems that are centered around the president and/or his interventions (such as the midterm elections). The exact periods I considered a crisis are stated in Table 1. Though this involves a degree of subjectivity, I have only included periods that are political events, and not purely personal problems (for example, the Stormy Daniels-Donald Trump scandal). Of the total 7,665 tweets, 1,323 (17.3%) tweets were sent in a crisis period and 6,342 (82.7%) were sent in a non-crisis period. The analysis in this paper thus includes the 103 variables as in Van Der Zee et al. (forthcoming) plus the three variables described above, totalling 106 variables.

Table 1: *All events in the data set time span that are marked as a crisis*

Event	Period
Midterm elections	October 15th 2018 - November 15th 2018
Government shutdown	December 22th 2018 - January 30th 2019
Mueller report	March 22th 2019 - April 18th 2019
First impeachment	December 18th 2019 - February 5th 2020

4 Methodology

In short, I first examine the different classification methods on the 7.665 tweets, allowing the consistency and overall performance of the models to be compared. For this comparison, I use several performance measures, including accuracy. As a robustness check of the results, I repeat the analysis of the classification methods on the same data set as in the forthcoming Van Der Zee et al., which appears in Appendix A.

For this analysis, I split the data into a training set and test test. The training set makes up 60% of the total sample (4,599 tweets) and covers the period November 2017 until mid July 2019. The test set ranges from mid July 2019 to April 2020 and consists of the other 40% (3,066 tweets). I follow the procedure by Van Der Zee et al. (forthcoming) and run a multivariate analysis of variance (MANOVA) to determine the significance of the 106 variables, such that a forward-stepwise logistic regression can select which significant variables to include in the model.[2] This stepwise-logistic regression uses the Akaike Information Criterion (AIC) to perform the model selection. The iterative process starts with only the variable that has the highest absolute Cohen's d value and then keeps adding a variable until the AIC no longer decreases. This direction of the step function is forward looking (i.e., starting with only a single variable). Alternatively, the backward stepwise model starts with all variables significant at 5% and then drops one at a time. I do not deviate from Van Der Zee et al. (forthcoming) and choose the forward stepwise-logistic model, which functions as the benchmark model in this research.

Then, I run the selected logit model on the training set with the dependent variable being the veracity variable. The predictive ability of this trained model is tested on the test set. Subsequently, the train and test process is repeated for the classification methods NB, SVM, kNN, rPart, rf, XGBoost and CNN. All methods use the same variables as the logit model, except for CNN, which uses the tweets themselves as input. Besides, all models are trained

[2]The p-values are adjusted with False Detection Rate (FDR) using the method by Benjamini and Hochberg (1995).

using 10-fold cross validation. The functioning of the different methods is explained below.

4.1 Classical methods

4.1.1 Support Vector Machine

This classification method, developed by Vapnik (Vapnik, 1995), takes in data points and outputs a hyperplane that best separates the observations. This hyperplane is called the decision boundary, where all points on a particular side of the boundary are classified as belonging to the same class. The optimal decision boundary is computed such that the distance between the different classes and the hyperplane is maximized. This can be done easily for linear data. However, nonlinear data requires a kernel trick that smartly represents the original data in order to compute the separating hyperplane. The kernel trick uses observations in the lower dimensional space (original data) and outputs their dot product in the higher dimensional space. This output is linearly separable and thus does not require the original data to be actually transformed to higher dimensions, which is computationally much more expensive than the dot product. Therefore, advantages of this method are that SVMs do not require expensive computations and that they are effective in high-dimensional data. Given the large number of explanatory variables in this research, the SVM tool is well suited. However, one disadvantage is that the SVM method lacks probabilistic interpretation. It classifies objects whenever they fall on a specific side of the hyperplane, which has no implications for the within-class distribution. The effectiveness is therefore hard to measure.

4.1.2 Naïve Bayes

Contrary to SVM, Naïve Bayesian classification methods are purely probabilistic. Naïve Bayes is based on the Bayesian theory, which is used to calculate conditional probabilities. This theorem is given by

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \tag{1}$$

where $P(A|B)$ is the conditional probability of event A occurring, given the event B and where $P(A)$ is the probability of event A occurring. In this research, event A is equivalent to 'false tweet' and event B is the set of explanatory variables. More specifically, let C_0 and C_1 denote the two classes, where the former corresponds to 'correct tweets' and the latter to 'false tweets'. Besides, the explanatory variables are given by $x_1, x_2, ..., x_n$, where n is the number of explanatory variables included in the model. The probability of flagging a tweet as incorrect, given the values of the explanatory variables is denoted as

$$P(C_1|x_1, x_2, ..., x_n) = \frac{P(x_1, x_2, ..., x_n|C_1)P(C_1)}{P(x_1, x_2, ..., x_n)}. \tag{2}$$

Applying the definition of conditional probabilities repeatedly yields

$$P(C_1|x_1, x_2, ..., x_n) = \frac{P(x_1|x_2, ..., x_n, C_1)P(x_2|x_3, ..., x_n, C_1)...P(x_{n-1}|x_n, C_1)P(x_n|C_1)P(C_1)}{P(x_1, x_2, ..., x_n)}. \tag{3}$$

However, Naïve Bayes assumes conditional independence, meaning that all explanatory variables are independent. Consequently, $P(x_k|x_{k+1}, ..., x_n, C_1)$ reduces to $P(x_k|C_1)$ for $k = 1$, ..., $n - 1$. Finally, the equation can be expressed as

$$P(C_1|x_1, x_2, ..., x_n) = \Pi_{j=1}^{j=n} P(x_j|C_1)\frac{P(C_1)}{P(x_1, x_2, ..., x_n)}. \tag{4}$$

Since all information on the right hand side of Equation 4 can be computed, all observations can be classified as containing false information or not. The assumption of conditional independence might be problematic in this research, because the LIWC variables likely capture similar linguistic aspects of the data. Hence, the independence of the explanatory variables might be violated.

4.1.3 k-Nearest Neighbors

The non-parametric classification method k-Nearest Neighbors assigns observations to a specific class based on the points closest to this observation. Figure 3 displays three visualised examples of the algorithm, where the star is the data point of interest. In the left example, the class of the star is determined on the basis of the nearest neighbor, which is in this case negative. Therefore, it will be classified as negative too. For the middle example, the class will be chosen randomly, as the two neighbors both belong to a different class. In the right example, the data point will be assigned to the negative class, since the majority of the three neighbors are in the negative class. This example highlights the importance of the choice of the number of neighbors included.

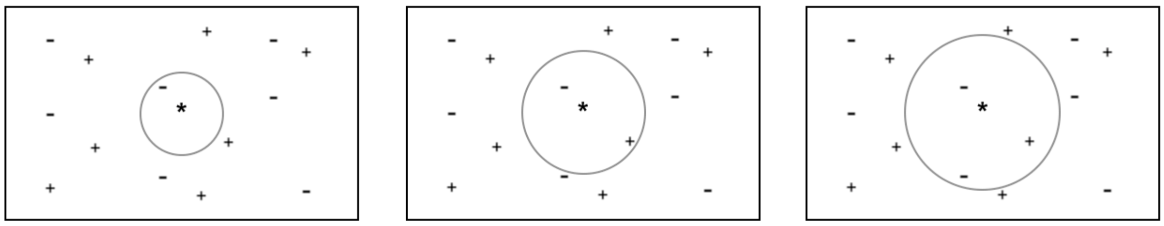


Figure 3: The 1-, 2-, and 3-nearest neighbors.

The many linguistic variables in this research are mapped to a high dimensional space (i.e., the number of variables included). The kNN algorithm relies on the crucial assumption that points corresponding to the same class are closely located to each other in this high dimensional space. For this research, the distance between the data points is computed with the Manhattan distance, as this is found to yield good results in high dimensional data (Mulak and Talhar, 2015). The Manhattan distance between two points $\mathbf{u} = (u_1, u_2, ..., u_n)$ and $\mathbf{v} = (v_1, v_2, ..., v_n)$ in the n-dimensional space is defined as the sum of the distance between the two points in each dimension:

$$d(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{n} |u_i - v_i|. \tag{5}$$

To prevent random classification from happening, as is the case in the middle example of Figure 3, I choose the number of neighbors to be an odd number. Besides, this number determines how conservative or optimistic the predictions are. With a number around the value of 10, the classifier has enough surrounding points to make a robust decision. On the other hand, 10 is small enough to identify the so-called clusters of false tweets. The probability to include a correct tweet when increasing the search neighborhood by one is larger than the probability to include an incorrect tweet, because approximately 70% of all tweets are correct. So, in order for the algorithm to classify tweets as incorrect, the neighborhood in which the search is conducted should not be too large. More specifically, the value of 10 is small enough to find at least 6 incorrect tweets when the algorithm is searching in a cluster of incorrect tweets. A value larger than 10 would require more incorrect tweets in the search neighborhood in order for the model to still classify the observation as incorrect, which becomes less likely as the number of neighbors increases. In other words, the model remains considerably optimistic with 10 neighbors.

Since the number of neighbors should be odd, I set k equal to be 11.

4.2 Decision-tree based methods

4.2.1 Recursive Partitioning

With the recursive partitioning method, a single decision tree is created, constructed from the training set. It starts with the most influential variable as a single node. The node then branches into the possible values corresponding to this variable. For example, if the first node of the tree is the age of a person, being older than 50 years or not would be an option for the branching. The two new nodes (younger than 50 years, older than 50 years) can then branch into additional nodes based on other variables (e.g., income, sex). This recursive process keeps adding new levels in the tree until a certain stopping condition is met. In this study, the stopping condition is determined by the fit of the model: if a split does not increase the fit of the model by a factor x, the split is not attempted. This factor x is called the complexity parameter. By default, the parameter equals 0.01. To allow the tree to grow more than by default, I reduce this value by half and set it equal to 0.005.

As this method concerns only the construction of a single tree, the method runs fast. The downside of using only a single tree is that it is prone to prediction mistakes.

4.2.2 Random Forests

Random Forests is a more sophisticated, but slower ensemble learning method. Random Forests grows a multitude of trees simultaneously. The output is then determined by the majority output of the trees. The constructed trees are similar to those described in the Recursive Partitioning section.

I grow 3000 decision trees simultaneously, to ensure that all levels of the tree get predicted at least once in the training set. This also allows me to obtain a stable overview of the variable importance in the classification. Moreover, I set the number of randomly selected variables as candidates at each split equal to the square root of the number of variables, because Breiman (2001) reaches optimum results in classification when doing so.

4.2.3 XGBoost

The more recently introduced XGBoost, which also incorporates decision trees, usually outperforms Random Forests (Piryonesi and El-Diraby, 2020; Hastie et al., 2009). In addition, the XGBoost algorithm runs significantly faster than Random Forests due to improvements by Chen and Guestrin (2016), such as better handling of sparse data. XGBoost takes a more iterative approach in the classification process: instead of training all trees at the same time independently of each other, XGBoost uses the information from the previous tree to construct a better tree next. In particular, trees are added that predict the residuals of preceding trees. Then, a combination of the newer trees and prior trees results in the final prediction. XGBoost is a so-called gradient tree boosting algorithm, because it minimizes the loss upon adding decision trees with the use of a gradient descent algorithm.

Regarding the parameter specification of the XGBoost model, I set the learning rate, or the shrinkage factor, equal to 0.05 (within the possible range of [0,1]). This prevents the model from overfitting, but makes it slightly slower to run. An advantage of this model is that it automatically determines the optimal number of trees to build in order to minimize the cross-validation accuracy error, which increases the quality of the output.

4.3 Deep learning method: Convolutional Neural Network

Unlike the other methods, the Convolutional Neural Network is the only algorithm that does not use the LIWC variables to determine the classification. Instead, it is trained entirely on the tweets themselves. This makes the neural network a black box, as it is unclear why the model assigns certain tweets to the true class and others to the false class. I base my Convolutional Neural Network on the model by Kim (2014), whose model is widely used in sentence classification. It consists of an embedding layer, which maps words to vectors, followed by a convolutional layer, a max pooling layer and three dense layers that eventually deliver the output.

To prepare the data, the tweets are first vectorized: all 9,901 unique words in the train set are represented by an integer.

The first hidden layer is the embedding layer, which turns these integers into dense vectors of fixed size. This ensures that the tweet is encoded as a sequence of integers, such that words with similar meanings are represented in a similar way in the vector space.

It is followed by the most important layer in this model; the convolutional layer. Simply stated, it detects features in the input data by applying different filters (or: kernels) repeatedly to the large matrices of data. Figure 4 shows an example of a convolutional process, where the input data is comparable to the embedded tweets in this research. The features are computed by multiplying the values that are on top of each other when the filter is 'placed' over the input data. The filter then slides over the entire data to obtain a map, the so-called feature map, which stores the characteristics of the input data. Different filters are trained in the training set, such that the features are found in the most efficient way possible.

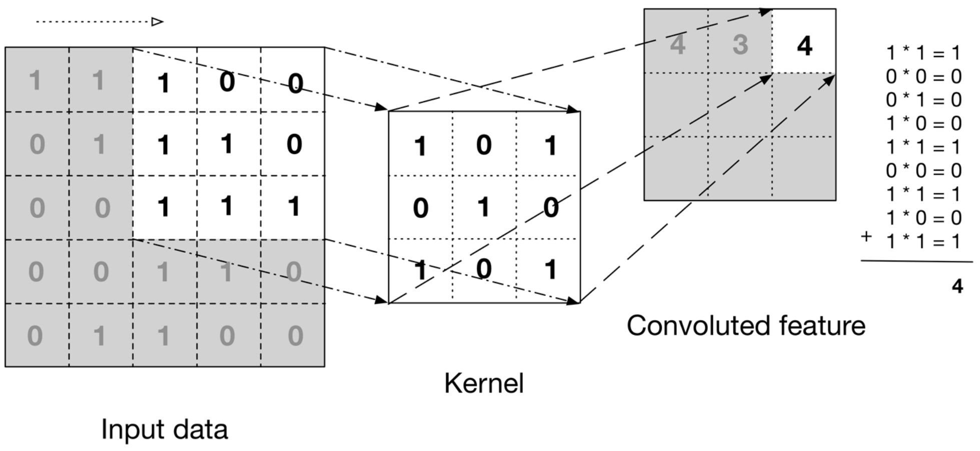


Figure 4: Example of a convolutional process. The input data resembles the embedded tweets of this research. Image taken from the book 'Deep Learning' by Gibson and Patterson (2017).

Next, the max pooling layer calculates the maximum value of each patch in the feature map, where a patch is a submatrix of the feature map. This pooling highlights the most prevalent features and thus reduces, or down samples, the number of features. Figure 5 shows an example of the max pooling operation.

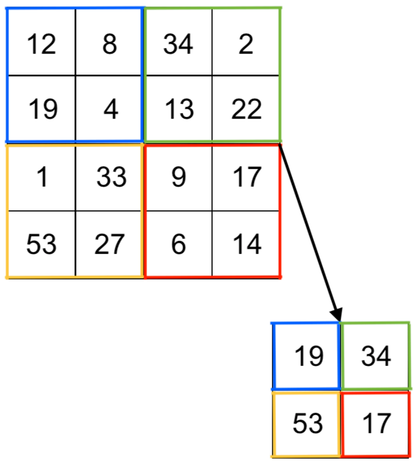


Figure 5: Example of a max pooling layer. The size of the data is reduced by only keeping the maximum feature values in each patch. The patches are indicated by the different colors.

Then, the features go through the fully-connected (dense) layers that interpret the different features. Finally, the sigmoid activation function in the last dense layer outputs a value between 0 and 1, representing the probability that a tweet is marked as correct (0), or incorrect (1).

For running the model, I use a validation split of 15%, meaning that 15% of the training set is used to validate the model that is trained on the other 85% of the training set. Moreover, I set the number of epochs equal to 12 to allow the model to train itself well enough, while also preventing overfitting from happening.

The results of the different classifiers are compared on the basis of a Receiver Operating Characteristic (ROC) curve and other performance measures. The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The TPR and FPR are also referred to as hit rate and false alarm rate, respectively. The other performance measures are in line with Van Der Zee et al., i.e. it includes the accuracy and the Area Under the Curve (AUC) measure, which shows how well the model performs relative to random guessing. A ranking of the classification methods then follows from the measures. However, the time complexity of the model and the application for which the model is meant matter too in determining the ranking.

5 Results

The means of the LIWC and Twitter-specific variables are compared on the basis of a MANOVA. Due to the large number of variables that are significant, the numerical results of the MANOVA are available in Table 10 in Appendix B. In addition to the p-value, the table reports the F-statistic too, as well as the Bayes factor. This factor follows from a Bayesian t-test and indicates the strength of the hypothesis that a variable in factually correct tweets is significantly different from the same variable in the factually incorrect tweets. The larger this number is, the stronger the evidence of a significant difference. Moreover, the table includes the effect size by means of the Cohen's d, accompanied by the confidence interval.

Of the 106 variables included in the analysis, 73 (68.9%) are significant at the 5% level. Of these, 64 (60.4%) are significant at the 1% level and 54 variables (50.9%) are even significant at the 0.1% level. The fact that more than half of the variables are highly significant indicates that there are clearly linguistic differences between factually correct and incorrect tweets. Both the retweet count and favorite count are significant too. As expected, people interact more with tweets that are deemed incorrect: the number of retweets and likes for these tweets is larger. No category of variables can be pointed out as being mainly insignificant. However, in the smaller data set of Van Der Zee et al. (forthcoming), the type of variables that turned out to be insignificant were mainly the punctuation variables, topical variables and pronouns (except for third-person plural). Those that were significant were typically variables that concerned emotions, word types (adverbs, auxiliary verbs, etc.) or Twitter-specific variables. Table 5 in Appendix A contains the MANOVA results for the smaller data set as in Van Der Zee et al.

Table 2 shows the metrics of the model selection by the stepwise-logistic regression (forward and backward). These metrics include the number of variables selected in the model, the log-likelihood, the AIC value and the area under the curve (AUC). For comparison purposes, the results for the model with all 5% variables, the model with all 1% variables and the model with the variables selected by a least absolute shrinkage and selection operator (LASSO) are displayed too. Most noticeably, the number of variables is substantially smaller for the forward and backward models compared to the others, though the log-likelihood, AIC and AUC do not differ a lot. It indicates that the stepwise logistic regression is able to make the model more parsimonious, while maintaining the quality of the model.

Table 2: *Model selection results, training set*

Model	Forward	Backward	Sig. 5%	Sig. 1%	LASSO
Number of variables	40	36	73	64	57
Log-likelihood	-2115.05	-2116.76	-2106.47	-2114.93	-2108.50
AIC	4312.10	4307.53	4360.93	4359.86	4333.00
AUC	0.804	0.804	0.807	0.804	0.807

The 40 variables that the forward stepwise-logistic model selected and their marginal effects are provided in Table 3, along with the stand error, the z-score and the p-value. It is remarkable that 21 of the 54 variables significant at 0.01% are excluded from the model, though there are seven variables included that are ‘only’ significant at the 5% significance level in the MANOVA. This shows that the AIC minimization approach takes into account multicollinearity: the selected variables are as little correlated with each other as possible. Such a procedure results in a parsimonious model, though attaining high prediction power. Nearly all types of words or symbols in Table 3 have little effect on the classification, except for At, relig, Crisis and Colon. For instance, the occurence of 1% more at signs (@) in a tweet decreases the likelihood of a false tweet by 4%.

Table 3: *Marginal effects of the variables selected by the forward stepwise logistic model selection on the likelihood of a tweet being incorrect*

LIWC variable	Variable name	Marginal Effect	Std. Err.	z-score	p-value
WC	Word Count	.004	.001	6.791	<.001
Retweet_count	Retweets	.000	.000	7.975	<.001
Tone	Emotional Tone	-.001	.000	-3.294	.001
At	@	-.040	.005	-7.357	<.001
compare	Comparison Words	.012	.002	6.887	<.001
relig	Religion	-.040	.010	-4.136	<.001
Crisis	Crisis period	.086	.017	5.171	<.001
money	Money	.009	.002	4.993	<.001

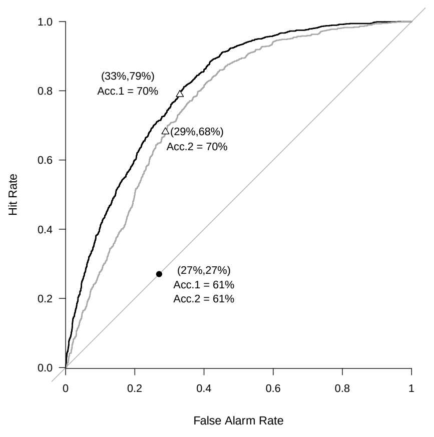
negate	Negation	.005	.002	2.032	.042
focusfuture	Future orientation	-.010	.002	-4.074	<.001
Colon	Colons	-.042	.012	-3.642	<.001
ppron	Personal pronouns	-.000	.002	-.142	.887
cogproc	Cognitive processes	.002	.001	1.553	.120
female	Female references	.011	.003	4.206	<.001
space	Space	.003	.001	2.787	.005
they	Third-person plural pronouns	.007	.003	2.610	.009
discrep	Discrepancy	-.009	.003	-3.473	.001
number	Numbers	.005	.002	2.887	.004
verb.	Common verbs	.004	.001	3.355	.001
time	Time	-.003	.001	-2.020	.043
risk	Risk focus	.004	.003	1.337	.181
Parenth	Parentheses (pairs)	.005	.002	2.135	.033
adverb	Adverbs	.005	.001	3.459	.001
Exclam	Exclamation marks	-.005	.001	-3.386	.001
inhib	Inhibition	.005	.002	2.429	.015
bio	Biology	-.007	.004	-1.920	.055
Quote	Quotation marks	.004	.002	1.764	.078
negemo	Negative emotions	.004	.002	2.065	.039
Dic	Dictionary words	-.001	.001	-1.012	.312
article	Articles	.004	.001	2.604	.009
Hashtag	#	-.012	.003	-3.507	<.001
OtherP	Other punctuation	.009	.003	3.152	.002
tentat	Tentative	.005	.002	1.848	.065
WPS	Average sentence length	-.001	.001	-1.584	.113
reward	Reward focus	-.003	.002	-1.839	.066
i_	First-person singular pronouns	-.013	.004	-3.289	.001
Clout	Clout	-.001	0.000	-2.826	.005
Self	Total first person	.004	.002	2.118	.034
function.	Total function words	-.002	.001	-1.577	.115

focuspast	Past orientation	.002	.002	1.442	.149

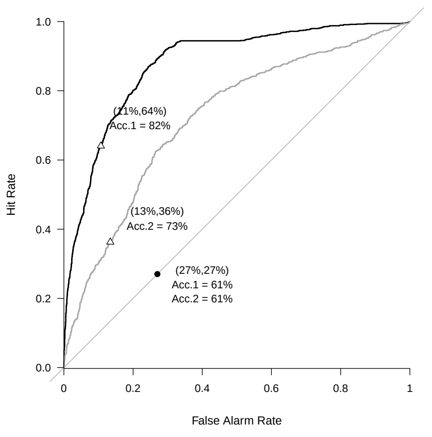
Note. Marginal effects obtained from the training set (N = 4599). They are expressed as a percentage of the total number of words in a tweet, except the retweets (Retweet_count) and the word count (WC). These are expressed as the total number of retweets and words, respectively.

For all other models too, I use the same variables as input variables to train the model. This has some minor consequences for the performance of the other methods, but makes the comparison fairer. The reason behind this is that the stepwise model selection is optimized for a logistic regression. Therefore, it is not guaranteed that the variable selection is also optimal for the other methods, although it is reasonable to assume it fits well. Since the models are more parsimonious this way, they are less likely to become overfitted and will suffer less from collinearity. However, perhaps some variables that might contain valuable information for the other models are left out, resulting in a marginally lower accuracy.

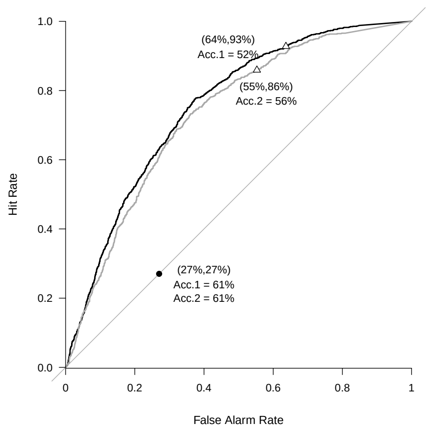
All other models are trained on the same training set (November 2017 - mid July 2019) as in the logistic case. The performance of the obtained logistic model, SVM, NB, kNN, rPart, rf, XGBoost and CNN are tested on the test set (mid July 2019 - April 2020). In all cases, two different cut-offs are used in the classification process: the standard cut-off of 50% and a cut-off of 27.03% that equals the percentage of incorrect tweets in the training set. The latter cut-off is lower and therefore the predictions will be more optimistic: the model will be more inclined to label a tweet as containing false information. Figure 6 displays the ROC curves of the different models, where the 45 degree line represents random guessing. The larger the area under the curve is, the better the predictor is. The grey curve represents the performance of the model on the test set, while the black curve represents the performance on the training set itself.



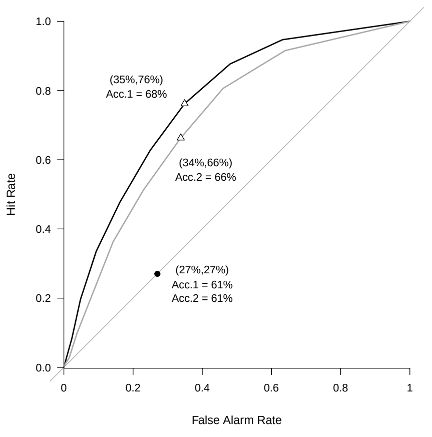
(a) Logistic regression



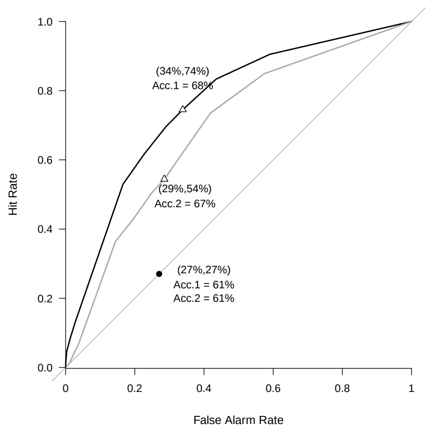
(b) Support Vector Machine



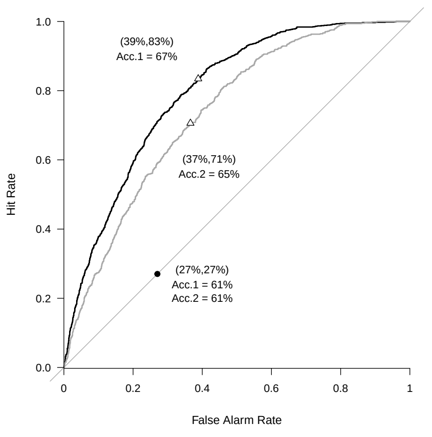
(c) Naïve Bayes



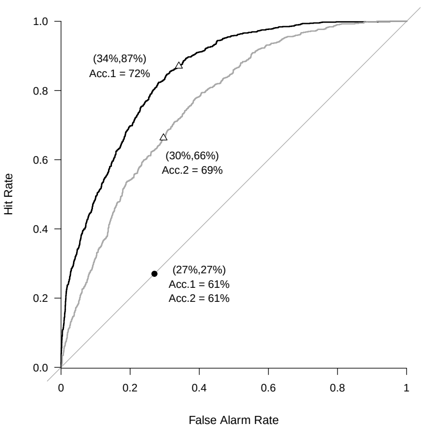
(d) k-Nearest Neighbors



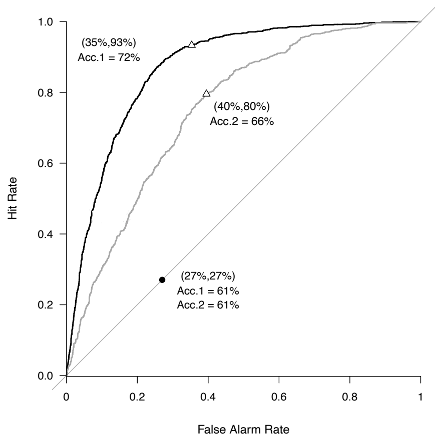
(e) Recursive Partitioning



(f) Random Forest



(g) XGBoost



(h) Convolutional Neural Network

Figure 6: Hit rates plotted against the false alarm rates (ROC curves) for the eight different classification methods. The in-sample predictions are in black (training set) and the out-of-sample predictions are in grey (test set). The actual coordinates of the points are given by the values in the brackets, with the white triangles corresponding to the prior as cut-off. The black dot corresponds to the coordinate when the prior is used as a random guess.

Most of the curves in Figure 6 follow a similar pattern. Especially the grey curves are comparable among all models. Only the performance of the Naïve Bayes is evidently the worst of all models in terms of accuracy. It is the sole method that does not outperform random guessing, which uses the prior probability of 27.03% to predict tweets as factually incorrect. Random guessing attains an accuracy of 61% on both the training set and the test set, whereas the accuracy of Naïve Bayes does not exceed 56%.

Regarding the remaining methods, no curve particularly stands out. It can be observed that the performance is quite similar on the training set and on the test test for the benchmark, the classical methods (except SVM) and the decision-tree based methods. This implies the consistency of the tweeting behaviour over the sample period and indicates that the model is not overfitted. However, the area under the training curve of SVM and Convolutional Neural Network is considerably larger than the area under the test curve for these methods. This implies that they perform especially well on data that shows substantial similarities, or that the models might be overfitted, which reduces the performance on out-of-sample observations. To inspect the possible overfitting, Appendix C repeats the analysis using half the number of variables. Generally, though, it is normal to see a better performance on the training set than on the test set. Overall, based on the graphs in Figure 6, the tree-based methods behave slightly better than the classical methods and the deep neural network, but do not necessarily perform better than the benchmark.

Table 4 contains some performance metrics corresponding to the grey ROC curves (test set) in order to compare the performances more formally. The full table, including the precision, recall, confidence intervals and the metrics of the black ROC curves (training set) is available in Table 11 in Appendix B. The bold values in the Accuracy and AUC column in Table 4 indicate the highest value among the models.

Table 4: *Selected performance metrics*

Cut-off	Accuracy	Hit rate	False alarm	AUC	Cut-off	Accuracy	Hit rate	False alarm	AUC
Logistic regression					Support Vector Machine				
50%	73.35%	27.70%	10.09%	0.757	50%	74.72%	12.87%	2.84%	0.721
27.03%	70.35%	68.14%	28.84%		27.03%	**73.19%**	36.27%	13.42%	
Naïve Bayes					k-Nearest Neighbors				
50%	56.72%	85.05%	53.56%	0.731	50%	72.99%	19.61%	7.64%	0.717
27.03%	55.68%	85.91%	55.29%		27.03%	66.24%	66.30%	33.78%	
Recursive Partitioning					Random Forest				
50%	72.50%	36.40%	14.40%	0.691	50%	74.17%	18.50%	5.64%	0.736
27.03%	66.93%	54.41%	28.53%		27.03%	65.33%	70.59%	36.58%	
XGBoost					Convolutional Neural Network				
50%	**74.69%**	19.36%	5.24%	**0.759**	50%	74.36%	19.0%	5.56%	0.751
27.03%	69.28%	66.30%	29.64%		27.03%	65.56%	79.53%	39.51%	

Note. The bold values in the Accuracy and AUC column denote the largest value among all models, which is done for both cut-offs.

For the cut-off of 50%, XGBoost attains the highest overall accuracy, nearing almost 75%. When a cut-off of 27.03% is applied, the Support Vector Machine has the highest accuracy. Besides, XGBoost also has the largest area under the curve (0.759), closely followed by the logistic regression (0.757). Recursive partitioning is the only method that has a value for AUC of less than 0.7, the others range from 0.715 to 0.759. Though this shows that the performance of the different models is comparable, the classical methods have relatively the lowest predictive power in terms of the AUC (only rPart has lower power).

Moreover, the table suggests a trade-off between accuracy and hit rate/false alarm rate: the cut-off of 50% results in an at least as high accuracy for all models compared to the 27.03% cut-off, but is accompanied by a lower hit rate and lower false alarm rate. In other words, the higher cut-off makes the model more conservative: the threshold to classify tweets as containing false information is higher, reducing the total number of tweets classified as incorrect. Accordingly, both the hit rate (true positive rate) and the false alarm rate (false positive rate) are lower. The choice of the cut-off is therefore crucial and should be set according to the intended use of the model. For instance, a more optimistic approach and therefore a lower cut-off is desired for a preliminary label on tweets.

Contrary to the other models, a major disadvantage of the Convolutional Neural Network is the inability to gain insights in the classification process or in the importance of certain aspects of the data. With the other models, the user is capable of ranking the importance of the variables. As an example, the variable importance of the logistic regression and the classical methods are displayed in Figure 7 and 8, respectively, where the importance of the variables is identical for the three classical methods. Figures 11 and 12 in Appendix B contain the variable importance graphs for the remaining models and the tree constructed by the Recursive Partitioning algorithm.

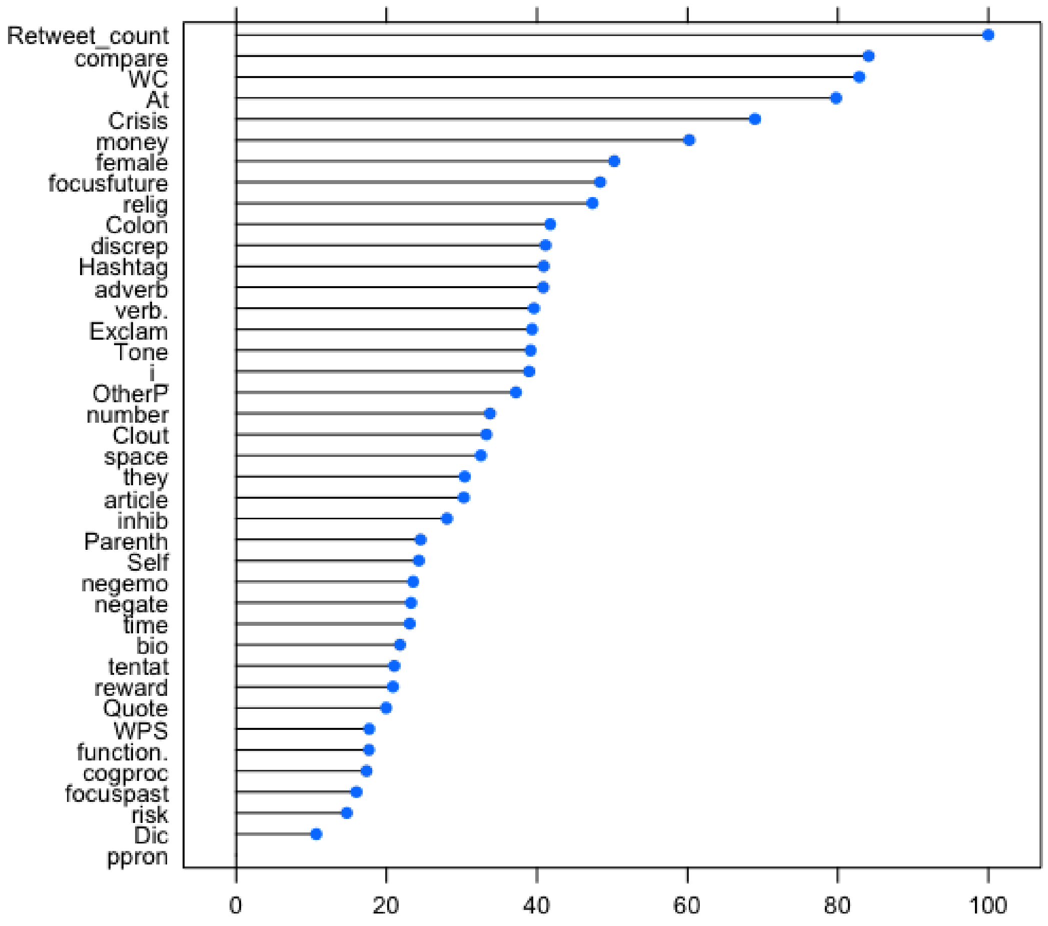


Figure 7: Variable importance in the benchmark. The importance is expressed as a percentage of the most important variable Retweet count.

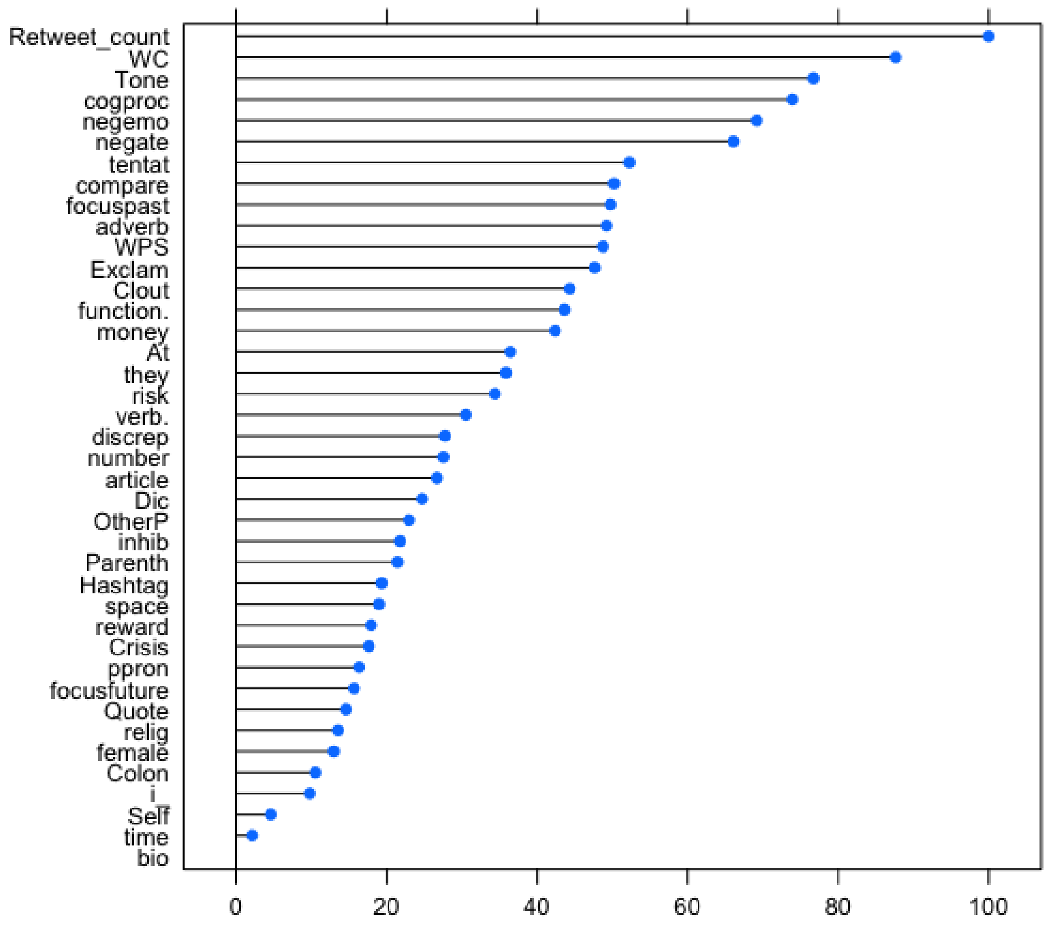


Figure 8: Variable importance in the classical models (SVM, kNN, NB). The importance is expressed as a percentage of the most important variable Retweet count.

The number of retweets and the word count per tweet (WC) are two of the most important variables in determining the correctness of a tweet. In the logistic regression, negative emotions (negemo), tone (Tone) and negations (negate) are relatively less influential. Besides, the personal pronouns (ppron) contribute nothing in Figure 7, whereas bio (biology) contributes nothing in Figure 8. Such insights support researchers in studying the behavior of individuals. It helps them understand which characteristics of written text are most helpful in the identification of false information. Even if the accuracy might not be the best yet, these graphs provide a foundation to build a better model.

Based on the results, an explicit ranking is not feasible due to the relatively small differences. However, a distinction can be made between best, mediocre and worst performers. As a simple heuristic, I base the decision on the value of the AUC, since this value is independent of the chosen cut-off. Doing so results in the following ranking. The best performers are XGBoost, the logistic regression and CNN, the mediocre performers are rf, NB, kNN and SVM. Then, the worst performer is rPart. In other words, the deep neural network and one tree-based method perform at least as well as the benchmark. All classical methods are mediocre performers and the three decision-tree methods all fall in another performance category.

However, the intended application plays a role too. For instance, if the goal is to place a preliminary label on tweets, knowing that the tweet will eventually be fact-checked, one is better off using a more optimistic predictor. A more optimistic model will put a flag on a higher percentage of tweets. Even though this results in more false positive cases, it also results in more true positive cases. If the preliminary label later turns out to be unjustified, it can be removed. On the other hand, if no label was placed, when it in hindsight should have been placed, the false information may already have caused irreversible damage. Figure 9 plots the efficiency in terms of implementation versus the predictive optimism of the different models schematically. The hit rate of the Naïve Bayes is the highest among all methods for both cut-offs, making it the most optimistic approach. So, if the intended use is in line with the description above, NB might actually be one of the better options despite it's low accuracy.

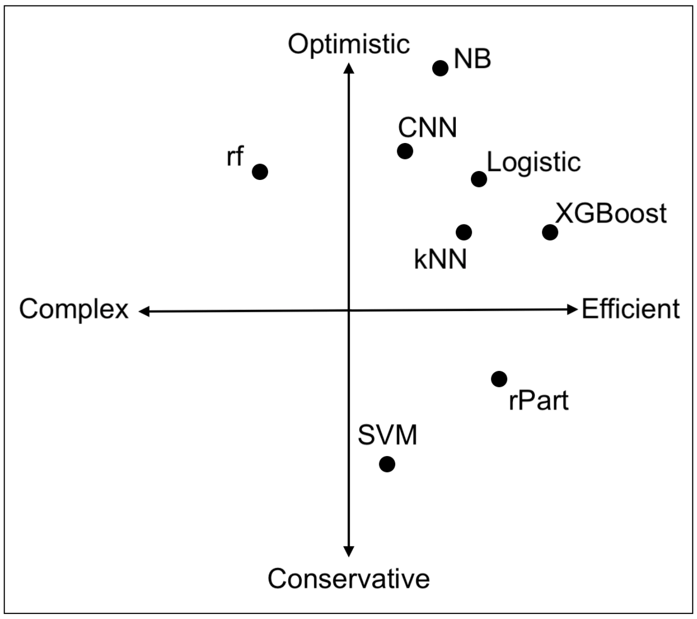


Figure 9: The computational complexity versus the optimism of all models.

A different type of model is desired when the classification of the tweet is permanent, i.e., when the tweet will not be manually fact-checked. In this situation, the model should be certain that a tweet contains false information, otherwise it should not be flagged. Therefore, a more conservative predictor should be used. Since the false alarm rate of the Support Vector Machine is only 3.42%, while maintaining an accuracy of 74.36%, this model is suited for more conservative applications. Figure 9 clearly shows that SVM is the most conservative option.

In terms of the computational complexity, most methods are equally efficient, with some exceptions. The classical algorithms are neither extremely fast nor inefficient and the same holds for the benchmark and the deep neural network. There is, again, a remarkable difference between the decision tree methods. Random Forests are the most complex due to the large number of trees (3000 in this research) that have to be constructed simultaneously. Recursive partitioning is comparable to the benchmark and the XGBoost is the most efficient.
Comparing the obtained results with the analysis on same, smaller data set as in Van Der Zee et al. (forhcoming), the performance is overall consistent. The ROC curves in Figure 10 of Appendix A do not deviate evidently from the ROC curves of the larger data set in Figure 6. Moreover, the AUC values in Table 8 in Appendix A fluctuate around the same value of approximately 0.74. However, the accuracy and the area under the curve of the Convolutional Neural Network is better in the small sample. Consequently, the ranking of best, mediocre and worst performs resembles the ranking of the large data set, with one difference. Namely, the best performers are the logistic regression and CNN, the mediocre performers are kNN, XGBoost, rf and SVM and the worst performers are NB and rPart. Only XGBoost and CNN have swapped places in this ranking. The consistency of the results is thus maintained using a larger sample.

6 Discussion

Out of the 7 new methods that were compared to the benchmark model, the logistic regression as in Van Der Zee et al. (forthcoming), no particular group of methods outperformed it in every respect. The accuracy of some methods exceeded the accuracy of the logistic regression, and only the area under the ROC curve of XGBoost exceeded the one of benchmark. XGBoost is the only method that could be considered to outperform the logistic regression. The performance of the classical methods, the deep neural network and one decision-tree based method (rf) was comparable to the performance of the logit model. The rPart classifier was the only method that did not perform well in terms of predictive power. These findings either mean that nearly all methods are more or less identical in their classification process, or that it has to do with classification of tweets in particular. The former case is highly unlikely, given that previous literature favors some methods over others. For instance, the research by Fernández-Delgado et al. (2014) that compares different classification methods proves that Random Forests outperform most methods, followed by Support Vector Machines. Besides, the newer model XGBoost has drawn a lot of attention by researchers due to it's outstanding performance in different Kaggle competitions, which is a renowned data science competition. A more plausible explanation for the comparable performance of all methods lies in the objective of detecting misleading tweets. In most text classification cases, the text under consideration conveys a deceptive/truthful message altogether. However, tweets are only short messages and are often classified by The Washington Post as incorrect only because of factually incorrect information (i.e., an incorrect year mentioned). Most models are not capable of identifying these cases, no matter the complexity or strength of the method. Therefore, there seems to be a certain maximum accuracy that can be attained in detecting incorrect tweets.

The skewed distribution of factually correct and incorrect statements in the data set makes it difficult to compare the findings with related literature. In most deception detection studies, the distribution of deceptive and truthful statements is 50/50 (Levine, 2018). On the other hand, it is not realistic to observe that 50% of all statements are deceptive in the real world. Therefore, the results based on a data set consisting of approximately 70% truthful tweets and 30% deceptive tweets reflect the reality, but make comparisons to previous research unfair. It would be interesting to study the performance of the models during the 2020 elections. In this period, Donald Trump likely tweeted relatively more deceptive statements than in the periods studied in this research, making a comparison to the literature fairer.

A limitation of this study is that the fact-checking of the tweets is done by a third party, because labelling statements as either correct or incorrect involves a degree of subjectivity. Additionally, the Washington Post did not analyse all tweets. For example, personal opinions cannot be verified and are therefore all classified as correct. This selective analysis of tweets might partly explain the importance of the number of retweets in the classification process. I find that the number of retweets for factually incorrect tweets is significantly larger than for factually correct tweets. However, reversed causality might be the case here: perhaps that the Washington Post was more likely to verify tweets that were often retweeted, which explains the difference in the number of retweets. One could therefore question the validity of this variable.

Throughout this paper, I have only considered the case in which tweets are marked as either containing false information or not (e.g., binary classification). In future research, one could implement a multi-class classification method that displays a warning message in accordance with the degree of certainty of the classifier. For instance, if the model expects a tweet to contain false information with a probability between 80 and 100%, a message stating that the tweet is “very likely” to contain false information should show. This enhances the usefulness in practice and makes the output less rigorous.

Besides, a sensitivity analysis by Zhang and Wallace (2015) shows that the Convolutional Neural Network is sensitive to minimal changes in the model architecture. Because this method requires extensive knowledge of the network to be able to specify the model architecture correctly, future research may find a specification that outperforms the one in this study.

7 Conclusion

In this research, I examined the performance of the logistic regression, functioning as benchmark, relative to seven other classification methods that are prevalent in the literature. These include three classical methods, consisting of Naïve Bayes, Support Vector Machine and k-Nearest Neighbors. Additionally, three decision-tree based methods are examined, consisting of Recursive Partitioning, Random Forests and XGBoost. Lastly, a deep neural network (Convolutional Neural Network) is included in the analysis. The aim was to improve the classification results applied in Van Der Zee et al. (forthcoming) and make the results more robust by using a larger data sample: 2.5 years of tweets by Donald Trump, compared to 6 months in Van Der Zee et al. Following from the literature, I expected the best performance to come from Random Forests, XGBoost or the Convolutional Network due to these being the most state-of-the-art models.

Out of the 106 linguistic variables that possibly explain differences between truthful and deceptive tweets, 73 were significant, proving the linguistic difference between factually correct and incorrect tweets. Instead of using all 73 significant variables, an AIC minimization process selected 40 variables to include in the analysis, making the models more parsimonious. Overall, the most important variables consisted of emotional variables (i.e., the emotional tone of tweets or inclusion of negative emotions in tweets) and Twitter specific variables (i.e., the number of retweets and the presence of the At sign in a tweet). Nonetheless, the model covered various other linguistic aspects too (such as topical variables, e.g. words about religion or money, or functional words, e.g. personal pronouns and verbs).

The main finding of this research is that no group of methods outperforms the benchmark in every respect. In terms of accuracy and predictive power, XGBoost is the most promising, outperforming the logistic regression in both respects. XGBoost attains an out-of-sample accuracy of 74.69%.

The two other decision-tree based methods, the classical methods and the deep neural network do not exceed the results of the benchmark or XGBoost. For instance, all three classical methods perform worse than the benchmark, possibly due to violations of critical assumptions (Naïve Bayes) or due to the method not being optimized for text classification (SVM and kNN).

Moreover, the single decision tree created by the Recursive Partitioning algorithm is a too simplistic method to accurately identify correct and incorrect tweets, as it has the least predictive power among all models. Random Forests performed decently, but they proved that plain decision-trees are not well suited for the purpose of classifying tweets, as they did not beat the benchmark. The ‘smarter’ method XGBoost builds on the information gained from previously constructed trees, explaining why this algorithm works much better than the other decision-tree based models.

Furthermore, the Convolutional Neural Network is equal to the benchmark, performance-wise. The current specification is not able to outperform XGBoost, but still does a good job. Some parameter tuning or model tweaks could enhance the neural network, potentially making it the best classifier. However, it lacks interpretation and is therefore not the favorite model, as it is desired to understand which linguistic characteristics contribute to a tweet being correct or incorrect.

Lastly, this study shows that the intended use of the model plays an important role too in determining which classifier to use. Naïve Bayes is more suited for optimistic applications due to it’s relatively high hit rate. For example, this method could be implemented in Twitter to place preliminary tags on tweets, since it identifies the largest percentage of incorrect tweets. On the other hand, Support Vector Machines have a relatively low false alarm rate, making this the preferred method for conservative applications, such as for permanent tags that will not be fact-checked. It is then desired to only mark tweets as incorrect if the model is confident to have identified an incorrect tweet. As there is no unambiguously answer to which model is the best, further research is required to make the classification of tweets more feasible in practice.

References

Allcott, H., & Gentzkow, M. (2017). Social media and fake news in the 2016 election. *Journal of economic perspectives*, *31*(2), 211–36.

Anil Kumar, D., & Ravi, V. (2008). Predicting credit card customer churn in banks using data mining. *International Journal of Data Analysis Techniques and Strategies*, *1*(1), 4–28.

Bachenko, J., Fitzpatrick, E., & Schonwetter, M. (2008). Verification and implementation of language-based deception indicators in civil and criminal narratives. *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, 41–48.

Barbaro, M. (2015). Pithy, mean and powerful: How donald trump mastered twitter for 2016. *The New York Times*, *5*.

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, *57*(1), 289–300.

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

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

De Silva, G. L., & Hull, J. J. (1994). Proper noun detection in document images. *Pattern Recognition*, *27*(2), 311–320.

Di Domenico, G., Sit, J., Ishizaka, A., & Nunan, D. (2021). Fake news, social media and marketing: A systematic review. *Journal of Business Research*, *124*, 329–341.

Duttagupta, R., & Cashin, P. (2008). The anatomy of banking crises. *IMF Working Papers*, *2008*(093).

Evans, A. M., & Krueger, J. I. (2009). The psychology (and economics) of trust. *Social and Personality Psychology Compass*, *3*(6), 1003–1017.

Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *The journal of machine learning research*, *15*(1), 3133–3181.

Gelfert, A. (2018). Fake news: A definition. *Informal Logic*, *38*(1), 84–117.

Gunn, S. R. et al. (1998). Support vector machines for classification and regression. *ISIS technical report*, *14*(1), 5–16.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). 10. boosting and additive trees. 337–387.

Hauch, V., Blandón-Gitlin, I., Masip, J., & Sporer, S. L. (2015). Are computers effective lie detectors? a meta-analysis of linguistic cues to deception. *Personality and social psychology Review*, *19*(4), 307–342.

James, K. E., White, R. F., & Kraemer, H. C. (2005). Repeated split sample validation to assess logistic regression and recursive partitioning: An application to the prediction of cognitive impairment. *Statistics in medicine*, *24*(19), 3019–3035.

Kessler, G., Rizzo, S., & Usero, A. (2021). *In four years, President Trump made 30,573 false or misleading claims*. The Washington Post.

Kim. (n.d.). Convolutional Neural Networks for Sentence Classification.

Kim, Han, K.-S., Rim, H.-C., & Myaeng, S. H. (2006). Some effective techniques for naive bayes text classification. *IEEE transactions on knowledge and data engineering*, *18*(11), 1457–1466.

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, *160*(1), 3–24.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. *Information*, *10*(4), 150.

Lee, T. (2019). The global rise of "fake news" and the threat to democratic elections in the usa. *Public Administration and Policy.*

Levine, T. R. (2018). Ecological validity and deception detection research design. *Communication Methods and Measures*, *12*(1), 45–54.

Mihalcea, R., & Strapparava, C. (2009). The lie detector: Explorations in the automatic recognition of deceptive language. *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, 309–312.

Morris, M. R., Counts, S., Roseway, A., Hoff, A., & Schwarz, J. (2012). Tweeting is believing? understanding microblog credibility perceptions. *Proceedings of the ACM 2012 conference on computer supported cooperative work*, 441–450.

Mulak, P., & Talhar, N. (2015). Analysis of distance measures using k-nearest neighbor algorithm on kdd dataset. *International Journal of Science and Research*, *4*(7), 2101–2104.

Ott, M., Choi, Y., Cardie, C., & Hancock, J. T. (2011). Finding deceptive opinion spam by any stretch of the imagination. *arXiv preprint arXiv:1107.4557.*

Patterson, J., & Gibson, A. (2017). *Deep learning: A practitioner's approach.* " O'Reilly Media, Inc."

Pennebaker, J. W., Boyd, R. L., Jordan, K., & Blackburn, K. (2015). *The development and psychometric properties of liwc2015* (tech. rep.).

Piryonesi, S. M., & El-Diraby, T. E. (2020). Data analytics in asset management: Cost-effective prediction of the pavement condition index. *Journal of Infrastructure Systems*, *26*(1), 04019036.

Rubin, V. L. (2017). Deception detection and rumor debunking for social media. *The SAGE Handbook of Social Media Research Methods*, 342–364.

Scavino, D. (2017). @DanScavino. *Twitter. https://twitter.com/DanScavino/status/846918912793083904.*

Suguna, N., & Thanushkodi, K. (2010). A k-nearest neighbor classification using genetic algorithm. *International Journal of Computer Science Issues*, *7*(2), 18–21.

Van Der Zee, S., Poppe, R., Havrileck, A., & Baillon, A. (forthcoming). A personal model of trumpery: Deception detection in a real-world high-stakes setting. *Psychological Science.*

Vapnik, V. (1995). *The nature of statistical learning theory.* Springer-Verlag.

Vrij, A. (2000). *Detecting lies and deceit: The psychology of lying and implications for professional practice.* John Wiley & Sons.

Waszak, P. M., Kasprzycka-Waszak, W., & Kubanek, A. (2018). The spread of medical fake news in social media–the pilot quantitative study. *Health policy and technology*, *7*(2), 115–118.

Zhang, Y., & Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820.*

Appendix A. Full analysis on smaller data set.

Table 5: *MANOVA results for the LIWC variables, training set (Feb - April 2018) as in Van Der Zee et al. (forthcoming)*

LIWC name	Variable name	Mean if correct	Mean if incorrect	F stat.	*p*-value	Sig.	Bayes Factor	Cohen's d	CI
achieve	Achievement	2.23	1.59	4.80	0.074		1.120	0.22	[0.02,0.42]
adj	Adjectives	6.45	6.16	0.21	0.780		0.120	0.05	[-0.15,0.24]
adverb	Adverbs	3.65	5.36	17.71	0.000	***	506.060	-0.42	[-0.62,-0.22]
affect	Emotions	9.76	7.79	7.12	0.027	*	3.390	0.27	[0.07,0.47]
affiliation	Affiliation	3.68	2.30	9.07	0.013	*	8.560	0.30	[0.1,0.5]
AllPunc	All punctuation	17.60	16.11	1.52	0.360		0.230	0.12	[-0.07,0.32]
Analytic	Analytic thinking	75.37	69.18	5.60	0.055		1.640	0.24	[0.04,0.44]
anger	Anger	0.39	0.77	9.66	0.011	*	11.330	-0.31	[-0.51,-0.11]
anx	Anxiety	0.12	0.25	3.28	0.148		0.540	-0.18	[-0.38,0.02]
Apostro	Apostrophes	0.71	1.26	7.08	0.027	*	3.320	-0.27	[-0.47,-0.07]
article	Articles	6.43	6.65	0.27	0.761		0.130	-0.05	[-0.25,0.15]
assent	Assent	0.10	0.04	0.36	0.730		0.130	0.06	[-0.14,0.26]
At	@	1.42	0.11	16.67	0.001	**	310.450	0.41	[0.21,0.61]
Authentic	Authentic	32.69	29.90	0.76	0.575		0.160	0.09	[-0.11,0.29]
auxverb	Auxiliary verbs	7.59	9.24	9.45	0.012	*	10.280	-0.31	[-0.51,-0.11]
bio	Biology	0.86	0.56	2.38	0.228		0.350	0.15	[-0.04,0.35]
body	Body	0.13	0.07	0.59	0.627		0.150	0.08	[-0.12,0.27]

cause	Causations	1.07	1.81	12.25	0.004	**	38.750	-0.35	[-0.55,-0.15]
certain	Certainty	1.80	2.54	6.20	0.040	*	2.190	-0.25	[-0.45,-0.05]
Clout	Clout	70.26	61.17	14.22	0.001	**	98.060	0.38	[0.18,0.58]
cogproc	Cognitive processes	7.07	10.59	36.89	0.000	***	>1,000	-0.61	[-0.81,-0.41]
Colon	Colons	0.40	0.07	5.50	0.057		1.570	0.24	[0.04,0.43]
Comma	Commas	0.01	0.03	0.74	0.575		0.160	-0.09	[-0.28,0.11]
compare	Comparison words	1.40	2.37	12.07	0.004	**	35.650	-0.35	[-0.55,-0.15]
conj	Conjunctions	4.76	5.00	0.41	0.706		0.140	-0.06	[-0.26,0.13]
Dash	Dashes	0.65	0.58	0.14	0.831		0.120	0.04	[-0.16,0.23]
death	Death	0.38	0.16	2.29	0.231		0.330	0.15	[-0.05,0.35]
Dic	Dictionary words	79.23	81.27	3.31	0.148		0.550	-0.18	[-0.38,0.01]
differ	Differentiation	1.62	2.63	15.37	0.001	**	168.740	-0.39	[-0.59,-0.19]
discrep	Discrepancy	1.17	1.73	6.40	0.039	*	2.400	-0.25	[-0.45,-0.06]
drives	Drives	13.24	10.93	8.35	0.016	*	6.080	0.29	[0.09,0.49]
excl	Exclusive	1.46	1.98	4.96	0.072		1.200	-0.22	[-0.42,-0.03]
Exclam	Exclamation marks	5.13	2.81	8.28	0.016	*	5.900	0.29	[0.09,0.49]
family	Family	0.23	0.06	4.45	0.086		0.950	0.21	[0.01,0.41]
Favorite_count	Favorites	93144.96	103355.14	8.60	0.015	*	6.860	-0.29	[-0.49,-0.10]
feel	Feeling	0.39	0.16	3.43	0.141		0.580	0.19	[-0.01,0.38]
female	Female references	0.47	0.50	0.01	0.971		0.110	-0.01	[-0.21,0.19]
filler	Fillers	0.01	0.00	0.43	0.696		0.140	0.07	[-0.13,0.26]
focusfuture	Future orientation	1.79	1.22	4.94	0.072		1.190	0.22	[0.03,0.42]

focuspast	Past orientation	2.48	3.59	10.27	0.009	**	15.190	-0.32	[-0.52,-0.12]
focuspresent	Present orientation	8.65	10.10	5.41	0.058		1.500	-0.23	[-0.43,-0.04]
friend	Friends	0.19	0.15	0.28	0.761		0.130	0.05	[-0.14,0.25]
function.	Total function words	43.37	47.58	14.63	0.001	**	119.380	-0.38	[-0.58,-0.19]
Hashtag	#	1.19	0.07	3.81	0.115		0.700	0.20	[0.00,0.39]
health	Health	0.38	0.41	0.03	0.906		0.110	-0.02	[-0.22,0.18]
hear	Hearing	0.32	0.41	0.68	0.596		0.150	-0.08	[-0.28,0.11]
home	Home	0.45	0.26	2.97	0.166		0.470	0.17	[-0.02,0.37]
humans	Humans	0.89	0.83	0.11	0.841		0.120	0.03	[-0.16,0.23]
i_	First-person singular pronouns	1.23	1.13	0.21	0.780		0.120	0.05	[-0.15,0.24]
incl	Inclusive	5.52	5.64	0.07	0.885		0.120	-0.03	[-0.22,0.17]
informal	Informal	0.27	0.23	0.11	0.841		0.120	0.03	[-0.16,0.23]
ingest	Ingestion	0.16	0.07	1.13	0.459		0.190	0.11	[-0.09,0.30]
inhib	Inhibition	0.78	1.19	3.85	0.115		0.710	-0.20	[-0.4,0.0]
insight	Insight	1.02	1.11	0.23	0.780		0.120	-0.05	[-0.25,0.15]
interrog	Interrogatives	0.91	1.34	4.74	0.075		1.090	-0.22	[-0.42,-0.02]
ipron	Impersonal pronouns	2.94	3.20	0.67	0.596		0.150	-0.08	[-0.28,0.12]
leisure	Leisure	0.64	0.32	2.85	0.175		0.440	0.17	[-0.03,0.37]
male.	Male references	0.83	0.77	0.06	0.885		0.110	0.02	[-0.17,0.22]
Metaph	Metaphysical	0.81	0.80	0.00	0.988		0.110	0.00	[-0.2,0.2]
money	Money	1.26	2.16	8.22	0.016	*	5.710	-0.29	[-0.49,-0.09]
motion	Motion	1.64	1.51	0.26	0.761		0.130	0.05	[-0.15,0.25]

negate	Negations	1.05	2.51	47.01	0.000	***	>1,000	-0.69	[-0.89,-0.49]
negemo	Negative emotions	2.25	3.66	10.01	0.010	*	13.410	-0.32	[-0.52,-0.12]
netspeak	Netspeak	0.06	0.11	0.92	0.514		0.170	-0.10	[-0.29,0.10]
nonflu	Nonfluencies	0.08	0.07	0.04	0.895		0.110	0.02	[-0.18,0.22]
number	Numbers	1.58	1.93	1.00	0.492		0.180	-0.10	[-0.3,0.1]
Optim	Optimism	1.04	1.13	0.19	0.789		0.120	-0.04	[-0.24,0.15]
Other	Total third person	1.56	1.93	1.95	0.281		0.280	-0.14	[-0.34,0.06]
OtherP	Other punctuation	3.02	1.17	7.21	0.027	*	3.530	0.27	[0.07,0.47]
Parenth	Parentheses (pairs)	0.62	1.09	3.87	0.115		0.720	-0.20	[-0.4,0.0]
percept	Perceptual processes	1.59	1.31	1.06	0.476		0.190	0.10	[-0.09,0.30]
Period	Periods	6.24	7.99	9.24	0.012	*	9.290	-0.31	[-0.50,-0.11]
posemo	Positive emotions	7.43	4.04	23.65	0.000	***	>1,000	0.49	[0.29,0.69]
power	Power	5.09	5.27	0.14	0.831		0.120	-0.04	[-0.24,0.16]
ppron	Personal pronouns	6.03	5.62	0.55	0.640		0.150	0.07	[-0.12,0.27]
prep	Prepositions	12.70	12.83	0.06	0.885		0.110	-0.02	[-0.22,0.17]
pronoun	Total pronouns	8.96	8.82	0.05	0.886		0.110	0.02	[-0.17,0.22]
QMark	Question marks	0.44	0.44	0.00	0.988		0.110	0.00	[-0.2,0.2]
quant	Quantifiers	1.91	2.31	1.65	0.338		0.250	-0.13	[-0.33,0.07]
Quote	Quotation marks	0.37	0.66	3.22	0.151		0.520	-0.18	[-0.38,0.02]
relativ	Relativity	14.38	13.79	0.46	0.689		0.140	0.07	[-0.13,0.27]
relig	Religion	0.67	0.05	9.35	0.012	*	9.810	0.31	[0.11,0.51]
Retweet_count	Retweets	21150.70	25448.30	19.03	0.000	***	937.850	-0.44	[-0.64,-0.24]

reward	Reward focus	2.92	1.74	8.47	0.016	*	6.450	0.29	[0.09,0.49]
risk	Risk focus	0.87	1.35	4.87	0.073		1.150	-0.22	[-0.42,-0.02]
sad	Sadness	0.49	0.87	4.41	0.087		0.930	-0.21	[-0.41,-0.01]
see	Seeing	0.86	0.73	0.34	0.736		0.130	0.06	[-0.14,0.26]
Self	Total first person	3.19	3.68	1.51	0.360		0.230	-0.12	[-0.32,0.07]
Senses	Sensory and Perceptual Processes	1.18	1.05	0.33	0.736		0.130	0.06	[-0.14,0.25]
sexual	Sexual	0.03	0.00	1.16	0.455		0.190	0.11	[-0.09,0.31]
shehe	Third-person singular pronouns	0.68	0.73	0.06	0.885		0.110	-0.02	[-0.22,0.17]
Sixltr	Six-letter words	22.90	20.35	6.27	0.040	*	2.260	0.25	[0.05,0.45]
social	Social processes	9.37	8.33	2.34	0.229		0.340	0.15	[-0.04,0.35]
space	Space	8.14	8.11	0.00	0.977		0.110	0.01	[-0.19,0.20]
swear	Swear words	0.01	0.01	0.01	0.971		0.110	-0.01	[-0.20,0.19]
tentat	Tentative	1.20	2.16	18.87	0.000	***	872.370	-0.44	[-0.64,-0.24]
they	Third-person plural pronouns	0.73	1.52	16.23	0.001	**	252.350	-0.40	[-0.60,-0.21]
time	Time	5.11	4.34	2.28	0.231		0.330	0.15	[-0.05,0.35]
Tone	Emotional tone	67.52	44.73	33.95	0.000	***	>1,000	0.59	[0.38,0.79]
verb.	Common verbs	13.43	15.70	9.88	0.010	*	12.620	-0.32	[-0.51,-0.12]
WC	Word quantity	31.22	39.56	36.96	0.000	***	>1,000	-0.61	[-0.81,-0.41]
we.	First-person plural pronouns	2.31	1.65	3.11	0.155		0.500	0.18	[-0.02,0.38]
work	Job/Work	5.08	5.12	0.01	0.971		0.110	-0.01	[-0.21,0.19]
WPS	Average sentence length	12.87	14.09	3.18	0.151		0.520	-0.18	[-0.38,0.02]
you	Total second-person pronouns	1.07	0.59	2.43	0.224		0.360	0.16	[-0.04,0.35]

Table 6: *Model selection results, small training set (Feb - April 2018).*

Model	Forward	Backward	Sig. 5%	Sig. 1%	LASSO
Number of variables	12	14	34	16	23
Log-likelihood	-218.44	-216.13	-210.58	-225.88	-212.44
AIC	462.88	462.26	491.15	485.77	472.87
AUC	0.819	0.823	0.834	0.806	0.831

The data set used in this Appendix is similar to the data sets in Study 1 and 2 by Van Der Zee et al. (forthcoming). The training set [test set] consists of 469 [484] observations ranging from February to April 2018 [November 2017 - January 2018]. Table 6 shows that the model selected by the forward stepwise logistic function is more parsimonious for this alternative data set. Overall, less variables are significant compared to the data set used in this research. Table 7 displays the marginal effects of the variables included in the model.

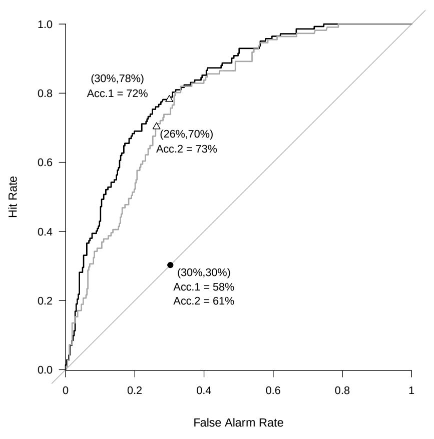
Table 7: *Marginal effects of the variables selected by the forward stepwise logistic model selection.*

LIWC variable	Variable name	Marginal Effect	Std. Err.	z-score	p-value
negate	Negation	.022	.008	2.617	.009
WC	Word Count	.005	.002	3.339	.001
Tone	Emotional Tone	-.002	0.000	-3.699	0.000
At	@	-.050	.019	-2.562	.010
relig	Religion	-.063	.032	-2.010	.044
compare	Comparison Words	.015	.006	2.412	.016
Period	Periods	.007	.003	2.336	.020
tentat	Tentative	.014	.008	1.727	.084
money	Money	.012	.005	2.202	.028
certain	Certainty	.011	.006	1.896	.058
they	Third-person plural	.014	.009	1.549	.121
adverb	Adverbs	.007	.005	1.471	.141

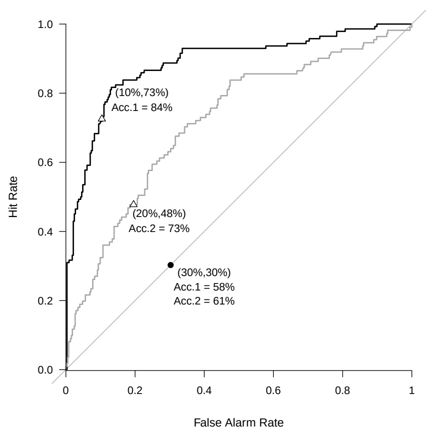
Note. Marginal effects obtained from the training set (N = 469). They are expressed as a percentage of the total number of words in a tweet, except the word count (WC), which is expressed as the total number words in a tweet.

Many of the important variables are included in this smaller model too, such as word count, negate and tone. However, most notably, the highly significant variable Retweet count is excluded from this list, while it was the most important variable for most models in the larger data set. The marginal effects are also larger in absolute terms than was previously observed.

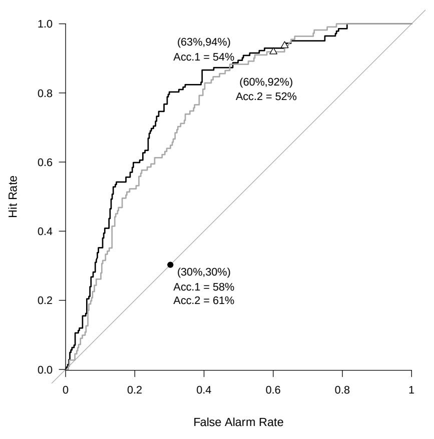
Figure 10 contains the ROC curves of all different models.



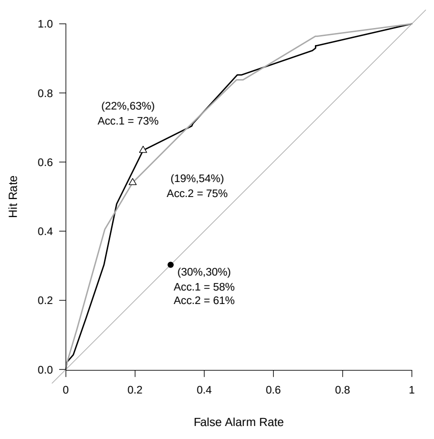
(a) Logistic regression



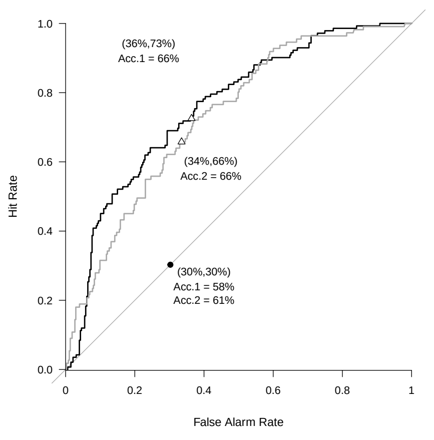
(b) Support Vector Machine



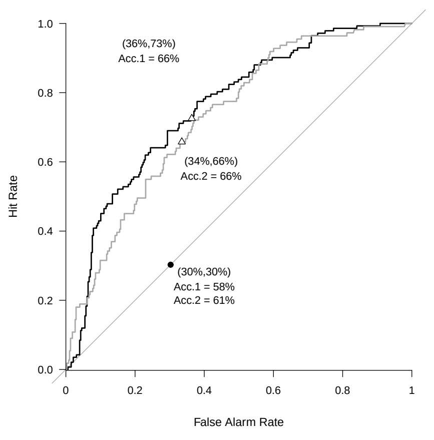
(c) Naïve Bayes



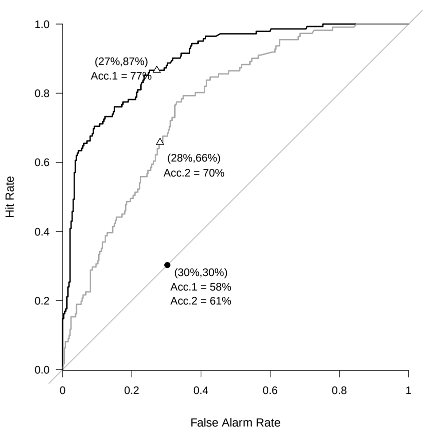
(d) k-Nearest Neighbors



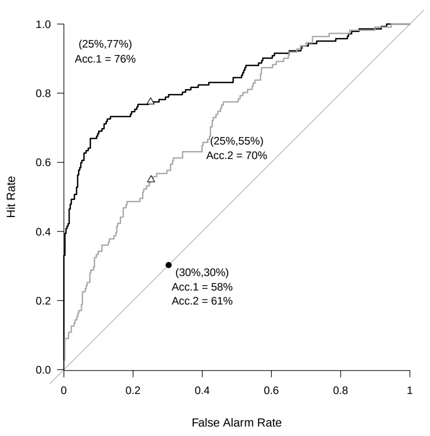
(e) Recursive Partitioning



(f) Random Forest



(g) XGBoost



(h) Convolutional Neural Network

Figure 10: Hit rates plotted against the false alarm rates (ROC curves) for the eight different classification methods. The in-sample predictions are in black (training set, Feb - April 2018) and the out-of-sample predictions are in grey (test set, Nov 2017 - Jan 2018). The actual coordinates of the points are given by the values in the brackets, with the white triangles corresponding to the prior as cut-off. The black dot corresponds to the coordinate when the prior is used as a random guess.

The lines are evidently more 'shocky' than was the case in the larger data set, due to the limited number of observations in the train and test set. Despite this, the curves behave similarly. The logistic regression, though, seems to be the best based on the graphs alone. This is proven by the performance metrics in Tables 8 and 9.

Table 8: *Selected performance metrics*

Cut-off	Accuracy	Hit rate	False alarm	AUC	Cut-off	Accuracy	Hit rate	False alarm	AUC
Logistic regression					Support Vector Machine				
0.5	**0.7748**	0.3694	0.1046	**0.784**	0.5	0.7645	0.2793	0.0912	0.72
0.2703	0.7293	0.7027	0.2626		0.2703	0.7314	0.4775	0.193	
Naïve Bayes					k-Nearest Neighbors				
0.5	0.562	0.8919	0.5362	0.755	0.5	**0.7748**	0.2523	0.0697	0.746
0.2703	0.5186	0.9189	0.6005		0.2703	**0.7459**	0.5405	0.193	
Recursive Partitioning					Random Forest				
0.5	0.7521	0.3153	0.118	0.66	0.5	0.7624	0.2793	0.0938	0.73
0.2703	0.7025	0.5495	0.252		0.2703	0.6632	0.6577	0.3351	
XGBoost					Convolutional Neural Network				

0.5	0.7727	0.2883	0.0831	0.762	0.5	**0.7748**	0.027	0.0027	0.717
0.2703	0.7045	0.6577	0.2815		0.2703	0.7025	0.5495	0.252	

For both cut-offs, k-Nearest Neighbors attains the highest overall accuracy, nearing almost 80% when a cut-off of 50% is applied. The logistic regression and Convolutional Neural Network attain the same accuracy for a 50% cut-off. The logistic regression has the largest area under the curve (0.784). Overall, the results in the tables are comparable to Table 4 and 11. The classical methods have a mediocre performance, though the performance by Naïve Bayes improved. Again, rPart is clearly the worst in terms of the area under the curve. Random Forests are assigned to the mediocre performers again too, and XGBoost belongs to the best performers. Taken together, the ranking is not much different than before.

Table 9: *All performance metrics of the different models on the Van Der Zee et al. (forthcoming) data set.*

Data	Cut-off	Accuracy	Acc. CI	Hit rate	False alarm rate	Precision	F1	AUC	AUC CI
Logistic regression									
train	0.5	0.7719	[0.727, 0.806]	0.5	0.1101	0.6636	0.5703	0.819	[0.781, 0.858]
	0.3028	0.7249	[0.7036, 0.7484]	0.7817	0.2997	0.5311	0.6325		
test	0.5	0.7748	[0.75, 0.7975]	0.3694	0.1046	0.5125	0.4293	0.784	[0.74, 0.827]
	0.3028	0.7293	[0.6963, 0.7541]	0.7027	0.2627	0.4432	0.5436		
Support Vector Machine									
train	0.5	0.8252	[0.791, 0.8679]	0.5493	0.055	0.8125	0.6555	0.882	[0.846, 0.918]
	0.3028	0.8443	[0.806, 0.8785]	0.7183	0.1009	0.7556	0.7365		
test	0.5	0.7645	[0.7438, 0.7955]	0.2793	0.0912	0.4769	0.3523	0.72	[0.665, 0.775]
	0.3028	0.7314	[0.7087, 0.7562]	0.4775	0.193	0.424	0.4492		
Naïve Bayes									

train	0.5	0.5736	[0.5565, 0.5842]	0.9296	0.581	0.4099	0.569	0.786	[0.743, 0.829]
	0.3028	0.5394	[0.5267, 0.5522]	0.9366	0.633	0.3912	0.5519		
test	0.5	0.562	[0.5475, 0.5764]	0.8919	0.5362	0.3311	0.4829	0.755	[0.709, 0.801]
	0.3028	0.5186	[0.5062, 0.531]	0.9189	0.6005	0.3129	0.4668		
k-Nearest Neighbors									
train	0.5	0.7122	[0.6795, 0.7663]	0.3028	0.1101	0.5443	0.3891	0.739	[0.691, 0.787]
	0.3028	0.7335	[0.6988, 0.7562]	0.6338	0.2232	0.5521	0.5902		
test	0.5	0.7748	[0.7524, 0.8033]	0.2523	0.0697	0.5185	0.3394	0.746	[0.697, 0.796]
	0.3028	0.7459	[0.7211, 0.7677]	0.5405	0.193	0.4545	0.4938		
Recursive Partitioning									
train	0.5	0.7612	[0.7217, 0.7897]	0.3873	0.0765	0.6875	0.4955	0.782	[0.737, 0.826]
	0.3028	0.7335	[0.7039, 0.7564]	0.7113	0.2569	0.5459	0.6177		
test	0.5	0.7521	[0.7281, 0.7789]	0.3153	0.118	0.443	0.3684	0.66	[0.601, 0.719]
	0.3028	0.7025	[0.6808, 0.7237]	0.5495	0.252	0.3935	0.4586		
Random Forests									
train	0.5	0.7591	[0.7271, 0.7868]	0.4789	0.1193	0.6355	0.5462	0.758	[0.712, 0.804]
	0.3028	0.6631	[0.6397, 0.6951]	0.7254	0.3639	0.464	0.5659		
test	0.5	0.7624	[0.7397, 0.7893]	0.2793	0.0938	0.4697	0.3503	0.73	[0.68, 0.781]

	0.3028	0.6632	[0.6384, 0.6901]	0.6577	0.3351	0.3687	0.4725		
XGBoost									
train	0.5	0.8465	[0.7569, 0.8849]	0.5775	0.0367	0.8723	0.6949	0.896	[0.866, 0.926]
	0.3028	0.7697	[0.7484, 0.7868]	0.8662	0.2722	0.5802	0.6949		
test	0.5	0.7727	[0.7438, 0.7996]	0.2883	0.0831	0.5079	0.3678	0.762	[0.716, 0.808]
	0.3028	0.7045	[0.6715, 0.7335]	0.6577	0.2815	0.4101	0.5052		
Convolutional Neural Network									
train	0.5	0.7846	[0.7783, 0.8401]	0.2887	0.0015	0.954	0.4481	0.842	[0.798, 0.886]
	0.3028	0.7569	[0.7356, 0.7762]	0.7746	0.2508	0.5729	0.6587		
test	0.5	0.7748	[0.7707, 0.8017]	0.027	0.0027	0.75	0.0522	0.717	[0.665, 0.77]
	0.3028	0.7025	[0.6777, 0.7232]	0.5495	0.252	0.3935	0.4586		

Note. 95% confidence intervals for the accuracy and AUC are provided. Precision is defined as the number of true positives divided by the total number of true positives and false positives. F1 is the harmonic mean of precision and hit rate.

Appendix B. Additional material, large data set.

Table 10: *MANOVA results for the LIWC variables, large training set (Nov 17 - mid July 19).*

LIWC name	**Variable name**	**Mean if correct**	**Mean if incorrect**	**F stat.**	**p-value**	**Sig.**	**Bayes Factor**	**Cohen's d**	**CI**
achieve	Achievement	2.08	1.79	6.48	0.018	*	0.940	0.08	[0.02,0.15]
adj	Adjectives	6.54	6.37	0.57	0.502		0.050	0.03	[-0.04,0.09]
adverb	Adverbs	4.16	5.08	33.67	0.000	***	>1,000	-0.19	[-0.26,-0.13]
affect	Emotions	9.62	7.61	53.26	0.000	***	>1,000	0.24	[0.18,0.31]
affiliation	Affiliation	3.19	2.38	33.35	0.000	***	>1,000	0.19	[0.13,0.26]
AllPunc	All punctuation	23.84	20.47	34.17	0.000	***	>1,000	0.19	[0.13,0.26]
Analytic	Analytic thinking	73.17	69.66	15.12	0.000	***	68.890	0.13	[0.06,0.19]
anger	Anger	0.63	0.78	3.34	0.096		0.200	-0.06	[-0.13,0.00]
anx	Anxiety	0.15	0.18	1.05	0.371		0.060	-0.03	[-0.10,0.03]
Apostro	Apostrophes	0.73	1.07	24.42	0.000	***	>1,000	-0.16	[-0.23,-0.10]
article	Articles	6.41	7.22	26.72	0.000	***	>1,000	-0.17	[-0.24,-0.11]
assent	Assent	0.10	0.05	1.85	0.224		0.090	0.05	[-0.02,0.11]
At	@	1.50	0.18	100.70	0.000	***	>1,000	0.33	[0.27,0.40]
Authentic	Authentic	33.97	37.68	11.28	0.001	**	10.200	-0.11	[-0.18,-0.05]
auxverb	Auxiliary verbs	7.61	9.02	57.69	0.000	***	>1,000	-0.25	[-0.32,-0.19]
bio	Biology	0.84	0.58	13.08	0.001	**	24.970	0.12	[0.05,0.19]
body	Body	0.12	0.08	3.07	0.112		0.170	0.06	[-0.01,0.12]

cause	Causations	1.28	1.60	12.41	0.001	**	17.880	-0.12	[-0.18,-0.05]
certain	Certainty	1.85	2.22	12.27	0.001	**	16.680	-0.12	[-0.18,-0.05]
Clout	Clout	68.50	61.12	81.51	0.000	***	>1,000	0.30	[0.23,0.37]
cogproc	Cognitive processes	7.47	10.46	184.28	0.000	***	>1,000	-0.45	[-0.52,-0.38]
Colon	Colons	0.32	0.03	32.30	0.000	***	>1,000	0.19	[0.12,0.25]
Comma	Commas	3.77	4.13	4.80	0.042	*	0.410	-0.07	[-0.14,-0.01]
compare	Comparison words	1.50	2.60	129.36	0.000	***	>1,000	-0.38	[-0.44,-0.31]
conj	Conjunctions	4.27	5.09	38.90	0.000	***	>1,000	-0.21	[-0.27,-0.14]
Crisis	Crisis period	0.16	0.24	31.76	0.000	***	>1,000	-0.19	[-0.25,-0.12]
Dash	Dashes	0.77	0.71	0.85	0.415		0.060	0.03	[-0.03,0.10]
death	Death	0.28	0.14	2.31	0.177		0.120	0.05	[-0.01,0.12]
Dic	Dictionary words	78.47	80.95	37.02	0.000	***	>1,000	-0.20	[-0.27,-0.14]
differ	Differentiation	1.68	2.96	168.54	0.000	***	>1,000	-0.43	[-0.50,-0.37]
discrep	Discrepancy	1.19	1.52	18.26	0.000	***	327.220	-0.14	[-0.21,-0.08]
drives	Drives	12.29	10.98	23.86	0.000	***	>1,000	0.16	[0.10,0.23]
excl	Exclusive	1.45	2.11	52.79	0.000	***	>1,000	-0.24	[-0.31,-0.18]
Exclam	Exclamation marks	5.78	3.26	72.17	0.000	***	>1,000	0.28	[0.22,0.35]
family	Family	0.17	0.09	5.01	0.038	*	0.450	0.07	[0.01,0.14]
Favorite_count	Favorites	79805.68	95362.96	163.49	0.000	***	>1,000	-0.42	[-0.49,-0.36]
feel	Feeling	0.31	0.28	0.37	0.594		0.040	0.02	[-0.04,0.09]
female	Female references	0.39	0.61	9.07	0.004	**	3.410	-0.10	[-0.17,-0.03]
filler	Fillers	0.01	0.02	4.23	0.057		0.310	-0.07	[-0.13,0.00]

focusfuture	Future orientation	1.99	1.41	32.43	0.000	***	>1,000	0.19	[0.12,0.25]
focuspast	Past orientation	2.50	3.59	76.67	0.000	***	>1,000	-0.29	[-0.36,-0.23]
focuspresent	Present orientation	9.28	9.49	0.87	0.413		0.060	-0.03	[-0.10,0.03]
friend	Friends	0.19	0.14	1.74	0.239		0.090	0.04	[-0.02,0.11]
function.	Total function words	43.92	47.68	90.22	0.000	***	>1,000	-0.32	[-0.38,-0.25]
Hashtag	#	1.56	0.15	26.72	0.000	***	>1,000	0.17	[0.11,0.24]
health	Health	0.35	0.33	0.11	0.775		0.040	0.01	[-0.05,0.08]
hear	Hearing	0.33	0.34	0.02	0.902		0.040	-0.01	[-0.07,0.06]
home	Home	0.37	0.23	9.81	0.003	**	4.920	0.10	[0.04,0.17]
humans	Humans	0.78	0.75	0.30	0.630		0.040	0.02	[-0.05,0.08]
i_	First-person singular pronouns	1.54	1.21	14.21	0.000	***	43.700	0.13	[0.06,0.19]
incl	Inclusive	4.95	4.90	0.14	0.746		0.040	0.01	[-0.05,0.08]
informal	Informal	0.35	0.30	1.16	0.353		0.070	0.04	[-0.03,0.10]
ingest	Ingestion	0.09	0.07	0.72	0.451		0.050	0.03	[-0.04,0.09]
inhib	Inhibition	0.77	1.02	8.33	0.006	**	2.350	-0.10	[-0.16,-0.03]
insight	Insight	0.95	1.22	14.55	0.000	***	51.860	-0.13	[-0.19,-0.06]
interrog	Interrogatives	0.86	1.28	50.82	0.000	***	>1,000	-0.24	[-0.30,-0.17]
ipron	Impersonal pronouns	2.94	3.76	51.52	0.000	***	>1,000	-0.24	[-0.30,-0.17]
leisure	Leisure	0.74	0.46	14.07	0.000	***	40.740	0.12	[0.06,0.19]
male.	Male references	0.84	0.68	5.33	0.033	*	0.530	0.08	[0.01,0.14]
Metaph	Metaphysical	0.49	0.25	9.96	0.003	**	5.290	0.10	[0.04,0.17]
money	Money	1.22	1.98	63.31	0.000	***	>1,000	-0.26	[-0.33,-0.20]

motion	Motion	1.60	1.65	0.29	0.633		0.040	-0.02	[-0.08,0.05]
negate	Negations	1.15	2.12	134.13	0.000	***	>1,000	-0.38	[-0.45,-0.32]
negemo	Negative emotions	2.11	3.43	80.61	0.000	***	>1,000	-0.30	[-0.36,-0.23]
netspeak	Netspeak	0.11	0.09	0.49	0.535		0.050	0.02	[-0.04,0.09]
nonflu	Nonfluencies	0.11	0.09	0.90	0.408		0.060	0.03	[-0.03,0.10]
number	Numbers	1.71	2.08	8.40	0.006	**	2.440	-0.10	[-0.16,-0.03]
Optim	Optimism	1.17	1.06	1.92	0.217		0.100	0.05	[-0.02,0.11]
Other_	Total third person	1.63	2.01	18.13	0.000	***	306.520	-0.14	[-0.21,-0.08]
OtherP	Other punctuation	3.72	1.44	52.41	0.000	***	>1,000	0.24	[0.18,0.31]
Parenth	Parentheses (pairs)	0.53	0.94	33.39	0.000	***	>1,000	-0.19	[-0.26,-0.13]
percept	Perceptual processes	1.58	1.36	4.65	0.045	*	0.380	0.07	[0.01,0.14]
Period	Periods	7.24	7.62	2.10	0.198		0.110	-0.05	[-0.11,0.02]
posemo	Positive emotions	7.46	4.13	160.97	0.000	***	>1,000	0.42	[0.36,0.49]
power	Power	4.88	4.95	0.19	0.703		0.040	-0.01	[-0.08,0.05]
ppron	Personal pronouns	6.61	5.46	30.71	0.000	***	>1,000	0.18	[0.12,0.25]
prep	Prepositions	12.47	12.47	0.00	0.979		0.040	0.00	[-0.06,0.07]
pronoun	Total pronouns	9.55	9.23	1.96	0.213		0.100	0.05	[-0.02,0.11]
QMark	Question marks	0.49	0.51	0.01	0.944		0.040	0.00	[-0.07,0.06]
quant	Quantifiers	1.67	2.20	32.17	0.000	***	>1,000	-0.19	[-0.25,-0.12]
Quote	Quotation marks	0.48	0.76	16.43	0.000	***	131.460	-0.13	[-0.20,-0.07]
relativ	Relativity	14.35	14.66	1.08	0.369		0.060	-0.03	[-0.10,0.03]
relig	Religion	0.46	0.06	27.86	0.000	***	>1,000	0.18	[0.11,0.24]

Retweet_count	Retweets	17364.81	22071.88	218.70	0.000	***	>1,000	-0.49	[-0.56,-0.43]
reward	Reward focus	2.95	2.07	37.60	0.000	***	>1,000	0.20	[0.14,0.27]
risk	Risk focus	0.75	1.21	53.97	0.000	***	>1,000	-0.24	[-0.31,-0.18]
sad	Sadness	0.38	0.57	12.82	0.001	**	21.920	-0.12	[-0.18,-0.05]
see	Seeing	0.90	0.71	5.47	0.031	*	0.570	0.08	[0.01,0.14]
Self	Total first person	3.31	3.03	4.84	0.042	*	0.420	0.07	[0.01,0.14]
Senses	Sensory and Perceptual Processes	1.20	1.12	1.15	0.353		0.070	0.04	[-0.03,0.10]
sexual	Sexual	0.08	0.08	0.01	0.927		0.040	0.00	[-0.06,0.07]
shehe	Third-person singular pronouns	0.79	0.73	1.02	0.376		0.060	0.03	[-0.03,0.10]
Sixltr	Six-letter words	22.98	20.82	29.81	0.000	***	>1,000	0.18	[0.12,0.25]
social	Social processes	9.26	8.16	20.17	0.000	***	841.980	0.15	[0.08,0.21]
space	Space	7.73	8.27	6.80	0.015	*	1.100	-0.09	[-0.15,-0.02]
swear	Swear words	0.02	0.03	0.80	0.428		0.060	-0.03	[-0.09,0.04]
tentat	Tentative	1.34	2.16	96.52	0.000	***	>1,000	-0.33	[-0.39,-0.26]
they	Third-person plural pronouns	0.81	1.34	64.50	0.000	***	>1,000	-0.27	[-0.33,-0.20]
time	Time	5.47	5.02	5.18	0.035	*	0.490	0.08	[0.01,0.14]
Tone	Emotional tone	63.39	43.46	228.43	0.000	***	>1,000	0.50	[0.44,0.57]
verb.	Common verbs	14.03	15.30	24.35	0.000	***	>1,000	-0.16	[-0.23,-0.10]
WC	Word quantity	30.25	39.45	346.63	0.000	***	>1,000	-0.62	[-0.68,-0.55]
we.	First-person plural pronouns	1.83	1.68	2.28	0.178		0.120	0.05	[-0.01,0.12]
work	Job/Work	4.55	4.37	0.63	0.481		0.050	0.03	[-0.04,0.09]
WPS	Average sentence length	13.03	15.26	79.07	0.000	***	>1,000	-0.30	[-0.36,-0.23]
you	Total second-person pronouns	1.64	0.51	54.74	0.000	***	>1,000	0.25	[0.18,0.31]

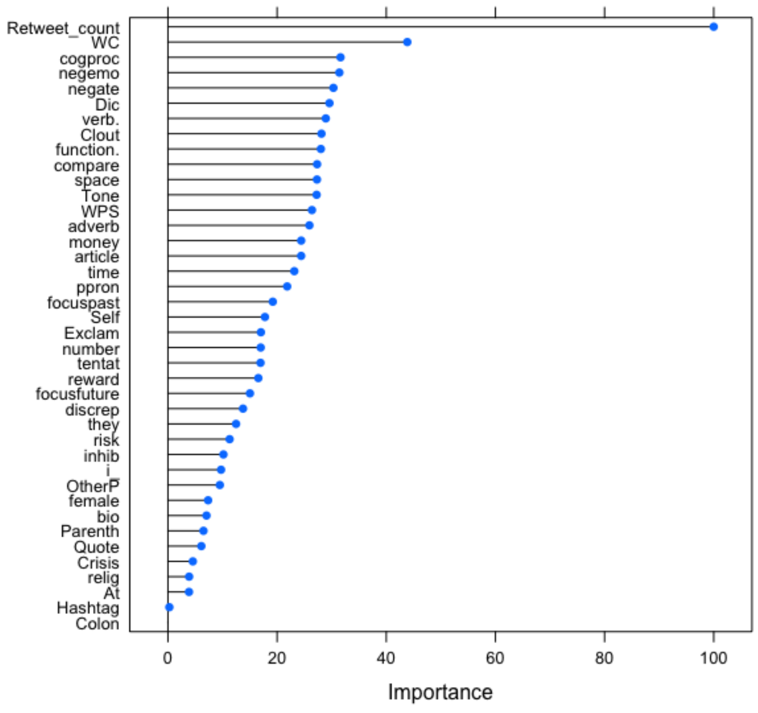
Table 11: *All performance metrics of the different models*

Data	Cut-off	Accuracy	Acc. CI	Hit rate	False alarm rate	Precision	F1	AUC	AUC CI
Logistic regression									
train	0.5	0.7665	[0.7558, 0.7771]	0.366	0.0852	0.614	0.4587	0.804	[0.791, 0.818]
	0.2703	0.7021	[0.6938, 0.7117]	0.7892	0.3302	0.4696	0.5888		
test	0.5	0.7335	[0.7231, 0.7453]	0.277	0.1009	0.4989	0.3562	0.757	[0.74, 0.775]
	0.2703	0.7035	[0.6911, 0.7153]	0.6814	0.2884	0.4614	0.5502		
Support Vector Machine									
train	0.5	0.8021	[0.7871, 0.8137]	0.3331	0.0241	0.8364	0.4764	0.882	[0.871, 0.892]
	0.2703	0.8269	[0.8171, 0.8378]	0.6484	0.107	0.6918	0.6694		
test	0.5	0.7466	[0.7391, 0.7541]	0.1275	0.0289	0.6154	0.2112	0.72	[0.7, 0.741]
	0.2703	0.7306	[0.7202, 0.743]	0.3627	0.136	0.4917	0.4175		
Naïve Bayes									
train	0.5	0.5234	[0.5192, 0.5279]	0.9204	0.6237	0.3534	0.5107	0.752	[0.738, 0.767]
	0.2703	0.5158	[0.511, 0.5203]	0.9276	0.6368	0.3505	0.5087		
test	0.5	0.5672	[0.5597, 0.5744]	0.8505	0.5356	0.3655	0.5112	0.731	[0.711, 0.75]
	0.2703	0.5568	[0.5496, 0.5639]	0.8591	0.5529	0.3604	0.5078		
k-Nearest Neighbors									
train	0.5	0.7521	[0.7423, 0.7611]	0.3355	0.0936	0.5705	0.4225	0.766	[0.751, 0.78]

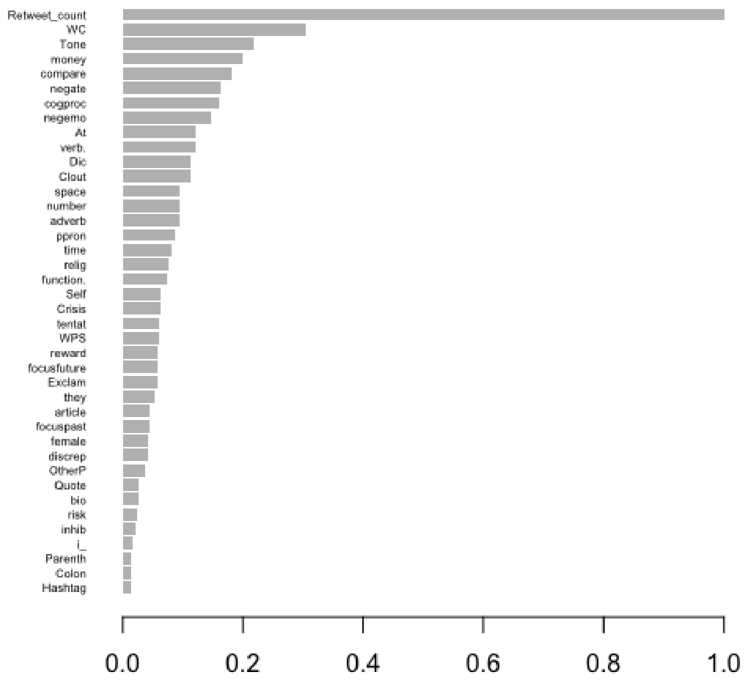
	0.2703	0.681	[0.6728, 0.6884]	0.7619	0.3489	0.4471	0.5635		
test	0.5	0.7299	[0.7209, 0.7398]	0.1961	0.0764	0.4819	0.2787	0.717	[0.698, 0.736]
	0.2703	0.6624	[0.6513, 0.672]	0.663	0.3378	0.4158	0.5111		
Recursive Partitioning									
train	0.5	0.7519	[0.7401, 0.7605]	0.5294	0.1657	0.542	0.5356	0.761	[0.746, 0.776]
	0.2703	0.6841	[0.6766, 0.6916]	0.745	0.3385	0.4491	0.5604		
test	0.5	0.725	[0.7117, 0.7351]	0.364	0.144	0.4783	0.4134	0.691	[0.67, 0.711]
	0.2703	0.6693	[0.6589, 0.6802]	0.5441	0.2853	0.4088	0.4669		
Random Forests									
train	0.5	0.761	[0.7497, 0.771]	0.3323	0.0802	0.6056	0.4291	0.795	[0.781, 0.808]
	0.2703	0.6719	[0.6636, 0.6784]	0.8343	0.3883	0.4432	0.5788		
test	0.5	0.7417	[0.7326, 0.7538]	0.185	0.0564	0.5432	0.2761	0.736	[0.717, 0.754]
	0.2703	0.6533	[0.6425, 0.6621]	0.7059	0.3658	0.4117	0.5201		
XGBoost									
train	0.5	0.7886	[0.7778, 0.7993]	0.3685	0.0557	0.7101	0.4852	0.846	[0.834, 0.857]
	0.2703	0.7162	[0.7102, 0.7225]	0.8705	0.3409	0.4861	0.6238		
test	0.5	0.7469	[0.7381, 0.758]	0.1936	0.0524	0.5725	0.2894	0.759	[0.741, 0.777]
	0.2703	0.6928	[0.6813, 0.7035]	0.663	0.2964	0.4478	0.5346		

Convolutional Neural Network									
train	0.5	0.8026	[0.7897, 0.8143]	0.4771	0.0769	0.6968	0.5664	0.871	[0.861, 0.882]
	0.2703	0.7241	[0.7199, 0.7282]	0.9332	0.3534	0.4945	0.6464		
test	0.5	0.7436	[0.7342, 0.7551]	0.19	0.0556	0.5536	0.2828	0.751	[0.733, 0.769]
	0.2703	0.6556	[0.6464, 0.6631]	0.7953	0.3951	0.422	0.5514		

Note. 95% confidence intervals for the accuracy and AUC are provided. Precision is defined as the number of true positives divided by the total number of true positives and false positives. F1 is the harmonic mean of precision and hit rate.



(a) Random Forest



(b) XGBoost

Figure 11: Variable importance of the included variables in the models. The importance is expressed as a percentage of the most important variable Retweet count.

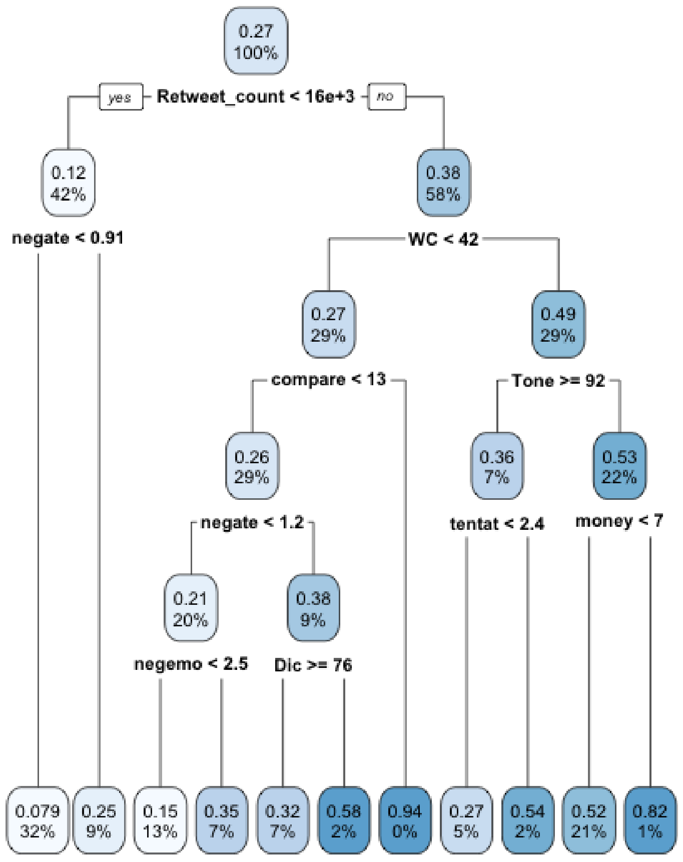
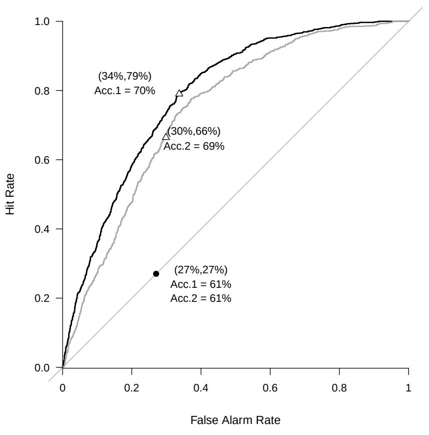


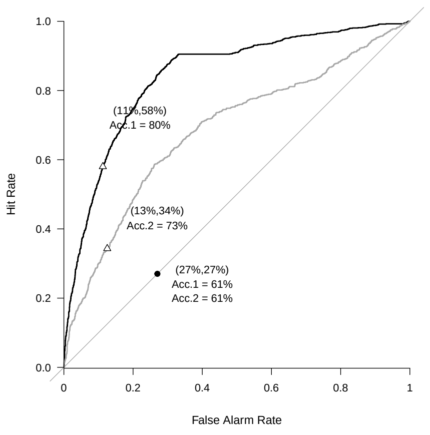
Figure 12: The tree constructed by the Recursive Partitioning algorithm. A white square corresponds to the correct tweets class, whereas blue corresponds to the incorrect tweets class.

Appendix C. Parsimonious approach.

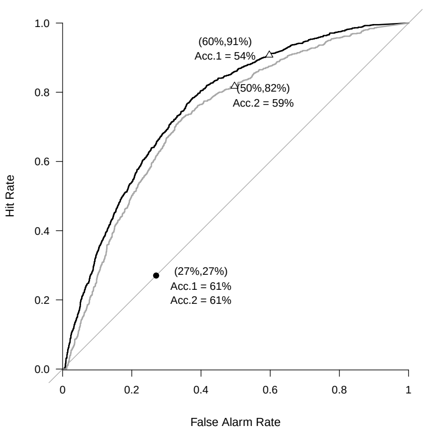
In this Appendix, the number of variables included in the models is reduced by 50% (except for CNN that does not takes variables as input). Instead of the 40 variables, as is shown in Tables 2 and 3, I provide only the 20 most important variables as input to the models (i.e., the 20 variables that reduce the AIC most). The purpose of reducing the dimensionality is to examine the presence of overfitting. Perhaps that a more parsimonious approach results in less overfitting, which was the case for the SVM model using 40 variables. The results are shown in Figure 13 and Table 12. Indeed, overfitting is less of a problem with the more parsimonious approach, at the cost of a slightly worse performance overall.



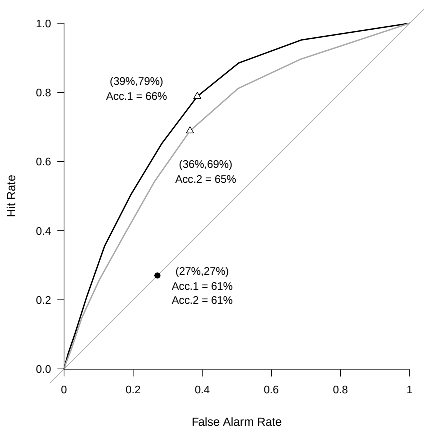
(a) Logistic regression



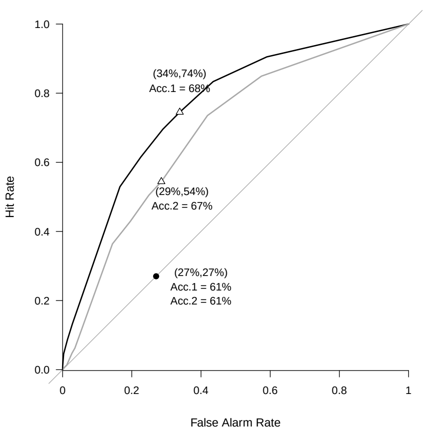
(b) Support Vector Machine



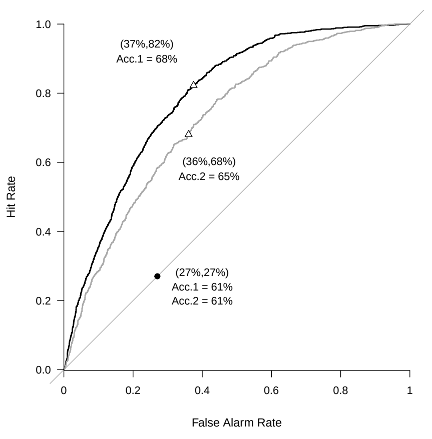
(c) Naïve Bayes



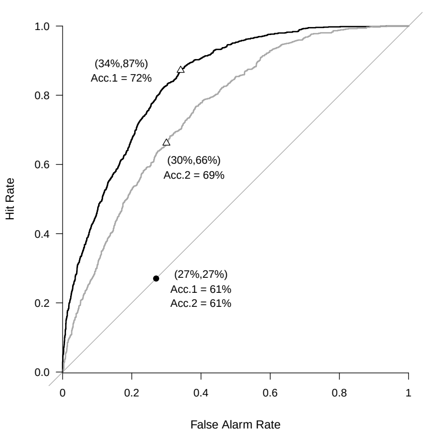
(d) k-Nearest Neighbors



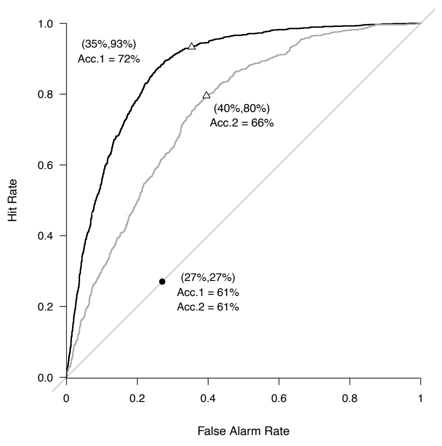
(e) Recursive Partitioning



(f) Random Forest



(g) XGBoost



(h) Convolutional Neural Network

Figure 13: Hit rates plotted against the false alarm rates (ROC curves) for the eight different classification methods. The in-sample predictions are in black (training set, Nov 17 - mid July 19) and the out-of-sample predictions are in grey (test set, mid July 19 - April 20). Only half of the variables of the original model are included, to examine possible overfitting. The actual coordinates of the points are given by the values in the brackets, with the white triangles corresponding to the prior as cut-off. The black dot corresponds to the coordinate when the prior is used as a random guess.

Table 12: *All performance metrics for the parsimonious models*

Data	Cut-off	Accuracy	Acc. CI	Hit rate	False alarm rate	Precision	F1	AUC	AUC CI
Logistic regression									
train	0.5	0.7541	[0.7454, 0.7632]	0.3298	0.0888	0.5791	0.4203	0.788	[0.775, 0.802]

	0.2703	0.6973	[0.6878, 0.7045]	0.79	0.337	0.4647	0.5852		
test	0.5	0.7384	[0.729, 0.7472]	0.2341	0.0787	0.519	0.3226	0.744	[0.726, 0.763]
	0.2703	0.6915	[0.6804, 0.7071]	0.6642	0.2987	0.4465	0.534		
Support Vector Machine									
train	0.5	0.7839	[0.7728, 0.795]	0.3049	0.0387	0.7446	0.4326	0.843	[0.83, 0.856]
	0.2703	0.8043	[0.7915, 0.8143]	0.58	0.1126	0.6561	0.6157		
test	0.5	0.7505	[0.7414, 0.7583]	0.1593	0.0351	0.622	0.2537	0.685	[0.663, 0.707]
	0.2703	0.7332	[0.7234, 0.7443]	0.3431	0.1253	0.4982	0.4064		
Naïve Bayes									
train	0.5	0.573	[0.5677, 0.5786]	0.8801	0.5408	0.3761	0.527	0.759	[0.744, 0.774]
	0.2703	0.539	[0.534, 0.544]	0.9075	0.5974	0.36	0.5155		
test	0.5	0.6164	[0.6076, 0.6239]	0.7966	0.4489	0.3916	0.525	0.726	[0.707, 0.746]
	0.2703	0.5868	[0.5796, 0.5952]	0.8174	0.4969	0.3737	0.5129		
k-Nearest Neighbors									
train	0.5	0.7402	[0.7297, 0.7491]	0.3556	0.1174	0.5287	0.4252	0.752	[0.738, 0.767]
	0.2703	0.6614	[0.6531, 0.6682]	0.7884	0.3856	0.431	0.5573		
test	0.5	0.7277	[0.7176, 0.7378]	0.2561	0.1013	0.4783	0.3336	0.699	[0.679, 0.719]
	0.2703	0.6497	[0.6391, 0.6592]	0.6887	0.3644	0.4067	0.5114		
Recursive Partitioning									

train	0.5	0.7519	[0.74, 0.7609]	0.5294	0.1657	0.542	0.5356	0.761	[0.746, 0.776]
	0.2703	0.6841	[0.6765, 0.6915]	0.745	0.3385	0.4491	0.5604		
test	0.5	0.725	[0.7121, 0.7344]	0.364	0.144	0.4783	0.4134	0.691	[0.67, 0.711]
	0.2703	0.6693	[0.6583, 0.6805]	0.5441	0.2853	0.4088	0.4669		
Random Forests									
train	0.5	0.753	[0.7432, 0.7632]	0.3298	0.0903	0.575	0.4192	0.791	[0.777, 0.804]
	0.2703	0.6784	[0.671, 0.6849]	0.8222	0.3749	0.4482	0.5802		
test	0.5	0.744	[0.7326, 0.7557]	0.201	0.0591	0.5522	0.2947	0.727	[0.708, 0.746]
	0.2703	0.6504	[0.6402, 0.6624]	0.6801	0.3604	0.4063	0.5087		
XGBoost									
train	0.5	0.7832	[0.7723, 0.7917]	0.3685	0.0632	0.6836	0.4788	0.838	[0.827, 0.85]
	0.2703	0.7165	[0.7075, 0.7223]	0.8721	0.3412	0.4863	0.6244		
test	0.5	0.7482	[0.7391, 0.7586]	0.2096	0.0564	0.5738	0.307	0.754	[0.736, 0.772]
	0.2703	0.6898	[0.6794, 0.7003]	0.6618	0.3	0.4444	0.5318		

Note. 95% confidence intervals for the accuracy and AUC are provided. Precision is defined as the number of true positives divided by the total number of true positives and false positives. F1 is the harmonic mean of precision and hit rate.