Universiteit Leiden
The Netherlands

Opleiding Informatica & Economie

Classifying medical misinformation on health forums

Ceyhan Deve

Supervisors:
Suzan Verberne & Anne Dirkson

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

25/07/2020

Abstract

Due to enormous rise in popularity of social media platforms, the dangers of medical misinformation are raised to a new level. As social media platforms have enabled internet users to create and share their own content in different forms, medical misinformation can spread more rapidly than ever before. The spread of medical misinformation can have a great impact on society and public health. This spread has therefore lead to public discourse and new challenges for governments, businesses and the academic world. One of these challenges is the development of tools for the automatic detection of (medical) misinformation. In this thesis, we studied the automatic detection of medical misinformation on health forums with the use of various predictive models. These models are built on messages from the online health platform MedHelp which consists of various forums. We replicated the feature selection approach of a related study on this subject and identified 63 features as being the most important feature set for the detection of medical misinformation on health forums. With this set of features we obtained an average accuracy of 86.3%. For our replication study we developed and employed a comparison process to compare the performances of our optimal feature set, that of the related study and the set of all useful numerical features from the used dataset. We found that all set of features yield nearly similar performances in average accuracy. Besides this replication study, we extracted three different feature sets based on forum posts and compared their performances in a 10-fold group cross-validation setup with the use of different classification algorithms and resampling methods. We obtained the highest F_1 score of 0.439 on a set of tf-idf features with the use of the logistic regression classifier. This score was more than 1.5 as high as a baseline approach which identifies all posts in the dataset as misinformative. With the use of nested cross-validation we tuned the hyperparameters of the best classification algorithms for each feature set. We were able to slightly increase the F_1 score corresponding to the misinformative class of our best performing classification algorithm (the logistic regression classifier on tf-idf feature set) to 0.443. However, the recall corresponding to this class decreased slightly in comparison to the obtained recall with the single cross-validation procedure.

Acknowledgements

The COVID-19 outbreak has changed our daily lives drastically in the past few months. All thesis projects had to be conducted online. Despite this unusual situation my supervisors Suzan Verberne and Anne Dirkson managed to guide me during my bachelor thesis very well. I would like to thank Suzan Verberne for always being available to answer my questions and pointing me in the right direction. Furthermore, I would like to thank Anne Dirkson for providing me with the dataset and giving me comprehensive feedback on my thesis.

Contents

Chapter 1

Introduction

On 31 December 2019, the Wuhan Municipal Health Commission alerted the World Health Organization (WHO) to several cases of unusual pneumonia in Wuhan, a city of 11 million people in the Hubei province in China (WHO, 2020). As of 3 January 2020, 44 patients of which 11 in critical condition had been diagnosed with this pneumonia-like illness. The majority of these cases were linked to Wuhan's Huanan Seafood Wholesale Market, a wet market selling especially fish and wild animals (ECDC, n.d.). Merely four days later on 7 January 2020, the Chinese Center for Disease Control and Prevention (CCDC) identified a novel coronavirus (2019-nCoV), later renamed as severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), as the cause of this mysterious illness. This illness is now known as the Coronavirus disease 2019 (COVID-19) and characterized with symptoms as fever, dry cough and shortness of breath.

More and more cases started to appear and the condition of some patients severely deteriorated leading to the first deaths due to COVID-19. Quickly cases also started to emerge in other countries like the United States, Singapore and Taiwan. The virus has spread rapidly worldwide and the numbers of cases and deaths continued to rise. Eventually on 11 March 2020, the WHO declared the COVID-19 outbreak as a pandemic urging countries to take immediate actions to minimize the spread of the virus. The COVID-19 outbreak had transformed from a small outbreak into a global health crisis.

At the same time, misinformation about the coronavirus started spreading and currently still is as public health scientists are trying to understand the spread and trying to find a vaccine. Misinformation can be defined as false information that is shared without malicious intent. A substantial part of the misinformation that is being spread about the coronavirus is health-related. Messages claiming that home remedies can prevent people from getting the virus or can even be a cure for the virus are widely spread (Mian & Khan, 2020). For example, there are claims circulating that gargling with salt, eating garlic and taking vitamin C can be cures for the virus. Most of this misinformation is shared via social media platforms like Twitter, Facebook and Instagram, but also via blogs and forums. Social media has changed the way people communicate as it allows users to create and share their

own content in different forms to many others.

Although this seems beneficial, it can be dangerous when it comes to sharing medical misinformation. Medical misinformation can spread rapidly and people can get misinformed easily as the internet has become a popular resource for health information (Swire-Thompson & Lazer, 2020). Misinformation such as medical misinformation can have a great impact on society and public health. Medical misinformation can lead to distrust between people and health authorities such as between patients and doctors (Zadrozny, 2019; Hill et al., 2019). This distrust manifests itself in not complying with health treatments and preventive measures. An example is the refusal of vaccinations, which reduces the vaccination coverage level and makes it harder to combat infectious diseases in society. A possible consequence of not complying with medical treatments is the turn to alternative remedies. In recent years this has led to numerous incidents as some misinformative claims might seem innocent while in reality they are not. A recent example of such incidents is the death of 300 and the severe illness of more than 1000 people in Iran because of a message that was circulating which claimed that alcohol can be a cure for the coronavirus (Associated Press , 2020). As alcohol consumption is forbidden in Iran people rely on bootleggers which led to methanol poisoning.

The spread of misinformation has therefore gained notoriety and has lead to public discourse and new challenges for governments, businesses and the academic world. One of these challenges is the development of tools to identify misinformation. This thesis focuses on identifying misinformation on health forums using various predictive models. To build these models comments and responses posted on MedHelp are used. MedHelp is a well-known online health platform consisting of various health forums that enables users to discuss about various health-related topics. The research question is thus as follows:

To which extent can medical misinformation be detected using posts on health forums?

This thesis is structured as follows. Chapter 2 provides more information about medical misinformation. Furthermore, this chapter discusses related work to this thesis. Chapter 3 describes both the construction and the content of the used dataset. Next, chapter 4 describes the employed methods. Chapter 5 presents all the obtained results after which these results are discussed in chapter 6. Finally, chapter 7 states the most important conclusions and recommendations for furture research.

Chapter 2

Background

2.1 Medical misinformation

Medical misinformation can be defined as false health-related information that goes against the consensus of the scientific community. In contrast to disinformation, misinformation is spread without malicious intent. The spread of medical misinformation is not a new phenomenon as it might be as old as public healthcare itself (Waszak, Kasprzycka-Waszak, & Kubanek, 2018). However, as a result of the increase in internet usage and especially the enormous rise in popularity of social media platforms like Twitter and Facebook, the dangers of medical misinformation have increased to a new level. Social media made health information open to the public, which was previously only available to health care providers (Li, Wang, Lin, & Hajli, 2016). According to a survey of Eurostat conducted in 2019 on the use of internet in European households, 53% of the participants sought health information online in the three months before this survey (Eurostat, 2020). Furthermore, social media enabled users to share their (personal) health information, such as their experience and symptoms. However, this abundance of information of which a large amount is inaccurate can cause social media users to get easily misinformed (Swire-Thompson & Lazer, 2020). This has a large impact on the behaviour of internet users which makes medical misinformation a threat to public health (Waszak et al., 2018). The spread of this inaccurate information affects the usage of medicine, the adoption of measures during epidemics, and contributes to vaccine hesitancy and distrust between patient and health authorities (Gyenes & Marrelli, 2020).

Medical misinformation can spread virally through social media platforms. When this occurs during the outbreak of a particular virus (or disease) causing the obstruction of the containment and a contribution to the spread of the virus, this is referred to as a misinfodemic. Misinfodemics have occurred during the outbreaks of Ebola and measles. More recently, the Covid-19 outbreak shows characteristics of a misinfodemic.

2.2 Related Work

There has been a substantial amount of research done on the identification of misinformation on social media platforms with the help of various machine learning techniques. However, many studies do not address the automatic identification of medical misinformation. Nevertheless, there are some studies that offer insights which can be relevant for the classification of health-related misinformation. Gupta and Kumaraguru (2012) conducted a research on the automatic assessment of credibility of tweets during high-impact news events. With the use of the logistic regression classifier, the researchers analysed which contact-based and user-based features are significant for the credibility assessment of tweets. Based on these features the researchers proposed an automatic ranking technique to rank the tweets according to their level of credibility. This technique is based on a combination of the ranking SVM algorithm and relevance feedback. The researchers found that content-based features were as significant as the user-based feature to assess the credibility of tweets. Both types of features therefore play a significant role in ranking the tweets.

A research which also uses tweets during a high-impact event for the detection of misinformation is conducted by Ghenai and Mejova (2017). However, this research is more directly related to this thesis in comparison with the previous mentioned study as this research focused on the detection of health-related misinformation on social media. In this research Ghenai and Mejova (2017) built a pipeline to detect health-related misinformation on social media with the use of tweets posted during the Zika virus epidemic. The researchers collected tweets about various types of rumors about the Zika virus after which the tweets were annotated. Based on the collected tweets they extracted features which can be divided into five different categories. The researchers determined which features are the best for this particular classification task. It was found that the most significant features are medical features. Based on the best ten features classifiers were built with the use of three different classification algorithms. The best results were found with the use of a random forest classifier.

A research on the automatic detection of health-related misinformation with a nearly similar approach is conducted by Kinsora, Barron, Mei, and Vydiswaran (2017). This thesis focuses especially on this study. The lack of labelled datasets makes identifying (medical) misinformation with supervised approaches a challenge. In order to improve this condition, Kinsora et al. (2017) created a labelled dataset containing both misinformative and non-misinformative posts from online health platform MedHelp. This required the identification of candidate misinformative post, the development of a codebook and labelling strategy, and the extraction of features. For the creation of the labelled dataset another dataset was used which provided meta-data and network data about the various posts and online health communities from MedHelp. A selection of candidate misinformative posts from this original dataset was labelled with the help of various annotators and the developed codebook. The features in the dataset were extracted based on content of the posts, the structure of the threads and network data. Based on these extracted features the researchers conducted and evaluated some supervised classification approaches for medical misinformation in health forums. To find the most important set of features for such a classification

task the researchers used recursive feature elimination with random forest in a 10-fold cross-validation setup. The labelled dataset created by Kinsora et al. (2017) is used in this thesis to build classification models for the identification of health-related misinformation. Besides, the feature selection procedure of this research is replicated. Therefore, the research by Kinsora et al. (2017) is described in more detail in further sections of this thesis.

Another research on automatic detection of medical misinformation is conducted by Hou, Pérez-Rosas, Loeb, and Mihalcea (2019). Similar to the research by Kinsora et al. (2017) the researchers created an annotated dataset after which various features were analysed for the classification of medical misinformation. The research of Hou et al. (2019), however, focuses on the detection of medical misinformation in Youtube videos related to prostate cancer. Based on keywords related to prostate cancer and some filtering on content-based criteria the researchers selected a set of 250 Youtube videos. After selecting these videos, each video was labelled as being trustworthy or misinformative. The researchers then obtained transcripts of the videos and extracted various sets of features that can be divided into three categories: viewer engagement, linguistic and acoustic features. Based on these extracted features various classification models were build with the use of linear kernel support vector machine (SVM). In the context of this thesis their use of linguistic features is particularly relevant. The researchers found that all the different sets of linguistic features except of Lexical Richness features outperform the majority baseline class. This baseline is defined as the classification of all instances in the dataset as trustworthy (the default class) which gives an accuracy of 52.8%. Among all the linguistic features, the N-grams features, which represent all unique words and words-pairs in the transcriptions, gave the highest accuracy of 71.61%. With the use of all the different sets of linguistic features combined the researchers obtained an accuracy of 72.41% which outperformed each individual feature set. A slightly higher accuracy of 74.41% was obtained by using various feature sets from different categories. This was the highest accuracy the researchers obtained in their research.

Chapter 3

The dataset

3.1 The construction of the dataset

3.1.1 The original dataset

The labelled dataset that is used in this research is created by Kinsora et al. (2017). For the construction of this dataset, another dataset was used, originally constructed by Vydiswaran et al. (2014) to study the dynamics of online communities in health forums. This dataset contains more than 4.7 million messages posted on MedHelp between 2001 and 2013. The messages in this dataset are divided into 2.8 million threads (health-related questions). Besides the messages, the dataset includes data about MedHelp members and the friendships between them.

3.1.2 Finding potentially misinformative messages

To determine in which threads medical misinformation could be located, the researchers retrieved a set of evaluated medical claims from Snopes, a website that evaluates various claims on the basis of their credibility. These claims were further verified with the use of other sources such as the website of ECDC. From each retrieved claim a set of keywords was extracted and used to search for messages that potentially contain misinformation. With these defined search queries hundreds of messages were obtained. Not all of these obtained messages are directly related to the retrieved claims of Snopes and the search queries. In some cases the search queries yielded messages related to other medical misinformative claims. Each found message that the researchers suspected of containing medical misinformation was flagged. The threads that contain at least one flagged message were reconstructed and identified as potentially containing misinformation. Using this approach 293 threads were found which contain in total 2250 messages.

3.1.3 The developed codebook & the annotation experiments

In order to annotate the messages within the identified threads and ensure consistency between the annotators, the researchers developed a codebook. This codebook consists of various criteria which define medical misinformation. The researchers defined medical misinformation as a medical relation that has not been verified by the medical community. For example, the false relation between vaccines and autism. The criteria in the codebook helped the annotators assess whether a message is misinformative or not. A non-misinformative message was annotated as '0' and a misinformative message was annotated as '1'. To test the robustness of the developed codebook the researchers conducted some annotation experiments. Two of the researchers annotated a test set of 250 messages independently. The degree of similarity between the annotations, which is called the inter-rater-agreement, was measured with the use of the Cohen's kappa coefficient (shortly called Cohen's κ). Cohen's κ indicates the degree of agreement between two raters corrected for the probability that this agreement occurred by chance (McHugh, 2012; Marston, 2010). In practice it can take a value between 0 and 1, where 0 means that the agreement is equivalent to an agreement that would have occured by chance and 1 means that there is a complete agreement between the raters.

After the annotation of the 250 messages, a very low inter-rater-agreement of 0.29 was obtained. This low agreement score was blamed on the lack of details in the developed codebook. All the annotations that the researchers disagreed on were therefore analysed. Each of these annotations was discussed until a definitive annotation decision was made and a corresponding criterion was added in the codebook. With the use of this revised codebook the researchers annotated a new test set of 250 messages and obtained a slightly higher agreement score of 0.55. Two new annotators then annotated this second set of messages independently with the same codebook. The agreement score between the first researcher and the two annotators was respectively 0.47 and 0.39. Based on these relatively low scores the researchers concluded that the annotation of comments as misinformative or not is a complex task.

3.1.4 The final annotation decisions

After the annotation experiment was completed, the messages in the identified threads were annotated by an annotation team. This annotation team consisted of three annotators, one of whom was is an expert. The comments were annotated based on a majority agreement. For most of the comments (1560 comments) all annotators agreed, and in the case of a tie between the non-experts, the annotation decision of the expert was decisive. Only 72 comments were annotated based on a majority formed by the two non-experts. It is important to note that the researchers have not described how the annotation decisions for the rest of the comments in the dataset were made. Besides, not all 2250 messages were annotated and can be found in the created dataset. The research does not indicate for what reason some messages were not annotated and left out of the dataset.

Besides the selection and annotation of messages, the researchers extracted various features. In section 3.2 the content of the dataset, among which the extraction of features, is eleborated.

3.2 The content of the dataset

The created dataset consists of 1558 labelled messages divided into various threads (health-related questions). Most of these messages, 1325, are labelled as non-misinformative which implies that the distribution of the class attribute is skewed and therefore the dataset is imbalanced. This is shown in Figure 3.1.

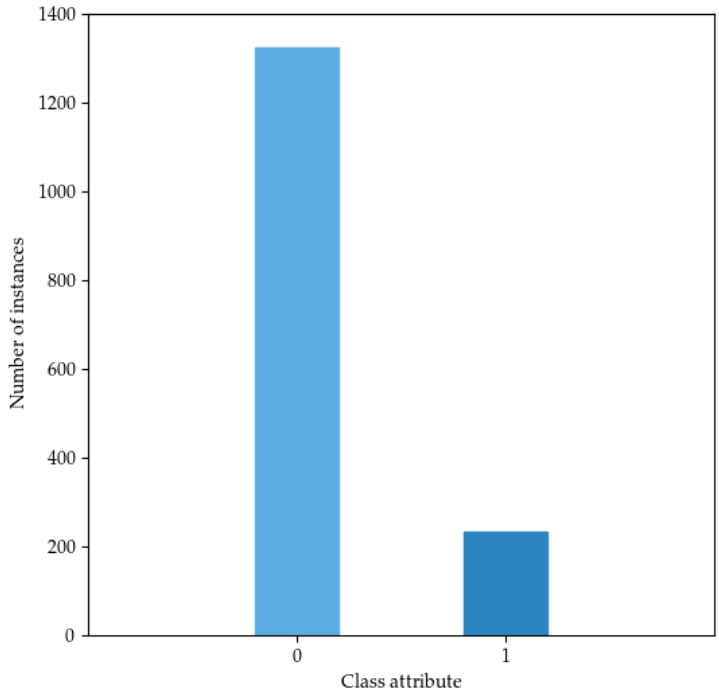

Figure 3.1: Distribution of the class attribute

The dataset contains 114 features. Most of these features are extracted with the help of LIWC (edition 2015). LIWC is a text analysis application which provides a method to study text based on its cognition, emotion and structure (Pennebaker, Boyd, Jordan, & Blackburn, 2015). With the use of a build-in dictionary, it determines to which extent various word categories are present in a certain text. LIWC2015 includes 90 different categories which can be divided into the following groups: summary language variables, linguistic dimensions, other grammar and psychological processes. For most word categories LIWC works as follows. LIWC analyses each word at a time in a (particular) piece of text. If there is a match with a word in the dictionary, the counter variable of the corresponding word category is incremented. After all words in the text are analysed, LIWC calculates the percentage of the total amount of words in each word category.

A word in the dictionary can correspond to more than one word category and some categories can be sub-categories of one or more broader categories. The categories are thus ordered hierarchically. For example, the word 'cried' corresponds to five different word categories. If 'cried' is in the analysed text, all the counter variables of these five categories will be incremented. All the words related to sadness such as 'cried' also belong to two broader categories.

Besides the features extracted with the use of LIWC, the dataset consists of some other features related to other concepts. The researchers for example developed various dictionaries to extract features which represent the number of terms related to medical professions, hearsay and equivocation in each message. In addition, the dataset contains two features related to the threads: a feature representing a running count of the misinformative messages in a thread and one representing a score indicating the distance between a message and the most recent misinformative message. Finally, several network features can be found in the dataset such as the implicit friendship and explicit friendship count of the poster of a message. All in all, the features in the dataset are extracted based on content of the messages, the structure of the threads and network data.

Chapter 4

Methods

4.1 Overview

The automatic identification of medical misinformation on health forums is a text classification task which involves various data mining techniques. Text classification is the multi-step process where text documents are assigned to various classes or categories. In order to detect medical misinformation on health forums, various predictive models are built which are able to classify posts as non-misinformative or misinformative. These predictive models are built with the use of supervised learning. Supervised learning is a form of machine learning in which a classifier learns from labelled data. The labelled data, which serves as a training set, consists of attributes (input) and the corresponding class attribute (output). To train the classifier, the classifier is fed with the input along with the corresponding output. The classifier then tries to learn a function that maps the input to the desired output. Using this function, the classifier can try to predict the output of unseen data. As the labels of the unseen data are known, it is possible to compare the predictions with these labels and assess the performance of the classifier. As stated before, the classifiers in this research are built on (labelled) forum posts of various threads on MedHelp.

This research can be divided into two parts: the replication of the feature selection process conducted by Kinsora et al. (2017) and the comparison of various feature sets. In the second part, the following feature sets, which represent the forum posts, are considered for comparison:

- **Bag-of-words features**
  - Countvectorizer features
  - Tf-idf features
- **Style-based features**

4.2 Replication study

4.2.1 Approach of Kinsora et al. (2017)

As stated before, Kinsora et al. (2017) determined which of their extracted features are the most important for the classification of forum posts as non-misinformative or misinformative. To find this most important set of features for such a classification task, the researchers used recursive feature elimination with a random forest classifier in a 10-fold cross-validation setup. Recursive feature elimination is a backwards feature selection approach in which features are removed recursively until an optimal set of features is obtained (Kuhn & Johnson, 2019). This approach begins with training a particular classifier on the entire set of features after which an importance score for each feature is computed. The least important feature is then removed and the classifier is retrained on the remaining features. The importance score for each of these features is computed again. This process repeats itself until a minimum number of features, which can be set by a parameter, is reached. Besides a minimum number of features to be selected, the number of features that are removed during each iteration can be set.

Recursive feature elimination is often combined with cross-validation (RFECV). When recursive feature elimination is combined with cross-validation, such as employed by Kinsora et al. (2017), the selection procedure is repeated for each iteration of the cross-validation process. A particular dataset is for example split in k folds, after which the classifier is trained on all the features and an importance score for each feature is calculated. The least important feature is then removed and the classifier is trained with the remaining features on the same folds. This process continues until the set minimum number of features is reached. The cross-validation process goes to the next iteration and the whole recursive selection procedure repeats until all the iterations of the cross-validation process are completed. After RFECV is completed, the mean predictive performance score for each number of features from all the cross-validation splits can be obtained. In the case of the research conducted by Kinsora et al. (2017), the researchers used the average accuracy to evaluate the mean predictive performance of the classifiers for each number of features. In this implementation RFECV selects the set of features which has attained the highest average accuracy during the selection procedure. The feature importances in research of Kinsora et al. (2017) were based on two metrics: the Gini importance and the permutation importance (Zhang & Ma, 2012).

4.2.2 Replication approach

Replication of RFECV

In order to replicate the feature selection procedure conducted by Kinsora et al. (2017) and evaluate their findings, their approach is adapted as closely as possible. As the researchers, however, have not described their approach in detail, assumptions are made during the replication.

To replicate the feature selection procedure, we employ recursive feature elimination with a random forest

classifier in a 10-fold cross-validation setup such as done by Kinsora et al. (2017). As the researchers did not specify which hyperparameters settings were used during the feature selection procedure, we assume that these hyperparameters were kept to their default settings. However, Kinsora et al. (2017) conducted the recursive feature elimination in R while in this thesis python in combination with scikit-learn package is used. The hyperparameters of the random forest classifier in R and scikit-learn do not fully correspond with each other which complicates the ability to fully replicate the default hyperparameter settings. A hyperparameter of the random forest classifier that actually does correspond and can have a great impact on the performance is the `n_estimator` parameter, which in R is called the `n_tree` parameter. The `n_estimator` hyperparameter is therefore set to 500 which is the default value of the `n_tree` parameter. The `random_state` parameter is set to 42. The rest of the parameters of the random forest classifier are kept on their default values.

As the research of Kinsora et al. (2017) does not state which particular set of features is used as an input for the feature selection procedure, we apply RFECV to all the useful numerical features in their dataset. Before RFECV is executed, all the non-numerical features are therefore removed from the dataset. In addition to the non-numerical features, the Last_misinfo_dist_score feature and the Running_misinfo_count feature are removed from the dataset to prevent data leakage. Data leakage occurs when classifiers are trained on data that is not available at the moment of classification. Data leakage causes classification models to be evaluated on data that these models have already seen which results in inaccurate performances. In this case the two removed features are directly related to the target attribute. Both features after all represent information about misinformative posts. The values of these features can not be available at the moment of classification as posts yet have to classified as misinformative or non-misinformative. The two features should therefore not be included in dataset. Besides these two features, the id feature is removed from the dataset as this feature represents index values and is therefore not useful.

The mean predictive performance of the classification models for each number of features during RFECV is assessed based on the average accuracy. In section 4.2.1 it is stated that in the initial research the feature importances are represented by two different measures. As the `feature_importances_` attribute of the random forest classifier in the scikit-learn package is only measured with the Gini importance, only this measure is used for expressing the importances of the features in this replication.

Comparison of the predictive performances

Besides validating if the same important set of features as Kinsora et al. (2017) can be obtained following their feature selection approach, it is essential to ascertain if the same predictive performance can be achieved using their findings and the findings of the replication in this thesis. Therefore, in this replication study a comparison process is developed and implemented, which provides in a single overview insight into the differences of the predictive performances based on these different findings. In this process various classification models are built

based on the following feature sets:

- All the useful numerical features in the dataset
- The most important set of features according to Kinsora et al. (2017)
- The most important set of features obtained in this replication study

The performances of the classifiers in the research of Kinsora et al. (2017) were obtained during the feature selection procedure which was implemented in a cross-validation setup. In order to remain as close as possible to this approach and be able to make a fair comparison, the classification models in the comparison process are built with the use of the random forest classifier and 10-fold cross-validation.

The values of the hyperparameters of the random forest classifier, except for the value of the `random_state` parameter, are set to the same values as in the replication of the feature selection approach. The performance of the classifier varies with the value of the `random_state` parameter. In order to obtain a general overall performance of the classifier for each set of features, the values of the `random_state` parameter are iterated over a range from 0 to 100. Therefore, for each of these values a predictive model is built for all three feature sets.

To assess the performance of the predictive models, the average accuracy is used per random state for each feature set. For each feature set also the average accuracy is calculated over all random seeds.

4.3 Comparing feature sets

4.3.1 Text preprocessing

Classification algorithms cannot handle text directly. In order to build the classification models based on forum posts, these posts have to be converted into a usable format. The forum posts in this thesis are represented with the use of different (numerical) feature sets as described in section 4.1. These posts are retrieved from the Comment_text feature column in the dataset.

Extraction of Bag-of-words features

In order to apply classification algorithms to text, text documents are transformed into vector representations which is called vectorization (Bengfort, Bilbro, & Ojeda, 2018). In practice, the set of numerical vectors is represented as a matrix in which each row corresponds to a text document and each column corresponds to a feature. One of the most common approaches to vectorize text is the bag-of-words approach. As evident by its name, the bag-of-words approach represents each text document as a set (a bag) consisting of words. Therefore, the text documents are split into words and each unique word in the text documents represents a feature.

To apply the bag-of-words approach and extract the bag-of-words features, the `CountVectorizer` function of the

scikit-learn package is used. The `min_df` parameter is set to 2. This parameter indicates the minimum number of text documents a word has to occur in to become a feature in the text-document matrix. The other parameters of the `CountVectorizer` are kept to their default values. The `CountVectorizer` converts each character in the posts to lowercase before it transforms these forum posts into a term-document matrix in which the columns correspond to the unique words in all the forum posts and the rows correspond to these posts. Each unique word represents a word-count feature which indicates the number of times that this word occurs in each forum posts.

The countvectorizer features do not take the length of each document and the occurrence of words in the entire collection of text documents into consideration (Sarkar, 2016). Words in longer text documents tend to occur more frequently than words in shorter text documents. Besides, words that occur in many text documents are likely to dominate other words that occur in less text documents although the words that occur in many text documents are less informative. In order to address these limitations, the term frequency–inverse document frequency, which is abbreviated as tf-idf, is often used. Therefore, besides extracting countvectorizer features, tf-idf features are extracted. The tf-idf normalizes the number of occurrences of a word and indicates the importance of this particular word in a text document considering the entire collection of text documents.

The tf-idf features in this thesis are extracted with the use of the `TfidfTransformer` function of the scikit-learn package in combination with the `CountVectorizer`. The `TfidfTransfomer` transforms the earlier obtained term-document matrix into a tf-idf representation. For all features of each forum post the tf-idf score is calculated. So the earlier obtained unique words (the features) in the matrix remain unchanged, only the values of the features are adapted. The parameters of the `TfidfTransformer` are kept to their default values.

The tf-idf is the product of the term frequency (*tf*) and the inverse document frequency (*idf*) (Sarkar, 2016). The term frequency is the number of occurrences of a term t in a particular text document. The inverse document frequency is the logarithmically scaled division of the total number of text documents (n) by the document frequency (df), which is the number of text documents that contain the term t. In order to prevent zero divisions the `TfidfTransformer` adds a '1' to the numerator and denominator of this division when the default settings are used. Besides, in scikit-learn '1' is added to the inverse document frequency to prevent that terms with a zero tf-idf are ignored. Thus, the `TfidfTransformer` calculates the tf-idf for a term t in a particular document d with the default settings as follows:

$$tf\text{-}idf = tf(t,d) \times idf(t), \tag{4.1}$$

where

$$tf(t,d) = \ number\ of\ occurences\ of\ term\ t\ in\ document\ d$$

$$idf(t) = \log \frac{(1+n)}{(1+df(t))} + 1$$

Extraction of style-based features

Besides extracting bag-of-words features, style-based features are extracted. As evident by their name, style-based features capture the style of the forum posts. These features are extracted with the use of the basic functionalities in Python and the NLTK package. The extracted style-based features with their descriptions are shown in Table 4.1

Table 4.1: Style-based features overview

Feature	Description
Word count	The number of words in a forum post
Sentence count	The number of sentences in a forum post
Avg sentence length	The average length of a sentence denoted in the number of characters
Avg words per sentence	The average number of words per sentence in a forum post
Avg word length	The average word length in a forum post denoted in characters
Uppercase word ratio	The proportion of words that begin with a capital letter of the total number of words
Spelling mistake ratio	The proportion of misspelled words of the total number of words
URL count	The number of URLs in a forum post
Uppercase ratio	The proportion of capital letters of the total number of capital letters
Question mark ratio	The proportion of question marks of the total number of characters
Exclamation mark ratio	The proportion of the number of exclamation marks of the total number of characters
Fullstop ratio	The proportion of the number of full stops of the total number of characters
Asterisk ratio	The proportion of the number of asterisks of the total number of characters

As forum posts resemble tweets in terms of content, the `TweetTokenizer` from the NLTK package is used for the extraction of the Word count feature. The `TweetTokenizer` splits the forum posts into tokens after which the obtained tokens are counted with a simple counter. The tokens obtained with this tokenizer are considered as words except for punctuation marks. A similar approach is used to extract the Sentence count feature. To split the forum posts into sentences, however, the `sent_tokenize` function in the NLTK package is used.

For the extraction of the Spelling mistake ratio feature, the `Spellchecker` function in the Pyspellchecker package is used to find which words in the forum posts are spelled incorrectly. The `Spellchecker` however identifies some words incorrectly as misspelled, such as contractions. To minimize this limitation, the contractions in the forum posts are expanded with the use of the Contractions package. Besides, the language dictionary of the Pyspellchecker package is expanded with text-based emoticons, a build-in dictionary in UNIX[1] and a word list of medical terms[2]. Furthermore, URLs and numbers in combination with some punctuation marks, such as '2019-2020' and '04/07', are filtered out of the identified misspelled words.

All the other features are extracted with basic functionalities in Python.

Handling class imbalance

Class imbalance refers to the case where the values of the class attribute in the dataset are unequally distributed. In the case of the class imbalance of a binary classification problem, the number of instances corresponding

[1]https://manpages.ubuntu.com/manpages/bionic/en/man5/british-english.5.html
[2]https://github.com/glutanimate/wordlist-medicalterms-en

to a certain class is much higher (majority class) than the number of instances corresponding to another class (minority class). In the case of class imbalance, the classification algorithms do not capture the characteristics of the instances belonging to the minority class sufficiently enough during training, which makes it harder to predict this underrepresented class. The algorithms are thus biased towards the majority class. This is problematic as in most cases the minority class has to be predicted and is thus of more interest.

As described in section 3.2, the dataset used in this thesis is imbalanced. In order to deal with class imbalance two approaches are used: undersampling and SMOTE. Undersampling is an approach in which the instances of the majority class are removed randomly such that the number of instances corresponding to this class becomes equal to the number of instances corresponding to the minority class (Padurariu & Breaban, 2019). In order to apply undersampling the `RandomUnderSampler` function in the imbalanced-learning package is used. The `random_state` parameter is set to 42, while the other parameters are kept to their default values.

SMOTE is an oversampling approach with which synthetic instances for the minority class are created (Padurariu & Breaban, 2019). For each instance of the minority class its k-nearest neighbors are identified after which some of these neighbors are randomly selected based on the required amount of oversampling. The new synthetic instances are generated between the current instance and the nearest neighbors. This can be visualised as generating these new instances somewhere along drawn lines between the current instance and the nearest neighbors. In order to implement and perform SMOTE in this thesis, the `SMOTE` function from the imbalanced-learning package is used. The `random_state` parameter is set to 42. The other parameters of this function are kept to their default values. In this implementation SMOTE is performed with 5-nearest neighbors.

4.3.2 The classification algorithms

In order to select which classification algorithms to use for the classification task in this thesis, typical approaches used in the field of text classification and previous work on the automatic detection of misinformation are considered. The following classification algorithms are used in this thesis:

- **Multinomial naive Bayes:** A simple probabilistic classification approach based on the Bayes theorem (Vanderplas, 2016). The Bayes theorem is used to calculate the probability of each class given the values of the features of a particular instance. The class with the highest conditional probability value determines the classification decision of that instance.
- **Support vector machine:** A binary classification approach which tries to find the most optimal hyperplane, a particular decision boundary, separating the instances (data points) in a space (Han, Kamber, & Pei, 2011). The most optimal hyperplane is the hyperplane which has the largest distance between the hyperplane and the nearest instances to the hyperplane (support vectors). The margin (the gap) between the classes is therefore maximized which leads to more reliable classifications. In the case of two features, the data points

are plotted in a 2D space in which the hyperplane is just a straight line separating the data points.

- **Logistic regression:** A binary probabilistic classification approach which fits the data to a logistic function. This logistic function is an s-shaped curve ranging from 0 to 1 (Sluijmers, 2020). The logistic regression algorithm gives the probability that a particular instance belongs to class '1'. The actual classification decision is made based on a decision threshold. An instance with a probability value below the decision threshold is classified as class '0' and an instance with a higher probability value is classified as class '1'.

- **Multilayer perceptron:** A feed-forward artificial neural network (ANN) which is a mathematical model based on the structure of the human brain (Hallinan, 2013). A multilayer perceptron (MLP) has three layers: the input layer, the hidden layer and the output layer. The layers consist of nodes, representing neurons, which are connected with each other between these layers. The output of particular node can serve as the input of other nodes. The input of a node is the weighted sum of its inputs. From this weighted sum the output value of that node is calculated using a non-linear function. In the case of a node in the output layer, this output value is the final output.

- **Random forest:** An ensemble approach which combines several decision trees (Han et al., 2011). An ensemble approach combines various classification models in order to obtain a more accurate classification performance than any of these models individually. The decision trees are trained on random subsets (of equal size) which is called bagging. During the training of the decision trees also the nodes are splitted based on of random subset of features are selected. The classification decisions are made based on a majority vote between the individual decision trees.

4.3.3 Training and evaluating the classification algorithms

In order to obtain an accurate indication of the generalizability and the overall performance of the classification algorithms, cross-validation is used. A commonly used form of cross-validation is k-fold cross-validation. With k-fold cross-validation the data is split into k equal subsets, which are called folds. The classification algorithm is then trained on k-1 folds and tested on the remaining fold. This procedure is repeated k times such that the classification algorithm is tested on each of the k folds once. The classifier is in each iteration thus trained and tested on different parts of the dataset. Training the classifier in each iteration on different training folds results in k classification models.

As the forum posts in the dataset are divided into threads, group 10-fold cross-validation is used. In contrast to k-fold cross-validation, group k-fold cross-validation considers pre-defined groups when splitting the dataset into folds, such that instances of these groups (threads in this case) remain together in the same folds. As instances of the same groups are not split over different folds, it is prevented that data of the test folds leaks into the training folds. Data leakage is thus prevented.

To employ 10-fold group cross-valdiation the `GroupKfold` function in the scikit-learn package is used. The dataset used in this thesis consists of 78 threads which are not of equal size. In the 10-fold cross-validation process the threads are divided into 10 equal sized folds. Each fold consists thus of an equal amount of forum posts of different threads.

4.3.4 Tuning the hyperparameters

After the overall performances of the classification algorithms are obtained with the use of 10-fold group cross validation, these classification performances are compared. The hyperparameters of the best classification algorithm for each feature set are tuned with the use of nested cross-validation. Nested cross-validation is an optimization approach which consists of two nested cross-validation loops: the outer and inner cross-validation loop. The inner cross-validation is used for hyperparameter optimization while the outer cross-validation is used to obtain an overall performance of the tuned classification models.

The outer cross-validation loop splits the dataset into k folds after which cross-validation is applied in the inner-loop on all the outer training folds (k-1 folds) for every possible combination of hyperparameter settings, in order to identify the best settings. In such an inner cross-validation process the outer training folds are split again into l inner folds. A classification model with particular hyperparameter settings is then trained and evaluated on an inner test fold which serves as a validation fold. The cross-validation for that particular combination of hyperparameter settings ends when each fold is used once as validation fold. After all the cross validations are performed for every combination of hyperparameters settings, the most optimal classification model (with the best hyperparameter settings) from all of these inner cross-validation processes is selected and trained on all the outer training folds. This model is then evaluated on an outer test fold. This process is repeated until the hyperparameters for each classification model of the outer loop are tuned.

The outer cross-validation loop of the nested-cross validation process in this thesis is conducted with 10-fold group cross-valdiation which is implemented with the `GroupKfold` function of the scikit-learn package. The inner-loop is conducted with the `GridSearchCV` function of this package. The `GridSearchCV` function performs an exhaustive search in a cross-validation setup to find the best hyperparameters of a classification algorithm from a hyperparameter grid. The `scoring` parameter of the `GridSearchCV` function is set to F_1 which results in the optimization of the hyperparameters for the F_1 score corresponding to the misinformative class. Besides, the `CV` parameter is configured such that `GridSearchCV` conducts 5-fold group cross-validation.

4.3.5 The performance metrics

In order to denote and assess the predictive performance of the built classification models, the following performance metrics (Jurafsky & Martin, 2019) are used:

- **Accuracy:** The proportion of the correctly classified instances of the total number of predictions. The accuracy can be calculated as follows:

$$\text{Accuracy} = \frac{\text{Correctly classified instances}}{\text{Total number of predictions}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{4.2}$$

  As cross-validation is used in this thesis, the average accuracy over all the different test folds is calculated. When dealing with an imbalanced dataset, accuracy can be a misleading performance metric (Jurafsky & Martin, 2019). Imagine a dataset which consists of 100 instances of which 90 instances belong to class A and 10 instances to class B. The latter is the class of interest. A classification model which classifies each instances as class A would achieve an accuracy of 90%. This classifier seems good however the classifier is useless as it is not able to detect class B. Besides the accuracy, it is therefore important to consider other performance metrics. Nevertheless, this metric is included in this thesis for completeness.

- **Precision:** The proportion of correctly classified instances as positive of the total number of instances classified as positive. The precision can be calculated as follows:

$$\text{Precision} = \frac{\text{Correct positive classifications}}{\text{Total number of positive classifications}} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4.3}$$

- **Recall:** The proportion of correctly classified instances as positive of the total number of actual positive instances. The recall can be calculated as follows:

$$\text{Recall} = \frac{\text{Correct positive classifications}}{\text{Total number of actual positive instances}} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.4}$$

- **F_1 score:** The harmonic mean of the precision and recall which can be defined as:

$$F_1 \text{ score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{2\text{PR}}{\text{P} + \text{R}} \tag{4.5}$$

These performance metrics are obtained with the use of the `accuracy_score` and `classification_report` functions in the scikit-learn package. The precision, recall and F_1 score are given for both classes. As this thesis focuses on the identification of medical misinformation, we consider the F_1 score for the misinformative class as the most important performance metric. However, in some cases the precison and recall for the misinformative class have to be taken into account when evaluating the performance of a classification algorithm. After all, some classification algorithms attain a high F_1 score while they either attain a low precision or low recall. These algorithms might not be useful for our classification task.

Chapter 5

Results

5.1 Replication study

Kinsora et al. (2017) found nine features with their feature selection approach as being the most distinctive for classifying medical misinformation on health forums. Besides the Word count (WC) feature, these features can be divided into three groups: features referring to person pronouns, authenticity and authority, and health. The identified features are shown with their descriptions (Pennebaker, Boyd, et al., 2015; Pennebaker, Booth, et al., 2015) in Table 5.1. The features in this table are ordered based on their feature importances except for the Bio feature. The feature importance of this feature is not denoted in the research of Kinsora et al. (2017) and therefore it is placed last in the table.

Table 5.1: Most distinctive set of features according to Kinsora et al. (2017). These LIWC features are ordered in the table based on their feature importance denoted in mean decrease in Gini except for the Bio feature. This feature is placed last in this order as the researchers did not denote its feature importance.

Feature	Description
WC	The number of words in a forum post
Dic	The percentage words occuring in the LIWC dictonary of the total number of words
Clout	A summary variable of LIWC2015 referring to social status, confidence or leadership denoted in a percentile score. The higher the score, the more a post is written with confidence and from an expertise perspective.
I	The percentage singular first-person pronouns of the total number of words
Authentic	A summary variable of LIWC2015 referring to authenticity of a post denoted in a percentile score. The higher the score, the more honest and personal a post is.
Ppron	The percentage personal pronouns of the total number of words
Health	The percentage words belonging to the category 'health' of the total number of words
You	The percentage second-person pronouns of the total number of words
Bio	The percentage words belonging to the category 'biological processes' of the total number of words

With these features the researchers obtained an average accuracy of 90.1% during the feature selection process. The highest accuracy, at 90.31%, during this process was achieved with 52 features. Although these features gave a higher accuracy than the set of nine features, the researchers stated that these nine features provides more insight into which textual characterises are significant for the classification of medical misinformation in health forums.

By replicating the feature selection approach of Kinsora et al. (2017) as closely as possible, we identify 63 features as being the most distinctive for classifying medical misinformation. Figure 5.1 depicts how the average accuracy varies with the number of features during this selection approach. The figure shows a rapid, absolute increase of approximately 7% in average accuracy as the number of features increases from one to nine. Subsequently the average accuracy increases with approximately more than 1% up to feature sets with 30 features. From that point on the average accuracy does not increase much further and remains fluctuating around 86%. With the most important 63 features we achieve an average accuracy of 86.3% as can be seen in Figure 5.1. A nearly similar classification performance can be achieved with a feature set of (approximately) 48 features. To be able to compare

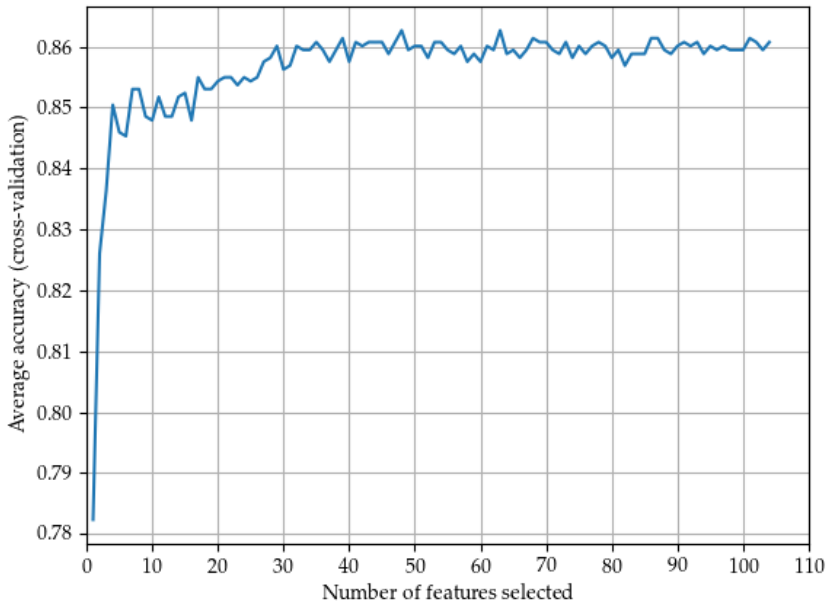

Figure 5.1: The average accuracy per number of features during RFECV with the use of a random forest classifier in a 10-fold cross-validation setup as was done by Kinsora et al. (2017). The set of features which yields the highest average accuracy is defined as the most optimal set for the identification of medical misinformation. During the replication study the highest accuracy at 86.3% is achieved with a set of 63 features.

our most optimal feature set with the found feature set of Kinsora et al. (2017), we select the nine most important features from the set of 63 features. In Figure 5.2 these nine features with their feature importances are shown. As the obtained feature importances are scaled differently than in the research of Kinsora et al. (2017), we only consider the order of importance (based on Gini importance) of the features when comparing both optimal feature sets.

In Figure 5.2 we can observe that several features occur in both of these feature sets. This is the case for the I, Authentic, WC, Dic, Clout and the Ppron features. Nevertheless, the order of importance of these six features is different in both feature sets. As can be seen in Figure 5.2 the length of the posts (comment_length) comes up as the most important feature in contrast to the total word count (WC), which was identified by Kinsora et al. (2017) as the most important one. In our replication the WC feature comes up as the fourth most important feature.

Besides the WC feature, the Dic and Clout features are ranked lower in our feature importance ranking than in the importance ranking of Kinsora et al. (2017) in which these features are respectively the second and the third most important features. In our feature ranking, however, the I and Authentic features comes up as second and third most important features which are slightly higher positions than their position in the ranking of Kinsora et al. (2017). The Health, You and Bio features can not be found in our set of nine most important features, but do come up in the full set of important features. All 63 most distinctive features with their feature importances can be found in appendix A.

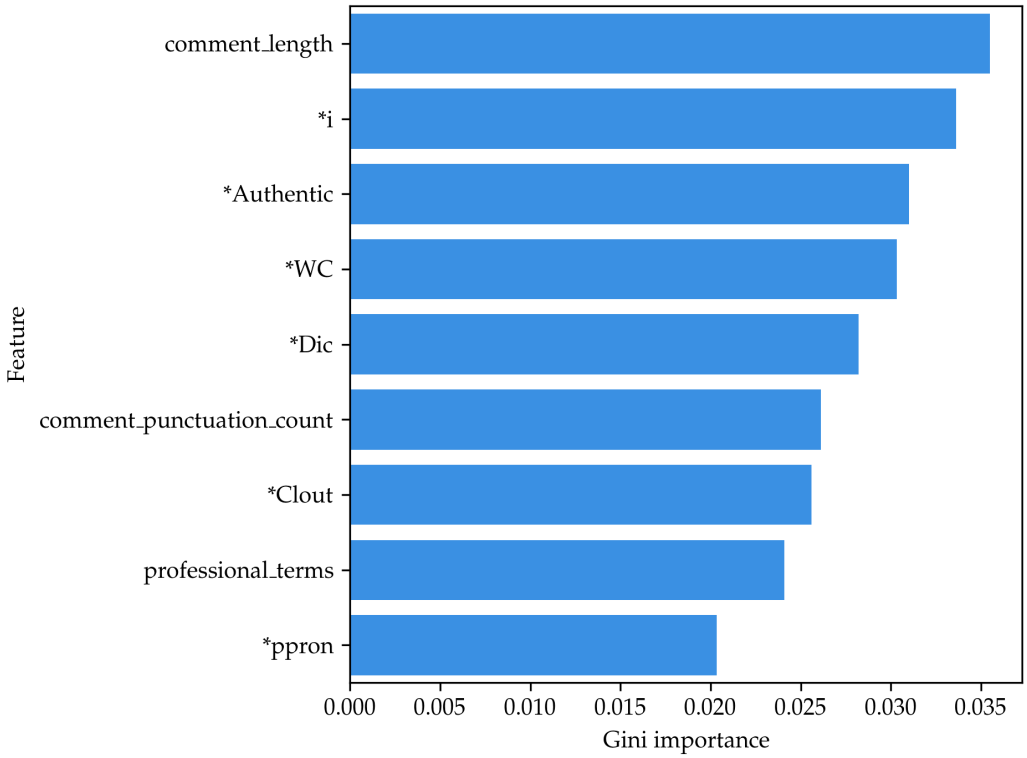

Figure 5.2: Top nine most important features with their feature importances denoted in mean decrease in Gini from the optimal set of 63 features. The features which correspond with the features in the optimal feature set of Kinsora et al. (2017) are marked with '*'. The comment_length comes up as the most important feature of the dataset based on this measure during RFECV.

As described in section 4.2.2, we developed and implemented a comparison process which provides an overview of the differences in classification performances when using the optimal feature sets and all the useful numerical features in the dataset for the detection of medical misinformation. In order to build the classification models we used a random forest classifier and 10-fold cross-validation. When building these models for each feature set we changed the value of the `random_state` parameter by iterating over random seeds ranging from 0 to 100. Table 5.2 shows the accuracy for each feature set averaged over all random seeds. From this table, we notice that the set with all useful numerical features and our optimal feature set obtain a similar average accuracy of 86.1%.

Table 5.2: Classification performance for each feature set obtained during the comparison process with the use of a random forest classifier and 10-fold cross-validation. For each feature set various classification models were build based on random seeds ranging from 0 to 100. The classification results are denoted in accuracy averaged over all random seeds.

Feature set	Average accuracy
All useful numerical features	86.1%
Most important feature set according to Kinsora et al. (2017)	85.5%
Most important feature set obtained during replication	86.1%

Figure 5.3 illustrates the dispersion of the average accuracies for each feature set obtained with the comparison process. We observe that the feature sets achieve a nearly similar performance in average accuracy for all different values of the `random_state` parameter. The average accuracies of all the feature sets fall within a small range from approximately 85.1% to 86.6%. We notice that the most optimal feature set of Kinsora et al. (2017) has the largest dispersion. Remarkably, we notice from Figure 5.3 that the median of the average accuracies obtained with the set of all numerical useful features is equal to the first quartile. This implies that the average accuracies remain approximately equal between the first quartile and the median. From our comparison process we find that we cannot obtain an average accuracy nearly equal to the average accuracy obtained by Kinsora et al. (2017) with any of the three feature sets regardless of the value of the `random_state` parameter.

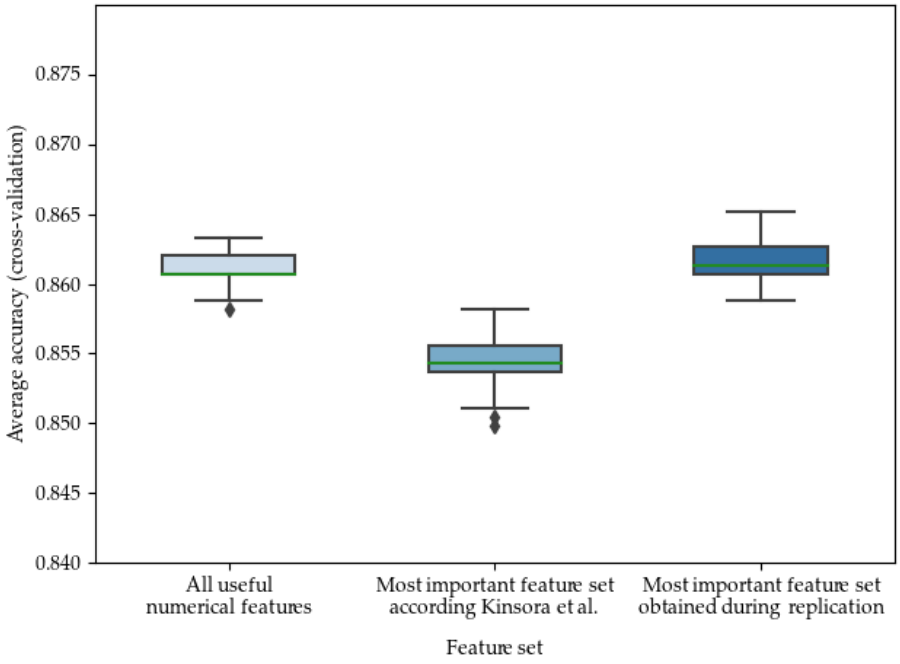

Figure 5.3: The dispersion of the average accuracies for each feature set obtained with random states in the range from 0 to 100 during the comparison process. Regardless of the random state values all feature sets obtain a nearly equal performance. The optimal feature set of Kinsora et al. (2017) has the largest dispersion in average accuracy. The median of the average accuracies achieved with the set of all useful numerical features is identical to the first quartile.

5.2 Comparing feature sets

5.2.1 Classification results with SMOTE

Besides replicating the feature selection approach of Kinsora et al. (2017), we explore the classification performances of different feature sets with the use of several classification algorithms. Table 5.3 represents the classification performances obtained with these feature sets for each classification algorithm when the class imbalance is handled with SMOTE. We compare the classification results of the algorithms on each feature set in order to find the best performing algorithm on each set.

Table 5.3: Classification results obtained with different classification algorithms and feature sets when the class imbalance is handled with SMOTE. For each class the precision, recall and the F_1 score are represented. These three performance metrics are calculated over all test folds. The best classification algorithm for each feature set is highlighted. The highest score for each metric is marked in bold.

			Non-misinformative			Misinformative		
Classification model	**Feature set**	**Avg accuracy**	**Precision**	**Recall**	**F_1 score**	**Precision**	**Recall**	**F_1 score**
Multinomial naive Bayes	Countvectorizer features	80.3%	0.887	0.882	0.885	0.350	0.361	0.355
	Tf-idf features	72.6%	**0.925**	0.741	0.823	0.308	**0.657**	0.420
	Style-based features	60.4%	0.876	0.623	0.728	0.188	0.498	0.273
Support vector machine	Countvectorizer features	69.8%	0.861	0.771	0.814	0.183	0.292	0.225
	Tf-idf features	86.4%	0.866	**0.993**	**0.925**	**0.769**	0.129	0.221
	Style-based features	67.4%	0.884	0.709	0.787	0.222	0.472	0.302
Logistic regression	Countvectorizer features	74.6%	0.886	0.805	0.843	0.270	0.412	0.327
	Tf-idf features	82.7%	0.903	0.894	0.898	0.429	0.451	**0.439**
	Style-based features	71.9%	0.902	0.753	0.821	0.277	0.536	0.365
Multilayer perceptron	Countvectorizer features	75.3%	0.879	0.822	0.849	0.260	0.356	0.301
	Tf-idf features	84.7%	0.890	0.937	0.912	0.485	0.339	0.399
	Style-based features	70.4 %	0.887	0.752	0.814	0.244	0.455	0.317
Random forest	Countvectorizer features	79.1%	0.866	0.894	0.880	0.259	0.210	0.232
	Tf-idf features	**85.2%**	0.868	0.976	0.919	0.529	0.155	0.239
	Style-based features	76.6%	0.871	0.852	0.862	0.252	0.283	0.267

From Table 5.3, we observe that the multinomial naive Bayes classifier achieves the best performance when comparing the performances of the set of countvectorizer features for each algorithm. The multinomial naive Bayes classifier reaches the highest accuracy of 80.3% and the highest F_1 scores of 0.885 and 0.355 corresponding respectively to the non-misinformative and misinformative classes.

On the set of tf-idf feature set, we observe based on the F_1 score that the logistic regression and the support vector machine classifier perform best for respectively the misinformative and non-misinformative class. Although the support vector machine classifier performs best for the non-misinformative class, we observe that the classifier performances poorly for the misinformative class on this feature set (and the other feature sets) as it attains the lowest F_1 score of 0.22. Remarkably, the support vector machine classifier attains a high precision (0.769) for this class. The low F_1 score is thus a result of the low recall (0.129) the classifier obtains for the misinformative class. This indicates that a significant proportion of all the instances identified as misinformative are really correct, but the classifier identifies only a small part of all the misinformative posts. The classifier is thus not able to detect many of these posts.

As the logistic regression and the support vector machine classifier perform best on either one of the classes, we compare both algorithms with each other. We notice that the F_1 score of the logistic classifier for the non-misinformative class is slightly lower than that of the support vector machine classifier, while the support vector machine classifier performs much worse for the misinformative class than the logistic regression classifier. As the misinformative class is of most interest, this slightly lower F_1 score for the non-misinformative class obtained with the logistic regression classifier is acceptable compared to the bad performance of the support vector machine for the misinformative class. We conclude that the logistic regression classifier is thus the best classifier on the tf-idf feature set (when handling the class imbalance with SMOTE) as it performs best on the misinformative class and only slightly worse than the best classifier on the non-misinformative class, the support vector machine classifier.

Considering the classification results on the style-based features for each classifier, we observe that such as on the tf-idf feature set, the highest F_1 score of 0.365 corresponding to the misinformative class is achieved with the logistic regression classifier. We notice that the logistic regression classifier achieves for this class a precision (0.277) which is more than two times lower than the recall (0.536). This implies that the classifier identifies more than half of all misinformative posts, but a significant proportion of all instances classified as misinformative are incorrect. The random forest classifier obtains the highest F_1 score of 0.862 corresponding to the non-misinformative class. However, the random forest classifier attains the lowest F_1 score on the style-based feature set for the misinformative class. Interestingly, the random forest classifier is the only classifier that obtains a precision (0.252) and recall (0.283) for this class which are approximately equal to each other.

We compare the best classifiers for both the non-misinformative and misinformative classes on the style-based feature set in case that the class imbalance is handled with SMOTE. For the non-misinformative class, the logistic regression classifier has a lower F_1 score than the random forest classifier. However, for the misinformative class, the logistic regression classifier performs substantially better than the random forest classifier. Nonetheless, we notice that both of their precisions are relatively low and approximately equal, indicating that a significant proportion of all forum posts classified as misinformative are incorrect with the use of both classifiers. However, the logistic regression classifier achieves a much higher recall, which results in a higher F_1 score. Following a similar argument as for the tf-idf feature set and the fact that the recall of the logistic regression classifier is much higher (while their precisions are approximately equal), we conclude that the logistic regression classifier is the best classifier on the style-based feature sets when the class imbalance is handled with SMOTE.

To summarize, we found that the multinomial naive Bayes classifier performs best on countvectorizer features when the class imbalance is handled with SMOTE. The logistic regression classifier attains the best performance on both the tf-idf and style-based features. The classification results of these best performing algorithms are highlighted in Table 5.3.

5.2.2 Classification results with undersampling

Besides handling the class imbalance with SMOTE, we applied random undersampling to deal with this class imbalance. Table 5.4 represents the classification results obtained with the use of the different feature sets for each classifier when the class imbalance is handled with random undersampling. We consider these classification results for each feature set as is done in section 5.2.1.

Table 5.4: Classification results obtained with different classification algorithms and feature sets when the class imbalance is handled with random undersampling. For each class the precision, recall and the F_1 score are represented. These three performance metrics are calculated over all test folds. The best classification algorithm for each feature set is highlighted. The highest score for each metric is marked in bold.

			Non-misinformative			Misinformative		
Classification model	**Feature set**	**Avg accuracy**	**Precision**	**Recall**	**F_1 score**	**Precision**	**Recall**	**F_1 score**
Multinomial naive Bayes	Countvectorizer features	67.9%	0.934	0.673	0.782	0.282	0.730	0.407
	Tf-idf features	60.1%	**0.964**	0.566	0.713	0.263	**0.880**	0.405
	Style-based features	59.6%	0.882	0.607	0.719	0.193	0.536	0.284
Support vector machine	Countvectorizer features	**76.9%**	0.899	**0.822**	**0.859**	0.320	0.476	0.383
	Tf-idf features	73.4%	0.932	0.744	0.828	0.322	0.691	**0.439**
	Style-based features	66.0%	0.875	0.700	0.778	0.203	0.433	0.276
Logistic regression	Countvectorizer features	73.2%	0.924	0.749	0.827	0.312	0.648	0.421
	Tf-idf features	72.1%	0.935	0.725	0.817	0.313	0.712	0.435
	Style-based features	72.4%	0.900	0.762	0.826	0.278	0.519	0.362
Multilayer perceptron	Countvectorizer features	73.0%	0.928	0.741	0.824	0.314	0.674	0.428
	Tf-idf features	67.8%	0.952	0.655	0.776	0.293	0.811	0.430
	Style-based features	65.0%	0.883	0.679	0.768	0.212	0.489	0.295
Random forest	Countvectorizer features	74.4%	0.915	0.773	0.838	0.314	0.592	0.411
	Tf-idf features	75.2%	0.921	0.778	0.843	**0.329**	0.622	0.430
	Style-based features	65.0%	0.892	0.673	0.767	0.224	0.536	0.316

When comparing the performances of the countvectorizer features for each classifier, we observe that the multilayer perceptron classifier identifies the misinformative class best as it obtains the highest F_1 score of 0.428. The highest F_1 score of 0.859 for the non-misinformative class is achieved with the support vector machine classifier. Interestingly, we notice from Table 5.3 that the multilayer perceptron classifier attains only a slightly lower F_1 score (0.824). However, we observe that the recall for the non-misinformative class obtained with the multilayer perceptron classifier is higher than that of the support vector machine classifier. This indicates that with the use of the multilayer perceptron classifier a larger proportion of all forum posts identified as non-misinformative are correct than with the use of the support vector machine classifier. Although the support vector machine classifier performs best for the non-misinformative class, the classifier performs worst for the misinformative class as it obtains the lowest F_1 score of 0.383. Remarkably, we notice however that the support vector classifier attains a nearly similar precision for the misinformative class as the best classifier, the multilayer perceptron classifier. This implies that the proportions of all forum posts identified as misinformative which are correct are approximately equal for both classifiers.

As the multilayer perceptron classifier performs slightly worse for the non-misinformative class than the best support vector machine classifier, while the support vector machine performs worst for the misinformative class, we conclude that the multilayer perceptron classifier is the best classifier for the countvectorizer feature set (when the class imbalance is handled with random undersampling), since the misinformative class is of most importance.

The slightly better performance of the support vector machine classifier for the non-misinformative compared to the multilayer perceptron classifier does not outweigh the bad performance for the misinformative class. Similar findings and implications were made in section 5.2.1.

Considering the set of tf-idf features, we notice that the support vector machine, logistic regression, multilayer perceptron and the random forest classifier achieve an approximately equal F_1 score for the misinformative class. The support vector machine classifier achieves the highest F_1 score of 0.439 and thus performs best on this class followed by logistic regression classifier which achieves a slightly lower F_1 score of 0.435. The multilayer perceptron and the random forest classifier both attain a F_1 score of 0.430 for the misinformative class. However, we notice that their precision and recall for this class differ from each other. The multilayer perceptron obtains a lower precision than the random forest classifier implying that with the use of multilayer perceptron a larger proportion of the forum posts classified as misinformative are incorrect. The multilayer perceptron classifier however attains a higher recall than the random forest classifier indicating that multilayer perceptron classifier identifies more misinformative posts correctly. The random forest classifier performs best on the non-misinformative class with a F_1 score of 0.843. Following a similar argument as for the set of countvectorizer features, we consider the multilayer perceptron as the best classifier on the tf-idf feature set when random undersampling is used to deal with the class imbalance.

For the set of style-based features, we observe that the logistic regression classifier performs best on both classes based on the F_1 score and outperforms thus the other classifiers.

In summary, we found that the multilayer perceptron classifier achieves the best performance on the countvectorizer features when the class imbalance is handled with random undersampling. The support vector machine classifier performs best on the set of tf-idf features. Finally, the logistic regression classifier is the best classifier on the style-based features. The classification results obtained with these best algorithms are highlighted in Table 5.4.

5.2.3 Selecting the best classification algorithms

In sections 5.2.1 and 5.2.2 we selected the best classification algorithms on each feature set for respectively SMOTE and random undersampling. The classification results of the selected algorithms are represented in Table 5.5. These classification algorithms are selected mainly based on their achieved results in the identification of medical misinformation. The results of the selected algorithms are compared in order to find the best classification algorithm on each feature set.

Table 5.5: Classification results obtained with the classification algorithms which are considered as the best for our classification task for each feature set and resampling method. These algorithms are mainly selected based on their achieved F_1 score corresponding to the misinformative class. The best classification algorithm for each feature set is highlighted. The highest score for each metric is marked in bold.

				Non-misinformative			Misinformative		
Feature set	**Resampling method**	**Classification model**	**Avg accuracy**	**Precision**	**Recall**	**F_1 score**	**Precision**	**Recall**	**F_1 score**
Countvectorizer features	SMOTE	Multinomial naive Bayes	80.3%	0.887	0.882	0.885	0.350	0.361	0.355
	Random undersampling	Multilayer perceptron	73,0%	0.928	0.741	0.824	0.314	0.674	0.428
Tf-idf features	SMOTE	Logistic regression	**82.7%**	0.903	**0.894**	**0.898**	**0.429**	0.451	**0.439**
	Random undersampling	Support vector machine	73.4%	**0.932**	0.744	0.828	0.322	**0.691**	0.439
Style-based features	SMOTE	Logistic regression	71.9%	0.902	0.753	0.821	0.277	0.536	0.365
	Random undersampling	Logistic regression	72.4%	0.900	0.762	0.826	0.278	0.519	0.362

Considering all the classification results in Table 5.5, we observe that the classifiers obtain higher scores for the non-misinformative class than for the misinformative class indicating that non-misinformative class is easier to predict. Furthermore, we notice that the classifiers obtain a higher recall for the misinformative class when the class imbalance is handled with random undersampling instead of SMOTE, except for the set of style-based features. However, the classifiers achieve a higher precision for the misinformative class when the class imbalance is handled with SMOTE, again except for the style-based feature set. Based on the F_1 scores, the best performances on both classes are obtained on the tf-idf feature set. With the use of this feature set, we obtain similar F_1 scores (0.439) corresponding to the misinformative class for both resampling methods.

When comparing the best classifiers on the set of countvectorizer features, we find based on the F_1 score that the multinomial naive Bayes classifier identifies the non-misinformative class best. The multilayer perceptron classifier, however, attains a higher F_1 score (0.428) for the misinformative class than the multinomial naive Bayes classifier. Nevertheless, the multinomial naive Bayes classifier attains a higher precision corresponding to this class than the multilayer perceptron classifier, as can be seen in Table 5.5. This indicates that a larger proportion of all forum posts classified as misinformative are incorrect with the use of the multilayer perceptron classifier than with the use of the multinomial naive Bayes. The multilayer perceptron classifier in turn attains a higher recall than the multinomial naive Bayes classifier.

The multilayer perceptron classifier obtains thus a high recall and low precision for the misinformative class indicating that the classifier identifies many misinformative forum posts correctly, while a large proportion of the forum posts classified as misinformative are incorrect. Although the multinomial naive Bayes classifier obtains a lower recall, the classifier is much more selective in identifying posts as misinformative which results in less posts that are classified falsely as misinformative. We believe that the multinomial naive Bayes classifier is more informative and useful for the detection of medical misinformation than the multilayer perceptron classifier, which identifies many forum posts easily as misinformative, while many of these posts are not. The higher precision of the multinomial naive Bayes classifier for the misinformative class does thus outweigh the higher recall of the multilayer perceptron classifier, and therefore we consider the multinomial naive Bayes classifier as the best classifier on the countvectorizer feature set for our classification task.

Considering the best classifiers on the tf-idf feature set, we observe that the highest F_1 score of 0.898 for the non-misinformative class is achieved with the logistic regression classifier. Furthermore, we notice that both classifiers obtain an equal F_1 score of 0.439 for the misinformative class. This is the highest F_1 score of all feature sets for this class. Although the classifiers obtain an equal F_1 score, their precision and recall differ from each other. The logistic regression classifier achieves a higher precision for the misinformative class than the support vector machine. The support vector machine however attains a much higher recall than the logistic regression classifier. Similar observations were made for the set of countvectorizer features. Following the same argument as for this feature set, we consider the logistic regression classifier as the best classifier on the tf-idf feature set.

We notice that on the style-based feature set the logistic regression classifier performs best for both resampling methods. When handling the class imbalance with SMOTE, we notice that we achieve a slightly lower F_1 score of 0.821 for the non-misinformative class than with random undersampling, with which we achieve a F_1 score of 0.826. However, with the use of SMOTE we obtain a slightly higher F_1 score of 0.365 for the misinformative class than with the use of random undersampling. We prefer the slightly higher F_1 score for the misinformative class obtained with SMOTE over the higher F_1 score for the non-misinformative class obtained with random undersampling. Therefore, we conclude that on the style-based features the logistic regression classifier performs best when the class imbalance handled with SMOTE.

In conclusion, we found that the multinomial naive Bayes classifier performs best on the set of countvectorizer features. The logistic regression classifier achieves the best performance on both the tf-idf and style-based features. All these classifiers were build on a dataset for which the class imbalance was handled with SMOTE. The logistic regression classifier on the tf-idf feature set performs best of all three classifiers for our classification task as it achieves the highest F_1 score of 0.439 for the misinformative class. The classifier also achieves the highest F_1 score corresponding to the non-misinformative class. The second highest F_1 score corresponding to the misinformative class is achieved with the logistic regression on the style-based features followed by the multinomial naive Bayes classifier which achieves the lowest F_1 score for this class. However, as the multinomial naive Bayes classifier on the countvectorizer features attains a higher precision than the logistic regression classifier on the style-based feature set, the multinomial naive Bayes classifier is much more selective in identifying posts as misinformative. Therefore, we conclude that the multinomial naive Bayes classifier is more useful and is considered as the second best classifier for our classification task. The worst performance of all the three algorithms is thus achieved with the logistic regression classifier on the style-based features.

5.2.4 Tuning the hyperparameters

With the use of nested cross-validation, we tuned the hyperparameter settings of the best classification algorithm for each feature set identified in section 5.2.3. The parameter values which are considered for each classification algorithm during this optimization process, the parameters grids, are shown in appendix B. The overall classification results obtained during the hyperparameter optimization of the classification algorithms are shown in Table 5.6.

Table 5.6: The overall classification results of the classification algorithms obtained during hyperparameter optimization of the F_1 score (corrresponding to the misinformative class) with the use of nested cross-validation. The outer loop of this nested cross-validation process is conducted with 10 fold group cross-validation while the inner loop is performed with 5 fold group cross-validation. The performance metrics are calculated over all inner test folds of the nested cross-validation process.

			Non-misinformative			Misinformative		
Classification model	**Feature set**	**Avg accuracy**	**Precision**	**Recall**	**F_1 score**	**Precision**	**Recall**	**F_1 score**
Multinomial naive Bayes	Countvectorizer features	80.3%	0.887	0.822	0.885	0.350	0.362	0.355
Logistic regression	Tf-idf features	83.6%	0.901	0.908	0.905	0.453	0.433	0.443
Logistic regression	Style-based features	70.5%	0.900	0.737	0.810	0.263	0.532	0.352

When tuning the hyperparameter settings for the multinomial naive Bayes classifier on the countvectorizer features, we find that for each model the value of the `alpha` parameter is '1'. This indicates that the multinomial naive Bayes classifier is stable on the set of countvectorizer features as the identified hyperparameter settings for each model do not change regardless of the change in training and test folds. We note that the value of '1' is the default value of the `alpha` parameter, and therefore the obtained classification results with nested cross-validation for this classifier are equal to the results obtained with single cross-validation, as can be seen in Table 5.6.

Table 5.7 shows the best hyperparameter settings obtained during the hyperparamater optimization for each logistic regression model on the tf-idf feature set of the outer cross-validation loop. From Table 5.7, we notice that the parameter optimization approach yields nearly equal hyperparameters settings for the logistic regression classifier accross the outer cross-validaton loops. For seven of the ten logistic regression models it is found that the values '1' and 'liblinear' are the optimal settings for respectively the `C` and `solver` parameters. The values '100' and 'saga' are found to be the most optimal values of the `C` and `solver`, respectively, for two of the remaining logistic regression models. Overall, we observe thus that the optimal hyperparameter settings do not differ much across the inner cross-validation loops and mostly occur in these two combinations. We therefore conclude that the logistic regression classifier is reasonably stable on the tf-idf feature set.

With nested cross-validation the logistic regression algorithm obtains on the tf-idf feature set an overall precision of 0.451, recall of 0.433 and a F_1 score of 0.443 for the misinformative class, as can be seen in Table 5.6. These precision and F_1 scores are slightly higher than those obtained with the single cross-validation process. The recall, however, is lower. This indicicates that with the use of nested cross-validation a larger proportion of the posts classified as misinformative are correct, while the algorithm identifies a smaller proportion of the misinformative posts.

Table 5.7: The best hyperparameters settings with corresponding F_1 score for each logistic regression model of the outer cross-validation loop on the tf-idf feature set obtained in the inner cross-validation loop with GridSearchCV. The F_1 score corresponding to particular hyperparameters settings of a classificication model is calculated over all the inner test folds of the corresponding cross-validation process in the inner cross-validation loop during GridSearchCV. This F_1 score belongs the misinformative class.

F_1 corresponding to the misinformative class (GridSearchCV)	Best hyperparameter settings
0.493	C: 1 , solver: liblinear
0.427	C: 1, solver: liblinear
0.448	C: 10, solver: liblinear
0.454	C: 1, solver: liblinear
0.458	C:100, solver: saga
0.445	C:1, solver: liblinear
0.464	C:100, solver: saga
0.451	C:1, solver: liblinear
0.444	C:1, solver: liblinear
0.468	C:1, solver: liblinear

Table 5.8 shows the found set of hyperparameter settings for each logistic regression model on the style-based feature set of the outer cross-validation loop. We notice from this table that the best hyperparameter settings of the logistic regression models on the style-based features differ much across the outer cross-validation loops.

Remarkably, we notice from the comparison of Tables 5.5 and 5.6 that the logistic regression classifier obtains a slightly worse overall performance for both classes with the tuned hyperparameter settings on both classes than with the default settings.

Table 5.8: The best hyperparameters settings with corresponding F_1 score for each logistic regression model of the outer cross-validation loop on the set of style-based features obtained in the inner cross-validation loop with GridSearchCV. The F_1 score corresponding to particular hyperparameters settings of a classificication model is calculated over all the inner test folds of the corresponding cross-validation process in the inner cross-validation loop during GridSearchCV. This F_1 score belongs the misinformative class.

F_1 corresponding to the misinformative class (GridSearchCV)	Best hyperparameter settings
0.353	C: 10 , solver: lbfgs
0.373	C: 0.001, solver: saga
0.377	C: 0.01, solver: liblinear
0.383	C: 0.001, solver: liblinear
0.370	C: 0.001, solver: sag
0.338	C: 0.001, solver: sag
0.385	C: 1, solver: saga
0.376	C: 0.01, solver: liblinear
0.358	C: 1, solver: lbfgs
0.366	C: 0.01, solver: saga

Considering all the classification results obtained with nested cross-validation in Table 5.6, we observe that the logistic regression classifier on the tf-idf feature set performs best for the classification of medical misinformation as it achieves the highest F_1 score for both classes. The logistic regression classifier is followed by the multinomial naive Bayes classifier on the set of countvectorizer features. The worst performance is obtained with the logistic regression classifier on the style-based features.

Chapter 6

Discussion

6.1 Replicating Kinsora et al. (2017)

By replicating the feature selection approach of Kinsora et al. (2017), we found 63 features as being the most important features for the classification of medical misinformation on health forums. With these features we obtained an average accuracy of 86.3%. As Kinsora et al. (2017) did not describe their feature selection approach in much detail, we made several assumptions in order to replicate their approach. We assumed that the researchers applied RFECV on all useful, numerical features. Furthermore, as Kinsora et al. (2017) did not specify which hyperparameter settings were used for the random forest classifier it was assumed that the hyperparameters were kept to their default values. The researchers, however, conducted their feature selection approach with R while our study is conducted with the use of the scikit-learn package. The parameters of the random forest classifiers of both data mining tools do not fully correspond with each other. Therefore, we only considered the `n_estimator` parameter (equivalent to the `n_tree` parameter in R), which has a significant impact on the classification performance of the logistic regression classifier, during our replication study. The lack of detail in the research of Kinsora et al. (2017) and the use of Python instead of R thus complicated the ability to fully replicate their feature selection approach.

Nevertheless, our replication study yielded nearly similar findings (expect for the average accuracy) as Kinsora et al. (2017). With RFECV the researchers identified nine features as being the most distinctive features for the classification of medical misinformation on health forums. With these nine features Kinsora et al. (2017) obtained an average accuracy of 90.1%. During RFECV the researchers, however, achieved the highest accuracy of 90.31% with a set of 53 features (a nearly similar number of features as in our found feature set). As stated in the research of Kinsora et al. (2017), this implies that the addition of more features to the set of nine features yields only a slight increase in average accuracy. With our replication study we found a nearly similar pattern from Figure 5.1. During RFECV the average accuracy increased up to 30 features after which the average accuracy fluctuated around 86%.

We suspect that this is a result of the hierarchical ordering of the LIWC features as described in section 3.2. This indicates that the LIWC features are highly correlated with each other. When more features are added, there is a high probability that the information captured by the newly added features is already incorporated in the classification model. Thus, from a certain number of features not much new information is incorporated in the model and the average accuracy stabilizes.

Despite the slightly higher accuracy of the set of 53 features, Kinsora et al. (2017) selected the set of nine features as they believed that these features are more informative and useful for the identification of medical misinformation. In order to be able to compare our most optimal feature set with the found feature set of the researchers, we selected the nine most important features from our set of 63 features. However, we were not able to fully compare both optimal feature sets with each other as the feature importances of the features in both sets are scaled differently. In addition, the random forest classifier from the scikit-learn package only denotes the feature importances of the features in Gini importance, while in the research of Kinsora et al. (2017) these importances are denoted in both the Gini importance and the permutation importance. Therefore, we only considered the order of importance (based on the Gini importance) for the comparison of both feature sets. However, Kinsora et al. (2017) only reported the feature importances for eight of the nine features from their optimal feature set. We assumed that the missing feature, which is the Bio feature, has the lowest importance and thus has the lowest position in the feature importance ranking of this set. We found from our set of nine most important features that six features also occur in the optimal feature set of Kinsora et al. (2017). The comment_length feature came up as the most important one.

We developed and implemented a comparison approach which provides an overview of the different classification results of both optimal feature sets and the set of all useful numerical features. Our optimal feature set and the set of useful numerical features attained a similar average accuracy of 86.1%, which is the highest achieved average accuracy during this process. We observe that the compared feature sets all attain a nearly equal average accuracies and that the variation in the average accuracy is small for the different random states. This can be attributed to the correlation of the LIWC features as described above. All of these feature sets are obtained in the region where the average accuracy has stabilized.

With the optimal feature set of Kinsora et al. (2017) we did not reach an average accuracy close to their obtained average accuracy of 90.1% (regardless of the value of the `random_state` parameter). We suspect that this is due to the inability to perfectly replicate their approach.

In our replication study the classification performance of the classification models is mainly assessed based on the average accuracy as done by Kinsora et al. (2017). However, it is important to note that this metric is not preferred as the dataset is imbalanced and it does not give much insight into the performance of the classifiers on

the misinformative class. The F_1 score might be better suited for the assessment of the classification models as this metric incorporates both the precision and recall.

6.2 Comparing feature sets

We explored the overall classification performances of three different feature sets for several classification algorithms which were obtained in a group 10-fold cross-validation setup. The class imbalance in the dataset was handled with two resampling methods: SMOTE and random undersampling. We observed that forum posts corresponding to the non-misinformative class are easier to classify than those corresponding to the misinformative class. We selected the best classification algorithm on each feature set for both resampling methods. We then compared these algorithms and selected the best classification algorithms for each feature set. We found that for all the selected algorithms the class imbalance was handled with SMOTE. Random undersampling removes instances randomly from the majority class such that the number of instances in both the minority and majority class is equal. The classification algorithm is trained on a much smaller number of instances than with the use of SMOTE. Consequently, we suspect that the algorithm does not capture the characteristics of the misinformative class sufficiently which results in a worse performance. This could explain the low precision that the algorithms achieve for the misinformative class when the class imbalance is handled with random undersampling. The algorithms in this case identify many posts as misinformative, while a large fraction of these posts is in fact non-misinformative. This in turn explains the higher recall in comparison with SMOTE, as the classification algorithm identifies many posts as misinformative. In general, we suspect that the small number of misinformative posts in the dataset had a large impact on the performance of the classification algorithms. The most ideal situation would be to have a dataset with a much larger and equal number of forum posts in both classes.

The best performance on the set of countvectorizer features was obtained with the multinomial naive Bayes classifier, while on both the tf-idf and style-based feature sets the best performance was obtained with the logistic regression classifier. The hyperparameter settings of these best classification algorithms were tuned with the use of nested-cross validation. For the multinomial naive Bayes classifier on the set of countvectorzier features, we found that the `alpha` parameter can best be set to '1', which is the default value. During the hyperparameter optimization we thus obtained an equal overall performance with this classifier as during single cross-validation in which all the hyperparameters are kept to their default values. Furthermore, we obtained with the logistic regression classifier on the tf-idf feature set a slightly better F_1 score for both classes than with single cross-validation. With the logistic regression classifier on the style-based feature set we achieved a lower performance. We were not able to explain this observation. Typically, this behaviour can be attributed to overfitting and underfitting. However, by comparing the train and test scores for each split during the hyperparameter optimization in the inner cross-validation loop no indications were found for these conditions.

Regardless of the small and imbalanced dataset, our highest obtained F_1 score of 0.439 corresponding to the misinformative class during 10 group cross-validation is more than 1.5 times as high as a F_1 score (0.26) which would be obtained with a baseline case in which every forum post is classified as misinformative. This highest F_1 score was obtained with the logistic regression classifier on the tf-idf feature set. With nested cross-validation we succeed to slightly increase this F_1 score although in expense of the recall, as described before.

Finally, it is important to note that we are restricted to the definition of medical misinformation stated by Kinsora et al. (2017). The built classification models in this study are therefore limited in identifying medical misinformation by this definition.

6.3 Practical application

In this study we obtained an overall and unbiased performances of several classifications algorithms with and without tuned hyperparameters for different feature sets and resampling methods. As these results give a general insight into the differences of various classification approaches, we believe that our findings can help with the development of tools for the automatic detection of medical misinformation on online (health) communities, such as health forums. These detection tools can serve as a filter which visually marks misinformative posts on such online communities causing misinformative posts to become more visible for users on these online platforms. As a result, users might get less easily misinformed and might spread these misinformative posts less quickly. Furthermore, these tools can help to identify which users share misinformative posts regularly. Online platforms could decide to block these users from their services.

From our findings these detection tools could be best developed with the logistic regression (or simalar classifiers) classifier on a tf-idf feature set. From our replication study we also found a set of optimal features. These features might also be considered during the development of such tools. As the built classification models logically identify some posts incorrectly as misinformative it may be preferable that the classifications are checked by moderators (with domain knowledge) on the online platforms. Their response and feedback on these classifications can then be used to improve the classification models.

Chapter 7

Conclusion

In this thesis, we studied the automatic detection of medical misinformation on health forums. We replicated the feature selection approach (RFECV) of Kinsora et al. (2017). Due to the lack of detail in the description of their approach several assumptions were made during our replication study. Nevertheless, we achieved nearly similar findings as the researchers. With their feature selection approach we identified a set of 63 features as being the most distinctive features for the detection of medical misinformation on health forums. Using this set of features we achieved an average accuracy of 86.3%. We observed that during this feature selection approach the average accuracy fluctuates stably around 86% after a set of 30 features. Furthermore, we developed and implemented for our replication study a comparison process which provides an overview of the performances of our most optimal feature set and that of Kinsora et al. (2017), and the set of useful numerical features. In this comparison process our optimal feature set and the set of numerical features obtain a similar average accuracy of 86.1% ,which was the highest score. All three feature sets, however, attained nearly similar average accuracies for various values of the `random_state` parameter during our comparison process. The stabilization of the average accuracy during RFECV and the lack of differences in performance between the feature sets in our replication study could be explained by the fact that the LIWC features are highly correlated with each other.

Besides our replication study, we extracted and compared the performances of three different sets of features in a 10-fold group cross-validation setup for several classification algorithms. To handle the class imbalance, two resampling approaches were employed: SMOTE and random undersampling. For each feature set and resampling method we selected the best performing classification algorithms. From this selection of algorithms we selected one best classification algorithm for each feature set. The multinomial naive Bayes algorithm performed best on the countvectorizer feature set, while the logistic regression algorithm performed best on both tf-idf and style-based feature sets. All these (best) algorithms were built on a (particular) set in which the class imbalance was handled with SMOTE. With random undersampling we achieved thus substantially worse performances. This could be the result of the fact that the algorithms do not capture the characterises of misinformative posts sufficiently as these algorithms are built on a small number of instances in both classes with this approach.

With 10-fold group cross-validation, we obtained the highest F_1 score of 0.439 corresponding to the misinformative class with the logistic regression algorithm on the tf-idf feature set. Regardless of the small and imbalanced dataset, this score is more than 1.5 times as high as a baseline approach in which all the instances are identified as misinformation (0.26).

In order to tune the hyperparameters of the best classification algorithms, we conducted nested-cross validation. We found that the best value of `alpha` parameter of the multinomial naive Bayes algorithm on the countvectorizer features for all outer test folds is '1', which is the default value of this parameter. Logically, the performance of this algorithm during the nested cross-validation was thus similar as during single cross-validation. For the logistic regression algorithm on the tf-idf feature set we were able to slightly increase the F_1 score for the misinformative class to 0.443.

In further research the feature selection approach of our replication study could be conducted to find the most important features based on the F_1 score instead of the accuracy as the F_1 score is a more preferable metric than the accuracy in the case of a small and imbalanced dataset. It would be interesting to remove highly correlated features from the dataset before applying this feature selection approach. Furthermore, as a small and imbalanced dataset can have a large impact on the classification results in future research the dataset could be expanded with more forum posts. In addition, the performances of other feature sets and classification algorithms could be explored. Classification algorithms could also be trained on combinations of feature sets. Finally, the feature importances could be obtained and studied for the built classification models when comparing different feature sets.

References

Associated Press . (2020, March 27). In Iran, hundreds die ingesting a poison they wrongly believed could fight coronavirus. *Los Angeles Times*. Retrieved from https://www.latimes.com/world-nation/story/2020-03-26/in-iran-false-belief-a-poison-fights-virus-kills-hundreds

Bengfort, B., Bilbro, R., & Ojeda, T. (2018). *Applied text analysis with python: Enabling language-aware data products with machine learning*. Sebastopol, CA: O'Reilly Media.

ECDC. (n.d.). *Event background COVID-19.* Retrieved from https://www.ecdc.europa.eu/en/novel-coronavirus/event-background-2019

Eurostat. (2020, March 27). *53% of EU citizens sought health information online.* Retrieved from https://ec.europa.eu/eurostat/web/products-eurostat-news/-/DDN-20200327-1

Ghenai, A., & Mejova, Y. (2017). Catching zika fever: Application of crowdsourcing and machine learning for tracking health misinformation on twitter. *2017 IEEE International Conference on Healthcare Informatics (ICHI)*, 518-518. doi:10.1109/ICHI.2017.58

Gupta, A., & Kumaraguru, P. (2012). Credibility ranking of tweets during high impact events. *Proceedings of the 1st Workshop on Privacy and Security in Online Social Media*. doi:10.1145/2185354.2185356

Gyenes, N., & Marrelli, M. (2020). *Health equity through health fact-checking: A primer.* Retrieved from https://health.meedan.com/primer.pdf

Hallinan, J. S. (2013). Computational intelligence in the design of synthetic microbial genetic systems. In C. Harwood & A. Wipat (Eds.), *Methods in microbiology* (pp. 1–37). Cambridge, MA: Academic Press.

Han, J., Kamber, M., & Pei, J. (2011). *Data mining: Concepts and techniques*. Waltham, MA: Morgan Kaufmann.

Hill, J., Agewall, S., Baranchuk, A., Booz, G., Borer, J., Camici, P., ... Vrints, C. (2019). Medical misinformation: Vet the message! *Journal of the American Heart Association*, *8*(3), e011838. doi: https://doi.org/10.1161/JAHA.118.011838

Hou, R., Pérez-Rosas, V., Loeb, S., & Mihalcea, R. (2019). Towards automatic detection of misinformation in online medical videos. *Conference: 2019 International Conference on Multimodal Interaction*, 235-243. doi: 10.1145/3340555.3353763

Jurafsky, D., & Martin, J. H. (2019, October 16). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.* Retrieved from https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf

Kinsora, A., Barron, K., Mei, Q., & Vydiswaran, V. (2017). Creating a labeled dataset for medical misinformation in health forums. *2017 IEEE International Conference on Healthcare Informatics (ICHI)*, 456-461. doi: 10.1109/ICHI.2017.93

Kuhn, M., & Johnson, K. (2019). *Feature engineering and selection: A practical approach for predictive models*. Boca Raton, FL: Chapman and Hall/CRC.

Li, Y., Wang, X., Lin, X., & Hajli, M. (2016). Seeking and sharing health information on social media: A net valence model and cross-cultural comparison. *Technological Forecasting and Social Change*, 28-40. doi: 10.1016/j.techfore.2016.07.021

Marston, L. (2010). *Introductory statistics for health and nursing using spss*. London,England: Sage Publications Ltd.

McHugh, M. (2012). Interrater reliability: The kappa statistic. *Biochemia medica*, 276-82. doi:10.11613/BM.2012.031

Mian, A., & Khan, S. (2020). Coronavirus: the spread of misinformation. *BMC Medicine*. doi:10.1186/s12916-020-01556-3

Padurariu, C., & Breaban, M. (2019). Dealing with data imbalance in text classification. *Procedia Computer Science*, *159*, 736-745. doi:10.1016/j.procs.2019.09.229

Pennebaker, J. W., Booth, R. J., Boyd, R. L., & Francis, M. E. (2015). *Linguistic inquiry and word count: Liwc2015*. Retrieved from https://liwc.wpengine.com/wp-content/uploads/2015/11/LIWC2015_OperatorManual.pdf

Pennebaker, J. W., Boyd, R. L., Jordan, K., & Blackburn, K. (2015). *The development and psychometric properties of liwc2015*. Austin, TX: University of Texas at Austin. doi:10.15781/T29G6Z

Sarkar, D. (2016). *Text analytics with python: A practical real-world approach to gaining actionable insights from your data*. New York City, NY: Apress.

Sluijmers, M. (2020, June 15). *Logistic Regression: Discover the Powerful Classification Technique.* Retrieved from https://towardsdatascience.com/logistic-regression-discover-the-powerful-classification-technique-d60c0ae4d3b1

Swire-Thompson, B., & Lazer, D. (2020). Public health and online misinformation: Challenges and recommendations. *Annual Review of Public Health*, *41*. doi:10.1146/annurev-publhealth-040119-094127

Vanderplas, J. (2016). *Python data science handbook: Essential tools for working with data*. Sebastopol, CA: O'Reilly Media.

Vydiswaran, V., Liu, Y., Zheng, K., Hanauer, D., & Mei, Q. (2014). User-created groups in health forums: What makes them special? *Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014*, 515-524. Retrieved from https://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/download/8045/8152

Waszak, P., Kasprzycka-Waszak, W., & Kubanek, A. (2018). The spread of medical fake news in social media – the pilot quantitative study. *Health Policy and Technology*, *7*, 115-118. doi:10.1016/j.hlpt.2018.03.002

WHO. (2020, January 5). *Pneumonia of unknown cause – China.* Retrieved from https://www.who.int/csr/don/05-january-2020-pneumonia-of-unkown-cause-china/en/

Zadrozny, B. (2019, December 29). Social media hosted a lot of fake health news this year. Here's what went most viral. *NBC News*. Retrieved from https://www.nbcnews.com/news/us-news/social-media-hosted-lot-fake-health-news-year-here-s-n1107466

Zhang, C., & Ma, Y. (2012). *Ensemble machine learning: Methods and applications*. New York,NY: Springer Science+Business Media.

Appendix

A Most distinctive features

The figure below shows the 63 most distinctive features with their feature importances obtained with RFECV during the replication study.

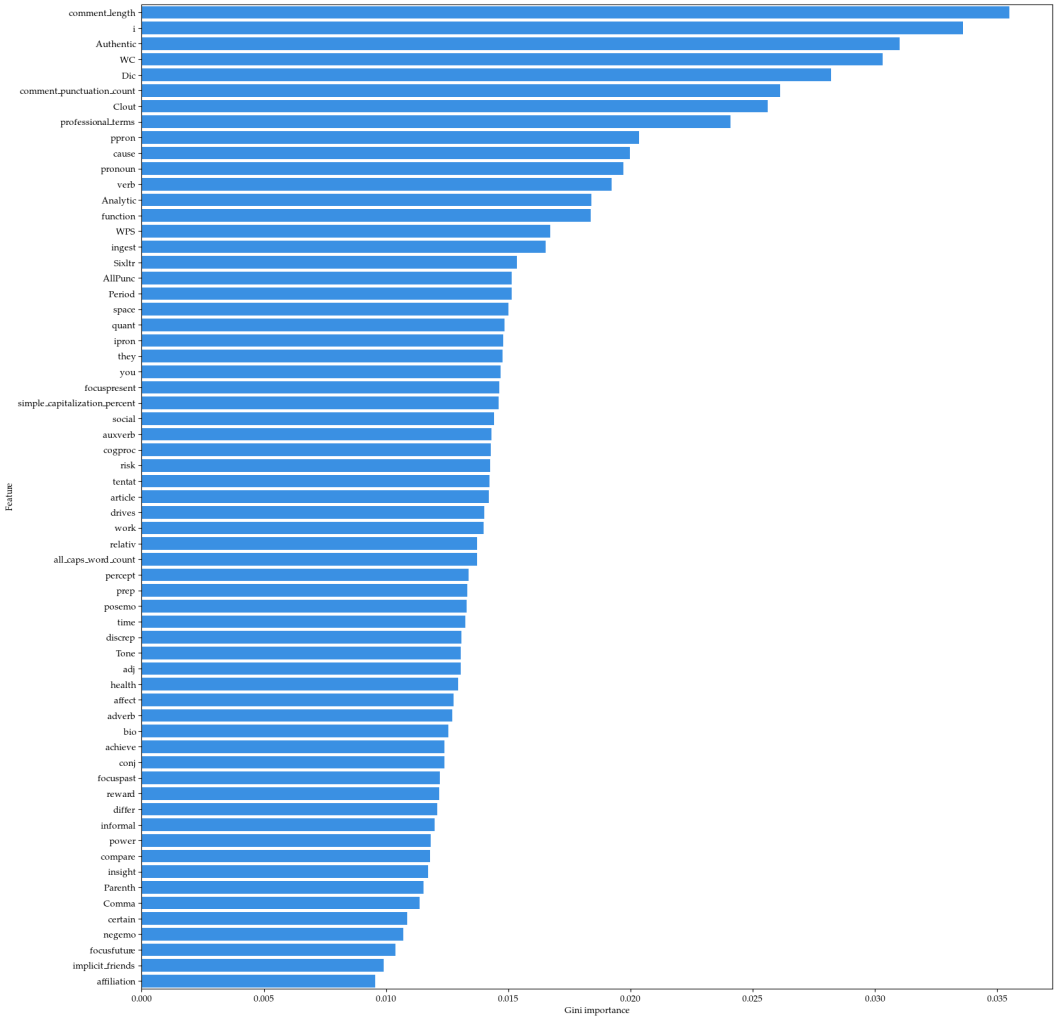

Figure A.1: The most important features with their feature importances denoted in mean decrease in Gini obtained during the replication study with RFECV.

B Parameter grids

The table below shows the hyperparameter grid of the best classification algorithm on each feature set used during the optimization of the hyperparameters.

Table B.1: The used parameter grids of the best classification algorithm on each feature set during the optimization of the hyperparameters with nested cross-validation. The same parameter grid is used for the optimization of the hyperparameter settings of the logistic regression classifier on both feature sets except for the newton-cg value of the solver parameter, which is excluded from the parameter grid of the classifier on the style-based feature. During hyperparameter optimization the random_state parameter of the logistic regression classifier is set to 42 for both sets. The random_state parameter is only used when the solver parameter is set to liblinear, sag or saga.

Classification algorithm	Feature set	Hyperparameter	Default value	Values
Multinomial naive Bayes	Countvectorizer features	Alpha	1.0	0.00001, 0.0001, 0.001, 0.01, 0.1, 1
Logistic regression	Tf-idf features	C	0.0001	0.001, 0.001, 0.01, 0.1, 1, 10, 100, 1000
		Solver	lbfgs	newton-cg, lbfgs, liblinear, sag, saga
Logistic regression	Style-based features	C	0.0001	0.001, 0.001, 0.01, 0.1, 1, 10, 100, 1000
		Solver	lbfgs	lbfgs, liblinear, sag, saga