Assessment of the project report

Student name: Lennart Schilling, lensc874

I have assessed your report with respect to the criteria defined below. For each criterion, I have assigned one of the scores 0 (below expectation), 3 (meets expectation), or 5 (exceeds expectation).

Clarity Is it clear what was done in this project, why it was done, and how it was done? Is the report well-written and well-structured?

0 There are some important questions about the method, results, or analysis that even expert readers are not able to resolve.

3 Any student who has successfully completed the course should understand what was done in this project, why it was done, and how it was done.

[✓ 5] The report is well-polished. In regard to clarity of presentation, it would be acceptable for an academic journal or conference proceedings.

Specific comments: very clear and detailed; great job!

Soundness and correctness Is the technical approach sound and well-chosen? Are the claims made in the report supported by proper experiments, and are the results of these experiments correctly interpreted?

0 Troublesome. There may be some ideas worth salvaging here, but the work should really have been done or evaluated differently. The claims made in the report have no support in the experimental results.

3 Fairly reasonable work. The approach is not bad, the methods are appropriate, and at least the main claims are probably correct. The report contains a discussion of the possibilities and limitations of the technical approach.

[✓ 5] The approach is very apt, and the claims are convincingly supported. The report contains a well-developed discussion of the possibilities and limitations of the technical approach, including the reliability and validity of the results.

Specific comments: the methodological decisions are carefully motivated and discussed. The experimental part shows significant craftsmanship.

Related work Does the report show awareness and understanding of related work documented in scientific sources? Is it clear where the work done in the project sits with respect to that related work?

0 The report shows little awareness and understanding of related work. References to scientific sources are missing or incomplete. There is no account of how the work done in the project compares to the related work.

3 The report shows some awareness and understanding of related work. Scientific sources are adequately referenced. The relation between the work done in the project and the work documented in the scientific sources is clear.

[✓ 5] The report features a precise and enlightening comparison with related work. References are complete and consistently formatted. The majority of scientific sources are peer-reviewed research articles.

Specific comments: Well done!

Creativeness How creative is the project? For example: Does the project target a new problem? Does it contribute a new data set? Does it use any machine learning models that were not covered in the course?

0 There are no creative elements in this project. The project is essentially a repetition of one of the lab assignments.

3 The project contains at least one creative element.

[✓ 5] There are many creative elements in this project. The project goes significantly beyond what has been covered in the course.

Specific comments: While a 'simple' classification project at its core, there is a lot of creativity in the research question and the design of the data set. There is also a classifier that was not used in class (XGBoost).

Substance Based on the report, does this project have enough substance, or would there have been room for more ideas, results, or analysis? (The expected amount of work for the project module is 88 hours.)

0 Seems thin. I (the examiner) would have expected significantly more ideas, results, or analysis for a project with this timeframe.

3 Represents an appropriate amount of work for a project in this course.

[✓ 5] Contains significantly more ideas, experiments, and analysis than what I (the examiner) would have expected for a project with this timeframe.

Specific comments: data collection, extensive experiments, careful analysis

Grade

For a passing grade, your score *for each criterion* must be at least 3. Your grade is determined by your total score, as specified in the table below.

Total score: 25 Grade: A

Linköping, 2020-04-08

Kuhlmann

Marco Kuhlmann, examiner

Total score	15	17	19	21	23
Grade 732A92	E	D	C	B	A
Grade TDDE16	3	3	4	5	5

Project report

Performing sentiment analysis of Twitter data to analyse people's happiness

A comparison between a highly and less developed region

732A92 Text Mining

Lennart Schilling (lensc874)

Division for Statistics and Machine Learning
Department of Computer and Information Science
Linköping University

30-01-2020

Abstract

According to the World Happiness Report 2019, people from less developed regions show significantly lower happiness than people from highly developed regions. Using the UK and a large region in southern Africa, this theory is examined by comparing Twitter data from the two areas. Multiple sentiment classifiers (Naïve Bayes, Decision Tree and XGBoost) using different combinations of vectorizer, pre-processing procedure and hyperparameter setting are trained and tested using the labelled Sentiment140 data set. A Naïve Bayes classifier identified as optimal is then used to classify previously self-crawled tweets from both regions. To increase the chance of a correct classification, classified tweets that are not assigned a minimum probability of 75% being in the positive or negative class are removed from the analyses. Following this procedure, the tweets considered are expected to be 85% correctly classified. The classifications of the self-crawled tweets are used to obtain the relative frequency of positive and negative tweets within both regions for different years. This is extended by the identification of named entities within the classified tweets. This combined information is used to analyse the happiness of the people in the two regions. The results obtained do not confirm the extreme differences in people's happiness between less and highly developed regions as presented in the World Happiness Report.

1. Introduction

1.1 Background and aim

Since 2012, the World Happiness Report [6] is yearly published by the United Nations Sustainable Development Solutions Network. Countries are given an overall score which aims to reflect the average happiness of the people living there. Within the report for 2019, it is generally noticeable that highly developed countries are ascribed significantly higher levels of happiness than developing countries (see figure 1). The score is built on various categories regarding for example the economy, corruption or social support. But does this really sum up people's happiness?

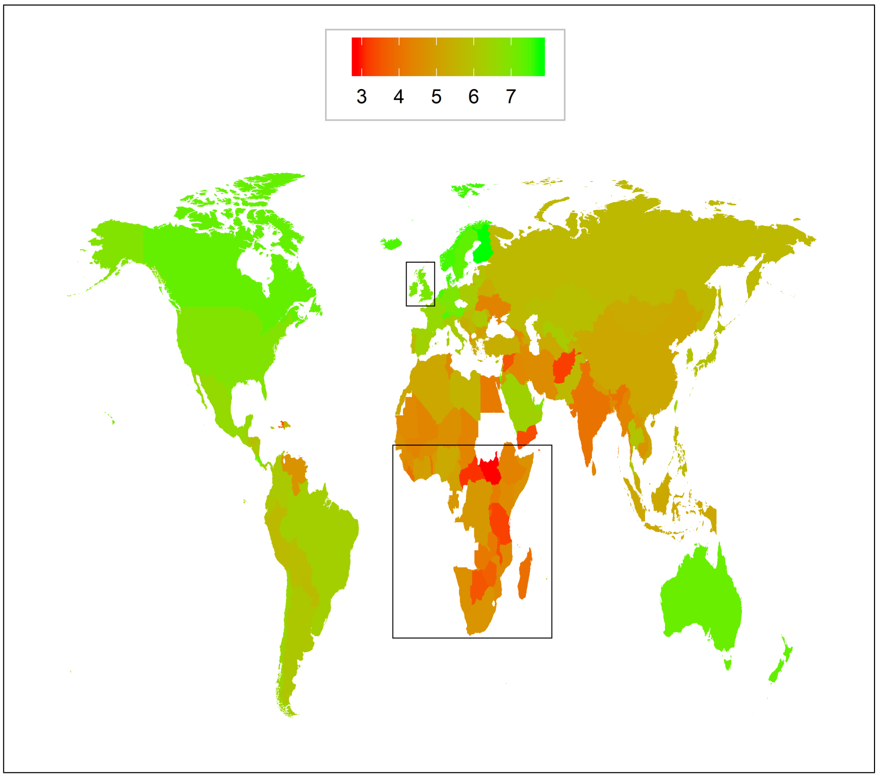

Figure 1: *World map to express the happiness of countries according to the World Happiness Report. Countries are coloured according to their happiness score. Countries that are left blank were not rated in the World Happiness Report. The UK and the southern part of Africa, on which the project focuses, are highlighted.*

Another approach is the analysis of human behaviour in social media. Posts on Twitter (called "tweets") could be used to assess the emotional state of a person. It is assumed that a positive tweet is an expression of a positive emotional state, whereas a negative tweet reflects a negative emotional state.

The project aims to derive and compare the happiness of people within (according to the World Happiness Report 2019) low- and high-rated countries by applying a Twitter Sentiment Analysis. A method is implemented which automatically classifies a tweet regarding its positive or negative sentiment. Due to data availability, the project focuses on tweets from the United Kingdom (UK) and the southern states of Africa.

1.2 Related work

Dodds et al. [2] also try to determine people's happiness with their implemented "hedonometer" using Twitter data. By assigning a happiness score on a scale from 1 (sad) to 9 (happy) to the most frequently used words in the English language, a context-free average happiness score is calculated for tweets. The research group uses this methodology to determine, among other things, the average happiness of people in near real-time as a timeline since 2009. By using geoinformation from the tweets, it also for example compares the happiness of different US states.[1]

A study by Frank et al. [3] use the same sentiment analysis instrument by Dodds et al. to analyse happiness of people as a function of distance from their expected location. They investigate that the happiness of people determined via tweets increases logarithmically with distance from their average position.

Reece et al. [10] use Twitter data to predict depression and Post-Traumatic Stress Disorder (PTSD) in Twitter users. By analyzing the Twitter behavior of 105 depressive and 99 healthy people, they implemented a combination of two different models (Random forest and state-space model) that enables them to detect signs of PTSD and depression many months prior to clinical diagnosis using tweets as input.

Go et al [4], the publishers of the data set used in this project to train the sentiment classifier, have also used this data set to classify tweets in terms of sentiment using various machine learning methods such as Naïve Bayes or Support Vector Machine.

There are many other studies that use social media for the psychological analysis of users. In addition to Twitter, Instagram, for example, is also analysed by Reece and Danforth [9] to predict user depression based on the pictures posted.

[1] For a practical insight into the different applications of the hedonometer, see http://hedonometer.org/index.html.

2. Theory

In order to classify a tweet regarding its sentiment, three models (Naïve Bayes, decision tree, XGBoost) are proposed. Naïve Bayes as a very fast and simple model is used as a baseline and the theory of this classifier is assumed to be known. This chapter presents the theory of decision tree and XGBoost classifiers.

2.1 Decision tree

Tree-based methods aim to divide the set of possible values of non-target features (predictor space) into different distinct and non-overlapping regions (called nodes). In case of classification, each observation receives the label of the majority class of the final node (called terminal node or leaf), into which the observation is assigned depending on its non-target features. The different nodes are defined by a set of splitting rules. In a top-down, greedy approach, known as recursive binary splitting, the root node, which contains all observations, is gradually split further, whereby a current end node is always split into two new nodes. In each step, the node that produces the best direct split is selected [7].

Criteria for measuring the purity within the nodes

To decide on the quality of a split, different criteria such as the Gini index or the Cross-entropy are selected. Both criteria examine the purity (homogeneity) of the split resulting nodes in relation to the target variable while the purity is aimed to be maximized.

$$\text{Gini index: } \sum_{k=1}^{K} \hat{p}_{mk} \left(1 - \hat{p}_{mk}\right) \tag{1}$$

$$\text{Cross-entropy: } -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk} \tag{2}$$

Both the Gini index and the Cross-entropy calculate the impurity of a node using the proportions $\hat{p}_{mk}$ of the observations belonging to each class k in node m. In both cases, the metric shows a value approaching 0 with increasing purity [5].

Pruning

To avoid making the model too complex, which would lead to a high probability of overfitting, different pruning options can be applied. A distinction is made between pre-pruning (during the recursive splitting process, defined stop criteria lead to a termination of the process) and post-pruning (after maximum purity within the nodes has been reached, the model is shortened). An option is to set the maximum depth of the tree in advance [8].

2.2 XGBoost

XGboost is an implementation of a tree boosting system. Multiple trees are generated by using information from previous trees to generate additional trees. The boosting approach leads to a slow learning process, which increases with each additional tree generated. Based on the initial probabilities with which the observations are assigned to a class, the residuals are calculated by comparing the target and the assigned probability. The trees, which are then iteratively greedily generated, are grown using these residuals instead of the actual values of the target variable [1].

Different parameters can be tuned to avoid overfitting. Beside the number of differently generated trees or a maximum depth for each tree, the learning rate eta as shrinking parameter plays a decisive role. It controls how fast the system learns. It determines the weight of the outputs of newly generated trees [7]. The influence of individual trees is reduced by shrinkage to leave space for future trees to improve the model [1].

3. Data

This chapter aims to present the data used. The labelled data set *Sentiment140* is used to train the sentiment classifier. The trained model is then applied to self-crawled tweets from the UK and the African region.

3.1 Sentiment140

Data source

Sentiment140 is a data set published by Go et al. [4]. The data consists of 1.6 million English tweets, covering the period from April 2009 to June 2009. Each tweet is labelled with either 0 (negative) or 4 (positive) regarding its general sentiment. The label is a result of how the data was collected. Using the Twitter API, Go et al. extracted the tweets with the help of emoticons. Querying data including positive emoticons such as ":)" are labelled as positive while data including negative emoticons like ":(" are labelled as negative. Tweets containing both positive and negative emoticons are removed. Since the emoticons are the only query criteria, the collected tweets refer to completely random posts without any specific context.

Data pre-processing

The original provided tweets sometimes include a username to direct the message to other users. This is done by specifying the @ symbol before the username. Links which reference to other web pages may also be part of posts by some users. Both elements are considered insignificant in relation to the analysis of a sentiment. In the following, they are removed from all provided tweets. An example for a tweet which is cleaned is shown in table 1. Other text elements which are usually pre-processed such as stop words are not necessarily considered as insignificant yet. The choice of a certain vectorizer and a certain set of hyperparameters might lead to better results when such a pre-processing is performed while other choices may not. Consequently, further pre-processing is not performed yet. The final pre-processing is decided in the further steps when different combinations of pre-processing elements and model settings are compared.

Tweet before cleaning	@nchokkan https://[...].com/search?[...] But all says not in stock
Tweet after cleaning	But all says not in stock

Table 1: *Example of a cleaned tweet.*

Data description

As a result of how Go et al. [4] collected the data, it follows that the Sentiment140 data set only consists of tweets either labelled as positive or negative, but not neutral. Regarding the sentiment label, the 1.6 million unique tweets are equally distributed with 800,000 tweets each. The pre-processing (removal of usernames and links) reduces the number of different tokens within all tweets from 684,358 to 287,922 down to 42.07 % of its original size.

3.2 Self-crawled tweets

Since the goal of the project is to analyse and compare the happiness of the people between (according to the World Happiness Report 2019) high and low rated regions by performing a Twitter sentiment analysis, Twitter data for these regions is required.

Considerations regarding the criteria for selecting suitable regions were directed towards the score in the World Happiness Report, the language and the Internet access of the people living there. The goal of the project requires finding two regions with very high and very low scores respectively. In addition, the aim is to access a large part of the people living there and not just a small subset. This implies that many people living there communicate (and therefore post) in English and have access to the Internet.

In addition, the speed at which the API (described below) returns tweets is extremely dependent on the defined region. This speed difference in the use of the API is another criterion for selection.

For the UK as a high rated country and the chosen part of Africa as a low rated region, the speed is acceptable. The African region was chosen because the percentage of English language tweets is relatively high compared to many other low rated regions. To focus on a single country from the African region, such as Kenya, would have taken too much tweet extraction time. This explains why all the work is focused on the analysis and comparison of these two regions.

Data source

Since no relevant data set for both regions could be found, it has been done by using Tweepy, a Python library for accessing the Twitter API.[2] The library allows to query location-based tweets by specifying a rectangle given by the four outer coordinates of the desired region.

However, the standard version of the Twitter API which Tweepy connects to only returns recent tweets published within the past seven days. Since the following investigation also includes an analysis over the past years for each region, Twitter data before 2019 is required. Another python library called GetOldTweets3 is able to extract the old tweets.[3] Both the time period and the location of the desired tweets are specified. In this case, the regions are not specified by a rectangle but by drawing a circle around a fixed location. In total, three fixed locations with a specified radius each have been chosen to collect tweets from a similar region compared to the one which has been used in Tweepy before (see figure 2 for a comparison of the African region). In order to make clear which extracted data a description refers to, the sources (Tweepy, GetOldTweets3) are occasionally mentioned in the following.

[2] For more information about Tweepy, see https://www.tweepy.org.

[3] For more information about GetOldTweets3, see https://pypi.org/project/GetOldTweets3.

Region	2015	2016	2017	2018	2019
UK	40 000	40 000	40 000	40 000	39 999
Africa	37 000	10 696	37 000	14 024	37 000

Table 2: *Extracted tweets via 'GetOldTweets3'.*

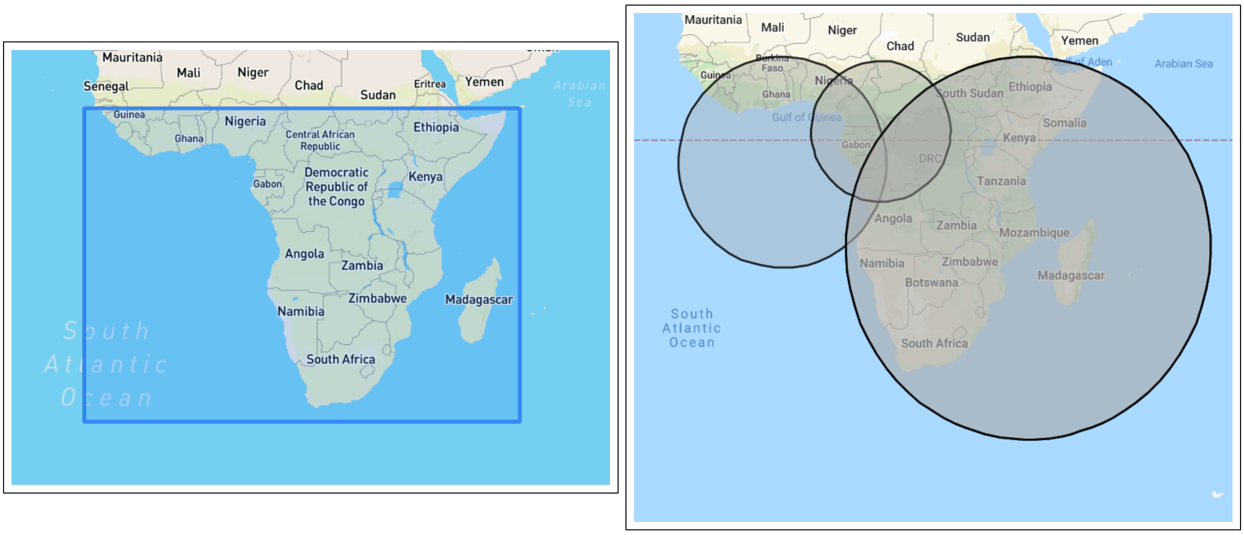

Figure 2: *Defined African region in 'Tweepy' and 'GetOldTweets3'. While 'Tweepy' uses four specified coordinates to build a rectangle, 'GetOldTweets3' uses one coordinate with a radius to form a circled area from which tweets are returned.*

Data pre-processing

Since the goal is to apply the classifier learned with the Sentiment140 data set on the self-crawled tweets, all of them are pre-processed in the same way. Usernames and links are removed. Since the emoticons which have been used to query the tweets from the Sentiment140 data set are already removed, emoticons are also removed from the self-crawled tweets.

Data description

For the UK, in total 102 158 recent tweets from three different days in 2019 (15th November, 19th November and 15th December) are collected using `Tweepy`. After pre-processing, the number of different tokens is 72 019.

For the African region, 102 345 recent tweets for again three days in 2019 (14th November, 15th November, 15th December) are collected via `Tweepy`. After pre-processing, the number of different tokens is 62 743.

Table 2 shows how many tweets for the years 2015-2019 are extracted via `GetOldTweets3`. Since, as Frank et al. [3] mention, only about 1% of all tweets are geolocated, extracting tweets by location may not return the desired quantity. Even if for the Africa region in the years 2016 and 2018 a significantly lower number of tweets is extracted via `GetOldTweets3`, this is still to be considered enough to use the tweets within the following analysis.

4. Method

4.1 Training sentiment classifier

The Sentiment140 data set is first divided into training data (70%; 1.12M tweets) and test data (30%; 480 000 tweets). The balance of the original data set (50% labelled as positive and 50% labelled as negative) is maintained in both data sets.

To fit the presented models to data, the functions `MultinomialNB`, `DecisionTreeClassifier` (both from the `sklearn` module) and `XGBClassifier` (from the `xgboost` module) are used. Each classifier is fit to the entire training data set. However, the optimal settings for each model must first be determined.

Tweets can be vectorized differently. Two vectorizers (`CountVectorizer`, `TfidfVectorizer`) are considered as possible reasonable fits.

Usernames and links are removed from all tweets. However, the tokens determined by a vectorizer can be further processed differently. The following options are implemented: Either no further processing or lemmatization. Two further specializations, lemmatization of strings consisting only of alphabetical characters with or without stop words, are added. Therefore, four different pre-processing variants are investigated.

For each of the three classifiers on which we focus, there are different hyperparameters for which values must be set before training. Reasonable values are determined in advance by an iterative procedure in which parameters are changed while the others are fixed. The changed model performance is used to move the parameters in the right direction to define a range of suitable values. In addition, qualitative reasons are given for which hyperparameters different values are to be tested. For Naïve Bayes, a choice other than a uniform prior (chosen by default) cannot be justified. The additive smoothing hyperparameter alpha is therefore the only parameter that is varied. For the decision tree, the splitting criterion and the maximum depth of the tree are varied. For the XGBoost classifier, different leaning rates (eta) and different maximum depths of the trees are tested.

To derive a suitable combination of vectorizer, pre-processing and hyperparameter values, Cross-validation is performed. The data to be used for this process is divided into three folds. In each iteration, two training folds and one validation fold are defined, so that each fold represents the validation fold once. The model is trained with the two training folds and then applied on the training folds and the validation fold. The average training and validation performance of the model (more information about the evaluation is presented in chapter 5) provides information about the suitability of the selected vectorizer-pre-processing-hyperparameter combination. Cross-validation is performed using the function `GridSearchCV` from the `sklearn` module.

Using Naïve Bayes as an example, figure 3 visualizes the enormous number of models that are trained in this process to identify optimal model settings. Due to the high time intensity when using the entire training data set, another “tuning data set” is used instead. Randomly, 10% of the training data set (120 000 tweets in total) is extracted while maintaining the balance (50% labeled positive, 50% negative).

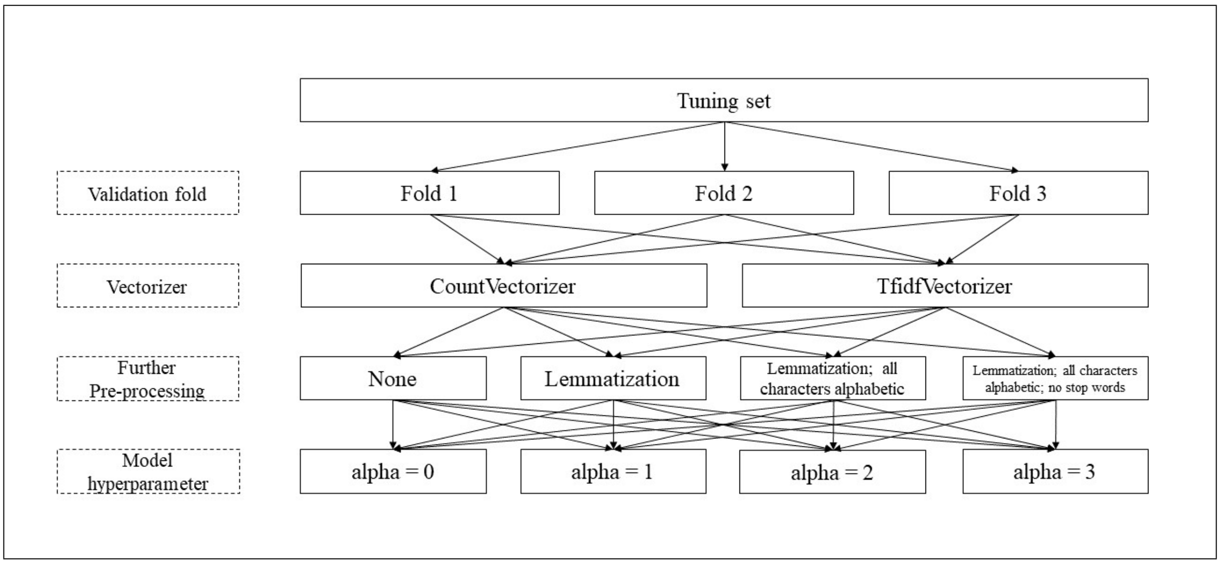

Figure 3: *Process to identify an optimal pipeline for Naïve Bayes. Three different validation folds, two vectorizers, four pre-processing procedures and four different choices of the additive smoothing model parameter leads to a total number of 96 trained models. One trained model is identified by following one path from the top to the bottom.*

For each model (Naïve Bayes, Decision tree, XGBoost), the most promising combination of vectorizer, pre-processing function and model hyperparameters is used to train the classifier on the entire training set afterwards. Classifying the tweets of the test data set using the trained models enables an evaluation of the performance of the model on previously unseen data. Classifying the tweets of the training data set is used to include the analysis of overfitting within the model comparison. The model considered to be the best is then used to classify the self-crawled tweets.

4.2 Classifying self-crawled tweets

The classifier is trained on data that is exclusively labelled positive or negative. A neutral classification is therefore not possible. Under this consideration as well as an increased probability of a correct classification, only tweets are taken into account that are assigned at least a 75% probability of being in the positive or negative class by the classifier. Tweets that fall into the positive class with a probability of at least 75% are classified as positive. In return, tweets that fall into the negative class with a probability of at least 75% are classified as negative. The probabilities of a tweet for both classes are returned by the function `predict_proba`.

In order to enable an up-to-date comparison between the UK and the African region using the classification of the self-crawled tweets extracted via `Tweepy` from 2019, the same number of tweets is set for both groups. This is achieved by random downsampling within that group in which a higher number of tweets remain after the classification and exclusion procedure.

A comparison of the happiness is made possible by the percentage of positively classified tweets within each group.

4.3 Identification of categories contained in the classified tweets

An extension of the happiness analysis is achieved with the library `spaCy`. Using an implemented model, the library is used to locate and classify named entities of different types (henceforth called categories) within the tweets.[4] The previously performed positive or negative classification of the tweets is now supplemented with additional information about contained categories. Since the sentiment is predicted for each tweet in which one or more categories are additionally identified, the relative frequencies of each category within the positive and negative tweets are obtained. It is calculated by dividing the absolute frequency of the category within a tweet collection (for example positive tweets from the UK) by the total number of all identified categories in the respective tweet collection. Consequently, it can be derived which categories people within both regions tend to express themselves more positively or more negatively about. In addition to the quantitative assessment of happiness, a qualitative investigation is thus also carried out.

4.4 Analysing happiness over time

Next to the detailed analysis for the 2019 tweets extracted using `Tweepy`, the tweets extracted using `GetOldTweets3` also enable to analyse the tweeting behaviour of people from the UK and Africa region over time.

In 3.2 it is shown that the same number of tweets has not been extracted for all years and both regions. In order to compare the collected tweets from 2015-2019 across both years and regions, this requires a uniform number of tweets.

It is achieved by randomly downsampling each collection of tweets for each region (UK, Africa) and each year (2015-2019). The number of tweets downsampled to is set to 10 696 tweets, which is the minimum number of tweets extracted (in 2016 for the African region).

The classifier is then applied to the downsampled tweets. As a result, the percentages of positively classified tweets per year and region are used to assess and compare people's happiness over time.

[4] For more information about the entity recognition system in `spaCy`, see https://spacy.io/usage/linguistic-features#named-entities.

5. Results

The different pipelines of the three classifiers are compared using the accuracy and F1 metric. By applying Cross-validation, the average of the metrics for both classifying the training and the validation folds are calculated. The metrics of the best ten pipelines of each classifier are shown in figures 9, 10 and 11 in the Appendix.

The optimal pipelines of the classifiers are described in table 3 in the Appendix. Training the models with their optimal settings on the entire training data set and classifying both training and test data afterwards leads to the scores shown in figure 4.

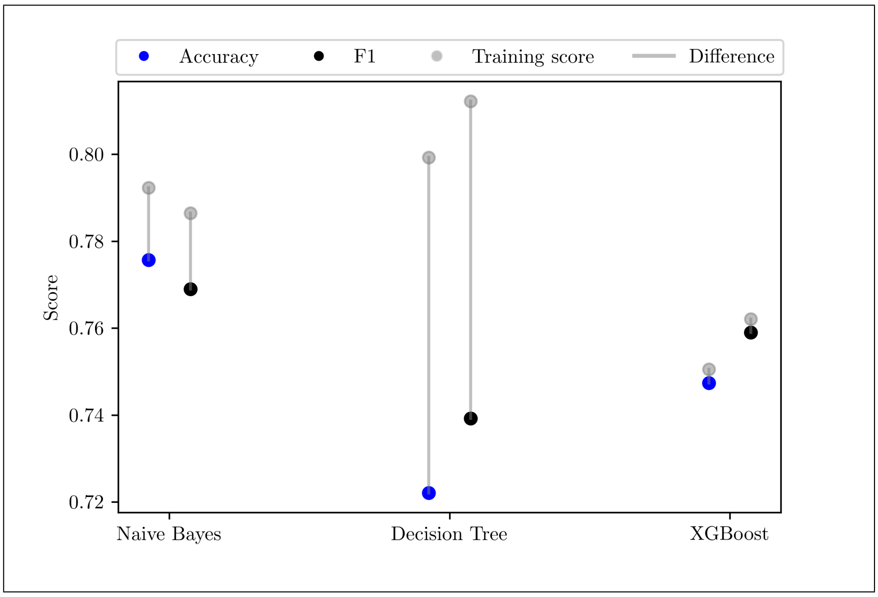

Figure 4: *A Comparison of the three classifiers with their optimal pipelines. For each trained model, the accuracy and F1 score after applying the models to the test data is displayed. The values after applying the models to the training data are also displayed to analyze over/under fitting. The vertical connection of the training and test score illustrates the extent of over/under fitting.*

Naïve Bayes has the highest scores for both test metrics (77.57% accuracy, 76.9% F1 score). The decision tree shows the lowest scores for both test metrics. It also shows the largest difference between the scores of the test and training metrics. The XGBoost classifier shows the smallest differences between test and training metrics, but has lower test scores (74.74% accuracy, 75.91% F1 score) than the Naïve Bayes classifier.

Excluding tweets that are not assigned a minimum 75% probability of a sentiment, the Naïve Bayes classifier even returns higher training and test scores for the metrics (e.g. 85 % test accuracy), as seen in figure 5 which also includes an analysis of precision and recall.

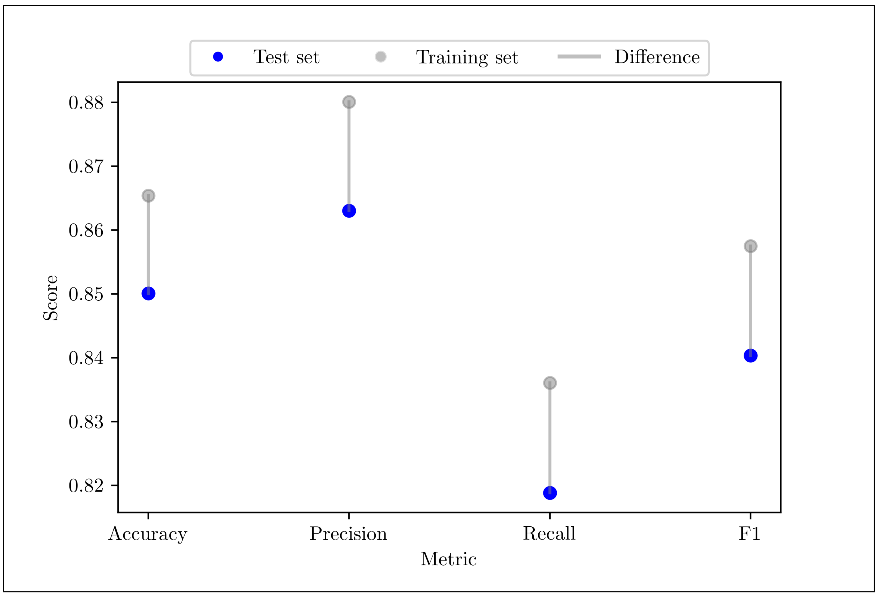

Figure 5: *Training and test scores of the Naïve Bayes classifier with its optimal pipeline when classified tweets not assigned a minimum 75 percent probability of a sentiment are excluded.*

exclusion

The Naïve Bayes classifier is then used to classify the self-crawled tweets extracted via `Tweepy`. The exclution procedure results in 60 579 and 50 449 remaining tweets for the UK and the African region respectively. Through random downsampling, both tweet quantities are reduced to 50 000 classified tweets. The remaining UK tweets are classified 66.15% positive and 33.85% negative. The remaining tweets of the African region are classified as positive by 62.61% and negative by 37.39%.

Figures 6 and 7 compare the relative frequencies of the predicted categories within the positive and negative tweets to determine which categories people in both regions tend to rate more positively and negatively.[5]

[5] For a description of the category types, see https://spacy.io/api/annotation#named-entities).

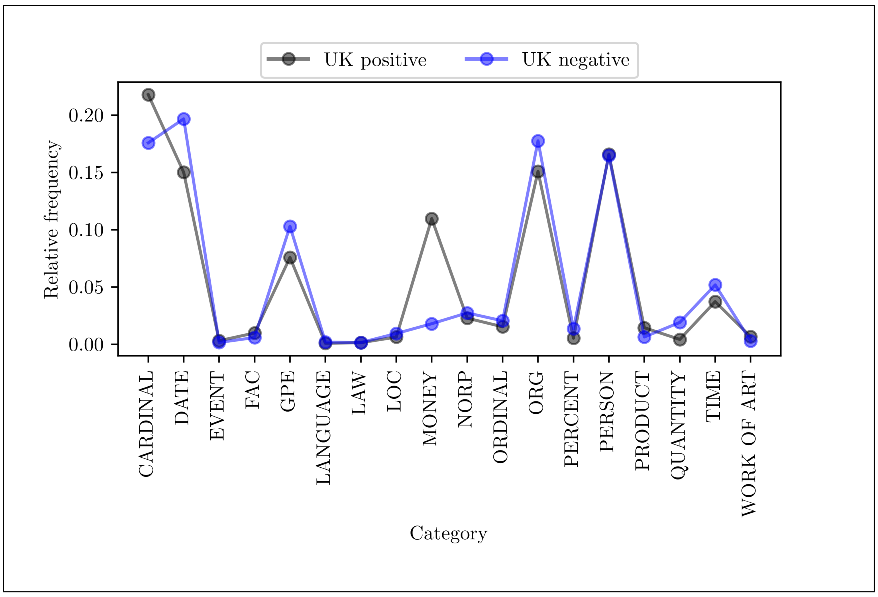

Figure 6: *Relative frequencies of identified categories within the tweets of people from the UK. The relative frequency of a category indicates how often the category is identified compared to the other identified categories within the tweet group (positive or negative tweets).*

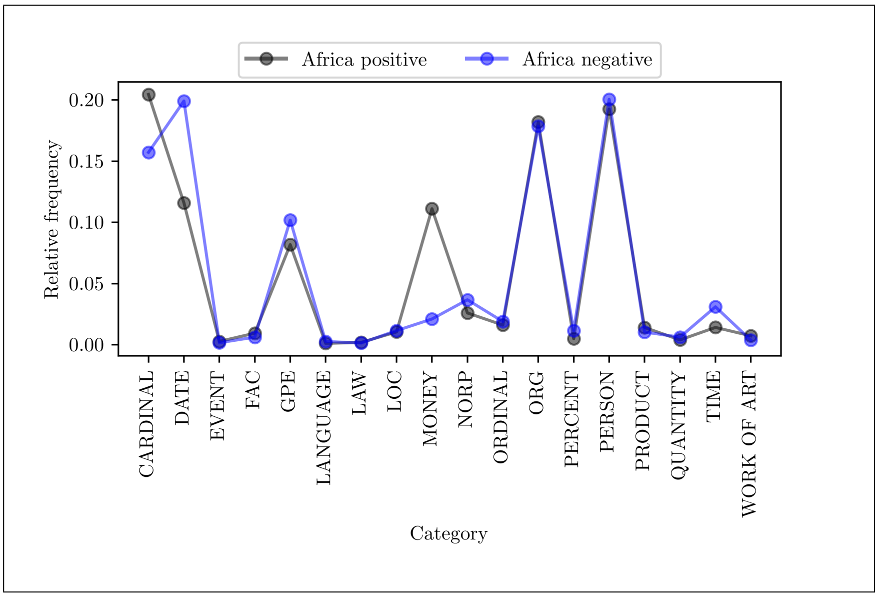

Figure 7: *Relative frequencies of identified categories within the tweets of people from the African region. The relative frequency of a category indicates how often the category is identified compared to the other identified categories within the tweet group (positive or negative tweets).*

For both the UK and the African region, tokens belonging to the category “MONEY” have a higher relative frequency in positive-classified tweets than in negative-classified tweets. Another example is the category “GPE” which is characterized in both cases by the fact that the relative frequency of the category within negative tweets exceeds the relative frequency within positive tweets.

A difference results from the category “ORG”. While the values for the African region are very similar, the relative frequency of tokens of the category within negatively classified tweets is higher than within positively classified tweets for the UK.

Figure 8 illustrates the results of the sentiment analysis for the years 2015 - 2019. The classification of tweets extracted using `GetOldTweets3` shows an approximate linear increase in the proportion of positively classified tweets for both regions until 2018. In contrast, for the year 2019, a significant decrease of positive-classified tweets can be seen for both regions. The calculated percentage of positive tweets for the year 2019 is for both regions (about 60% for the UK, about 55% for the African region) below the results from before when `Tweepy` tweets are used (66.15% for the UK, 62.61% for the African region). Moreover, in 2015, 2017 and 2018, the proportion of positively classified tweets is higher among people from the African region than among people from the UK.

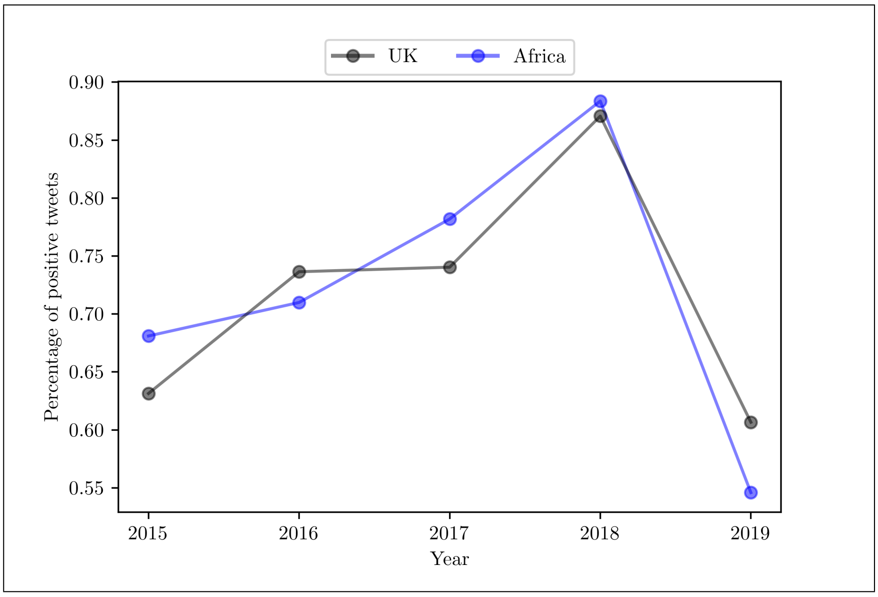

Figure 8: *Proportion of positive-classified tweets over the years 2015 - 2019. The percentages are the result of the classified tweets extracted via ‘GetOldTweets3‘.*

6. Discussion

Analysing figure 9 in the Appendix, the pipeline which belongs to model index 8 is considered as the best combination for the Naïve Bayes classifier. The reason is that the average validation scores of accuracy and the F1 metric are very close to the maximum scores. However, since the differences between the validation and training scores are significantly smaller than for other models, less overfitting is assumed so that this model is expected to work best for unseen data.

Since for the Decision Tree classifier the pipeline with the highest validation scores (Model index 13) does not show significantly larger differences between the training and validation metrics (see figure 10 in the Appendix), this model is considered the optimal model.

For the XGBoost classifier (see figure 11 in the Appendix), the pipelines with the model indices 6 and 8 have the highest validation scores. The differences between the training and validation metrics are also considered acceptable for both. Since the value for the hyperparameter eta of the model index 8 corresponds to the default value (0.3), this model is considered the optimal model for simplification.

When comparing the optimal pipeline of each classifier (figure 4), the difference between training and test scores is considered acceptable for Naïve Bayes, so that no overfitting is assumed. Naïve Bayes has higher test scores for both metrics (accuracy and F1 score) compared to XGBoost, which also provides acceptable but worse test results. Since Naïve Bayes is also the less complex model, together with its combination of vectorizer, pre-processing function and hyperparameter setting (see table 3 in the Appendix for details), it is used as the final model to classify the self-crawled tweets.

Since only classified self-crawled tweets that are assigned a minimum 75 % probability of a sentiment are considered within all analyses, figure 5 expresses how the classifier is expected to perform on previously unseen data (the self-crawled tweets). An accuracy of 85 % is assumed to be sufficiently high to expect the classifier's sentiment predictions to be mostly correct.

The classification of tweets extracted via `Tweepy` shows that the proportion of positive classifications of tweets from the UK is slightly higher than for tweets from the African region. However, the difference (66.15% vs. 62.61%) is not interpreted as sufficient to assume an extremely different level of happiness among people, as the World Happiness Report shows. The classification of the data extracted via `GetOldTweets3` leads to the same conclusions, as here too the proportions of positively classified tweets do not differ significantly between the two regions over several years (for the years 2015, 2017 and 2018, the proportions of positively classified tweets are even higher for the African region). The qualitative analysis of the tweets of both regions by identifying categories does not produce the significant differences either. Interesting results, such as that tweets containing country or city names are classified as more negative by both regions and that tweets in which monetary values are identified are classified as rather positive, are transferable to both regions. However, these similar results of both regions support the argument that no extreme differences in people's happiness can be assumed.

Go et al. [4] achieve a test accuracy of 82.7 % with a Naïve Bayes classifier, without excluding tweets that are assigned too similar class probabilities. This score cannot be directly compared with the result of the Naïve Bayes classifier of this project, because Go et al. used different test data. Nevertheless, this finding could be seen as an impulse to analyse further improvement potential.

As Dodds et al. [2], Reece et al. [10] and Frank et al. [3], the two selected vectorizers (`CountVectorizer`, `TfidfVectorizer`) are exclusively induced to use unigrams as features. The choice of bigrams or even the combination of unigrams and bigrams could, as with Go et al. (test accuracy improved for Naïve Bayes from 81.3% from to 82.7%), lead to an increase in model performance.

Further pre-processing steps such as shortening of words with repeating letters, as implemented by Go et al., may also lead to positive effects. Removing links could be handled differently, by always replacing the link with the same string (for example "URL") instead. The idea here is that posting internet links could also be an expression of sentiment. Dodds et al. [2], Reece et al. [10] and Frank et al. [3] have all made use of assigned happiness scores of individual words. The used *language assessment by Mechanical Turk (labMT)* can also be used to generate further classification features.

The methodology used to decide about the final classifier can be criticized. Classified tweets that are assigned a probability of a sentiment of less than 75% are not excluded in this process. The methodology used is therefore not the same as the one used in the final application (classification of self-crawled tweets). Since about 100 models are trained in this process for each classifier, this simplification was used for time reasons.

The Sentiment140 data set provided by Go et al. [4] does not allow the classification of neutral sentiments. Since tweets can be neutral, many tweets cannot be classified correctly. Considering this problem, many classified self-crawled tweets that are assigned too similar probabilities are removed from the analysis. To use this data, it would have required a different training data set with three labels (negative, neutral, positive). In general, it must also be accepted that the labeling of the Sentiment140 data set is not done manually but based on included emoticons.

Also it must be questioned whether the used libraries `Tweepy` and `GetOldTweets3` extract reasonable tweets regarding the input query (i.e. only tweets from the specified region in the specified time period). This is controlled manually by random sampling. However, the significant difference in the proportion of positively classified tweets for the year 2019 (e.g. `Tweepy`, UK: 66.15%; `GetOldTweets3`, UK: 60%) or the significant differences in positively classified `GetOldTweets3` tweets (2018: both regions around 85%; 2019: both regions around 30% less) indicate that the extracted tweets need to be investigated more thoroughly.

The basic assumption, to measure the happiness of a group of people by the proportions of positive and negative tweets, can also be questioned. Since a quantitative assessment of happiness is not fixed, there is a lot of room for debate and different views. The chosen assessment is considered reasonable in this project.

7. Conclusion

The analysis of different combinations of vectorizer, pre-processing function and classifier with its different hyperparameter settings results in the Naïve Bayes model, originally chosen as the baseline, as the final classifier to classify the self-crawled tweets. Using the example of the UK and the selected African region, the results of the performed sentiment analyses do not confirm the extreme differences in people's happiness between less and highly developed regions as presented in the World Hapiness Report.

Various potential improvements in the methodology used, as discussed in chapter 6, could lead to a more precise classification of the self-crawled tweets. In addition, the correctness of the results obtained depends partly on factors such as the correct functioning of the libraries used to extract the tweets, which could not be circumvented within the short time available.

References

[1] Tianqi Chen and Carlos Guestrin. *XGBoost: A Scalable Tree Boosting System.* 2016. DOI: 10.1145/2939672.2939785.

[2] Peter Sheridan Dodds et al. “Temporal patterns of happiness and information in a global social network: hedonometrics and Twitter”. In: *PloS one* 6.12 (2011), e26752. DOI: 10.1371/journal.pone.0026752.

[3] Morgan R. Frank et al. “Happiness and the patterns of life: a study of geolocated tweets”. In: *Scientific reports* 3 (2013), p. 2625. DOI: 10.1038/srep02625.

[4] Alec Go, Richa Bhayani, and Lei Huang. *Twitter sentiment classification using distant supervision.* Stanford, 2009. URL: https://www-cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf.

[5] Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. *The elements of statistical learning: Data mining, inference, and prediction.* Second edition, corrected at 12th printing 2017. Springer series in statistics. New York, NY: Springer, 2017. ISBN: 9780387848587.

[6] John F. Helliwell, Richard Layard, and Jeffrey D. Sachs. *World Happiness Report 2019.* New York, 2019. URL: https://worldhappiness.report/ed/2019/.

[7] Gareth James et al. *An introduction to statistical learning: With applications in R.* Springer texts in statistics. New York et al.: Springer, 2013. ISBN: 9781461471387.

[8] Nikita Patel and Saurabh Upadhyay. “Study of Various Decision Tree Pruning Methods with their Empirical Comparison in WEKA”. In: *International Journal of Computer Applications* 60.12 (2012), pp. 20–25. DOI: 10.5120/9744-4304.

[9] Andrew G. Reece and Christopher M. Danforth. “Instagram photos reveal predictive markers of depression”. In: *EPJ Data Science* 6.1 (2017), p. 157. DOI: 10.1140/epjds/s13688-017-0110-z. URL: https://doi.org/10.1140/epjds/s13688-017-0110-z.

[10] Andrew G. Reece et al. “Forecasting the onset and course of mental illness with Twitter data”. In: *Scientific reports* 7.1 (2017), p. 13006. DOI: 10.1038/s41598-017-12961-9.

Appendix

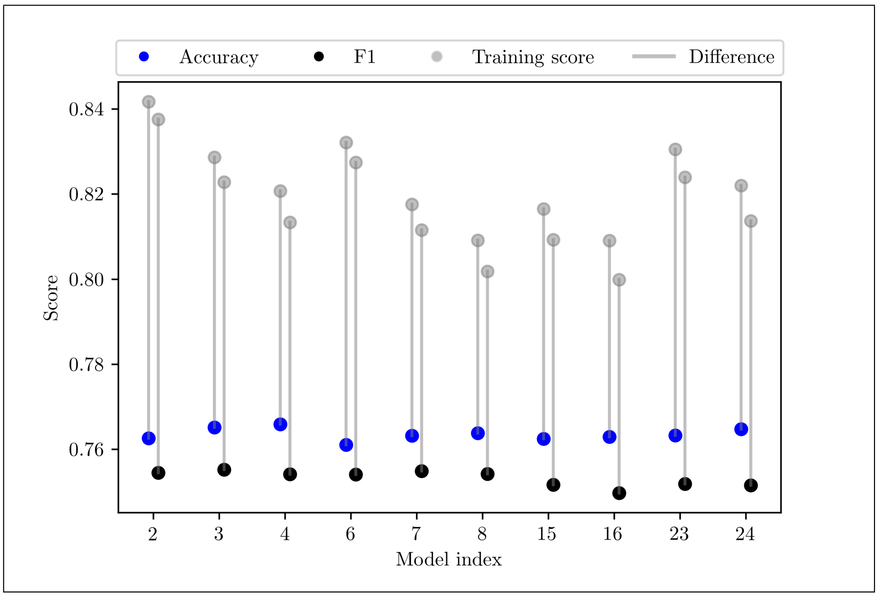

Figure 9: *Top ten pipelines for the Naive Bayes classifier. Each model index refers to a different pipeline. Both average validation and training scores from the Cross-Validation process are shown.*

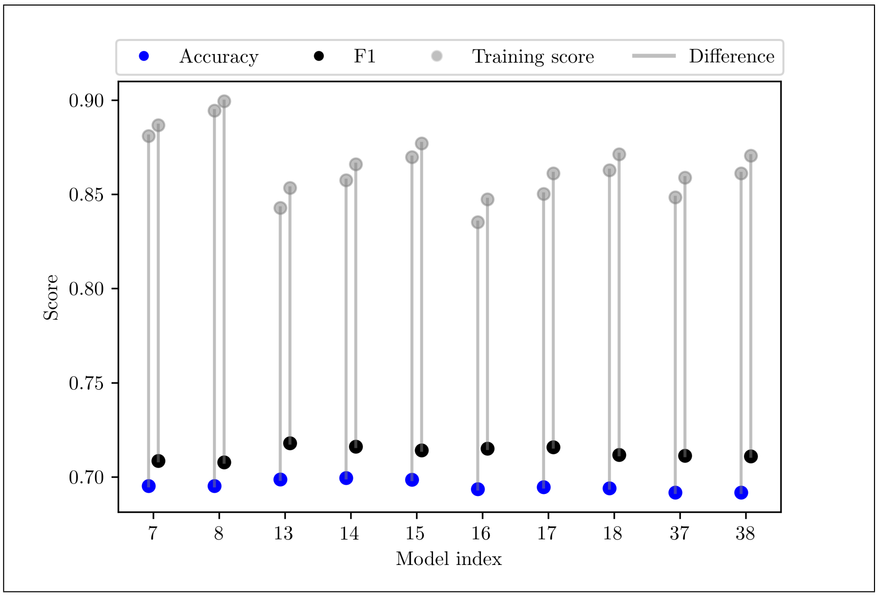

Figure 10: *Top ten pipelines for the decision tree classifier. Each model index refers to a different pipeline. Both average validation and training scores from the Cross-Validation process are shown.*

			Pre-processing			Hyperparameter setting			
Model	**Model index**	**Vectorizer**	**Lemma-tization?**	**Discard strings with non-alphabetic characters?**	**Discard stop words?**	**Alpha**	**Eta**	**Criterion**	**Max depth**
Naive Bayes	8	Count Vectorizer	Yes	No	No	3	-	-	-
Decision Tree	13	Count Vectorizer	Yes	Yes	No	-	-	Gini	45
XGBoost	8	Count Vectorizer	Yes	No	No	-	0.3	-	8

Table 3: *Description of the best pipeline for each classifier identified via Cross-validation*

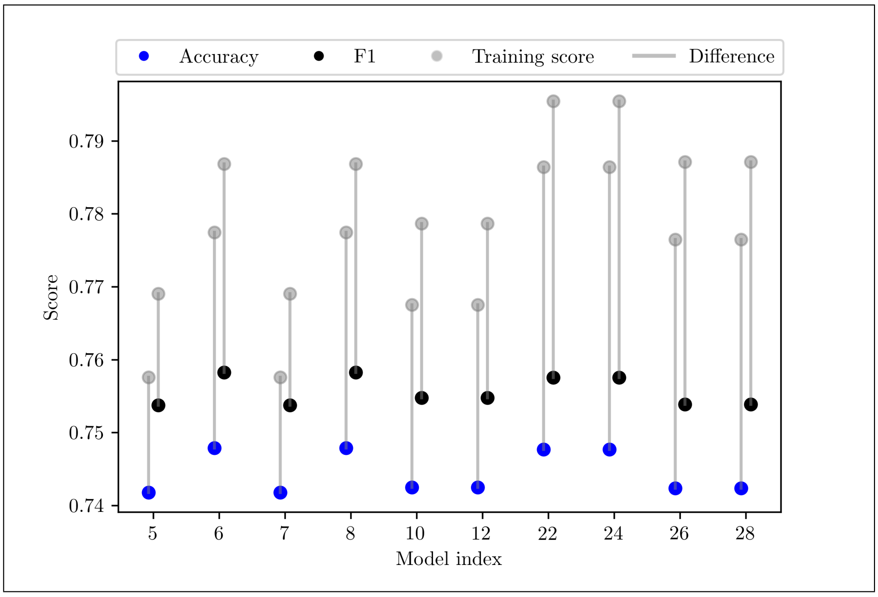

Figure 11: *Top ten pipelines for the XGBoost classifier. Each model index refers to a different pipeline. Both average validation and training scores from the Cross-Validation process are shown.*