Using dated training sets for classifying recent news articles with Naive Bayes and Support Vector Machines

- An experiment comparing the accuracy of classifications using test sets from 2005 and 2017
Abstract

Text categorisation is an important feature for organising text data and making it easier to find information on the world wide web. The categorisation of text data can be done through the use of machine learning classifiers. These classifiers need to be trained with data in order to predict a result for future input. The authors chose to investigate how accurate two classifiers are when classifying recent news articles on a classifier model that is trained with older news articles. To reach a result the authors chose the Naive Bayes and Support Vector Machine classifiers and conducted an experiment. The experiment involved training models of both classifiers with news articles from 2005 and testing the models with news articles from 2005 and 2017 to compare the results. The results showed that both classifiers did considerably worse when classifying the news articles from 2017 compared to classifying the news articles from the same year as the training data.

**Keywords:** News Articles, Machine Learning, Naive Bayes, Support vector machine, SVM, Text categorisation
Preface

We would like to thank our peer-reviewers at Linnaeus university and our supervisor Johan Hagelbäck for helpful feedback and ideas.
# Contents

1 Introduction 4
   1.1 Background 5
   1.2 Previous research 8
   1.3 Problem formulation 9
   1.4 Motivation 9
   1.5 Research Question 10
   1.6 Scope/Limitation 10
   1.7 Target group 11
   1.8 Outline 11

2 Method 12
   2.1 Tool description 12
   2.2 Dataset description 13
   2.3 Reliability and Validity 13

3 Implementation 14
   3.1 Settings 14
   3.2 Naive Bayes 14
   3.3 Support Vector Machine 14
   3.4 Application run order 15

4 Result and Analysis 17
   4.1 Naive Bayes 18
      4.1.1 Cross-validation 18
      4.1.2 Test set from 2005 19
      4.1.3 Test set from 2017 19
   4.2 Support vector machine 21
      4.2.1 Cross-validation 21
      4.2.3 Test set from 2017 23

5 Discussion 27

6 Conclusion 29
   6.1 Further research 29

References 31
1 Introduction

Text categorisation is a key feature for organising text data in today's world of online information. This feature can be used to solve multiple problems such as finding interesting information on the world wide web and automatically classifying news articles. This can be done manually, but also algorithmically through the use of text classifiers such as Naive Bayes and Support Vector Machines [1] [2].

In this paper, the authors will explore the use of the Naive Bayes and Support Vector Machine classifiers within the subject of classifying news articles. The purpose is to find out how the accuracy of the classifiers' models changes as time goes by. The authors will look at the result of a dataset with articles from the years of 2004 and 2005 and then use that model with modern day articles to see how the accuracy changes.

1.1 Background

Using machine learning, programs and computers have the ability to learn without being explicitly programmed. Machine learning can recognise patterns using specific algorithms to make predictions on a dataset. The methods to teach an algorithm is either through a supervised or an unsupervised learning method [3].

Unsupervised learning does not use a training dataset to learn. Instead, it uses unlabeled data to describe the structure of it. Therefore it does not have an expected output as there are no correct answers and the method does not use a teacher. The algorithms within unsupervised learning are left to discover and present any interesting patterns they may find within the data.

Supervised learning uses a training dataset to learn patterns. The learning process within supervised learning consists of inserting data and expecting a specific result. During the learning process, the supervisor corrects the algorithm if it makes a faulty prediction. When the algorithm performs at an acceptable level, the learning stops. At this point, the optimal scenario is that the algorithm can be used in order to correctly determine and classify unseen instances of data. Two techniques within supervised learning that have been used for classifying texts are the Naive Bayes classifier [1] as well as Support Vector Machines [4].

The Naive Bayes classifier is a technique that uses probabilities to make a prediction. This algorithm assumes that the probability of each element belonging to a given category value is independent of all other elements. A Naive Bayes model is easy to build and can be trained efficiently in a supervised learning setting [5]. This makes it useful for large
datasets [5] [6]. To describe the probability of an event the Naive Bayes classifiers are based on a rule called Bayes’ theorem:

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

To explain how this could work in the case of categorising news articles an example using 200 articles and an article that simply states “Microsoft patches serious bug”. For each of the 200 articles, each word is counted and saved in a frequency table. This frequency table can then be turned into a probability table:

<table>
<thead>
<tr>
<th>Class</th>
<th>Microsoft</th>
<th>patches</th>
<th>serious</th>
<th>bug</th>
<th>Total</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>34</td>
<td>15</td>
<td>5</td>
<td>45</td>
<td>99</td>
<td>= 99/200 → ~0.5</td>
</tr>
<tr>
<td>Sport</td>
<td>0</td>
<td>8</td>
<td>10</td>
<td>0</td>
<td>18</td>
<td>= 18/200 → 0.09</td>
</tr>
<tr>
<td>Entertainment</td>
<td>12</td>
<td>0</td>
<td>23</td>
<td>0</td>
<td>35</td>
<td>= 35/200 → ~0.18</td>
</tr>
<tr>
<td>Politics</td>
<td>4</td>
<td>0</td>
<td>56</td>
<td>0</td>
<td>60</td>
<td>= 60/200 → 0.3</td>
</tr>
<tr>
<td>Business</td>
<td>15</td>
<td>2</td>
<td>49</td>
<td>2</td>
<td>68</td>
<td>= 68/200 → 0.34</td>
</tr>
<tr>
<td>Probability</td>
<td>65/200 → 0.325 ~ 0.33</td>
<td>25/200 → 0.125 ~ 0.13</td>
<td>143/200 → 0.715 ~ 0.72</td>
<td>47/200 → 0.235 ~ 0.24</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1.1: Probability table of the words in the phrase “Microsoft patches serious bug” based on 200 articles

For this example, we will only look at the likelihood whether the phrase “Microsoft patches serious bug” most likely belongs to the “Technology” or the “Business” category. Table 1.1 shows that out of the 65 times that the name “Microsoft” was mentioned, it was mentioned 34 times in articles categorised as “Technology” and 15 times it was mentioned in articles categorised as Business. To classify the phrase, the Bayes’ theorem is used to calculate the probability of the article belonging to each category.

The probability that the phrase “Microsoft patches serious bug” belongs to the “Technology” category:

$$P(Tech | Microsoft, patches, serious, bug)$$

$$= \frac{P(Microsoft|Tech)P(patches|Tech)P(serious|Tech)P(bug|Tech)P(Tech)}{P(Microsoft)P(patches)P(serious)P(Bug)}$$

$$= \frac{0.34 \times 0.15 \times 0.05 \times 0.45 \times 0.50}{0.33 \times 0.13 \times 0.72 \times 0.24} = 0.0773965…$$
The probability that the phrase “Microsoft patches serious bug” belongs to the “Business” category:

\[
P(\text{Business} | \text{Microsoft, patches, serious, bug})
\]

\[
= \frac{P(\text{Microsoft})P(\text{patches})P(\text{serious})P(\text{Business})P(\text{bug})P(\text{Business})}{P(\text{Microsoft})P(\text{patches})P(\text{serious})P(\text{Business})}
\]

\[
= \frac{0.22 \times 0.03 \times 0.72 \times 0.03 \times 0.34}{0.33 \times 0.13 \times 0.72 \times 0.24} = 0.0065384\ldots
\]

Since the result of the Bayes’ theorem is greater for “Technology” than “Business” it would classify the article as “Technology”.

The other set of classifiers, Support Vector Machines, SVMs for short, are given a set of training examples, each marked as belonging to a specific class, that is used to build a model that assigns new examples to a category. When a new example is given, the classifier compares it to what it learned from the training set in order to make an assumption. SVMs are built on finding a hyperplane that divides a dataset into classes. An SVM model represents the training examples as points in space, mapped separate from other categories, and a classifier represented by a hyperplane separating the points. The two points closest to the hyperplane are the so-called support vectors. The further away from the hyperplane that the support vectors are the more confident we can be that the data are correctly identified. When new data is inserted, whatever side of the hyperplane it falls decides what category it is assigned. For the classifier to be optimised we want the hyperplane to be placed as far away as possible from the points while still correctly separating them [7].

Fig 1.2. Red and blue points represents different categories from the training set and the hyperplane represents the classifier. The left, unoptimised, figure works for the training set but may prove inaccurate for many new examples as the margin is small. The right, optimised, figure have a bigger margin and will, most likely, be more accurate.
The result of a supervised learning method can vary greatly based on how the learning data have been pre-processed. Data pre-processing is a technique for transforming raw data, such as plain text, into a more comprehensible format. A common practice in data preprocessing is to remove certain words that are common and repeatable (called stop words) [8]. The product of the data pre-processing is the final dataset used for training the algorithms, the quality of which will significantly affect the accuracy and result of the classifications made.

Document classification is the task of assigning a document one or multiple categories [4]. This task is important for news sites as they often categorise their articles within a category such as sports, business, entertainment and politics. Machine learning has already made its way to being used for automating the classification of news articles. For example, google news uses machine learning to algorithmically find and recommend news stories for their users [9] [10] [11].

The accuracy of a classification can be measured with True positives (TP), True negatives (TN), False positives (FP) and false negatives (FN). By evaluating this data in a confusion matrix [12], it is possible to see what has been classified correctly and faulty.

1.2 Previous research

Previous research regarding automatic text classification using machine learning are plenty. Texts of multiple different languages have been classified using machine learning [4] [13]. The authors could not find any research about how different algorithms’ accuracy change when classifying current news articles with a model trained with dated articles. However, the previous research gives a good overview of what did and what did not work well in terms of classifiers and settings.

T. Joachims conducted an experiment, on Reuters-21578 test collection for text categorization, with five different classifiers: NB, Rocchio, C4.5, k-NN, and SVM [4]. The SVMs were tested twice, once using a polynomial kernel setting and once using a radial basis function (RBF) setting. The experiment showed that the SVMs were more accurate than the other classifiers. Using an RBF setting, SVM reached a combined accuracy of 86.4% and using a polynomial kernel setting it reached a combined accuracy of 86.0%. These results are better than the four other classifiers. The second most accurate classifier was k-NN which achieved an accuracy of 82.3% while Naive Bayes did worst with an accuracy of 72.0%

To avoid a too large feature vector, Joachims considered words that occurred at least three times in a text to be a feature. This consideration resulted in a set of 9962 features. The experiment showed that both SVM and NB
performed best in accuracy when using all features, while classifiers like C4.5 achieved the highest accuracy at 1000 features.

S. Alsaleem made a similar study comparing only SVM and NB classifiers on arabic news articles [13]. The dataset he used contained 5121 articles divided into roughly 730 articles over eight categories. Alsaleem’s experiment resulted in an accuracy of 77.9% for SVM and 74.1% for Naive Bayes. SVM outperformed NB on six of the eight categories.

1.3 Problem formulation

The goal of this study is to determine how the accuracy of Support Vector Machine and Naive Bayes models change as the time goes on. This is done by studying the resulting accuracy of a given model when the data of the test set is from the same time period as the dataset, compared to when the test set is from a later time period.

The dataset [14] contains 2225 articles from the British newspaper BBC that was collected in the year 2005. The accuracy of the model using the old data will be classified using 80% of the dataset as the training set and 20% as the test set, the test set containing newer articles will be collected and used on the same training set.

The result in accuracy of the different classifications will hopefully show if there is any difference in accuracy between the dated set and the newly created set. The result will therefore show which one of the two classifiers is the most accurate and most optimal over time.

The most commonly preferred type of algorithm for automating classification of texts is supervised learning. Two algorithms that are commonly used for automatic classification of a text are Support Vector Machine and Naive Bayes Classifier [1] [2]. Each classifier is sensitive to parameter optimisation which along with different datasets can cause varying results.

1.4 Motivation

Automatically classifying news articles can be used by news agencies to simplify categorisation on their sites, such as, uploading a new article and the system labels it in a category without the writer specifying it. The automatic categorisation of news articles could also be used by third party sites that gather news articles from multiple sources. If the accuracy of the automatic classifications is good enough, the news outlets can use it to easier categorise their articles and the third party sites can use it in order to avoid mislabeled articles. In their choice of classifier they may need to think about how much
maintenance is needed. Therefore, knowing how the classifier handles newer data without having been updated to specifically handle new terms.

The classification of news articles can be a challenging task as there are a lot of variables that can greatly affect the outcome and result of the classification such as what algorithm is being used, settings being used, how the pre-processing of the texts are done as well as what dataset is used.

Naive Bayes classifiers are a very popular method for text classification and are often the go to classifier when you want to categorise texts [1]. Another classification technique that has been used for classification of texts is SVM [2] [4]. The authors found these two classifiers to be ideal subjects in order to find out how you should set up a machine learning classifier if you are looking for the most accurate results regarding TP, TN, FP and FN.

1.5 Research Question

| RQ1                  | How accurate are Naive Bayes and Support Vector Machine when classifying recent news articles on a model trained with dated articles? |

1.6 Scope/Limitation

There are other classifiers that also are interesting in this particular topic but due to the limited amount of time, the authors have decided to limit the study to evaluate NBC and SVM.

The dataset the authors chose as the training set consists of 2225 unprocessed news articles which are divided into five different categories: politics, tech, sports, entertainment, and business. The authors believe that the amount of data will be sufficient for this study. Therefore this dataset won’t be broadened during the study.

The test set containing the news articles from 2017 were limited to approximately 400 articles that were divided equally into the five categories. The number of articles is set to 400 since it is about 20% of the old dataset and collecting more than that will be too time-consuming.

Due to the limited amount of time and resources, the lightweight application will use the WEKA [15] framework. WEKA does not support SVM without using a wrapper. Therefore, a wrapper class [16] for LIBSVM [17] in WEKA will be used.

The limited time will also put a constraint on how much the authors are able to test and do. SVM and Naive Bayes have a lot of different settings that can be used and there is no way near enough time to test and report
every single combination. Therefore, the authors will only select the settings they find to be the most important and relevant for the result.

1.7 Target group

This study may prove interesting for developers and news agencies that are looking at the possibility of integrating machine learning in news services. Hopefully, the study will provide the developers and news agencies with a good overview of how accurate Naive Bayes and SVMs may be after years of no maintenance. If the study is successful it could be helpful for developers to decide what algorithm they should use.

1.8 Outline

The following method and implementation chapters describe which method, techniques, and settings the authors used to generate a result. The result and analysis chapter describes the results the authors achieved and the analyses made of the result. The chapter discussion contains the author's own thoughts about the result, and the final chapter describes the authors’ conclusions of the study and their thoughts for further research within the area.
2 Method

In the study, an experiment with one dependent and one independent variable was conducted to reach a result. The independent variable is the dataset which is the input to the classifiers. The dependent variable is the accuracy of a given classifier. The accuracy of a classification was evaluated by the percentage of correctly classified instances.

To calculate the accuracy of the trained classifiers based on the dataset, the authors decided to create an application using LIBSVM and WEKA. This application trains the classifier and then tests it with a given test set.

The experiment was executed in two stages. The first stage looked at the accuracy of the classifiers when the test set was from the same time period as the dataset. This was done by first getting an average result of the classifiers by using a cross-validation method and then by using a percentage split method. The second and last stage consisted of testing the new test set, containing 473 articles from 2017, on the old dataset. When the two stages were complete the results were evaluated and presented.

2.1 Tool description

Two machine learning tools have been used in this project: Weka and LIBSVM. Weka is a free software licensed under the GNU General Public License that has been developed at the University of Waikato in New Zealand. Weka provides a set of algorithms and tools that can be used for analysing data and creating predictive models. 

LIBSVM is a popular open source library developed at the National Taiwan University for machine learning used to implement algorithms for support vector machines. LIBSVM is released and licensed under the BSD license.

The WEKA and LIBSVM frameworks allowed the authors to experiment with the two classifiers and simplified the implementation phase since no algorithms and settings had to be written from scratch.

In both frameworks and for both classifiers cross-validation will be used to get a more accurate result. Cross-validation runs and tests the classifier several times and the test data with part of the training data through each iteration. This will give an average measurement of the classifier instead of only measuring one time with just one set of test data and therefore increase the validity of the reported accuracy.
2.2 Dataset description

The SVM and Naive Bayes models were given data from a dataset of 2225 documents from the BBC news website in five areas from 2004-2005 [14]. The models were fed and tested with different amount of data from the 2225 documents by using cross-validation and a percentage split method. The results of cross-validation showed how accurate the model is on average and the percentage split divided the 2225 documents into one training set and one test set. The training set was used to train two identical models that were tested with two different test sets, one retrieved from the percentage split and a newly created test set.

The newly created test set consists of 473 articles from 2017 that were collected from BBC news with a web scraper made by the authors. The newly created test set and the test set provided by the percentage split contains the same amount of articles.

There are plenty of different settings for the models as well as the pre-processing of data that can be applied. A setting could be only using words that occur more than three times and to filter out common words, such as stop words, that are used in all types of articles as they could influence the result in a negative way. The number of documents used to teach the models, the settings used for pre-processing and the settings used for the algorithms will be further detailed in the implementation chapter of the essay.

2.3 Reliability and Validity

The collection of a dataset or test set brings the issue of how the articles are chosen. If the collection is done manually there could be a bias of what the collector thinks will do well. To circumvent this, the authors only chose the latest articles from the categories present in the old dataset and did not consider the content of the articles. This was done automatically by a web scraper, making the bias of picking articles no issue.

Another issue regarding the experiment is that all articles are from the same news agency. The result is therefore not representative for all English news agencies. In order to show a more general result, the experiment has to be conducted using other news agencies articles.
3 Implementation

This section will describe the different settings that were used during the pre-processing of the articles as well as describing any classifier settings that have been used unless the default values were used.

3.1 Settings

Two different preprocessing settings were used for each classification:

**Tokenizer** - Tokenization is a process in which you break down a stream of text into meaningful elements called tokens. These tokens can be words, phrases, symbols, and more. The built-in tokenizer setting “word-tokenizer” was used, which delimits the content of the text when one of the following characters appears: “ \n\t.;::"()?!

**Stopword Removal** - Stop words occur frequently throughout almost every document. These are words that are meaningless when used for classifying texts as they are only used to join other words together in a sentence. A stop word does not contribute to the context of the text and as such may only cause confusion for the classifier. To remove stop words the stopword handler “Rainbow” was used to remove words such as: “and”, “or”, “it”, “for”, and “a”.

**NGRAM** - Is a setting for a classifier which sets a sequence of words to one attribute, but since Alsaleem did not use NGRAM in his experiment [13] and still achieved a high accuracy, the decision to not use the setting was made. This means that one word was considered as one attribute during the classification.

3.2 Naive Bayes

The Naive Bayes implementation in the WEKA framework was used in the application. The Naive Bayes classifiers give a result based on probability. In the case of this implementation, the Naive Bayes classifier will calculate the probability of an article belonging to a specific category and then label the article to the one that it most likely belongs to.

3.3 Support Vector Machine

As previously described in the introduction, a Support Vector Machine is built on finding a hyperplane that best divides a dataset into its different categories. SVMs are based on Structural Risk Minimization(SRM) principle
[18]. The LIBSVM framework for WEKA provided SVM implementation solutions which significantly simplified the SVM implementation in the Java application. LIBSVM provided an easy solution to implement pre-existing SVM algorithms, with multiple different classifier settings that could be changed to reach an acceptable accuracy. The authors noticed that the default classifier settings reached an acceptable result of 94%. The authors decided, however, to slightly change the gamma setting to “7”, reaching an even higher percentage of about 96%. The remaining twelve settings of the SVM classifier were the default values for the LIBSVM framework [17].

3.4 Application run order

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Train</th>
<th>Test</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive Bayes</td>
<td>2005 articles</td>
<td>2005 articles</td>
<td>Cross-validation(5 Folds)</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>2005 articles</td>
<td>2005 articles</td>
<td>Train with 78.75% of the 2005 dataset, test with rest</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>2005 articles</td>
<td>2017 articles</td>
<td>Train with 78.75% of the 2005 dataset, test with 2017 dataset</td>
</tr>
<tr>
<td>SVM</td>
<td>2005 articles</td>
<td>2005 articles</td>
<td>Cross-validation(5 Folds)</td>
</tr>
<tr>
<td>SVM</td>
<td>2005 articles</td>
<td>2005 articles</td>
<td>Train 78.75% of the 2005 dataset, test with rest</td>
</tr>
<tr>
<td>SVM</td>
<td>2005 articles</td>
<td>2017 articles</td>
<td>Train with 78.75% of the 2005 dataset, test with 2017 dataset</td>
</tr>
</tbody>
</table>

Table 3.1 Experiment procedure.

To gather the results the classifiers were executed three times each with different model testing methods.
The first time a classifier's model is created, it is tested using cross-validation. Cross-validation is used for evaluating a model to give a more reliable accuracy of the selected options and training set. This method will train and test the entire dataset. If our dataset is applied to the 5-cross-fold example in figure 3.1, the first fold trains the first 20% of the articles and test the rest. For the second fold, it trains the next 20% and tests the rest etc.

The first run used cross-validation and was done to evaluate the average accuracy of the classifier and its settings. The second and third run will use the same training set, this way the trained models will be the same but tested with different data, one time with data from 2005 and one time with data from 2017.

The second time a classifier’s model is created and tested, percentage split is used using the test set from 2005. The percentage split will split the dataset into a training set that consists of 78.75% of the articles as a training set and the remaining 21.25% are used as a test set.

The third and last time a classifier’s model is created and tested it will use 78.75% of the original dataset from 2005 as the training set. Unlike the second time, the test set is now replaced by an equal amount of articles from 2017, instead of the original from 2005.
4 Result and Analysis

The result is presented in two different sections, one for each classifier. Each classifier result is divided into three sections based on the methods; cross-validation, percentage-split, and 2017 articles as test set. Each result contains:

- Correctly classified instances (amount and percentage)
- Incorrectly classified instances (amount and percentage)
- Confusion matrix, which displays what label has been classified to which category

The method percentage-split will be referred to as 2005 test set since it has exactly the same training data as the 2017 test set.

A confusion matrix is created for every test. The confusion matrix provides a table layout that can be used to describe the performance of a classifier. Each column in a confusion matrix shows the total amount of articles predicted as a class while each row shows the result of an actual class. Each classification for a given class can be either TP(True Positive), TN(True Negative), FP(False Positive) or FN(False negative). TP are the sum of elements classified as the correct class. Excluding the TP, the sum of the values in a class’ row represents the FN while the sum of the values in a class’ column represents the FP for that class. The TN for a class is the sum of all the columns and rows in the matrix, excluding the column and row for the class.
4.1 Naive Bayes

The classifications of Naive Bayes are divided into three subcategories.

4.1.1 Cross-validation

The result of cross validating the Naive Bayes model with five folds.

Using cross validation, 2225 articles were classified. Out of the 2225 articles, 2114 (~95.01%) were assigned the correct class and 111 (~4.99%) were assigned an incorrect class:

- 401 articles were originally labeled as technology, 380 of which were classified as technology, 8 as business, 6 as politics, 5 as entertainment, and 2 as sport.
- 511 articles were originally labeled as sport, 503 of which were classified as sport, 4 as politics, 2 as business, and 2 as entertainment.
- 510 articles were originally labeled as business, 481 of which were classified as business, 16 as politics, and 13 as technology.
- 386 articles were originally labeled as entertainment, 348 of which were classified as entertainment, 17 as technology, 11 as business and 10 as politics.
- 417 articles were originally classified as politics, 402 of which were classified as politics, 8 as business, 4 as technology, 2 as sport and 1 as entertainment.

<table>
<thead>
<tr>
<th>Classified as</th>
<th>Technology</th>
<th>Sport</th>
<th>Business</th>
<th>Entertainment</th>
<th>Politics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>380</td>
<td>2</td>
<td>8</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Sport</td>
<td>0</td>
<td>503</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Business</td>
<td>13</td>
<td>0</td>
<td>481</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>Entertainment</td>
<td>17</td>
<td>0</td>
<td>11</td>
<td>348</td>
<td>10</td>
</tr>
<tr>
<td>Politics</td>
<td>4</td>
<td>2</td>
<td>8</td>
<td>1</td>
<td>402</td>
</tr>
</tbody>
</table>

Fig 4.1 Confusion Matrix of the Naive bayes classification with cross-validation method.
4.1.2 Test set from 2005

The result of using a percentage-split on the Naive Bayes model. The percentage-split is set to train on 78.75% of the dataset, then test on the remaining 21.25%.

<table>
<thead>
<tr>
<th>Classified as</th>
<th>Technology</th>
<th>Sport</th>
<th>Business</th>
<th>Entertainment</th>
<th>Politics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>90</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Sport</td>
<td>0</td>
<td>68</td>
<td>3</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>Business</td>
<td>1</td>
<td>0</td>
<td>94</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Entertainment</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>95</td>
<td>1</td>
</tr>
<tr>
<td>Politics</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>88</td>
</tr>
</tbody>
</table>

Fig 4.2 Confusion Matrix of the Naive Bayes classification with the percentage-split method.

Using the test set from 2005, 473 articles were classified with a model trained using 1747 articles. Out of the 473 articles, 435 (~91.97%) were assigned the correct class and 38 (~8.03%) were assigned an incorrect class:

- 97 articles were originally labeled as technology, 90 of which were classified as technology, 3 as politics, 2 as business, and 2 as entertainment.
- 86 articles were originally labeled as sport, 68 of which were classified as sport, 9 as entertainment, 6 as politics, and 3 as business.
- 96 articles were originally labeled as business, 94 of which were classified as business, 1 as politics, and 1 as technology.
- 100 articles were originally labeled as entertainment, 95 of which were classified as entertainment, 3 as politics, 2 as business and 2 as technology.
- 94 articles were originally classified as politics, 88 of which were classified as politics, and 6 as business.

4.1.3 Test set from 2017

The result of testing the 2017 dataset on the Naive Bayes model. The classifier is set to train 78.75% of the 2005 dataset, then test the 2017 dataset.
Using the test set from 2017, 473 articles were classified with a model trained using 1747 articles. Out of the 473 articles, 380 (~80.34%) were assigned the correct class and 93 (~19.66%) were assigned an incorrect class:

- 97 articles were originally labeled as technology, 80 of which were classified as technology, 9 as business, 6 as entertainment, and 2 as politics.
- 86 articles were originally labeled as sport, 71 of which were classified as sport, 7 as politics, 4 as entertainment, 3 as technology, and 1 as business.
- 96 articles were originally labeled as business, 61 of which were classified as business, 21 as technology, and 14 as politics.
- 100 articles were originally labeled as entertainment, 80 of which were classified as entertainment, 10 as technology, 7 as politics, 2 as sport, and 1 as business.
- 94 articles were originally labeled as politics, 88 of which were classified as politics, 3 as business, 2 as technology, and 1 as entertainment.
4.2 Support vector machine
The classifications of Support vector machine are divided into three subcategories.

4.2.1 Cross-validation
The results of cross validating the Naive Bayes model with five folds.

<table>
<thead>
<tr>
<th></th>
<th>Technology</th>
<th>Sport</th>
<th>Business</th>
<th>Entertainment</th>
<th>Politics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>384</td>
<td>2</td>
<td>8</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>Sport</td>
<td>0</td>
<td>506</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Business</td>
<td>8</td>
<td>0</td>
<td>488</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Entertainment</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>376</td>
<td>3</td>
</tr>
<tr>
<td>Politics</td>
<td>1</td>
<td>2</td>
<td>11</td>
<td>3</td>
<td>400</td>
</tr>
</tbody>
</table>

Fig 4.4 Confusion Matrix of the SVM classification with cross-validation method.

Using cross validation, 2225 articles were classified. Out of the 2225 articles, 2154 (~96.81%) were assigned the correct class and 71 (~3.19%) were assigned an incorrect class:

- 401 articles were originally labeled as technology, 384 of which were classified as technology, 8 as business, 6 as entertainment, 2 as sport, and 1 as politics.
- 511 articles were originally labeled as sport, 506 of which were classified as sport, 2 as business, 2 as politics, and 1 as entertainment.
- 510 articles were originally labeled as business, 488 of which were classified as business, 10 as politics, 8 as technology, and 4 as entertainment.
- 386 articles were originally labeled as entertainment, 376 of which were classified as entertainment, 4 as business, 3 as technology, and 3 as politics.
- 417 articles were originally labeled as politics, 400 of which were classified as politics, 11 as business, 3 as entertainment, 2 as sport, and 1 as technology.
4.2.2 Test set from 2005

The results of using a percentage-split on the SVM model. The percentage-split is set to train on 78.75% of the dataset, then test on the remaining 21.25%.

<table>
<thead>
<tr>
<th>Classified as →</th>
<th>Technology</th>
<th>Sport</th>
<th>Business</th>
<th>Entertainment</th>
<th>Politics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>90</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Sport</td>
<td>0</td>
<td>78</td>
<td>1</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>Business</td>
<td>2</td>
<td>0</td>
<td>93</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Entertainment</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>97</td>
<td>1</td>
</tr>
<tr>
<td>Politics</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>2</td>
<td>86</td>
</tr>
</tbody>
</table>

Using the test set from 2005, 473 articles were classified with a model trained using 1747 articles. Out of the 473 articles, 444 (~93.87%) were assigned the correct class and 29 (~6.13%) were assigned an incorrect class:

- 97 articles were originally labeled as technology, 90 of which were classified as technology, 4 as business, 2 as entertainment, and 1 as politics.
- 86 articles were originally labeled as sport, 76 of which were classified as sport, 6 as entertainment, 1 as business, and 1 as politics.
- 96 articles were originally labeled as business, 93 of which were classified as business, 2 as technology, and 1 as politics.
- 100 articles were originally labeled as entertainment, 97 of which were classified as entertainment, 1 as business, 1 as technology, and 1 as politics.
- 94 articles were originally labeled as politics, 86 of which were classified as politics, 6 as business, 2 as entertainment.
4.2.3 Test set from 2017

The result of testing the 2017 dataset on the SVM model. The classifier is set to train 78.75% of the 2005 dataset, then test the 2017 dataset.

<table>
<thead>
<tr>
<th>Classified as</th>
<th>Technology</th>
<th>Sport</th>
<th>Business</th>
<th>Entertainment</th>
<th>Politics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>66</td>
<td>1</td>
<td>18</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>Sport</td>
<td>2</td>
<td>78</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Business</td>
<td>11</td>
<td>0</td>
<td>72</td>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>Entertainment</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>85</td>
<td>5</td>
</tr>
<tr>
<td>Politics</td>
<td>2</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>86</td>
</tr>
</tbody>
</table>

Fig 4.6 Confusion Matrix of the SVM classification with the 2017 articles as test set.

Using the test set from 2017, 473 articles were classified with a model trained using 1747 articles. Out of the 473 articles, 387 (~81.82%) were assigned the correct class and 86 (~18.18%) were assigned an incorrect class:

- 97 articles were originally labeled as technology, 66 of which were classified as technology, 18 as business, 8 as entertainment, 4 as politics, and 1 as sport.
- 86 articles were originally labeled as sport, 78 of which were classified as sport, 3 as entertainment, 3 as politics, and 2 as technology.
- 96 articles were originally labeled business, 72 of which were classified as business, 11 as technology, 11 as politics, and 2 as entertainment.
- 100 articles were originally labeled as entertainment, 85 of which were classified as entertainment, 5 as politics, 4 as sport, 3 as technology and 3 as business.
- 94 articles were originally labeled as politics, 86 of which were classified as politics, 6 as business, and 2 as technology.
4.3 Summary of the results

The cross-validation results from the classifiers show that the models are very accurate with 96.81% accuracy for SVM and 95.01% for Naive Bayes. However, these results cannot be compared directly with the two other methods since they use different training and test data.

<table>
<thead>
<tr>
<th></th>
<th>Naive Bayes Correctly classified instances(%)</th>
<th>SVM Correctly classified instances(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2005 test set</td>
<td>91.97%</td>
<td>93.87%</td>
</tr>
<tr>
<td>2017 test set</td>
<td>80.35%</td>
<td>81.82%</td>
</tr>
<tr>
<td>Change(%)</td>
<td>-12.63%</td>
<td>-12.83%</td>
</tr>
</tbody>
</table>

Fig 4.7 Accuracy change between 2005 test set and 2017 test set using NB and SVM.

As seen in figure 4.7, the result shows a clear decrease in accuracy when using the test set from 2017 when compared to the test set from the same year as the training set. Both classifiers experienced a similar drop in accuracy, SVM with a 12.83% drop and Naive Bayes with a 12.63%. This shows that the average quality of the models deteriorates at a similar rate. However, some of the specific results of each category for Naive Bayes and SVM varied greatly.
As seen in figure 4.8, the Naive Bayes classifier and the SVM had difficulties classifying articles from 2017 compared to those from 2005, especially in some categories like business and technology.

When classifying technology using the original test set from 2005, both SVM and Naive Bayes showed the same accuracy of 92.78%, however, when the newer test set from 2017 was used the result differed greatly. The accuracy of Naive Bayes decreased to an adequate 82.45% and SVM accuracy decreased to a low 68.04%.

For the business category, the results using the test set from 2005, both classifiers had roughly the same accuracy, Naive Bayes 97.91%, and SVM 96.88%. Similar to technology, using the test set from 2017 resulted in two rather different accuracies. Unlike technology, the results shifted. SVM was affected least by the change of test set with an accuracy of 75% while Naive Bayes only reached an accuracy of 63.54%.

The entertainment category was classified with roughly the same accuracy using the 2005 test set. Naive Bayes had 95% and SVM 97%.
Similar to the classifications of business and sport, the accuracy dropped using the 2017 test set. SVM achieved an accuracy of 85% and Naive Bayes 80%.

The results for each classifier and test set differed the least for the politics category. Both classifiers achieved an accuracy of 93.62% using the original test set from 2005. Both classifiers also achieved the same result when classifying the test set from 2017, with an accuracy of 91.49%.

When classifying sports articles, Naive Bayes achieved a higher accuracy of 82.56% when classifying articles from 2017 compared to 79.07% when classifying the older articles from 2005. The accuracy of SVM did not differ from changing between the two test sets, both resulting in 90.70% accuracy.
5 Discussion

The result of this study clearly shows an overall decrease in accuracy for both the Naive Bayes and the Support Vector Machine classifier when classifying newer articles with a model using an old dataset. The drop in accuracy for the Naive Bayes was about the same for Support Vector Machine which shows that both classifiers had about the same performance for both the new and the old test sets.

As seen in the result, the classifier's accuracy differed a lot regarding classifying specific categories. Technology and business were the two categories where the result of both classifiers differed the most. SVM did a lot worse than Naive Bayes when classifying the new technology articles and Naive Bayes did a lot worse than SVM when classifying the new business articles.

The classifiers experienced a lot of difficulties when classifying technology and business articles. Technology articles were often incorrectly classified as business, and business articles were often classified as technology. A likely reason for this is that both categories are closely related. This can be seen on BBC's website where the page for business contains some technology articles [19].

As the two test sets only contain 473 articles, the margin of error is rather large. The size of the test sets could have affected the result negatively and could be an explanation to why Naive Bayes was more accurate at classifying newer sports articles. Further testing is needed to verify the result of the classifiers, but this was not possible with the limited amount of time in this study.

We believe that trends in the news world could have had an impact on the result. If there are a lot of articles about a current event such as a sudden death of a celebrity and the classifiers would incorrectly classify the first article of this topic, it is likely that the rest of articles regarding this topic would also be incorrectly classified. This problem could of course also occur during longer periods, such as the result of the 2016 US presidential election.

As a classifier's result vary depending on the test set, the result of the study may have been affected by many articles writing about the same occurrence or event. Current major events could be widespread throughout
the test, such as Donald Trump becoming the new president of the United States and Great Britain leaving the European Union. The 2017 test set only contains articles written between January and March which enhances the issue of the same events being widespread throughout the test set. To reduce this possible negative effect and achieve a more representative result the test set could have been enlarged and contained articles from a longer time period, such as the 2005 dataset.

As both Naive Bayes and SVM reached a similar performance in terms of accuracy, we can see a clear advantage in using an updateable classifier such as Naive Bayes over SVM. The possibility of training the Naive Bayes model with new training data without access to the old data makes it a lot easier to maintain an accurate model when compared to SVM.
6 Conclusion
The goal of this study was to find out how accurate the Naive Bayes and Support Vector Machine classifiers are when classifying new news articles on a model that is trained with old news articles.

The study shows that the accuracy of both the Naive Bayes and Support Vector Machine classifiers is lower when classifying new articles on a classifier trained with old news articles compared to classifying articles from the same period as the training set.

The study also shows that some categories were more likely to be incorrectly classified than others by both Naive Bayes and Support Vector Machine. This was seen in the result where the classifiers' accuracy for technology, business, and entertainment were affected a lot more than sport and politics. The result for sports and politics were barely affected at all by the change of test set.

To increase the validity of the result we would have liked to use a larger dataset, both for training and testing the classifiers. We believe that the relatively small test set of 478 articles can be affected too much by current events. Using a larger test set would ensure that the result would be more representative of reality.

The authors believe this study can be helpful for news agencies which have an interest in adopting machine learning into their system. The results provide information on how the classifiers differ in accuracy when categorising news articles and shows which categories that are harder to classify correctly. As the study shows a similar result in accuracy for both Naive Bayes and SVM, it gives Naive Bayes an advantage as it can be continuously trained over time to maintain an acceptable accuracy.

6.1 Further research
For further research on this topic, we would recommend expanding the dataset size to increase the validity of the result and hopefully decrease the amount of incorrectly classified instances.

This study was limited to BBC news only which means that the result represents one source. By collecting articles from several news sites, the result could be more general. However, this could cause more issues
regarding the categories since all news sites don't categorise their content the same way.

In the current results, we can see that the overall accuracy of the two classifiers is similar but the accuracy for some categories are very different. Therefore, an interesting topic to further research upon would be to find out why Naive Bayes and SVMs results deviated from each other on some categories, such as business and technology.

Another interesting direction for further research would be to evaluate the training time of the different models and what settings that can be applied to reduce training time.
References


