WORKING PAPER

NEWS ARTICLE ANALYSIS USING NAIVE BAYES CLASSIFIER

Ana Vujović

Available at www.nbs.rs

Economic Research and Statistics Department

NATIONAL BANK OF SERBIA

Belgrade, 12 Kralja Petra Street

Telephone: (+381 11) 3027 100

Belgrade, 17 Nemanjina Street

Telephone: (+381 11) 333 8000

www.nbs.rs

News article analysis using Naive Bayes classifier

Ana Vujović

Abstract: The paper presents the Naive Bayes classifier (NBC), one of the standard models used for solving classification problems, in the context of textual analysis. The model is examined first from a theoretical perspective and then from a practical one. An empirical study was conducted with the aim of carrying out a thematic classification of news articles using the NBC. The results of our research confirm that the NBC has a high predictive power despite the simplified assumptions on which it is based. These findings suggest a potential for further application of the NBC in the thematic classification of texts, which may have significant implications for economic research.

Keywords: Naive Bayes classifier, thematic classification, natural language processing
[JEL Code]: C13, E37.

Non-technical summary

In recent times, natural language processing has become an increasingly relevant topic in economics, especially following the outbreak of the pandemic, during a period of elevated uncertainty. This has highlighted the importance of and need for models that utilise more frequent data. Since news articles or social media comments track current events, analysing this type of data is useful for monitoring economic changes in real time.

With the popularisation of natural language processing and the increasing availability of textual data sources, classical classification models have been adapted to handle textual data. A model that has proven to be exceptionally useful in this domain is the Naive Bayes classifier (NBC).

This paper presents the NBC from both a theoretical and practical perspective, demonstrating how it can be applied in textual analysis. A part of the paper is also dedicated to the processing of textual data, given that such data are more challenging for modelling compared to numerical data. We conducted empirical research aimed at thematic classification of news articles using the NBC.

The analysis was performed on a sample of 2,225 news articles from the BBC website, spanning five different categories. Since the NBC is a supervised machine learning method, it requires training on pre-labelled data, where the model "learns" how to map given inputs to known outputs. After training, the model performs the same task on unlabelled data. We used part of the data to train the model and tested its predictive power on the remaining data. The results indicate that the NBC has a high predictive power, despite the simplified assumptions on which it is based. This finding opens up opportunities for further application of this model in textual analysis, which has proven to be highly useful for various economic analyses.

Contents:

1 Introduction

In recent years, natural language processing has become an increasingly relevant topic in economics, particularly following the pandemic, in the period of elevated uncertainty. This has highlighted the importance of and need for models that utilise more frequent data. Since news articles and social media comments track current events, analysing this type of data is useful for monitoring economic changes in real time.

Investing in techniques for analysing textual data can be a profitable investment for central banks, as these techniques enable the management of various data sources important for assessing monetary and financial stability, which cannot be quantitatively analysed using other methods (Bholat D., Hans, S., Santos, P., & Schonhardt-Bailey, C. (2015)). Therefore, textual analysis is an area of growing importance in the National Bank of Serbia. A news-based inflation sentiment was created by counting expressions related to price changes and inflation, and a statistically significant relationship with inflation was confirmed, indicating that this sentiment can be used as an indicator of the strength of inflationary pressures (Đukić, 2022). A significant relationship was found between the frequency of certain topics in news articles over time and household inflation expectations (Đukić, 2024). The thematic classification of news articles is also the subject of this paper.

The media coverage of certain topics can serve as a good indicator of economic movements, which is why news articles are the most common source of data for natural language processing in economics. Information from news articles can be useful not only for tracking inflation but also for monitoring economic growth, investments, unemployment, or uncertainty, as the sentiment is closely linked to these variables. For short-term forecasting of real GDP growth in the euro area, the sentiment indicator from news articles has outperformed official ECB projections and the composite Purchasing Managers' Index (PMI), especially during the financial crisis (2008/09) and the pandemic lockdown period (2020) (Ashwin, J., Kalamara, E., Saiz, L. (2021)). When it comes to uncertainty, the frequency of terms such as *economy*, *uncertainty*, *deficit*, etc., can be used to construct the Economic Policy Uncertainty Index for United States, whose movement is aligned with economic theory about the relationship between uncertainty and key macroeconomic variables (Baker, S. R., Bloom, N., & Davis, S. J. (2016)). According to more recent research (Kalamara, E., Turrell, A., Redl, C., Kapetanios, G., & Kapadia, S. (2020)), news articles contain more relevant signals of economic sentiment than uncertainty, although literature has primarily focused on the use of textual data for assessing uncertainty.

This paper presents the Naive Bayes classifier (NBC), one of the standard models used for solving classification problems in the analysis of news articles. It aims to explore the NBC model and evaluate its potential application in textual data analysis, which could have significant practical implications for economic research.

The paper is divided into several sections. Following the Introduction, we outline the standard procedure for preparing and processing textual data. We then present the NBC, along with its theoretical foundations and key assumptions, and give an overview of empirical literature on its application. The third section contains the results of empirical research in

which we used the NBC to classify news articles from the BBC website[17] into five categories. The final section summarises our findings.

2 Preparation and processing of textual data

Given that the focus of this paper is on textual data, which requires more complex preparation than numerical data, this section outlines the standard steps for processing and preparing textual data. We will first explain the steps for normalising textual data, followed by the transformation of textual into numerical data using feature extraction and feature selection methods.

2.1 Textual data processing

The standard procedure for processing textual data involves several phases: converting uppercase to lowercase; excluding all characters except words; stemming or lemmatisation; and removing stop words.

To ensure that the programming language does not treat a word differently because it starts with a capital letter (e.g. at the beginning of a sentence), it is necessary to convert all uppercase letters to lowercase. Next, punctuation marks and other characters (such as newline characters \n) that are not words should be excluded, as the program might otherwise treat them as significant information due to their frequent occurrence. These transformations can be performed using **tokenisation** – a method that divides text into tokens. **Tokens** are the basic units of natural language analysis. They can be words, sentences, multiword expressions, or characters. In practice, words are most commonly used as tokens, although other types of tokens may be required in some cases. During tokenisation, other unnecessary characters (i.e. non-word characters in the text) are automatically removed, and this function in many programs also offers the option of converting uppercase to lowercase.

Similarly, programming languages treat different forms of the same word differently, such as: *play*, *playing*, *plays*, *played*... Since these words convey the same or similar information, for analysis, it is necessary to reduce them to their base form – i.e. the root of the word, by removing suffixes. There are two approaches to normalising words: stemming or lemmatisation:

1. Stemming involves removing suffixes from words. This method is relatively simple as it implies setting a small number of rules for normalising words. The drawback is that in some cases, the output may not be a meaningful word, as illustrated in Table 1.

2. Lemmatisation, on the other hand, is based on a dictionary, so its result is always a valid word. However, this method is more demanding as it requires first defining a dictionary and then processing the text according to that dictionary. Lemmatisation can be time-consuming and software-intensive, especially for languages or topics where pre-defined dictionaries for lemmatisation are not available.

We will illustrate the difference between these two methods with a simple example: in the R programming language, we define a list of words (in English – due to dictionary availability)

[17] Data downloaded from Kaggle platform.

that we will first stem and then lemmatise. Table 1 shows the results of stemming and lemmatisation for five words of the same origin: the result of stemming[18] in four cases is not a complete word but the root of the word, while lemmatisation produces a better result – all words are valid, and two are treated as having the same meaning.

Table 1 **Difference between stemming and lemmatisation**

Words	*Stemming*	*Lemmatisation*
creation	*creation*	*creation*
create	*creat*	*create*
created	*creat*	*create*
creative	*creativ*	*creative*
creatively	*creativ*	*creatively*

The final step in textual data processing is the removal of stop words – which frequently appear in text but do not provide relevant information about the content (Claudia, E., Scott B. (2022)). They are eliminated from the text because, otherwise, the algorithm might mark them as relevant due to their frequent occurrence, which could distort the analysis. Examples of stop words include: *and*, *or*, *to*, *how*, *therefore*, *then* etc. While Serbian does not use articles, in English, it is important to remove *the*, *a*, and *an* before conducting any analysis. Most programming languages include lexicons of stop words, making this step straightforward.

2.2 Feature extraction from text

Texts contain a large amount of information that is, in a sense, impossible to abstract using models. Therefore, before modelling, feature extraction from textual data is applied. The goal is to transform textual data into numerical vectors that machine learning models can process. There are various feature extraction methods, with the most commonly used being: part-of-speech tagging (POS tagging), identification, negation, bag-of-words, and term frequency–inverse document frequency (TF-IDF). We have applied here the TF-IDF method, which we will present in this section.

The TF-IDF value indicates the relative importance of a specific token in a document compared to the entire collection of documents. It helps determine the weight of common words that appear frequently across the document collection relative to specific words that appear less frequently, making them more significant for document classification (Tripathy, A., Agrawal, A., Rath, S., K. (2015)). Simply put, just because a word appears many times in a particular document does not mean it is significant if it also appears frequently in all documents (e.g. words such as *and*, *or*, *if*...). Therefore, the main assumption of this approach is that the relative importance of a word is inversely proportional to its frequency across all documents in the collection (Raschka, S. (2014)).

To explain the derivation of this statistical measure, it is necessary to first explain all its components:

- Total number of documents in the collection (N);
- Term frequency (TF) – the frequency of a word in document *d*:

[18] We used the Porter stemmer.

$$TF(t,d) = \frac{Number\ of\ times\ word\ t\ appears\ in\ document\ d}{Number\ of\ words\ in\ document\ d} \quad (1)$$

- Document frequency (DF) – the number of documents in which a specific word appears in the entire collection;
- Inverse document frequency (IDF) – a measure of the importance of a specific word:

$$IDF(t,D) = log\ (\frac{N}{DF}) \quad (2)$$

Finally, term frequency–inverse document frequency (TF-IDF):

$$TF - IDF(t,d,D) = TF(t,d) \times IDF(t,D) \quad (3)$$

2.3 Elimination of rare words

One of the common methods for reducing the number of observations when dealing with words is to eliminate those that do not appear in a large number of documents. This is because such words can be highly specific and, therefore, not very significant for classification, as there is a low probability that they will appear in the prediction data. In practice, words that appear in fewer than five or ten documents in the entire collection are usually excluded. For this, document frequency (DF) is used, and a rule is set to exclude all observations where DF<5 (or DF<10).

3 Naive Bayes classifier

In general, classification problems involve categorising observations based on their attributes into appropriate categories. In the case of classification problems in natural language processing, texts are also grouped into different categories based on their attributes, where the attributes are the words themselves.

The NBC is a probabilistic model because it is based on probabilities, and it operates based on the **Bayes' rule**. The NBC determines the probability that an observation belongs to a specific class based on the features of that observation. As inputs, the model uses **prior probabilities** that an observation belongs to a specific class, and **conditional probabilities** of the features within the classes. When the observations are texts, based on the probabilities that a text belongs to a category and the conditional probabilities of the text's features within a category, the model determines the **posterior probability** of the category to which the text belongs. The model assigns the text to the category where this probability is highest.

3.1 Bayes' rule

To gain insight into this model, it is first necessary to present the Bayes' theorem, formulated by the British statistician Thomas Bayes (1701–1761):

$$P(A\ |\mathrm{B}) = \frac{P(B|A)P(A)}{P(B)} \quad (4)$$

where:

- $P(A \mid B)$ – posterior probability, representing the probability of hypothesis A under the condition that event B has occurred, which serves as evidence;
- $P(B \mid A)$ – conditional probability of event B under the condition that hypothesis A is true;
- P(A) – prior probability of hypothesis A without any knowledge of event B;
- P(B) – probability of event B, i.e. the state indicated by the data (evidence).

3.2 Application of Bayes' rule in classification problems

After presenting Bayes' formula in its general form, we will present it in the context of a classification problem:

$$P(C_i \mid x_1, x_2 \dots x_q) = \frac{P(x_1, x_2 \dots x_q \mid C_i) P(C_i)}{P(x_1, x_2 \dots x_q \mid C_1) P(C_1) + P(x_1, x_2 \dots x_q \mid C_2) P(C_2) + \dots + P(x_1, x_2 \dots x_q \mid C_p) P(C_p)} \quad (5)$$

In this case, the posterior probability is the probability that an observation belongs to class *i* given that it has attributes *x_1,x_2…x_q*. The attributes of the observation are known, and based on them, we need to determine which class the observation belongs to. The goal is to find the posterior probabilities for all classes (from 1 to *p*) and assign the observation to the class with the highest probability. This decision rule can be represented by the formula:

$$assigned\ class \leftarrow \underset{i=1,2\dots,p}{\operatorname{argmax}} P(C_i \mid x_1, x_2 \dots x_q) \quad (6)$$

The conditional probability represents the probability that an observation has features *x_1,x_2…x_n* and belongs to class *i*. The prior probability is the probability that an observation belongs to a specific category. The denominator of formula 5 contains the formula for total probability, which calculates the probability of the observation's attributes, i.e. the probability that the observation has certain attributes regardless of the class it belongs to. In cases where the model encounters an observation in the test dataset that was not present in the training dataset, this expression equals 0. To avoid division by zero, the **Laplacian smoothing factor** is added to the numerator and the same factor multiplied by the total number of attributes is added to the denominator of the formula for calculating the posterior probability.

3.3 Model assumptions

To simplify the calculation of the posterior probability, the **naive assumption** is introduced, which gives the model its "naive" attribute. The simplification assumes that the **attributes are mutually independent**. In reality, this assumption does not hold, but it does not significantly affect the performance of the model, which has proven to be effective in numerous classification tasks. This can be explained by the fact that optimality in the context of **classification error** (0/1 loss) does not depend on how well the data fit the assumed probability distribution but rather on whether the realised and assumed distributions correspond to the most probable class (Rish, I. (2001)). A more significant indicator of the NBC's accuracy is the amount of information about the class that is lost due to the attribute

independence assumption (Rish, 2001). This assumption facilitates the calculation of conditional probabilities, which in this case are calculated as follows:

$$P(x_1, x_2 \dots x_q \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i) \tag{7}$$

thus, the formula for calculating the posterior probability becomes:

$$P(C_1 \mid x_1, x_2 \dots x_q) = \frac{P(x_1 \mid C_1)\mathrm{P}(x_2 \mid C_1)\dots P(x_q \mid C_1)P(C_1)}{P(x_1 \mid C_1)P(x_2 \mid C_1)\dots P(x_q \mid C_1)P(C_1)+\dots+ P(x_1 \mid C_p)P(x_2 \mid C_p)\dots P(x_q \mid C_p)P(C_p)} \tag{8}$$

For calculation of conditional probabilities, an additional assumption is introduced – they can be computed as follows:

$$P(x_k \mid C_i) = \frac{n}{N} \tag{9}$$

This means that the probability that an observation with attribute x_k belongs to class C_i is expressed as the ratio of the number of observations with attribute x_k that belong to class C_i (in the training data) – n, to the total number of observations in class C_i – N.

3.4 NBC in textual analysis

When it comes to applying the NBC in textual analysis, the attributes x_k are words, and the classes C_i are text categories. The conditional probability $P(x_1 \mid C_1)$ j is the probability that a specific word appears in a given category. This probability is calculated as the ratio of the number of words x_i that appear in category C_i to the total number of words in the observed category. The posterior probability $P(C_1 \mid x_1, x_2 \dots x_q)$ is the probability that a text containing specific words belongs to the category of interest. We will illustrate this with the following sentence: “The central bank has tightened monetary policy”. Here, the posterior probability is the probability that this sentence belongs to the category of economic texts, and it is calculated based on the conditional probabilities that each word from the sentence belongs to this category.

3.5 Evaluation metrics for classification models

Before delving into examples where the NBC is applied, it is essential to introduce the evaluation metrics for classification models to better understand the model performance. These metrics are used for all classification models, regardless of the type of data.

To derive classification evaluation metrics, it is first necessary to create a **confusion matrix**, which shows the relationship between correctly and incorrectly classified observations. This matrix can also be applied when the classification problem involves more than two classes, but for simplicity, the table below shows a confusion matrix for a binary classification problem where the classes are labelled as positive and negative.

Table 2 **Confusion matrix**

		Actual category	
		Positive	Negative
Predicted category	Positive	True positive (TP)	False positive (FP)
	Negative	False negative (FN)	True negative (TN)

Four categories of observations can be derived from this matrix:

a) **True positive** (TP): observations that are truly positive and are correctly classified as positive by the model.

b) **False negative** (FN): observations that are positive but are incorrectly classified as negative by the model.

c) **True negative** (TN): observations that are negative and are correctly classified as negative by the model.

d) **False positive** (FP): observations that are negative but are incorrectly classified as positive by the model.

Based on these four categories of observations, the following classification evaluation metrics are derived:

a) Accuracy: the ratio of correctly classified observations to the total number of observations in the test dataset. This metric is most commonly cited when discussing the success of a classifier. It is calculated using the formula:

$$accuracy = \frac{TP+TN}{TP+TF+FP+F} \tag{10}$$

b) Error rate (estimated misclassification rate): the opposite of accuracy. It expresses the ratio of incorrectly classified observations to the total number of classified observations:

$$err = \frac{FP+FN}{TP+TF+FP+FN} \tag{11}$$

These two metrics are general indicators and are not sufficiently informative in cases where one class is more important than the other. For example, in medicine, it can be very dangerous if a patient has a disease, the test is positive, but the model classifies it as negative (false negative). Therefore, the following metrics are also calculated:

c) **Precision** – positive predicted value (ppp): measures the precision of the model. It refers to the proportion of correctly classified positive observations relative to all observations classified as positive:

$$Precision = \frac{TP}{TP+F} \tag{12}$$

d) **Negative predicted value** (npp): calculated similarly to precision but refers to negative observations:

$$npp = \frac{TN}{TN+} \tag{13}$$

e) **Recall** (sensitivity): indicates the model's ability to classify observations of the class of interest. It measures the completeness of the model, i.e. the proportion of correctly classified positive observations relative to all positive observations:

$$Recall = \frac{TP}{TP+FN} \tag{14}$$

f) **Specificity**: similar to recall but refers to negative observations:

$$Specificity = \frac{TN}{TN+FP} \tag{15}$$

g) **F-value**: a metric that combines precision and recall, calculated as the harmonic mean of the two:

$$F - value = \frac{2 \times precision \times recall}{precision+recall} \tag{16}$$

3.6 Overview of empirical literature on the Naive Bayes classifier

The NBC has proven to be most accurate in binary classification problems. This is because the evaluation of a classification model depends on the sign of the estimated function (McCallum, A., Nigam, K. (1998)). This also explains the success of this model despite the violation of the attribute independence assumption. Therefore, it is not surprising that the NBC is most commonly applied in sentiment analysis, where the goal is typically to classify comments as positive or negative, although in some cases, a third category for neutral comments is also included. For companies, strongly positive and negative opinions about products/services are certainly more important, as more actionable information can be extracted from such comments. Hence, companies often use this model to classify reviews of their products/services and to monitor their reputation. For example, in the classification of movie reviews as positive or negative, the NBC achieved a precision of 80% for negative reviews and 87% for positive reviews (Tripathy, A., Agrawal, A., Rath, S., K. (2015)). This study is not an isolated case where the NBC is more accurate in classifying positive reviews compared to negative ones. A similar result was obtained in the classification of restaurant reviews, leading to the proposal of a modified version of the NBC model that reduced the gap in classification accuracy between positive and negative reviews by 3.6% (Kang, H., Yoo, S. J., & Han, D. (2012)). This limitation of the model should not be an obstacle to its use in natural language processing, given the proposed modification. On the other hand, in both cases, the support vector machine (SVM) model – a linear classification model that works on the principle of determining the best linear boundary (plane) to separate categories – proved to be more accurate, suggesting that a combined approach of these two classification models might be an optimal solution.

Another common application of the NBC in text classification is email sorting and spam detection. In such cases, the model is trained on pre-sorted emails that differ based on the words that most frequently appear in standard emails versus those that indicate junk emails.

The principle is similar to the examples mentioned above, except that instead of looking for positive or negative words, the focus is on words characteristic of different types of emails, so the analysis should focus on other text features. Email filtering based on the NBC showed good performance, achieving a precision of 92% for junk email classification and 95% for standard email classification (Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998)).

As the evaluation of written responses is time-consuming and costly for universities and other institutions, automated essay grading has gained popularity in recent times. This method also falls under classification tasks, where grades represent different classes. Therefore, it is possible to train a model to classify ungraded essays based on the characteristics of already graded essays and assign grades accordingly. This idea is not entirely new, but with the development of artificial intelligence, it has gained significance in recent years – as early as 2002, research was conducted showing that the NBC could classify ungraded student essays with an accuracy of 80% (Rudner, L. M., Liang, T. (2002)).

The field where the NBC has proven useful and has been applied for a longer period is medicine. This model is most commonly used to predict the risk of developing certain diseases based on patient risk factors. When comparing the performance of the NBC and SVM models for predicting the risk of liver disease, the SVM proved to be more accurate, while the NBC was faster (Vijayarani, S., Dhayanand, S. (2015)). On the other hand, a modified version of the NBC – the weighted NBC – proved to be exceptionally good in breast cancer detection, achieving a sensitivity of 99.11%, accuracy of 98.54%, and specificity of 98.25% (Karabatak, M. (2015)). Similarly, the application of this classifier for predicting the risk of recurrence or progression of brain cancer led to significant results that can be further used for practical purposes (Kazmierska, J., & Malicki, J. (2008)).

4 Author's analysis: text classification by topic using the NBC model

The focus of our analysis is the classification of news articles from the BBC website into five categories based on their topics using the NBC. The news articles belong to the fields of business, entertainment, politics, sports, and technology. These data were taken from the Kaggle platform. The analysis was conducted in the R programming language.

The entire dataset contains 2,225 news articles. Given that this is a relatively small dataset for data science purposes, we consider the NBC suitable for classification, as it is generally used when the training dataset is of a smaller size (Wankhade, M., Rao, A., Kulkarni, C. (2022)). The research was conducted in the following stages:

1. Data analysis, loading, and preparation: this involves all the steps of textual data processing outlined in the second section;
2. Text feature extraction using the TF-IDF method;
3. Data splitting: dividing data into training and testing sets;
4. Application of the NBC on the training dataset;
5. Model testing: classifying texts from the testing dataset;

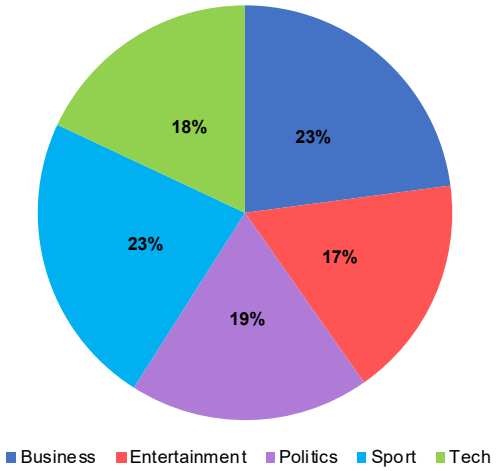
6. Examination of model characteristics using the evaluation metrics for classification models presented in the previous section;

7. Finally, determining the economic and marketing significance of the obtained model.

4.1 Data analysis

As some topics are always more prevailing than others in journalism, it is expected that an equal number of articles from all five fields were not published during the observed period. In our corpus of texts, there are 510 articles from business, 386 from entertainment, 417 from politics, 511 from sports, and 401 articles covering technology. Chart 1 shows that this dataset is approximately balanced, with business and sports being the most represented topics. We do not expect these small differences in the relative representation of topics to lead to bias during model training and testing.

Chart 1 **Share of different topics in the collection of texts**



4.2 Data preparation

Since the NBC is a supervised machine learning model, it is necessary to label the categories of the texts after loading the dataset. Once the dataset is loaded with clearly labelled categories, we can proceed to process the texts.

As already explained, textual data in its "raw" form are not ready for modelling. Data preparation is carried out in three phases: tokenisation, removal of stop words, and stemming. Although it is clear from Table 1 that lemmatisation yields better results, due to the complexity of this method, stemming had been applied in this paper.

The first step in data preparation is tokenisation. As stated, tokens can be individual words, n-grams, or sentences. For the purposes of thematic classification, the tokens of interest are words. This process significantly increases the number of observations – from 2,225 (articles) to 861,355 (words). However, in the next step, this number is reduced because words that appear frequently but are not informative – stop words – are not needed for text classification.

By removing stop words, the number of data was reduced from 861,355 to 268,178, highlighting the importance of eliminating this type of words. The final step in data preparation is stemming, which results in words in their base form, i.e. with suffixes removed.

To illustrate the importance of this type of textual data processing, Table 3 shows the first 30 observations of a randomly selected text in four preparation phases. The first phase involves the simplest extraction of words from texts, where a word is defined as anything between two white spaces. This type of text separation into words was applied only to this small sample, as tokenisation already provides text separated into words, which are also reduced to lowercase, and unnecessary characters are automatically excluded. In this case, we have only shown this phase for comparison purposes.

Table 3 **Text preparation phases**

Words	*Tokens*	*No stop words*	*Stemming*
Lennon	*lennon*	*lennon*	*lennon*
brands	*brands*	*brands*	*brand*
Rangers	*rangers*	*rangers*	*ranger*
favourites\n\nCeltic's	*favourites*	*favourites*	*favourit*
Neil	*celtic's*	*celtic's*	*celtic'*
Lennon	*neil*	*neil*	*neil*
admits	*lennon*	*lennon*	*lennon*
Rangers	*admits*	*admits*	*admit*
could	*rangers*	*rangers*	*ranger*
be	*could*	*considered*	*consid*
considered	*be*	*slight*	*slight*
\"slight	*considered*	*favourites*	*favourit*
favourites\"	*slight*	*firm*	*firm*
for	*favourites*	*cis*	*ci*
the	*for*	*cup*	*cup*
Old	*the*	*clash*	*clash*
Firm	*old*	*insists*	*insist*
CIS	*firm*	*win*	*win*
Cup	*cis*	*lennon*	*lennon*
clash,	*cup*	*concedes*	*conced*
but	*clash*	*rangers*	*ranger*
insists	*but*	*form*	*form*
his	*insists*	*moment*	*moment*
side	*his*	*failed*	*fail*
can	*side*	*beat*	*beat*
still	*can*	*celtic*	*celtic*
win.\n\nLennon	*still*	*meetings*	*meet*
concedes	*win*	*rangers*	*ranger*
Rangers	*lennon*	*run*	*run*
are	*concedes*	*recent*	*recent*

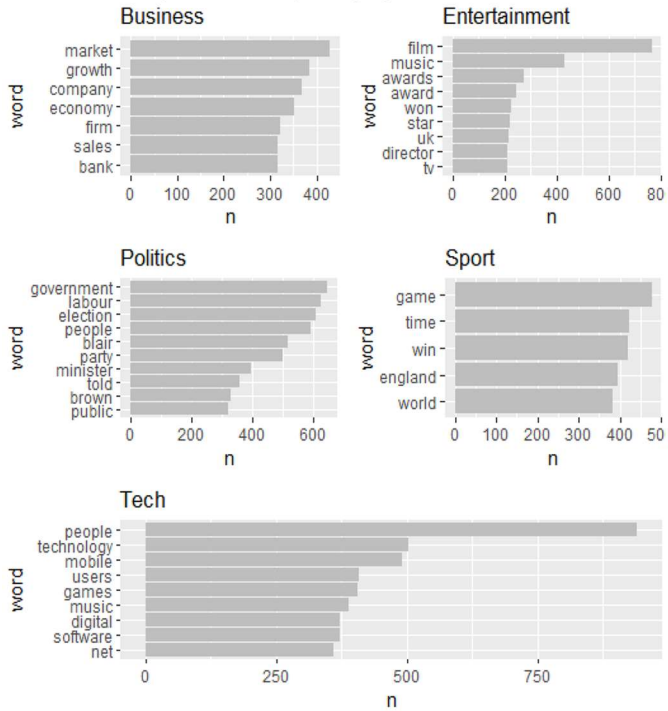
When comparing the first and second columns, it is clear why the first phase was not applied to all data. Tokenisation results in whole words reduced to lowercase, without periods, quotation marks, and newline characters (\n). Therefore, tokenisation is a logical first step in text processing. In the next step, words such as *could*, *be*, *but, his*, etc., which are not relevant

for text classification, are removed. Finally, after stemming, we have words that will be used in further analysis. It is evident that the result of stemming is not ideal – for example, the word *considered* is reduced to *consid*, which has no meaning. However, in most cases, this step is significant – we need the algorithm to treat the words *admits* and *admit* in the same way because they have the same meaning. Moreover, the imprecision of stemming can be neglected given the number of words being analysed.

4.3 Text feature extraction and selection

To classify a text by topic, attributes of the texts that are used as inputs to the model are needed. Intuitively, the most important attributes are the words that appear most frequently in articles within a particular topic. Before extracting features from texts, it is necessary to first see which words these are. Chart 2 shows the words that appear most frequently in news articles across all five categories. The results are in line with expectations: in the business category, the most frequent words are ***market***, ***growth***, and ***company***, while in the entertainment category, they are ***music***, ***film***, and ***awards***. Surprisingly, even though suffixes have been removed through stemming, *awards* and *award* are treated as different words even though they have the same meaning. This problem can be ignored if it occurs in isolated cases, but if it occurs frequently, it can lead to data dimensionality issues, which may adversely affect model evaluation. Another potential problem concerns the categories of politics and technology – in both categories, the word *people* is among the most frequent words. This could lead to the model incorrectly assigning a text from the politics category to the technology category.

Chart 2 **Most frequent words by category**



As already emphasised in the second section, the frequencies of words themselves are not as significant as the relative occurrence of certain words in a text compared to the entire collection of texts. Therefore, we applied the TF-IDF method for feature extraction from textual data. In this way, each word is assigned a TF-IDF value in the range of 0 to 1.

Subsequently, to reduce the number of observations, we excluded all words that appear in fewer than ten texts in the entire collection of texts. Thus, the total number of observations was reduced from 268,178 to 56,529, which weakens the problem of data dimensionality.

4.4 Model training and testing

Before training the model, it is necessary to first split the data into training and testing sets, which is a standard procedure for all supervised machine learning methods. Since the data are processed at the word level rather than at the text level, it is necessary to separate them by texts to avoid overlap, i.e. to prevent situations where the model is trained on a certain number of words from one text and predicts on other words from the same text. After separating the data by texts, we can split them according to the common practice of using 70% of the data for model training and the remaining 30% for testing.

The next step is training the NBC on the training dataset. The dependent variable is the category, and the independent variables are all other variables (words and their TF-IDF values). We set the Laplacian smoothing factor to 1 to avoid zero probabilities.

On the portion of the data set aside for prediction (30% of observations), we will test the predictive power of this model. The testing results are illustrated in Table 4, which shows a sample of 11 words, the category assigned to each of these words, the estimated probabilities that the word belongs to each of the five categories, and the category to which the text from which the word was extracted belongs.

Table 4 **Word classification based on NBC**

Word	Assigned category	Probabilities					Relevant category
		Business	*Entertainment*	*Politics*	*Sport*	*Tech*	
parti	*Politics*	0.00	0.00	0.99	0.01	0.00	*Politics*
time	*Sport*	0.00	0.00	0.00	1.00	0.00	*Sport*
game	*Tech*	0.00	0.00	0.00	0.00	1.00	*Tech*
sale	*Entertainment*	0.00	1.00	0.00	0.00	0.00	*Businesss*
film	*Entertainment*	0.00	1.00	0.00	0.00	0.00	*Entertainment*
govern	*Entertainment*	0.00	0.96	0.00	0.00	0.04	*Politics*
product	*Businesss*	1.00	0.00	0.00	0.00	0.00	*Businesss*
campaign	*Entertainment*	0.24	0.76	0.00	0.00	0.00	*Entertainment*
win	*Sport*	0.00	0.00	0.39	0.61	0.00	*Politics*
futur	*Tech*	0.00	0.00	0.00	0.01	0.99	*Tech*
start	*Tech*	0.00	0.00	0.00	0.01	0.99	*Tech*

4.5 Examination of model characteristics

To evaluate the model characteristics, we have constructed a confusion matrix, based on which we calculate the classification evaluation metrics. The confusion matrix and the table of classification evaluation metrics for the obtained model are presented below.

The confusion matrix shows that the NBC has classified well the words into categories. The accuracy of this model is **95.77%**, while the estimated misclassification rate is **4.23%**.

Table 5 **Confusion matrix for obtained model**

		Relevant category				
		Business	Entertainment	Politics	Sport	Tech
Assigned category	Business	**4174**	131	0	0	0
	Entertainment	94	**1774**	68	1	0
	Politics	0	91	**3423**	115	0
	Sport	0	0	64	**3158**	85
	Tech	0	0	2	52	**3407**

Table 6 **Classification evaluation metrics**

	Business	*Entertainment*	*Politics*	*Sport*	*Tech*
Sensitivity	0.978	0.889	0.962	0.95	0.976
Specificity	0.989	0.989	0.984	0.989	0.996
Pos. predicted value	0.97	0.916	0.943	0.955	0.984
Neg. predicted value	0.992	0.985	0.99	0.987	0.993
F-measure	0.974	0.902	0.952	0.952	0.980

Table 5 shows that the words from the technology category were classified with the highest precision. This is because this category is the least related to the other four categories. On the other hand, the lowest classification precision was achieved in the entertainment category, which is somewhat expected, as this category had the smallest number of texts, leading to a slight bias towards other categories. The largest number of misclassified words were from the entertainment category, which were incorrectly assigned to the business category (out of 1,996 words belonging to the entertainment category, 131 were assigned to the business category). Similarly, out of 4,268 words from the business category, 94 were classified as belonging to the entertainment category. This indicates an overlap of key words from these two categories, which also affected the precision of word classification in the entertainment category.

The highest sensitivity, or the model's ability to assign observations to the relevant categories, was achieved for the business category (97.8%). In contrast, the lowest value for

this metric was observed in the entertainment category (88.9%). Although the highest number of errors occurred in this category, based on Tables 6 and 7, we can conclude that the NBC demonstrated good performance in word classification overall.

Table 7 **Text-level confusion matrix**

		Relevant category				
		Business	Entertainment	Politics	Sport	Tech
Assigned category	Business	**160**	8	0	0	0
	Entertainment	0	**97**	0	0	0
	Politics	0	3	**118**	2	0
	Sport	0	0	1	**162**	0
	Tech	**0**	0	0	0	**121**

Table 8 **Text-level classification evaluation matrics**

	Business	*Entertainment*	*Politics*	*Sport*	*Tech*
Sensitivity	1	0.9	0.992	0.99	1
Specificity	0.984	1	0.991	0.998	1
Pos. predicted value	0.952	1	0.959	0.994	1
Neg. predicted value	1	0.981	0.998	0.996	1
F-measure	0.975	0.947	0.975	0.991	1.000

However, the goal of this research is text-level and not word-level classification. Therefore, it is necessary to recombine words into texts and assign the category to which the majority of words in that text belong. Although we did not create a separate model for text-level classification, the tables below show the confusion matrix and classification evaluation metrics for texts classified in this manner.

The accuracy of text-level classification is higher compared to word-level classification, standing at 97.92%. The classification error was reduced by 2.13 percentage points, or 2.1%. The improvement in classification was achieved because words were first classified into categories, and then texts were classified by summing the categories of the words within them. In other words, a text was assigned to the category to which the majority of its words belong. Text classification is an averaged word-level classification, which is why it is expected to be more accurate.

5 Conclusion

In this paper, we presented the thematic classification of 2,225 news articles from the BBC website into five categories using the NBC. The model was trained on a pre-labelled portion of data and tested on the remaining unlabelled portion, which is standard practice for

supervised machine learning methods. This approach provides the simplest way to gain insight into the characteristics, advantages, and limitations of the model.

The most significant advantages of the NBC are its simplicity, robustness, and low memory usage. The training phase is relatively slower, while the prediction phase is faster. Another limitation of the NBC is its weaker adaptability to new data – the model is trained based on the available training dataset, so adding new data requires retraining the model. In addition, the naive attribute independence assumption does not significantly affect the accuracy of predictions.

Although the dataset used for the analysis is relatively small, the results of our research are useful as they confirm the predictive power of the NBC despite the simplicity of its underlying assumptions – the accuracy of the obtained model is 97.9%, with a classification error of 2.1%. Besides accuracy, an additional advantage of this model is that, due to its simple assumptions, it can be easily adapted to different types of data and to various domains.

Since news articles are the most important source of data for textual analysis in economics, in this paper, we conducted an analysis on this data source and determined that the NBC can be very effectively adapted to it. This insight opens up opportunities for further application of the NBC in the analysis of news articles in economic research, which can be highly practical given the availability and timeliness of these data sources. In this context, economic news articles and datasets in the Serbian language will be the subject of our future analyses.

Literature

Ashwin, J., Kalamara, E., Saiz, L. 2021. Working Paper Series. *Nowcasting euro area GDP with news sentiment: a tale of two crises*. ECB Working Paper No. 2021/2616.

Baker, S. R., Bloom, N., & Davis, S. J. 2016. *Measuring economic policy uncertainty*. The Quarterly Journal of Economics, 131(4), 1593–1636.

Bholat D., Hans, S., Santos, P., & Schonhardt–Bailey, C. 2015. *Text mining for central banks*. Handbooks, Centre for Central Banking Studies, Bank of England, number 33.

Claudia, E., Scott B. 2022. *Text Analysis with R*. https://cengel.github.io/R-text-analysis/.

Đukić, M. 2022. *Assessment of inflationary pressures using newspaper text analysis*. Working Papers Bulletin of the National Bank of Serbia, III.

Đukić, M. 2024. *Topic classification of economic newspaper articles in a highly inflectional language – the case of Serbia*. Working Papers Bulletin of the National Bank of Serbia, VI.

Kalamara, E., Turrell, A., Redl, C., Kapetanios, G., & Kapadia, S. 2020. *Making text count: economic forecasting using newspaper text*. Staff Working Paper No. 865, Bank of England.

Kang, H., Yoo, S. J., & Han, D. 2012. *Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews*. Expert Systems with Applications, 39(5), 6000–6010.

Karabatak, M. 2015. *A new classifier for breast cancer detection based on Naïve Bayesian*. Measurement, 72, 32–36.

Kazmierska, J., & Malicki, J. 2008. *Application of the Naïve Bayesian classifier to optimize treatment decisions*. Radiotherapy and Oncology, 86(2), 211–216.

McCallum, A., & Nigam, K. 1998. *A Comparison of Event Models for Naive Bayes Text Classification*. In AAAI-98 workshop on learning for text categorization (Vol. 752, No. 1, pp. 41–48).

Raschka, S. 2014. *Naive Bayes and Text Classification I – Introduction and Theory*. arXiv preprint arXiv:1410.5329. https://arxiv.org/abs/1410.5329.

Rish, I. 2001. *An empirical study of the naive Bayes classifier*. In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol. 3, No. 22, pp. 41–46).

Rudner, L. M., & Liang, T. 2002. *Automated essay scoring using Bayes' theorem*. The Journal of Technology, Learning and Assessment, 1(2).

Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. 1998. *A Bayesian approach to filtering junk e-mail*. In Learning for Text Categorization: Papers from the 1998 workshop (Vol. 62, pp. 98–105).

Source: BBC Full Text Document Classification. Downloaded on 27 April 2024: https://www.kaggle.com/datasets/shivamkushwaha/bbc-full-text-document-classification.

Tripathy, A., Agrawal, A., Rath, S., K. 2015. *Classification of sentimental reviews using machine learning techniques*. Procedia Computer Science 57:821–829.

Vijayarani, S., & Dhayanand, S. 2015. *Liver disease prediction using SVM and Naïve Bayes Algorithms*. International Journal of Science, Engineering and Technology Research (IJSETR), 4(4), 816–820.

Wankhade, M., Rao, A., Kulkarni, C. 2022. *A survey on sentiment analysis methods, applications, and challenges*. Artificial Intelligence Review 55:5731–5780.