MCIT
Modeling, control &
information technologies

Analysis of machine learning methods using spam filtering

https://doi.org/10.31713/MCIT.2021.06

Nataliya Boyko
department of artificial intelligence
Lviv Polytechnic National University
Lviv, Ukraine
Nataliya.i.boyko@lpnu.ua

Oleksandra Dypko
department of artificial intelligence
Lviv Polytechnic National University
Lviv, Ukraine
oleksandra.dypko.knm.2018@lpnu.ua

***Abstract* — The paper considers methods of the naive Bayesian classifier. Experiments that show independence between traits are described. Describes the naive Bayesian classifier used to filter spam in messages. The aim of the study is to determine the best method to solve the problem of spam in messages. The paper considers three different variations of the naive Bayesian classifier. The results of experiments and research are given.**

***Keywords* — classification; naive Bayes; machine learning; spam filtering.**

I. Introduction

Spam e-mail is a problem that determines the significant economic impact on society. Spam is a waste of time, storage space and communication bandwidth. The problem of e-mail with the accumulation of spam is gaining relevance. According to the latest statistics, 54.61% of all emails are spam. This is about 15.4 billion emails per day. That is why spam filtering using the naive Bayesian classifier is a topical issue for research [1, 5].

The choice of naive Bayesian classifiers to solve the problem of spam filtering is due to the fact that today they are the most widely used filters for spam classification. They are used in free webmail servers and open source systems [3, 8].

The naive Bayesian classifier method has some advantages over other classifiers that can be used for spam filtering. The paper describes: prediction of the test data set class and the naive Bayes classifier, which works in multi-class forecasting [6, 10, 12].

II. Methods Overview

A. Statement of the classification problem

Spam filtering is a type of text classification task, so the following model of the classification task should be used. Statement of the problem of text classification [7, 12]:

Let $\epsilon = \{d1, d2, ..., dm\}$ be a set of text documents with a set of features $W = \{w1, w2, ..., wn\}$ (ie each text document $di = (w1, w2, .., wn)$ and a given function of the distance (metric) between objects $\rho\ (di, dj)$, where $di, dj \in D$ objects.

The classification function is a function $C = \{C1, C2, ..., Ck\}$, which unambiguously corresponds f to each object $d \in D$ cluster number $y \in Y = \{1, ..., k\}$, $k \leqslant m$. It is necessary to find such a function f * that Q (f*,C,ρ) = min($Q(f,\ C,\ \rho)$), where $Q\ (f, C, \rho)$ is the chosen criterion of classification quality.

B. Naive classifier of Bayes

The Bayesian classifier is a broad class of classification algorithms based on the principle of maximum posterior probability. For a classified object, the probability functions of each of the classes are calculated, as well as the back probability of the class. The object belongs to the class for which the rear probability is maximum [2, 4].

The Bayesian classifier is based on the Bayesian theorem – one of the basic theorems of probability theory, which allows to determine the probability of any event A provided that there is another statistically interrelated event B (Formula 1) [1, 6]:

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

Using formula 1, we can calculate the conditional probability $P\ (A \mid B)$ that an event A took place, if event B was observed as a result of the experiment, with the known probabilities of occurrence of events – $P\ (A)$ and $P\ (B)$, and the conditional probability of occurrence events A with existing B - $P\ (B \mid P)$.

Suppose that for each class $Cj \in C$ we know the a priori probability $P\ (Cj)$ of the appearance of an object of class Cj and the distribution density $P\ (d \mid Cj)$ of each of the classes, also called the likelihood functions of the classes.

According to Bayes' theorem, the probability that document d belongs to the class Cj is calculated as follows (Formula 2) [1, 8, 11]:

$$P(Cj|d) = \frac{P(d|Cj)P(Cj)}{P(d)}, \tag{2}$$

where $P\ (Cj \mid d)$ – the probability that the document d belongs to the class Cj, this is what we need to calculate; $P\ (d \mid Cj)$ – probability to find document d among documents of class Cj (density of class Cj distribution); $P\ (Cj)$ – a priori (unconditional) probability of the class of occurrence of the document of class Cj; $P\ (d)$ – unconditional probability of appearance of the document d in the body of documents.

The value $P\ (Cj \mid d) = P\ (d \mid Cj)\ P\ (Cj)$ is interpreted as an a posteriori probability that the object d belongs to the class $Cj \in C$.

The most probable class $C\ *$ to which the document d belongs is the class for which the conditional probability of belonging of the document d class Cj is maximum (Formula 3) [1, 9]:

$$C * = arg\max_j P\ (Cj \mid d). \tag{3}$$

It is necessary to calculate the probability for all classes and select the class for which the probability has the maximum value. By Bayes' theorem (Formula 4):

$$C * = arg\max_j \frac{P\ (d \mid Cj)\ P(Cj)}{P\ (d)}. \tag{4}$$

According to the solved classification problem, each document $d \in D = \{d1,\ d2,\ ...,\ dm\}$ is given by signs with $W = \{w1,\ w2,\ ...,\ wn\}$, ie each text document $di = (w1,\ w2\ ,\ ...,\ wn)$. For this model, the features of the document should be considered some characteristics associated with the words contained in it.

The next step is to make an assumption, which explains why this algorithm is called naive. It reads as follows: the denominator can be omitted, because for the same document вір the probability $P\ (d)$ will be the same, which means that it can be ignored (Formula 5):

$$C * = arg\max_j P\ (w1, w2, \ldots, wn \mid Cj)\ P\ (Cj). \tag{5}$$

Also in the model of the naive Bayesian classifier it is assumed that all the features $w1$, $w2$, ..., wn of the document d are independent of each other. It is clarified that the position of the term in the sentence is not important.

Thus, the conditional probability $P\ (w1,\ w2,\ ...,\ wn \mid Cj)$ for the features $w1$, $w2$, ..., wn can be represented as follows (Formula 6):

$$P\ (w1, w2, \ldots, wn \mid Cj)\ =\ \prod_j P\ (wi \mid Cj). \tag{6}$$

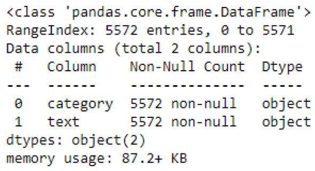
Thus, to find the most probable class for the document $d = \{w1,\ w2,\ ...,\ wn\}$ using the available Bayesian classifier, it is necessary to calculate the conditional probabilities of the document d for each of the presented classes and choose the class with the maximum probability (principle of maximum a posteriori probability) (Formula 7):

$$C * = arg\max_j [P\ (Cj)\ \prod_j P\ (wi \mid Cj)], j = 1, 2. \tag{7}$$

Therefore, it all comes down to calculating the probabilities P (C) and P (w | C). Calculating these parameters is called classifier training.

III. Review and Analysis of Data

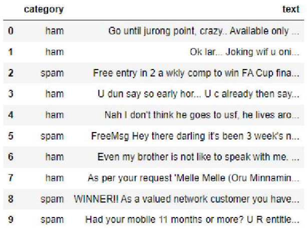
A dataset consisting of a set of SMS messages should be used for this study. The dataset contains information about SMS-messages, each of which belongs to one of the categories – "spam" or "ham". The total number of records is 5572. After performing a preliminary analysis of the dataset, you need to remove the attributes that will not be necessary for remote data processing. In Fig. 1 below shows the structure of the dataset.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   category  5572 non-null   object
 1   text      5572 non-null   object
dtypes: object(2)
memory usage: 87.2+ KB
```

Figure 1. Figure 1. Dataset structure

In fig. 2 shows the first 10 records of the dataset, namely the three features. Each entry in the columns is text, in the first column of unique values only two: "spam", "ham"; in the second more, because they are text messages from different people, so the same messages are very rare.



	category	text
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...
5	spam	FreeMsg Hey there darling it's been 3 week's n...
6	ham	Even my brother is not like to speak with me. ...
7	ham	As per your request 'Melle Melle (Oru Minnamin...
8	spam	WINNER!! As a valued network customer you have...
9	spam	Had your mobile 11 months or more? U R entitle...

Figure 2. The first 10 records of the dataset

A. Experiments with the Bayesian classifier

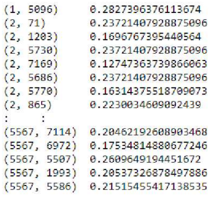
During the experiments, different variations of the Bayesian classifier were compared, namely the Bernoulli classifier, the Gaussian classifier and the polynomial classifier. You need to determine the correctness and accuracy of each of them to understand which one is best for filtering spam in messages.

You first need to analyze and review the dataset, and then you need to remove any characters that are not letters except the space, and reduce the case to lowercase for better classifier performance.

The next step is to remove words that are very common but do not make sense in the sentences. In English, these are articles, pronouns, prepositions that stand next to words.

The naive Bayesian classifier uses the TF-IDF statistic, which exists to evaluate important words in documents.

TF (word frequency) characterizes the ratio of the number of input specific words to the total set of words in the documents. IDF (inverse frequency of the document) characterizes the inversion of the frequency with which a particular word is used in the text.



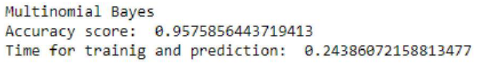
```
(1, 5096)       0.2827396376113674
(2, 71)         0.23721407928875096
(2, 1203)       0.1696767395440564
(2, 5730)       0.23721407928875096
(2, 7169)       0.12747363739866063
(2, 5686)       0.23721407928875096
(2, 5770)       0.16314375518709073
(2, 865)        0.2230034609092439
:     :
(5567, 7114)    0.20462192608903468
(5567, 6972)    0.17534814880677246
(5567, 5507)    0.2609649194451672
(5567, 1993)    0.20537326878497886
(5567, 5586)    0.21515455417138535
```

Figure 3. The value of the statistical indicator TD-IDF for each word

In fig. 3 presents two columns, the first of which is a pair of numbers: the number of the sample element and the unique token of this element; the numbers in the second column are the calculated TF-IDF value, which means how important the word is in the text.

The next step is to learn the classifier model and the prediction itself. To do this, the dataset is divided into two samples – test and training in the ratio of 25% to 75%, respectively. As already mentioned, we will teach three models of the Bayesian classifier.

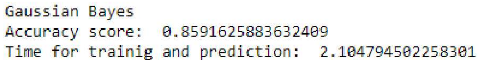
The results of the prediction of different variations of the classifier (Fig. 4):



```
Multinomial Bayes
Accuracy score:  0.9575856443719413
Time for trainig and prediction:  0.24386072158813477
```

Figure 4. Estimation of correctness of prediction by polynomial model

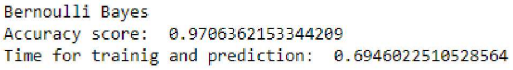
The result of the polynomial model, presented in Fig. 4, shows that the accuracy reached approximately 95.8% and 0.2439 seconds were spent to perform the classification (Fig. 5).



```
Gaussian Bayes
Accuracy score:  0.8591625883632409
Time for trainig and prediction:  2.104794502258301
```

Figure 5. Estimation of the correctness of the prediction by the Gaussian model

In fig. 5 shows that the accuracy of the classification by the Gaussian model reached approximately 85.9% and 2,104 seconds were spent on the classification (Fig. 6).



```
Bernoulli Bayes
Accuracy score:  0.9706362153344209
Time for trainig and prediction:  0.6946022510528564
```

Figure 6. Estimation of the correctness of the prediction of the Bernoulli model

The result of the Bernoulli model, presented in Fig. 6, shows that the accuracy reached approximately 97.06% and it took 0.695 seconds to perform the classification.

IV. Discussion of the results of the experiments

Given all the results of the experiments conducted in the previous section, we can say that the naive Bayesian classifier Bernoulli best solves the problem of classification of spam messages, as this problem is most common in various messengers and online platforms, where spam attacks spread quickly across the platform, so to resist, that is, to block at the beginning of the attack, you need to quickly adapt the template to new words.

Consider the difference between the three different functions of the Naive Bayes family that we used in the previous section, namely the Bernoulli classifier, the Gaussian class, and the polynomial classifier. The paper determined the correctness and accuracy of each of them to understand which of them is the best for filtering spam in messages.

To compare different variations of the naive Bayesian method classifier, accuracy estimates, program execution time, and a confusion matrix should be used, which shows the percentage of positive / negative true and false values.

First, a comparison of the accuracy of each method in tabular and graphical form should be presented. Each algorithm first learns from the training data, and then tests on the test. To determine the accuracy of the algorithms, you need to use the built-in sklearn.naive_bayes library .score () function. The input function takes two parameters: X and Y, which are the educational part of the sample. The function returns the average accuracy of the given data.

TABLE 1. Comparison Of Accuracy Of Performance Of Different Variations Of Naive Bayes

Method	Accuracy	Time
MultinomialNB	0.9575856	0.3036868
GaussianNB	0.86405655	2.02284264
BernoulliNB	0.974986405	0.8595001

In the table. 1 presents a comparison of the accuracy and execution time obtained during testing of each method from the naive Bayes family. For a better understanding, each of the relationships should be presented in the form of bar charts (Fig. 7).

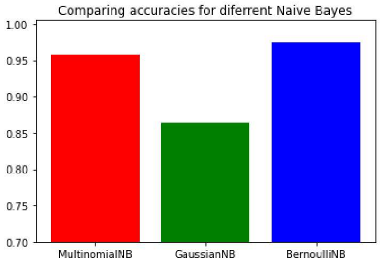


Figure 7. Comparison of the accuracy of different models of the Bayesian classifier

In fig. Figure 7 presents a comparison of the accuracy of classification with different variations of the Bayesian classifier models. There are three variations, namely: MultinomialNB, GaussianNB, BernoulliNB on the X axis. On the B axis, the value of accuracy. From fig. 7 shows that Bernoulli's model performed the prediction with greater accuracy. In turn, the polynomial model performed the prediction with slightly less accuracy than Bernoulli. Specifically, the Gaussian method of classification of the naive Bayes family lags behind in comparison with other models.

In fig. 8 presents a visualization of the comparison of the execution time of each classifier model.

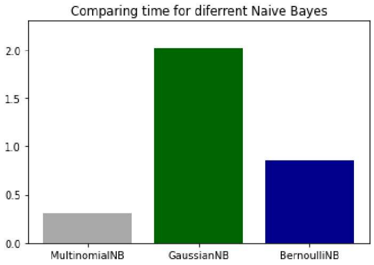


Figure 8. Comparison of program execution time by different models

In fig. 8 presents a comparison of the classification execution time with different variations of the Bayesian classifier models. Variations such as MultinomialNB, GaussianNB, BernoulliNB along the X axis are presented. From fig. 8 shows that the Gaussian model performed the classification for the longest, more than two seconds, in turn, the multinomial model performed the classification on the same data in about 0.3 seconds. The execution time of the Bernoulli model is about 0.9 seconds.

Conclusion

The presented study considered widely used variations of the naive Bayes method and their belonging to the problem of classification of e-mail that contains spam. Descriptions of algorithms, and also comparison of their productivity are presented. The results of the experiments showed very promising results, especially in two of the three functions of the Bayesian classifier.

It is established that the current problem of spam filtering in e-mails is best solved by the Bernoulli model of the naive Bayes method, because this model has achieved the highest accuracy and productivity. The Bayesian multinomial model, despite its slightly lower accuracy, is also noteworthy.

The naive Bayesian classifier method has some advantages over other classifiers that can be used for spam filtering. Thus, we can distinguish the main ones: the prediction of the test data set class and the naive Bayes classifier. Predicting the independence of the naive Bayes classifier compared to other models, such as logistic regression, the method works better. Accordingly, less training data is required for work. The classifier works well in the case of categorical input variables compared to numeric variables, because the numeric variables have a normal distribution.

References

[1] A. McCallum, K. Nigam, et al., “A comparison of event models for naive bayes text classification,” in AAAI-98 workshop on learning for text categorization, vol. 752, 1998, pp. 41–48.

[2] G. Cormack, “Email spam filtering: a systematic review”, Found Trends Inf Retr, 1(4), 2008, pp. 335–455.

[3] L. Wenbin, N. Zhong, “Spam Filtering and Email-Mediated Applications”, Web Intelligence Meets Brain Informatics Lecture Notes in Computer Science, pp. 382–405. URL: https://doi.org/10.1007/978-3-540-77028-2_23

[4] N. Boyko, P. Telishevskyi, B. Kushka, “Analysis of recommendation system methods for accuracy of predicted estimates”, CEUR Workshop Proceedings, 2021, pp. 1878–1888.

[5] I. Androutsopoulos, J. Koutsias, K.V. Chandrinos, C.D. Spyropoulos, “An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with encrypted personal e-mail messages”, in: Proc. of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000), 2000, pp. 160–167. URL: https://doi.org/10.1145/345508.345569

[6] I. Androutsopoulos, P. Georgios, E. Michelakis, “Learning to filter unsolicited commercial e-mail”, Technical Report 2004/2, NCSR Demokritos00, 2004.

[7] R. Bergman, M. Griss, C. Staelin, “A personal email assistant. Technical Report HPL-2002-236”, HP Labs Palo Alto, 2002. URL: http://citeseer.ist.psu.edu/bergman02personal.html.

[8] T. Berners-Lee, J. Hendler, O. Lassila, “The Semantic Web: a new form of Web content that is meaningful to computers will unleash a revolution of new possibilities”, Scientific American, 284(5), 2001, pp. 34–43. URL: https://doi.org/10.1038/scientificamerican0501-34.

[9] P. O. Boykin, V. Roychowdhury, “Personal email networks: an effective anti-spam tool”, IEEE Computer, 38(4), 2005, pp. 61–68. URL: https://doi.org/10.1109/MC.2005.132

[10] D. Chris, C.H. Robert, “Cost curves: an improved method for visualizing classifier performance”,. Machine Learning, 65(1), 2006, pp. 95–130. URL: https://doi.org/10.1007/s10994-006-8199-5.

[11] N. Boyko, “Application of mathematical models for improvement of “cloud” data processes organization”, in Mathematical Modeling and Computing, Vol. 3(2), 2016, pp. 111–119. DOI: https://doi.org/10.23939/mmc2016.02.111

[12] P. Domingos and M. Pazzani, “On the optimality of the simple bayesian classifier under zero-one loss,” Machine learning, vol. 29, no. 2 –3, 1997, pp. 103–130.