COMPUTATIONAL MATHEMATICS

MSC 68T50 **DOI: 10.14529/jcem210203**

ANALYSIS OF MACHINE LEARNING MODELS BY SOLVING THE TEXT DATA CLASSIFICATION PROBLEM

A. V. Pchelin[1], andrej.pchelin@yahoo.com,
N. A. Kononov[1], kononoff.174@yandex.ru,
V. S. Serova[1], vladislava.serova.98@mail.ru,
E. V. Bunova[1], albv70@mail.ru,
A. D. Marchenko[1], vetrenik1@gmail.com,
A. E. Shevchenko[1], aleksandr_shevchenko_2012@mail.ru
[1]South Ural State University, Chelyabinsk, Russian Federation

The article presents a study of usage of machine-learning models for the classification of text data on the example of the problem of classifying requests to technical support through a chat bot of a mobile application. The following methods were considered: Naive Bayes classifier, K-Nearest Neighbors algorithm (KNN)), Decision Tree, Random Forest, Support Vector Machines (SVM) and the method of Logistic Regression (Logistic Regression), as well as 21 models based on above methods. The best machine-learning model for classifying text requests to the technical support chat bot turned out to be a model, based on the Logistic Regression method, and model, based on the Random Forest Classifier. The Complement Naive Bayes model of the Naive Bayes group of models showed the shortest tuning time among the trained models with an acceptable accuracy. The proposed methodology can be used to analyze and classify text data.

Keywords: text classification; machine learning methods; regression; natural language; text data analysis.

Introduction

Nowadays, the natural language processing is an actively developing branch of information technology. Natural language processing issues are especially relevant when implementing the tasks of automating the functions of the execution of public administration, in solving socially significant issues, as well as to improve quality and reduce cost of companies' services provision. Natural language processing methods are based on mathematical statistics, probability theory, computational mathematics and graph theory, are an integral part of the field of artificial intelligence knowledge and used as a platform for deep analysis and text processing. Their use allows us to find patterns in large datasets, reduce data processing time and carry out classification.

The task of classifying text in natural language is relevant and has great development prospects, since it provides the ability to fully automate a wide range of business processes, for example, automatic text classification and organizing a collection of text questions and answers for faster processing of user requests. Text classification is used to solve a wide range of problems as sorting texts according to their content, placing them in certain groups, classifying records with given attributes, etc. [1]. When solving problems of textual

data classification, often-used tools are machine-learning methods that effectively solve the problem of assigning an object to one of the predefined classes based on its formalized features.

1. Research of Text Classification Methods

At the moment, the technology stack for performing the task of classifying text data is very diverse and is based on the use of various machine learning methods. Some of the most common methods used to solve text classification problems are:

1) Method "Naive Bayesian classifier (Naive Bayes)";
2) K-nearest neighbors method;
3) Decision Tree Classifier method;
4) Random Forest method;
5) Support Vector Machine;
6) Logistic Regression method.

The topic of description and use of machine learning methods for text classification is investigated intel works of many Russian and foreign authors. In [13–19], comparative characteristics of the usage of the above mentioned methods are given, which show that at present there is no single generally accepted effective method.

For example, in [13] the results of the experiment are presented, which show that the Naive Bayesian classifier and the Support Vector Machine have good accuracy of use, which is 90 and 91.5%, respectively, in the recognition of spam messages compared to the Method of K-nearest neighbors, the accuracy of which is 55.75%.

In [14], at the stage of classification of textual data in the field of medicine, the authors used seven machine learning methods (Seven machine learning methods Bagging, J48, Jrip, Logistic Model Trees (LMT), Logistic Regression, Linear Regression and Support Vector Machine (SVM)).

The accuracy of the best classifier (Logistic Model Trees) when analyzing the authenticity of notes was 74%. The Bayesian classifier and the k-nearest neighbors classifier showed lower accuracy. In [15–17], the K-nearest neighbor algorithm (KNN) for classification was used, which showed good classification accuracy, and in [8], KNN method was used to calculate the level of customer satisfaction with commercial banks in the region, which made it possible to improve the quality of services provided. At the stage of classification and evaluation of electronic correspondence in [18], 5 algorithms were tested: Naive Bayes classifier, KNN, SVM, Decision tree, Random forest. As a result of the study of these machine learning algorithms, the Random Forest algorithm showed the best result, which is 93%. The article [10] describes the application of random forest ("random forest") and SVM algorithms from the scikit learn library for the classification of news texts. The best obtained classifier for the classification of media texts turned out to be a model based on the support vector machine. And the most common, according to the authors of [19], a useful method for solving the problem of binary classification is a logistic regression. The logistic regression can be used for various classification tasks such as spam detection, disease prediction, loan outcome prediction, etc.

An analysis of the usage of machine learning methods showed that they have advantages and disadvantages, which are shown in Table 1.

Table 1

Advantages and disadvantages of groups of machine learning methods for text classification

Machine learning method name	Dignity	disadvantages
Naive Bayes	classification is performed in a short time, requiring a minimum of computing power; requires less training data, for example, compared to logistic regression.	the problem of "zero frequency"; the assumption of the independence of features.
K-nearest neighbors	does not require pre-training; ease of use.	"Curse of Dimension"; high complexity of one forecast.
Decision Tree Classifier	does not require data preparation and is easy to understand; uses a "white box" model, that is, the results can be explained using Boolean logic, in contrast to some artificial neural networks.	the problem of retraining due to the creation of complex structures when building a tree; inefficient distribution of weights for data that includes categorical variables.
Random Forest	efficient processing of data with a large number of features and classes; internal assessment of the generalizability of the model; high parallelizability and scalability.	large size of the resulting models. It takes O (K) memory to store the model, where K is the number of trees.
Support Vector Machines	the principle of the optimal dividing hyperplane leads to maximization of the dividing strip width, and, therefore, to a more confident classification; the convex quadratic programming problem is well studied and has a unique solution.	instability to noise; it is necessary to select the constant C using cross-validation; there is no feature selection.
Logistic Regression	fast learning; ease of use	only applicable if the solution is linear; the algorithm assumes that the input functions are mutually independent (no collinearity).

Thus, we will conduct a study of the methods presented in Table 1 for the most effectiveness for solving the problem and creating a classification system.

2. Methodology for the Classification of Text Data

For efficient and fast execution of the task of classifying text data, consider a methodology that contains the following steps.

Stage 1. Data preprocessing.

The initial data for the study is a database of text records (or calls) to the technical support chatbot on problems that arose during the use of a banking application of a company engaged in bank payments.

The marked data contains the following fields: Header - header; Body - the appeal itself; Class - the class of appeal (qr-codes, commission, checks, call back, refund, reviews, payment failed). User requests are initially manually assigned to a class, marking up the dataset, and then the model will be trained on this dataset and will itself determine which class this or that request belongs to. The data volume includes 2504 records, 2315 of which are unique records containing 5867 word forms.

The classification accuracy, in addition to the selected machine learning model and its parameters, is influenced by the amount of initial data for training the classifier model, the quality and purity of this data. Therefore, preliminary processing of the initial data is one of the most important stages. Thus, in the course of preliminary processing of text data, it is necessary to implement preliminary automatic processing, including:

1) splitting the text into words;
2) correction of typos in words;
3) replacing dates, times and amounts with keywords;
4) removal of unrecognized numeric data;
5) making substitutions from the dictionary of substitutions;
6) removal of punctuation marks;
7) removal of service parts of speech;
8) reduction of words to their initial form;
9) combining words back into text.

To date, there is no ready-made tool for correcting typos that would show close to 100% efficiency for any input data. For example, there are the following solutions to fix typos: Yandex.Speller, JamSpell, PyEnchant and DeepPavlov. The first solution, Yandex.Speller, requires an Internet connection and has restrictions on commercial use, and JamSpell is a paid product, in the free version of which efficiency is sufficient only for correcting words with one typo. DeepPavlov is a powerful and free tool that takes into account the context, does not require an Internet connection, but, firstly, it requires a slightly outdated version of Python with a virtual environment to work, and, secondly, it requires a lot of RAM and processing power.

Therefore, to correct typos in this study, we used the implementation of the Enchant library for the Python programming language – PyEnchant, designed to correct errors in words of various languages, which also has support for correcting words in Russian with high accuracy. It is free, does not require an Internet connection (after downloading and installing on a PC / server), and also does not demand the power of your PC / server. Disadvantages of this solution are no contextualization, no correction of separated words by mistake, correction of words with more than one typo is difficult.

In the set of initial data, a lot of words glued and separated by mistake, words with two or more typos, borrowed words with typos that the system did not recognize were found by mistake. To solve this problem, a number of substitution dictionaries were compiled for words and word forms, for which the algorithm does not pick up the necessary corrections. The replacement for borrowed words was carried out immediately by a word in the initial form. The percentage of such cases of word forms is 3% to 7%. After applying these methods, it was possible to filter out more than 97% of words with typos.

Then the service parts of speech were removed, because they do not have a significant effect on the meaning of the text, but they increase the noise level and the number of word forms. All words are reduced to their initial form, because different forms of the same word increase the number of word forms in the text. When solving both problems, a free library for the Python programming language, PyMorphy2, was used. This library implements the functionality of morphological parsing, and also supports the reduction of words to their initial form based on the data obtained during morphological analysis. The disadvantages of the library can also be attributed to the lack of context. With the help of this library, a morphological analysis of all words in the initial data was carried out: the words of the service parts of speech were removed, and the rest were reduced to their initial form.

Regular expressions have been applied to replace meaningful numeric data such as dates, times, amounts, quantities, and remove unrecognized ones, as well as to remove punctuation marks. First, a search was made for dates and times, replacing them with the words date and time, then the search for sums and quantities, they were replaced with the words sum and quantity. Unrecognized numeric data has been removed, as having punctuation marks.

After applying all the above methods, 2,176 unique records remained, and the number of word forms was reduced to 2951. Preliminary word processing gives significant results even on a small amount of data. On larger amounts of data, the effect will be more noticeable as the parameters of the initial data are significantly improved, which subsequently influenced the quality of training a machine learning model for text classification.

Algorithm and program code for preprocessing data.

The preprocessing stage contains the following steps, implemented using the Python programming language.

Step 1. Import .csv raw data into a DataFrame.

Step 2. Removing null lines.

Step 3. Casting the DataFrame object to a list for ease of processing, followed by converting the strings to lowercase.

Step 4. Removing all extraneous characters except alphabetic and numeric by means of a regular expression using the Re library.

Step 5. Splitting each element of the list into separate words using the split () method.

Step 6. Correcting mistakes in words using a dictionary of substitutions.

Step 7. Correcting mistakes in words using the Enchant library.

Step 8. Using the PyMorphy2 library, lemmatization of each word, ie. reduction of a word to its initial form, which increases the classification accuracy due to a sharp reduction in

word forms in the text.

Step 9. Using the PyMorphy2 library, determine the part of speech for each word, and, if it turns out to be a service part of speech, remove it from the list.

Step 10. Using the Re library, replacing meaningful information such as dates, times, sums with the words DATE, SUM, TIME.

Stage 2. Data splitting into training and test samples.

The data were divided into training and test samples in a ratio of 70% (training) to 30% (test). Splitting is done randomly using the train_test_split () function of the Scikit-Learn library.

Stage 3. Training models on the training sample.

To create a classification system, six groups of machine learning were selected, presented in Table 1.

The Scikit-Learn package was used to implement the software implementation of the group of methods. Scikit-Learn specializes in machine learning algorithms for solving supervised learning problems: classification (predicting a feature, the set of valid values of which is limited) and regression (predicting a feature with real values), as well as for unsupervised learning tasks: clustering (splitting data into classes , which the model will determine itself), dimensionality reduction (presentation of data in a space of a lower dimension with minimal loss of useful information) and anomaly detection.

Training models using the LogisticRegression model, which showed the best accuracy analysis and classification of text using Scikit-Learn and Pandas library includes the following steps.

Step 1. Replacing class names with numbers.

Step 2. Vectorization of data (translation of data into vector representation).

Step 3. Building a classifier to assess its accuracy.

Step 4. Application of the classifier.

This line allows you to determine which class this message belongs to. The rest of the classifiers based on other machine learning models have similar program code.

Stage 4. Testing of models on a test sample.

Testing was carried out as follows: using the model predictions on the test sample, an error matrix was built. The error matrix shows which class the call belongs to (horizontal axis) and to which class the machine learning model assigned it (vertical axis). The color of the cell shows the number of correct answers: the color of the cell is lighter if the number of answers it contains is greater. Based on the error matrix, the accuracy of the model can be calculated. But it is much more convenient to calculate the accuracy using a function from the sklearn library: score (X_test.toarray (), y_test). For example, the obtained accuracy value of 0.7 means that 70% of the model's predictions turned out to be correct.

Fig. 1 shows an example of the resulting error matrix for the LogisticRegression model. An error matrix is a matrix that is used to evaluate machine learning classifier models based on test data for which the true values are known.

The main indicators of the matrix are:

1) true answers, when the model recognized the desired object class;

2) false answers, when the model assigned the recognized object to the wrong class.

In the error matrix, one can see the number of correctly recognized objects of each

class, the number of incorrectly recognized objects. One can also see to which class the model has assigned the incorrectly recognized objects.

Based on the error matrix, various model metrics can be calculated: overall model accuracy, total model error probability, individual accuracy and error probability for each of the classes.

The error matrix allows us to fully evaluate the models under consideration.

The error matrix, shown in Figure 1, diagonally displays the number of predictions corresponding to the correct class, and the remaining values can be used to determine the number of false answers for each class.

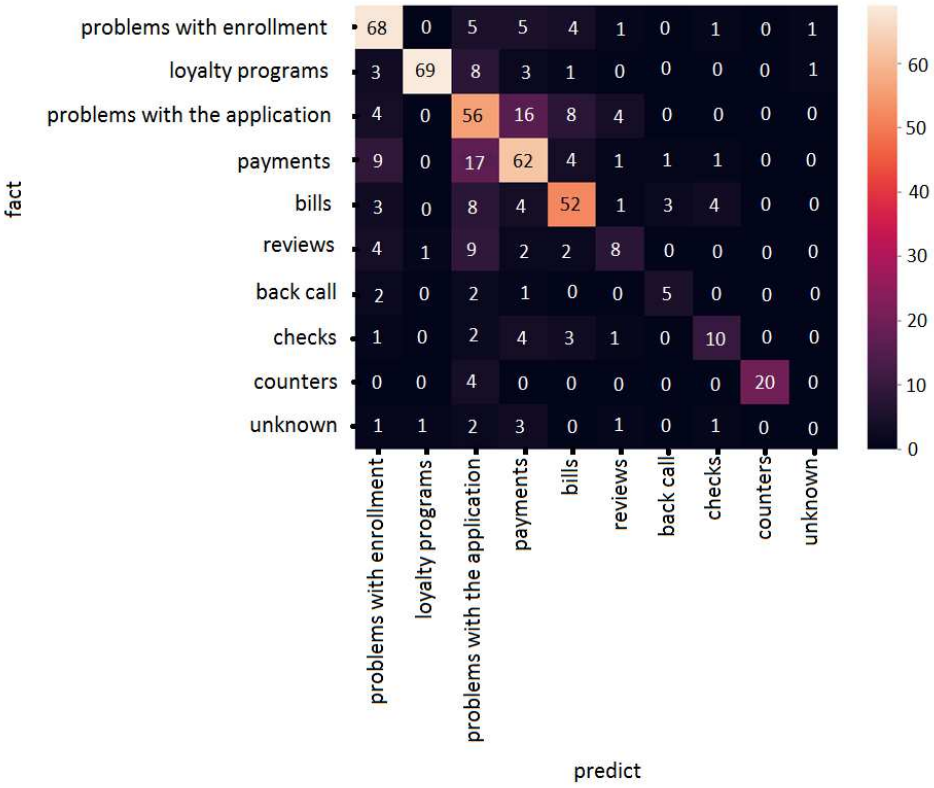

Fig. 1. Matrix of errors for the LogisticRegression model

As can be seen from the error matrix, that there is a small number of false positive and false negative values related to the true positive and true negative values. Accordingly, the accuracy of building the model is 0.7, and, therefore, this model, in general, correctly classifies all the data of the test sample. Accordingly, the accuracy of the model is 0.7, and, therefore, this model correctly classifies the test sample data in 70% of cases.

3. Analysis of Experimental Results

Table 2 shows the results of testing models in terms of accuracy and training duration. Accuracy is determined by the ratio of the number of correctly identified positive objects to the total number of positive objects that the classifier attributed to this class.

Table 2

Analysis of experimental results based on the classification by accuracy and running time of the trained model

Group	Model	Accouracy	Time (sec)
Logistic Regression	LogisticRegression	0.7	1.95
	LogisticRegressionCV	0.7	64.74
Random Forest	RandomForestClassifier	0.68	1.59
	RandomForestRegressor	0.54	19.7
Support Vector Machine	LinerSVC	0.67	0.52
	NuSVC	0.65	8.57
	SVC	0.63	8.58
	NuSVR	0.25	3.24
Naïve Bayes	ComplementNB	0.65	0.03
	MultinomialNB	0.64	0.03
	GaussianNB	0.44	0.36
	BernoalliNB	0.41	0.06
Decision Tree Classifer	DecisionTreeClassifer	0.6	0.37
	ExtraTreeClassifer	0.59	0.05
	ExtraTreeRegressor	0.36	0.39
	DecisionTreeRegressor	0.34	0.42
K-nearest neighbors	KNeighborsClassifer	0.56	1.99
	KNeighborsRegressor	0.42	1.99
	RadiusNeighborsClassifer	0.17	3.13
	NearestCentroid	0.13	0.06

Experiments show that the most accurate methods of text analysis and classification when solving the problem are the Logistic Regression models with an accuracy of 0.7 and the Random Forest Classifier with an accuracy of 0.68. A lower accuracy (0.67), but also a shorter training time (0.52 s), was shown by the LinearSVC model of the Support Vector Machines. The shortest running time of the trained model (0.03 s) with an acceptable accuracy (0.65) was shown by the ComplementNB model of the Naive Bayes group of models.

Conclusion

As a result, the work was studied and 6 methods of classification of text data have been tested: NaiveBayes classifier, KNN, Decision tree, Random forest, SVM, the method of logistic regression, as well as 21 models that are part of the above methods. The best obtained classifier for classifying text records (or calls) to the technical support chatbot turned out to be a model based on the Logistic Regression method. Also, to solve the problem, a code in the Python programming language was written, which contains powerful

tools for solving various problems of data science, using which various models can be built for obtaining the required results. The proposed methodology can be used to analyze and classify text data.

References

1. Lakshmi R., Baskar S. DIC-DOC-K-means: Dissimilarity-based Initial Centroid Selection for DOCument Clustering using K-means for Improving the Effectiveness of Text Document Clustering. *Journal of Information Science*, 2019, vol. 45, pp. 818–832. DOI: 10.1177/0165551518816302.
2. Celyh V. R., Shishkovec S. S., Uskov M. O., Voroncov K. V. [Additive Regularization of a Naïve Bayes Classifier]. *MLSD' 2016. Proceedings of the Ninth International Conference. Moscow, October 03–05, 2016*. Moscow, 2016, pp. 414–416. (in Russian)
3. Han-joon Kim, Jiyun Kim, Jinseog Kim, Pureum Lim. Towards Perfect Text Classification with Wikipedia-based Semantic Naïve Bayes Learning. *Neurocomputing*, 2018, vol. 315, pp. 128–134. DOI: 10.1016/j.neucom.2018.07.002.
4. Zinnatullin A. F. [Using a Naïve Bayes Classifier in Text Processing]. *Information Technology and Mathematical Modeling. Proceedings of the XVI International Conference named after A. F. Terpugova. Kazan, September 29, 2017*. Kazan, 2017, pp. 50–54. (in Russian)
5. Burlakov M. E., Golubykh D. A., Osipov M. N. Naive Bayesian Classifier Adaptatoin for E-Mail Classification Mechanism. *Infokommunikacionnye tehnologii*, 2016, vol. 14, no. 2, pp. 199–203. DOI: 10.18469/ikt.2016.14.2.15. (in Russian)
6. Denisova D. S. [Automatic Language Processing. Classification of the Text. Naïve Bayes Classifier]. *Synergy of Science*, 2018, no. 19, pp. 1410–1414. (in Russian)
7. Sajfullin M. A., Sulejmanova A. M. Realization of a Naïve Bayesian Classifier for Financial News Articles in Ruby Programming Language. *Information Technologies in Management and Economics*, 2019, no. 4, pp. 34–44. (in Russian)
8. Kunitsyna N. N., Metel Yu. A. Applying the Method of Intellectual Text Analysis in Estimating the Level of Customer Satisfaction at Commercial Banks. *Vestnik Saratov State Socio-Economic University*, 2018, no. 2 (71), pp. 149–155. (in Russian)
9. Grishanov K. M., Belov Yu. S. [Method of Classification K-NN and its Application in Character Recognition]. *Fundamental Problems of Science. Proceedings of the International Scientific and Practical Conference. Tyumen', May 15, 2016*. Ufa, 2016, pp. 30–33. (in Russian)
10. Motovskikh L. V. Mediatexts Classification Using Machine Learning. *Vestnik of Moscow State Linguistic University. Humanities*, 2020, issue 12 (841), pp. 124–130. (in Russian)
11. Strelnikov V. G., Trunov A. S. [Application of Logistic Regression to the Problem of Classification of Texts of Court Decisions]. *Telecommunications and Information Technologies*, 2017, vol. 4, no. 2, pp. 75–78. (in Russian)
12. [*Logistic Regression and ROC Analysis*], available at: https://basegroup.ru/community/articles/logistic (accessed on May 5, 2021). (in Russian)

13. Ping Zhang, Yajie Fang. Research on Text Classification Algorithm Based on Machine Learning. *Journal of Physics: Conference Series, Advanced Algorithms, Analysis Model, Optimization Design and Its Application*, 2020, vol. 1624, pp. 10–18. DOI: 10.1088/1742-6596/1624/4/042010.

14. Duy Van Le, James Montgomery, Kenneth C Kirkby, Joel Scanlan. Risk Prediction using Natural Language Processing of Electronic Mental Health Records in an Inpatient Forensic Psychiatry Setting. *Journal of Biomedical Informatics*, 2018, vol. 86, pp. 49–58. DOI:10.1016/j.jbi.2018.08.007.

15. Pestian J., Nasrallah H., Matykiewicz P. , Bennett A., Leenaars A., Suicide Note Classification Using Natural Language Processing: A Content Analysis. *Biomedical Informatics Insights*, 2010, vol. 3, pp. 19–28. DOI: 10.4137/BII.S4706

16. Amer Ali A., Abdalla Hassan I. A Set Theory Based Similarity Measure for Text Clustering and Classification. *Journal of Big Data*, 2020, vol. 7, article no. 74. DOI: 10.1186/s40537-020-00344-3.

17. Aggarwal C. C., Zhai C. A Survey of Text Classification Algorithms. *Mining Text Data*. Boston, MA, Springer, 2012, pp. 163–222. DOI: 10.1007/978-1-4614-3223-4_6.

18. Radivanovich D. A., Sukhorukova I. G. [Multi-criteria email filtering]. *Information technologies. Proceedings of the 83rd Scientific and Technical Conference of the Teaching Staff, Researchers and Graduate Students*, 2019, no. 74, pp. 175–178. (in Russian)

19. Palmin P. A. [Using Logistic Regression to Solve a Classification Problem using Python as an Example]. *Scientific and Practical Research*, 2019, no. 8-7 (23), pp. 51–57. (in Russian)

Andrey V. Pchelin, Student at the Department of Applied Mathematics and Software Development, South Ural State University (Chelyabinsk, Russian Federation), andrej.pchelin@yahoo.com.

Nikita A. Kononov, Student at the Department of Applied Mathematics and Software Development, South Ural State University (Chelyabinsk, Russian Federation), kononoff.174@yandex.ru.

Vlada S. Serova, Undergraduate at the Department of Applied Mathematics and Software Development, South Ural State University (Chelyabinsk, Russian Federation), vladislava.serova.98@mail.ru.

Elena V. Bunova, PhD (Techn), Docent, Associate Professor at the Department of Applied Mathematics and Software Development, South Ural State University (Chelyabinsk, Russian Federation), albv70@mail.ru.

Anton D. Marchenko, Graduate Student at the Department of Applied Mathematics and Software Development, South Ural State University (Chelyabinsk, Russian Federation), vetrenik1@gmail.com.

Aleksandr Y. Shevchenko, Student at the Department of Applied Mathematics and Software Development, South Ural State University (Chelyabinsk, Russian Federation), aleksandr_shevchenko_2012@mail.ru.

Received June 10, 2021.

УДК 004.02 **DOI: 10.14529/jcem210203**

АНАЛИЗ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ ПРИ РЕШЕНИИ ЗАДАЧИ КЛАССИФИКАЦИИ ТЕКСТОВЫХ ДАННЫХ

А. В. Пчелин, Н. А. Кононов, В. С. Серова, Е. В. Бунова, А. В. Марченко, А. Е. Шевченко

В статье представлено исследование использования моделей машинного обучения для классификации текстовых данных на примере задачи классификации обращений в техническую поддержку через чат-бот мобильного приложения. Были рассмотрены следующие методы: наивный бейсовский классификатор (Naive Bayes classifier), метод k-ближайших соседей (k-nearest neighbors algorithm, KNN)), дерево принятия решений (Decision tree), метод случайный лес (Random forest), метод опорных векторов (SVM), метод логистическая регрессия (Logistic Regression), а также 21 модель, входящая в состав вышеперечисленных методов. Наилучшей моделью машинного обучения для классификации текстовых записей (обращений) в чат-бот технической поддержки оказалась модель на основе метода логистическая регрессия (Logistic Regression), которая была построена средствами языка программирования Python.

Ключевые слова: классификация текста; методы машинного обучения; регрессия; естественный язык; анализ текстовых данных.

Литература

1. Lakshmi, R. DIC-DOC-K-means: Dissimilarity-based Initial Centroid Selection for DOCument Clustering using K-means for Improving the Effectiveness of Text Document Clustering / R. Lakshmi, S. Baskar // Journal of Information Science. – 2019. – V. 45. – P. 818–832.
2. Целых, В. Р. Аддитивная регуляризация наивного байесовского классификатора / В. Р. Целых, С. С. Шишковец, М. О. Усков, К. В. Воронцов // Управление развитием крупномасштабных систем (MLSD'2016). Материалы Девятой международной конференции. Москва, 03–05 октября 2016 года. – М., 2016. – С. 414-416.
3. Kim, Han-joon. Towards Perfect Text Classification with Wikipedia-based Semantic Naïve Bayes Learning / Han-joon Kim, Jiyun Kim, Jinseog Kim, Pureum Lim // Neurocomputing. – 2018. – V. 315. – P. 128–134.
4. Зиннатуллин, А. Ф. Применение наивного байесовского классификатора в обработке текстов / А. Ф. Зиннатуллин // Информационные технологии и математическое моделирование (ИТММ-2017). Материалы XVI Международной конференции имени А. Ф. Терпугова. Казань, 29 сентября 2017 года. – Казань, 2017. – С. 50–54.

5. Бурлаков, М. Е. Адаптация наивного байесовского классификатора к механизму классификации электронных сообщений / М. Е. Бурлаков, Д. А. Голубых, М. Н. Осипов // Инфокоммуникационные технологии. – 2016. – Т. 14, № 2. – С. 199–203.

6. Денисова, Д. С. Автоматическая обработка языка. Классификация текста. Наивный байесовский классификатор / Д. С. Денисова // Синергия Наук. – 2018. – № 19. – С. 1410–1414.

7. Сайфуллин, М. А. Реализация наивного байесовского классификатора новостных публикаций в финансовой сфере на языке программирования Ruby / М. А. Сайфуллин, А. М. Сулейманова // Информационные технологии в управлении и экономике. – 2019. – № 4. – С. 34–44.

8. Куницына, Н. Н. Применение метода интеллектуального текстового анализа в оценке уровня удовлетворенности клиентов коммерческих банков / Н. Н. Куницына, Ю. А. Метель // Вестник Саратовского государственного социально-экономического университета. – 2018. – № 2 (71). – С. 149–155.

9. Гришанов, К. М. Метод классификации K-NN и его применение в распознавании символов / К. М. Гришанов, Ю. С. Белов // Фундаментальные проблемы науки. Сборник статей Международной научно-практической конференции. Тюмень, 15 мая 2016 года. – Уфа, 2016. – С. 30–33.

10. Мотовских, Л. В. Классификация медиатекстов с использованием машинного обучения / Л. В. Мотовских // Вестник МГЛУ. Гуманитарные науки. – 2020. – Вып. 12 (841). – С. 124–130.

11. Стрельников, В. Г. Применение метода логистической регрессии для задачи классификации текстов судебных решений / В. Г. Стрельников, А. С. Трунов // Телекоммуникации и информационные технологии. – 2017. – Т. 4, № 2. – С. 75–78.

12. Логистическая регрессия и ROC-анализ [Электронный ресурс] – URL: https://basegroup.ru/community/articles/logistic (дата обращения: 05.05.2021).

13. Zhang, P. Research on Text Classification Algorithm Based on Machine Learning / Ping Zhang, Yajie Fang // Journal of Physics: Conference Series, Advanced Algorithms, Analysis Model, Optimization Design and Its Application. – 2020. – V. 1624. – P. 10–18.

14. Le, Duy Van. Risk Prediction using Natural Language Processing of Electronic Mental Health Records in an Inpatient Forensic Psychiatry Setting / Duy Van Le, James Montgomery, Kenneth C Kirkby, Joel Scanlan // Journal of Biomedical Informatics. – 2018. – V. 86. – P. 49–58.

15. Pestian, J. Suicide Note Classification Using Natural Language Processing: A Content Analysis / J. Pestian, H. Nasrallah, P. Matykiewicz, A. Bennett, A. Leenaars // Biomedical Informatics Insights. – 2010 – V. 3. – P. 19–28.

16. Amer, A. A Set Theory Based Similarity Measure for Text Clustering and Classification / A. Amer, H. Abdalla // Journal of Big Data. – 2020. – V. 7. – Article № 74.

17. Aggarwal, C. C. A Survey of Text Classification Algorithms / C. C. Aggarwal, C. Zhai // Mining Text Data. – Boston, MA: Springer, 2012. – P. 163–222.

18. Радиванович, Д. А. Многокритериальная фильтрация электронной корреспонденции / Д. А. Радиванович, И. Г. Сухорукова // Информационные технологии. Материалы 83-й научно-технической конференции профессорско-преподавательского состава, научных сотрудников и аспирантов. – 2019. – № 74. – С. 175–178.

19. Пальмин, П. А. Использование логистической регрессии для решения задачи классификации на примере Python / П. А. Пальмин // Научно-практические исследования. – 2019. – № 8-7 (23). – С. 51–57.

Пчелин Андрей Владимирович, студент кафедры прикладной математики и программирования, Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), andrej.pchelin@yahoo.com.

Кононов Никит Александровича, студент кафедры прикладной математики и программирования, Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), kononoff.174@yandex.ru.

Серова Влада Сергеевна, магистрант кафедры прикладной математики и программирования, Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), vladislava.serova.98@mail.ru.

Бунова Елена Вячеславовна, кандидат технических наук, доцент, доцент кафедры прикладной математики и программирования, Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), albv70@mail.ru.

Марченко Антон Дмитриевич, аспирант кафедры прикладной математики и программирования, Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), vetrenik1@gmail.com.

Шевченко Александр Евгеньевич, студент кафедры прикладной математики и программирования, Южно-Уральский государственный университет (г. Челябинск, Российская Федерация), aleksandr_shevchenko_2012@mail.ru.

Поступила в редакцию 10 июня 2021 г.