DOI 10.20544/TTS2021.1.1.21.p44
UDC 004.85:519.248]:656.062
Professional paper

Forecasting Number of Calls to the Call Center Using Machine Learning

Pavle Bugarčić[1], Slađana Janković[2] and Snežana Mladenović[3]

Abstract – **This paper presents a forecast of a number of call arrivals in the call center per hour using supervised machine learning. For the forecast, the WEKA machine learning software tool was used. The results of the forecast are verified using several methods, which shows very good results. Finally, the results of the forecast are presented graphically using Excel diagrams.**

Keywords – **Machine learning**, **Forecasting, WEKA**

I. Introduction

The call center has always been the electronic face of the company, and with the advancement of technology and the trend of digitalization, it has become a vital service for interacting with existing and future customers. In the modern world, call centers have become one of the most important forms of communication between companies and customers. The most common form of call centers is inbound call centers, where customers initiate calls and call center agents respond to those calls. The traffic profile of incoming calls is usually very dynamic. More precisely, the intensity of the incoming calls varies in different parts of the year, month, week and day. For this reason, different amounts of human and technical resources of the call center are required at different periods. This is mainly related to the optimum number of call center agents working simultaneously. To optimally plan the resources of the call center, it is first necessary to forecast the number of incoming calls in the future. This is exactly the goal of this paper.

Considering the very large number of incoming calls to the call center, for the problem of forecasting the number of calls to the call center can be said that is a Big Data problem. For this reason, one of the Big Data analytics techniques should be selected for the forecast. An important class of Big Data analytics is a predictive analytics based on supervised machine learning. This technique is used for the forecast in this paper, using the WEKA (Waikato Environment for Knowledge Analysis) software tool [1]. This data mining software is a collection of machine learning algorithms used in data mining operations. Two algorithms are used in this paper - RandomForest and Bagging. The implementation of the machine learning process in this paper is done on the example of a call center of a company TNT, which provided us relevant incoming calls number information, in the previous period of three years.

This paper is organized as follows. In section II the method of supervised machine learning is described. Section III shows a numerical example of forecasting the number of calls to the call center using a WEKA software tool. Concluding remarks are given in section IV.

II. Supervised Machine Learning Method

Machine learning solves the problem of how to make computers that automatically improve through experience. Today it is one of the fastest-growing technical fields, based on computer science and statistics, and is the core of artificial intelligence and data science. Recent progress in machine learning has been driven both by the development of new learning algorithms and theory and by the ongoing explosion in the availability of online data and low-cost computation [2].

There are several applications for machine learning, the most significant of which is predictive data mining. Every instance in any dataset used by machine learning algorithms is represented using the same set of features. The features may be continuous, categorical or binary. If instances are given with known labels (the corresponding correct outputs) then the learning is called supervised, in contrast to unsupervised learning, where instances are unlabeled [3].

The goal of supervised machine learning is to build a concise model of the distribution of class labels in terms of predictor features. The resulting classifier is then used to assign class labels to the testing instances where the values of the predictor features are known, but the value of the class label is unknown [4].

The machine learning process consists of the following stages: data preparation, model building, model validation, model testing and model implementation. Machine learning is an iterative process in which all of the above phases are repeated as many times as necessary. The repetition of these phases ends when all combinations of attributes, all available algorithms and algorithm parameter values are exhausted, or when a satisfactory performance model is reached. Once the model testing shows that the model has satisfactory performance, it can begin with its use for forecasting of the selected variable [5].

The most important step in the machine learning process is data preparation, which most influences the success of the process. Data preparation consists of clearing raw data from incomplete records or records with incorrect values, converting the data to the appropriate format, etc. Some of the

[1]Pavle Bugarčić is with the Faculty of Transport and Traffic Engineering, University of Belgrade, Vojvode Stepe 305, Belgrade, Serbia, p.bugarcic@sf.bg.ac.rs

[2]Slađana Janković is with the Faculty of Transport and Traffic Engineering, University of Belgrade, Vojvode Stepe 305, Belgrade, Serbia, s.jankovic@sf.bg.ac.rs

[3]Snežana Mladenović is with the Faculty of Transport and Traffic Engineering, University of Belgrade, Vojvode Stepe 305, Belgrade, Serbia, snezanam@sf.bg.ac.rs

input dataset file formats supported by WEKA are csv, arff, etc.

The building of each machine learning model consists of the following stages:

- defining the goal of the model, in line with the goals of predictive analytics;
- selection of the target variable, i.e. attribute from the dataset whose value we want to predict using machine learning model;
- selection of supervised machine learning algorithm, following the nature of the target variable and attributes;
- selection of relevant attributes of the dataset;
- preparation of datasets for learning (training) and testing models, according to the requirements of the chosen algorithm;
- model adjustment, i.e. values of hyperparameters specific to each type of machine learning algorithm;
- model training, that is, applying the selected machine learning algorithm to a training dataset to obtain model hyperparameters.

After that, it is possible to validate the model. WEKA provides several types of validation, such as the use of a training dataset, the use of a test dataset, cross-validation, and percentage splitting. The cross-validation type is used in this paper. Usually, the input dataset is divided into a training dataset and a test dataset. The cross-validation process only uses a training dataset. This process consists of the following stages:

- the available model training dataset is divided into *K* equal folds. It is usually divided into 10 folds (10-fold cross-validation);
- the model is trained on *K*-1 folds of data;
- the model is evaluated on one remaining subset of data;
- steps 2 and 3 are repeated *K* times. In each iteration, one fold of the data is taken for model validation purposes, while the rest (*K*-1 folds) is used for training. A different fold is always selected to be used to validate the model.
- model performance is calculated as the arithmetic mean of the performance obtained in the *K* iterations.

Several different measures can be used to evaluate the forecast, such as mean squared error, root mean squared error, mean absolute error, relative squared error, root relative squared error, relative absolute error and correlation coefficient [6].

The mean squared error is the most commonly used measure and is calculated as follows:

$$\boldsymbol{Mean\ squared\ error} = \frac{(p_1-a_1)^2+\cdots+(p_n-a_n)^2}{n} \qquad (1)$$

The values p_1, p_2, ..., p_n are the projected values of the target variable, and the values a_1, a_2, ..., a_n are the actual values of the target variable. The parameter n indicates the total number of elements of the input dataset.

The root mean squared error is calculated as follows:

$$\boldsymbol{Root\ mean\ squared\ error} = \sqrt{\frac{(p_1-a_1)^2+\cdots+(p_n-a_n)^2}{n}} \qquad (2)$$

Mean absolute error represents the average value of individual errors, without considering their sign. It is calculated as follows:

$$\boldsymbol{Mean\ absolute\ error} = \frac{|p_1-a_1|+\cdots+|p_n-a_n|}{n} \qquad (3)$$

The relative squared error represents the normalized total squared error and is calculated as follows:

$$\boldsymbol{Relative\ squared\ error} = \frac{(p_1-a_1)^2+\cdots+(p_n-a_n)^2}{(a_1-\bar{a})^2+\cdots+(a_n-\bar{a})^2} \qquad (4)$$

The mean values of the variables p and a are denoted by $\bar{\boldsymbol{p}}$ and $\bar{\boldsymbol{a}}$.

The root relative squared error is calculated as follows:

$$\boldsymbol{Root\ relative\ squared\ error} = \sqrt{\frac{(p_1-a_1)^2+\cdots+(p_n-a_n)^2}{(a_1-\bar{a})^2+\cdots+(a_n-\bar{a})^2}} \qquad (5)$$

The relative absolute error represents the normalized total absolute error and is calculated as follows:

$$\boldsymbol{Relative\ absolute\ error} = \frac{|p_1-a_1|+\cdots+|p_n-a_n|}{|a_1-\bar{a}|+\cdots+|a_n-\bar{a}|} \qquad (6)$$

The last measure of accuracy is the correlation coefficient, which takes values from -1 to 1. The highest correlation is shown by models where the absolute value of the correlation coefficient tends to be 1. On the other hand, when the value of this coefficient is closer to 0, the correlation is smaller and the model is worse. This coefficient is calculated as follows:

$$\boldsymbol{Correlation\ coefficient} = \frac{S_{PA}}{\sqrt{S_P S_A}} \qquad (7)$$

where:

$$S_{PA} = \frac{\sum_{i=1}^{n}(p_i-\bar{p})(a_i-\bar{a})}{n-1},\ S_P = \frac{\sum_{i=1}^{n}(p_i-\bar{p})^2}{n-1},\ S_A = \frac{\sum_{i=1}^{n}(a_i-\bar{a})^2}{n-1} \qquad (8)$$

The previous steps in the cross-validation process are done on the training dataset. The test dataset didn't play a role in the building of the model, nor in the validation of the model, so it is possible to evaluate the performance of the model obtained through the training using this dataset.

The next step is to compare the performance of the model in the case of using the training dataset and in case of using the test dataset. The model should not give significantly better results when using the training dataset than when using the test dataset. If the model performs very well on the training dataset but significantly worse on the test dataset, then there is a problem of over-matching.

Finally, to forecast the target variable for the future period, it is necessary to prepare an appropriate input dataset and apply the selected model to it.

III. A NUMERICAL EXAMPLE OF FORECASTING NUMBER OF CALLS TO THE CALL CENTER

This chapter presents a numerical example of forecasting the number of calls to a company call center using supervised machine learning. As mentioned above, in this paper the

WEKA software tool was used to this forecast. WEKA contains a large number of algorithms that can be used for this forecast. Some of the basic classes of algorithms implemented in WEKA are bayes, functions, lazy, meta, rules and trees.

The inbound dataset, based on which the forecast was made, is a collection of incoming call center calls over the last three years. The input dataset is divided into a training dataset and a test dataset. The training dataset consists of data for the first two years (October 2016 - September 2018), and the test dataset consists of data for the third year (October 2018 - September 2019). The number of calls per hour was monitored and therefore the incoming calls were grouped per hour. Each call is described by the following group of attributes: year, month, day of the month, day of the week, and the hour at which the call was made. So the target variable, in this case, is the number of calls per hour, and the attributes mentioned are influential factors.

As explained in the previous chapter, it is first necessary to form a model based on training dataset. The previous section describes ways to verify forecasting models. In this paper, the correlation coefficient, mean absolute error, root mean squared error, relative absolute error and root relative squared error are used for verification. The verification was first performed on the training dataset. A number of algorithms have been tested on the training dataset, including IBk, AdditiveRegression, Bagging, M5P, RandomCommittee, RandomSubSpace, DecisionTable, M5Rules, RandomForest, RandomTree and many others. Models based on the Bagging and RandomForest algorithms showed the best performances, so these algorithms are selected to form a model. The results of performance measurement in case of the application of the RandomForest and Bagging algorithms are shown in Table I.

TABLE I
PERFORMANCES OF THE TWO BEST MODELS FOR FORECASTING THE NUMBER OF CALLS TO THE CALL CENTER MEASURED ON THE TRAINING DATASET

Algorithm	Correlation coefficient	Mean absolute error	Root mean squared error	Relative absolute error (%)	Root relative squared error (%)
Random Forest	0.9588	1.5897	3.7153	15.4259	28.4004
Bagging	0.9573	1.5878	3.7827	15.4071	28.916

The parameter that has decisively influenced the choice of the mentioned algorithms is the correlation coefficient, so a comparison will be made based on it. It can be observed that the correlation coefficient for both algorithms is very close to 1, which indicates a very high correlation and therefore very high forecast accuracy when both algorithms are used. Slightly better performance is indicated by the RandomForest algorithm.

The verification was then performed on the test dataset as well. The verification results for the RandomForest and Bagging algorithms are shown in Table II. As expected, both algorithms show slightly worse results compared to results from Table I. However, the difference is not very large, so it cannot be said that there was a over-matching problem. In this case, the correlation coefficients are also close to 1, so it can be said that the forecast is very reliable. In this case, the RandomForest algorithm also shows slightly better performance.

TABLE II
PERFORMANCES OF THE TWO BEST MODELS FOR FORECASTING THE NUMBER OF CALLS TO THE CALL CENTER MEASURED ON THE TEST DATASET

Algorithm	Correlation coefficient	Mean absolute error	Root mean squared error	Relative absolute error (%)	Root relative squared error (%)
Random Forest	0.9222	2.0517	5.0113	20.3268	39.0217
Bagging	0.9198	1.9663	5.0907	19.4807	39.6404

It is then possible to test the model on the test dataset. Fig. 1 shows the actual values of the number of calls per hour for the third year, as well as the forecast values of the number of calls per hour obtained using models based on the RandomForest and Bagging algorithms. As can be observed, certain seasonal variations of actual data are noticeable, and forecasting models closely follow these variations.

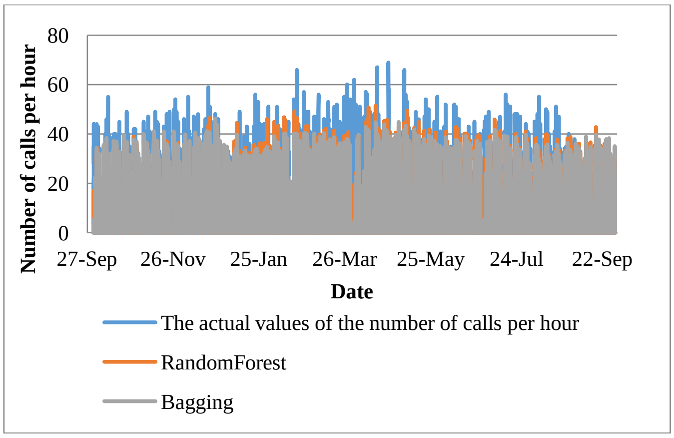

Fig. 1. The actual and forecasted number of calls per hour for each hour of the day and every day of the year

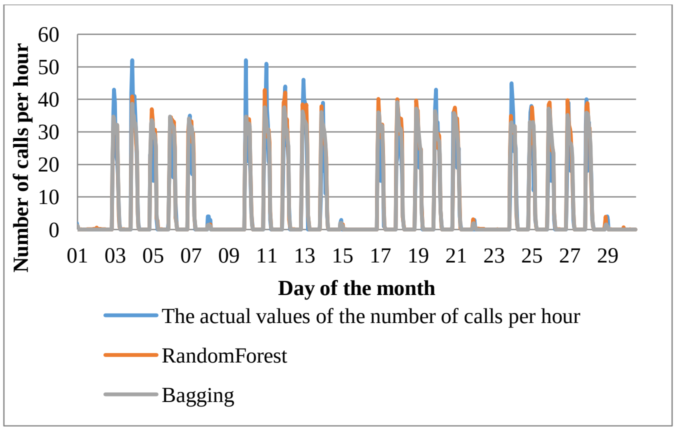

Fig. 2. The actual and forecasted number of calls per hour for each hour of the day and every day of the month

For better clarity, Fig. 2 shows the actual and forecast values of the number of calls per hour for an arbitrarily chosen month, within the third year, based on the same models as in the previous case. It can be noted that the number of calls per hour fluctuates greatly within one month, and forecasting models pretty much follow these fluctuations.

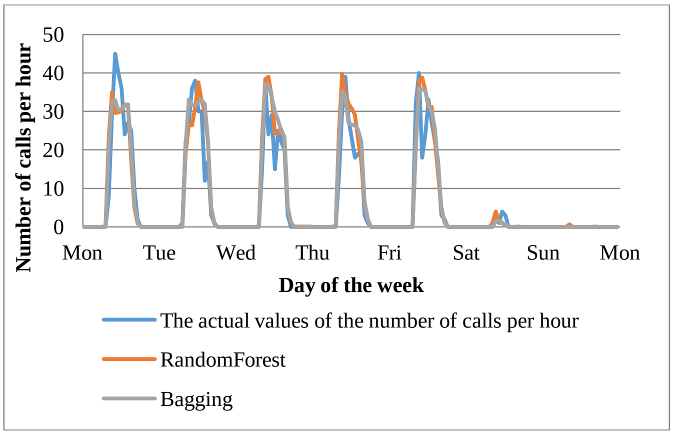

Fig. 3. The actual and forecasted number of calls per hour for each hour of the day and every day of the week

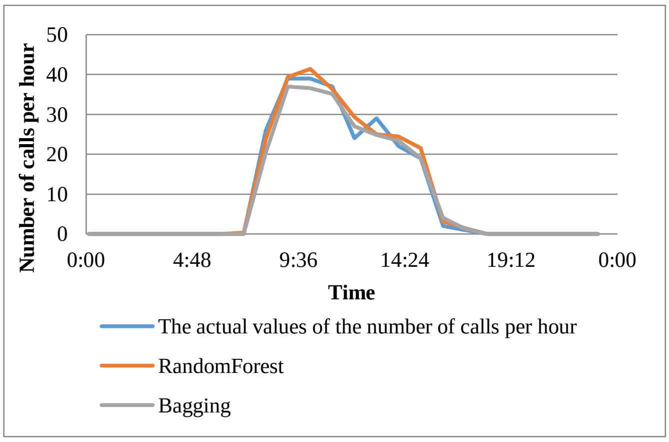

Fig. 4. The actual and forecasted number of calls per hour for each hour of one day

Fig. 3 shows the actual and forecast values of the number of calls per hour over one arbitrarily chosen week, within the third year, based on the same models as in the previous cases. Significant fluctuations in the number of calls per hour can also be observed here. As expected, the number of calls is significantly higher on working days than on weekends. The forecast values follow this trend very well.

Fig. 4 shows the actual and forecast values of the number of calls per hour over one arbitrarily chosen day, within the third year, based on the same models as in the previous cases. During daylight hours the number of calls per hour is much higher than during the night. And in this case, forecasting models are largely following this trend.

IV. Conclusion

This paper presents the application of two models for forecasting the number of calls per hour to the call center, based on machine learning. For this, two algorithms provided by WEKA software were used, namely RandomForest and Bagging algorithms. These algorithms were chosen because they showed the highest correlation coefficients in the cross-validation process. In this case, a few more algorithms showed very good performance, so it is possible to extend this study using other algorithms.

The most important and demanding part of the job with such forecasts is the preparation of an input dataset. The input data should be well grouped and described with the right attributes. The success of creating an effective forecasting model largely depends on the combination of attributes that describe each data. In this paper, all attributes are numeric and refer to the time of user calls to the call center. This set of attributes was chosen because significant variations were observed in the number of incoming calls on a yearly, monthly, weekly and daily basis.

For some future research, it is possible to include information about location from which calls were made or call type information, depending on whether the calls are from a mobile or landline telephone network, in the forecast. Since the forecasting methods applied have shown very good performance in such a dynamic time series, similar methods can be used for many other cases where traffic is extremely dynamic and not easy to predict. Such dynamic performance has IP traffic, which is increasingly used in the modern world. This may be one of the direction of our future research.

Acknowledgement

This paper has been partially supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia project under No. 32025.

References

[1] M. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, “The WEKA Data Mining Software: An Update”, ACM SIGKDD Explorations Newsletter, vol. 11, pp. 10-18, 2009.

[2] M. I. Jordan, T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects”, American Association for the Advancement of Science, vol. 349, no. 6245, pp. 255-260, 2015.

[3] A. K. Jain, M. N. Murty, P. J. Flynn, “Data clustering: a review”, *ACM Comput Surveys*, vol. 31, no. 3, pp. 264-232, 1999.

[4] S. B. Kotsianitis, “Supervised Machine Learning: A Review of Classification Techniques”, IOS Press, Emerging Artificial Intelligence Applications in Computer Engineering, vol. 160, pp. 249-268, 2007.

[5] S. Janković, K. Kukić, A. Uzelac, V. Maraš, "Predikcija saobraćaja u lokalnoj računarskoj mreži primenom nadgledanog mašinskog učenja", PosTel, Conference Proceedings, pp.181-192, Belgrade, Serbia, 2019.

[6] I. H. Witten, E. Frank, M. A. Hall, C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques, Fourth Edition*, Burlington, USA, Morgan Kaufmann, 2017.