

Comparative Analysis of Conversion Series Forecasting in E-commerce Tasks

Lyudmyla Kirichenko^(✉), Tamara Radivilova, and Illya Zinkevich

Kharkiv University of Radioelectronics, Kharkiv, 61166, Ukraine
Lyudmyla.kirichenko@nuze.ua, tamara.radivilova@gmail.com

Abstract. The characteristic features of time series conversion, which arise in the tasks of e-commerce are described. It is shown that these series are weakly correlated, which does not allow to use traditional methods for their prediction. Forecasting of the series is performed by methods of exponential smoothing, neural network and decision tree using data from an online store. A comparative analysis of the results is carried out. The advantages and disadvantages of each method are considered.

Keywords: Time series · Conversion rate · Machine learning · Forecasting · Exponential smoothing · Decision tree · Long-term memory neural network

1 Introduction

Time series describe a wide range of phenomena, for example, they are the stock prices, solar activity, the overall incidence rate and much more. Economic indicators can also be considered as time series and you can try to find not visible at first glance laws, hidden periodicity, to predict the moments when peaks appear, etc. At the moment is urgent time-series analysis in the field of e-commerce. E-commerce is in process of development, which is facilitated by new technologies, services and tactical tools [1]. For successful sale in online stores, web analytics is used, which allows to work on optimization, increase conversion and attendance of the electronic store.

To “survive” and stand out among the many online stores, it is important to understand the user’s behavior from the moment of the first arrival on the site: to track his movements, to know what products he looked at, put in the basket, where he clicked, what saw, the time he left, how and when he returned. Web-analytics will help in this, which involves ongoing collection, analysis and interpretation of data about visitors, work with basic metrics. Careful analysis of the online store and user behavior is a necessary stage of business development.

Quality web analytics of online store always begins with the visitor’s way that he passed before making a purchase. The order processing consists of the following steps: (1) product search; (2) add item to shopping cart; (3) go to the checkout page; (4) fill out and submitting the form; (5) go to the page of the order, payment. The main task of analytics is to periodically find and fix the weak points in this chain. At each stage, the

user can stop without having made a purchase. Each of these stages is represented in the form of a time series.

It is impossible to work on optimization, increase of conversion and site attendance without web analytics. By using key performance indicators, it can be significantly improved profit site. These indicators show how quickly and efficiently the business grows. One of the advantages of running an online store is the transparency of key performance indicators tracking and the ability to optimize processes for business growth [2]. The competition in the field of e-commerce is so great that an online store that does not use analytics and metrics will not last long. Here are the main success indicators that any online store should measure:

1. Attendance of the site - the number of users visiting the site, which are measured in terms of daily audience, weekly and monthly. This will allow to evaluate the incidence and bursts of site traffic and identify their causes.
2. Views commodity page – which pages visitors view on the site often and which less. Analyzing the attendance of commodity pages, it is possible to understand the customers' shopping preferences and the way of their interacting with the site.
3. The average time on the site and the average number of pages viewed – if these indicators are low, it is worth assessing the quality of traffic on site.
4. Exit pages - by analyzing the exit points of the site visitors (registration, shopping cart, ordering), can better understand the reasons for the low conversion and optimize the site so that users stay on the site and complete purchases.
5. Channels to attract visitors - you need to track not just a source of attraction of visitors, and their impact.
6. The overall conversion rate of an online store is the number of visitors who made a purchase.
7. Indicator of return of visitors – the number of not only new, but also returned visitors to the online store are analyzed. These will allow evaluate how the site is interesting for the target audience.
8. Profit from the buyer – profit minus costs. This indicator provides an understanding of how successful online store.
9. The failure rate is the number of orders that have been started, but not completed.
10. Number of products in the order – the number of products per order.
11. Average order value - total sales/orders quantity.

A store that is already selling on the Internet, often to increase profits, it is sufficient to pay attention to only a few metrics, one of which is the conversion. Conversion is the most important parameter that characterizes the effectiveness of the website promotion process. Conversion is the ratio of the number of users who made purchase of a product or service on your site to the number of users who came to your site for an advertising link, ad, or banner. For example, if a site was visited by 100 people, but only 2 people bought the product, then the conversion rate is equal to 2%. The increase of conversion percent depends on many factors: from the design of the page to its functionality. Monitoring conversion allows to understand in time that the e-shop needs to be improved. Conversion is the main metric in the web analytics of all commercial sites [3].

Analysis and forecasting of time series of daily value conversion percent plays crucial value for optimizing the efficiency of the online business [4]. However, it should be noted that almost all of the classical methods of time series forecasting based on the calculation of the correlation between the time series values [5]. In the case of weakly correlated time series, and also in the case when the time series has sparse zero values structure, which is typical for many electronic sales sites, these methods do not fit or have a large error.

Neural network approach has been widely used to solve forecasting problems. Neural networks allow you to model complex relationships between data as result of learning by examples. However, the prediction of time series using neural networks has its drawbacks. First, for training the majority of neural networks, time series of a large length are required. Secondly, the result essentially depends on the choice of the architecture of the network, as well as the input and output data. Third, neural networks require preliminary data preparation, or preprocessing. Preprocessing is one of the key elements of forecasting: the quality of the forecast of a neural network can depend crucially on the form in which information is presented for its learning. The general goal of preprocessing is to increase the information content of inputs and outputs. An overview of the methods for selecting input variables and preprocessing is contained in [6].

Recently, for the analysis of the regularities of the time series, the methods of machine learning [7, 8] have been increasingly used to detect various patterns in the time series. In this case, logical methods are of particular value in the detection of such patterns. These methods allow us to find logical if-then rules. They are suitable for analyzing and predicting both numerical and symbolic sequences, and their results have a transparent interpretation.

The goal of the presented work is to carry out a comparative analysis of weakly correlated time series forecasting, based on classical prediction methods and machine learning ones such as neural networks and decision trees, using the data of a real online store.

2 Input Data

Input data in the work were daily data from the online sales site, which included the number of clicks on the site from social networks, the number of sales and the corresponding conversion rate. In addition, there was information about which language the customer used, from which country the order was and other data [9].

Figure 1(at the top) shows typical time series of conversion rate. Series of conversion rate are characterized by zero values, which significantly complicates forecasting the next day. The correlation function of the rate series is shown on Fig. 1(at the bottom). Obviously, there is no correlation between the time series values.

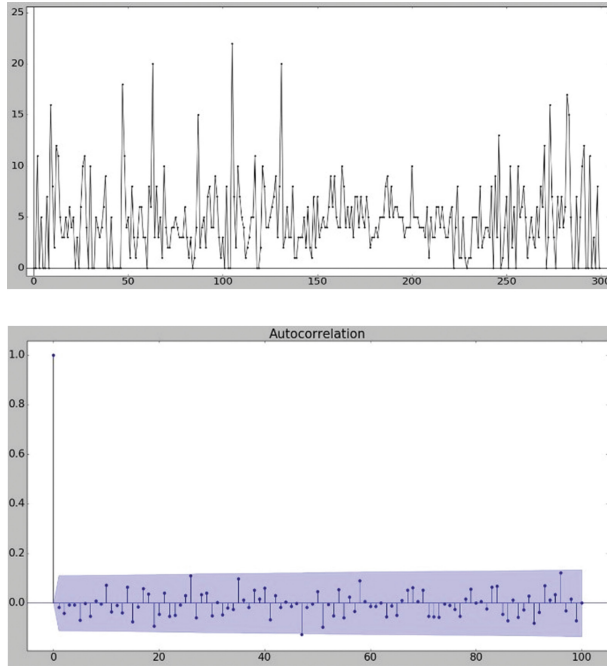


Fig. 1. Time series of conversion rate and the correlation function

Figure 2 shows the histogram of the distribution density of typical conversion rate series. It is easy to see that the percent from 0 to 2 is the highest, then there is a more even distribution, but for each series of conversion rate there are bursts that are most difficult to forecast.

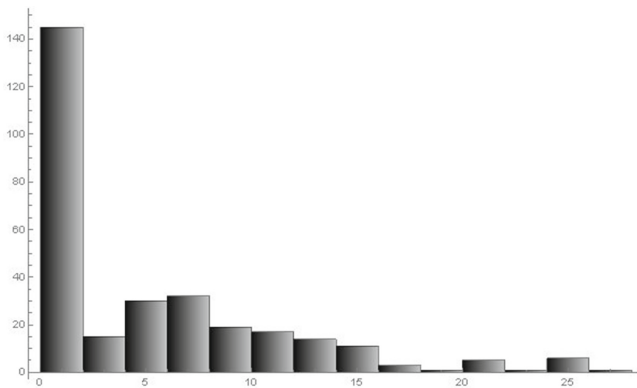


Fig. 2. Distribution density of typical conversion rate series

3 Forecasting Methods

E-commerce is constantly evolving, it is facilitated by new technologies, services and tactical tools. Suppliers, range of buyers, assortment of goods change regularly, that leads to a rapid obsolescence of information. Therefore, forecasting methods that require time series of great length, such as, for example, autoregressive and moving average models, work poorly [10].

Methods of Exponential Smoothing. The basis for exponential smoothing is the idea of a constant revision of the forecast values as the actual ones arrive. The model of exponential smoothing assigns exponentially decreasing weights to observations as they become outdated [5, 11]. Thus, the latest available observations have greater influence on the forecast value than older observations.

The model of exponential smoothing has the form:

$$Z(t) = S(t) + \varepsilon_t, S(t) = \alpha \cdot Z(t - 1) + (1 - \alpha) \cdot S(t - 1), \quad (1)$$

where α is smoothing factor; $0 < \alpha < 1$; $Z(t)$ is projected time series; $S(t)$ is smoothed time series; initial conditions are defined as $S(1) = Z(0)$. In this model, each subsequent smoothed value $S(t)$ is the weighted average between the previous value of the time series $Z(t-1)$ and the previous smoothed value $S(t-1)$.

The value α is determined by how much the current series value should affect the next value. The closer α is to unity, the stronger the forecast takes into account the value in the previous step. To find the optimal value α , it is required to minimize the mean error of the forecast. When the value α is automatically selected, all forecasts are calculated for α that change with a given step, the mean error is calculated, and the value α at which the error has the smallest value is selected.

3.1 Decision Tree Method

Methods of machine learning are an extremely broad and dynamically developing field of research using a lot of theoretical and practical methods. One of these methods is the decision tree method [7, 12, 13]. The decision tree is a decision support tool used in statistics and data analysis for predictive models.

In intellectual data analysis, decision trees can be used as mathematical and computational methods to help describe, classify and summarize a set of data that can be written as follows: $(x, Y) = (x_1, x_2, x_3, \dots, x_k, Y)$. The dependent variable Y is the objective variable that needs to be analyzed, forecast and generalized. A vector x consists of input variables x_1, x_2, x_3 , etc., which are used to perform the task.

The decision tree method for classification or prediction task is the process of dividing the original data into groups until homogeneous (pure) subsets are obtained. The set of rules, due to which there is such division, allows to make a forecast obtained as a result of evaluating some input features x_1, x_2, x_3 for new data.

The decision tree is a model that represents a set of rules for decision-making. Graphically, it can be represented in tree structure form, where decision-making

moments correspond to the decision nodes. The data to be classified are at the root of tree. In nodes, depending on decision made, a branching process occurs. Terminal nodes are called leaf nodes. Each leaf is final result of consistent decision-making and represents value of objective variable, which was modified during movement from root to leaf. Each internal node corresponds to one of input variables. Depending on decision made at nodes, the process eventually stops in one of leaves, where a variable of response is assigned a particular value.

The algorithm of learning (forming the tree) operates according to the principle of recursive partitioning. The partitioning of data set (i.e., splitting into disjoint subsets) is performed on the basis of the most suitable for this feature. A corresponding decision node is created in tree, and process continues recursively to the stopping criterion.

There are various numerical algorithms for constructing decision trees. One of the most famous is the algorithm called C5.0, developed by the programmer J.R. Quinlan. In fact, the C5.0 is the standard for construction of decision trees. This program is implemented on a commercial basis, but version built into the Python (and some other packages) is available for free.

The algorithm implements the principle of recursive partitioning. The algorithm starts with empty tree and complete data set. In nodes, starting from the root node, feature is selected whose value is used to divide all data into two classes. After the first iteration, the tree appears with one node dividing the data set into two subsets. After that, this process can be performed repeatedly, with respect to each of subsets for creating subtrees. To separate data, we use conditions of form: $\{x < a\}$, $\{x > a\}$, where x is feature, and a is some fixed number. Such partitions are called “axis-parallel splits”. Essentially, with each condition check, the data samples are sorted in such way that each data element is determined to correspond only to one branch. Decision criteria divide the original data set into disjoint subsets. The recursion terminates if subset in node has same values of objective variable, so it does not add values to forecasts.

To create a decision tree, its needed to determine the features by which the partition will be performed. In the case of the classification of data samples, these values may be the sampling values. From the set of attributes for the partition, it is required to choose those that would allow to obtain as homogeneous (pure) sets as possible. Algorithm C5.0 uses as impurity measure of entropy concept, which is a measure of data disorder.

Using the entropy as measure of impurity sets that are result of partitioning, algorithm can select feature on which partitioning will give the purest set (i.e., the set with lowest entropy). These calculations are called “information gain”. The feature is determined by the search method. For each feature, the value of information gain is calculated as difference in entropy of sets before and after partitioning.

The higher information gain for selected feature, better this feature is suitable for partitioning, since such partition will ensure that the most pure set is obtained. If, for the selected characteristic, the value of information gain is close to zero, it means that the partition by this feature is unpromising, since it does not lead to entropy decrease. On the other hand, the maximum possible value of information gain is equal to the value of entropy before the partition. This means that the entropy after partition will be zero, i.e. resulting sets will be completely pure.

The main advantages of the C5.0 algorithm for forecasting tasks: it is universal, it solves well the problems of classification and forecasting from different areas; to construct a decision tree, it selects from set of features only those that strongly influence the result; it requires a relatively small amount of training sample. One of the significant advantages of the C5.0 algorithm is its ability to post-pruning of the built decision tree, that is, cutting off those nodes and branches that have little impact on the forecast results. The disadvantages are that the algorithm “gravitates” to split based on a large number of levels; inaccuracies in classification can arise from the fact that only “parallel-axis axes” splits are used; decision trees sometimes turn out to be very large.

Neural Networks. It can be said that any neural network (NN) acts as follows: iteration after iteration, it deforms the vector of input data in this way that as a result of deformation, the input data fall into the zones where we expect to see them at the output. In ordinary neural networks, each individual sample is processed without taking into account the influence of past information on current result. To solve this problem, recurrent neural networks (RNN) were developed in the 1980s. These are networks that contain feedback and allow to take into account the previous iterations. A recurrent network can be viewed as several copies of the same network, each of which transmits information of a subsequent copy. RNN resembles a chain, and its architecture is well suited for working with sequences, lists and time series.

The scheme of the RNN operation looks like this: there is an input layer of neurons that is projected onto a hidden layer (one or several), the outputs of the hidden layer are transferred to output layer, and also copied to context layer, which at the next iteration is perceived together with input layer and connected to hidden layer. Accordingly, cycle is produced: a hidden layer - a context layer - a hidden layer. In progress new samples will come in RNN, they will change the context and context circulating within the network, will retain this information, which will affect the current classification. Over the past few years, RNN has successfully applied to a lot of tasks: speech recognition, language modeling, translation, image recognition, etc. [14]. But the main disadvantage of classic RNN is to reduce the effect of samples with increasing time delay. As a rule, the maximum impact on response of RNN samples that were at previous iteration, two iterations back, etc. have and the further, the less this effect decreases. While quite often there are situations when information important for correct forecasting is not on the nearest samples, but on 10-20-30 iterations back.

Recently, the architecture of RNN which are called long short-term memory neural networks (LSTM) has become popular. This is a special kind of recurrent neural networks, which are capable of learning long-term dependencies. LSTM are specifically designed to avoid the problem of long-term dependencies. Remembering information for a long period of time is practically their default behavior [6, 15, 16].

A method that some elements in the context in the previous iterations provide a greater influence on the result, while other elements have a smaller effect was suggested. In LSTM, it is proposed to extend the classical RNN schema with notion gate, which is a memory gate and forget gate, and which determines how likely the given sample should be forgotten or remembered for next iteration. The previous samples affect saving or deleting of sample. If they indicate that this information is important for future

classification, more importance will be given. If the current information plays a weak role for forecasting at subsequent iterations, impact will decrease.

Currently LSTM work incredibly well on a wide variety of tasks and are widely used. Many impressive results of work of RNN were achieved precisely on the basis of LSTM architecture [16].

The Forecast Errors. To obtain quantitative characteristics of the comparative analysis of the models, the following characteristics of forecast errors were chosen [5, 11, 17]. The Mean Absolute Deviation (MAD) measures the accuracy of the forecast by averaging the values of the forecast errors. Using MAD is most useful when the analyst needs to measure the forecast error in the same units as the original series. This error is calculated as follows:

$$MAD = \frac{1}{n} \sum_{t=1}^n |X(t) - \hat{X}(t)|. \quad (2)$$

The average deviation (Mean Deviation, MD) allows to see how the forecast value is overvalued or undervalued on average:

$$MD = \frac{1}{n} \sum_{t=1}^n X(t) - \hat{X}(t). \quad (3)$$

Mean squared error (MSE) is another way of estimating the forecasting method. Since each deviation value is squared, this method emphasizes large forecast errors. The MSE error is calculated as follows:

$$MSE = \frac{1}{n} \sum_{t=1}^n (X(t) - \hat{X}(t))^2. \quad (4)$$

The Mean Absolute Percentage Error (MAPE) is calculated by finding the absolute error at each time and dividing it by the actual observed value, with subsequent averaging of the obtained absolute percent errors. This error is calculated as follows:

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|X(t) - \hat{X}(t)|}{X(t)}. \quad (5)$$

This approach is useful when the size or value of the predicted value is important for estimating the accuracy of the forecast. MAPE emphasizes how large the forecast errors are in comparison with the actual values of the series. This approach is useful when the size or value of the predicted value is important for estimating the accuracy of the forecast. MAPE emphasizes how large the forecast errors are in comparison with the actual values of the series.

4 Software Implementation of Machine Learning Methods

The methods and algorithms of Data Mining and machine learning must be implemented in a certain programming language, in a certain environment, calculated on a certain type of computer elements, etc. There are many tools available today to implement Data Mining and Machine Learning algorithms [18].

One of the most widely used programming languages for solving application problems is the Python language. Python is a general purpose programming language, which means that people have built modules to create websites, interact with a variety of databases, and manage users. Python uses a large number of people and organizations around the world, so it develops and is well documented; it is cross-platform and you can use it free [12].

This language has several advantages. It is quite easy to learn and, as a rule, is a language that needs a low entry level. A person who has basic knowledge in the theory of algorithms and mathematics can simply master the basic functionality, methods and syntax to solve applied problems. Python has implemented a large number of libraries, which provide most of the available algorithms in a convenient way.

In general, for machine learning there are several basic Python libraries, which have quite a big advantage compared to libraries of other programming languages. The main one is: very detailed and qualitative documentation. Most of these libraries use the NumPy library [19]. This is a library that allows you to quickly and effectively work with numeric data matrices, tables of numbers in different formats, to carry out a large number of typical operations that are required in the process of solving applied machine learning tasks.

One of the great documentation libraries that implements most of the typical machine learning methods is scikit-learn [20]. Dozens of algorithms are implemented in this library for clustering, regression, classification, reference vector method, linear and logistic regression, and dozens of other algorithms. Each of the available algorithms has a large number of parameters that can be customized to your task.

One very convenient language libraries in Python, which help to work with lots of tabular data (often training and testing sample look like .csv-table with hundreds of thousands and millions of rows and columns parameters) is a Pandas library [21]. It allows to download data very quickly, preprocess it (to prepare it in a suitable format), to send in a convenient form for processing by our algorithm, which we, for example, have chosen from the Scikit-learn library.

Currently, due to the popularity of neural networks in Python, there are many libraries at quite different levels of abstraction (from low to high-abstraction architecture descriptions) allow construction of various neural networks.

One of the most commonly used libraries for low-level operations for the implementation of neural network algorithms is the Theano Library [22]. It implements complex matrix cartoons, rapid methods of convolution with multiplication, sampling, regression methods, and all backend-logic of neural networks.

One key bonus of libraries is that in addition to CPU-realization (i.e., implementation of algorithm's operation on processor), Theano or TensorFlow libraries (by Google) are open source (you can see and add modules that you need).

It should be noted that between the graphics card, the Python language and the library written in this language there is one more layer - CUDA - a set of libraries, implemented by NVidia, which allow effectively and quickly perform calculations on its graphics cards.

To work with web resources, the Scrapy library is used [23]. Scrapy is an application framework for crawling web sites and extracting structured data which can be used for a wide range of useful applications, like data mining, information processing or historical archival. Scrapy provides a lot of powerful features for making scraping easy and efficient, such as: built-in support for selecting and extracting data from HTML/XML sources using extended CSS selectors and XPath expressions, with helper methods to extract using regular expressions; for generating feed exports in multiple formats (JSON, CSV, XML) and storing them in multiple backends (FTP, S3, local filesystem); an interactive shell console (IPython aware) for trying out the CSS and XPath expressions to scrape data, very useful when writing or debugging your spiders.

5 Research Results

The analysis of the conversion series for compliance with ARMA models was performed. Figure 3 shows the typical values of the Akaike information criterion (AIC), which is used exclusively for the selection of several statistical models for one set of data [5, 9]. The values of the criterion indicate that the use of ARMA models in this case is not appropriate.

	Candidate	AIC
1	MAProcess[0]	1411.31
2	MAProcess[1]	1412.81
3	ARProcess[1]	1412.85
4	ARMAProcess[1, 1]	1414.7

Fig. 3. Values of the Akaike information criterion

To carry out the forecasting, time series were divided into two parts, where the first one was used to train the model, and the second one was applied to assess its plausibility. The models were trained on the S last values of time series.

Checking the models for forecasting m values was carried out in the following way: take the window of last S values from the first part series and will do the forecast one value ahead; then will move window one value forward, including the forecast of new value in the window, and will again do the forecast, and so m times.

Figures 4, 5 and 6 presents the results of the forecasts of each model for 7 values ahead or $S = 20$ and $m = 1$. The solid line shows the actual values. Figure 4 shows the values obtained by the method of exponential smoothing. On Fig. 5 presents ones based on decision tree. On Fig. 6 values obtained with the help of the LSTM neural network are shown.

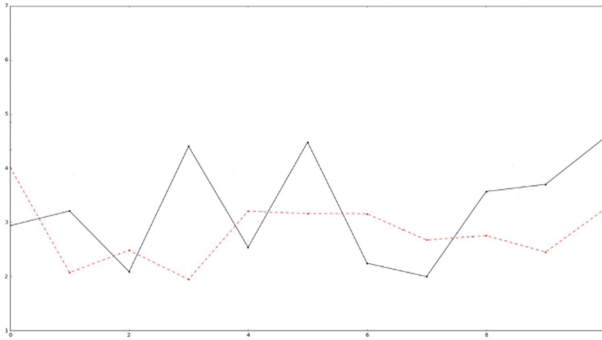


Fig. 4. Forecasted values obtained by the method of exponential smoothing

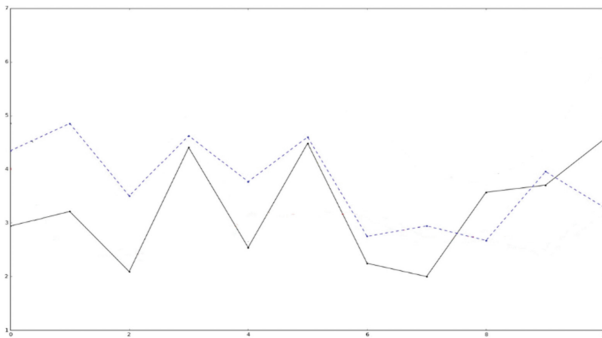


Fig. 5. Forecasted values based on decision tree

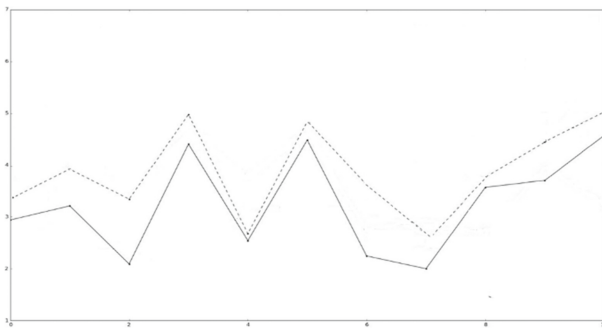


Fig. 6. Forecasted values obtained with the help of the LSTM

The predicted values for $S = 20$ and $m = 1$ (this choice of parameters is determined by the requirements of the online store) were computed for 100 values of the daily data conversion rate and corresponding values of clicks and sales number and other data. The results of calculations typical for most series are given in Table 1.

Table 1. Forecast errors

	ES	DT	LSTM
MAD	7.65	6.08	1.2
MA	-1.99	1.32	0.91
MSE	64	53	22
MAPE	0.49	0.38	0.07

As a result of the analysis of the forecasts of different values of S and m , the following was established. The method of exponential smoothing, in spite of its simplicity and non-exactingness in the amount of data, has in most cases comparatively small prediction errors. But at the same time, with the use of this method, some predicted values are significantly removed from real ones.

The decision tree method has proved to be inconvenient in the choice of parameters and has errors comparable with errors of exponential smoothing, but without strongly remote forecast values. The LSTM neural network, which has a more complex structure and needs to be preliminarily trained on a rather large time series, has shown good results, as well as in the overall forecast error, and in the remoteness of forecasts from real time series values.

6 Conclusion

The results of a study of methods for predicting weakly correlated time series typical of e-commerce conversion series have shown that exponential smoothing is the simplest, fastest and most convenient to set up predictive method, but in the cases of complex or long-term dependencies, it does not apply. The decision tree method is fast in learning, not difficult to understand, but inconvenient in the choice of parameters and does not work well when learning on data that have many characteristics. The LSTM neural network is a cumbersome, long learning, requires a lot of parameters that need to be selected, but has a very good performance in forecasting and order of magnitude smaller errors.

References

1. LPgenerator - Professional Landing Page is a platform to increase sales of your business. <http://lpgenerator.ru/blog/2015/07/02/kakoj-dolzha-byt-veb-analitika-internet-magazina>. Accessed 10 July 2017
2. Stillwagon, A.: 14 Key Performance Indicators (KPIs) to Measure Customer Service. <https://smallbiztrends.com/2015/03/how-to-measure-customer-service.html>. Accessed 10 July 2017
3. Conversion Probability Forecast. https://www.searchengines.ru/prognoz_veroyat.html. Accessed 10 July 2017
4. Wei, D., Geng, P., Ying, L., Shuaipeng, L.: A prediction study on e-commerce sales based on structure time series model and web search data. In: 26th Chinese Control and Decision Conference, Changsha, China, pp. 1–4. IEEE (2014)

5. Hanke, J.E., Wichern, D.: *Business Forecasting*, 9th edn. Pearson Prentice Hall, Upper Saddle River (2008)
6. Guyon, I.J., Elisseeff, A.: An introduction to variable and feature selection. *J. Mach. Learn. Res.* **3**, 1157–1182 (2003)
7. Ian, H.W., Frank, E., Hall, M.A., Pal, C.J.: *Data Mining: Practical Machine Learning Tools and Techniques*, 4th edn. Morgan Kaufmann, Elsevier (2011)
8. Ismail, M., Mansur Ibrahim, M., Mahmoud Sanusi, Z., Nat, M.: Data Mining in electronic commerce: benefits and challenges. *Int. J. Commun. Netw. Syst. Sci.* **8**, 501–509 (2015)
9. Kirichenko, L., Radivilova, T., Zinkevich, I.: Forecasting weakly correlated time series in tasks of electronic commerce. In: *12th International Conference Computer Sciences and Information Technologies*, Lviv, Ukraine. IEEE (2017)
10. Qiang, X., Rui-Chun, H., Hui, L.: Data mining research on time series of e-commerce transaction. *Int. J. u- and e- Serv. Sci. Technol.* **7**(1), 9–18 (2014)
11. Hyndman, R., Koehler, A.B., Keith Ord, J., Snyder, R.D.: *Forecasting with Exponential Smoothing*. Springer, Heidelberg (2008)
12. Cielen, D., Meysman, A., Ali, M.: *Introducing data science: Big Data, machine learning, and more, using Python tools*. Manning Publications, Greenwich (2016)
13. Lantz, B.: *Machine Learning with R*, 2nd edn. Packt Publishing, Birmingham (2015)
14. The Unreasonable Effectiveness of Recurrent Neural Networks. <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>. Accessed 10 July 2017
15. Understanding LSTM Networks. <http://colah.github.io/posts/2015-08-Understanding-LSTMs>. Accessed 10 July 2017
16. LSTM - network of long short-term memory. <https://habrahabr.ru/company/wunderfund/blog/331310/>. Accessed 10 July 2017
17. White, D.S., Ariguzo, G.: A time-series analysis of U.S. e-commerce sales. *Rev. Bus. Res.* **11**(4), 134–140 (2011)
18. Hongjiu, G.: Data mining in the application of e-commerce website. In: Du, Z. (ed.) *Intelligence Computation and Evolutionary Computation. Advances in Intelligent Systems and Computing*, vol. 180, pp. 493–497. Springer, Heidelberg (2013)
19. Numpy and Scipy Documentation. <https://docs.scipy.org/doc/>. Accessed 10 July 2017
20. Documentation of scikit-learn 0.18. <http://scikit-learn.org/stable/documentation.html>. Accessed 10 July 2017
21. pandas: powerful Python data analysis toolkit. <http://pandas.pydata.org/pandas-docs/stable/>. Accessed 10 July 2017
22. Theano Python library. <http://deeplearning.net/software/theano/>. Accessed 10 July 2017
23. Scrapy 1.4 documentation. <https://docs.scrapy.org/en/latest/>. Accessed 10 July 2017