Ministry of Science and Higher education

of the Russian Federation

ITMO University

Faculty of Digital Transformations

Educational program 01.04.02. Applied mathematics and informatics

Subject area (major) Big Data and Machine learning

REPORT

Modeling and forecasting sales in retail

Student: Victor Voronin, J41324c

Supervisor: Gladilin P.E.

Date: 01.03.2021

St. Petersburg

2021

TABLE OF CONTENT

[Introduction 3](#_Toc65279693)

[Literature review 3](#_Toc65279694)

[Models, Algorithms and Datasets 4](#_Toc65279695)

[ARIMA 4](#_Toc65279696)

[Error, Trend, Seasonality model (ETS) 5](#_Toc65279697)

[Long-short term memory (LSTM) 6](#_Toc65279698)

[Gated Recurrent unit (GRU) 7](#_Toc65279699)

[XGBoost 7](#_Toc65279700)

[Evaluation measurements 8](#_Toc65279701)

[Experimental research 8](#_Toc65279702)

[ETS 9](#_Toc65279703)

[ARIMA 9](#_Toc65279704)

[LSTM 10](#_Toc65279705)

[GPU 11](#_Toc65279706)

[XGBoost 11](#_Toc65279707)

[Conclusions 12](#_Toc65279708)

[References 13](#_Toc65279709)

# Introduction

Sales forecasting is one of the most important issues that is beyond all strategic and planning decisions in any retail business. The importance of accurate sales forecasts to efficient inventory management at both disaggregate and aggregate levels has long been recognized [1]. Poor forecasts usually lead to either too much or too little inventory directly affecting the profitability and the competitive position of the company. At the organizational level, sales forecasting is very important to any retail business as its outcome is used by many functions in the organization: finance and accounting departments are able to project costs, profit levels and capital needs; sales department is able to get a good knowledge of the sales volume of each product; purchasing department is able to plan short- and long-term purchases; marketing department is able to plan its actions and assess the impact of different marketing strategies on sales volume; and finally logistics department is able to define specific logistic needs [2]. Accurate forecasts of sales have the potential to increase the profitability of retailers by improving the chain operations efficiency and minimizing wastes. Moreover, accurate forecasts of retail sales may improve portfolio investors’ ability to predict movements in the stock prices of retailing chains [3].

# Literature review

Aggregate retail sales time series are usually preferred because they contain both trend and seasonal patterns, providing a good testing ground for comparing forecasting methods, and because companies can benefit from more accurate forecasts. Retail sales time series often exhibit strong trend and seasonal variations presenting challenges in developing effective forecasting models. How to effectively model retail sales series and how to improve the quality of forecasts are still outstanding questions.

The authors of study [4] use ARIMA models to predict the number of newspapers sold for a real case study of a newspaper company in Surakarta. As newspapers are supplied with demand, the switch from printed newspapers to electronic news have caused that many newspapers are returned. The authors do mention that ARIMA is used for forecasting short term, making the result of the study invalid for long term modeling. The authors select an appropriate ARIMA model for the sales forecasting of newspapers. Using evaluation measures such as RMSE and MAE as well as mean absolute percentage error, the results of [4] shows that the best ARIMA model was with the parameters (1, 1, 0) without constant. Furthermore, the authors conclude that the suggested ARIMA model can be used in practice for short-term forecasting of the sales of the newspapers as well as to reduce the amount of newspapers that are not sold.

In the study [5] the authors test a LSTM network on real sales data. The model was built to experiment on 66 products with 45 weeks worth of data. The model predicts sales in week level, using four consecutive weeks to predict the sales of the fifth week. Also, normalization between [0, 1] is used as well as iterating the experiment ten times to get the average mean square error to calculate the performance. The results obtained in [5] show that only a fourth of the products have relatively low forecasting errors. However, the authors motivated that the limitation in form of lack of data, lack of long-term seasonality as only 45 weeks worth of data was accessible and using one LSTM network for all products, could have affected the results immensely. The authors argue that the network was not optimized with the respect to characteristics of the specific products as it was a generalized implementation. Nonetheless the authors conclude that the LSTM network still shows potential and is a domain that should be further looked at, considering the seasonality of products and data from a longer time period, among other things, when developing a new network.

Artificial neural networks (ANNs) along with time series models have been widely applied in the short-term forecasting [6,7] and they have been shown to be a promising tool for forecasting financial times series. The reason is that the ANN is a universal function approximator which is capable of mapping any linear or non-linear functions. A large number of successful applications have shown that neural networks can be a very useful tool for time-series modelling and forecasting[8]. Neural networks are particularly good at capturing complex nonlinear characteristics of time series [9].

XGBoost is used for binary classification in supervised machine learning system capable of matching internet devices to web cookies in [10]. XGBoost is used in [11] to improve the marketer’s ability to identify individual users as they switch between devices and show relevant content/recommendation to users wherever they go.

The impact of the system has been widely recognized in a number of machine learning and data mining challenges. Take the challenges hosted by the machine learning competition site Kaggle for example. Among the 29 challenge winning solutions 3 published at Kaggle’s blog during 2015, 17 solutions used XGBoost.[12]

# Models, Algorithms and Datasets

## ARIMA

Exponential smoothing and Autoregressive Integrated Moving Average (ARIMA) models are the most widely used approaches to time series forecasting and provide complementary approaches to the problem. While exponential smoothing methods are based on a description of trend and seasonality in the data, ARIMA models aim to describe the autocorrelations in the data. The ARIMA framework to forecasting originally developed by Box et al. [13] involves an iterative three-stage process of model selection, parameter estimation and model checking. A statistical framework to exponential smoothing methods was developed based on state space models called ETS models [14].

ARIMA is composed of 3 parts namely Auto regressive (AR), Integrated (I) and Moving Average (MA):

1. AR is used to predict future values by extracting the influence of past values
2. MA is used to predict future values by extracting the influence of past errors
3. Integrated (I) is means differencing the series to make the time series stationary.

Th multiplicative seasonal ARIMA model, denoted as ARIMA(p, d, q) x (P, D, Q)m has the following form [15]

Where

And m is the seasonal frequency, B is the backward shift operator, d is the degree of ordinary differencing, and D is the degree of seasonal differencing and are the regular autoregressive and moving average polynomials of orders p and q, respectively, and are the seasonal autoregressive and moving average polynomials of order P and Q, respectively, where is a mean of process with variance .

The main task in ARIMA forecasting is selecting an appropriate model order, that is the values of p, q, P, Q, d and D. Usually the following steps are used to identify a tentative model [16,17]:

* Plot the time series, identify any unusual observations and choose the proper variance-stabilizing transformation. A series with nonconstant variance often needs a logarithm transformation (more generally a Box–Cox transformation may be applied

There are some modifications of ARIMA model:

* SARIMA:Seasonal ARIMA,
* SARIMAX:Seasonal ARIMA with exogenous variables

## Error, Trend, Seasonality model (ETS)

Exponential smoothing methods have been used with success to generate easily reliable forecasts for a wide range of time series since the 1950s [13]

The most common representation of these methods is the component form. Component form representations of exponential smoothing methods comprise a forecast equation and a smoothing equation for each of the components included in the method. The components that may be included are the level component, the trend component and the seasonal component. By considering all the combinations of the trend and seasonal components 15 exponential smoothing methods are possible. Each method is usually labeled by a pair of letters (T, S) defining the type of “Trend” and “Seasonal” components. The possibilities for each component are Trend={N, A, Ad, M, Md} and Seasonal={N, A, M}. For example (N, N) denotes the simple exponential smoothing method, (A, N) denotes Holt's linear method, (A , N) denotes the additive damped trend method, (A, A) denotes the additive Holt–Winters’ method and (A, M) denotes the multiplicative Holt–Winters’ method, to mention the most popular ones.

The additive Holt-Winters model consists of three smoothing equations and a forecasting equation as follow:

Where α, γ, and δ are the smoothing parameters, and St, Tt, and It are the smoothing equations known as level, trend, and seasonality. The information from the observed values (Xt) is projected through the last equation k steps ahead to obtain predictions (Xt (k)).

The trend and seasonal equations can be expressed in an additive or multiplicative form. The way the equations are combined allow a total of 30 possible combinations.

Taylor [18,19] introduced the double and triple seasonal Holt–Winters models (HWT). These models are characterized by capturing the information contained in the seasonal component, split into several seasonalities of different lengths, as well as including an adjustment of the forecast including the first autocorrelation error. García-Díaz and Trull generalized these models to adapt to an indeterminate number of seasonalities, proposing the multiple seasonal Holt–Winters models (nHWT) [20].

The Holt–Winters models are recursive, and thus an initialization value is needed to feed the model. The way to obtain the initial values must match the trend and seasonal method, as it would impact later on the accuracy of the model. The proposed methods for the initialization of the Holt–Winters models are separated into three groups. The first ones calculate the seeds for the level and trend separately. The last one implements each seasonality separately. The latter first obtains a seed, as the trend, and then obtains another one on a recurring basis, such as that of the level.

* Level methods:
  + Moving average from Holt [21]
  + First value [21]
  + First period’s average adapted from Winters [22]
* Trend methods
  + Newbold [26]
  + Taylor [18]
* Seasonality calculation method
  + Granger and Newbold [23]
  + Brockwell and Davis [15]. proposed another method, in which the calculation of seasonality is done by obtaining the weights of the series against the average values of the cyclic pattern. This method is certified by NIST
  + Winters [4]. The calculation of seasonality consists in calculating the series averages over the series without a trend.

## Long-short term memory (LSTM)

The LSTM networks implements a gated cell to store information, like computer memory. Unlike the aforementioned networks, the LSTM cells also learn when to allow reads and writes of information from previous time-steps. Hence the LSTM model solves the problem of a vanishing or exploding gradient and makes it possible for the network to correctly remember information far back in the sequence. Figure 2.2 presents the inner components of the LSTM-cell. One cell handles one time step worth of data and pass along chosen information to the next cell at the next time step, depending on the gates. Figure 2.3 depicts an example where three consecutive time-steps are used to predict the fourth. Notice the additional output Ct compared to 2.1, which only uses ht.

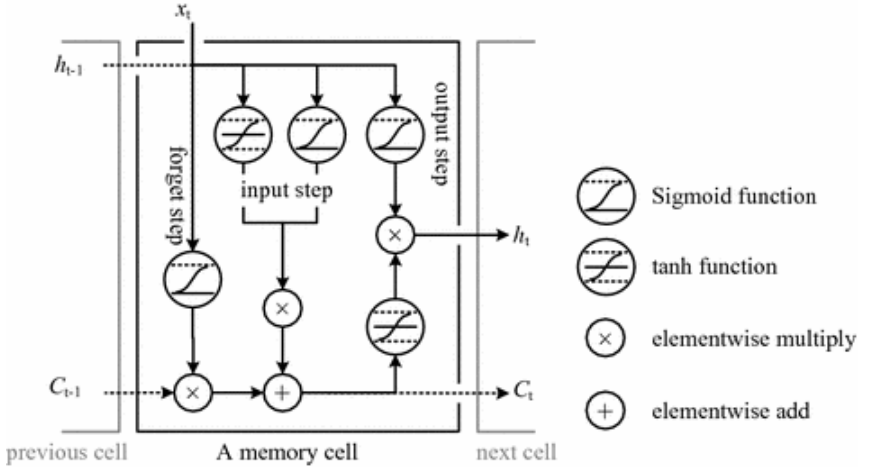


Figure An LSTM Cell and its components.[5]

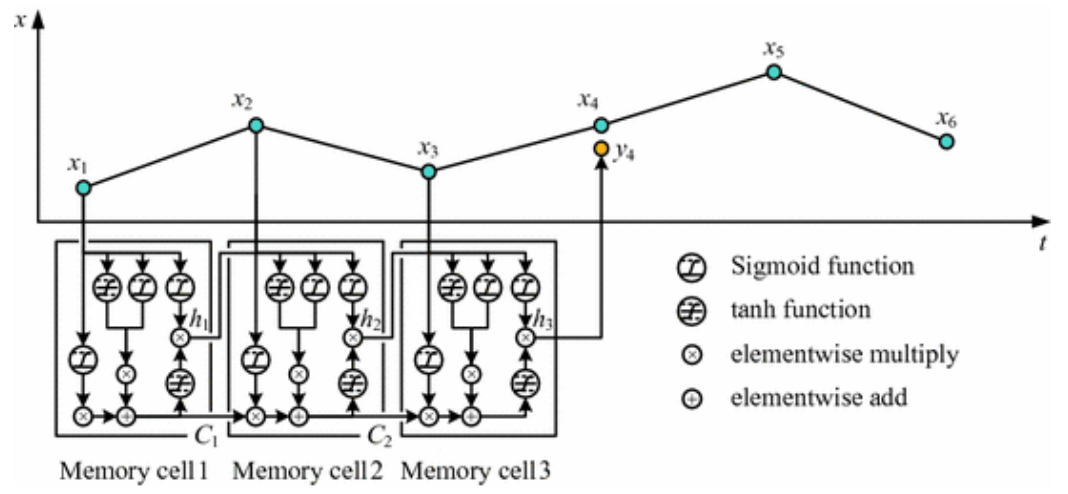


Figure Depicts an example of how time-series forecasting might look like using LSTM-cells [5]

## Gated Recurrent unit (GRU)

GRU is some variation of LSTM. GRU networks do not leverage the cell state and use the hidden state to transfer information. Unlike LSTM, GRU networks contain only three gates and do not maintain an internal cell state. The information that is stored in the internal cell state in an LSTM recurrent unit is incorporated into the hidden state of the GRU. This combined information is aggregated and transferred to the next GRU [24].

The first two gates reset gate and update gate, help solve the vanishing gradient problem of a standard RNN: these gates are two vectors that decide what information should be passed to the output [24]. The unique characteristics about them is that they can be trained to keep information from long ago, without removing it through time, or delete information that is irrelevant to the prediction.

* Update gate – The update gate helps the model to determine how much of the past information (from previous time steps) needs to be passed along to the future. That is really powerful because the model can decide to copy all the information from the past and eliminate the risk of the vanishing gradient problem.
* Reset gate ‒ the reset gate is another gate used to decide how much past information to forget.
* Current memory gate – This gate is incorporated into the reset gate and is used to initiate some nonlinearity into the input and develop the input zero-mean. Another motivation to incorporate it into the reset gate is to decrease the impact that previous information has on the current information that is being passed into the future.

## XGBoost

XGBoost is short for eXtreme Gradient Boosting. It is based on gradient boosting framework. Gradient boosting is the machine learning technique to deal with classification, regression and ranking problems. Xgboost has regularized model formalization to control overfitting, which boost its performance. It provides a good result for most of the datasets involving linearity and nonlinearity. It is efficient as it supports parallel computation on a single machine. Model of Xgboost is based on the concept of Gradient boosting which believes that single trees are not strong enough to give accurate prediction. Hence, the ensembles of decision trees are used, where trees are added in such a way that they optimizes the current error. [25].

For given dataset with n examples and m features a tree ensemble model uses K additive functions to predict the output [12]:

Where Is the space of regression trees. Here q represents the structure of each tree that maps an example to the corresponding lead index. T is number of leaves in the tree. Each fk corresponds to an independent tree structure q and leaf weights w. Unlike decision tree, each regression tree contains a continuous score on each of the leaf using wi to represent score on i-th leaf. It will use the decision rules in the trees to classify it into the leaves and calculate the final prediction by summing up the score in the corresponding leaves.

To learn set of functions used in the model, it needs to minimize the following regularized objective:

Where

Here l is a differentiable convex loss function that measures the difference between the prediction and the target. The second term Ω penalizes the complexity of the model (i.e., the regression tree functions). The additional regularization term helps to smooth the final learnt weights to avoid over-fitting. Intuitively, the regularized objective will tend to select a model employing simple and predictive functions.

## Evaluation measurements

To compare the performance models, two different evaluation measures were used: Mean absolute percentage error (MAPE) and root mean square error (RMSE) [26]. A lower value for both measures implies better accuracy. Using Ypred as the forecast value (the prediction), Ytrue as the actual value and n as the number of time steps, RMSE and MAPE can be defined as follows:

When comparing the performance of forecast methods on a single data set, the MAPE is interesting as it is easy to understand but the RMSE is more valuable as is more sensitive than other measures to the occasional large error (the squaring process gives disproportionate weight to very large errors).

# Experimental research

For our experiments we used follow datasets:

* Rossmann store sales dataset which provided historical data of 1115 Rossmann’s stores. The task is to forecast “Sales” column for test’s store for the last 90 days.

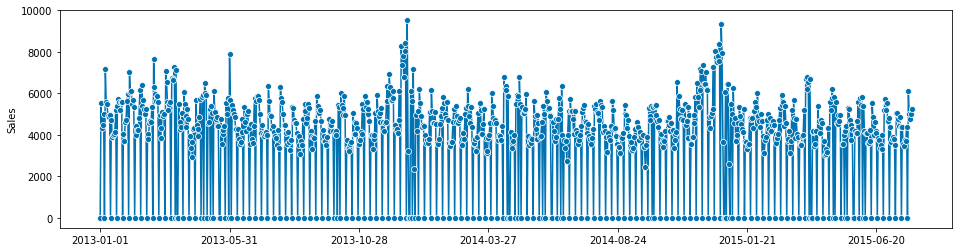


Figure Initial Rossmann time-series

## ETS

For implement ETS model we imported AutoETS from sktime library. First, we need to explain to Python that our series is daily time series, but not in example monthly with 29 missing values. It is performed with pandas.to\_datatime() function. After that split the unput data into y\_train and y\_test with temporal\_test\_split function. This is special version for time series data of well-known train\_test\_split function. Set the parameters ETS model in according with AAA abbreviation, namely, error, trend and seasonality have all additive nature. Seasonality is set to 7 days, because a priori we know that every Monday we have increase of sales due to fact that the store was closed on weekend. Fit the model and then get the predicted sales (fig.4).

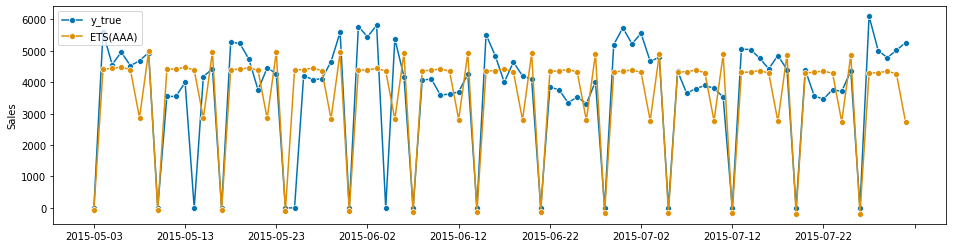


Figure True sales and predicted ones with ETS(AAA)

Calculate the evaluate metrics for ETS:

* + RMSE = 1201.412
  + MAPE = 0.258

As we can see, ETS model perfectly catch close state of the store, but has difficulties with peaks. If play with sp parameter in model then we have significantly gain of RMSE: from 1201 for sp=7 to ~1800 for sp is equal to 6 or 8 days. This fact again proves that initial guess for ETS is very important. As it turns out, the ETS model will show the best values of evaluate metrics from all considered models.

## ARIMA

For implementation ARIMA we need to install pmdarima and import from there auto\_arima.

The important moment is to determine initial order of ARIMA method, i.e. (p,q,d)(P,Q,D,s) where (p,d,q) order of the model for the number of AR parameters, differences, and MA parameters and (P,D,Q,s) order of the seasonal component of the model for the AR parameters, differences, MA parameters, and periodicity. Determine it with auto\_arima and get optimal orders in a form (0, 1, 1)(2, 1, 1, 12). After that fit model on y\_train and get predict for y\_test.

Calculate the valuation metrics with sklearn.metrics:

* + RMSE = 1824.009
  + MAPE = 0.361

Plot values of sales for the last 90 days of the survey (fig. 5)

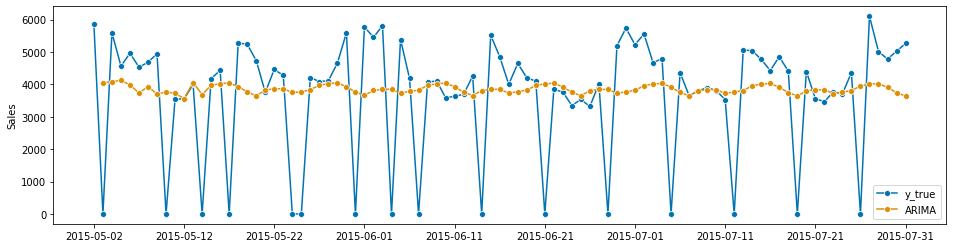


Figure True sales and predicted ones with ARIMA

As we can see from the figure, ARIMA for the most part of observations underestimates sales. ARIMA also could not catch peaks and related with them “seasonality of Monday”, when after weekends when stores are closed sales increases the first days of week.

## LSTM

This model requires more complicate data preprocessing, than abovementioned linear models.

LSTM are sensitive to the scale of input data, so we have to normalize it with MinMaxScaler. Moreover, input data must be represented in form a tensor [samples, time series, features] instead of current [samples, time series]. Transformation of the input data performed with numpy.reshape.

Finally, it began to create network structure. The structure of the model is sequential. The first layer is LSTM with 50 neurons. The second layer is Dense. Compile the model with optimizer Adam and MAE loss function.

As far as we normalized the input data, so predicted value of sales is normalized too. To convert them to absolute values we used inverse.transform.

Calculate the valuation metrics:

* + RMSE = 1694.804
  + MAPE = 0.349

And draw a plot (fig. 6)

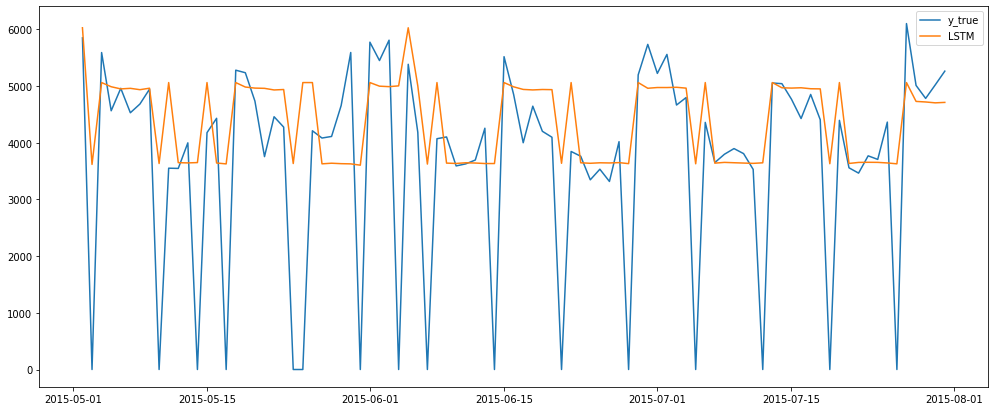


Figure True sales and predicted ones with LSTM

The presence of “memory” in the model can help to predict sales more exactly. Model quite well works with peaks.

## GPU

This model requires almost the same setup steps as LSTM model, but instead of LSTM layer consist GPU layer.

It is not surprised that we got almost the same results as for LSTM version. May be a little bit better:

* + RMSE = 1691.129
  + MAPE = 0.349

However, the predict curve looks more coherent (fig.7):

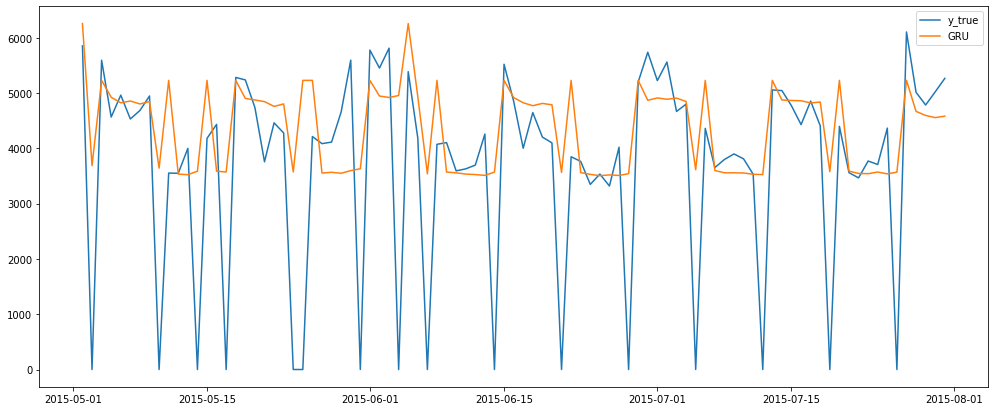


Figure True sales and predicted ones with GRU

## XGBoost

The implementation of XGBoost was performed with XGBRegressor from xgboost library. In contrast with LSTM and GRU, XGBoost operates with tensor in a form [samples, features]. We the same way normalized and inverse transformed data.

The evaluate metrics for XGBoost looks more better than for all previous models:

* + RMSE = 1466.571
  + MAPE = 0.329

If visualize the predicted data we can see that the model catch seasonality and even can predict the day-off of the store (fig.8):

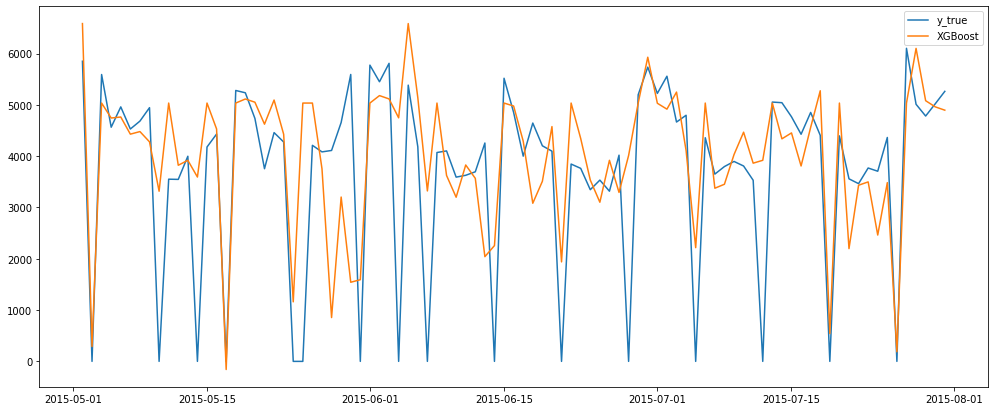


Figure True sales and predicted ones with XGBoost

# Conclusions

The most main lesson of this project: data preprocessing is important. Qualitative processing of data can supply both work model and its perfect results.

In table 1 evaluate metrics and on figure 9 predicted sales for all model represented.

Table 1 ‒ Evaluate metrics for models.

|  |  |  |
| --- | --- | --- |
| Model | RMSE | MAPE |
| ETS | 1201.412 | 0.258 |
| XGBoost | 1466.571 | 0.329 |
| GRU | 1691.129 | 0.349 |
| LSTM | 1694.804 | 0.349 |
| ARIMA | 1824.009 | 0.361 |

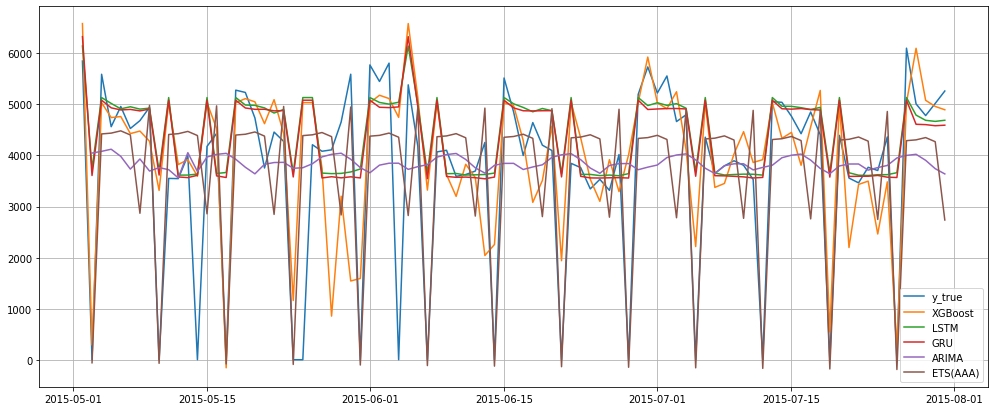


Figure True and predicted sales with different methods

ETS is the one of simplest and at the same time the more exact model. However, the crucial fact is to correct determine initial seasonality. Non-correctly estimation seasonality lead to significantly increase of errors (RMSE+50%).

ARIMA has a long procedure of fitting and setting. At the same time this model does not differ distinctive quality of predictions.

LSTM and GPU thanks to fact that they have very similar underpinning principles show about identical results. These models require a notable data preprocessing in contrast with other models.

# References

1. Zhao X., Xie J., Lau R. S. M. Improving the supply chain performance: Use of forecasting models versus early order commitments // Int. J. Prod. Res. 2001. Т. 39. № 17. С. 3923–3939.
2. Doganis P., Aggelogiannaki E., Sarimveis H. A combined model predictive control and time series forecasting framework for production-inventory systems // Int. J. Prod. Res. 2008. Т. 46. № 24. С. 6841–6853.
3. Alon I., Qi M., Sadowski R. J. Forecasting aggregate retail sales: A comparison of artifcial neural networks and traditional methods // J. Retail. Consum. Serv. 2001. Т. 8. № 3. С. 147–156.
4. Permatasari C. I., Sutopo W., Hisjam M. Sales forecasting newspaper with ARIMA: A case study // AIP Conference Proceedings. , 2018.
5. Yu Q. и др. Application of Long Short-Term Memory Neural Network to Sales Forecasting in Retail—A Case Study // Lecture Notes in Electrical Engineering. , 2018. С. 11–17.
6. Hwang R. C., Huang H. C., Hsieh J. G. Short-term power load forecasting by neural network with stochastic back-propagation learning algorithm // 2000 IEEE Power Engineering Society, Conference Proceedings. , 2000. С. 1790–1795.
7. Leung M. T., Chen A. S., Daouk H. Forecasting exchange rates using general regression neural networks // Comput. Oper. Res. 2000. Т. 27. № 11–12. С. 1093–1110.
8. Li X. и др. Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation // Environ. Pollut. 2017. Т. 231. С. 997–1004.
9. R. Rojas, Neural networks: a systematic introduction. Springer Science & Business Media, 2013.
10. Kim M. S. и др. Connecting Devices to Cookies via Filtering, Feature Engineering, and Boosting // Proceedings - 15th IEEE International Conference on Data Mining Workshop, ICDMW 2015. , 2016. С. 1690–1694.
11. Renov O., Anand T. R. Machine Learning Approach to Identify Users Across Their Digital Devices // Proceedings - 15th IEEE International Conference on Data Mining Workshop, ICDMW 2015. , 2016. С. 1676–1680.
12. Chen T., Guestrin C. XGBoost: A scalable tree boosting system // Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. , 2016.
13. Geurts M., Box G. E. P., Jenkins G. M. Time Series Analysis: Forecasting and Control // J. Mark. Res. 1977. Т. 14. № 2. С. 269.
14. Hand D. J. Forecasting with Exponential Smoothing: The State Space Approach by Rob J. Hyndman, Anne B. Koehler, J. Keith Ord, Ralph D. Snyder // Int. Stat. Rev. 2009. Т. 77. № 2. С. 315–316.
15. Brockwell P. J., Davis R. A. Introduction to Time Series and Forecasting. // Biometrics. 1998. Т. 54. № 3. С. 1204.
16. Lund R. Time Series Analysis and Its Applications: With R Examples // J. Am. Stat. Assoc. 2007. Т. 102. № 479. С. 1079–1079.
17. Hillmer S. C., Wei W. W. S. Time Series Analysis: Univariate and Multivariate Methods. // J. Am. Stat. Assoc. 1991. Т. 86. № 413. С. 245.
18. Taylor J. W. Short-term electricity demand forecasting using double seasonal exponential smoothing // J. Oper. Res. Soc. 2003. Т. 54. № 8. С. 799–805.
19. Taylor J. W. Triple seasonal methods for short-term electricity demand forecasting // Eur. J. Oper. Res. 2010. Т. 204. № 1. С. 139–152.
20. Trull O., García-Díaz J. C., Troncoso A. Initialization methods for multiple seasonal holt-winters forecasting models // Mathematics. 2020. Т. 8. № 2.
21. Holt C. C. Forecasting seasonals and trends by exponentially weighted moving averages // Int. J. Forecast. 2004. Т. 20. № 1. С. 5–10.
22. Winters P. R. Forecasting Sales by Exponentially Weighted Moving Averages // Manage. Sci. 1960. Т. 6. № 3. С. 324–342.
23. Granger, C.W.J.; Newbold, P. Forecasting Economic Time Series, 2nd ed.; Academic Press, Inc.: New York, NY, USA, 1986
24. Cho K. и др. Learning phrase representations using RNN encoder-decoder for statistical machine translation // EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference. , 2014. С. 1724–1734.
25. Gumani M. и др. Forecasting of sales by using fusion of machine learning techniques // 2017 International Conference on Data Management, Analytics and Innovation, ICDMAI 2017. , 2017. С. 93–101.
26. Willmott C. J. On the validation of models // Phys. Geogr. 1981. Т. 2. № 2. С. 184–194.