Feature Importance Study on Forecasting Hourly Retail Time Series

Bruno Vilela Mendes, Ana Maria Tomé, and José Moreira

DETI / IEETA – Universidade Aveiro
3810-193 Aveiro

Abstract. This work presents a predictive linear approach to hourly retail time series. The preprocessing steps must cope with time schedule normalization. After model fitting, a feature ranking strategy is applied to study the relevance of the lags for the prediction. An experimental study was conducted with different stores and different periods for training and testing. This study resulted in a low-complexity and robust regression pipeline which can be applied to this type of time series.

Keywords: Feature Ranking, Forecasting, Retail Time Series, Support Vector Regression (SVR)

1 Introduction

Machine learning has been applied to retail time series in various problems [1–3]. Better business decisions can be made if, for example, one can predict future revenues or assess how much stock a particular store should have available of a particular product. However, for these two examples, a weekly or even a monthly time series is sufficient in order to take accurate business decisions.

The present study aims to provide useful information on forecasting the hourly customer flow. Forecasting the customer flow on a hourly base can be very useful for any workforce management application. If a manager can provide an optimized labor planning, not only will it represent reduced costs due to not having unnecessary workers in the store but it will also avoid workers on day off being called to do extra hours because the store is overflown with customers. To overcome this problem a hourly customer flow forecasting could facilitate the workforce planning. Energy consumption is an example of such kind of high-frequency time series. Forecasting energy demands have been addressed with statistical or machine learning based methods [4]. The statistical models are considered likely appropriate for daily or weekly predictions. While machine learning based methods are considered more flexible and presenting good performances with different levels of aggregation.

Still, there are specific challenges when working with this type of high-frequency data which will be addressed in the next sections.

This paper is organized as follows. In section 2 a background regarding the main concepts discussed in this work as well as the specific challenges of this

topic are presented. Section 3 presents the workflow of this study. It starts with an overview of the datasets and the methods used to deal with this type of time series. Next, a linear machine learning model is optimized to forecast the customer flow. The input of the model is a feature vector formed by sliding windows of the time series and its output is the following window segment. Then, a feature importance study is presented. Section 4 presents a full implementation of a regression pipeline and the results obtained, which are discussed in section 5. Finally, section 6 summarizes the paper and presents future work ideas.

2 Forecasting Retail Time Series

A time series can be defined as a set of data collected at successive points in time or over successive periods of time [5]. It can also be decomposed into simpler components for analysis [6]. In retail time series analysis, the seasonal components are widely studied. In fact, theses series are known for having multiple seasonal components. The most common ones to come across in the literature are the annual seasonality and the weekly seasonality.

Multiple works have been made on forecasting in retail time series. Most of them focus on forecasting revenue [1], sales [2] and stock demand [3]. All these studies focus time series with a period no smaller than a day.

The challenge in working with retail time series on an hourly base is that all of them will be inherently irregular. In general, the stores have different opening hours and even the same store might have schedules that depend on the day of the week. The time series might also have a daily seasonality.

That happens because stores usually close during night time. It is also very common for daily schedules to change across the time series. A holiday or even a weekly close day will have a much greater impact in an hourly time series than what would have in a weekly or even a daily time series. Another particularity of hourly time series is that, as they have multiple data points during a day, they present another seasonal component: the daily seasonality.

To our knowledge up until now, the only other study made regarding this topic was [7]. This paper aims to make predictions of half-hourly inconsistent time series. The first step taken is the standardization of the daily store schedules. The present work will also start with this preprocessing step. However, it will go even further in this preprocessing step and standardize not only the days but also the weeks. The present work will also go beyond just forecasting and, through feature ranking, study and optimize this process.

2.1 Support Vector Regression

Support Vector Machines (SVM) [8] were proposed in the 90s primarily as a two-group classification model and are still widely used nowadays [9, 10]. Throughout the years, many adaptations were made from these SVMs, one of them being the Support Vector Regression (SVR) [11], which aims to solve regression problems instead of classification. SVR performs well even for small data sets. The model

is also applicable with linear and non-linear constraints without needing an exhaustive hyper-parameter tuning. The linear SVR model can be formulated with the following equation:

$$x[n] = \sum_{k=1}^{P} w_k x[n-k] + w_0, \quad n = P+1 \dots \tag{1}$$

where $n-th$ sample is a weighted sum of P past samples and bias w_0. The parameters of the model w_k are optimized using the LinearSVR from Scikit Learn [12]. In this work, the coefficients w_k will be used to study the relevance of the features (e.g. the lags of the model). In this study the size of the training sets highly surpassed the number of features (P), the model was configured to solve the primal optimization problem and the loss function used was the squared epsilon-insensitive loss.

2.2 Training and Test Strategies

Considering a training sequence with length M. For convenience let's assume that the sequence is $x[m], m = 0, \dots M-1$.

The training set is formed applying a sliding window to read $P+1$ samples of $x[m]$. In that case P represents the length of the feature vector as well as the number of samples in past to forecast one single value. Then the input and the output of the model are obtained by moving the window forward one time step. The training set has $N = M+1-(P+1)$ elements, where each input and output pairs are:

- the $n-th$ input is formed with the samples $\{x[n+0], x[n+1], \dots x[n+P-1]\}$, where $n = 0, 1, M-(P+1)$.
- the corresponding output is $\{x[n+P]\}$

This process is often referred to as transforming the time series into a supervised machine learning problem.

During the test phase, the model should predict a sequence of future values (often called Horizon). In this case it will be used the Autoregressive Predictions strategy [13]: the model makes single step predictions and each output is back as its input. Therefore, the prediction of a sequence of samples is performed iteratively by updating the input vector of the model with the previous prediction. The input feature vector works a a tapped delay line where in each iteration the samples move one step. That way lag 1 is fulfilled with the most recent prediction.

Standard Scaler In order to use the SVR efficiently, the data needs to go through a standardization process first. In this work, the z-score normalization was used, following the StandardScaler implementation from Scikit-Learn [12]. The scaling object was first fitted to the training data, then used to transform that training data and after to transform the test data. In the end, the predictions returned by the SVR model were transformed back to the original scale.

2.3 Performance Measures

There is a lot of discussion around which is the best performance measure to evaluate a regression model. This paper will not dive into this discussion. Instead, more than one performance measure will be taken into account when taking conclusions about the results obtained. Those performance measures will be:

- MAE: Mean Absolute Error;
- R2 Score: the Coefficient of Determination;
- RMSE: Root Mean Squared Error;
- MAPE: Mean Absolute Percentage Error;
- MedAE: Median Absolute Error.

The implementation of these measures from Scikit-Learn [12] were used for the present work. More information regarding the implementation itself can be found in [14].

3 Case Study and Methods

The present work comprises a total of seven time series, all of them consisting of customer flow data from real retail stores. All of the store have data recorded between January 2015 and October 2020. The first five were used for the study itself. After a feature study, a regression pipeline was build. Stores 6 and 7 were used later for validation. Table 1 describes these retail time series with more detail.

Table 1. Detailed description of the time series used in this study.

	store 1	store 2	store 3	store 4	store 5	store 6	store 7
Mean	113.07	91.16	85.92	70.57	58.94	76.69	90.91
Std	54.43	50.64	48.82	37.58	27.82	35.76	35.69
Min	1	1	1	1	1	1	1
25%	69	50	49	44	39	48	62
50%	106	78	79	67	57	71	85
75%	150	129	117	96	79	100	120
Max	316	262	245	228	164	224	224

Three months were chosen as targets for test predictions. Those were: November 2018, June 2019 and February 2020. The SVR model was fitted with the time series data up to evaluation month. Therefore the sizes of the training data are increasing. However, before forming the training data, the time series should be pre-processed in order to normalize the stores time schedules and imputing missing values.

3.1 Time Schedule Normalization

A preprocessing phase will then take place in order to keep all the days and the weeks with the same number of samples. After a preliminary analysis, it was noticed that the Sundays were almost every time close days. It was noted that, in the best case, one store was opened on around 20% of the Sundays and, in the worst case, one store was opened on around 2% of the Sundays. Based on these percentages, it was decided to remove all the Sundays from all the time series and work with 6 day weeks.

As mentioned in 2, hourly retail data is inherently irregular. A machine learning model does not read actual dates and times, but only ordered samples. Therefor, a preprocessing phase is needed in order to keep all the days and the weeks with the same number of samples.

First, all of the days are to be processed so that the working hours of the entire dataset are the same. In order to do that, a function was developed to find the most predominant day-schedule throughout each time series. Then, each time series was filtered using that information. As a consequence, some null gaps appeared in each time series. There are two kinds of missing values: days with gaps and even a full day might be missing. The latter can be caused by a weekday holiday. The gaps during the day are solved by linear interpolation with the two closest neighbour values within the day. Note that, for example, if a day time schedule is 9am-10pm, if the missing value is 9am the imputation is only based on the 10am value of the same day.

In the case of a missing day, the values of day are imputed with the values of the closest corresponding weekday of the previous week. If a day is missing during the first week of data, then the next corresponding weekday is used.

3.2 Feature Importance Study

As previously mentioned, all time series were transformed into a supervised machine learning problem. The main goal is to predict an entire month using an autoregressive prediction strategy. All the stores returned the same most regular working schedule in the time schedule normalization step, which was between 9am and 9pm inclusive. This interval comprises a total of 13 hours, which means 13 lags of information per day. The SVR model order is P=390 which corresponds to a window with 5 weeks of duration, the prediction of the next sample is the output of the model.

Table 2 displays the first results obtained. Each performance measure shown represents the mean value for the 3 horizons tested. These results were encouraging and led to the conclusion that this model was able to capture the time series behaviour.

Next, the coefficients assigned to the features were analysed. Note that the prediction is a weighted sum of the values corresponding to five weeks. Therefore if the absolute value coefficient is very close to zero has not the corresponding lag/feature have not much influence on the predicted value.

Table 2. Performance measures for the first SVR implementation.

	store 1	store 2	store 3	store 4	store 5
MAE	10.64	9.37	9.22	8.77	7.94
R2	0.92	0.94	0.93	0.89	0.84
RMSE	13.78	12.11	11.65	11.36	10.14
MAPE	0.11	0.13	0.13	0.13	0.13
MedAE	8.66	7.54	7.39	7.18	6.63

With that goal the coefficients were ordered by decreasing order of magnitude. Figure 1 shows the 20 largest values of the 15 trained models. And all present a similar profile. After the 10th feature, the value of the coefficients decreased around one order of magnitude. Due to this, no more than the 20 top features in each model were studied.

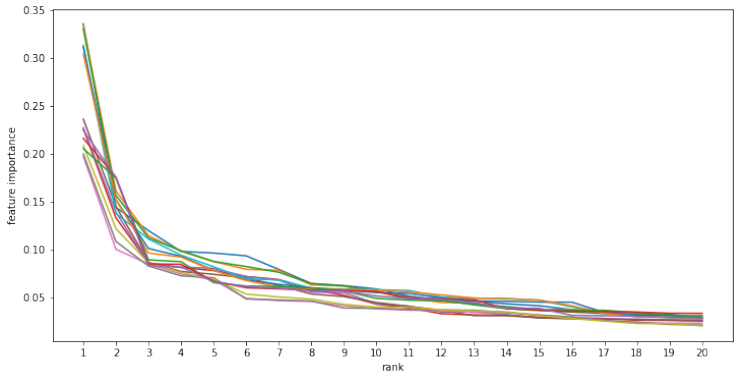

Fig. 1. Feature importance decay on the top 20 features.

Across every model, when predicting a target hour, the feature with the highest weight was always the one that represents the most recent lag, followed by the feature that represents the same hour of the target from one week ago. Another pattern quickly noticed was that, across all 15 cases, there were 11 features that were always ranked in the top 20. These were:

- The one lag referring to the previous hour;
- The five lags referring to the target hour from the previous five weeks;
- The five lags referring to the target hour from the previous five days.

Apart from these features, there were five more that appeared in the top 20 rank but their occurrence was not as regular as the ones mentioned above. These last five features referred to the hours previous to the target hour and their presence in the top 20 rank went as follows:

- 2 hours ago: 13 occurrences;
- 3 hours ago: 12 occurrences;
- 6 hours ago: 10 occurrences;
- 5 hours ago: 7 occurrences;
- 4 hours ago: 6 occurrences.

This analysis revealed that, besides the lag referring to the previous hour, the seasonal lags, be the daily or weekly, have a very strong weight in the decision making process, even stronger than much recent lags. Note that the model is constructed using only the working hours of the store, then lags less 13 might also be related with the previous day.

Training size A different type of analysis was performed regarding the size of the training data to see if larger sizes would benefit the overall performance of the SVR model. For this, only the R2 and MAPE performance measures were used because, as both of them represent percentages, they allow comparison between different time series data. Table 3 presents the mean of these performance measures for the time series for the three different test horizons.

Table 3. Performance measures for different sizes of training data.

	Nov 2018	Jun 2019	Feb 2020
R2	0.910	0.902	0.900
MAPE	0.128	0.132	0.120

For the first test horizon, there was already a significant number of data samples for the model to be trained upon (from January 2015 to October 2018). From the results in table 3, one can infer that the SVR model is not improving if the training data set is larger.

Autoregressive test error The model is fitted using a single step prediction error. However with the autoregressive test strategy the learned model is iteratively applied, feeding through the previous output as its new input. Therefore, the errors are naturally propagated for future predictions. In this study it was concluded that lag 1 is the most relevant for the predictions. Therefore it could be expected that it compromises the possibility of forecasting long horizons. Figure 2 shows the evolution of the error across the first test horizon for store 1. The values of the error are in the same range along the horizon, as there is not any rising trend. The profiles of the mean absolute daily error (see figure 2 on right) also present the same range of values in all the weeks of the period.

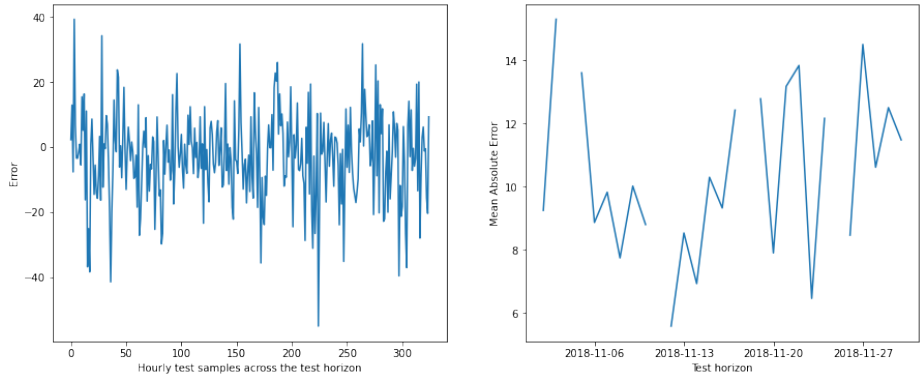

Fig. 2. Prediction Error for store. *Left*: time series error; *Right*: mean absolute error per day.

4 Regression Pipeline

Based on this study, a regression pipeline was developed in order to aggregate and implement all the steps mentioned above. Figure 3 illustrates that implementation.

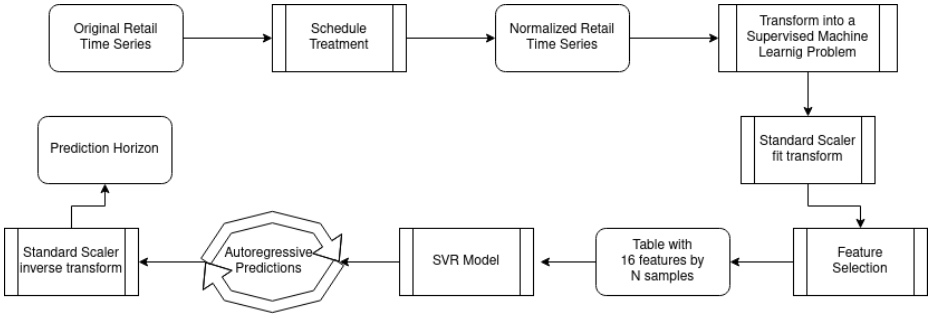

Fig. 3. Regression pipeline developed.

To summarize, the raw time series will first enter the time schedule normalization step. Here, the size (in hours) of its days and weeks are normalized. Next, the normalized time series are transformed into a supervised machine learning problem, to follow an autogressive strategy, and standardized with the z-score methodology. This step results in a table with all the hours from the previous month as input.

Next, a feature selection step is implemented in order to provide a future SVR model only the desired time lags. Following the analysis discussed in 3.2, a total of 16 features are to be selected. These are:

- The six lags referring to the six hours immediately before the target hour;
- The five lags referring to the target hour from the previous five weeks;
- The five lags referring to the target hour from the previous five days;

After fitted, the model performs autoregressive predictions and these predictions are then inverse transformed to the time series original scale. The output of this pipeline are the hourly predictions for the following month.

4.1 Results

That regression pipeline was first used on the five time series that served as ground for this study. Note that the complexity of the SVR model used in this pipeline is highly reduced due to the feature selection. For these five time series, the number of features to evaluate went from 390 to only 16. The performance measures of these tests are displayed in table 4.

Table 4. Performance measures of the implemented regression pipeline in the first five stores used for the feature study.

	store 1	store 2	store 3	store 4	store 5
MAE	11.75	9.65	9.97	11.67	9.58
R2	0.90	0.94	0.92	0.82	0.78
RMSE	15.17	12.48	12.55	14.58	11.90
MAPE	0.12	0.13	0.14	0.18	0.17
MedAE	9.46	7.86	8.38	9.90	8.18

Validation The final phase of this study will be to use the regression pipeline developed on two stores left for validation, stores 6 and 7. The configurations of the SVR model were the same as before. The results are displayed in table 5.

Table 5. Performance measures of the implemented regression pipeline in the validation set.

	store 6	store 7
MAE	7.98	11.01
R2	0.85	0.84
RMSE	10.42	13.82
MAPE	0.15	0.13
MedAE	6.33	9.70

5 Discussion

The application of the regression pipeline was proved to be a resilient solution. Despite cutting the number of features from 390 to only 16, stores 1 to 3 barely suffered any performance penalty. In stores 4 and 5, the loss of performance was a bit higher than in the first 3 stores but they were nevertheless good results. One can note that, despite the R2 and MAPE reporting a bigger error in stores 4 and 5, the non-percentage metrics didn't differ much.

The same logic was applied to data which was never analysed before - stores 6 and 7. The pipeline also proved very solid in these cases with quite good performance measures to support it.

All predictions results were also plotted in order make sure the values of the performance measures were not misleading. Figure 4 show the plot representation of a prediction example made with the presented regression pipeline featuring the first test horizon of store number 6 for comparison.

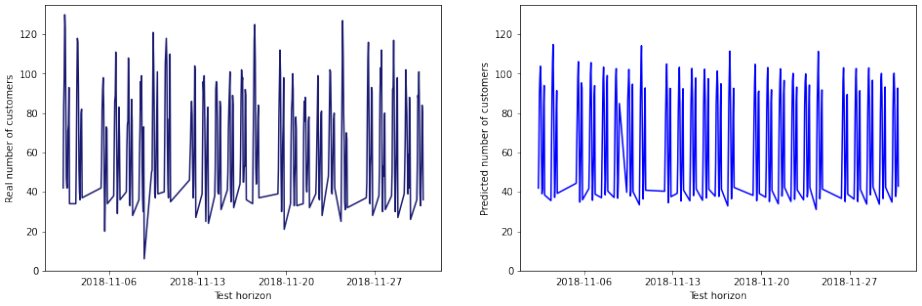

Fig. 4. Graphic representation of the predictions results obtained with the regression pipeline for store 6 in the first horizon test. To the left, the real customer flow is represented and the predicted values are shown to the right.

In the figure, it's possible to see that the pipeline was able to capture both weekly and daily seasonalities. In fact, if analysing the selected features, 10 out of 16 represent seasonal values, with 5 referring to the daily seasonality and 5 referring to the weekly. This capture is only possible because of the time schedule normalization performed early on. Without it, this seasonal representation would not be present in these exact lags.

Because none of the stores had a significant trend value, several artificially rising trends were added to see if the pipeline performance would drop. The results were exactly the same as when there was no trend, proving that this regression pipeline is robust to the presence of a trend in a time series.

6 Conclusion

In this paper, a linear SVR model was used in order to study the feature importance of hourly retail time series. This specific type of time series has particular differences from the regularly studies time series:

- they usually have multiple seasonalities;
- they are inherently irregular.

To minimize the irregularity problem, a time schedule normalization was performed, both at week and day level in order to preserve seasonality. The time series were then transformed into a supervised machine learning problem, which the lags of the past month being used as the input features. Next, a SVR model was fitted to five different time series in three different periods of time each. Finally, the coefficients provided by the SVR models were studied in order find out which features were considered the most important throughout the models. Out of a total of 390 features, 16 features were a common presence in almost every model. These features can be grouped as follows:

- the previous 6 time lags;
- the lags of the same hour of the previous 5 days;
- the lags of the same hour of the previous 5 weeks.

Based on the information gathered, a pipeline was implemented which performed the steps previously mentioned and a feature selection where only these 16 features would serve as input vector for SVR models. This complexity reduction proved quite efficient.

One topic not approached by this paper is the presence of outliers. This was because none of the analysed time series had a number of outliers that justified treatment. However, if a time series would be too contaminated with outliers, this could eventualy lead to performance loss.

References

1. Pundir, A.K., Ganapathy, L., Maheshwari, P., and Kumar, M.N.: Machine Learning for Revenue Forecasting: A Case Study in Retail business. 11th Annual IEEE Information Technology, Electronics and Mobile Communication Conference, IEMCON 2020 (2020)
2. Xie, X., Ding, J., Hu, G.: Forecasting the retail sales of china's catering industry using support vector machines. Proceedings of the World Congress on Intelligent Control and Automation (WCICA), IEEE. (2008)
3. Arunraj, N.S., Ahrens, D.: A hybrid seasonal autoregressive integrated moving average and quantile regression for daily food sales forecasting. International Journal of Production Economics. (2015)
4. Mouakher, A., Inoubli, W., Ounoughi, C., Ko, A.: Expect: EXplainable Prediction Model for Energy ConsumpTion. Mathematics. (2022)
5. Dodge, Y.: The Concise Encyclopedia of Statistics. Springer, New York, NY. (2008)
6. West, M. Time series decomposition. Biometrika. (1997)

7. Gusmão, P., Tomé, A.M., Moreira, J.: Forecasting Retail Client Flow with LSTMs on Inconsistent Time Series. 21th Portuguese Association for Information Systems Conference. (2021)
8. Cortes, C., Vapnik, V.: Support-Vector Networks. Machine Learning. (1995)
9. Mendes, B.V., Tomé, A.M., Santos, I.M., Bem-Haja, P.: Analysis of eyewitness testimony using electroencephalogram signals. 43rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). (2021)
10. Maktabi, M., Köhler, H., Ivanova, M., Neumuth, T., Rayes, N., Seidemann, L., Chalopin, C.: Classification of hyperspectral endocrine tissue images using support vector machines. International Journal of Medical Robotics and Computer Assisted Surgery. (2020)
11. Drucker, H., Surges, C. J. C., Kaufman, L., Smola, A., Vapnik, V.: Support vector regression machines. In Advances in Neural Information Processing Systems. Neural information processing systems foundation. (1997)
12. Pedregosa et al.: Scikit-learn: Machine Learning in Python. JMLR 12. (2011)
13. Lütkepohl, H.: Handbook of Research Methods and Applications in Empirical Macroeconomics. Chapter 6: Vector autoregressive models. Edward Elgar Publishing (2013)
14. Metrics and scoring: quantifying the quality of predictions, Scikit-learn, `https://scikit-learn.org/stable/modules/model_evaluation.html`