

Forecasting in Shipments: Comparison of Machine Learning Regression Algorithms on Industrial Applications for Supply Chain

Nunzio Carissimo[1], Raffaele D'Ambrosio[2], Milena Guzzo[1], Sabino Labarile[1], and Carmela Scalone[2(✉)]

[1] Code Architects, Via Campania, 1-3, 70029 Santeramo in Colle (Bari), Italy
{ncarissimo,mguzzo,slabarile}@codearchitects.com
[2] Department of Information Engineering and Information Science and Mathematics, University of L'Aquila, L'Aquila, Italy
{raffaele.dambrosio,carmela.scalone}@univaq.it

Abstract. Supply chains are very complex systems and their correct and efficient management represents a fundamental challenge, in which the practical needs of the corporate world can find answers together with the advanced skills of the academic world. This paper fits exactly in this area. In particular, starting from a project by the company Code Architects, we will illustrate how it is possible to make forecasts on shipments with machine learning tools, which can support business decisions.

Keywords: Supply Chain · Forecasting · Random Forest

1 Introduction

A supply chain is an entire network of producing and delivering a commodity or a service. It includes all the stages, from the very beginning of the source of the raw materials to the final delivery to customers. Since it involves all the aspects of the production process, such as, materials, informations, human resources and activities, the study and management of the supply chain are as important as it is complex for its correct functioning, see [19] and reference therein. This is also the reason why supply chain management is applied at different levels, with respect to multiple aspects and involves different technical figures. The present work is an example of interaction between academic and industrial cooperation. This is a key point, since, it is fundamental to understand the scopes and the targets from the industrial point of view, while it is decidedly more correct that the modeling, experimental and commentary phase of the results be taken care of by the academic component. The aim of this paper is to provide an analysis, comparing the behaviour of different machine learning algorithms, of an industrial case of study provided by Code Architects company. As pointed out in other works, see [4] and reference therein, it ideally should be interesting

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023, LNCS 13957, pp. 462–470, 2023.
https://doi.org/10.1007/978-3-031-36808-0_33

to involve in the study the whole supply chain, but, in the majority of the cases this is very difficult to do, due to several reasons, including convergence of business interests, sharing informations, complexity of large scale supply chain, etc. We focus on the part of the supply chain regarding the delivery of the final products to the end-users. In particular, we aim to select an efficient forecasting approach with respect to two fundamental aspects: days of delay in delivery and customer demand. These are typical problems in industrial applications, see for instance [9,20,22]. The products involved in the supply chain delivery step, are from food industry, therefore it appears clear that the delay issue is particularly crucial, not only for customer satisfaction, which is always a variable in play, but also for correct planning that avoids the deterioration of the goods as much as possible. Similar problems are presented in the works [13,14] and [18]. Forecasting, therefore, has an important role in preventing critical situations that have occurred in the past. Another fundamental point is related to the prevision of the demand from the customers.

Forecasting informations on the future demand, i.e. the quantities of commoditities ordered, is useful in terms of organization of the stocks, in particular to help in preventing the Bullwhip Effect, i.e. the phenomenon for which small fluctuations in demand at the retail level provokes progressively larger fluctuations in demand at the wholesale, distributor, manufacturer and raw material supplier levels. See [2,8,12], for a deepening on the Bullwhip effect. It is important to highlight that these considerations, both as regards the forecasts of the ordered quantities and as regards possible delays, also have a significant repercussion from an environmental point of view. As anticipated above, since the industrial project concerns food products subject to perishability, an organized and efficient delivery system avoids unnecessary waste. It is well known that these issues are of great actuality and importance.

We proceed with our analysis assuming our forecasting considerations based on the previous data in the delivery process. We consider three different Machine Learning Regression algorithms, basing our choices on the related literature, see [20,22] for instance.

The paper is organized as follows: in Sect. 2, we describe the composition of the data available for our analysis; in Sect. 3 we provide the motivations and brief descriptions of the choosen algorithms. Section 4 is dedicated to the experimental details and results.

2 Data

The dataset used, is made up of 180520 rows, each of which represent a shipment. To be more precise, for each shipment (row) we have the following informations:

1. **Customer Id**: an identification number associated to the customer;
2. **Customer Zipcode**: takes into account the distance for shipping;
3. **Product Id**: an identification number associated to the product;
4. **Product Quantity**: the ordered product units;

5. **Product Price**;
6. **Benefit per order**: net income derived from the shipment;
7. **Late Risk**: a binary variable associated to the shipment;
8. **Scheduled days for shipment**;
9. **Real days for shipment**.

The data related to Product Price and Benefict per order are real numbers. The remaining data associated to the other variables are integers. In particular, the Late Risk is binary: 1 is associated to a shipment for which there is a high risk associated to the delay (connected to the particularly high perishability), 0 otherwise. Based on the available data, we are interested in providing an efficient Machine Learning approach able to predict two crucial target variables: real days for shipment and quantities of product ordered. The prevision on the real days for shipment is important to prevent delay situations, comparing real and scheduled days. Preventing delays is important both because we work with perishable food products and for full customer satisfaction. Similarly, forecasting quantities is crucial for efficiently organizing the shipments and prevent the bullwhip effect. Being able to manage shipments in a harmonious way in terms of requests is therefore fundamental for all the correct functioning of the supply chain, and to have an algorithm of machine learning that supports the decisions turns out very profitable.

3 Machine Learning Algorithms

The ability of Machine Learning of learn complex relations among data, just using the data, i.e. without any other preliminary and sometimes expensive information (data distributions for instance) is extremely convenient. However, it is well known that, the quality of training dataset is a crucial factor to reach accurate results. In general, Machine Learning algorithms need large volume of data, for better processing and modeling results. This is the reason why, they are widely used in big data analysis. We aim to understand the applicability of Machine Learning models in delay and demand forecasting for our industrial case of study. With the awareness that there is no absolute algorithm better than others, but that the result depends on considerations relating to the company, the market, the product and the availability and structure of the dataset, we analyze and comment on the behavior of different Regression Algorithms, which are suited for our problems. In fact, these are Supervised Learning algorithms, to make predictions and Regression is a process of finding the correlations between dependent and independent variables. In other words, the problem of regression is to find a function that approximates mapping from an input domain to real numbers on the basis of a training sample. In our case, we have as dependent variables quantity/days for shipment and as independent the remaining ones. The model is trained on the "past data", and once the training is completed, it can easily predict the "future data". We compare the following Regression Algorithms:

1. **Support Vector Regression - Linear Kernel**
2. **Gradient Boosting Regression**
3. **Random Forest Regression**

This choice is due to the fact that Gradient Boosting and Random Forest have proved very efficient on problems similar to our. Linear Regression is a classical algorithm and it is interesting to test it and compare. There are several works in litaìerature based on the comparison of algorithms, see [11,16,22]. Now we propose a brief explanation for all the considered algorithms.

3.1 Random Forest

As we will see in the next Section, Random Forest is particularly important for our analysis, therefore we are pleased to provide a brief introduction following [5] and [21], which are optimal references for a deepening on this algorithm. Random forests is a supervised machine learning algorithm made up by a combination of tree predictors. The final prediction is made by averaging the predictions of each decision tree. The averaging makes a Random Forest better than a single Decision Tree, in terms of accuracy and reduction of overfitting. As observed in [22], Random Forest is a very convenient and easy to use algorithm, because of default hyper-parameters often produce a good prediction result.

As suggested by the name, a Random Forest is a tree-based ensemble with each tree depending on a collection of random variables. More precisely, for a p-dimensional random vector $X = (X_1, ..., X_p)^T$ representing the real-valued input or predictor variables and a random variable y representing the real-valued response, we assume an unknown joint distribution $P_{XY}(X, Y)$. The goal is to find a prediction function $f(X)$ for predicting Y. The prediction function is determined by a loss function $L(Y, f(X))$ and defined to minimize the expected value of the loss $\mathbb{E}_{XY}L(Y, f(X))$ where the subscripts denote expectation with respect to the joint distribution of X and Y. Intuitively, $L(Y, f(X))$ is a measure of how close $f(X)$ is to Y; it penalizes values of $f(X)$ that are a far from Y. Typical choices of L are squared error loss

$$L(Y, f(X)) = (Y - f(X))^2$$

for regression. It is known that, minimizing $\mathbb{E}_{XY}L(Y, f(X))$ for squared error loss gives the conditional expectation

$$f(x) = \mathbb{E}_{XY}(Y|X = x)$$

otherwise known as the regression function. Ensembles construct f in terms of a collection of so-called "base learners" $h_1(x), ..., h_J(x)$ and these base learners are combined to give the "ensemble predictor" $f(x)$. In regression, the base learners are averaged

$$f(x) = \frac{1}{J}\sum_{j=1}^{J} h_j(x).$$

In Random Forests the j-th base learner is a tree denoted $h_j(X, \Theta_j)$, where Θ_j is a collection of random variables and the Θ_j's are independent for $j = 1, .., J$. This is the basic idea of Random Forest, but there are several ways to realize this algorithm. The main feature is a good knowledge of the type of trees used as base learners. See also [3,21] for further informations and descriptions of the Random Forest algorithm.

3.2 Gradient Boosting

In Machine Learning setting, "Boosting" is referred to a combination of multiple simple models into a single composite model. This is also the reason why boosting is called additive model, since simple models (also called weak learners) are added one at a time, while keeping existing trees in the model unchanged. This combination of more and more simple models makes the complete finalmode a stronger predictor. The term "gradient" in "gradient boosting" refers to the fact that the algorithm uses gradient descent to minimize the loss. The function consists of a random output or response variable y, and a set of random input variables $X = x_1, .., x_n$. Given a training sample, the aim is to find a function, $F(X)$ that maps X to y, such that the loss function is minimized. Gradient Boosting is an efficient Machine Learning algorithm and a greedy additive strategy.

It is an iterative procedure in which, at each step, the residuals of the previous step are fitted, employing a Bass learning model. Gradient Boosting is highly considered and used in scientific literature, see [11,22].

3.3 Support Vector Regression - Linear Kernel

Support Vector Regression is a supervised machine learning algorithm, used to predict discrete values. The main features related to Support Vector Regression are:

1. **Kernel**: the function used to map a lower dimensional data into a higher dimensional data. We consider a linear kernel.
2. **Hyper Plane**: the line to predict the continuous value or target value.
3. **Boundary line**: there are two lines other than Hyper Plane which creates a margin. The support vectors can be on the Boundary lines or outside it. This boundary line separates the two classes.
4. **Support vectors**: this are the data points closest to the boundary. The considered distance of the points is minimum or least.

In Support Vector Regression, the best fit line is the hyperplane that has the maximum number of points. In simple regression, the idea is to minimize the error rate while in Support Vector Regression the idea is to fit the error inside a certain threshold, say ε, which means, work of Support Vector Regression is to approximate the best value within a given margin called ε - tube. The best fit line is the line hyperplane that has maximum number of points. We refer to [1,6,10] and reference therein, for further explanations on the Support Vector Regression algorithms.

4 Experimental Setting and Results

We analyze the data-types, dimensions, and missing values in the dataset. We have scaled the variables so that they are normalized with 0 mean and with a standard deviation equal to 1. To do that, we use the MinMaxScaler function in sklearn in Python. We have split the train data-set into a train and a validation data-set using the function train_test_split in Python. We consider the 80% of the dataset as train set (i.e. 144416 data rows). The remaining 20% are used as test set for validation.

4.1 Performance Estimation

Let us denote by $\{y\}_{i=1,..n}$ the true variables and by $\{\hat{y}\}_{i=1,..n}$ the predicted variables. For comparing and evaluating the performances of the algorithms we use the following measures:

$$\text{MSE} = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}$$

$$\text{MAE} = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n}$$

$$\text{R}^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}$$

where $\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$.

The mean square error (MSE) is a risk metric corresponding to the expected value of the squared (quadratic) error or loss. The mean absolute error (MAE) is the risk metric corresponding to the expected value of the absolute error loss or 1-norm loss. We compute the previous errors by using the Python functions mean_absolute_error and mean_absolute_error. R^2 si called coefficient of determination and it is computed by the Python function r2_score. R^2 represents the proportion of variance (of y) that has been explained by the independent variables in the model. It is an indicator of goodness of fit and therefore a measure of how well unseen samples are likely to be predicted by the model, through the proportion of explained variance. It ranges over interval 0 and 1. Best possible score is 1 and it can be negative (because the model can be arbitrarily worse).

4.2 Experimental Results

This section is dedicated to show and comment the experimental results. All instances are implemented in Python 3.6.2. All the learning and modeling processes are run on a PC with a 3.5 GHz Intel CoreTM i7-4770 processor and 16 GB RAM. We refer to Python library Scikit-learn, [15]. It should be pointed out that each algorithm has run five times and the outcomes are presented in Tables 1 and 2 are the average values. The metrics of performance are computed

with respect to the test set. In Table 1, the results regarding the prediction of demand commodities quantities are reported. We can see that Random Forest and Gradient Boosting have very good performances, with respect all the metrics. Anyway Random Forest show the best performace with respect all the parameters, confirming to be an accurate and stable algorithms. In this case, the Support Vector Regression does not show a good behaviour; in particular the determination coefficient is negative.

In Table 2, we provide the results for prediction of the three choosen Algorithms of the real days for shipping. In this case, Random Forest is not as excellent as in the previous case. However, we get a coefficient of determination equal to 0.8 which makes the algorithm absolutely good. Again Gradient Boosting is slightly below the results of Random Forest, while Support Vector Linear Regression shows very modest accuracy.

Table 1. Experimental results of the listed Algorithms for quantity forecasting.

Algorithms	MSE	MAE	R^2
Random Forest	0.0595	0.053	0.97
Gradient Boosting	0.1724	0.053	0.92
Support Vector Regression	2.2072	1.068	−0.05

Table 2. Experimental results of the listed Algorithms for real days for shipment forecasting.

Algorithms	MSE	MAE	R^2
Random Forest	0.5578	0.546	0.8
Gradient Boosting	0.5874	0.546	0.78
Support Vector Regression	1.0388	0.721	0.61

5 Conclusion

Demand forecasting is one of the most vital tasks in supply chain management and also a prerequisite of meeting customers' needs. We can comment on the results obtained by saying that evidently the dataset used is well structured for forecasting demand by the Random Forest algorithm. With less excellent performances, we are able to obtain a good feedback also on the real delivery times, in order to act in advance and avoid delays. This is particularly important for industrial data related to perishable foods. This paper may be seen as a good example of joint academic and industrial research, since a practical industrial issue found a satisfactory answer in the advanced tools of the academic world.

Further data informations may stimulate projects involving classification or cluster algorithms to understand customers behaviour and provide recommendation systems, starting from suited numerica analysis methods (as [7,17]). This may be subject for future research.

Acknowledgements. The authors thank the project PON "Ricerca Innovazione" 2014-2020 (PON) risorse FSE-REACT EU - DM 10 agosto 2021, n. 1062.

R. D'Ambrosio and C. Scalone are supported by GNCS-INDAM project and PRIN2017-MIUR project 2017JYCLSF "Structure preserving approximation of evolutionary problems".

This paper is part of the project: **OR.F.E.O.- ORchestrator For Enterprise Omniplatform** Decreto n r.0001497 del 11/03/2020, Progetto n. F/190189/00/X44 Fondo per la Crescita Sostenibile - Sportello "FABBRICA INTELLIGENTE" PON I&C 2014-2020, di cui al D.M. 5 marzo 2018 Capo III.

References

1. Awad, M., Khanna, R.: Support vector regression. In: Efficient Learning Machines. Apress, Berkeley (2015). https://doi.org/10.1007/978-1-4302-5990-9_4
2. Bhattacharya, R., Bandyopadhyay, S.: A review of the causes of bullwhip effect in a supply chain. Int. J. Adv. Manuf. Technol. **54**, 1245–1261 (2011)
3. Breiman, L.: Random forests. Mach. Learn. **45**, 5–32 (2001)
4. Carbonneau, R., Laframboise, K., Vahidov, R.: Application of machine learning techniques for supply chain demand forecasting. Eur. J. Oper. Res. **1843**, 1140–1154 (2008)
5. Cutler, A., Cutler, D.R., Stevens, J.R.: Random forests. In: Zhang, C., Ma, Y.Q. (eds.) Ensemble Machine Learning, pp. 157–175. Springer, New York (2012). https://doi.org/10.1007/978-1-4419-9326-7_5
6. David Sánchez, A.V.: Advanced support vector machines and kernel methods. Neurocomputing **55**(1–2), 5–20 (2003)
7. Guglielmi, N., Scalone, C.: An efficient method for non-negative low-rank completion. Adv. Comput. Math. **46**(2), 1–25 (2020). https://doi.org/10.1007/s10444-020-09779-x
8. Fransoo, J.C., Wouters, M.J.F.: Measuring the bullwhip effect in the supply chain. Supply Chain Manag. **5**(2), 78–89 (200)
9. Keung, K.L., Lee, C.K.M., Yiu, Y.H.: A machine learning predictive model for shipment delay and demand forecasting for warehouses and sales data. In: 2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) (2021). https://doi.org/10.1109/IEEM50564.2021.9672946
10. Kavitha, S., Varuna, S., Ramya, R.: A comparative analysis on linear regression and support vector regression. In: 2016 Online International Conference on Green Engineering and Technologies (IC-GET) (2016). https://doi.org/10.1109/GET.2016.7916627
11. Islam, S., Amin, S.H.: Prediction of probable backorder scenarios in the supply chain using distributed random forest and gradient boosting machine learning techniques. J. Big Data **7**(1), 1–22 (2020). https://doi.org/10.1186/s40537-020-00345-2
12. Lee, H.L., Padmanabhan, V., Whang, S.: The bullwhip effect in supply chains. Sloan Manag. Rev. **38**(3), 93–102 (1997)

13. Lin, Q., Zhao, Q., Lev, B.: Cold chain transportation decision in the vaccine supply chain. Eur. J. Oper. Res. **283**(1), 182–195 (2020)
14. Mercier, S., Uysal, I.: Neural network models for predicting perishable food temperatures along the supply chain. Biosys. Eng. **171**, 91–100 (2018)
15. Pedregosa, F., et al.: Scikit-learn: machine Learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
16. Raczko, E., Zagajewski, B.: Comparison of support vector machine, random forest and neural network classifiers for tree species classification on airborne hyperspectral APEX images. Eur. J. Remote Sens. **50**(1), 144–154 (2017)
17. Scalone, C., Guglielmi, N.: A gradient system for low rank matrix completion. Axioms **7**(3), 51 (2018)
18. Siddh, M.M., Soni, G., Jain, R.: Perishable food supply chain quality (PFSCQ): a structured review and implications for future research. J. Adv. Manag. Res. **12**(3), 292–313 (2015)
19. Stadtler H.: Supply chain management - an overview. In: Stadtler, H., Kilger, C. (eds) Supply Chain Management and Advanced Planning, pp. 9–36. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-74512-9_2
20. Vairagade, N., Logofatu, D., Leon, F., Muharemi, F.: Demand forecasting using random forest and artificial neural network for supply chain management. In: Nguyen, N.T., Chbeir, R., Exposito, E., Aniorté, P., Trawiński, B. (eds.) ICCCI 2019. LNCS (LNAI), vol. 11683, pp. 328–339. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-28377-3_27
21. Zhang, C., Ma, Y.: Ensemble Machine Learning: Methods and Applications. Springer, New York (2012). https://doi.org/10.1007/978-1-4419-9326-7
22. Zohdi, M., Rafiee, M., Kayvanfar, V., Salamiraad, A.: Demand forecasting based machine learning algorithms on customer information: an applied approach. Int. J. Inf. Technol. **14**(4), 1937–1947 (2022)