



HAL
open science

Demand Forecasting for an Automotive Company with Neural Network and Ensemble Classifiers Approaches

Eleonora Bottani, Monica Mordonini, Beatrice Franchi, Mattia Pellegrino

► **To cite this version:**

Eleonora Bottani, Monica Mordonini, Beatrice Franchi, Mattia Pellegrino. Demand Forecasting for an Automotive Company with Neural Network and Ensemble Classifiers Approaches. IFIP International Conference on Advances in Production Management Systems (APMS), Sep 2021, Nantes, France. pp.134-142, 10.1007/978-3-030-85874-2_14 . hal-04030416

HAL Id: hal-04030416

<https://inria.hal.science/hal-04030416>

Submitted on 16 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

Demand forecasting for an automotive company with neural network and ensemble classifiers approaches

Eleonora Bottani¹[0000-0002-6319-8080], Monica Mordonini²[0000-0002-5916-9770],
Beatrice Franchi³[0000-0002-7829-6855] and Mattia Pellegrino⁴[0000-0002-6592-7451]

^{1,2,3,4} Department of Engineering and Architecture, University of Parma, Viale delle Scienze
181/A, 43124 Parma, Italy

¹eleonora.bottani@unipr.it, ²monica.mordonini@unipr.it
³beatrice.franchi@unipr.it, ⁴mattia.pellegrino@unipr.it

Abstract. This work proposes the development and testing of three machine learning technique for demand forecasting in the automotive industry: Artificial Neural Network (ANN) and two types of Ensemble Learning models, i.e. AdaBoost and Gradient Boost. These models demonstrate the great potential that machine learning has over traditional demand forecasting methods. These three models will be compared to each other on the basis of the coefficient of determination R^2 and it will be shown which model has the greatest accuracy.

Keywords: demand forecasting; automotive; ANN; AdaBoost; Gradient Boost

1 Introduction

In economics terms, “demand” describes the consumers’ desire and willingness to pay a price for receiving specific goods or services and can be interpreted as the amount of desire expressed by the buyers. Accurate demand forecasting allows the company’s managers to improve their market performance, increasing profit and delineating internal policies and procedures on a sound basis. To this end, various tools have been proposed in the last decades for dealing with sales forecasting and making the process more accurate. These tools range from the traditional time series/statistical forecasting methods (e.g., exponential smoothing, moving average, ARIMA, or multivariate regressions) to the more recent applications of artificial intelligence (AI), including, among others, artificial neural networks, fuzzy logic or genetic algorithms [1]. As a matter of fact, research activities intended to develop more effective business forecasting techniques have recently evoked the usage of tools from the computational AI area. Machine learning (ML), as a branch of AI, is currently one of the most popular tools applied to demand forecasting. ML describes a suite of approaches able to provide systems with the ability to automatically improve their performance by learning from experience without being explicitly programmed. The present work proposes the comparison of various ML tools applied for the purpose of demand forecasting in the particular context of the automotive industry. In particular, the following ML approaches will be

discussed in this paper: an Artificial Neural Network (ANN) implementation and two types of Ensemble Learning, i.e. AdaBoost and Gradient Boost.

For an exhaustive review of the relevant literature on the use of ANN in demand forecasting, see [2], who have proposed an ANN implementation in the automotive context with the purpose of forecasting the demand of finished products. In the automotive field there are very few studies focusing on the issue of demand forecasting. To the best of the authors' knowledge, besides [2], the studies expressively focusing on this topic have been by [3] and [4]. In all studies, ANNs have been implemented as the demand forecasting tool. For literature review on AdaBoost used as an efficient predictor see [5], who demonstrated that AdaBoost is an appropriate model to evaluate the financial risk of Korean construction companies. For studies regarding the usage of the Gradient Boost algorithm in the context of sales forecasting, the reader is referred to [6]. In that work, Gradient Boost has shown good accuracy in forecasting and future B2B sales prediction.

Because the studies applying ANNs, AdaBoost and Gradient Boost approaches in the automotive field are very limited, this specific sector was chosen as the implementation context for this study. Besides this aspect, it is critical for an automotive company to forecast the future demands for finished products or spare parts, and it is also essential for optimizing supply chain operations and reducing costs [4].

The remainder of the paper is organized as follows. Section 2 details the methodological approach followed for carrying out the research. Section 3 shows the implementation of the chosen approaches and the corresponding outcomes, together with the comparison with the ANN approach previously applied. The relating results are discussed in Section 4. Conclusions, implications and future research directions are finally delineated in section 5.

2 Background Modelling

2.1 Framework

The approach proposed by [2] consists of several steps, which represent a useful guideline for building new methods to analyze a demand forecasting problem. The steps are reported in Fig. 1.

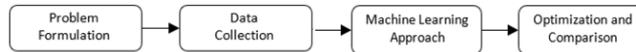


Fig. 1. Methodology logic tree.

1. *Problem Formulation*: The proposed context concerns the demand forecasting problem in an automotive company, called Company A for the sake of confidentiality.
2. *Data collection*: Company A provided us with a dataset for working about the context's task (see section 2.2 for a detailed description of the dataset).
3. *ML approach*: we built a new neural network to work with our dataset, and we implemented other two algorithms to work with: AdaBoost and Gradient Boost.

4. *Optimization and comparison*: we compared the estimates obtained the used methods.

2.2 Dataset description

The dataset contains three-monthly data from 2013 to 2018 (670 observations overall). Each row in the dataset represents a different part of the car: nose, front spoiler, machine structure, arms, brake discs, upright and bearings. Some components can be “right” or “left”, “upper” or “lower” and, consequently, the final number of components analysed is eighteen. Seven key variables are to be taken into account as inputs:

1. Components’ category, items number that can be grouped into a main class;
2. Championship to which sales data are related, since different championships correspond to different climatic conditions and cultural factors, which may affect accidents or spare parts demand;
3. Number of cars competing in each championship;
4. Number of races in each championship;
5. Ageing, simply attributable as YES or NO depending on the component;
6. Car life cycle, i.e. first year of life, interim period and last year of life;
7. Trimester, i.e. July-September, October-December, January-March, etc.

The following variables will be used to predict sales: number of cars; number of races; ageing.

2.3 Machine Learning Techniques

Neural Network. The ANN created in this study is a feed-forward neural network. As it can be seen in Fig. 2(a), a feed-forward NN consists of an input layer of neurons, an arbitrary number of hidden layers, and an output layer. Each neuron of a certain layer is connected to each neuron of the next layer. Fig. 2(b) shows one single neuron, which receives inputs (i_n signals weighted by weights w_n) from each neuron of the previous layer. Additionally, it receives a so-called bias input i_{bias} with weight w_{bias} . The transfer function $f(\Sigma)$ of a neuron translates the sum of all the weighted inputs into an output signal, which serves as input for all the neurons of the following layer [7].

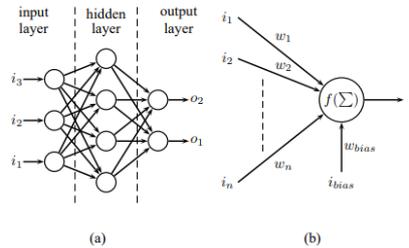


Fig. 2. (a) structure of a feed-forward neural network. (b) model of a single neuron.

The most used method to train a neural network is to present a set of samples (training set) as input to the network. The answer provided by the network for each example is compared to the desired answer; the difference (error) between the two is then evaluated and based on its value, the weights are adjusted. This process is repeated across

the entire training set until the network outputs return an error lower than a predetermined threshold.

Boosting. The ensemble learning is a way to combine different basic classifiers to derive a new one, more complex and more efficient. Moreover, these basic classifiers can be different in terms of hyper-parameters, representation and training set. The boosting algorithm belongs to this particular category and includes the AdaBoost and the Gradient boost techniques.

AdaBoost. Adaptive Boosting (AdaBoost, [8]) was the first boosting algorithm developed for classification and regression. It fits a sequence of weak learners on different weighted training data. The algorithm starts by predicting the original data set and giving equal weight to each observation. If the prediction is incorrect, using the first learner, the algorithm will give a higher weight to the observation. This procedure is iterated until the model reaches a predefined (threshold) value of accuracy. However, decision stumps are used with AdaBoost. They split the samples into two subsets based on the value of one generic feature. Each stump sets a feature and a threshold, and then splits the data into two new groups on each threshold side. Fig. 3 provides an example of the classic AdaBoost working.

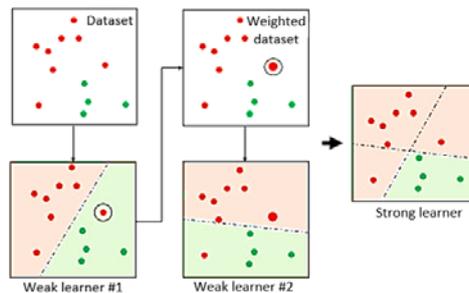


Fig. 3. Example of implementation of an AdaBoost classifier on a dataset with two features and two classes. Weak learner #1 improves on the mistake made by Weak learner #2.

AdaBoost has many advantages and disadvantages. It is typically easy to use because it does not need complex parameters' tuning procedures; similarly, it shows low sensitivity to the overfitting phenomenon. It is able to learn from a little set of features and add, in an incremental way, new information on which to work about. However, AdaBoost is quite sensitive to data having noises or abnormal values and, typically, it works with binary data, which can make it hard to adapt the algorithm to a categorical classification.

Gradient Boosting. This algorithm proposes a predictive model in the form of a combined set of weak predictive models, typically decision trees. After its original formulation, explicit regression gradient boosting algorithms were developed by [9-10]. Gradient boosting trains many models in a sequential way. Each new model gradually minimizes the whole system loss function, using the Gradient descent method, i.e. a first-

order iterative optimization algorithm for finding a local minimum of a differentiable function. The criterion is to take repeated steps in the gradient opposite direction of the function at its current point, because it is the steepest descent (see Fig. 4). The learning procedure fits new models to provide a better accuracy of the response variable. The aim is to construct new base learners which can be correlated with the loss function negative gradient.

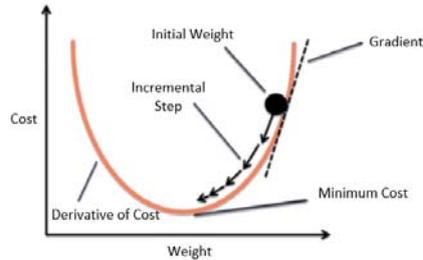


Fig. 4. Gradient descent on a series.

Gradient Boost too has some characteristics that makes it optimal for the targeted implementation. It often provides predictive accuracy that cannot be beat, it is very flexible, it can optimally deal with different loss functions and provide some hyper-parameter tuning that make the function fit very flexible. Unlike AdaBoost, it works with both categorical and binary data without difficulties. Moreover, it can handle missing data. Nevertheless, Gradient Boost always tries to minimize all errors, which can bring to overemphasize outliers and cause overfitting. It is also computationally expensive, and its high flexibility requires many parameters to be set.

3 Case study

3.1 Context

The company in which this study has been carried out (referred to as Company A for the sake of confidentiality) is based in the North of Italy and operates in the production of racing cars, by assembling components which can be either supplied or made in-house. Typically, cars are directly sold to teams participating to worldwide championships, while for components, the market is wider since also other companies or privates can be reached.

3.2 Neural Network Implementation

A feed-forward neural network was used in this work. The procedure used for the training process was iterated 1000 times and it is as follows: first of all, the input data were taken from the training dataset, adjusted based on their weights, and retrieved *via* a method that computes the output. After, the back-propagated error rate is computed as the difference between the predicted output of the neuron and the expected output of

the training dataset. Finally, some minor weight adjustments are made using the error weighted derivative formula.

3.3 AdaBoost Implementation

The implementation of this algorithm follows the steps listed below:

1. Normalization of each entry in the dataset;
2. Grouping of the data into the “train” and “test” sets;
3. Definition of the AdaBoost regressor;
4. Tuning of the hyper-parameters of a grid search. A grid search defines a search space as a grid of hyper-parameter values and evaluate every position in the grid. Hyper-parameters specification taken into account in this work include decision trees dimension and learning rate values.
5. Score computation and analysis.

3.4 Gradient Boost Implementation

For the targeted implementation, Gradient Boost had been chosen in the light of its optimal accuracy. The implementation of Gradient Boost follows the same steps listed for AdaBoost in the previous subsection.

4 Results

For an evaluation of the three different models implemented (ANN, AdaBoost and Gradient Boost algorithms) it is paramount to determine their goodness and accuracy. To do this, the coefficient of determination R^2 and RMSE were computed. R^2 index was because, for the purpose of our study, it is important that the algorithm is able to predict the real numerical data (i.e. to forecast the sales data). Indeed, R^2 is a statistical index that aims at evaluating whether the regression model can actually be used to make reliable predictions. R^2 is calculated as the ratio between the model deviance (σ_{model}^2) and the total deviance (σ^2) (Formula 1). R^2 measures the accuracy of the models implemented, in terms of its ability to fit the data. A value close to 1 means that there is an almost perfect correlation between the model and the data; on the contrary, a value close to 0 indicates absence of correlation between them, meaning that the adaptation of the mean is equivalent to the model created. Therefore, the more R^2 approaches 1, the greater the goodness of the model.

The root mean square error, on the other hand, show the difference between predicted values and those observed in the model (Formula 2).

$$(1) R^2 = \frac{\sigma_{\text{model}}^2}{\sigma^2} = \frac{\sum(y_i - \bar{y})^2}{\sum(y_i - \bar{y})^2} \quad (2) \sqrt{\frac{1}{n} \sum_{i=1}^n (f_i - o_i)^2}$$

Formula 1. Computation of R^2 **Formula 2.** Computation of Root Mean Squared Error

Where n is the number of samples, f are the predicted values and o are the empirical observed data. Therefore, the more RMSE approaches 0, the greater the goodness of

the model. Moreover, the algorithms were validated using a k -fold cross validation. K -fold cross validation consists in splitting a dataset in K sections, where each fold is used as a testing set at some point.

The general procedure is as follows:

1. Shuffle the dataset randomly;
2. Split the dataset into k groups;
3. For each unique group:
 1. Take the group as a hold out or test data set;
 2. Take the remaining groups as a training data set;
 3. Fit a model on the training set and evaluate it on the test set;
 4. Retain the evaluation score and discard the model;
4. Summarize the skill of the model using the sample of model evaluation scores.

Table 1 shows the R^2 and RMSE values for the models implemented in our study:

Table 1. R^2 and RMSE values for each model implemented.

	Neural network	AdaBoost	Gradient Boost
R^2	-0.10	0.37	0.27
RMSE	0.14	0.07	0.06

As can be seen, the best combination of R^2 and RMSE values (respectively equal to 0.37 and 0.07) is observed for the AdaBoost algorithm. Taking into account the relatively small sample of observations, the value denotes an excellent result. For Gradient Boost, R^2 scores 0.27 and RMSE scores 0.06. These values are way lower than AdaBoost, it is because AdaBoost works very well with weak learners and its exponential loss function fits well with the proposed model and generally it works better than Gradient Boost with small datasets. Nonetheless, in the light of the small size of the dataset, they could still be considered as acceptable. Finally, the ANN shows a negative R^2 (-0.10); this result clearly denotes the inability of the neural network model to fit to the data. Negative values of R^2 can be observed if the predictions that are compared with the correspondent results were not obtained from a procedure adhering model using such data. Once again, the possible reason is to be searched in the narrowness of the series used, which prevents the correct adaptation of the ANN to the specific problem. In addition, Figure 6 shows the predictions made by the previously described algorithms. The blue curve represents the real sales values, the red one instead is the prediction. The x -axis reflects the number of samples analyzed, while the y -axis reports the sales value.

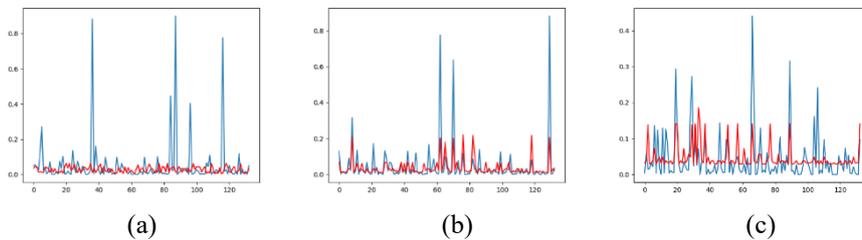


Fig. 6. Algorithms' predictions. (a) Neural Network, (b) AdaBoost, (c) Gradient Boost

5 Conclusions

The aim of this study was the implementation of three ML models for demand forecasting of eighteen car components for an automotive company located in the North of Italy, i.e. Neural Network, AdaBoost and Gradient Boost. The model that provides greater prediction accuracy turned out to be the AdaBoost algorithm, with an R^2 of 0,37 and a RMSE of 0,07 respectively. This means that AdaBoost algorithm fits well the categorization model proposed in this work and it is the best configuration among the three. Further insights can be considered. To have sales forecast data yet more accurate and precise, it would be interesting to repeat the study with larger datasets, consisting of thousands of rows. Furthermore, it could be interesting to repeat the study using a different time span (e.g. annual), as well as to study the demand forecasting for a greater number of car components. As a suitable future research direction, this study could be replicated in fields other than the automotive one, to take into account the presence of different components and to test its validity in various contexts.

References

1. Salais, T., Saucedo, J, Rodriguez Aguilar, R., Vela-Haro, J. Demand Prediction Using a Soft-Computing Approach: A Case Study of Automotive Industry. *Applied Sciences*, 10(3), article no.829 (2020).
2. Bottani, E., Tebaldi, L., Pindari, S. Demand forecasting in an automotive company: An Artificial Neural Network approach. In: Affenzeller, M., et al. (Eds.), *The 31st European Modeling & Simulation Symposium*, vol.1, pp.162-167 (2019).
3. Shahrabi, J., Mousavi, S.S., Heydar, M. Supply chain demand forecasting: a comparison of machine learning techniques and traditional methods. *Journal of Applied Sciences*, 9(3), 521-527 (2009).
4. González Vergas, C.A., Cortés, M.E. Automobile spare-parts forecasting: A comparative study of time series methods. *International Journal of Automotive and Mechanical Engineering*, 14(1), 3898-3912 (2017).
5. Heo, J., Yang, J. AdaBoost based bankruptcy forecasting of Korean construction companies. *Applied Soft Computing*, 24, 494-499 (2014).
6. Wisesa, O., Adriansyah, A., Osamah, I. and Khalaf, O. Prediction Analysis Sales for Corporate Services Telecommunications Company using Gradient Boost Algorithm. In: Adriansyah, A. (Ed.), *2nd International Conference on Broadband Communications, Wireless Sensors and Powering*, vol.1, pp.101-106 (2020).
7. Blum, C., Socha, K. Training feed-forward neural networks with ant colony optimization: An application to pattern classification. In: Gomide, F. (Ed.), *5th International Conference on Hybrid Intelligent Systems*, vol.1, pp.233-238. IEEE. (2005).
8. Freund, Y., Schapire, R. E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1), 119-139 (1997).
9. Friedman, J. H. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5), 1189-1232 (2001).
10. Friedman, J. H. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4), 367-378 (2002).