

Demand Forecasting for Inventory Management using Limited Data Sets: A Case Study from the Oil Industry

Jorge Ivan Romero-Gelvez, Esteban Felipe Villamizar, Olmer Garcia-Bedoya and Jorge Aurelio Herrera-Cuartas

Universidad de Bogotá Jorge Tadeo Lozano, Bogotá, Colombia

Abstract

This document's main focus is to present a way to solve forecasting issues using open source tools for time series analysis. First, we present an introduction to the hydrocarbon sector and time series analysis, later we focus on the solution methods based on supervised learning trained (support vector regression) with bio-inspired algorithms (Particle swarm optimization). We expose some benefits of use support vector machines and open source tools that focuses on variables like trend and seasonality. In this work, we chose the fb-prophet package and support vector regressor with scikit-learn as the primary tools because they have representative results dealing with limited data sets and Particle swarm optimization as training algorithm because of their speed and adaptability. Finally, we show the results and compare them with their RMSE obtained.

Keywords

Hydrocarbon, Forecasting, small time-series, support vector regressor, particle swarm optimization

1. Introduction

The hydrocarbon sector is the protagonist of the world economy's growth, seeking to expand as a critical piece of economic development through energy consumption, exploration, and production of oil where its primary producers are the USA, Russia, Saudi Arabia Iraq, Canada, and Iran. In 2019, 83 million barrels per day were produced, and Colombia reached participation of 863 thousand barrels, being the country number 22 within the scale of producers worldwide. In Colombia, the hydrocarbon sector has contributed significant growth, standing out as the engine of the country's development. The ACP estimates that the investment in production in 2020 will be around USD 4,050 million, that is, 25% higher than that of 2019 (USD 3,250 million). By region, 90% of the investment will be carried out in the Llanos Orientales, Valle Magdalena, and Caguán - Putumayo basins. By department, the following stand out: Goal USD 1,924 million, Santander USD 560 million, Casanare USD 496 million, Putumayo USD 239


ICAIW 2020: Workshops at the Third International Conference on Applied Informatics 2020, October 29–31, 2020, Ota, Nigeria

✉ jorgei.romerog@utadeo.edu.co (J.I. Romero-Gelvez); estebanf.villamizar@utadeo.edu.co (E.F. Villamizar); olmerg@gmail.com (O. Garcia-Bedoya); jorgea.herrerac@utadeo.edu.co (J.A. Herrera-Cuartas)

🆔 0000-0002-5335-0819 (J.I. Romero-Gelvez); 0000-0002-6964-3034 (O. Garcia-Bedoya); 0000-0003-0273-4043 (J.A. Herrera-Cuartas)



© 2020 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

million, and Arauca USD 101 million.

Also, the ACP projects that in 2020 there will be an investment in exploration and production of oil and gas of USD 4,970 million, 23% higher than in 2019. Indeed, the largest company in the country and the leading oil company in Colombia belong to the group of the 39 largest oil companies in the world and is one of the top five in Latin America. They have hydrocarbon extraction fields in the center, south, east and north of Colombia, two refineries, ports for the export and import of fuel and crude oil on both coasts, and a transportation network of 8,500 kilometers of pipelines and pipelines to throughout the entire national geography, which interconnect production systems with large consumption centers and maritime terminals.

Given this panorama, the industries must plan strategies to manage and control their inventories since their importance lies in obtaining profits. Inventory management plays a vital role within the business chain, which is the buffer between two processes, supply, and demand. This can be known or unknown, variable, or constant. The sourcing process contributes goods to inventory while demand consumes the same inventory. This is necessary due to differences in rates and times between supply and demand, and this difference must be attributed to internal or external factors. Endogenous factors are policy issues, but exogenous factors are uncontrollable. Internal factors include economies of scale, smoothing of operation, and customer service, the most important exogenous factor being uncertainty.

Given that inventory control is a critical aspect for effective management and administration, in order to guarantee the availability of equipment, spare parts, and materials to meet the needs of expenses and projects with the expected quality, cost, and opportunity. Likewise, the materials management process is to ensure that the materials that register stock in the warehouse correspond to optimal levels of inventories, such quantity must fully meet the needs of the company with the minimum investment.

2. Literature review

There are applications in forecasts using SVR and it is more common to find them in recent years, since there is a special interest in machine learning applications to make predictions on time series. An example of this can be seen in [1] with financial forecasts, [2, 3] over rainfall predictions, electric load forecasting [4, 5, 6], forecasting carbon price [7] among many others. Next, we present a brief introduction to the time series and later we will deal with the comparison of the applied models.

2.1. Time series analysis

The analysis of time series and demand forecasts becomes the primary input for the MRP model. For this reason, it is proposed to contrast different methods that allow considering seasonal periods, and that may also include new demand observations to adjust the model in real-time. According to [8] Time-series methods refers to a set of observations of real phenomena (like mathematical, biological, social sciences, physical, economic among others) given as part of a discrete set in time. The main thought is that past data can be utilized to generate future estimations. In time series analysis is common to try to get patterns of data as trend, seasonality, cycles, and randomness as inputs to modeling the phenomena.

- Trend: The data set exhibits a stable pattern of growth or decrease.
- Seasonality: A seasonality pattern are those that are repeated at fixed intervals.
- Cycles: The variation of cycles is similar to seasonality, except that the duration and the magnitude of the cycle varies.
- Randomness: A random series is where you do not have a recognized pattern of data. One can generate random series of data that have a specific structure. The data that seems to have apparently a randomness , actually have a specific structure. Actually the really random data fluctuate around a fixed average.

2.2. Support Vector machine and support vector regressor

According with [9] In machine learning, support vector machine proposed by Vapnik is one of the most popular approaches for supervised learning [10, 11]. This model resembles logistic regression in that a linear function $\mathbf{w}^\top \mathbf{x} + b$ drives both. Its main difference is that the logistic regression produces probabilities, the support vector machine produces a class identity. The SVM predicts that the positive class is present when $\mathbf{w}^\top \mathbf{x} + b$ is positive. Likewise, it predicts that the negative class is present when $\mathbf{w}^\top \mathbf{x} + b$ negative. A notable feature of support vector machines is the kernel, considering that many algorithms can be written in the form of a dot product. As an example, the rewritten SVM linear function is shown as:

$$\mathbf{w}^\top \mathbf{x} + b = b + \sum_{i=1}^m \alpha_i \mathbf{x}^\top \mathbf{x}^{(i)} \quad (1)$$

where $x^{(i)}$ is a training example and α is a vector of coefficients. Rewriting the learning algorithm this way allows us to replace x by the output of a given feature function $\phi(\mathbf{x})$ and the dot product with a function $k(\mathbf{x}, \mathbf{x}^{(i)}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{x}^{(i)})$ called a kernel. The (\cdot) operator represents an inner product analogous to $\phi(\mathbf{x})^\top \phi(\mathbf{x}^{(i)})$. For some feature spaces, we may not use literally the vector inner product. In some infinite dimensional spaces, we need to use other kinds of inner products, for example, inner products based on integration rather than summation. After replacing dot products with kernel evaluations, we can make predictions using the function

$$f(\mathbf{x}) = b + \sum_i \alpha_i k(\mathbf{x}, \mathbf{x}^{(i)}) \quad (2)$$

This function is nonlinear with respect to x , but the relationship between $\phi(\mathbf{x})$ and $f(\mathbf{x})$ is linear. Also, the relationship between α and $f(\mathbf{x})$ is linear. The kernel-based function is exactly equivalent to preprocessing the data by applying $\phi(\mathbf{x})$ to all inputs, then learning a linear model in the new transformed space. The kernel trick is powerful for two reasons. First, it allows us to learn models that are nonlinear as a function of x using convex optimization techniques that are guaranteed to converge efficiently. This is possible because we consider ϕ fixed and optimize only α , i.e., the optimization algorithm can view the decision function as being linear in a different space. Second, the kernel function k often admits an implementation that is significantly more computational efficient than naively constructing two vectors

and explicitly taking $\phi(x)$ their dot product. In some cases, $\phi(x)$ can even be infinite dimensional, which would result in an infinite computational cost for the naive, explicit approach. In many cases, $k(x, x')$ is a nonlinear, tractable function of x even when $\phi(x)$ is intractable. As an example of an infinite-dimensional feature space with a tractable kernel, we construct a feature mapping $\phi(x)$ over the non-negative integers x . Suppose that this mapping returns a vector containing x ones followed by infinitely many zeros. We can write a kernel function $k(x, x^{(i)}) = \min(x, x^{(i)})$ that is exactly equivalent to the corresponding infinite-dimensional dot product.

2.3. Support vector regressor

According to [12] the basic idea of SVR is to map the data x into a high dimensional feature space \mathcal{F} by nonlinear mapping ϕ and to do linear regression in this space. Also, according to [13] we can consider a set of training data $f(x) = (w \cdot \phi(x)) + b$, where each $x_i \in R^n$ denotes the input space of the sample and has a corresponding target value $y_i \in R$ for $i = 1, \dots, l$, where l corresponds to the size of the training data. The idea of the regression problem is to determine a function that can approximate future values accurately. The generic form of SVR can be seen as follows

$$f(x) = (w \cdot \phi(x)) + b \quad (3)$$

where, $w \in R^n$, $b \in R$ and ϕ denotes a nonlinear transform to high-dimensional space. Our goal is to find the value of w and b such that values of $f(x)$ can be determined by minimizing the regression risk

$$R_{\text{reg}}(f) = \sum_{i=1}^{\ell} C(f(x_i) - y_i) + \lambda \|w\|^2 \quad (4)$$

where $C(\cdot)$ is a cost function, λ is a constant, and vector w can be written in terms of data points as

$$w = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) \phi(x_i) \quad (5)$$

By substituting eq.5 into eq.3, the generic equation can be rewritten as

$$f(x) = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) (\phi(x_i) \cdot \phi(x)) + b = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) k(x_i, x) + b \quad (6)$$

In eq.6, the dot product is replaced with a kernel function $k(x_i, x) = (\phi(x_i) \cdot \phi(x))$. Kernel functions enable the dot product to be performed in high-dimensional feature space using low-dimensional space data input without knowing the transformation ϕ . All kernel functions must satisfy Mercer's condition that corresponds to the inner product of some feature space. The radial basis function RBF is commonly used as the kernel for regression

$$k(x_i, x) = \exp \left\{ -\gamma |x - x_i|^2 \right\} \quad (7)$$

2.4. Particle swarm optimization

Particle swarm optimization is a computational technique that optimizes a problem by iteratively attempting to promote a candidate solution concerning a given degree of quality. It solves a problem by producing a population of candidate solutions called particles and moving them around the search-space with particular position and velocity. Each particle's movement is influenced by its local best-known place but is also guided toward the best-known positions in the search-space, which are renewed as another particle found better solutions. This is expected to move the swarm toward the best solutions. According to [14] the main inputs for the formulation of this algorithm can be seen as follows:

- D is the dimension of the search space.
- E is the search space, a hyperparallelepiped defined as the euclidean product of D real intervals.

$$E = \bigotimes_{d=1}^D \left[\min_d, \max_d \right] \quad (8)$$

The standard form of the algorithm composes by a set of particles (called swarm), made of a position in the search space, the fitness value at each position, a velocity for displacement, a memory that contains the best position and last the fitness value of the previous best. The search is performed in two phases, initialization of the swarm and a cycle of iterations. The main steps can be seen as follows:

- Initialisation of the swarm: pick a random position in search space and pick a random velocity.
- Iteration: compute the new velocity, move and compute the new fitness.
- Stop .

2.5. The prophet forecasting model

Prophet is an open source tool for forecasting time series observations based on an additive model in which nonlinear trends adjust to seasonality. Their results improve with data that includes compromises with strong seasonal effects and a considerable amount of historical data. We use Prophet open-source software in Python [15] based in a decomposable time series model [16] components: trend, seasonality, and vacations. They are combined as follows:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \quad (9)$$

Were $g(t)$ is the trend function which models non periodic changes in the value of the time series, $s(t)$ represents periodic changes, and $h(t)$ represents the effects of vacations which occur on potentially irregular schedules over one or more days. The error term ϵ_t represents any idiosyncratic changes which are not accommodated by the model; We will make a parametric assumption with normally distributed t .

2.5.1. Prediction model

The development of an algorithm is necessary to obtain a data history in which the breakdown of the monthly inventory value is represented by the BSE material and transit corresponding to two years. In this analysis of generated data, factors such as inventory value and material dependence are taken into account. These have a significant influence on the model's behavior since they provide realism and specific variations of the trend line, which are of interest to optimize its management. All these data have been obtained from the materials management system. Subsequently, these data will be processed and analyzed in order to see the interaction between them, carrying out a parameterization that allows characterizing the logic of generation of the historical.

3. Data description

The warehouse corresponds to the code assigned in the information system that represents an organizational unit or warehouse, corresponds to the physical place where the materials are stored, which allows differentiation of material stocks. In this case, the logistics center 2000 is taken, as a Detail describes the types of warehouse: Imported: corresponds only to material in transit of expenses or projects. Expenses: the types of warehouses are associated with new material in good condition and repaired material in good condition. These materials are characteristic of the operation and maintenance, such as spare parts, consumables, and supplies for the operation, equipment, lubricants, consumption tools, parts from manufacturers, spare parts. Projects: this material is part of the business investment with a physical location in patios and covered warehouse, this material is acquired according to the project's requirement, given that the material is no longer required by the project, it is assigned as not required material or in the process of sale which is offered to other projects of the business group.

4. Solution method and results

In order to solve the problem, we contrast two methods, the fb-prophet and PSO-SVM.

- **Forecasting Model selection:** In order to use the method that generates less error ϵ_t . First, we apply support vector machine with particle swarm optimizations as global optimization algorithm. In addition, Fb-prophet (black box method) is also used. Next, we select the method with the least error of them
- **IDE:** IPython/Jupyter notebooks and Google-Colab.

4.1. Results

5. Conclusions

The main forecast topic with a variety of backgrounds should make more forecasts than they can do manually—the first component of our forecast. The system is the new model that we

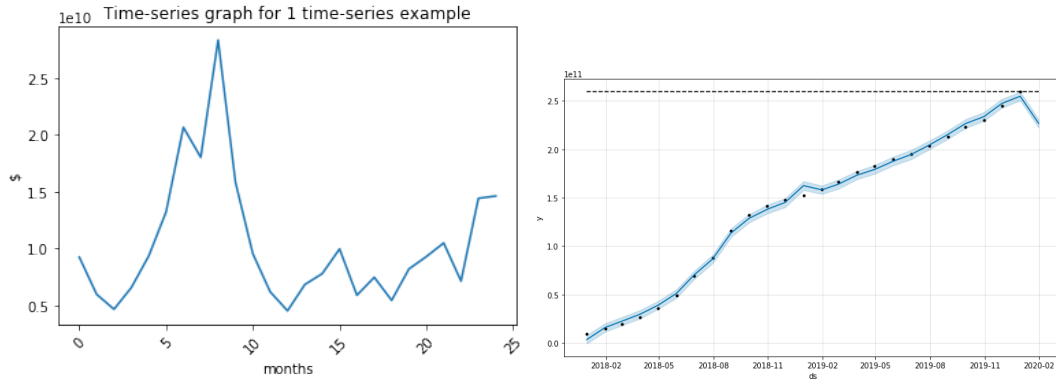


Figure 1: Fb-prophet black box model implementation

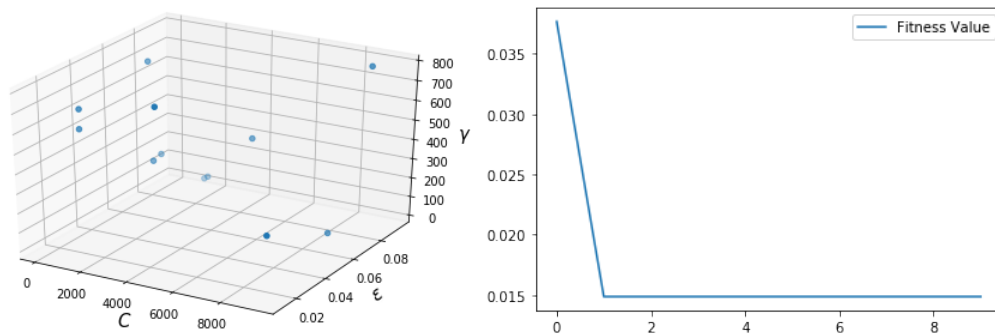


Figure 2: Initialization of the PSO-SVM model

Table 1
RMSE error

	RMSE
fb-prophet	3471658053
pso-svm	0.022833

have developed in many prediction iterations of a variety of data in FbProphet. We use a simple modular regression model that often works well with predetermined parameters, and that allows you to select the components that are relevant to your forecast problem and quickly make adjustments as needed. The success of model lies in its ability to adjust the positions of all particles in an area of search space with satisfactory solutions.

According to a determined objective function to minimize, in this case, the root mean square error. It measures the dispersion of error in forecast, this value is the difference between real demand and the forecast squaring, disabling those periods where the difference was higher compared to others. From this calculation, decisions against forecast models and their results

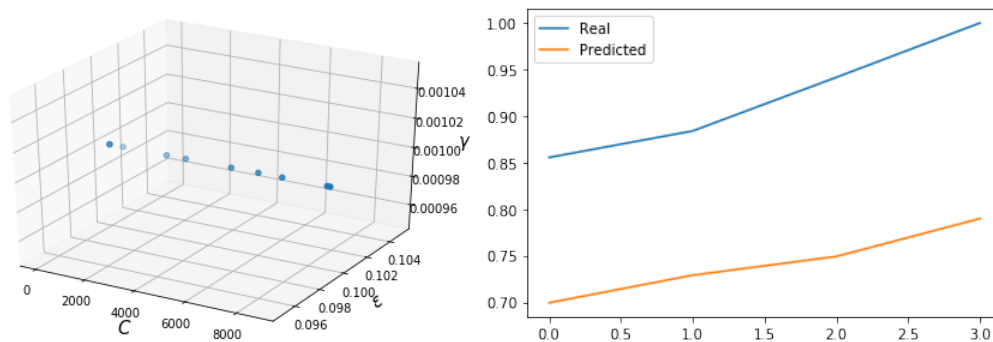


Figure 3: Results of PSO-SVR

are guided to the best choice. Likewise, it is established that this type of problems are solved by evolutionary algorithms, the importance of using these algorithms as the swarm of particles lies in the high efficiency in generating predictions with better performance, the formulation is reduced to characterize the movement of the particles based on a speed operator that must associate exploration and convergence by decomposing the speed into three components in order to decipher the study behavior.

References

- [1] C.-J. Lu, T.-S. Lee, C.-C. Chiu, Financial time series forecasting using independent component analysis and support vector regression, *Decision support systems* 47 (2009) 115–125.
- [2] A. D. Mehr, V. Nourani, V. K. Khosrowshahi, M. A. Ghorbani, A hybrid support vector regression–firefly model for monthly rainfall forecasting, *International Journal of Environmental Science and Technology* 16 (2019) 335–346.
- [3] C. Balsa, C. V. Rodrigues, I. Lopes, J. Rufino, Using analog ensembles with alternative metrics for hindcasting with multistations, *ParadigmPlus* 1 (2020) 1–17.
- [4] Z. Zhang, W.-C. Hong, Electric load forecasting by complete ensemble empirical mode decomposition adaptive noise and support vector regression with quantum-based dragonfly algorithm, *Nonlinear Dynamics* 98 (2019) 1107–1136.
- [5] Y. Yang, J. Che, C. Deng, L. Li, Sequential grid approach based support vector regression for short-term electric load forecasting, *Applied Energy* 238 (2019) 1010–1021.
- [6] Z. Zhang, W.-C. Hong, J. Li, Electric load forecasting by hybrid self-recurrent support vector regression model with variational mode decomposition and improved cuckoo search algorithm, *IEEE Access* 8 (2020) 14642–14658.
- [7] B. Zhu, D. Han, P. Wang, Z. Wu, T. Zhang, Y.-M. Wei, Forecasting carbon price using empirical mode decomposition and evolutionary least squares support vector regression, *Applied energy* 191 (2017) 521–530.
- [8] F. R. Jacobs, *Manufacturing planning and control for supply chain management*, McGraw-Hill, 2011.

- [9] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016. <http://www.deeplearningbook.org>.
- [10] B. E. Boser, I. M. Guyon, V. N. Vapnik, A training algorithm for optimal margin classifiers, in: Proceedings of the fifth annual workshop on Computational learning theory, 1992, pp. 144–152.
- [11] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (1995) 273–297.
- [12] K.-R. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, V. Vapnik, Predicting time series with support vector machines, in: International Conference on Artificial Neural Networks, Springer, 1997, pp. 999–1004.
- [13] C.-H. Wu, J.-M. Ho, D.-T. Lee, Travel-time prediction with support vector regression, IEEE transactions on intelligent transportation systems 5 (2004) 276–281.
- [14] M. Clerc, Beyond standard particle swarm optimisation, in: Innovations and Developments of Swarm Intelligence Applications, IGI Global, 2012, pp. 1–19.
- [15] S. J. Taylor, B. Letham, Forecasting at scale, The American Statistician 72 (2018) 37–45.
- [16] A. C. Harvey, S. Peters, Estimation procedures for structural time series models, Journal of Forecasting 9 (1990) 89–108.