

# How We Combined Different Methods to Create Advanced Time Series Prediction

Taras Firman  
Data Scientist at ELEKS

Today, businesses need to be able to predict demand and trends to stay in line with any sudden market changes and economy swings. This is exactly where forecasting tools, powered by Data Science, come into play, enabling organizations to successfully deal with strategic and capacity planning. Smart forecasting techniques can be used to reduce any possible risks and assist in making well-informed decisions. One of our customers, an enterprise from the Middle East, needed to predict their market demand for the upcoming twelve weeks. They required a market forecast to help them set their short-term objectives, such as production strategy, as well as assist in capacity planning and price control. So, we came up with an idea of creating a custom time series model capable of tackling the challenge. In this article, we will cover the modelling process as well as the pitfalls we had to overcome along the way.

## There is a number of approaches to building time series prediction ....and neither fit us

With the emergence of the powerful forecasting methods based on Machine Learning, future predictions have become more accurate. In general, forecasting techniques can be grouped into two categories: qualitative and quantitative. Qualitative forecasts are applied when there is no data available and prediction is based only on expert judgement. Quantitative forecasts are based on time series modeling. This kind of models uses historical data and is especially efficient in forecasting some events that occur over periods of time: for example prices, sales figures, volume of production etc.

The existing models for time series prediction include the ARIMA models (<https://onlinecourses.science.psu.edu/stat510/node/49>) that are mainly used to model time series data without directly handling seasonality; VAR models

(<https://onlinecourses.science.psu.edu/stat510/node/79>), Holt-Winters

(<https://www.otexts.org/fpp/7/5>) seasonal methods, TAR models

(<https://onlinecourses.science.psu.edu/stat510/node/82>) and other. Unfortunately, these algorithms may fail to deliver the required level of the prediction accuracy, as they can involve raw data that might be incomplete, inconsistent or contain some errors. As quality decisions are based only on quality data, it is crucial to perform preprocessing to prepare entry information for further processing.

## Why combining models is an answer

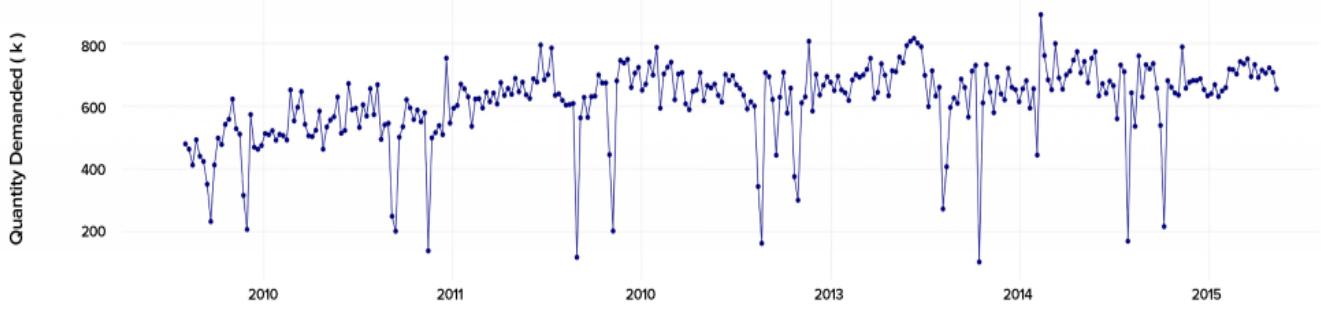
It is clear that one particular forecasting technique cannot work in every situation. Each of the methods has its specific use case and can be applied with regard to many factors (the period over which the historical data is available, the time period that has to be observed, the size of the budget, the preferred level of accuracy) and the output required. So, we faced the question: which method/methods to use to obtain the desired result? As different approaches had their unique strengths and weaknesses, we decided to combine a number of methods and make them work together. In this way, we could build a time series model capable of providing trustworthy predictions to ensure data reliability and time/cost saving. And this is how we did it.

## The modeling process; let's dive into the details

The demand data depends on various factors that can influence the result of the forecast, such as the price and types of goods, geographical location, the country's economics, manufacturing technology, etc. As we wanted our time series model to provide the customer with high-accuracy predictions, we used the interpolation

(<https://ece.uwaterloo.ca/~dwharder/NumericalAnalysis/05Interpolation/>) method for missing values to ensure that the input is reliable.

When conducting the time series analysis in Python 2.7., we analyzed the past data starting from 2010 to 2015 to calculate precisely the demand and predict its behavior in the future.



(<https://labs.eleks.com/wp-content/uploads/2016/10/Fig1-The-demand-data-over-2010-2015-timeframe.png>)

*Fig. 1. The demand data over the 2010-2015 timeframe*

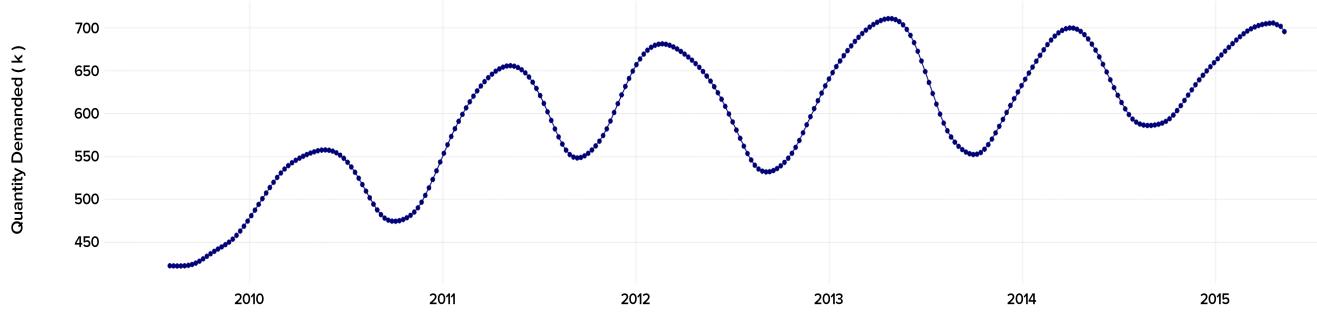
At first sight, it may seem that there is no constant demand value, as the variance goes up and down, making the prediction hardly possible. But, there is a method that can help here.

We used the decomposition method to separately extract trend (the increase or decrease in the series over a period of time), seasonality (the fluctuation that occurs within the series over each week, each month, etc.) and residuals (the data point that falls outside of the expected data range). With these three components we built the additive model:

$$y_t = T_t + S_t + R_t$$

where  $y_t$  is the data,  $T_t$  is the trend-cycle component,  $S_t$  is the seasonal component and  $R_t$  is the residual component, all defined over the time period  $t$ .

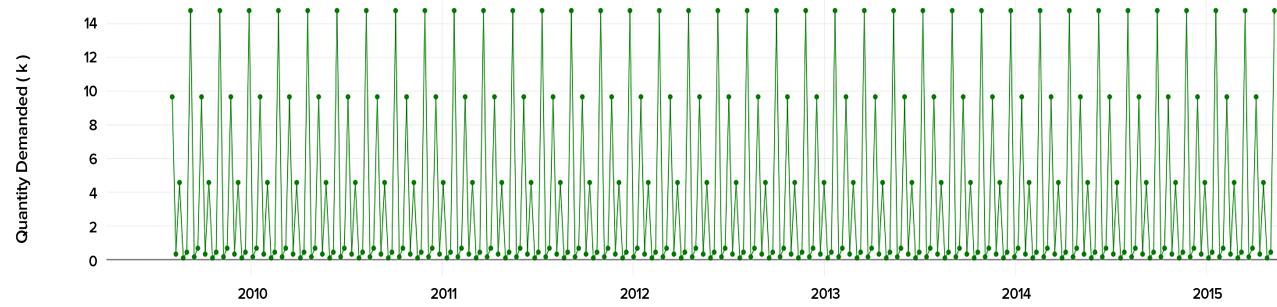
An important first step in describing various components of the series is smoothing, although it does not really provide you with a ready-to-use model. In the beginning, we estimated the trend (behavior) component. Such methods as Moving Average, Exponential Smoothing, Chow's Adaptive Control, Winter's Linear and Seasonal Exponential Smoothing methods did not provide us with the trend estimation accuracy we expected. The most reliable result was obtained using the Hodrick-Prescott Filter (<https://cran.r-project.org/web/packages/mFilter/mFilter.pdf>) technique.



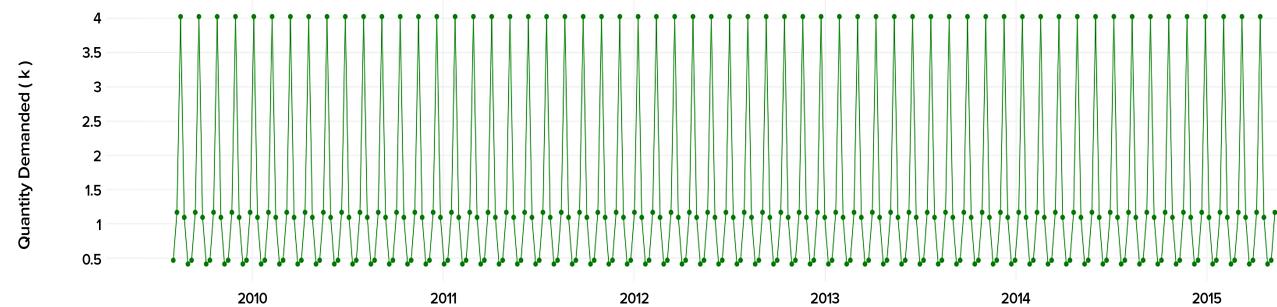
(<https://labs.eleks.com/wp-content/uploads/2016/10/Fig2-The-estimated-trend.png>)

*Fig. 2. The estimated trend*

Then, we defined the seasonality from the available data. This component could change over time, so we applied a powerful tool for decomposing the time series – the Loess method (<https://www.otexts.org/fpp/6/5>). This approach can handle any type of seasonality, and the rate of change can be controlled by a user.



(<https://labs.eleks.com/wp-content/uploads/2016/10/Fig3-Multi-seasonality.png>)



(<https://labs.eleks.com/wp-content/uploads/2016/10/Fig3-Multi-seasonality2.png>)

Fig. 3. Multi-seasonality

We obtained a multi-seasonal component with some high and low variances, causing large fluctuations.

After applying Elastic Net Regression ([http://scikit-learn.org/stable/modules/generated/sklearn.linear\\_model.ElasticNet.html](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html)) and Fourier transformation (<http://www.le.ac.uk/users/dsgp1/COURSES/LEIMETZ/FOURIER.pdf>), we built a forecast for the trend based on the results obtained. The approximation of the trend can be found from the formula below,

$$y_{trend} = P_n(t) + \sum_{\alpha_m \in A_k} \left( a \sin \left( \frac{2\pi\alpha_m t}{T} \right) + b \cos \left( \frac{2\pi\alpha_m t}{T} \right) \right),$$

where  $P_n(t)$  is a degree polynomial and  $A_k$  is a set of indexes, including the first  $k$  indexes with highest amplitudes.

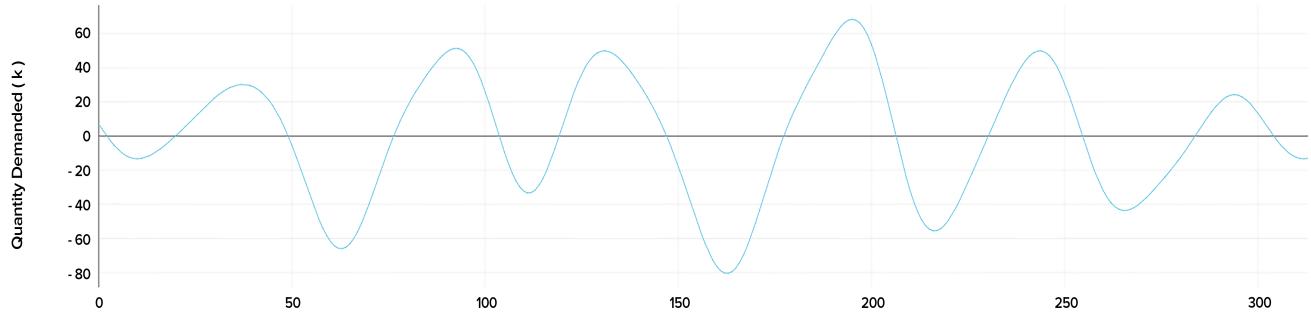
Then, we calculated the Fourier coefficients using The Discrete Fourier Transform (DFT).

```
def fourierExtrapolation(x, n_predict=12, n_harm=12):
    """
    :param x: input series
    :param n_predict: forecasting period
    :param n_harm: number of harmonics in model
    :return: restored series
    """

    n = x.size
    xf = np.fft.fft(x)
    f = np.fft.fftfreq(n)
    indexes = range(n)
    indexes.sort(key=lambda i: np.absolute(f[i]))
    t = np.arange(0, n + n_predict)
    restored_ser = np.zeros(t.size)
    for j in indexes[:1 + n_harm * 2]:
        ampl = np.absolute(xf[j]) / n    # amplitude
        phase = np.angle(xf[j])          # phase
        restored_ser += ampl * np.cos(2 * np.pi * f[j] * t + phase)
    return restored_ser
```

Fig. 4. The example of code of the DFT in Python

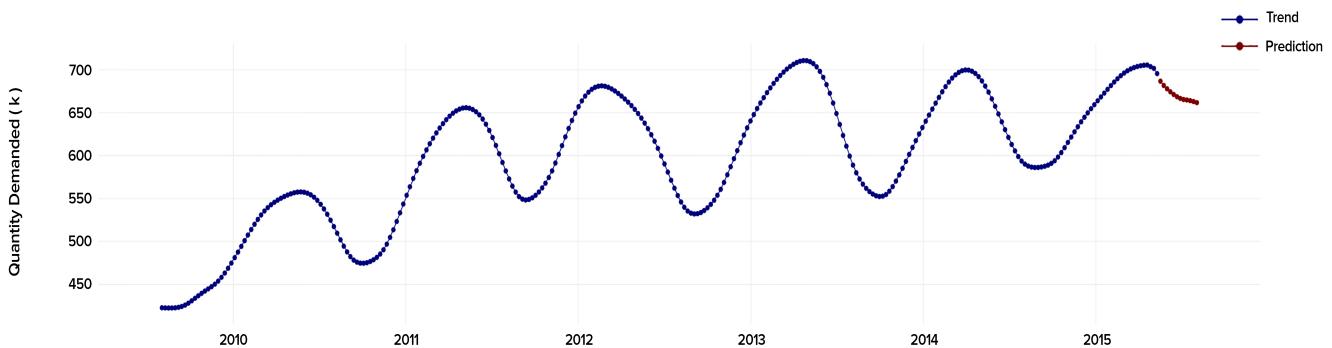
The effect of the Fourier terms, used as external regressors in the model, is visualised below.



(<https://labs.eleks.com/wp-content/uploads/2016/10/Fig5-The-visualised-effect-of-Fourier-terms.png>)

*Fig. 5. The visualised effect of Fourier terms*

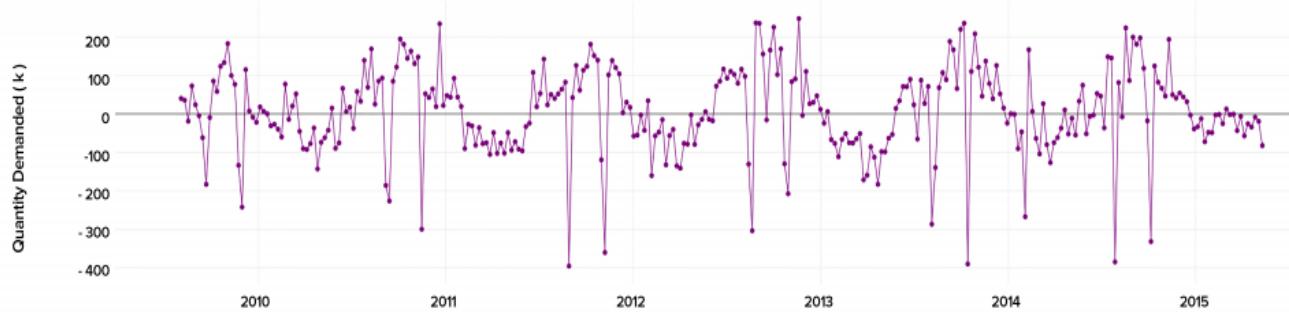
We built the trend prediction using the additive model.



(<https://labs.eleks.com/wp-content/uploads/2016/10/Fig6-Trend-prediction.png>)

*Fig. 6. Trend prediction*

When the trend and seasonal components are removed from the model, we can obtain the residuals (the difference between an observed value and its forecast based on other observations) from the remaining part to validate and fit our mathematical model.

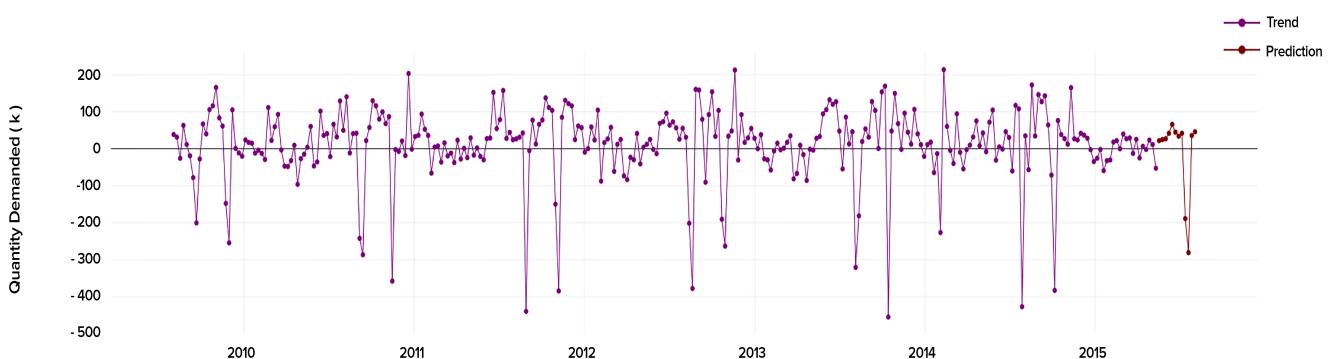


(<https://labs.eleks.com/wp-content/uploads/2016/10/Fig7-Obtained-residuals.png>)

*Fig. 7. Obtained residuals*

You may notice that there are some negative values present, showing that something unusual was happening during that period of time. We aimed to find out the circumstances causing such behaviour, so we came up with an idea to compile the outliers with a simple calendar and discovered that the negative values tightly correlate with such public holidays as Ramadan, Eid Al Fitr and other. Having collected and summarized all the data, we applied Machine Learning methods based on previous data points as entry features and Machine Learning Strategies for Time Series Prediction.

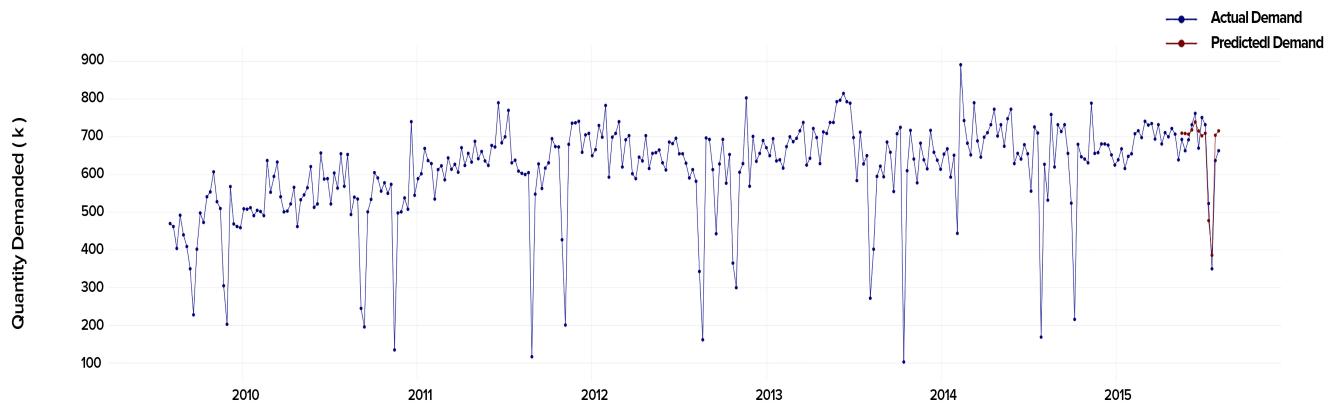
After a few training sessions conducted with ML models, we built a prediction for residuals that can be observed below.



(<https://labs.eleks.com/wp-content/uploads/2016/10/Fig8-Prediction-for-residuals.png>)

*Fig. 8. Prediction for residuals*

As a result, we got a final forecasting model that minimizes the mean absolute percentage error (MAPE) to 6% for one particular city and 10% for the entire country in general.



(<https://labs.eleks.com/wp-content/uploads/2016/10/Fig9The-forecast-at-the-original-scale.jpg>)

*Fig. 9. The forecast at the original scale*

## A 24-times faster prediction? Yes, it's possible

When building our model, we attempted not only to use the available information, but also discover the factors which could affect the results. This approach helped us develop the model generating more accurate forecasting results faster than the existing models. For example, to train the developed model to make a prediction for 300 different cities, we need about 15 minutes, while other methods require about 6 hours.

Also, the fact that the deviation between the actual demand and the predicted demand was only 6% resulted in possibilities to resolve mismatches between supply and demand. Now, the customer can more quickly and more easily plan the capacity, minimize future risks and optimize inventory.

## What's next?

Well, the results are quite promising. And there is a long way we can go further in improvement of this model, so it could provide accurate long-term forecasts as well. As for now, the degree of error for long-term predictions is still quite high. Sounds like a challenge? So stay tuned! Some new experiments are coming!

---

**Taras Firman**

Taras Firman is a Data Scientist at ELEKS, working with statistical analysis, optimization, forecasting and other serious stuff. A mathematician by education and by calling, Taras develops his professional skills at a postgraduate program, and teaches operations research and data analysis. Moreover, he never misses any interesting data science or machine learning news in blogs, and having a lot to contribute, he also decided to blog. When he's done with math and data science, Taras is all into football and music.

---

---

1999-2017 (C) [ELEKS](#), All Rights Reserved. ELEKS is a trusted global organization that provides offshore, full-cycle, software product engineering services, from ideation to finished products. Since 1991, we've been helping businesses innovate and build advanced software products that matter, leveraging technologies such as Data Analytics, the Internet of Things, mobile applications and wearables.

---

