# Walk-Through Demand Sales Time Series Forecasting

**Aditya Kumar**

*UG Student, Dept. of Computer Science Engineering, Chandigarh University, Mohali, India*

---

**Abstract –** *Time Series Forecasting is the scientific prediction of future observations from the historical time-stamped data. It helps an organization to build models for past analysis and create business plans that can result in high revenue in the future. Demand Sales Forecasting is one of the applications of time series where a combination of strategies is used. In Demand forecasting, we make predictive analysis on past historic data to estimate and predict customer demand in the future and the sales of the product. This helps organizations and widely spread businesses to manage their resources, create revenue plans with high gains and risk estimation. Various time series classical models can be used in predicting future demand sales. These pre-defined time series state-of-art-model create a base on which we can do our modifications according to our input data and business plan and get good results. When having seasonality in the input data, models like ARIMA, SARIMA, and SARIMAX can prove to be a better option. The techniques like Moving Averages, Weighted Average, Auto-Regression can also be used while performing feature engineering. This survey shows how these techniques have been used in-demand sales forecasting for a type of input data taken from an external source.*

***Key Words*** **:(Demand Sales Forecast, Time Series, ARIMA, SARIMA, SARIMAX, Moving Average, Revenue, Data Leakage, Mean Absolute Error, Symmetric Mean Absolute Error)**

## 1. INTRODUCTION

*Time Series Forecasting is predicting the future from past data. Time Series Data is being used in various realistic domains such as stock price forecasting, weather forecasting, invoice delay forecasting, business planning, and many others. Time-series Forecasting is a subset of supervised regression problems, where we study and perform analysis of temporal features. The proper study of time series data is very important to find changes in trends and seasonality in data. Demand sales forecasting is one of the applications where time-series data is used.*

*We are given past sales data in the Demand Sales forecast, and we have to find future 2-3 month sales. This type of forecasting helps different organizations to make a decision based on facts and human factors about everything from resource planning to running flash sales. Demand Forecasting helps an organization to have an estimate of the total sales and revenue for a future period.*

### 1.1 WHAT IS TIME SERIES?

A time series is a system in which we have to make observations at certain times, and the outcome, the observed value at each time which is a random variable. A time series consists of multiple assessments of a specific outcome measure, at group level, or at regular differenced time intervals. The "interruption" or "change" or "seasonality" in the time series data is an important identifiable and real-world event that needs to be analyzed and monitored. In forecasting, we are at theta(t) at time t and we are interested in knowing the outcome theta (t +h) at time t +h using only information available at time t.

## 2. WHAT IS DEMAND FORECASTING?

Demand forecasting is the process of using predictive analysis [1] of past data to estimate and predict customers' future demand and sales for the company or organization. It helps the business make better-informed decisions that estimate the total sales and profit for a future period. Through Demand Forecasting, different business and organization can optimize their resources and make business plans that will lead to better revenue estimates in the future. The uniqueness in studying demand forecasting time series data is that you have limited features from which the estimation has to be made. A target variable which is sales at time t is given and is of most importance to us. We are having a feature value of sales at time t, given as S(t). We have to forecast sales for S (t +h). In demand forecasting, the Date feature which is being denoted as t has a lot of hidden information stored. A deep analysis of sales concerning date is required to have insights regarding the changing trends and seasonality of data.

### 2.1 IMPORTANCE OF DEMAND FORECASTING

Without demand, there is no business. If a business doesn't have a proper understanding of demand, they aren't capable of making informed decisions and proper business plans that will lead to higher resource usage and less revenue in the future. Demand forecasting is not 100% accurate, but it gives business organizations an estimate and approximation regarding the revenue that they will be having in the future "t" month period if they follow a current business plan.

These estimations help the organization to make amendments to the current business plans so that they can increase their revenue in the future. It helps to improve production lead times, resource usage, and cost reduction.

Demand Forecasting also helps to reduce the risk factor and make efficient financial informed decisions, profit and cost margins, and allocation of resources. It lets you analyze the products that your customers want and when they want them.

**2.2 LEVELS OF DEMAND FORECASTING**

Demand Forecasting is done at various levels-:

- **Short-Term Demand Forecasting-** It is carried out for a range of periods between 3 months to 12 months. The seasonal pattern and the effect of statistical decisions on customer demand are taken into consideration.

- **Medium to Long-Term Demand Forecasting-** It is carried out mainly for a period of more than 12 months to 24 months which could sometimes increase to 36-48 months in certain businesses and organizations.

- **Macro-Level Demand Forecasting-** This type of forecasting deals with the macroeconomic movements in the market which is widely spread.

- **Passive Demand Forecasting-** This type of forecasting is generally carried out for stable business that aims to grow invested capital over the long term.

- **Active Demand Forecasting-** This type of forecasting is generally carried out for large business that seeks to return the highest capital gains.

- **Internal Business Level Demand Forecasting-** It is carried out for businesses such as product category, sales, etc. The annual sales forecast is one of the examples that fall in this category.

**2.3 CLASSICAL METHODS OF TIME SERIES USED IN-DEMAND SALES FORECASTING**

- **Auto-Regression (AR)-** This method states itself as a next step in the sequence as a linear function of the observation at past time steps. [2] It is a regression of the variable against itself. The method works for univariate time series without trend and seasonal components.

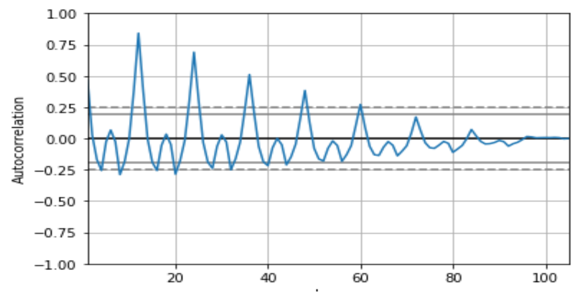
$$h(t)= \beta+ \phi 1h(t-1) +\phi 2h(t-2) + .... + \phi xh(t-x) +£t,$$

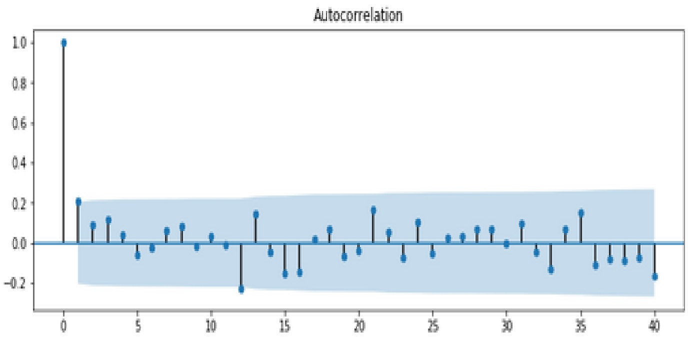
*where £t is the white noise and* β is a constant.

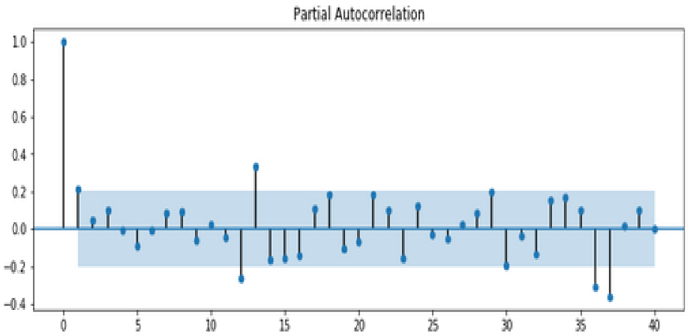
ϕ1, ϕ2, …… ϕx are the autoregressive coefficients.

These models are excellent at handling a wide range of different time-series patterns.

The identification of the AR model is best done with a Partial Autocorrelation Factor. For an AR model, the theoretical PACF pushes off the past order of the model. Here push-off means PACF's are equal to 0 beyond that point. In another way, the number of non-zero partial autocorrelations gives the order of the AR model. By the “order of the model” we mean the most extreme lag of p that is used as a predictor.




- **Moving Averages (MA)-** This method states itself as a next step in the sequence as a linear function of the residual errors from a mean process at past time steps. [2] A series of the mean of different subsets of the entire dataset is calculated for the analysis. A top-down sliding window is made of fixed items for which the average is calculated by dropping the top items of the previous window and adding new items from the below. It is also called Rolling Mean(RM) or Moving Mean(MM). If the dataset contains *n* entries with X1, X2, X3, . . . ,

$X_n$, then the moving average of over the last m entries is calculated as -:

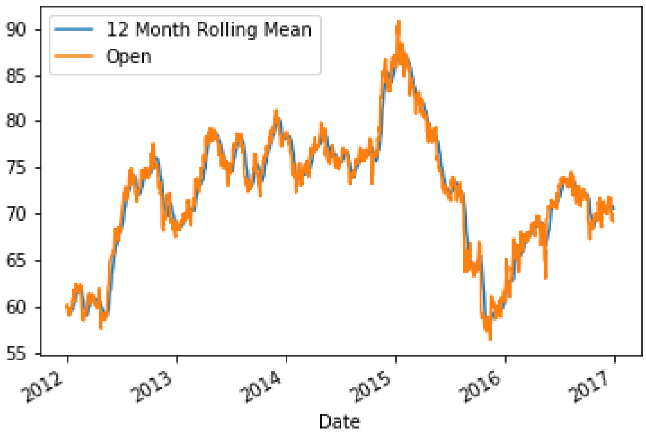
$MA(m) = 1/m \sum X_i$, where $i = n-m+1$ to n.

Now when calculating the next average, the same sampling width of m is taken which will range from n-m+2 to n+1.

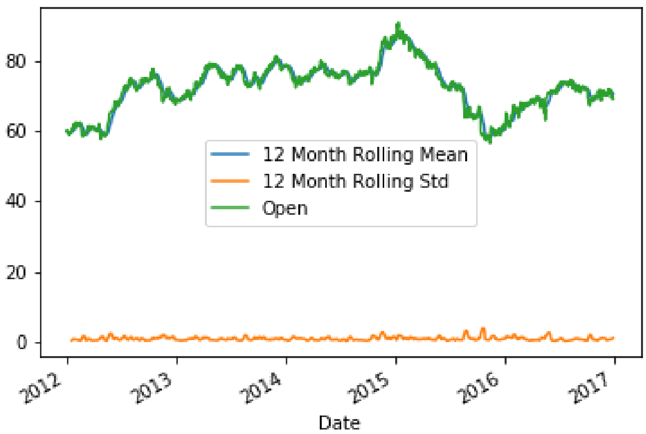
$MA(m, next) = 1/m \sum X_i$, where $i = n-m+2$ to $n+1$ .

On expansion,

$MA(m, next) = MA(m, prev) + 1/m [X_{n+1} - X_{n+k-1}]$



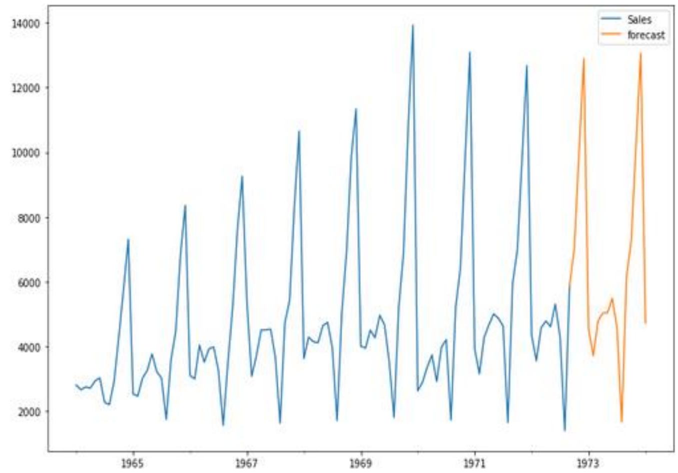

Rolling Mean Example 1.1



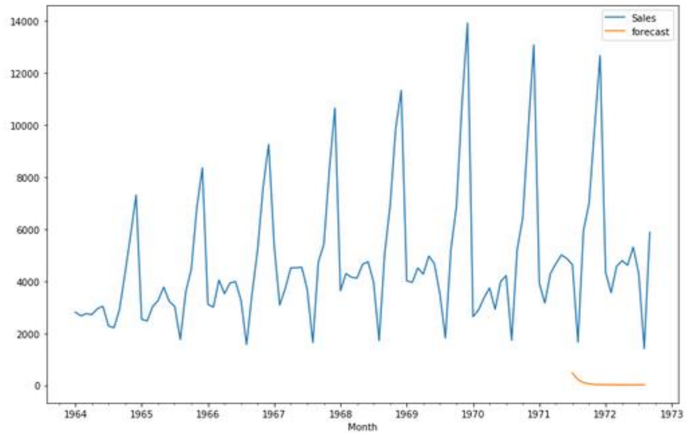

Rolling Mean with Rolling Standard Deviation Example 1.2

- **Autoregressive Integrated Moving Average (ARIMA) and Seasonal Autoregressive Integrated Moving Average (SARIMA) –** This method states itself as the next step in the sequence as a linear function of the differenced observation and residual errors at prior time steps. It is a combination of AR and MA models as well as the differencing pre-processing step of the sequence to make it stationary. The ARIMA model can be imported from ***statsmodel.tsa.arima_model*** and SARIMA model by importing ***statsmodel.api*** in Python. The general purpose of the ARIMA models is -:

1. Visualize time series data.
2. Make the time series data stationary.
3. Plot the correlation and autocorrelation charts.
4. Construct ARIMA and Seasonal ARIMA models based on the data.
5. Use the model to make predictions.




Seasonal ARIMA Extended forecast




ARIMA forecast

### 2.4 FEATURE ENGINEERING

Various methods can be used to perform feature engineering when working on-demand sales forecasting datasets. But the correct method will be one that can lead you to the right analysis and less error.

| | date | store | item | sales |
|---|---|---|---|---|
| 0 | 2013-01-01 | 1 | 1 | 13 |
| 1 | 2013-01-02 | 1 | 1 | 11 |
| 2 | 2013-01-03 | 1 | 1 | 14 |
| 3 | 2013-01-04 | 1 | 1 | 13 |
| 4 | 2013-01-05 | 1 | 1 | 10 |

The temporal feature like date plays a very important role in this case. Normally, we shouldn't hesitate to generate many features even if we think that it will not have a point. It's better to generate as much as we can and then by looking 'feature_importance' plot we can eliminate some of them that we see that they are not useful features for the model. The weekdays can influence sales. The study of such features is important for correct analysis and estimation of future sales. There are cases where sales are seasonal for some product items. For example, products only bought by customers in summers have fewer chances that they will be bought in winter too. Such trends and seasonal factors are very important to be taken care of. So decomposition of date into a month, year, day, and weekday is required.

- **Lag/Shift Feature-** Adding lag features that will tell 1 day before sales is also a technique that we can use. Since we are creating these features from the target variable which is sales, we need to add noise too to avoid overfitting and data leakage. The reason for that **data leakage** problem is that in our case, normally we shouldn't generate features by using target variables when we are working on an ML project. Because it causes **overfitting** to the train data. Model notices target variable base features explains "target" column well and focuses more on that columns. Consequently, it loses its "generalization" ability.

  Since we feel obliged to generate features from the target column in our case (because we have only this feature in our hands and from statistical time series models like ARIMA uses that values for forecasting and we also have to use to represent time series patterns), for avoiding "overfitting" situation, as a solution we add "Random, Gaussian noise" to "Lag/Shifted Features" on purpose. This will cause, the model not to learn the exact values of the target variable and as a result, we avoid an "overfitting" situation

| | sales | lag1 | lag2 | lag3 | lag4 |
|---|---|---|---|---|---|
| 0 | 13 | NaN | NaN | NaN | NaN |
| 1 | 11 | 13.0 | NaN | NaN | NaN |
| 2 | 14 | 11.0 | 13.0 | NaN | NaN |
| 3 | 13 | 14.0 | 11.0 | 13.0 | NaN |
| 4 | 10 | 13.0 | 14.0 | 11.0 | 13.0 |
| 5 | 12 | 10.0 | 13.0 | 14.0 | 11.0 |
| 6 | 10 | 12.0 | 10.0 | 13.0 | 14.0 |
| 7 | 9 | 10.0 | 12.0 | 10.0 | 13.0 |
| 8 | 12 | 9.0 | 10.0 | 12.0 | 10.0 |
| 9 | 9 | 12.0 | 9.0 | 10.0 | 12.0 |

As we are predicting sales for the future 3 months in our case, we have created a list for which lag features will be created [91, 98, 105, 112, 119, 126, 182, 364, 546, 728].

As we are creating more than 1 "Lag/Shifted Features". So this list represents all "Lag/Shifted Features" that will be created. We started with a '90' days gap because we are aiming to predict the "test" set in the final stage. Our ultimate goal is to be successful in the "test" set. So if we don't start the '90' days gap, the majority of our new generated lag columns will be 'NaN'.

| sales_lag_91 | sales_lag_98 | sales_lag_105 | sales_lag_112 | sales_lag_119 | sales_lag_126 |
|---|---|---|---|---|---|
| NaN | NaN | NaN | NaN | NaN | NaN |
| NaN | NaN | NaN | NaN | NaN | NaN |
| NaN | NaN | NaN | NaN | NaN | NaN |
| NaN | NaN | NaN | NaN | NaN | NaN |
| NaN | NaN | NaN | NaN | NaN | NaN |

Lag/Shift Features for first five rows in the dataset

| sales_lag_91 | sales_lag_98 | sales_lag_105 | sales_lag_112 | sales_lag_119 | sales_lag_126 | sales_lag_182 |
|---|---|---|---|---|---|---|
| 78.749209 | 70.436030 | 82.666390 | 79.734337 | 78.531102 | 104.787398 | 90.539611 |
| 83.286495 | 91.452435 | 87.410392 | 78.798285 | 88.802146 | 88.225447 | 121.684762 |
| 87.684363 | 96.601679 | 93.293318 | 98.004843 | 97.987213 | 97.950486 | 110.741330 |
| 101.001542 | 97.159184 | 101.838894 | 93.173178 | 93.056028 | 92.067970 | 120.473773 |
| 99.344188 | 98.507640 | 107.873938 | 98.199088 | 101.931610 | 115.242301 | 121.202208 |

Lag/Shift Features for last five rows in the dataset

For instance, if we intend to create the 'lag1' feature, this feature will be almost 'NaN' for the 'test' data, and we probably will not be successful in the test set, since this column will be almost 'NaN' for the 'test' column. Only one observation (the observation that comes after the last observation of the 'train' set because only for that observation do we have the 'lag1' feature) will not be 'NaN' others will be. So for that reason, we start with a '90' days gap. In this case, we will not have any 'NaN' in the "Lag/Shifted Features" in the 'test' set.

We add more than 1 "Lag/Shifted Features" because we will try which "Lag/Shifted Features" makes sense for our data, which means you can try other values in this list and see the effect.

We need such features as we have to generate some features that represent "Time series" patterns. By using "Lag/Shifted Features" we add that kind of feature to our data. Because if you are familiar with well-known "Time Series" statistical models like ARIMA, Holt-Winters, etc, in almost all these models' formulas we see that "Lag/Shifted Features" actually is used.

$$h(t) = \beta + \phi 1 h(t-1) + \phi 2 h(t-2) + \ldots + \phi x h(0) + \pounds t,$$

Here h(t-1), h(t-2) are nothing but lag features.

- **Rolling Mean Features-** "Moving Average Method" is used for forecasting "Time Series" problems. This method simply takes "n" previous target variables and averages them and returns them as a new value.

  So since we know that, this kind of method is used for forecasting "Time Series" problems, again we generate a new feature by using that method.

  You may ask, why we use traditional "Time Series Forecasting methods" when we generate new features. Because normally when we work on Machine Learning problems we try to generate features that we think these new features can predict target variables.

  Since this kind of traditional method has been used for forecasting target variables, when we want to generate new features by looking at these methods we become sure that these new features will have predictive ability for predicting target variables. Because they have been used in the traditional "Time Series Method", that means they have a predictive ability for target variables.

  So since we said that while using the ML approach we have to generate features that represent time series patterns, we act help from traditional methods for that purpose.

We try to add our variety of predictions for target variables to the columns as a new feature by using the traditional "Time Series Methods" approach.

So since these varieties of predictions may cause overfitting to training data we add again random, Gaussian noise to these new generated features on purpose.

| | sales | roll2 | roll3 | roll5 |
|---|---|---|---|---|
| 0 | 13.0 | NaN | NaN | NaN |
| 1 | 11.0 | NaN | NaN | NaN |
| 2 | 14.0 | 12.0 | NaN | NaN |
| 3 | 13.0 | 12.5 | 12.666667 | NaN |
| 4 | 10.0 | 13.5 | 12.666667 | NaN |
| 5 | 12.0 | 11.5 | 12.333333 | 12.2 |
| 6 | 10.0 | 11.0 | 11.666667 | 12.0 |
| 7 | 9.0 | 11.0 | 10.666667 | 11.8 |
| 8 | 12.0 | 9.5 | 10.333333 | 10.8 |
| 9 | 9.0 | 10.5 | 10.333333 | 10.6 |

| sales_roll_mean_365 | sales_roll_mean_546 |
|---|---|
| 88.501497 | 85.213259 |
| 90.742862 | 87.007006 |
| 85.698984 | 85.710129 |
| 87.104656 | 89.584973 |
| 87.169300 | 83.586291 |

Rolling Mean Feature

- **Exponentially Weighted Mean Feature-** Another traditional "Time Series Method" is the "Exponentially Weighted Mean" method. This method has a parameter called *alpha* used as a smoothing factor. This parameter ranges between [0, 1]. If *alpha* is close to 1 while taking the average for last for instance 10 days (rolling mean features also was taking averages but without giving weight), it gives more *weight* to the close days and decreases the *weight* when going to more past days.

You can read about this method more on the internet, but briefly normally in time series forecasting, it's better to give more *weight* to the more recent days rather them giving the same *weight* to all past days.

Because more recent days have more influence to the current day. Therefore, giving more *weight* to the more recent days makes sense.

This method uses that formula behind in its calculations (xt : past days values)

$$y_t = \frac{x_t + (1-\alpha)x_{t-1} + (1-\alpha)^2 x_{t-2} + \ldots + (1-\alpha)^t x_0}{1 + (1-\alpha) + (1-\alpha)^2 + \ldots + (1-\alpha)^t}$$

As we see when it goes more past values it decreases the *weight.*

| | sales | roll2 | ewm099 | ewm095 | ewm07 | ewm01 |
|---|---|---|---|---|---|---|
| 0 | 13.0 | NaN | NaN | NaN | NaN | NaN |
| 1 | 11.0 | NaN | 13.000000 | 13.000000 | 13.000000 | 13.000000 |
| 2 | 14.0 | 12.0 | 11.019802 | 11.095238 | 11.461538 | 11.947368 |
| 3 | 13.0 | 12.5 | 13.970201 | 13.855107 | 13.287770 | 12.704797 |
| 4 | 10.0 | 13.5 | 13.009702 | 13.042750 | 13.084686 | 12.790637 |
| 5 | 12.0 | 11.5 | 10.030097 | 10.152137 | 10.920146 | 12.109179 |
| 6 | 10.0 | 11.0 | 11.980301 | 11.907607 | 11.676595 | 12.085878 |
| 7 | 9.0 | 11.0 | 10.019803 | 10.095380 | 10.502722 | 11.686057 |
| 8 | 12.0 | 9.5 | 9.010198 | 9.054769 | 9.450748 | 11.214433 |
| 9 | 9.0 | 10.5 | 11.970102 | 11.852738 | 11.235259 | 11.342672 |

### 2.5 CUSTOM COST FUNCTION

- **MAE (mean absolute error)-** is the absolute value of the difference between the forecasted value and the actual value. It tells us how big an error can we expect from the forecast on average. MAE= 1/n * ∑|yi-xi| where i=1 to n.

  yi= predicted value, xi= actual value, n= total number of data points.

- **MAPE (mean absolute percentage error)-** It represents accuracy as a percentage of error. As it is a percentage, it is better to understand than the other accurate statistics. MAPE= 1/n * ∑|(At-Ft)/At| where i=1 to n.

  At=Actual Value, Ft= Forecast Value, n= number of time summation iteration happens.

- **SMAPE (symmetric mean absolute percentage**) error (adjusted MAPE). It is a relative percentage error.

  SMAPE = 2/n * ∑|(Ft- At)/(At+Ft)| where i=1 to n.

### 2.6 HOW TO VALIDATE TIME SERIES MODEL?

We have to define a validation set for optimizing our model. Normally in typical ML projects, we define validation set either 'Hold-out validation set' approach or 'K-fold cross validation' approach. Since it's a time series problem and the order of the time matters 'K-fold cross validation' approach will not be useful. So we had used Hold-Out Approach. For this approach to work, we have to create a validation set similar to the test set, as our ultimate goal will be assessed in the test set. As in our case we are forecasting sales for the beginning 3 months of the next year (2019 as in the dataset). We have taken the first 3 months of the previous year (2018 as in the dataset) as a validation set.

## 3. CONCLUSION

As, a result of this analysis, out of all the features created, the rolling mean feature and lag feature had higher feature importance. Also the year, weekday, and weekend days also affected our forecasting but not by much percentage. In our case, the SMAPE was found to be 11.234. It can be further reduced by creating a more refined and tuned machine learning model, which can be done by cross-validation and hyper-parameter tuning. There are a lot of combination strategies other than what we have used, that can be implemented in demand forecasting problems. The strategy to be chosen and to be used best depends upon the input data and the business goal, but you can use the pre-defined state-of-the-art approaches to create your own business.

## REFERENCES

[1] Seyedan, M., Mafakheri, F. Predictive big data analytics for supply chain demand forecasting: methods, applications, and research opportunities. J Big Data **7,** 53 (2020). https://doi.org/10.1186/s40537-020-00329-2M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[2] Athiyarath, S., Paul, M. & Krishnaswamy, S. A

Comparative Study and Analysis of Time Series Forecasting Techniques. SN COMPUT. SCI. **1,** 175 (2020).

Elissa, "Title of paper if known," unpublished.

[3] Jan G. De Gooijer, Rob J. Hyndman,25 years of time series forecasting, International Journal of Forecasting, Volume 22, Issue 3, 2006, Pages 443-473, ISSN 0169 2070.

[4] Ghalehkhondabi, I., Ardjmand, E., Weckman, G.R., et al. An overview of energy demand forecasting

methods published in 2005–2015. Energy Syst **8,** 411–447 (2017).

**BIOGRAPHIES**

Aditya Kumar is a grounded and solution-oriented Data Scientist, pursuing a Bachelor of Engineering in Computer Science at Chandigarh University. Efficient in Performing Exploratory Data Analysis and Feature Engineering, data modeling as well as in scripting languages like python.