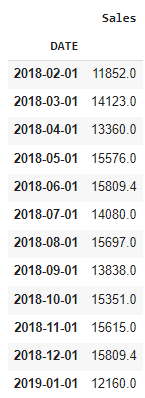
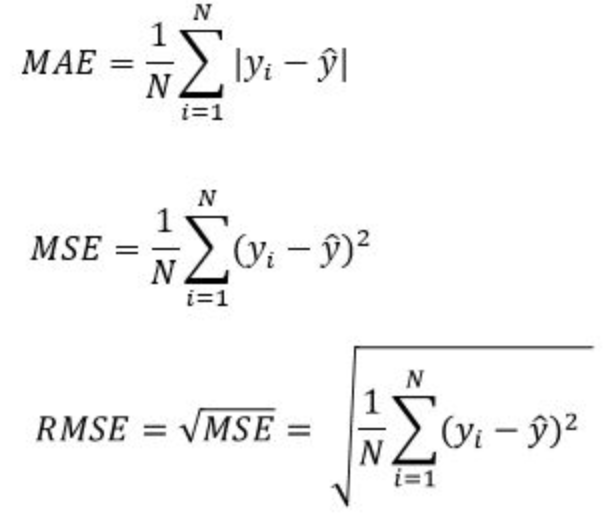
# **Train test split**

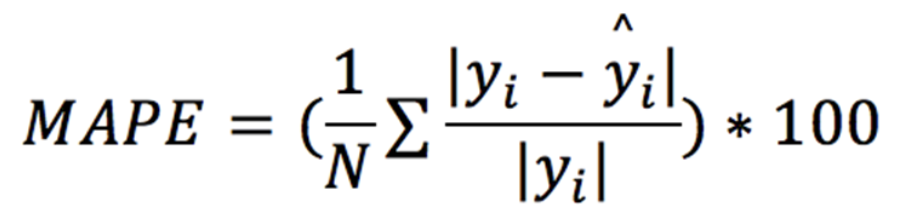
* For time series data, we do not perform a random shuffle for train and test data split.
* Random shuffling on time series data would be meaningless since it is a collection of data over time and we look at the past data to predict future values.
* Therefore, we do a **time-based splitting**.
* In the train-test split we hold out the most recent data to be as the test data.
* If there is seasonality present in the data atleast two full seasons of data are to be taken as the test data.
* For example, for our sales dataset, out of the 18 years of data, we decide to train on 17 years of data and use the last year, i.e. the last 12 values for test data.



# **Measures of Forecast accuracy**

* **MAE** (Mean absolute error) represents the difference between the original and predicted values extracted by averaging the absolute difference over the data set.
* **MSE** (Mean Squared Error) represents the difference between the original and predicted values extracted by squaring the average difference over the data set.
* **RMSE** (Root Mean Squared Error) is the error rate by the square root of MSE.
* **MAPE**(Mean Absolute Percentage Error) is the mean or average of the absolute percentage errors of forecasts. Error is defined as the actual or observed value minus the forecasted value.



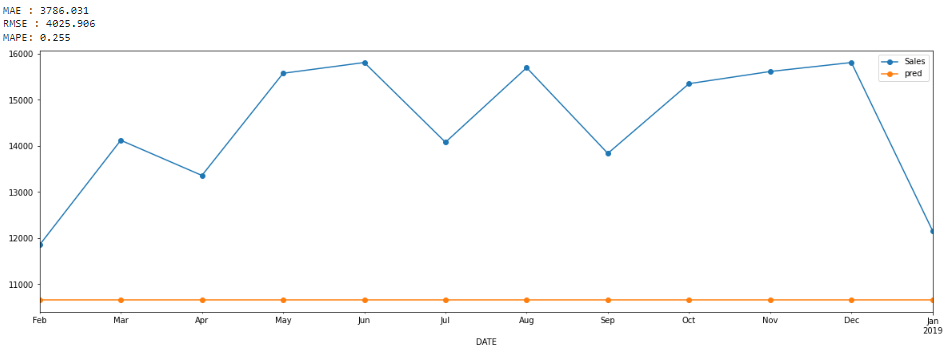


# **Simple Forecast Methods (not-so-intelligent approaches)**

## **Mean forecast**

* We take the average of all past *k* values of the time series to forecast the value at time *t=k+1*.

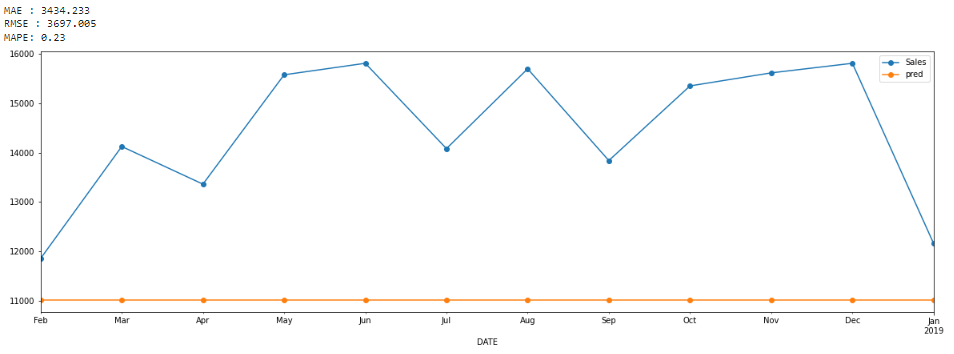




* The MAPE metric is showing that we have a 25.5% error, with respect to its own value. Thus, it is not a good model.
* A similar argument can be made for the median. So, we can rule out using the median of the entire data as future forecast values.
* This method gives equal importance to each observation.

## **Naive Approach**

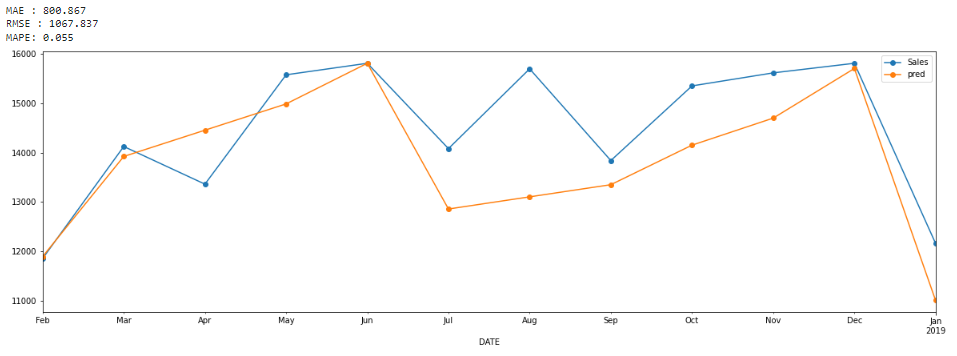
In this approach, we simply take the value of the series at time *t=k* and forecast that for the future.



* This model's performance is slightly better(23%), but we can see it's still really bad.
* But if we get a different value for series at t=k, this would change all the future forecasts we had made earlier. Hence, this is not a very intelligent approach.
* These forecasts are based only on the past values of the variable
* Thus forecast is optimal when data follow a random walk.

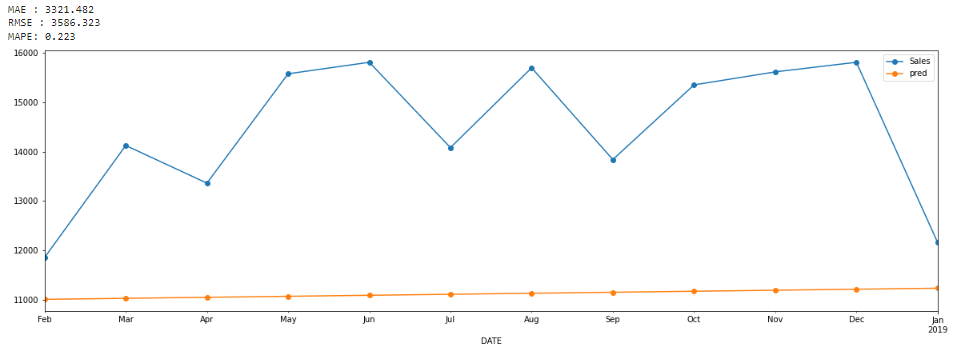
## **Seasonal Naive Forecast**

* It is a smarter approach to optimize the Naive approach.
* We set each forecast to be equal to the last observed value from the same season (e.g., the same month of the previous year).
* This way, we essentially forecast the future values to be exactly the same as last season.



## **Drift Method**

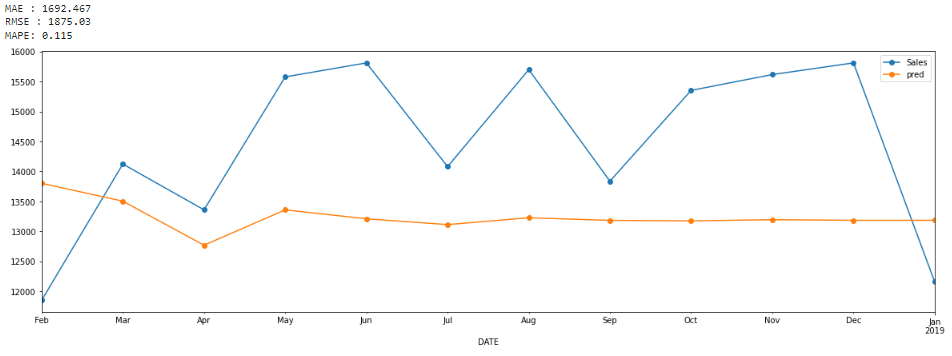
* A variation on the naïve method is to allow the forecasts to increase or decrease over time, where the amount of change over time (called the **drift**) is set to be the average change seen in the historical data.
* In this method of forecasting, we take the first and last point of the data and draw a straight line between those points, and then extend this line into the future to get a forecast (**Linear extrapolation**).
* So, instead of just picking up some value from the past, we let our values increase or decrease over time.
* Depending on what the last point is, the drift (slope) may change significantly. thus, this is highly sensitive to the last value available.



## **Moving average**

* The average of the last k data points in our series is used to forecast the next point in this approach.
* We cannot consider the data points outside the window.
* For our sales data, our model has an 11.5 % error when using this model.

This is much better than the other simple models we saw.



* Moving average as a forecast can give misleading results when there is underlying seasonality present in the data.
* These were some simple models that do not give any smart forecasts.

# **Smoothing-based methods**

## **Exponential Smoothing**

* The exponential smoothing technique is a weighted moving average procedure where the exponential decline of weights happens as the data becomes older.
* More weightage is given to the recent observations and less weightage is given to the past/old observations.
* This method overcomes the shortcomings of the moving average method.
* The smoothing parameters control how fast the weights decay and these parameter values lie between 0 and 1.
* There are three types of exponential smoothing methods:
  + - SES
    - DES
    - TES

### **Simple Exponential Smoothing (SES)**

* The key idea of this method is to keep some memory of the entire time series, but also, we want to give more value to the recent data and less value to the past value. This forms a decaying trend.
* This method is used when there is no trend or seasonality present in the data.
* Let's consider the weight we assign to the recent most value be .

is called the **smoothing parameter**.

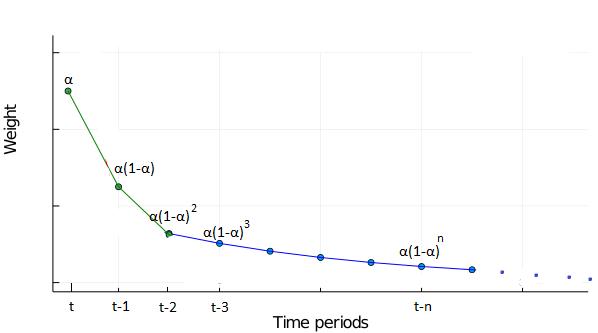
So, our forecast at time *t* for the time t+1 is:



* The above formulation is recursive in nature and expands in the following form:



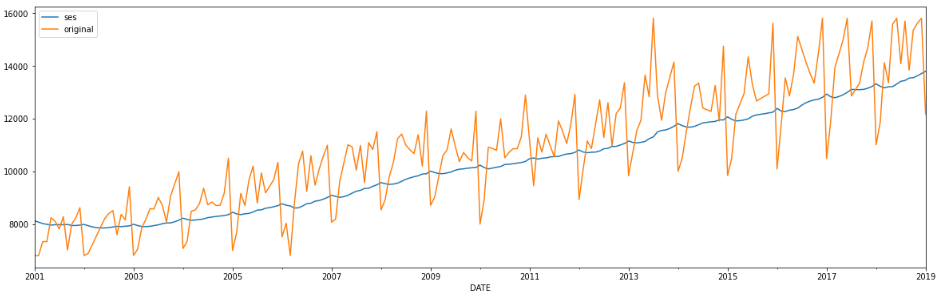
* We can observe from the formulation that the weights are exponentially decaying. Therefore, we give more weightage to the most recent values, and this weightage keeps decreasing for earlier values.



* The recommended starting value of is:

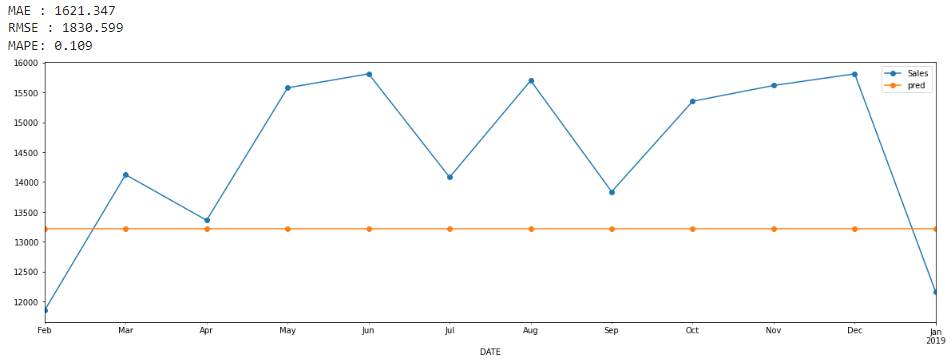


* Let’s fit the SES model on the sales data:



Unlike the moving averages, it does not have the offset at the beginning and end, because this method is initialized properly.

* The forecast plot of this model on the sales test data is given below:



* The advantage of the above forecast is that the **level** of the forecasted values is right.
* However, the forecast is a completely straight line This is because we don't have the previous actual value available for horizon > 1. So the current forecast is used for all the next values.
* The prediction is a straight line, but the error is 10% which is less than the error of the moving average.
* The higher the value of  **(i.e, nearer to 1)** the forecast becomes more sensitive to the latest observations.
* The lower the value of  **(i.e, nearer to 0)** the forecast will be less sensitive to the latest observations.
* Disadvantage of this model is that it is missing both trend and seasonality
* Now, we have the right levels, if we can predict the trend and seasonality right, we should get a good forecast.