**Types of time series models**

Now that you have seen the main specificities of time series data, it is time to look into the types of models that can be used for predicting time series. This task is generally referred to as forecasting.

**Classical time series models**

Classical time series models are a family of models that have been traditionally used a lot in many domains of forecasting. They are strongly based on temporal variation inside a time series and they work well with univariate time series. Some advanced options exist to add external variables into the models as well. These models are generally only applicable to time series and are not useful for other types of machine learning.

**Supervised models**

Supervised models are a family of models that are used for many machine learning task. A machine learning model is supervised when it uses clearly defined input variables and one or more output (target) variables.

Supervised models can be used for time series, as long as you have a way to extract seasonality and put it into a variable. Examples include creating a variable for a year, a month, or a day of the week, etc. These are then used as the X variables in your supervised model and the ‘y’ is the actual value of the time series. You can also include lagged versions of y (the past value of y) into the X data, in order to add autocorrelation effects.

**Deep learning and recent models**

The increasing popularity of deep learning over the past years has opened new doors for forecasting as well, as specific deep learning architectures have been invented that work very well on sequence data.

Cloud computing and the popularization of AI as a service have also provided a number of new inventions in the domain. Facebook, Amazon, and other big tech companies are open-sourcing their forecasting products, or making them available on their cloud platforms. The availability of those new “black- box” models gives forecasting practitioners new tools to try and test, and can sometimes even beat previous models.

**Going deeper into classical time series models**

In this part, you will discover classical time series models in depth.

**ARIMA family**

The ARIMA family of models is a set of smaller models that can be combined. Each part of the ARMIA model can be used as a stand-alone component, or the different building blocks can be combined. When all of the individual components are put together, you obtain the SARIMAX model. You will now see each of the building blocks separately.

**1. Autoregression (AR)**

Autoregression is the first building block of the SARIMAX family. You can see the AR model as a regression model that explains a variable’s future value using its past (lagged) values.

The order of an AR model is denoted as p, and it represents the number of lagged values to include in the model. The simplest model is the AR(1) model: it uses only the value of the previous timestep to predict the current value. The maximum number of values that you can use is the total length of the time series (i.e. you use all previous time steps).

**2. Moving average (MA)**

The Moving Average is the second building block of the larger SARIMAX model. It works in a comparable way to the AR model: it uses past values to predict the current value of the variable.

The past values that the Moving Average model uses are not the values of the variable. Rather, the Moving Average uses the prediction error in previous time steps to predict the future.

This sounds counter-intuitive, but there is a logic behind it. When a model has some unknown but regular external perturbations, your model may have a seasonality or other pattern in the error of the model. The MA model is a method to capture this pattern without even having to identify where it comes from.

The MA model can use multiple steps back in time as well. This is represented in the order parameter called q. For example, an MA(1) model has an order of 1 and uses only one time step back.

**3. Autoregressive moving average (ARMA)**

The Autoregressive Moving Average, or ARMA, model combines the two previous building blocks into one model. ARMA can therefore use both the value and the prediction errors from the past.

ARMA can have different values for the lag of the AR and MA processes. For example an ARMA(1, 0) model has an AR order of 1 ( p = 1) and an MA order of 0 (q=0). This is actually just an AR(1) model. The MA(1) model is the same as the ARMA(0, 1) model. Other combinations are possible: ARMA(3, 1) for example has an AR order of 3 lagged values and uses 1 lagged value for the MA.

**4. Autoregressive integrated moving average (ARIMA)**

The ARMA model needs a stationary time series. As you have seen earlier on, stationarity means that a time series remains stable. You can use the Augmented Dickey-Fuller test to test whether your time series is stable and apply differencing if it is not the case.

The ARIMA model adds automatic differencing to the ARMA model. It has an additional parameter that you can set to the number of times that the time series needs to be differenced. For example, an ARMA(1,1) that needs to be differenced one time would result in the following notation: ARIMA(1, 1, 1). The first 1 is for the AR order, the second one is for the differencing, and the third 1 is for the MA order. ARIMA(1, 0, 1) would be the same as ARMA(1, 1).

**5. Seasonal autoregressive integrated moving-average (SARIMA)**

SARIMA adds seasonal effects into the ARIMA model. If seasonality is present in your time series, it is very important to use it in your forecast.

SARIMA notation is quite a bit more complex than ARIMA, as each of the components receives a seasonal parameter on top of the regular parameter.

For example, let’s consider the ARIMA(p, d, q) as seen before. In SARIMA notation, this becomes SARIMA(p, d, q)(P, D, Q)m.

m is simply the number of observations per year: monthly data has m=12, quarterly data has m=4 etc. The small letters (p, d, q) represent the non-seasonal orders. The capital letters (P, D, Q) represent the seasonal orders.

**6. Seasonal autoregressive integrated moving-average with exogenous regressors (SARIMAX)**

The most complex variant is the SARIMAX model. It regroups AR, MA, differencing, and seasonal effects. On top of that, it adds the X: external variables. If you have any variables that could help your model to improve, you could add them with SARIMAX.

**Smoothing**

Exponential Smoothing is a basic statistical technique that can be used to smoothen out time series. Time series patterns often have a lot of long term variability, but also short term (noisy) variability. Smoothening allows you to make your curve smoother so that long term variability becomes more evident and short term (noisy) patterns are removed.

This smooth version of the time series can then be used for analysis.

**1. Simple moving average**

The **simple moving average** is the simplest smoothing technique. It consists of replacing the current value by the average of the current and a few  past values. The exact number of past values to take into account is a parameter. The more values you use, the smoother the curve will become. At the same time, you will lose more and more variation.

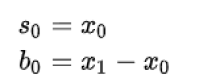
**2. Simple exponential smoothing (SES)**

**Exponential smoothing** is an adaptation of this simple moving average. Rather than taking the average, it takes a weighted average of past values. A value that is further back will count less and a more recent value will count more.

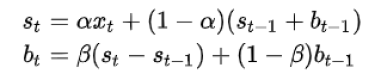
**3. Double exponential smoothing**

When trends are present in your time series data, you should avoid using Simple Exponential Smoothing: it does not work well in this case, as the model cannot make the distinction between variation and trend correctly. However, you can use **double exponential smoothing**.

In DES, there is a recursive application of an exponential filter. This allows you to remove trend problems. This works using the following formulas for time zero:



and the following formulas for subsequent time steps:



In which alpha is the data smoothing factor and beta is the trend smoothing factor.

**4. Holt Winter’s exponential smoothing (HWES)**

If you want to go even further, you can use Triple Exponential Smoothing, which is also called **Holt Winter’s exponential smoothing**. You should use this only when there are three important signals in your time series data. For example, one signal could be the trend, another one could be a weekly seasonality and a third one could be a monthly seasonality.

**Going deeper into supervised machine learning models**

Supervised Machine Learning models work very differently than classical machine learning models. The main difference is that they consider that variables are either dependent variables or independent variables. Dependent variables, or target variables, are the variables that you want to predict. Independent variables are the variables that help you to predict.

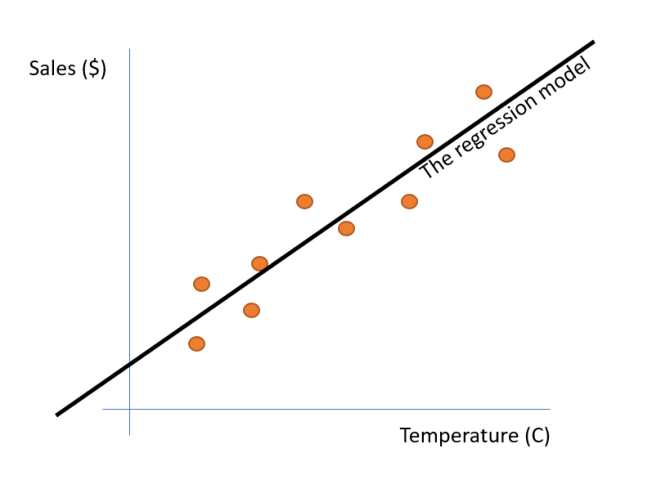
Supervised Machine Learning models are not made especially for time series data. After all, there are often no independent variables in time series data. Yet it is fairly simple to adapt them to time series by converting the seasonality (based on your time stamps for example) into independent variables.

**Linear regression**

Linear Regression is arguably the simplest supervised machine learning model. Linear Regression estimates linear relationships: each independent variable has a coefficient that indicates how this variable affects the target variable.

Simple Linear Regression is a Linear Regression in which there is only one independent variable. An example of a Simple Linear Regression model in non-time series data could be the following: hot chocolate sales that depend on the outside temperature (degrees Celsius).

The colder the temperature, the higher the hot chocolate sales. Visually, this could look like the graph below.

Linear regression | Source: Author

In Multiple Linear Regression, rather than using only one independent variable, you use multiple independent variables. You could imagine the 2d graph converting into a 3d graph, where the third axis represents the variable Price. In this case, you would build a Linear Model that explains the sales using temperature and price. You can add as many variables as you need.

Now, of course, this is not a time series data set: there is no time variable present. So, how could you use this technique for time series? The answer is fairly straightforward. Rather than only using temperature and price in this data set, you could add the variables year, month, day of the week, etc.

If you build a supervised model on time series, you have the disadvantage that you need to do a little bit of feature engineering to extract seasonality into variables in a way or another. An advantage is, however, that adding exogenous variables becomes much easier.

### Random forest

The Linear model is very limited: it can only fit linear relationships. Sometimes this will be enough, but in most cases, it is better to use more performant models. Random Forest is a much-used model that allows fitting nonlinear relationships. It is still very easy to use.

## Going deeper into advanced and specific time series models

In this part, you will discover two more advanced and specific time series models called GARCH and TBATS.

### GARCH

GARCH stands for Generalized Autoregressive Conditional Heteroskedasticity. It is an approach to estimating the volatility of financial markets and is generally used for this use case. It is seldom used for other use cases.

The model works well for this, as it assumes an ARMA model for the error variance of the time series rather than for the actual data. In this way, you can predict variability rather than actual values.

There exist a number of variants to the GARCH family of models, for example, check [this](https://en.wikipedia.org/wiki/Autoregressive_conditional_heteroskedasticity) out. This model is great to know, but should only be used when forecasting variability is the need and it is therefore relatively different from the other models that are presented in this article.

### TBATS

TBATS stands for the combination of the following components:

* Trigonometric seasonality
* Box-Cox transformation
* ARMA errors
* Trend
* Seasonal components

The model was created in 2011 as a solution to forecast time series with multiple seasonal periods. As it is relatively new and relatively advanced, it is less widespread and not as much used as the models in the ARIMA family.

## Going deeper into deep learning-based time series models

You have now seen two relatively different model families, each of them with its specific ways of fitting the models. Classical time series models are focused on relations between the past and the present. Supervised machine learning models are focused on relations between cause and effect.

You will now see three more recent models that can be used for forecasting as well. They are even more complex to apprehend and master and may (or may not) produce better results, depending on the data and the specifics of the use case.

### LSTM (Long Short-Term Memory)

LSTMs are Recurrent Neural Networks. Neural Networks are very complex machine learning models that pass input data through a network. Each node in the network learns a very simple operation. The neural network consists of many such nodes. The fact that the model can use a large number of simple nodes makes the overall prediction very complex. Neural Networks can therefore fit very complex and nonlinear data sets.

RNNs are a special type of Neural network, in which the network can learn from sequence data. This can be useful for multiple use cases, including understanding time series (which are clearly sequences of values over time), but also text (sentences are sequences of words).

LSTMs are a specific type of RNNs. They have proven useful for time series forecasting on multiple occasions. They require some data and are more complicated to learn than supervised models. Once you master them, they can prove to be very powerful depending on your data and your specific use case.

To go into LSTMs, the [Keras](https://keras.io/api/layers/recurrent_layers/lstm/" \t "_blank) library in Python is a great starting point.

### Prophet

Prophet is a time series library that was open-sourced by Facebook. It is a black-box model, as it will generate forecasts without much user specification. This can be an advantage, as you can almost automatically generate forecasting models without much knowledge or effort.

On the other hand, there is a risk here as well: if you do not pay close enough attention, you may very well be producing a model that seems good to the automated model building tool, but that in reality does not work well.

Extensive model validation and evaluation are recommended when using such black-box models, yet if you find that it works well on your specific use case, you may find a lot of added value here.