**Quick Response Code**

QR codes were first used in 1994 by Denso Wave, a Toyota subsidiary company in Japan. QR codes provided a quick and convenient approach to track vehicles during manufacturing process at automotive industry. After its successful implementation at Denso Wave, other industries attempted to adopt this technology too. Denso Wave has patented the QR code, but it’s open for worldwide use. In 2011 the QR code became commercial for the very first time through telecommunications industry (Walker, 2011).

In an article titled “Two-Level QR Code for Private Message Sharing and Document Authentication” (Tkachenko et. al, 2016), it was mentioned that the popularity of these codes is mainly due to the following features: they are robust to the copying process, easy to read by any device and any user, they have a high encoding capacity enhanced by error correction facilities, they have a small size and are robust to geometrical distortions. However, those undeniable advantages also have their counterparts:

1) Information encoded in a QR code is always accessible to everyone, even if it is ciphered and therefore is only legible to authorized users (the difference between “see” and “understand”).

2) It is impossible to distinguish an originally printed QR code from its copy due to their insensitivity to the Print-and-Scan (P&S) process.

QR code is a two-dimensional encoding of information and it is also called matrix code. This matrix code is machine-readable that consists of black and white squares. It can store information in the form of URL (Uniform Resource Locator), contact information, link to videos or photos, plain text and other types of content (International standard ISO/IEC 18004, 2000).

In a paper entitled “Exploring concept of QR Code and the benefits of using QR Code for companies” (Qianyu, 2014), discusses the QR code architecture. Each QR code symbol looks like a square pattern. This square pattern consists of two regions: encoding region and function patterns. The function patterns concentrate on the positioning where the encoding region represents the data encoding. The function pattern comprises finder patterns, timing patterns and alignment patterns. Three common structures on the three corners of QR code symbol are called finder patterns. Finder pattern is used for deciding the correct orientation of the symbol. Timing patterns are used by the decoder software to find the side of pattern. Alignment patterns are used in the case of image distortion to correctly decode the symbol by decoder software. The rest of the region i.e. other than function pattern is the encoded region where data code words and error correcting code words are stored.

Furthermore, the format information areas contain error correction level and mask pattern. The code version and error correction bits are stored in the version information areas. The QR code generation algorithm consists of information encoding using Reed-Solomon error correction code, information division on codewords, application of mask pattern, placement of codewords and function patterns into the QR code. The QR code recognition algorithm includes the scanning process, image binarization, geometrical correction and decoding algorithm (Tkachenko et. al, 2016)

Below are the key Characteristics of QR Code (Qianyu, 2014):

1) High Storage Capacity

A QR code symbol can store up to 7,089 characters of information, which is a huge amount as compared to 1-D barcode.

2) Encodable Character Set

* Numeric data (Digits 0-9)
* Alphanumeric data (upper case letters A-Z; Digits 0 - 9; nine other characters: space, : % \* + - / \_ $)
* Kanji characters

3) Small Printout Size

The information in QR code is stored in both horizontal and vertical directions. Due to this feature, for the same amount of data, space acquired by QR code is one fourth times less than the space acquired by 1-D barcode.

4) 360 Degree Reading

QR code is readable from any direction. This feature is provided by the finder patterns present at three corners of the symbol. The finder pattern helps to locate the QR code.

5) Capability of Restoring and Error Correction

If the part of code symbol is damaged or dirty, data can be recovered. The error detecting procedure can focus on the region of correct information. There are four levels of error correction of QR code that are L, M, Q and H. The level L has the weakest and level H has the strongest error correction capability.

**Time Series Forecasting**

Forecasting future values of an observed time series plays an important role in nearly all fields of science and engineering, such as economics, finance, business intelligence, meteorology and telecommunication (Palit and Popovic, 2005).

A forecasting method is a procedure for computing forecasts from present and past values. As such it may simply be an algorithmic rule and need not depend on an underlying probability model. Alternatively, it may arise from identifying a particular model for the given data and finding optimal forecasts conditional on that model. Thus, the two terms ‘method’ and ‘model’ should be kept clearly distinct. It is unfortunate that the term ‘forecasting model’ is used rather loosely in the literature and is sometimes wrongly used to describe a forecasting method (Chatfield, 2000).

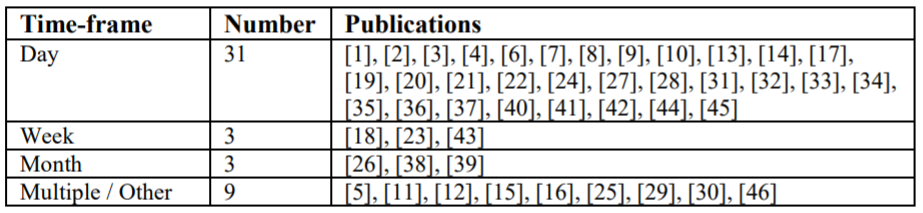
Chatfield on his book “Time-series Forecasting” (2000) broadly classified forecasting methods into three types:

(a) Judgmental forecasts based on subjective judgement, intuition, ‘inside’ commercial knowledge, and any other relevant information.

(b) Univariate methods where forecasts depend only on present and past values of the single series being forecasted, possibly augmented by a function of time such as a linear trend.

(c) Multivariate methods where forecasts of a given variable depend, at least partly, on values of one or more additional time series variables, called predictor or explanatory variables.

In an article titled “Financial time series forecasting with machine learning techniques: A survey” (Krollner et.al, 2010), authors discussed how different intervals were used in various literature. Figure 2.x below gives an overview of the different forecasting intervals used in the literature. The prediction periods are categorised into one day, one week, and one month ahead predictions. Publications using multiple or different time-frame are listed under ’Multiple / Others’. Most papers make one day ahead predictions e.g. predicting the next day’s closing price. However, being able to predict the stock index one day ahead does not necessarily mean that an investor can take advantage of this information in terms of trading profit, especially since the index itself cannot be traded.



Let *Y* = {*y*1, . . . , *yn*} denote a time series. Forecasting denotes the process of estimating the future values of *Y , yn+h*, where h denotes the forecasting horizon. Quantitative approaches to time series forecasting are split into two categories: univariate and multivariate. Univariate methods refer to approaches that model future observations of a time series according to its past observations. Multivariate approaches extend univariate ones by considering additional time series that are used as explanatory variables. We will focus on univariate approaches in this work. The forecasting horizon is another aspect to consider when addressing time series prediction problems. Forecasting methods usually focus on one step ahead forecasting, i.e., the prediction of the next value of a time series (*yn*+1). Sometimes one is interested in predicting many steps into the future. These tasks are often referred to as multi-step forecasting (Taieb et al., 2012).

In a paper titled “Machine Learning vs Statistical Methods for Time Series Forecasting: Size Matters” (Cerqueira et.al, 2019), the authors discussed common time series models:

1. Naive method, also known as the random walk forecast, predicts the future values of the time series according to the last known observation:

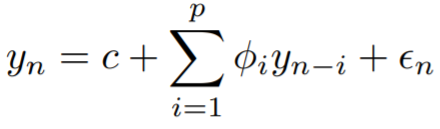
*yˆn+h = yn*

1. Seasonal naive model works similarly to the naive method. The difference is that the seasonal naive approach uses the previously known value from the same season of the intended forecast:

*yˆn+h = yn+h−m*

where *m* denotes the seasonal period.

1. ARMA (Auto-Regressive Moving Average) is one of the most commonly used methods to model univariate time series. ARMA(p,q) combines two components: AR(p), and MA(q). According to the AR(p) model, the value of a given time series, yn, can be estimated using a linear combination of the p past observations, together with an error term  and a constant term *c*.



1. Exponential smoothing model is similar to the AR(p) model in the sense that it models the future values of time series using a linear combination of its past observations. Exponential smoothing methods produce weighted averages of the past values, where the weight decays exponentially as the observations are older.

*yn*+1 = *ynβ*0 + *yn*−1*β*1 + *yn*−2*β*2 + · · ·

There are three main strategies for Multi-step Time Series Forecasting: Recursive, Direct and DirRec. The Recursive strategy (Weigend and Gershenfeld, 1994) trains first a one-step model *f*

*yt*+1 = *f*(*yt*,...,*yt*−*n*+1) + *wt*+1,

with *t* ∈ {*n*, . . . , *N* − 1} and then uses it recursively for returning a multistep prediction. A well-known drawback of the recursive method is its sensitivity to the estimation error, since estimated values, instead of actual ones, are more and more used when we get further in the future.

The Direct strategy (Cheng et.al, 2006) learns independently *H* models *fh*

*yt*+*h* = *fh*(*yt*,...,*yt*−*n*+1) + *wt*+*h*,

with *t* ∈ {*n*, . . . , *N* − *H*} and *h* ∈ {1,...,*H*} and returns a multi-step forecast by concatenating the H predictions. Since the Direct strategy does not use any approximated values to compute the forecasts, it is not prone to any accumulation of errors. Notwithstanding, it has some weaknesses. First, since the *H* models are learned independently no statistical dependencies between the predictions ˆ*yN+h* is considered. Second direct methods often require higher functional complexity than iterated ones in order to model the stochastic dependency between two series values at two distant instants. Last but not least, this strategy demands a large computational time since the number of models to learn is equal to the size of the horizon.

The DirRec strategy (Sorjamaa and Lendasse, 2006) combines the architectures and the principles underlying the Direct and the Recursive strategies. DirRec computes the forecasts with different models for every horizon (like the Direct strategy) and, at each time step, it enlarges the set of inputs by adding variables corresponding to the forecasts of the previous step (like the Recursive strategy). However, note that unlike the two previous strategies, the embedding size *n* is not the same for all the horizons. In other terms, the DirRec strategy learns *H* models *fh* from the time series [*y*1,...,*yN* ] where

*yt*+*h* = *fh*(*yt*+*h*−1,...,*yt*−*n*+1) + *wt*+*h*,

with *t* ∈ {*n*, . . . , *N* − *H*} and *h* ∈ {1,...,*H*}.

**ARIMA**

Auto Regressive Integrated Moving Average (ARIMA) processes are a class of stochastic processes used to analyze time series (Box and Jenkins, 1994). The general scheme is as follows:

Step 0) A class of models is formulated assuming certain hypotheses.

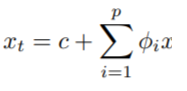
Step 1) A model is identified for the observed data.

Step 2) The model parameters are estimated.

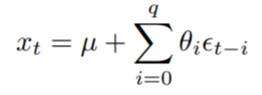
Step 3) If the hypotheses of the model are validated, go to Step 4, otherwise go to Step 1 to refine the model.

Step 4) The model is ready for forecasting.

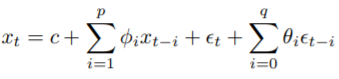
A simple form of an AR model of order p, i.e., *AR(p)*, can be written as a linear process given by:



Where *xt* is the stationary variable, *c* is constant, the terms in *φi* are autocorrelation coefficients at lags 1, 2,,*p* and *t* , the residuals, are the Gaussian white noise series with mean zero and variance *σ*2. An MA model of order q, i.e., *MA(q)*, can be written in the form:



Where *μ* is the expectation of *xt* (usually assumed equal to zero), the *θi* terms are the weights applied to the current and prior values of a stochastic term in the time series, and *θ*0 = 1. We assume that *t* is a Gaussian white noise series with mean zero and variance *σ2* . We can combine these two models by adding them together and form an ARIMA model of order (*p, q*):



Where *φi* = 0, *θi* = 0, and *σ*2 > 0. The parameters p and q are called the AR and MA orders, respectively. ARIMA forecasting, also known as Box and Jenkins forecasting, is capable of dealing with non-stationary time series data because of its “integrate” step. In fact, the “integrate” component involves differencing the time series to convert a non-stationary time series into a stationary. The general form of a ARIMA model is denoted as ARIMA(*p, d, q*) (Siami-Namini et.al, 2018).