**1. INTRODUCTION:**

**1.1. What is Times Series:**

Time series is a sequence of observations recorded at regular time intervals. Depending on the frequency of observations, a time series may typically be hourly, daily, weekly, monthly, quarterly and annual.

Here time is the independent variable while the dependent variable might be

* Stock market data
* Sales data of companies
* Data from the sensors of smart devices
* The measure of electrical energy generated in the powerhouse.

Time series forecasting is basically the machine learning modeling for Time Series data (years, days, hours…etc.)for predicting future values using Time Series modeling.

To gain some useful insights from time-series data, you have to decompose the time series and look for some basic components such as trend, seasonality, cyclic behaviour, and irregular fluctuations. Based on some of these behaviours, we are deciding on which model to choose for time series modelling.

**1.2. What makes Time Series Special?**

Time Series is different from a regular regression problem in two points:

* It is time dependent. So the basic assumption of a linear regression model that the observations are independent doesn’t apply in this case.
* most Time Series have some form of seasonality trends, i.e. variations specific to a particular time frame. For example, if you see the sales of a woolen jacket over time, you will invariably find higher sales in winter seasons.

Because of the inherent properties of a Time Series, there are various steps involved in analyzing it.

Let’s get a better understanding by exploring somw Basic concept of Time Series

**2. BASIC CONCEPTS OF TIME- SERIES**

**2.1. Seasonality:**

A data pattern that repeats itself at regular intervals is called **Seasonality**. Seasonal patterns can be very useful in scenarios like predicting network traffic, road traffic, sales patterns of certain commodities that have high sales in certain seasons, etc.

Seasonal data with a slightly increasing trend.

**2.2. Trend:**

A long-term increasing or decreasing pattern in the data points indicates a trend. It could be linear/non-linear. For example, global temperature is at an increasing trend due to global warming.

Global temperature through the years with an increasing trend.

**2.3. Cyclic:**

A cycle occurs when the data exhibits rises and falls that are not of a fixed frequency. A cycle is different from seasonality by way of its irregularity in frequency.

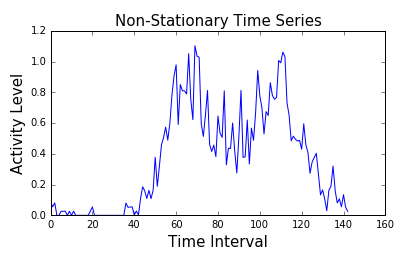
**2.4. Random:**

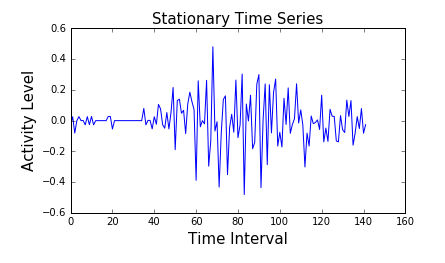
We know that data cannot be perfect, and we always need to provide leeway for some noise.

**3. STATIONARY**

**3.1 What is stationary?**

In the most intuitive sense, stationarity means that the statistical properties of a process generating a time series do not change over time. In other words all its statistical properties (mean,variance, standard deviation) remain constant over time.





If you keenly observe the above images you can find the difference between the two plots. In stationary time series the mean, variance, and standard deviation of the observed value over time are almost constant whereas in non-stationary time series this is not the case.

There are a lot of statistical theories to explore stationary series than non-stationary series.

In practice we can assume the series to be stationary if it has constant statistical properties over time and these properties can be:

• Constant mean

• Constant variance

• An auto co-variance that does not depend on time.

### 3.2 How to make a time series stationary?

You can make series stationary by:

* Differencing the Series (once or more)
* Take the log of the series
* Take the nth root of the series
* Combination of the above

The most common and convenient method to stationarize the series is by **differencing the series at least once until it becomes approximately stationary**.

So what is differencing? If Y\_t is the value at time ‘t’, then the first difference of Y = Yt – Yt-1. In simpler terms, differencing the series is nothing but subtracting the next value by the current value. If the first difference doesn’t make a series stationary, you can go for the second differencing. And so on.

For example, consider the following series: [1, 5, 2, 12, 20]

First differencing gives: [5-1, 2-5, 12-2, 20-12] = [4, -3, 10, 8]

Second differencing gives: [-3-4, -10-3, 8-10] = [-7, -13, -2]

### 3.3 Why make a non-stationary series stationary before forecasting?

The stationarity of a series can be established by looking at the plot of the series.

Another method is to split the series into 2 or more contiguous parts and computing the summary statistics like the mean, variance and the autocorrelation. If the stats are quite different, then the series is not likely to be stationary.

Nevertheless, you need a method to quantitatively determine if a given series is stationary or not. This can be done using statistical tests called ‘Unit Root Tests’. There are multiple implementations of Unit Root tests like:

* Augmented Dickey Fuller test (ADH Test)
* Kwiatkowski-Phillips-Schmidt-Shin – KPSS test (trend stationary)
* Philips Perron test (PP Test)

**The most commonly used is the ADF test**,In this test, First we consider the null hypothesis: the time series is non- stationary. The result from the rest will contain the test statistic and critical value for different confidence levels. The idea is to have Test statistics less than critical value, in this case we can reject the null hypothesis and say that this Time series is indeed stationary.

# 4. FORCASTING A TIME SERIES:

Now that we have made the Time series stationary, let’s make models on the time series using differencing because it is easy to add the error , trend and seasonality back into predicted values .

**We will use statistical modelling method called ARIMA to forecast the data where there are dependencies in the values.**

Auto Regressive Integrated Moving Average(ARIMA) — It is like a liner regression equation where the predictors depend on parameters (p,d,q) of the ARIMA model .These three parameters account for seasonality, trend, and noise in data.

We can dive into this part more intensively in the Code Implementation section.

**LSTM**

Long Short Term Memory is a kind of recurrent neural network. In RNN output from the last step is fed as input in the current step. LSTM was designed by Hochreiter & Schmidhuber. It tackled the problem of long-term dependencies of RNN in which the RNN cannot predict the word stored in the long-term memory but can give more accurate predictions from the recent information. As the gap length increases RNN does not give an efficient performance. LSTM can by default retain the information for a long period of time. It is used for processing, predicting, and classifying on the basis of time-series data.

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) that is specifically designed to handle sequential data, such as time series, speech, and text. LSTM networks are capable of learning long-term dependencies in sequential data, which makes them well suited for tasks such as language translation, speech recognition, and time series forecasting.

A traditional RNN has a single hidden state that is passed through time, which can make it difficult for the network to learn long-term dependencies. LSTMs address this problem by introducing a memory cell, which is a container that can hold information for an extended period of time. The memory cell is controlled by three gates: the input gate, the forget gate, and the output gate. These gates decide what information to add to, remove from, and output from the memory cell.

The input gate controls what information is added to the memory cell. The forget gate controls what information is removed from the memory cell. And the output gate controls what information is output from the memory cell. This allows LSTM networks to selectively retain or discard information as it flows through the network, which allows them to learn long-term dependencies.

LSTMs can be stacked to create deep LSTM networks, which can learn even more complex patterns in sequential data. LSTMs can also be used in combination with other neural network architectures, such as Convolutional Neural Networks (CNNs) for image and video analysis.

### Structure Of LSTM:

LSTM has a chain structure that contains four neural networks and different memory blocks called **cells**.

Information is retained by the cells and the memory manipulations are done by the **gates.** There are three gates –

**1. Forget Gate:** The information that is no longer useful in the cell state is removed with the forget gate. Two inputs *x\_t* (input at the particular time) and *h\_t-1* (previous cell output) are fed to the gate and multiplied with weight matrices followed by the addition of bias. The resultant is passed through an activation function which gives a binary output. If for a particular cell state the output is 0, the piece of information is forgotten and for output 1, the information is retained for future use. The equation for the forget gate is:

 f\_t = σ(W\_f · [h\_t-1, x\_t] + b\_f)  
 where:

* W\_f represents the weight matrix associated with the forget gate.
* [h\_t-1, x\_t] denotes the concatenation of the current input and the previous hidden state.
* b\_f is the bias with the forget gate.
* σ is the sigmoid activation function.

**2. Input gate:** The addition of useful information to the cell state is done by the input gate. First, the information is regulated using the sigmoid function and filter the values to be remembered similar to the forget gate using inputs *h\_t-1* and *x\_t*. Then, a vector is created using *tanh* function that gives an output from -1 to +1, which contains all the possible values from h\_t-1 and *x\_t*. At last, the values of the vector and the regulated values are multiplied to obtain the useful information. The equation for the input gate is:

* i\_t = σ(W\_i · [h\_t-1, x\_t] + b\_i)
* Ĉ\_t = tanh(W\_c · [h\_t-1, x\_t] + b\_c)
* C\_t = f\_t ⊙ C\_t-1 + i\_t ⊙ Ĉ\_t
* where
* ⊙ denotes element-wise multiplication
* tanh is tanh activation function

**3. Output gate:** The task of extracting useful information from the current cell state to be presented as output is done by the output gate. First, a vector is generated by applying tanh function on the cell. Then, the information is regulated using the sigmoid function and filter by the values to be remembered using inputs *h\_t-1* and *x\_t*. At last, the values of the vector and the regulated values are multiplied to be sent as an output and input to the next cell. The equation for the output gate is:

 o\_t = σ(W\_o · [h\_t-1, x\_t] + b\_o)

**Advantages of LSTM**

1. Long-term dependencies can be captured by LSTM networks. They have a memory cell that is capable of long-term information storage.
2. In traditional RNNs, there is a problem of vanishing and exploding gradients when models are trained over long sequences. By using a gating mechanism that selectively recalls or forgets information, LSTM networks deal with this problem.
3. LSTM enables the model to capture and remember the important context, even when there is a significant time gap between relevant events in the sequence. So where understanding context is important, LSTMS are used. eg. machine translation.

**Disadvantages of LSTM**

1. Compared to simpler architectures like feed-forward neural networks LSTM networks are computationally more expensive. This can limit their scalability for large-scale datasets or constrained environments.
2. Training LSTM networks can be more time-consuming compared to simpler models due to their computational complexity. So training LSTMs often requires more data and longer training times to achieve high performance.
3. Since it is processed word by word in a sequential manner, it is hard to parallelize the work of processing the sentences.

**Some of the famous applications of LSTM includes:**

1. Long Short-Term Memory (LSTM) is a powerful type of Recurrent Neural Network (RNN) that has been used in a wide range of applications. Here are a few famous applications of LSTM:
2. Language Modeling: LSTMs have been used for natural language processing tasks such as language modeling, machine translation, and text summarization. They can be trained to generate coherent and grammatically correct sentences by learning the dependencies between words in a sentence.
3. Speech Recognition: LSTMs have been used for speech recognition tasks such as transcribing speech to text and recognizing spoken commands. They can be trained to recognize patterns in speech and match them to the corresponding text.
4. Time Series Forecasting: LSTMs have been used for time series forecasting tasks such as predicting stock prices, weather, and energy consumption. They can learn patterns in time series data and use them to make predictions about future events.
5. Anomaly Detection: LSTMs have been used for anomaly detection tasks such as detecting fraud and network intrusion. They can be trained to identify patterns in data that deviate from the norm and flag them as potential anomalies.
6. Recommender Systems: LSTMs have been used for recommendation tasks such as recommending movies, music, and books. They can learn patterns in user behavior and use them to make personalized recommendations.
7. Video Analysis: LSTMs have been used for video analysis tasks such as object detection, activity recognition, and action classification. They can be used in combination with other neural network architectures, such as Convolutional Neural Networks (CNNs), to analyze video data and extract useful information.

Data Used- on the problem of forecasting future web traffic for approximately 145,000 Wikipedia articles.

Compared performances of ARMA, ARIMA, & LSTM and achieved best RMSE of 1.72 on ARMA