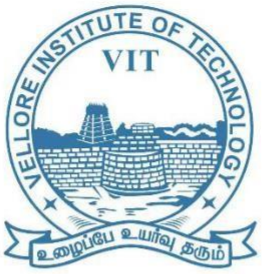


VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

TITLE: DATA VISUALIZATION ON WATER CRISIS IN CHENNAI

PROJECT REPORT

DATA VISUALIZATION

SUBMITTED TO: PROF. RAJKUMAR R

GROUP MEMBERS:

19MCB0008 – Lavanya Bandla
19MCB0011 – Kalyani V.Anjankar

Abstract

Chennai also known as Madras is the capital of the Indian state of Tamil Nadu. The city is facing an acute water shortage now (June 2019). Chennai is entirely dependent on ground water resources to meet its water needs.

There are four reservoirs in the city, namely, Red Hills, Cholavaram, Poondi and Chembarambakkam, with a combined capacity of 11,057 mcft.

These are the major sources of fresh water for the city.

Apart from the reservoirs, the other sources of fresh water water are desalination plants at Nemelli and Minjur; aquifers in Neyveli, Minjur and Panchetty; Cauvery water from Veeranam lake;

Here is an attempt to put together a dataset that has the information about the various water sources available in the city.

At the end, we'll get to know which method is best to predict the needs required to fulfil the water requirement till next monsoon and this method will also help many other states and countries.

Objective

The idea is to see if we can use this dataset to

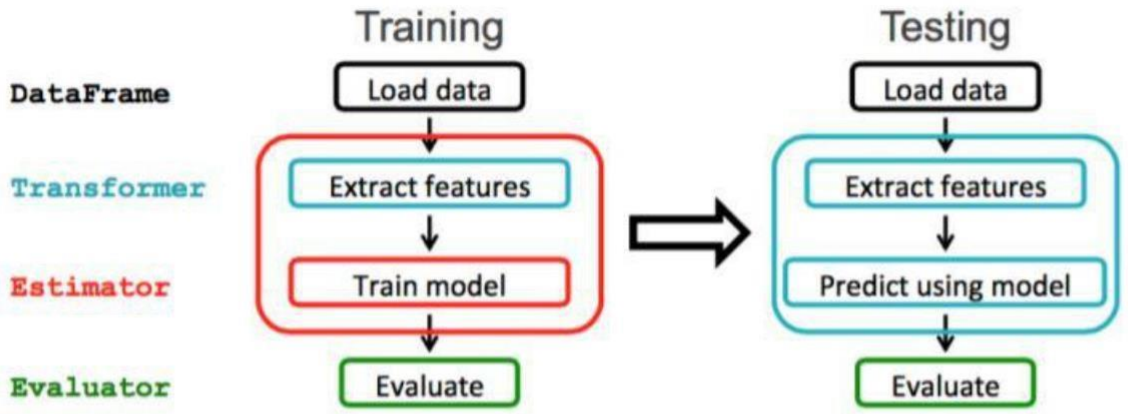
1. Visualize the water need / usage of the city
2. Identify whether the water sources availability will be able to meet the needs till the subsequent monsoon?
3. How bad is the current water crisis compared to previous years?

Methodology

In this project, we worked with historical data about the Chennai water reservoir levels. We have algorithms to predict whether the available water resources will be able to meet the water need of the Chennai and nearby people till next monsoon or not, starting with simple algorithms like Time Series forecasting.

Time series analysis comprises methods for analysing time series data in order to extract meaningful statistics and other characteristics of the data. Time series forecasting is the use of a model to predict future values based on previously observed values.

Time series are widely used for non-stationary data, like economic, weather, stock price.



The Algorithms used to predict the Cyber Frauds are:

- Time Series Forecasting

Dataset Description

This dataset has details about the water availability in the four main reservoirs over the last 15 years

Poondi

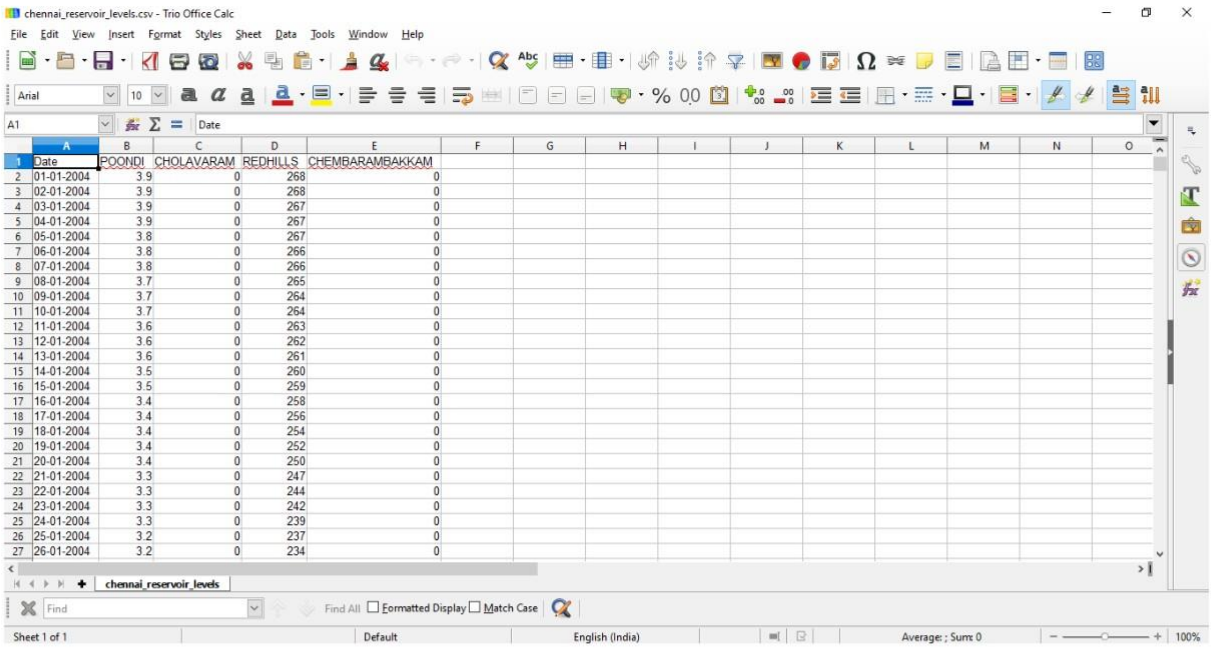
Cholavaram

Redhills

Chembarambakkam

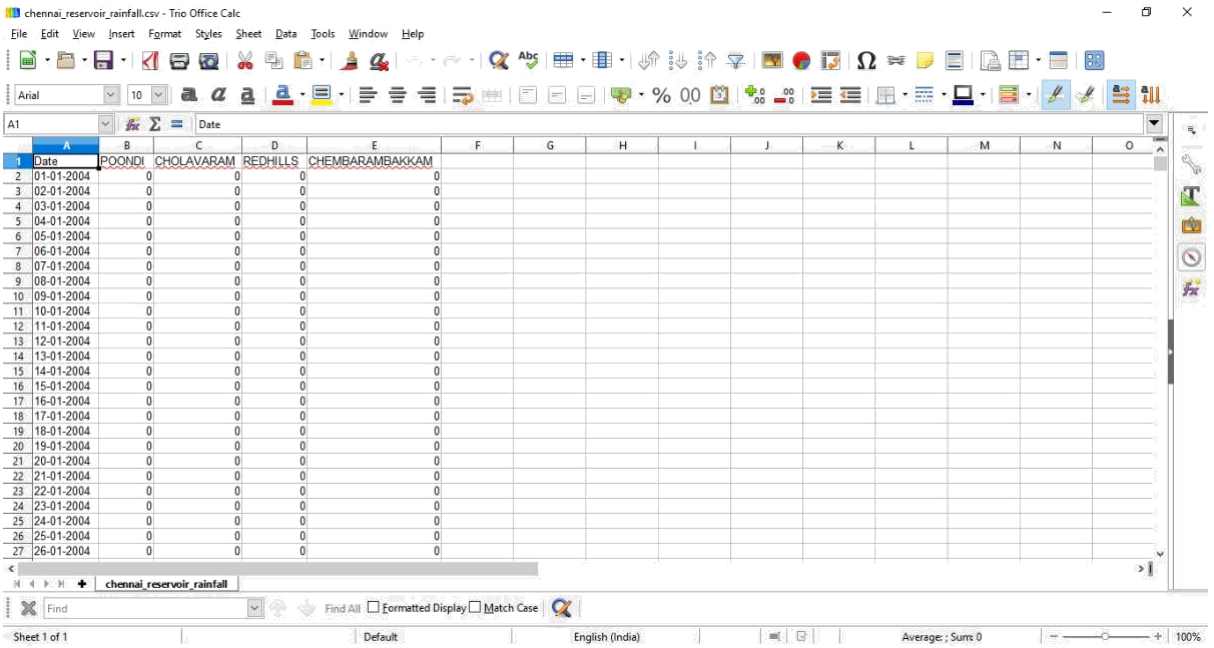
The data is available on a daily basis and the unit is million cubic feet.

URL - https://www.kaggle.com/sudalairajkumar/chennai-water-management



Date	POONDI	CHOLAVARAM	REDHILLS	CHEMBARAMBAKKAM
01-01-2004	3.9	0	268	0
02-01-2004	3.9	0	268	0
03-01-2004	3.9	0	267	0
04-01-2004	3.9	0	267	0
05-01-2004	3.8	0	267	0
06-01-2004	3.8	0	266	0
07-01-2004	3.8	0	266	0
08-01-2004	3.7	0	265	0
09-01-2004	3.7	0	264	0
10-01-2004	3.7	0	264	0
11-01-2004	3.6	0	263	0
12-01-2004	3.6	0	262	0
13-01-2004	3.6	0	261	0
14-01-2004	3.5	0	260	0
15-01-2004	3.5	0	259	0
16-01-2004	3.4	0	258	0
17-01-2004	3.4	0	256	0
18-01-2004	3.4	0	254	0
19-01-2004	3.4	0	252	0
20-01-2004	3.4	0	250	0
21-01-2004	3.3	0	247	0
22-01-2004	3.3	0	244	0
23-01-2004	3.3	0	242	0
24-01-2004	3.3	0	239	0
25-01-2004	3.2	0	237	0
26-01-2004	3.2	0	234	0

Chennai Water Reservoir level



Date	POONDI	CHOLAVARAM	REDHILLS	CHEMBARAMBAKKAM
01-01-2004	0	0	0	0
02-01-2004	0	0	0	0
03-01-2004	0	0	0	0
04-01-2004	0	0	0	0
05-01-2004	0	0	0	0
06-01-2004	0	0	0	0
07-01-2004	0	0	0	0
08-01-2004	0	0	0	0
09-01-2004	0	0	0	0
10-01-2004	0	0	0	0
11-01-2004	0	0	0	0
12-01-2004	0	0	0	0
13-01-2004	0	0	0	0
14-01-2004	0	0	0	0
15-01-2004	0	0	0	0
16-01-2004	0	0	0	0
17-01-2004	0	0	0	0
18-01-2004	0	0	0	0
19-01-2004	0	0	0	0
20-01-2004	0	0	0	0
21-01-2004	0	0	0	0
22-01-2004	0	0	0	0
23-01-2004	0	0	0	0
24-01-2004	0	0	0	0
25-01-2004	0	0	0	0
26-01-2004	0	0	0	0

Chennai Rainfall levels

Data Pre-processing

For Time Series Forecasting, the date is encoded into a proper date time format before processing because the time series algorithm needs the date to be in a specific format for its proper working.

Pre-processing the dataset

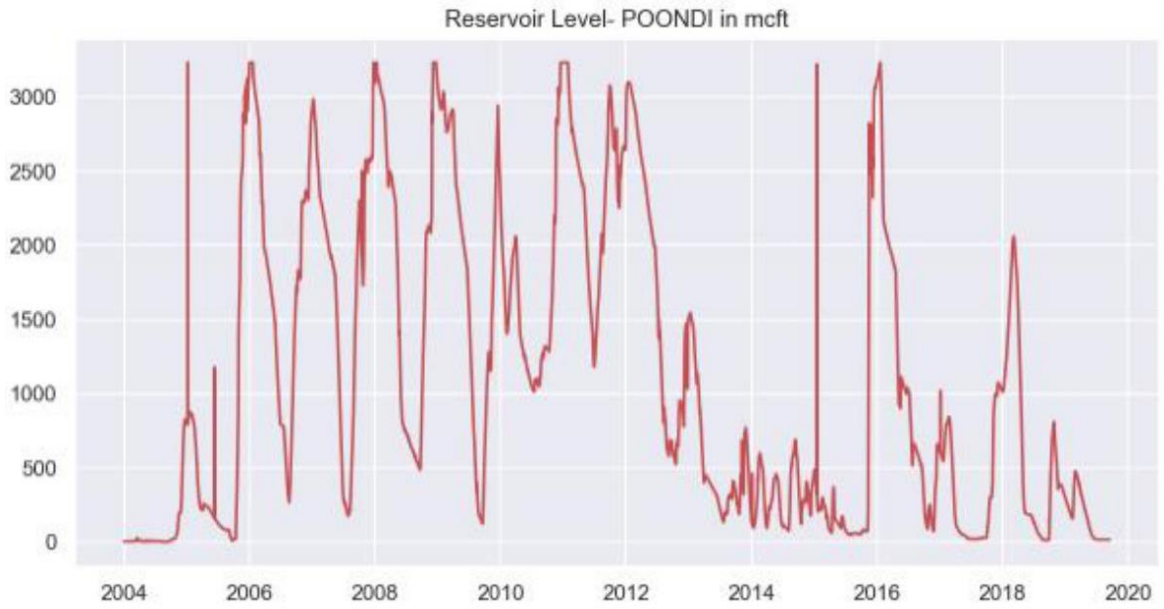
In [5]:
```
dataset_rainfall['Date']=pd.to_datetime(dataset_rainfall['Date'], infer_datetime_format=True)
indexDataset = dataset_rainfall.set_index(['Date'])
```

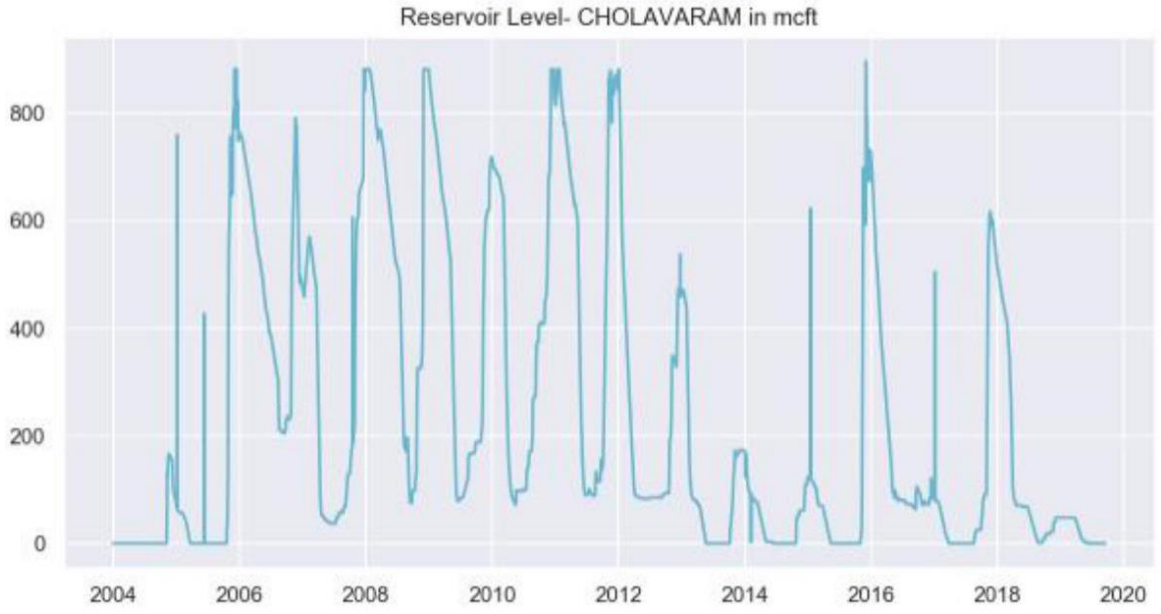
In [6]:
```
from datetime import datetime
indexDataset.head()
```

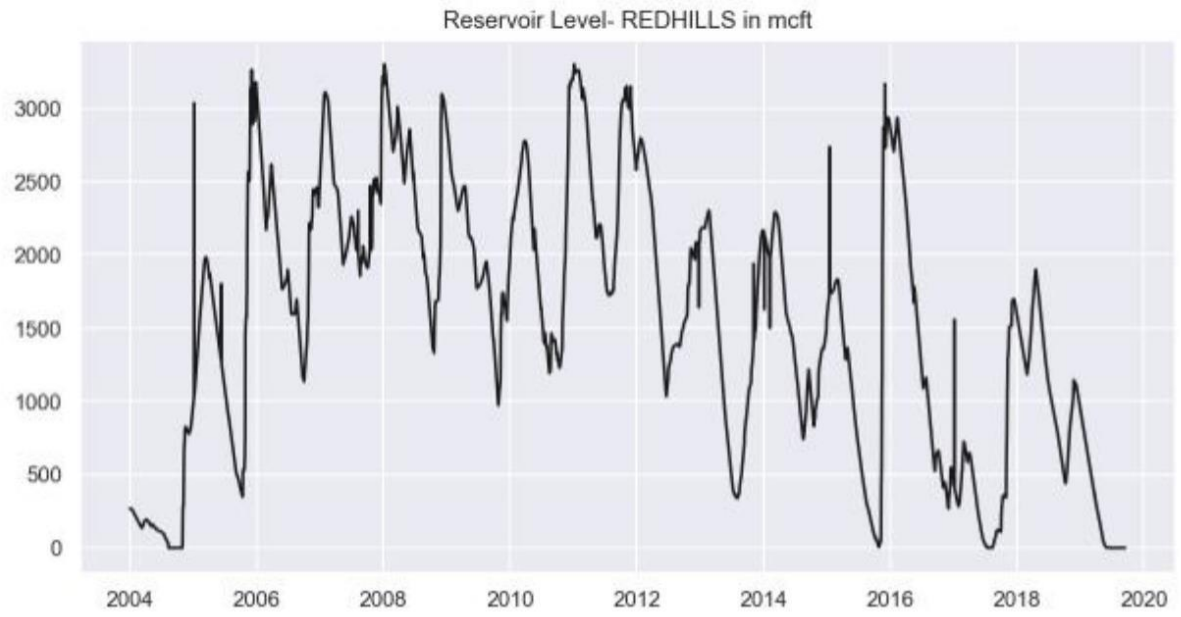
Out[6]:

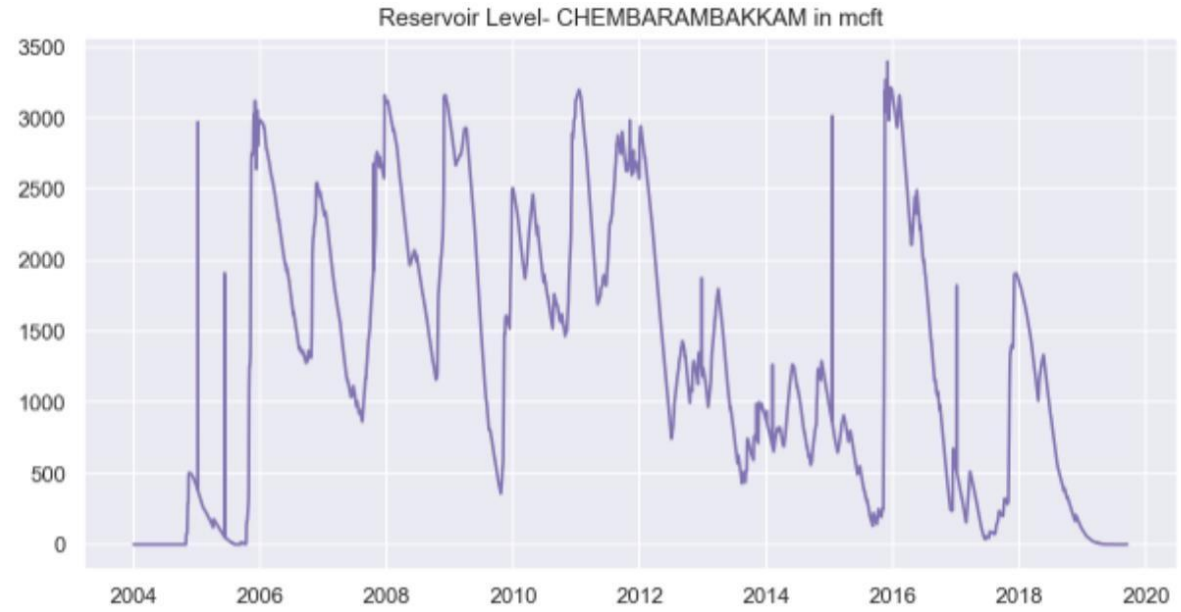
Date	Total
2004-01-01	0.0
2004-02-01	0.0
2004-03-01	0.0
2004-04-01	0.0
2004-05-01	0.0

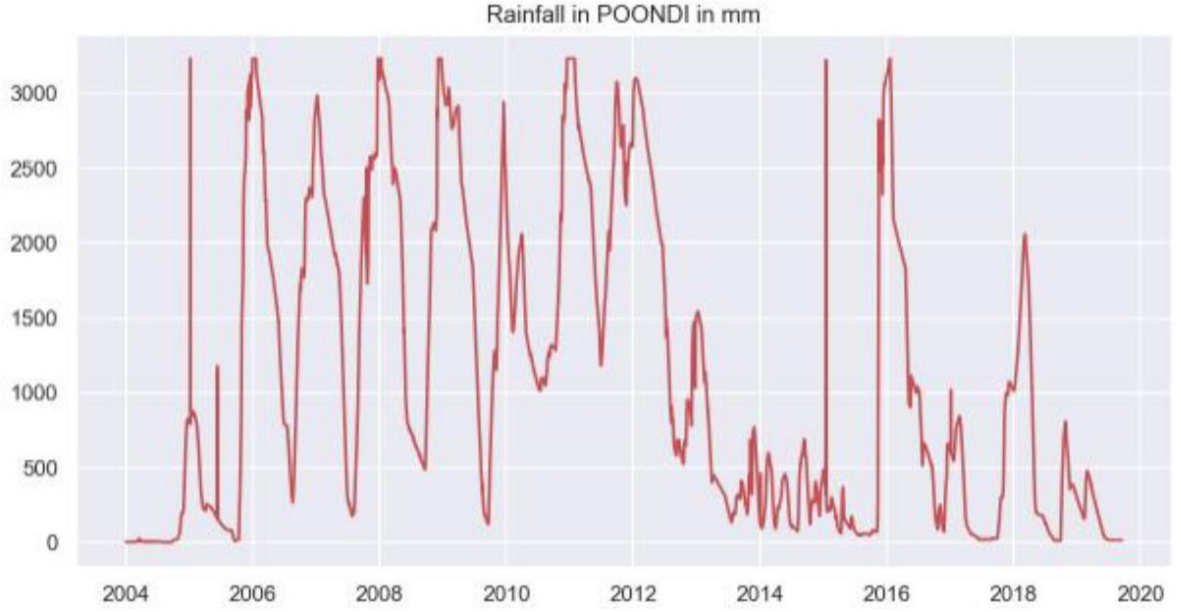
Data Visualization

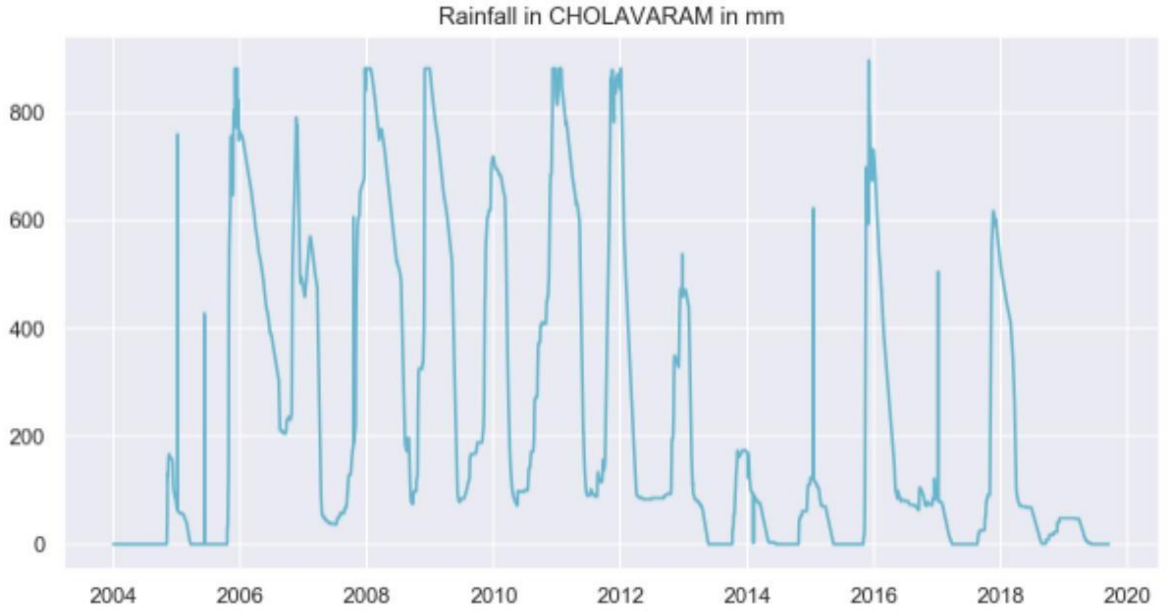


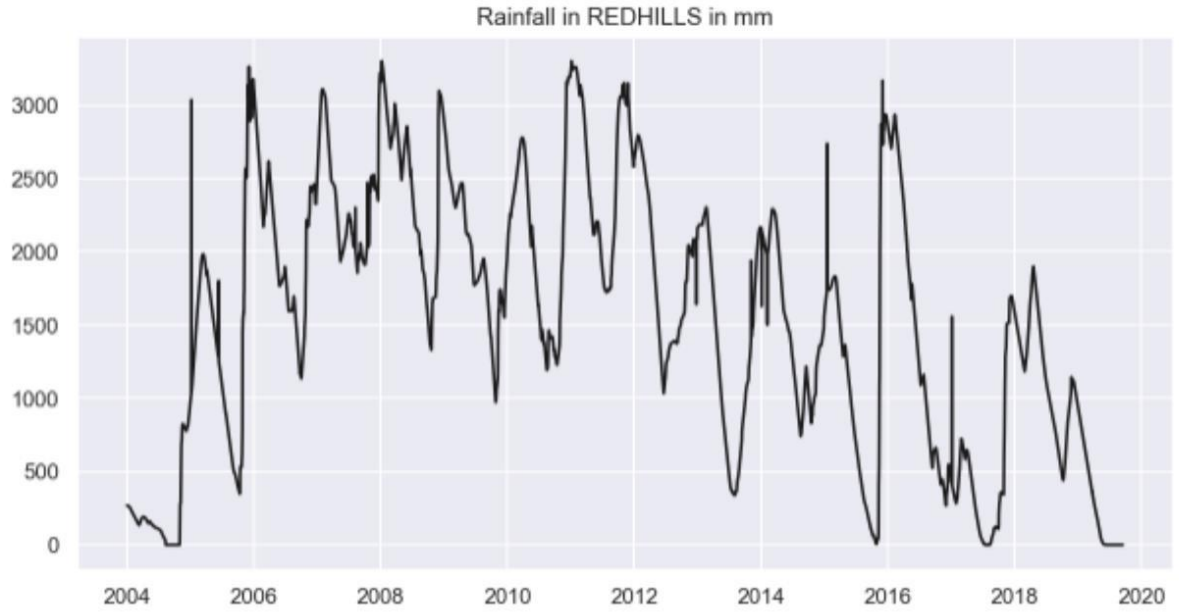


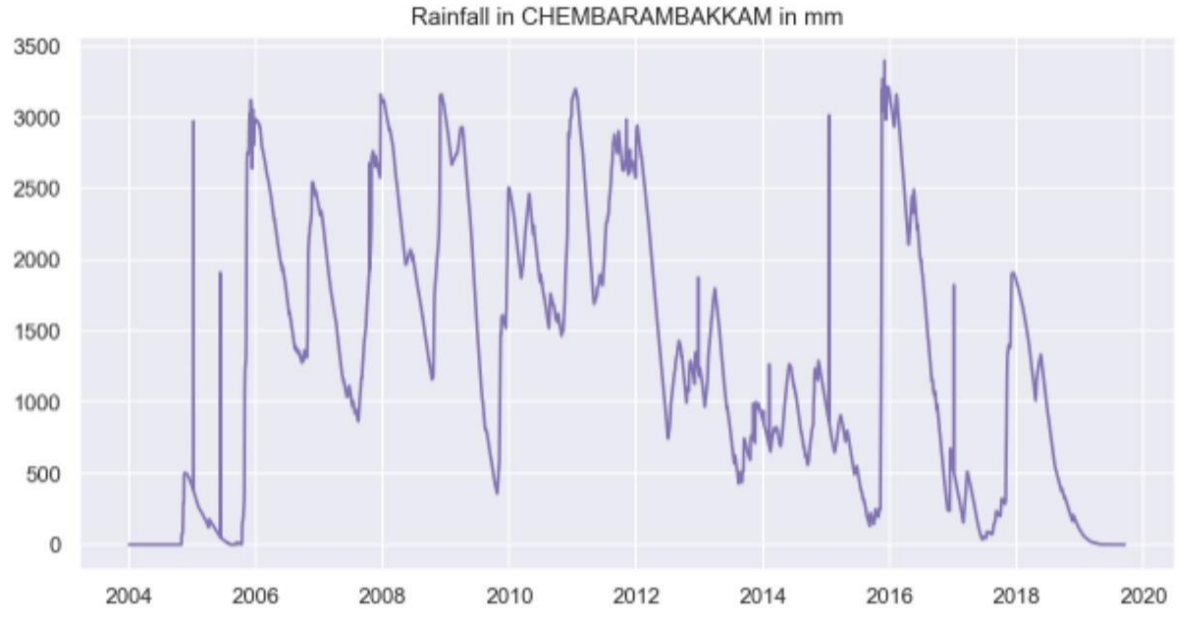

Reservoir Level- REDHILLS in mcft
3000
2500
2000
1500
1000
500
0
2004
2006
2008
2010
2012
2014
2016
2018
2020


Reservoir Level- CHEMBARAMBAKKAM in mcft
3500
3000
2500
2000
1500
1000
500
0
2004
2006
2008
2010
2012
2014
2016
2018
2020


Rainfall in POONDI in mm
3000
2500
2000
1500
1000
500
0
2004
2006
2008
2010
2012
2014
2016
2018
2020


Rainfall in CHOLAVARAM in mm
800
600
400
200
0
2004
2006
2008
2010
2012
2014
2016
2018
2020


Rainfall in REDHILLS in mm
3000
2500
2000
1500
1000
500
0
2004
2006
2008
2010
2012
2014
2016
2018
2020


Rainfall in CHEMBARAMBAKKAM in mm
3500
3000
2500
2000
1500
1000
500
0
2004
2006
2008
2010
2012
2014
2016
2018
2020

Approach

TIME SERIES ANALYSIS

In this Algorithm, Sequence of Data is stored and recorded at a specific interval of time and then that data is analysed to forecast future.

Components of Time Series?

There are four components of Time Series,

Trend – It is the overall change in the pattern throughout a period of time. It can have three possible value like, If the value increases after certain period of time then trend is Upward, if value decreases then it Downwards and if in case the value remains the same then trend is horizontal constant.

Seasonality – It is the periodic changes over a period of time. E.g. Sale of Warm clothes increases during every winter seasons. Hence here is a fixed pattern.

Cyclicity – It is almost similar to the Seasonality but here the time period is not Fixed. E.g. There is no fixed time – period for recession.

Irregularity - When there is an unpredictable patter in the graph is called as Irregularity. E.g. Suppose a Natural Calamity happened and during that time there is an steep growth in sale of the medicines or ointments. But as it is completely unpredictable so this type of patterns falls under the category of Irregularity.

Time Series works on the stationary data, And whenever Non- Stationary data is supplied it first convert it Stationary form and then process it further.

For a data to be stationary, it must follow three constraints –

i) Mean should not vary with time

ii) Variance should not vary with time

iii) The covariance of the I th and (i + m) th term should not vary with time.

ARIMA Model

ARIMA Model is a statistical model used for analysing and calculating time series. It is formed by the integration of two words “AR – Auto Regression” and “MA – Moving Average”.

AR- This model uses the dependence relation between the current and new class.

MA- This model dependency between an observation and a residual error from a moving average

Components of ARIMA Model -

P - Number of Autoregressive lags
Q – Size of the Moving Average
D – Order of Differentiation

Why Time Series Forecasting?

It helps to understand the past behaviour and would be helpful for future predictions and helps us to compare the present performance of the series with that of the past. Moreover, it also make the analyst to analyse the factor that influences the fluctuation of the series.

Accuracy:

Rainfall prediction

```
In [23]: import numpy as np
         from sklearn.metrics import mean_squared_error
         from sklearn.metrics import r2_score
         test_set_rmse = (np.sqrt(mean_squared_error(test, predictions)))
         test_set_r2 = r2_score(test, predictions)

In [24]: print(test_set_rmse)
Out[24]: 643.9896

In [25]: print(test_set_r2)
Out[25]: 0.76331
```

Water Reservior Prediction:

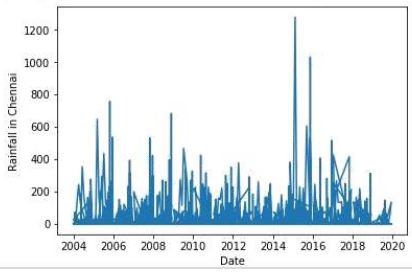
```
In [24]: mean_squared_error(test,predictions)
Out[24]: 360.114

In [21]: from sklearn.metrics import accuracy_score

         print(accuracy_score(test, predictions.round()))
Out[21]: 0.80375
```

Rainfall prediction:

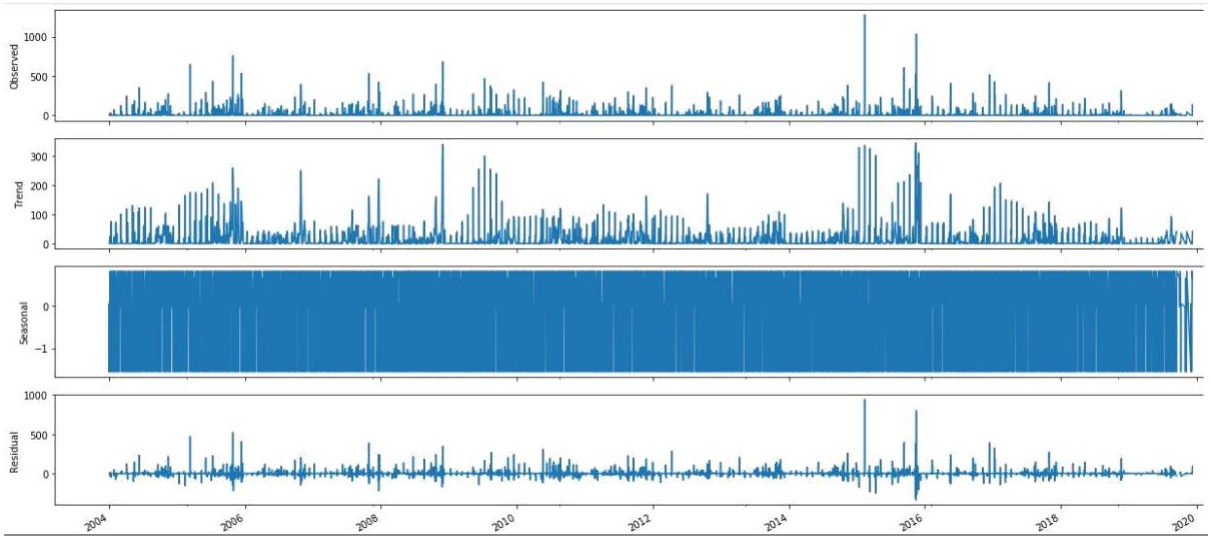
▾ Plotting the dataset to check the stationarity of the data

```
[ ]  plt.xlabel("Date")
     plt.ylabel("Rainfall in Chennai")
     plt.plot(indexDataset['Total'])
```

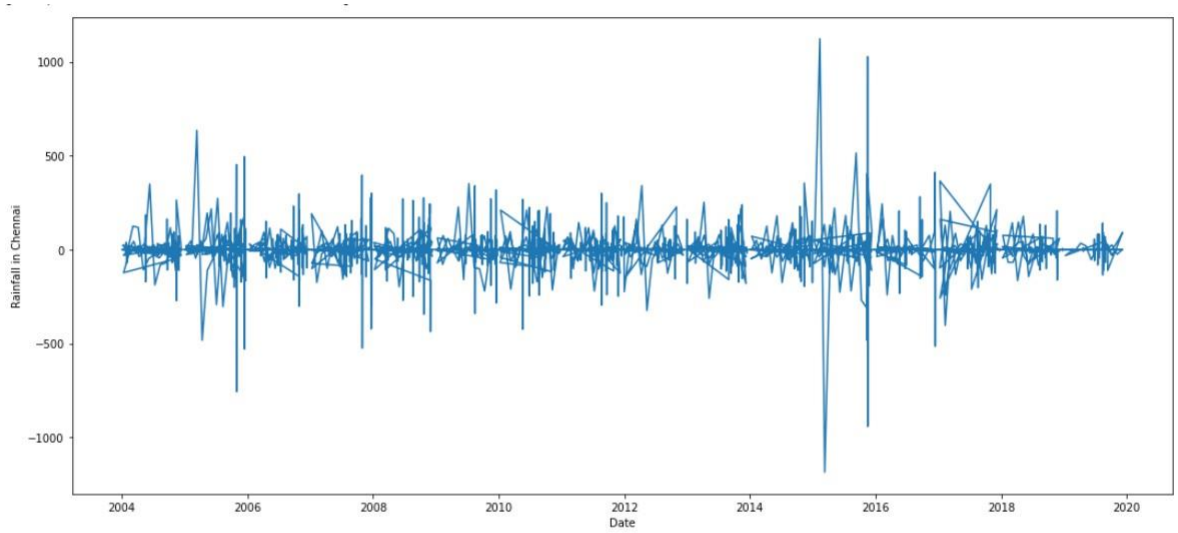
```
[<matplotlib.lines.Line2D at 0xa32ae70>]
```



Plot to check the stationarity of the data

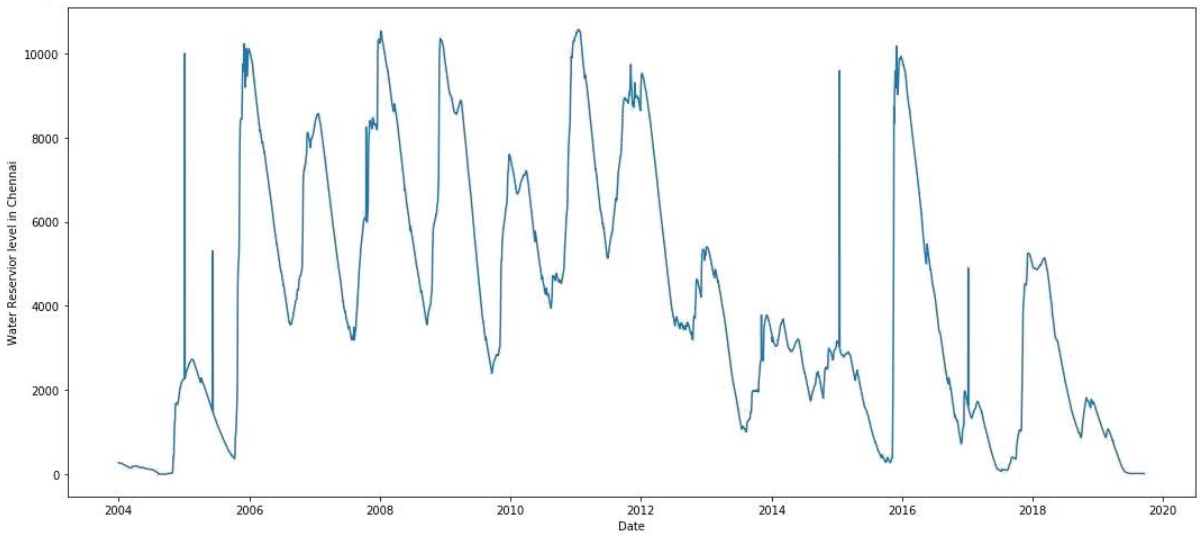


Different components of Time Series Forecasting

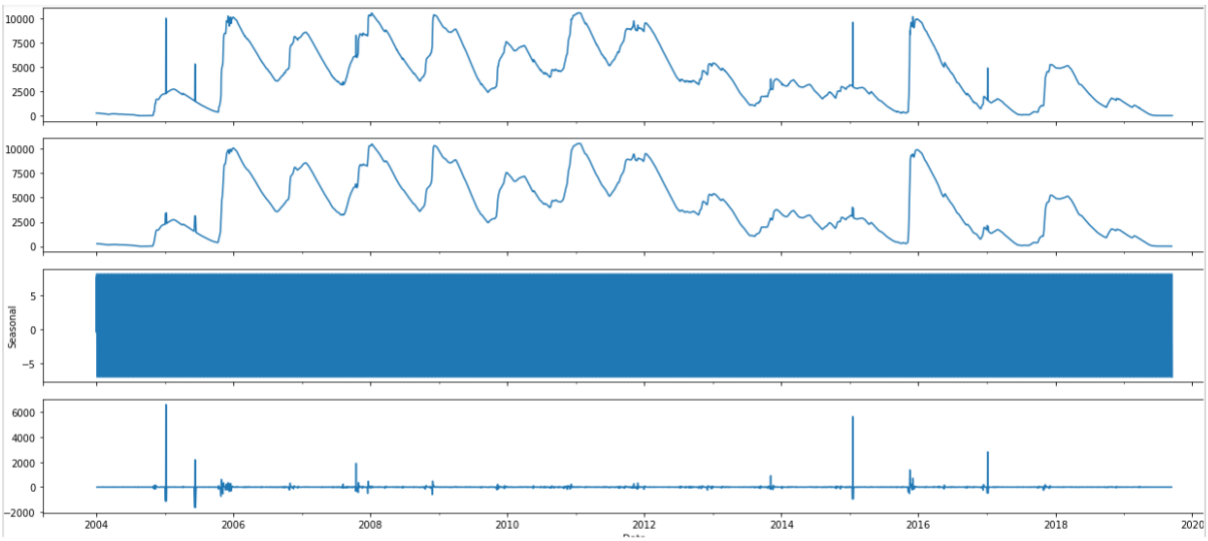


Plot after making the dataset stationary

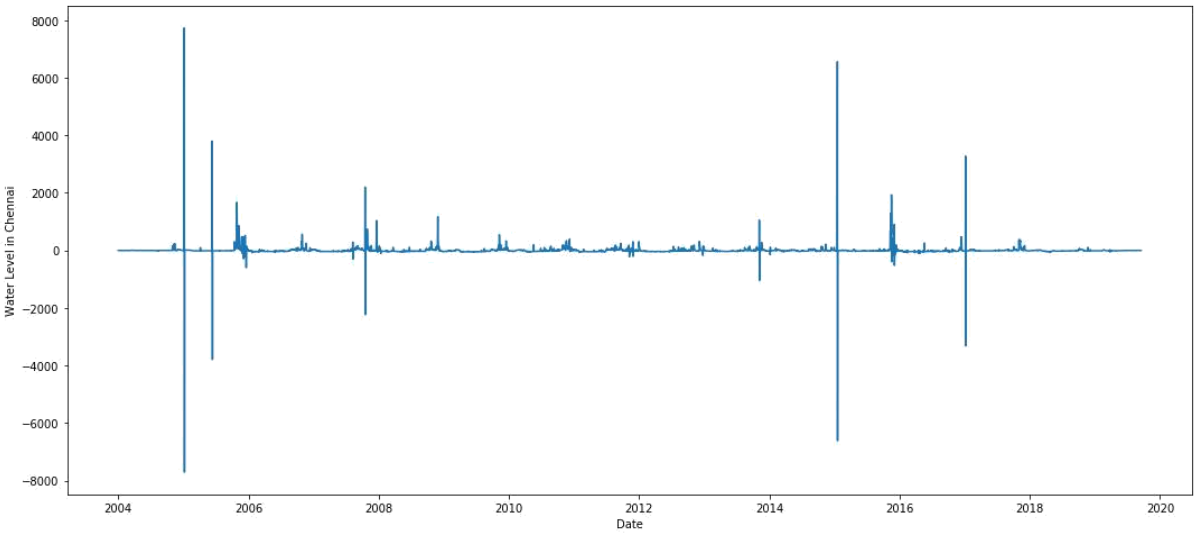
Water Reservoir Prediction:



Plot to check the stationarity of the data



Different components of Time Series Forecasting



Plot after making the dataset stationary

Result

SL NO:	ALGORITHM		RMSE	ACCURACY
1	Time Series Analysis	Rainfall prediction	898.45	71.32084
		Water Reservoir prediction	360.114	80.375

Inference

Since, Time Series is having the less RMSE error as compare to the Multiple Linear Regression hence it is working best on the dataset. The reason for this might be that the time series algorithm takes several factors into consideration like seasonality, trend, cyclicity, irregularity into consideration before making any prediction but multiple linear regression doesn't take these factors into consideration.

Hence **Time Series Forecasting** worked best for this dataset.