Fergusson College, Pune
Department of Statistics

A Statistical View on Water :

The First and Foremost Medicine

A PROJECT ON

THE STATISTICAL VIEW ON WATER : THE FIRST AND FOREMOST MEDICINE

Submitted by:

Gunavant Thakare	214547
Krutika Ahire	214526
Prasad Nikam	214553
Sakshi Kolge	214532
Aditya Akhade	214501
Ketaki Sathe	214549
Snehal Patil	214557

Guided by

Dr. Subhash Shende

FERGUSSON COLLEGE (AUTONOMOUS)
DEPARTMENT OF STATISTICS

2021-22

Group Members:

Sr.no	Name	Roll No.
1.	Gunavant Thakare	214547
2.	Krutika Ahire	214526
3.	Prasad Nikam	214553
4.	Sakshi Kolge	214532
5.	Aditya Akhade	214501
6.	Ketaki Sathe	214549
7.	Snehal Patil	214557

Index

1. Acknowledgement

Any project completed successfully offers a great sense of achievement and satisfaction. The project would remain incomplete if the people who made it possible and whose guidance and encouragement go without mention.

First and foremost, we offer our sincere phrases of thanks to our project mentor and Head of the department **Dr. Subhash Shende Sir**, for providing help and support to carry out the project. We would like to express our gratitude to our **principal**, **Dr. Ravindrasingh Pardeshi Sir** for providing us a congenial environment for completion of our project and also for permitting us to utilize all the necessary facilities in the institution.

We gratefully acknowledge the help and cooperation offered by all the teaching and non-teaching staff members of department of Statistics. A whole hearted acknowledgement to **Mr. Unde Sir** and **Pophale sir** from Drainage department and the staff of the Drainage department, Pune Municipal Corporation for providing us with all the data and stats required for the successful completion of our project.

Finally, we would like to extend a deep appreciation to all those associated with this project for having shared a genuine desire to make a positive contribution to address the challenges associated with every element of this project.

2.Introduction

The Urban development Authorities in the state of Maharashtra have installed manySewage treatment plants (STP) in various cities which aim to remove contaminants from sewage to produce an effluent that is suitable for discharge to the surrounding environment or an intended reuse application, thereby preventing water pollution from raw sewage discharges. The most important task in waste-water treatment is to monitor the variations in the quantity of inflow water in the STP. Based on the generation of waste, population and the respective area, the STPs are designed.

The crucial parameter used in the design of STP is the “Per person organic matter load”. The inflow of wastewater is affected by many factors such as varied climatic conditions, population rise, vacations and tourists. Therefore, forecasting of sewage inflow is necessary to determine the average and peak flow rates, which help in advancing the STP for future conditions. The accurate forecast of STP predicts the plant behaviour to support process designs and controls, improves system reliability, reduces operational cost and endorses optimization of overall performances. Forecasting wastewater inflow is based on the current observed values of inflow recorded at regular intervals of time.

3. Motivation

The Mula-Mutha river is considered as one of the most populated river in India .Over the decades the condition of the rivers have decayed due to discharge of untreated domestic waste water into river owing to inadequate sewerage system ,dumping construction material and open defecation on the river banks . The polluted water affects environment greatly and leads to many waterborne diseases such as Cholera, Diarrhoea , Typhoid, Hepatitis and various skin diseases . Entry of pollutants raises temperature of water, promote the spread of algae blooms and lead to dissolved oxygen depletion which causes death of aquatic animals.To reduce the river water pollution, PMC has installed Sewage Treatment Plants in various parts of city**.**

Motto behind the project is to check the efficiency of STP and to forecast the inflow of wastewater so that in near future due to increase in population the STPs may get overloaded with waste water and PMC might have to install more STPs or to increase the capacities of currently working treatment plants. The treated water is analysed in the lab before it is discharged in river for the parameters like pH, TSS , COD, BOD which have standard set of values . For these values we are testing a hypothesis for all parameters so that we can conclude, whether the treated water is potable or not and lastly we are doing model fitting in which we are searching for best fitting model using which we can classify potability for given data.

4. Terminology :

AR : Auto regression

MA: Moving average

Arima : Auto regressive integrated moving average model

pH = Potential Of Hydrogen

TSS = Total Suspended Solid

COD= Chemical Oxygen Demand

BOD= Biological Oxygen Demand

5.Objectives:

.

1.To study the daily inflow of wastewater in STP.

2.To study the potability of treated water from STP.

3.To classify the water potability by using different models.

4.To fit the best model for our data

6. Study Area:

Pune is a sprawling city in the western Indian state- Maharashtra. Situated 560 metres above sea level on the Deccan plateau, on the banks of the Mula- Mutha river. It is situated at approximately 18° 32" north latitude and 73° 51" east longitude. Pune Municipal Corporation covers geographical area of about 516.18sq.km.

The Pune city corporation has currently 9 working sewage treatment plants (STP). Namely

1. New & Old Naidu
2. Erandwane
3. Baner
4. Tanajiwadi
5. Mundhwa
6. Kharadi
7. Bhairoba
8. Bopodi
9. Vitthalwadi

Figure 1: STP Plant

6.1 Background of the present study and data collection:

The Pune Municipal corporation have installed many STPs till now, out of which 9 are in working condition. However, the frequency and increased quantity of the waste water inflow affects the efficiency of sewage treatment, so it is necessary to predict the inflow changes to have anticipatory control over the wastewater treatment systems.

The Sewage Treatment Plant of *New Naidu* was selected as a study area. The plant is located near Naidu Hospital. Its present capacity is 115 MLD. The sewage generated from the central part of the city is collected at Kasba pumping station & then treated in this STP. The process used in this plant is activated sludge process followed by anaerobic digestion.

To perform Time Series Analysis and forecasting of inflow of waste water, the recorded 304 days of daily inflow data which was read by the flow meter on daily basis (From 1st June 2021 to 31st March 2022) was collected from the New Naidu STP. Average of daily inflow was calculated and the obtained time series is used for further analysis and the ARIMA model is developed.

Figure 2: New Naidu STP Plant

7. Statistical Methods

7.1 Time series – Time series is a series of statistical observations arranged in chronological order where observations are taken at a regular successive intervals or points of time.

The data we used is related to time and as we want to forecast the inflow of sewage in near future that's why we use time series as a statistical tool for forecasting purpose.

Forecasting model :- We use time series analysis to forecast the inflow of waste water in the STP's under PMC.

7.2 Augmented Dickey-Fuller Test:

For forecasting, time series must be stationary hence we need to check whether our time series is stationary or not .

Using Augmented Dickey-Fuller Test (ADF test) to check the stationarity.

7.3 Configuring AR and MA:

Now to check which model is suitable for our data, we plot ACF and PACF.

Two diagnostics plots can be used to choose the p and q parameters of the ARMA or ARIMA. They are:

7.4 Autocorrelation Function (ACF):

The plot summarizes the correlation of an observation with lag values. The x-axis shows the lags and the y-axis shows the correlation coefficient.

Partial Autocorrelation Function (PACF):

The plot summarizes the correlations for an observation with the lag values that is not accounted for by prior lagged observations.

7.5 Box -Jenkins model (ARIMA Model)

The Box-Jenkins model is a mathematical model designed to forecast data ranges based on inputs from a specified time series, also it can analyse several different types of time series data for forecasting purposes. This methodology allows the model to identify trends using autoregression, moving averages and seasonal differencing to generate forecasts. Autoregressive Integrated moving average (ARIMA) models are also called as Box-Jenkins model. A best fitted Autoregressive Integrated Moving Average model is one which can give almost accurate prediction values to achieve success in controlling and planning of wastewater treatment in future.

This model forecasts data using three principles:

a) Autoregression
b) Differencing
c) Moving average

These three principles are known as p, d and q respectively. Each principle used in the Box-Jenkins analysis together, they are collectively shown as ARIMA (p, d, q).

The autoregression (p) process tests the data for value of p for our model. If the data being used is stationary, it can simplify the forecasting process and if it is non-stationary it will needed to be differenced (d). the data is also tested for its moving average fit (q). Overall, initial analysis of the data prepares it for forecasting by determining the parameters (p, d, q), which are then applied to develop a forecast.

- **Autoregressive Integrated Moving Average model:**

ARIMA model is used to forecast the values using past data. An ARIMA model can be understood by outlining each of its component as follows:

AR: *Autoregression.* A model that uses the dependent relationship between an observation and some number of lagged observations. A stochastic process called as Autoregressive process of order p is defined as follows:

If $\{Z_t\}$ – errors are purely random with mean= 0 and variance = σ^2

Then process $\mathbf{Y_t}$ is said to be autoregressive process of order t if it is given by:

$$\mathbf{Y_t = \alpha_1 Y_{t-1} + \alpha_2 Y_{t-2} + \ldots + \alpha_p Y_{t-p} + Z_t} \qquad ; \ |\alpha_i| < 1 \ , \text{ for all } i=1,2,\ldots,p$$

AR(1) – it is sometimes called as Markovian process. It is defined as follows for p=1:

$$\mathbf{Y_t = \alpha_1 Y_{t-1} + Z_t} \qquad |\alpha| < 1$$

I: *Integrated*. The use of differencing for data observations (i.e. subtracting an observation from an observation at the previous time step) in order to make the time series stationary.

MA: *Moving Average*. A model that uses the dependency between an observation and residual errors from a moving average model applied to lagged observations. The model of the MA is given by:

$$\mathbf{Y_t = \beta_0 Z_t + \beta_1 Z_{t-1} + \beta_2 Z_{t-2} + \ldots + \beta_q Z_{t-q}}$$

Where, Z: residuals of the past values

Y: value with which we correlate the past residuals

q: lag value

β: coefficients of residuals

Each of these components are explicitly specified in the model as a parameter. A standard notation is used as ARIMA (p, d, q) where the parameters are substituted with integer values to quickly indicate the specific ARIMA model by using AIC criteria. The parameters of the ARIMA model are defined as follows:

p: the number of lagged observations included in the autoregressive model.

d: the number of times that the time series is differenced, also called the degree of differencing.

q: the number of lagged observation of the moving average model.

Aic is **Akaike Information Criterion-** it used for evaluating how well a model fits the data. In statistics, it used to compare different possible models and determine which one is the best fit for the data. The formula for the AIC is :

$$AIC = 2K - 2\ln(L)$$

Where, K: the number of independent variables used

L: the log-likelihood estimate

The less AIC value indicates less information of our dataset is loss. Hence, the model with less AIC value is the best fit. Therfore, model with AIC=1889.11 value is the best fit for our data.

Model Development:

ARIMA model used in this study consists of the following steps: Identification, Diagnostic checking and Forecasting. The model was estimated using the R software (version 4.1.2).

We need to install the packages and libraries as follows:

1. readxl
2. tseries
3. forecast
4. ggplot2

R code:

#Importing data set

```
View(flow)

library(readxl)

flow <- read_excel("C:/Users/ASUS/Downloads/flow.xlsx")

View(flow)
```

Library install for model forecasting

```
library(tseries)

library(forecast)

attach(flow)
```

Split data as train and test

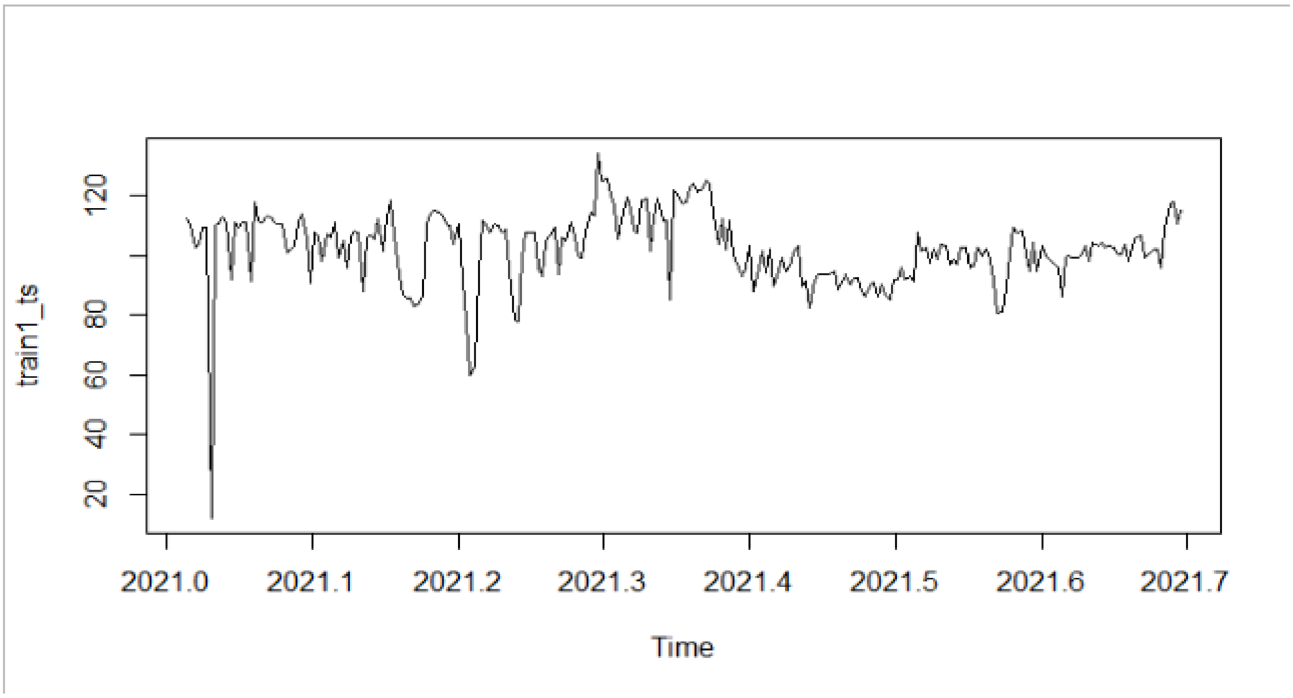
```
train1=flow[1:250,2]

train1=unlist(train1)

train1=as.numeric(train1)

train1_ts=ts(train1,start=c(2021,06,01),frequency = 365)

train1_ts
```

 # Test data

```
test1=flow[251:304,2]

test1=unlist(test1)

test1=as.numeric(test1)

test1_ts=ts(test1,start=c(2022,02,06),frequency = 365)

test1_ts
```

To plot time series

plot(train1_ts)



To check stationarity

adf.test(train1_ts)

Augmented Dickey-Fuller Test

data: train1_ts

Dickey-Fuller = -5.1182, Lag order = 6, p-value = 0.01

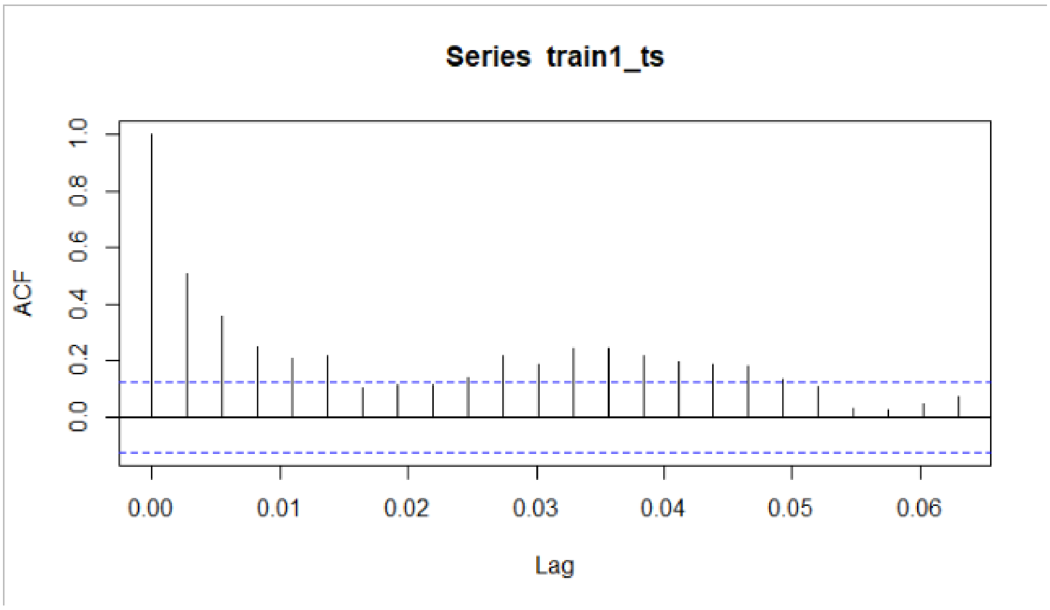
alternative hypothesis: stationary

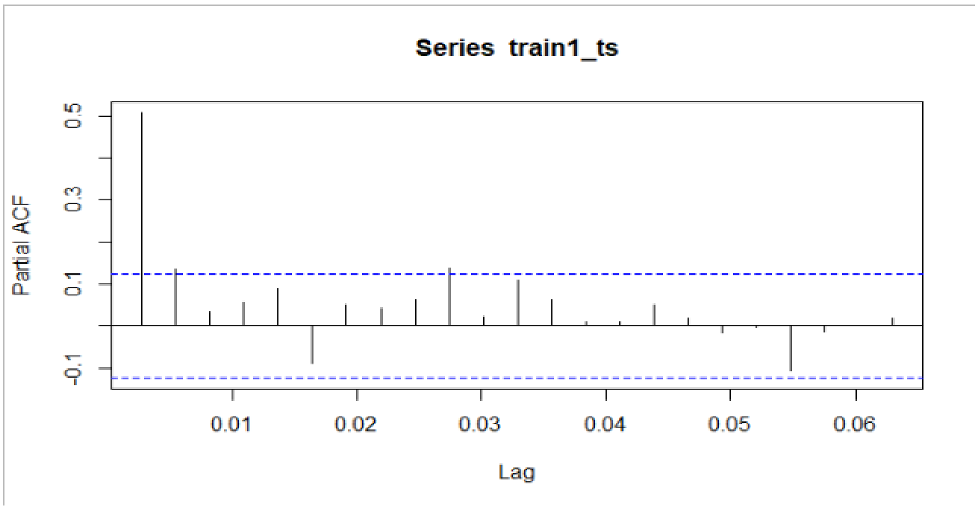
Conclusion : By using ADF test we conclude that our time series is stationary by using test statistics.

As our time series is stationary we can apply forecasting model for our data.

To check collinearity

acf(train1_ts)





pacf(train1_ts)

The ACF and PACF plots should be considered together to define the process. From the above fig we observed that, both the graphs shows geometrical decreasing pattern hence mixed ARIMA model is considered for modelling

TO FIND BEST MODEL FIT FOR OUR DATA

```
z=auto.arima(train1_ts,ic='aic',trace=TRUE)
z
```

```
Fitting models using approximations to speed things up...

ARIMA(2,0,2) with non-zero mean : Inf
ARIMA(0,0,0) with non-zero mean : 1965.568
ARIMA(1,0,0) with non-zero mean : 1892.571
ARIMA(0,0,1) with non-zero mean : 1915.624
ARIMA(0,0,0) with zero mean     : 3028.958
ARIMA(2,0,0) with non-zero mean : 1890.713
ARIMA(3,0,0) with non-zero mean : 1893.208
ARIMA(2,0,1) with non-zero mean : 1890.507
ARIMA(1,0,1) with non-zero mean : 1889.078
ARIMA(1,0,2) with non-zero mean : 1890.839
ARIMA(0,0,2) with non-zero mean : 1898.61
ARIMA(1,0,1) with zero mean     : Inf

Now re-fitting the best model(s) without approximations...

ARIMA(1,0,1) with non-zero mean : 1889.108

Best model: ARIMA(1,0,1) with non-zero mean
```

To have prior knowledge about the inflow rate of water to the STP, we have forecasted the values. The best fitted ARIMA(1, 0, 1) was used to forecast the inflow rate next 84 days (54 observations of test set and and 30 for prediction purpose). The inflow values obtained do not show so much fluctuation in inflow rate as we have daily data .The observed and predicted values with the confidential limits are shown in the table.

```
m=arima(train1_ts,order = c(1,0,1))
m
```

```
Call:
  arima(x = train1_ts, order = c(1, 0, 1))

Coefficients:
  ar1          ma1          intercept
0.7254      -0.3002      102.4784
s.e.  0.0864   0.1231     1.6644

sigma^2 estimated as 108.3:  log likelihood = -940.55,  aic = 1889.11
```

```
a=forecast(m,h=84)
> a
```

Point	Forecast	Lo 80	Hi 80	Lo 95	Hi 95
2021.6986	109.7196	96.38116	123.0580	89.32023	130.1189
2021.7014	107.7312	93.23697	122.2254	85.56421	129.8981
2021.7041	106.2888	91.22196	121.3556	83.24607	129.3315
2021.7068	105.2425	89.88290	120.6021	81.75204	128.7329
2021.7096	104.4835	88.97208	119.9949	80.76084	128.2061
2021.7123	103.9329	88.34220	119.5236	80.08898	127.7768
2021.7151	103.5335	87.90124	119.1658	79.62602	127.4410
2021.7178	103.2438	87.58970	118.8979	79.30292	127.1847
2021.7205	103.0336	87.36806	118.6992	79.07521	126.9921
2021.7233	102.8812	87.20957	118.5528	78.91352	126.8488
2021.7260	102.7706	87.09581	118.4454	78.79808	126.7431
2021.7288	102.6904	87.01392	118.3668	78.71530	126.6654
2021.7315	102.6322	86.95484	118.3095	78.65577	126.6086
2021.7342	102.5900	86.91217	118.2678	78.61285	126.5671
2021.7370	102.5593	86.88130	118.2374	78.58185	126.5368
2021.7397	102.5371	86.85896	118.2153	78.55944	126.5148
2021.7425	102.5210	86.84278	118.1992	78.54323	126.4988
2021.7452	102.5093	86.83106	118.1876	78.53148	126.4872
2021.7479	102.5008	86.82256	118.1791	78.52298	126.4787
2021.7507	102.4947	86.81640	118.1730	78.51681	126.4726
2021.7534	102.4902	86.81193	118.1685	78.51234	126.4681
2021.7562	102.4870	86.80869	118.1653	78.50910	126.4649
2021.7589	102.4846	86.80634	118.1630	78.50675	126.4625
2021.7616	102.4829	86.80464	118.1613	78.50505	126.4608
2021.7644	102.4817	86.80340	118.1600	78.50381	126.4596
2021.7671	102.4808	86.80251	118.1591	78.50291	126.4587
2021.7699	102.4802	86.80186	118.1585	78.50226	126.4581
2021.7726	102.4797	86.80139	118.1580	78.50179	126.4576
2021.7753	102.4794	86.80104	118.1577	78.50145	126.4573
2021.7781	102.4791	86.80080	118.1574	78.50120	126.4570

2021.7808	102.4789	86.80062	118.1572	78.50102	126.4568
2021.7836	102.4788	86.80049	118.1571	78.50089	126.4567
2021.7863	102.4787	86.80039	118.1570	78.50080	126.4566
2021.7890	102.4786	86.80032	118.1569	78.50073	126.4565
2021.7918	102.4786	86.80027	118.1569	78.50068	126.4565
2021.7945	102.4785	86.80024	118.1568	78.50064	126.4564
2021.7973	102.4785	86.80021	118.1568	78.50062	126.4564
2021.8000	102.4785	86.80019	118.1568	78.50060	126.4564
2021.8027	102.4785	86.80018	118.1568	78.50058	126.4564
2021.8055	102.4785	86.80017	118.1568	78.50057	126.4564
2021.8082	102.4785	86.80016	118.1568	78.50057	126.4564
2021.8110	102.4785	86.80015	118.1568	78.50056	126.4564
2021.8137	102.4785	86.80015	118.1568	78.50056	126.4564
2021.8164	102.4785	86.80015	118.1568	78.50055	126.4564
2021.8192	102.4785	86.80015	118.1568	78.50055	126.4564
2021.8219	102.4785	86.80014	118.1568	78.50055	126.4564
2021.8247	102.4785	86.80014	118.1568	78.50055	126.4564
2021.8274	102.4784	86.80014	118.1568	78.50055	126.4564
2021.8301	102.4784	86.80014	118.1568	78.50055	126.4564
2021.8329	102.4784	86.80014	118.1568	78.50055	126.4563
2021.8356	102.4784	86.80014	118.1568	78.50055	126.4563
2021.8384	102.4784	86.80014	118.1568	78.50055	126.4563
2021.8411	102.4784	86.80014	118.1568	78.50055	126.4563
2021.8438	102.4784	86.80014	118.1568	78.50055	126.4563
2021.8466	102.4784	86.80014	118.1568	78.50055	126.4563
2021.8493	102.4784	86.80014	118.1568	78.50055	126.4563
2021.8521	102.4784	86.80014	118.1568	78.50055	126.4563
2021.8548	102.4784	86.80014	118.1568	78.50055	126.4563
2021.8575	102.4784	86.80014	118.1568	78.50055	126.4563
2021.8603	102.4784	86.80014	118.1568	78.50055	126.4563
2021.8630	102.4784	86.80014	118.1568	78.50055	126.4563
2021.8658	102.4784	86.80014	118.1568	78.50055	126.4563
2021.8685	102.4784	86.80014	118.1568	78.50055	126.4563
2021.8712	102.4784	86.80014	118.1568	78.50055	126.4563

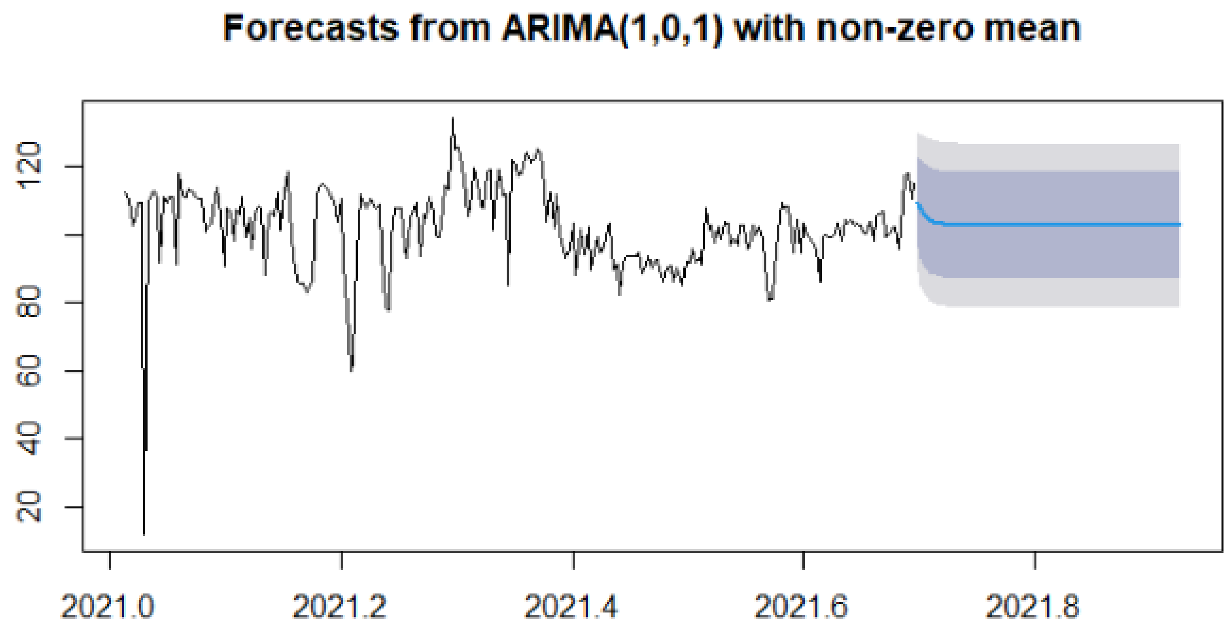
```
2021.8740       102.4784 86.80014 118.1568 78.50055 126.4563
2021.8767       102.4784 86.80014 118.1568 78.50055 126.4563
2021.8795       102.4784 86.80014 118.1568 78.50055 126.4563
2021.8822       102.4784 86.80014 118.1568 78.50055 126.4563
2021.8849       102.4784 86.80014 118.1568 78.50055 126.4563
2021.8877       102.4784 86.80014 118.1568 78.50055 126.4563
2021.8904       102.4784 86.80014 118.1568 78.50055 126.4563
2021.8932       102.4784 86.80014 118.1568 78.50055 126.4563
2021.8959       102.4784 86.80014 118.1568 78.50055 126.4563
2021.8986       102.4784 86.80014 118.1568 78.50055 126.4563
2021.9014       102.4784 86.80014 118.1568 78.50055 126.4563
2021.9041       102.4784 86.80014 118.1568 78.50055 126.4563
2021.9068       102.4784 86.80014 118.1568 78.50055 126.4563
2021.9096       102.4784 86.80014 118.1568 78.50055 126.4563
2021.9123       102.4784 86.80014 118.1568 78.50055 126.4563
2021.9151       102.4784 86.80014 118.1568 78.50055 126.4563
2021.9178       102.4784 86.80014 118.1568 78.50055 126.4563
2021.9205       102.4784 86.80014 118.1568 78.50055 126.4563
2021.9233       102.4784 86.80014 118.1568 78.50055 126.4563
2021.9260       102.4784 86.80014 118.1568 78.50055 126.4563
>
```

```
b=accuracy(test1,a)
b
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	-0.0406814	10.40802	6.742089	-3.603815	9.574069	1.031275	0.007472207
Test set	6.9014058	13.71923	11.185635	5.075130	10.149107	1.710964	NA

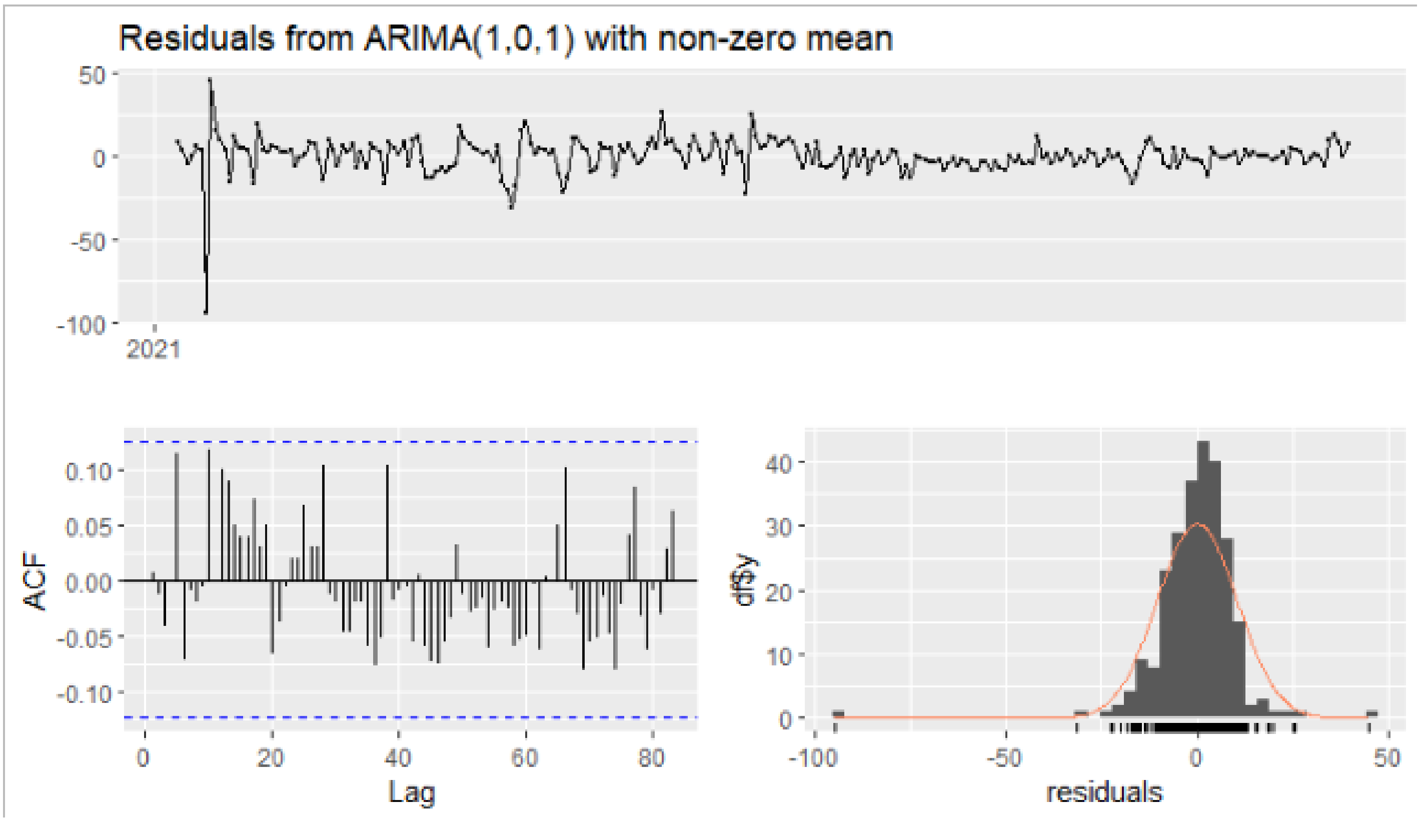
#forecasted time series

plot(a)



- **Residual Analysis :**

Checkresiduals(z)



Ljung-Box test

data: Residuals from ARIMA(1,0,1) with non-zero mean

Q* = 39.635, df = 47, p-value = 0.7684

Model df: 3. Total lags used: 50

Augmented Dickey-Fuller Test

data: z$residuals

Dickey-Fuller = -7.0107, Lag order = 6, p-value = 0.01

alternative hypothesis: stationary

By above graph and using adf test we conclude that the residuals are stationary and are uncorrelated and also we interpret that residuals follow normality.

7.6 Holt Winters Forecasting Method:

It is used for exponential smoothing for level,trend and seasonal components . Trend and seasonal components are absent in our data so we are going for Simple exponential smoothing for removing irregularities . Gamma and beta implies coefficients of trend smoothing and seasonal smoothing.

R code :

s=HoltWinters(train1_ts,gamma=F,beta=F)

> s

Holt-Winters exponential smoothing without trend and without seasonal component

Output:

HoltWinters(x = train1_ts, beta = F, gamma = F)

Smoothing parameters:

alpha: 0.3779311

beta : FALSE

gamma: FALSE

Coefficients:

[,1]

a 112.9779

Model:

$$\hat{Y}_{t+1} = \alpha Y_t + (1-\alpha)\hat{Y}_t$$

$$\hat{Y}_{t+1} = 0.3779 Y_t + 0.6221 \hat{Y}_t$$

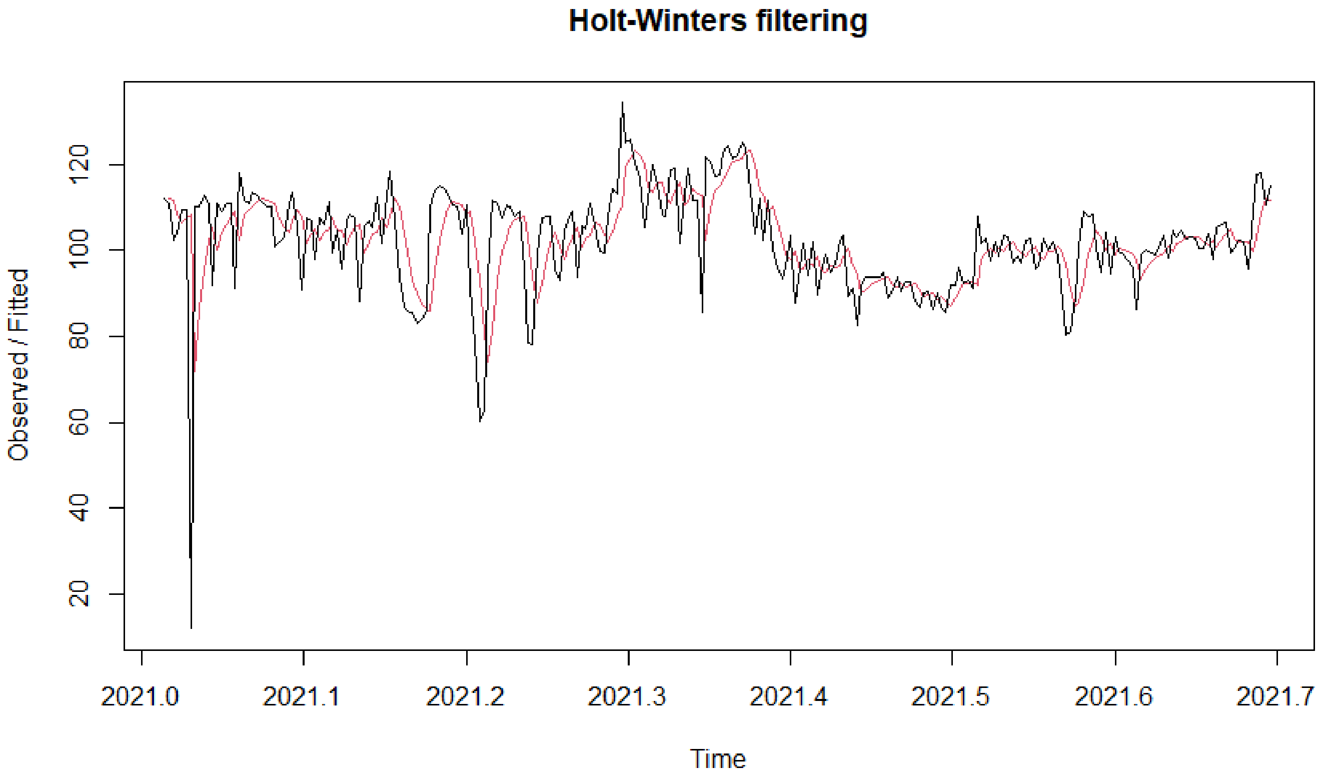
```
> q=forecast(s,h=84)
> q
```

Point	Forecast	Lo 80	Hi 80	Lo 95	Hi 95
2021.6986	112.9779	98.99241	126.9634	91.58894	134.3668
2021.7014	112.9779	98.02695	127.9288	90.11240	135.8434
2021.7041	112.9779	97.12016	128.8356	88.72558	137.2302
2021.7068	112.9779	96.26249	129.6933	87.41389	138.5419
2021.7096	112.9779	95.44673	130.5091	86.16629	139.7895
2021.7123	112.9779	94.66727	131.2885	84.97421	140.9816
2021.7151	112.9779	93.91967	132.0361	83.83086	142.1249
2021.7178	112.9779	93.20031	132.7555	82.73069	143.2251
2021.7205	112.9779	92.50621	133.4496	81.66915	144.2866
2021.7233	112.9779	91.83488	134.1209	80.64245	145.3133
2021.7260	112.9779	91.18422	134.7716	79.64735	146.3084
2021.7288	112.9779	90.55244	135.4033	78.68112	147.2747
2021.7315	112.9779	89.93797	136.0178	77.74137	148.2144
2021.7342	112.9779	89.33947	136.6163	76.82604	149.1297
2021.7370	112.9779	88.75575	137.2000	75.93333	150.0225
2021.7397	112.9779	88.18577	137.7700	75.06162	150.8942
2021.7425	112.9779	87.62861	138.3272	74.20951	151.7463
2021.7452	112.9779	87.08343	138.8724	73.37573	152.5801
2021.7479	112.9779	86.54950	139.4063	72.55915	153.3966
2021.7507	112.9779	86.02614	139.9296	71.75874	154.1970
2021.7534	112.9779	85.51275	140.4430	70.97358	154.9822
2021.7562	112.9779	85.00878	140.9470	70.20283	155.7530
2021.7589	112.9779	84.51373	141.4420	69.44572	156.5101
2021.7616	112.9779	84.02715	141.9286	68.70156	157.2542
2021.7644	112.9779	83.54862	142.4072	67.96970	157.9861
2021.7671	112.9779	83.07774	142.8780	67.24955	158.7062
2021.7699	112.9779	82.61416	143.3416	66.54057	159.4152
2021.7726	112.9779	82.15755	143.7982	65.84225	160.1135
2021.7753	112.9779	81.70761	144.2482	65.15413	160.8017
2021.7781	112.9779	81.26406	144.6917	64.47576	161.4800

2021.7808	112.9779	80.82662	145.1292	63.80676	162.1490
2021.7836	112.9779	80.39505	145.5607	63.14674	162.8090
2021.7863	112.9779	79.96913	145.9867	62.49534	163.4604
2021.7890	112.9779	79.54863	146.4072	61.85225	164.1035
2021.7918	112.9779	79.13336	146.8224	61.21714	164.7386
2021.7945	112.9779	78.72312	147.2327	60.58973	165.3660
2021.7973	112.9779	78.31773	147.6381	59.96975	165.9860
2021.8000	112.9779	77.91704	148.0387	59.35694	166.5988
2021.8027	112.9779	77.52087	148.4349	58.75105	167.2047
2021.8055	112.9779	77.12907	148.8267	58.15186	167.8039
2021.8082	112.9779	76.74152	149.2143	57.55914	168.3966
2021.8110	112.9779	76.35806	149.5977	56.97270	168.9831
2021.8137	112.9779	75.97858	149.9772	56.39233	169.5635
2021.8164	112.9779	75.60296	150.3528	55.81786	170.1379
2021.8192	112.9779	75.23107	150.7247	55.24910	170.7067
2021.8219	112.9779	74.86280	151.0930	54.68590	171.2699
2021.8247	112.9779	74.49807	151.4577	54.12808	171.8277
2021.8274	112.9779	74.13676	151.8190	53.57550	172.3803
2021.8301	112.9779	73.77877	152.1770	53.02801	172.9278
2021.8329	112.9779	73.42403	152.5318	52.48548	173.4703
2021.8356	112.9779	73.07244	152.8833	51.94777	174.0080
2021.8384	112.9779	72.72392	153.2319	51.41476	174.5410
2021.8411	112.9779	72.37840	153.5774	50.88632	175.0695
2021.8438	112.9779	72.03579	153.9200	50.36234	175.5934
2021.8466	112.9779	71.69602	154.2598	49.84271	176.1131
2021.8493	112.9779	71.35902	154.5968	49.32733	176.6285
2021.8521	112.9779	71.02474	154.9310	48.81608	177.1397
2021.8548	112.9779	70.69309	155.2627	48.30887	177.6469
2021.8575	112.9779	70.36403	155.5918	47.80561	178.1502
2021.8603	112.9779	70.03749	155.9183	47.30621	178.6496
2021.8630	112.9779	69.71341	156.2424	46.81058	179.1452
2021.8658	112.9779	69.39174	156.5640	46.31863	179.6372
2021.8685	112.9779	69.07243	156.8834	45.83029	180.1255
2021.8712	112.9779	68.75543	157.2004	45.34547	180.6103

2021.8740	112.9779	68.44068	157.5151	44.86410	181.0917
2021.8767	112.9779	68.12814	157.8276	44.38611	181.5697
2021.8795	112.9779	67.81776	158.1380	43.91143	182.0444
2021.8822	112.9779	67.50950	158.4463	43.43999	182.5158
2021.8849	112.9779	67.20332	158.7525	42.97172	182.9841
2021.8877	112.9779	66.89917	159.0566	42.50657	183.4492
2021.8904	112.9779	66.59702	159.3588	42.04447	183.9113
2021.8932	112.9779	66.29682	159.6590	41.58535	184.3704
2021.8959	112.9779	65.99854	159.9572	41.12917	184.8266
2021.8986	112.9779	65.70214	160.2536	40.67587	185.2799
2021.9014	112.9779	65.40759	160.5482	40.22539	185.7304
2021.9041	112.9779	65.11485	160.8409	39.77769	186.1781
2021.9068	112.9779	64.82389	161.1319	39.33270	186.6231
2021.9096	112.9779	64.53468	161.4211	38.89039	187.0654
2021.9123	112.9779	64.24719	161.7086	38.45071	187.5051
2021.9151	112.9779	63.96138	161.9944	38.01360	187.9422
2021.9178	112.9779	63.67722	162.2786	37.57903	188.3768
2021.9205	112.9779	63.39470	162.5611	37.14695	188.8088
2021.9233	112.9779	63.11378	162.8420	36.71731	189.2385
2021.9260	112.9779	62.83443	163.1214	36.29008	189.6657

> plot(s)



accuracy(q,test1)

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	0.008053699	10.89100	6.852565	-3.508512	9.765141	1.048174	0.1204965
Test set	-3.109706901	12.01853	9.698354	-4.130214	9.519942	1.483468	NA

Results and Conclusions:

By using arima and holtwinter method we forecasted the inflow rate. RMSE values for testing set of both the models were compared. RMSE value of arima model is greater than that of Holt-Winter, so we concluded, Holt-winter method as best model for forecasting our data.

8.Exploratory data Analysis:

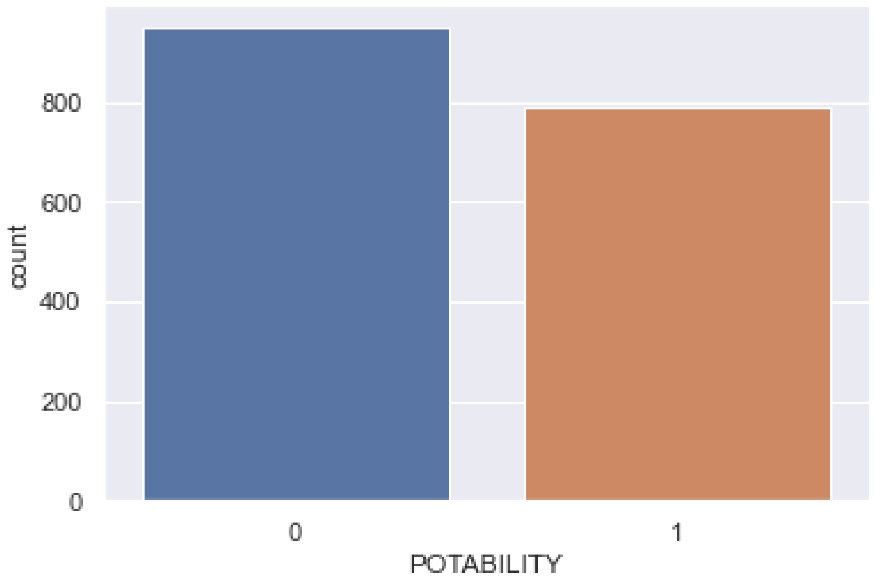


Figure 3: Count plot for Potability

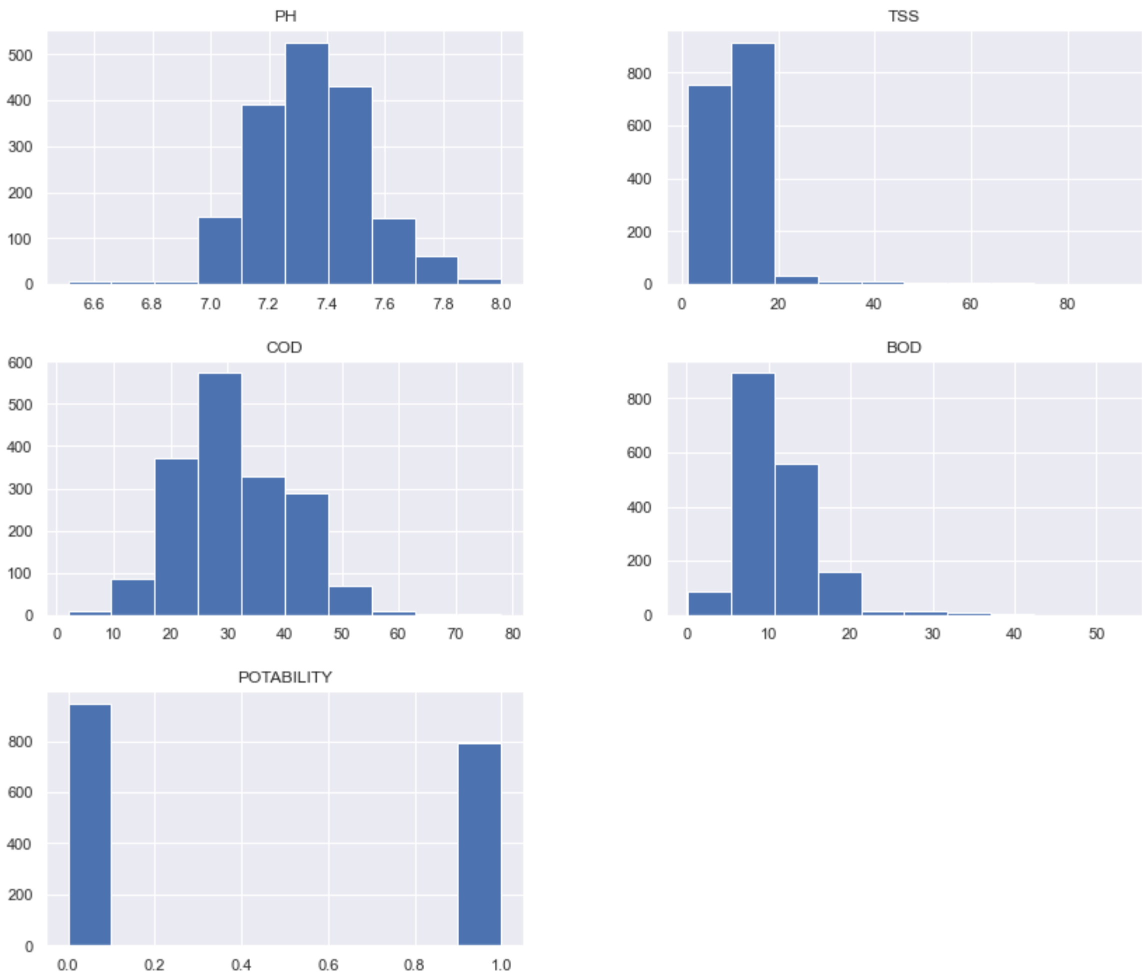


Figure 4: Histogrammic representation

#countplot for potability

```
sns.countplot(data['POTABILITY'])

plt.show()
```

#graphical representation of parameters using histogram

```
data.hist(figsize=(14,12))

plt.show()
```

Interpretation:

1. (Figure 3) data['POTABILITY'].value_counts()

```
0    948
1    791
Name: POTABILITY, dtype: int64
```

2. (Figure 4): By looking at the histograms of all the parameters, we can't say about the normality of the parameters. So we have checked by Shapiro-Wilk test. (refer page no 27)

9.Hypothesis Testing-

Hypothesis testing is one of the most important concepts in Statistics which is heavily used by Statisticians, Machine Learning Engineers, and Data Scientists. In hypothesis testing, Statistical tests are used to check whether the null hypothesis is rejected or not rejected. These Statistical tests assume a null hypothesis of no relationship or no difference between groups.

Parametric and Non-Parametric Test-

Parametric tests are those tests for which we have prior knowledge of the population distribution (i.e. normal), or if not then we can easily approximate it to a normal distribution which is possible with the help of the Central Limit Theorem.

In **Non-Parametric tests**, we don't make any assumption about the parameters for the given population or the population we are studying. In fact, these tests don't depend on the population.

As our project aim is to check whether working of STP is efficient or not i.e treated water is potable or not potable. So for testing this claim we have to perform the hypothesis testing for each parameter **(PH, TSS, COD, BOD)** . The standard ranges for the given parameter for which the potability of water are given by the STP itself and are as follows.

pH=(6.5,8.5) (Potential Of Hydrogen)

TSS(<20) (Total Suspended Solid)

COD(<50) (Chemical Oxygen Demand)

BOD(<10) (Biological Oxygen Demand)

If all the parameters value are in the specified range then we can conclude that the water is potable, otherwise it is not potable.

So for checking this claim we perform the hypothesis testing for a given sample data. The first step to proceed by parameter testing we want to check whether the given sample is coming from normal population or not.

Let's check the normality of each parameter by Shapiro test.

Import Data set:

```
library(readxl)

stpexcelfinal <- read_excel("C:/Users/ASUS/Desktop/stpexcelfinal.xlsx")

View(stpexcelfinal)

shapiro.test(stpexcelfinal$PH)
```

9.1Shapiro-Wilk normality test (l.o.s.=5%)

To test : H_0 = PH is normally distributed.

H_1 = PH is not normally distributed.

data: stpexcelfinal$PH

W = 0.98505, p-value = 1.768e-12

> shapiro.test(stpexcelfinal$TSS)

Shapiro-Wilk normality test

data: stpexcelfinal$TSS

W = 0.64475, p-value < 2.2e-16

> shapiro.test(stpexcelfinal$BOD)

Shapiro-Wilk normality test

data: stpexcelfinal$BOD

W = 0.85976, p-value < 2.2e-16

> shapiro.test(stpexcelfinal$COD)

Shapiro-Wilk normality test

data: stpexcelfinal$COD

W = 0.98289, p-value = 1.491e-13

Conclusion:

As p-value for each test is less than 0.05 so we reject null hypothesis at 5% l.o.s.

By seeing the output of shapiro test we can easily conclude that the data does not follow normal distribution. So we should go with corresponding non-parametric test.

There exist a suitable non parametric test for checking median(median is measure of central tendency for non parametric test) value which is known as one sample Wilcoxon signed rank test.

The following is the information about the test.

9.2 Wilcoxon's Signed Rank Test:

It is one of the non-parametric tests used to test the location of a population based on a sample of data or to compare the locations of two populations using two samples. The sign test for location utilizes only the signs of difference of observations from hypothesized median (or the difference of observations in the pairs) without considering the magnitude of the difference. If the information regarding magnitude is available then a test procedure that takes into account the size and the relative magnitude of the differences as well, is expected to give a better performance. Wilcoxon's signed rank test is based on this consideration. However, the better performance is obtained at the cost of additional assumption of symmetry of the population about true median.

Testing Problem:

Suppose $X_1, \ldots$, Xn is a random sample of size n from the distribution of random variable X. Let $F_x(.)$ be the distribution function and M be the median of X.it is required to test the hypothesis.

$H_0 : M = M_0$ against one of the alternatives,

1) $H_1 : M > M_0$
2) $H_1 : M < M_0$
3) $H_1 : M \neq M_0$

Assumptions:

1. $F_x(.)$ is continuous
2. $F_x(.)$ is symmetric about M.

Test Statistic:

Let T^+ = sum of positive ranks

T^- = sum of negative ranks.

Note that, T^+ and T^- both are non-negative numbers and

$$T^+ + T^- = \sum_{i=1}^{n} \frac{n(n+1)}{2}$$

Under H_0, the distributions of T^+ and T^- are identical and each distribution is symmetric about the common mean n(n+1)/4. So, any one of the T^+ or T^- can be used as the test statistic.If the alternative hypopthesis is :
$H_1 : M > M_0$ then test statistics is T^-
$H_1 : M < M_0$ then test statistics is T^+
$H_1 : M \neq M_0$ then test statistics is $\min\{T^+, T^-\}$

Decision Rule:

If $H_1 = M < M_0$ then smaller value of T^+ favours the alternative hypothesis i.e Reject H_0 if $T^+ <= T\alpha,n$ where $T\alpha,n$ is lower α % point of T^+ .

If $H_1 = M > M_0$ the larger value of T^+ favours the alternative hypothesis i.e Reject H_0 if $T^+ >= T\alpha,n$ where $T\alpha,n$ is upper α % point of T^+ .

If $H_1 = M \neq M_0$ too larger value of T^+ too smaller values of T^+ favours the alternative hypothesis i.e Reject H_0 if $T^+ >= T\alpha_{/2},n$ or $T^+ <= T'\alpha_{/2},n$.

R code: **Wilcoxon signed rank test**

```
> a=wilcox.test(stpexcelfinal$PH,mu=6.5,alternative ="greater")
> a
data:  stpexcelfinal$PH
V = 1512930, p-value < 2.2e-16
alternative hypothesis: true location is greater than 6.5

> a=wilcox.test(stpexcelfinal$PH,mu=8.5,alternative ="less")
> a
data:  stpexcelfinal$PH
V = 0, p-value < 2.2e-16
alternative hypothesis: true location is less than 8.5

> a=wilcox.test(stpexcelfinal$TSS,mu=20,alternative ="less")
> a
data:  stpexcelfinal$TSS
V = 67830, p-value < 2.2e-16
alternative hypothesis: true location is less than 20

> a=wilcox.test(stpexcelfinal$COD,mu=50,alternative ="less")
> a
data:  stpexcelfinal$COD
V = 8378, p-value < 2.2e-16
alternative hypothesis: true location is less than 50

> a=wilcox.test(stpexcelfinal$BOD,mu=10,alternative ="less")
> a
data:  stpexcelfinal$BOD
V = 561760, p-value = 5.407e-08
alternative hypothesis: true location is less than 10
```

As all the null hypothesis is rejected for the given parameters, we can conclude that all the parameters are in suitable range in short the treated water of STP is potable for the given sample at this stage. But in future the decision may or may not be same as it is in present because it depends on the given sample.

Note: Suppose in the future if we get similar data and we want to Check the given water is potable or not we can use appropriate machine learning model for checking purpose.

9.3 Benefits of using model over the hypothesis:

The hypothesis is possible if and only the given sample is considerably large. sometimes it is very costly to get the large sample but if you have given only one data point we can't use the hypothesis but we can use the model to get idea about the census.

Scope of using the machine learning models in day to day life

We can easily see that in summer season some villages face the water problem. Sometimes water provided to them maybe collected from river or from lake are from some well which is not tested chemically whether it is potable or not because by the naked eyes we can't figure out the water as potable or not.

So if we have provided the parameters value it will be very difficult for human being to check each value in the parameter space and give conclusion about the sample. Sometimes it will reject the sample even it satisfy all the require conditions. So to increase the efficiency of work we use the machine learning models.

10. MACHING LEARNING:

10.1 What is Machine Learning?

Machine learning (ML) is basically the study of computer algorithms that can improve automatically through experience and by the use of past data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to make decisions and test its accuracy with the help of test data. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, computer vision, etc.

Nowadays the demand of statistics in machine learning is increasing day by day. In models, Statistical methods are required in the preparation of train data and test data and also to check the accuracy of the models.

This includes:

- Outlier detection.
- Missing value imputation.
- Data sampling.
- Data scaling.
- Variable encoding.

This all can be done in machine learning by applying the proper statistical tools.

10.2 Why we use it?

The response variable of our data was in the form of classification type. So we classify our data in two groups namely potable water or non potable water as like a binary variable.

Potable water=1

Non potable water=0

There are also some classification models that are used in machine learning.
Example of those models are
1.Logistic Regression.
2.K-Nearest Neighbor
3.Support Vector Machines
4.Kernel SVM
5.Naive Bayes
6.Decision Tree Classification
7.Random Forest Classification
8.ANN
9.CNN

we want to develop a model that can predict the values of potability. Our focus is on both accuracy of the predictions and interpretability of the model.
Therefore we have choose the models that suits our data best. We will evaluate three different models covering the complexity spectrum.
1.Logistic Regession.
2.K-Nearest Neighbors.
3.Decision tree

To head-start the ML process, the cleaning of data is must.

10.3 Why data cleaning is important?

To reduce the errors and to increase the efficiency of model we need to clean our data.

Lets clean our data set using python:

Codes for cleaning data :

```
import pandas as pd                    #to import and analyse data
import numpy as np                     #to work with array -mathematical operations
import matplotlib.pyplot as plt        #data visualization and graphical plotting
import seaborn as sns;sns.set()        #data visualisation and exploratory data analysis
import math                            #mathematical calculations
data=pd.read_csv(r'C:\Users\admin\Desktop\stpfinal.csv')    #importing data in csv format
data
```

	PH	TSS	COD	BOD	POTABILITY
0	7.20	13.0	31.0	7.0	1
1	7.40	13.0	26.0	9.0	1
2	7.46	14.0	43.0	8.0	1
3	7.46	12.0	46.0	10.0	0
4	7.44	13.0	44.0	9.0	1
...	...	...	...	...	...
1734	7.18	13.0	36.0	7.0	0
1735	7.05	10.0	26.0	16.0	1
1736	7.03	11.0	21.0	9.0	0
1737	7.07	11.0	26.0	9.0	0

1739 rows × 5 columns

1738	7.14	12.0	18.0	14.0	1

```
data.shape       #rows and columns
```

(1739, 5)

```
#data cleaning
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1739 entries, 0 to 1738
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   PH          1739 non-null   float64
 1   TSS         1739 non-null   float64
 2   COD         1739 non-null   float64
 3   BOD         1739 non-null   float64
 4   POTABILITY  1739 non-null   int64
dtypes: float64(4), int64(1)
memory usage: 68.1 KB
```

In [65]:

```
data.isnull().sum()     #checking null values
PH            0
TSS           0
COD           0
BOD           0
POTABILITY    0
dtype: int64
```

As data cleaning is done so we can move further.

To use the machine learning model the basic assumptions is that there should no multicollinearity between the regressor . So in our data type let X1,X2,X3, X4 be PH , TSS, COD, BOD respectively. These are the regressor in our data which affects the value of the response variable. So to check the multicollinearity between the regressors we use the Heat map as statistical tool.

10.4 HEATMAP

What is heat map?

A heatmap is basically the representation of two dimensional information (data) with the help of colours . It gives warm-to-cool colour spectrum to show which parts of a data has the most attention. We use Heatmap as a correlation matrix .In heatmap correlation matrix, both the axis has same variables and we check the correlation between them by using it .The dark colour represent the positive correlation and the medium light colour gives no correlation between the variable.

As it gives visual as well as numerical value to check the correlation . The values in the cell indicate the strength of the relationship, with positive values indicating a positive relationship and negative values indicating a negative relationship. In addition, correlation plots can be used to identify outliers and to detect linear and nonlinear relationships. The color-coding of the cells makes it easy to identify relationships between variables at a glance.

Check the multicollinearity between the regressors:

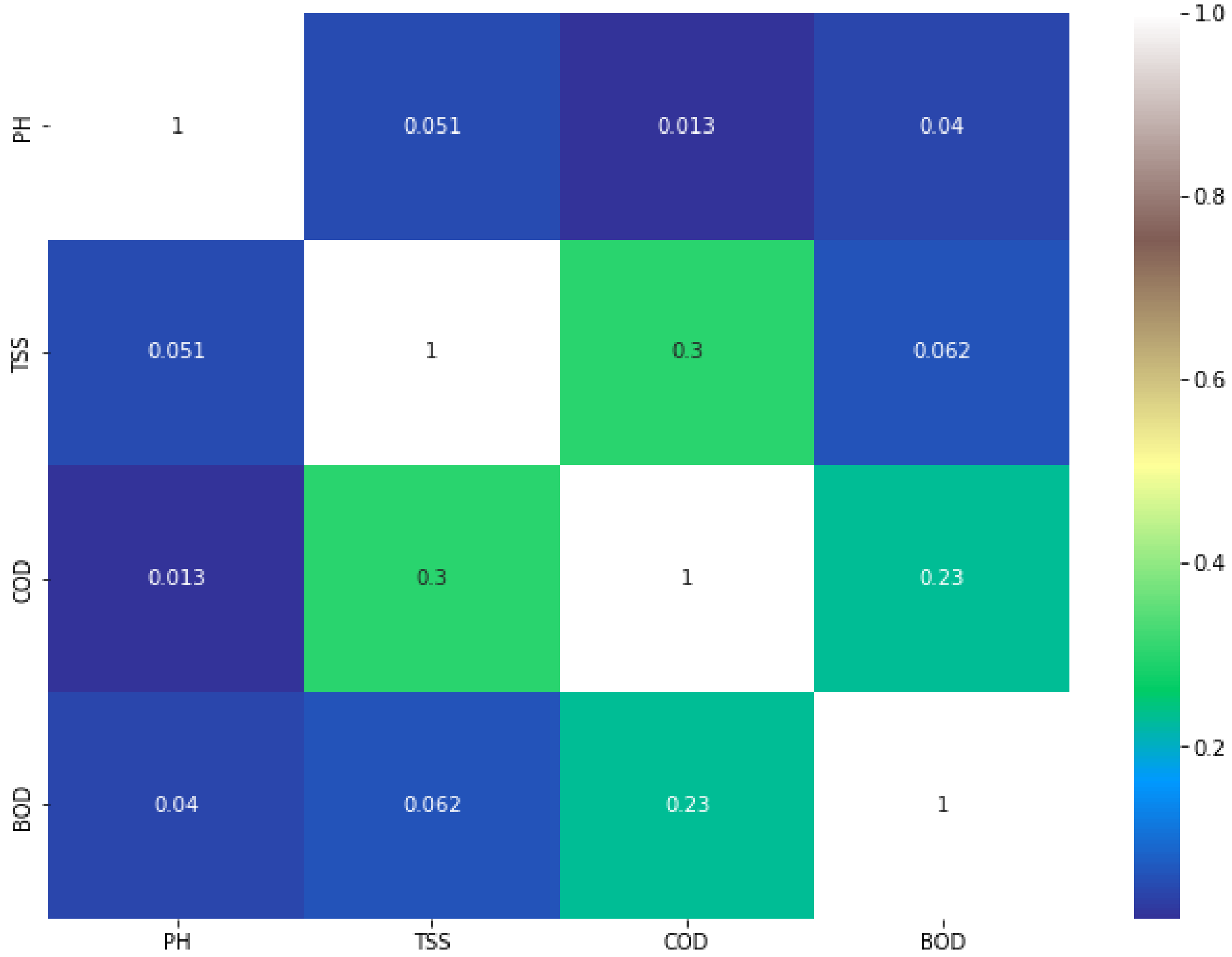
By using python we plot the heatmap for our data. The respective commands are as follows.

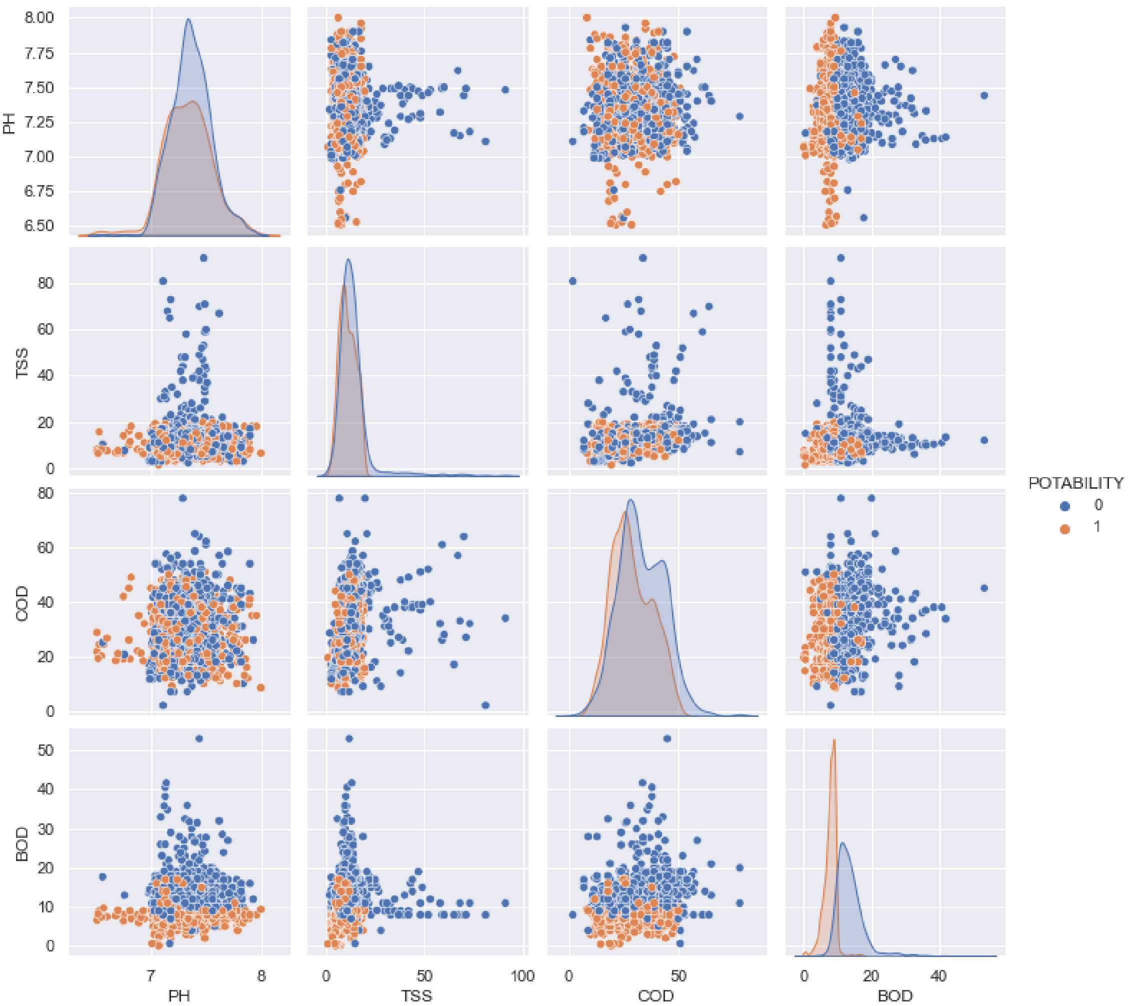
#**correlation using heatmap**

```
data=data.drop(['POTABILITY'],axis=1)

data

sns.heatmap(data.corr(),annot=True,cmap='terrain')

fig=plt.gcf()

fig.set_size_inches(11,8)

plt.show()
```

#**graphical representation of relationship between the parameters using pairplots**

```
sns.pairplot(data,hue='POTABILITY')

plt.show()
```


PH
TSS
COD
BOD
1
0.051
0.013
0.04
0.3
0.062
0.23
1.0
0.8
0.6
0.4
0.2


PH
TSS
COD
BOD
POTABILITY
0
1

Conclusion of heat map: As the correlation coefficient are neglible, we can conclude that the parameters are uncorrelated. The correlation coefficient for COD and TSS as well as for BOD and COD are considerably large due to numerical variations but there no such relation between them.

#Partitioning of data

>X=data.drop('POTABILITY',axis=1) #inputs variable

>X

	PH	TSS	COD	BOD
0	7.20	13.0	31.0	7.0
1	7.40	13.0	26.0	9.0
2	7.46	14.0	43.0	8.0
3	7.46	12.0	46.0	10.0
4	7.44	13.0	44.0	9.0
...	...	...	...	...
1734	7.18	13.0	36.0	7.0
1735	7.05	10.0	26.0	16.0
1736	7.03	11.0	21.0	9.0
1737	7.07	11.0	26.0	9.0
1738	7.14	12.0	18.0	14.0

11.Models development

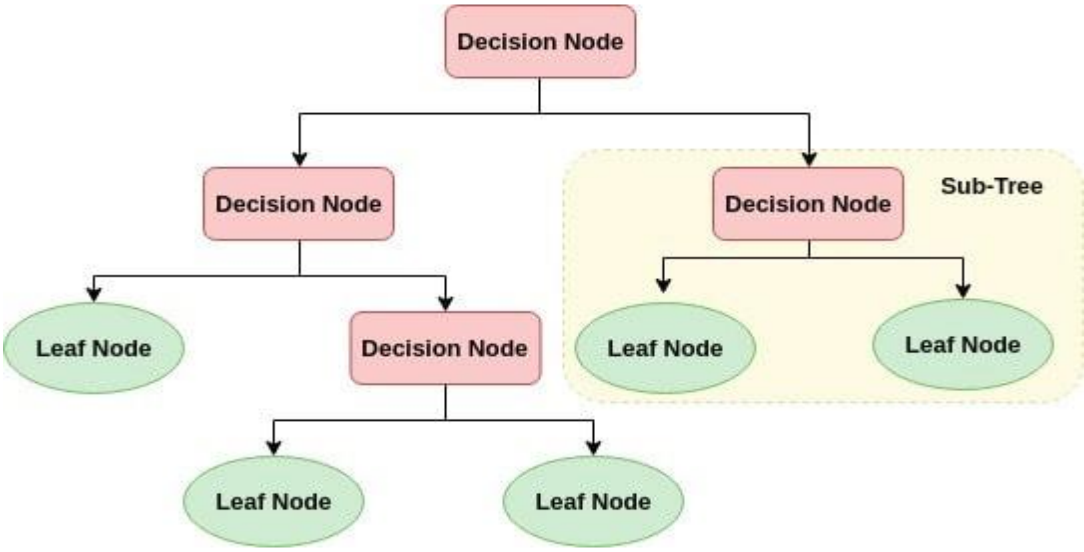
11.1 Decision Tree.

What is decision tree?

Decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource**.** It is Supervised Machine learning algorithm which uses set of rules to make decisions. It is one of the classification algorithms which uses rule-based approach.

For example: Planning the next vacation which depends on various factors such as time, no. of members, budget.

It can perform both classification and regression tasks so referred as CART algorithm (Classification and Regression Tree).



Intuition: Need of use of dataset features to create YES/NO type questions (In our case water potability) until we isolate all data points belonging to each class.

Model characteristics:

1)Fewer the splits more the accuracy.

2)Algorithm assigns only one class to each leaf node.

3)It picks best split to minimize loss function on basis of purity – "GINI Impurity"

$$\mathbf{G}=\sum_{k=1}^{c} \mathrm{P}(1-\mathrm{P})$$

4)Uses greedy approach

5)It can be linearized into decision rules

6)It should be paralled by a probability model as a choice model

7)Descriptive means for calculating conditional probabilities.

8)Categorical variable decision tree.

Advantages:

1. Simple to understand and to interpret.
2. It can handle both numerical as well as categorical data.

Disadvantages:

1)Unstable: Change sensitive

2)Relatively inaccurate

3)Bias in favour of attributes with more level

4) Calculations can get very complex

```
#Model fitting Decision tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score,confusion_matrix,precision_score
data=DecisionTreeClassifier(criterion= 'gini', min_samples_split=6, splitter= 'best') # quality support criterio
n-Gini
data
DecisionTreeClassifier(min_samples_split=6)
data.fit(X_train,Y_train)
DecisionTreeClassifier(min_samples_split=6)

#Prediction for test dataset
prediction=data.predict(X_test)
```

prediction

```
array([1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0,
       1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1,
       1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0,
       1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0,
       0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1,
       1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0,
       1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0,
       0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0,
       0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1,
       0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0,
       0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1,
       0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1,
       1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0,
       0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0], dtype=int64)
```

```
#Accuracy for decision tree

accuracy_score(prediction,Y_test)

print('accuracy_score:',accuracy_score(prediction,Y_test)*100,'%')
```

accuracy_score: 96.26436781609196 %

```
print("feature importances:\n{}".format(data.feature_importances_))
```

feature importances:

[0.01270699 0.07330349 0.02190262 0.8920869]

```
print("Accuracy on training set :{:.3f}".format(data.score(X_train,Y_train)*100),'%')

print("Accuracy on test set :{:.3f}".format(data.score(X_test,Y_test)*100),'%')
```

Accuracy on training set :98.994 %

Accuracy on test set :96.264 %

confusion_matrix(prediction,Y_test)   # describes performance of classification model on set of test data for which true values are known

```
array   ([184,   6],
       [  7, 151]], dtype=int64)
```

```
#Prediction on only one set of data

X_DT=data.predict([[7.5,25,39,15]])

X_DT
```

array([0], dtype=int64)

11.2 K -NEAREST NEIGHBOUR (KNN)

What is K-NN?

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. It was 1st used for classification task by Fix and Hodges in 1951.K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data. It maps an Input to an Output based on example of Input-Output pairs. i.e it stores all the available data and classifies a new data point based on the similarity.

Euclidean distance-

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$$

When do we use K-NN Algorithm?

1. When data is **labelled-** We already know the results of data for particular data set and based on this we try to classify future unknown data
2. When data is **noise free**-Noise is unwanted data items , features or records which don't help in explaining relationship between feature and target variable.

How does K-NN work?

The K-NN working can be explained on the basis of the below algorithm:

- **Step-1:** Select the number K of the neighbors.
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Model is ready.

Advantages of K-NN-

1. It is simple to implement
2. It is robust to noisy training data
3. It can be more effective if the training data is large

Disadvantages of K-NN-

1. Always needs to determine the value of k which may be complex sometimes.
2. The computation cost is high because of calculating the distance between the data points for all the training samples.

#KNN Model from sklearn.neighbors import KNeighborsClassifier

from sklearn.neighbors import KNeighborsClassifier

math.sqrt(len(Y_test)) #to find the value of neighbour in KNN model

18.65475810617763
knn=KNeighborsClassifier(metric='manhattan',n_neighbors=19) #k nearest neighbors algorithm

knn

KNeighborsClassifier(metric='manhattan', n_neighbors=19)
knn.fit(X_train,Y_train)

KNeighborsClassifier(metric='manhattan', n_neighbors=19)
#Prediction of test data set

prediction_knn=knn.predict(X_test)

prediction_knn

```
array([1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0,
       1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1,
       1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0,
       1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0,
       0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1,
       1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0,
       1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0,
       0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0,
       0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1,
       0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0,
       0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1,
       0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1,
       1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0,
       0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0], dtype=int64)
```

#Accuracy of KNN

accuracy_knn=accuracy_score(Y_test,prediction_knn)*100

print('accuracy_score:',accuracy_knn,'%')

accuracy_score: 93.67816091954023 %

confusion_matrix(prediction,Y_test)

```
array([[184,   6],
       [  7, 151]], dtype=int64)
```

11.3 Logistic Regression Model

What is logistic model?

It is a statistical method which is used to Predict a "binary output such as Yes or No (in our case 1 or o). Logistic regression model predicts dependent variable of data using regressors which are independent.

It is basically a supervised classification algorithm use in classification problems. As in linear regression, it is assume that the data follows linear function similarly logistic model builds a regression model to predict the probability that given data entry belongs to Category numbered as **"1" OR "0"**

Assumptions:

1. Absence of Multicollinearity- one of the most important assumptions.
2. The dependent variable must be dichotomous.

Why this model?

As in our data, response variable is in the form of binary type and also there is no collinearity between the regressors (Using heatmap we can observed) , hence we have use this model for testing quality of water i.e whether it is potable or not.

Model of Logistic regression:

1. $Y= E(Y|x) + \varepsilon$
2. $Y = \Pi(x) + \varepsilon$

Where, ε is Bernoulli random variable with

a. $E(\varepsilon)=0$

b. $var(\varepsilon) = \pi(x)(1-\pi(x))$

$$\pi(x) = \frac{e^{\beta_0+\beta_1X_1+\beta_2X_2+\beta_3X_3+\beta_4X_4}}{1+e^{\beta_0+\beta_1X_1+\beta_2X_2+\beta_3X_3+\beta_4X_4}}$$

$$Y=\Pi(x)+\varepsilon$$

$$Y=\frac{e^{\beta_0+\beta_1X_1+\beta_2X_2+\beta_3X_3+\beta_4X_4}}{1+e^{\beta_0+\beta_1X_1+\beta_2X_2+\beta_3X_3+\beta_4X_4}} + \varepsilon$$

To get the regressor coefficient

```
Call:
glm(formula = POTABILITY ~ PH + TSS + COD + BOD)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.12243  -0.34230   0.01814   0.33814   2.45516

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.1544201  0.3360372   6.411 1.86e-10 ***
PH          -0.1082749  0.0456846  -2.370   0.0179 *
TSS         -0.0093846  0.0013258  -7.078 2.11e-12 ***
COD         -0.0013837  0.0009614  -1.439   0.1503
BOD         -0.0684741  0.0020448 -33.487  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 0.1411137)

    Null deviance: 431.21  on 1738  degrees of freedom
Residual deviance: 244.69  on 1734  degrees of freedom
AIC: 1536.8

Number of Fisher Scoring iterations: 2

>
```

$$Y=\frac{e^{2.1544-0.10827X1-0.0092846X2-0.0013837X3-0.06847x4}}{1+e^{2.1544-0.10827X1-0.0092846X2-0.0013837X3-0.06847x4}}+\varepsilon$$

Where , β_1 ,β_2, β_3, β_4 are regression coefficients and variables are

POTABILITY	Y
PH	X_1
TSS	X_2
COD	X_3
BOD	X_4

Logistic model considers probability using which we are going to allocate new observation to specify class.For this purpose the threshold probability is decided and by default it is consider as P=0.5

#**Logistic Regression Model**

```
#Model fitting

from sklearn.linear_model import LogisticRegression

model = LogisticRegression()

model.fit(X_train , Y_train)

LogisticRegression()
#Prediction for test data set

predictlog=data.predict(X_test)

predictlog

array([1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0,
       1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1,
       1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0,
       1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0,
       0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1,
       1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0,
       1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0,
       0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0,
       0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1,
       0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0,
       0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1,
       0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1,
       1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0,
       0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0], dtype=int64)
#Accuracy of logistic

test_acc = accuracy_score(Y_test,prediction)

test_acc

print("The accuracy for Test Set is {}".format(test_acc*100),'%')
```

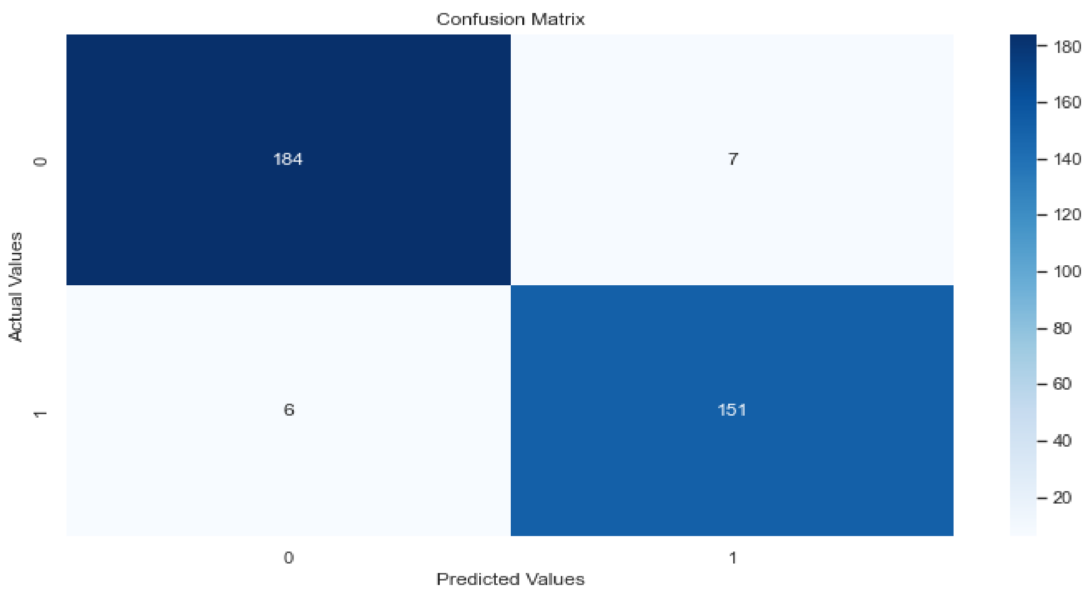
The accuracy for Test Set is 96.26436781609196 %

```
#one sample prediction

X_DT=data.predict([[7.5,19,15,85]])

X_DT

array([0], dtype=int64)


#confusion matrix

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
print(classification_report(Y_test,prediction))


cm=confusion_matrix(Y_test,prediction)

cm
```

```
          precision    recall  f1-score   support

           0       0.97      0.96      0.97       191
           1       0.96      0.96      0.96       157

    accuracy                           0.96       348
   macro avg       0.96      0.96      0.96       348
weighted avg       0.96      0.96      0.96       348

array([[184,   7],
       [  6, 151]], dtype=int64
```

```
#confusion matrix
plt.figure(figsize=(12,6))
plt.title("Confusion Matrix")
sns.heatmap(cm, annot=True,fmt='d', cmap='Blues')
plt.ylabel("Actual Values")
plt.xlabel("Predicted Values")
plt.savefig('confusion_matrix.png')
```



```
#RMSE of logistic regression model
import math
mse=np.square(np.subtract(Y_test,predictlog)).mean()
rmse=math.sqrt(mse)
print(rmse)
0.1932778358712671
```

```
#knn
import math
mse=np.square(np.subtract(Y_test,prediction_knn)).mean()
rmse=math.sqrt(mse)
print(rmse)
0.25143267648537193
```

#decision tree

```
import math
mse=np.square(np.subtract(Y_test,prediction)).mean()
rmse=math.sqrt(mse)
print(rmse)
0.1932778358712671
```

Model	Accuracy	RMSE
KNN	**93.67816091954023 %**	0.251432
Decision tree	**96.26436781609196 %**	0.193277
Logistic Model	**96.26436781609196 %**	0.193277

As RMSE of logistic model is smaller than other models and also we can say, which regressor affect the potability of water using this model. We conclude that, logistic model is best.

To check the significance of regressor

To test:

H_0 =the regressors PH is not significant vs

H_1= the regressors PH is significant

```
> a=glm(formula=POTABILITY~PH,family="binomial")
> a

Call:  glm(formula = POTABILITY ~ PH, family = "binomial")

Coefficients:
(Intercept)           PH
     5.4758      -0.7701

Degrees of Freedom: 1738 Total (i.e. Null);  1737 Residual
Null Deviance:      2397
Residual Deviance: 2387         AIC: 2391
```

From the above table we interpret that, the regressors pH is significant.

As null deviance -residual deviance =2397-2387

$=10> \chi^2_{1,0.05}$

Conclusion:

As $10> \chi^2_{1,0.05}$ we reject the null hypothesis, our regressors PH is significant.

To test:

H_0 =the regressors TSS is not significant vs

H_1= the regressors TSS is significant

```
> b=glm(formula=POTABILITY~TSS,family="binomial")
> b

Call:  glm(formula = POTABILITY ~ TSS, family = "binomial")

Coefficients:
(Intercept)          TSS
    0.80549     -0.08239

Degrees of Freedom: 1738 Total (i.e. Null);  1737 Residual
Null Deviance:      2397
Residual Deviance: 2320         AIC: 2324
```

From the above table we interpret that, the regressors TSS is significant.

As null deviance -residual deviance =2397-2320

$=77> \chi^2_{1,0.05}$

Conclusion:

As $70> \chi^2_{1,0.05}$ we reject the null hypothesis, our regressors TSS is significant.

To test:

H_0 =the regressors BOD is not significant vs

H_1= the regressors BOD is significant

```
> c=glm(formula=POTABILITY~BOD,family="binomial")
Warning message:
glm.fit: fitted probabilities numerically 0 or 1 occurred
> c

Call:  glm(formula = POTABILITY ~ BOD, family = "binomial")

Coefficients:
(Intercept)          BOD
     13.278       -1.373

Degrees of Freedom: 1738 Total (i.e. Null);  1737 Residual
Null Deviance:      2397
Residual Deviance: 889   AIC: 893
```

From the above table we interpret that, the regressors BOD is significant.

As null deviance -residual deviance =2397-889

$=1508 > \chi^2_{1,0.05}$

Conclusion:

As $1508 > \chi^2_{1,0.05}$ we reject the null hypothesis, our regressors BOD is significant.

To test:

H_0 =the regressors COD is not significant vs

H_1= the regressors COD is significant

```
> d=glm(formula=POTABILITY~COD,family="binomial")
> d

Call:  glm(formula = POTABILITY ~ COD, family = "binomial")

Coefficients:
(Intercept)          COD
    1.19762     -0.04469

Degrees of Freedom: 1738 Total (i.e. Null);  1737 Residual
Null Deviance:      2397
Residual Deviance: 2315           AIC: 2319
```

From the above table we interpret that, the regressors BOD is significant.

As null deviance -residual deviance =2397-2315

$=82 > \chi^2_{1,0.05}$

Conclusion:

As $82 > \chi^2_{1,0.05}$ we reject the null hypothesis, our regressors COD is significant.

12.CONCLUSIONS:

The foremost part in our project is understanding the Time series and ARIMA model. This study helps in understanding the variations in the sewage inflow of New Naidu STP. According to ARIMA (1, 0, 1) model forecasted the values for next 30 days, which shows there may be increase in the waste water inflow up to 126 MLD for upcoming days and by using Holt-winter model the waste water inflow can be 189 MLD in the upcoming days . The forecasted values of the inflow rate help to monitor the sewage load and for future planning of STPs.

By using testing of hypothesis, we conclude that STPs are efficiently working which means the treated water is potable.

Our project also includes the understanding of the Machine Learning and its basic types. The classification models were used to analyse the water quality. The supervised classification models namely decision tree, KNN model and Logistic Model were fitted to our sample data of 1739 sample points. The water quality analysis is based on the parameters present in it, which are pH, TSS, BOD and COD, there standard ranges were provided by WHO and lab reports. Of the three models that were fitted to this data, Logistic model proved to be the best fit with accuracy of 96.26 %. With this accuracy, it concludes that our data is overfitted.

13.Scope and limitations :

Scope:

1. If we compare the relationship between the waste water inflow and population rate we can correctly interpret our result about forecasting.
2. If we have given all parameter values for given water sample then we can use this model for drinkable or non-drinkable water.
3. Using machine learning models we can easily interpret the result for future data also.

Limitations:

1. To apply time series data should be large.
2. The models can be apply if and only if the data can be classify into two groups.
3. We can't use some classification model e.g naive bayes if our data is not normally distributed.

14.References:

BOOKS:

1. ANALYSIS OF TIME SERIES AN INTRODUCTION(Fifth edition)

   by Chris Chatfield

2. FUNDAMENTALS OF APPLIED STATISTICS by S.C.GUPTA V.K KAPOOR

3. Montgomery , D.C.and Johnson L.A.(1976):Forecasting and Time Series Analysis , McGraw Hill

4. Data Mining Concepts and Techniques (Third Edition) by Jiawei Han , Micheline Kamber, Jian Pei

5.Fundamentals of Python Programming by Richard L.Halterman.

LINKS:

1.https://www.geeksforgeeks.org/

2.https://www.analyticsvidhya.com/blog/2021/06/hypothesis-testing-parametric-and-non-parametric-tests-in-statistics/

3.https://www.javatpoint.com/machine-learning

4.https://www.wikipedia.org/

5. https://github.com/python

6. https://www.analyticsvidhya.com/blog/2020/11/popular-classification-models-for-machine-learning/

7. https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html