INDIAN WATER QUALITY TRACKING AND PREDICTION USING WSN AND MACHINE LEARNING

by Anbarivan N.I

Submission date: 02-Apr-2019 12:44PM (UTC+0530)
Submission ID: 1104364254
File name: LITY_TRACKING_AND_PREDICTION_USING_WSN_AND_MACHINE_LEARNING.docx (2.74M)
Word count: 5352
Character count: 29538

ABSTRACT

Water contamination is one of the greatest issues confronting India at the present time. As might be clear, untreated sewage is the greatest wellspring of such type of contamination in India. There are different wellsprings of contamination, for example, overflow from the rural division just as unregulated units that have a place with the little scale industry. The circumstance is serious to the point that maybe there is no water body in India that isn't contaminated to some degree or the other. Truth be told, it is said that practically 80% of the waterbodies in India are exceptionally dirtied. This is particularly material of ones that some structure or the other of human residence in their quick region. Ganga and Yamuna are the most dirtied waterways in India.

we proposed a water pollution tracking and prediction system in order to reduce the growing pollution and tack down the pollution hotspots over a wider area. Various types of sensors are used to collect data various location; the data are reduced on various aspects to improvise the the system efficiency. To ensure the system redundancy, weight-based node failure detection and tracking are implemented. Collected Data are used to train machine learning algorithm to predict the future pollution rates. This model is optimized using gradient descent on varying datasets.

CHAPTER 1

BACK GROUND

iii

LIST OF FIGURES:

LIST OF TABLES

LIST OF ACRONYMS xii

WQI- water quality index

Chapter 1

Introduction

1.1 BACKGROUND -

Water contamination is one of the greatest issues confronting India at the present time. As might be clear, untreated sewage is the greatest wellspring of such type of contamination in India. There are different wellsprings of contamination, for example, overflow from the rural division just as unregulated units that have a place with the little scale industry. The circumstance is serious to the point that maybe there is no water body in India that isn't

contaminated to some degree or the other. Truth be told, it is said that practically 80% of the waterbodies in India are exceptionally dirtied. This is particularly material of ones that some structure or the other of human residence in their quick region. Ganga and Yamuna are the most dirtied waterways in India.

The single main motivation for water contamination in India is urbanization at an uncontrolled rate. The rate of urbanization has just gone up at a quick pace in the most recent decade or somewhere in the vicinity, however and still, at the end of the day it has left a permanent imprint on India's sea-going assets. This has prompted a few ecological issues in the long haul like scarcity in water supply, age and accumulation of wastewater to give some examples. The treatment and transfer of wastewater has likewise been a noteworthy issue in such manner. The zones close waterways have seen a lot of towns and urban areas come up and this has likewise added to the developing force of issues.

Uncontrolled urbanization in these territories has additionally prompted age of sewage water. In the urban regions water is utilized for both modern and local purposes from waterbodies, for example, waterways, lakes, streams, wells, and lakes. Most exceedingly bad still, 80% of the water that we use for our household reasons for existing is passed out as wastewater. In a large portion of the cases, this water isn't dealt with appropriately and all things considered it prompts colossal contamination of surface-level freshwater. This dirtied water additionally leaks through the surface and toxic substances groundwater. It is evaluated that urban communities with populaces of more than one lakh individuals produce around 16,662 million liters of wastewater in multi day. For some odd reason, 70% of the general population in these urban communities approach sewerage offices. Urban areas and towns situated on the banks of Ganga create around 33% of wastewater produced in the nation.

The Central Pollution Control Board (CPCB) in relationship with State Pollution Control Boards (SPCBs)/Pollution Control Committees(PPCs) is observing the nature of water bodies at 2500 areas the nation over under National Water Quality Monitoring Program (NWQMP) which demonstrate that natural contamination is the overwhelming reason for water contamination. In view of the size of natural contamination, CPCB in 2008 recognized 150 dirtied stream extends which expanded to 302 of every 2015. The waterways extends are dirtied fundamentally because of release of untreated/in part treated sewage and release of modern wastewater. CPCB surveyed the all out volume of civil wastewater age in the nation at around 61,948 MLD as against the introduced sewage treatment limit of 23,277 MLD leaving a wide hole of more than 38,671 MLD.Similar perceptions were made by WHO in its reports on water contamination.

2. LITERATURE SURVEY:

This work represents the implementation of the two well-known power efficient data gathering and aggregation protocols: PEDAP and PEDAP-PA. Simulations are used to show that both the algorithms perform near optimal. The simulations show that keeping all the nodes to work together is important. PEDAP-PA performs best among

others but where the lifetime of the last node is important, PEDAP is a good alternative. [1]

This paper introduces us a MAC and cross-layer routing approach to QoS assessment in a WSN. The investigation is primarily based on two methods: the best-effort and latency constraint. These approaches can be used for rapid assessment of expected quality of service in the networks as well as finding of time division multiplexing schedules for utilization in network. The final simulation that is done shows that reliability in substantial gain is achieved. [2]

In this paper, data-centric routing is statistically assessed and its performance is compared with traditional end to end routing schemes. The impact of source to destination placement and network density on the energy costs is carefully examined in this paper. The significance of data-centric routing that offers high performance across variety of range of operational scenarios. [3]

This paper compares the various algorithms that make predictions in time series by data mined from WSN. A simulation is performed that shows the nature of the data and their entropy deeply influences the performance of the selected algorithm. After the implementation and observation of results, it is concluded that gradually changing data is best for ARMA, and for data with sharp changes, MA is the most suitable. [4]

In this paper, a novel technique, DBP, is applied to over 13 million data points from four real world applications. The assessment shows that the technique vanquishes 99 percent of the application data and its performance is often better than the other common approaches. [5]

Huge reductions in communication is hence automatically allowed in this technique. Practical use of DBP includes improving system lifetime from every aspect. The paper is very well explained and the important terms are beautifully highlighted in detail.[6]

Grouping the sensors into clusters is very well explained in this paper. The technique used here is heterogeneous clustering, which is very energy-efficient. This is done by selecting the cluster head from the cluster with respect to the residual energy of the nodes, transmission range and number of transmissions. The connectivity, considered as a measure of QoS, is ensured by Route identification technique. [7]

This paper analyses the wireless sensor networks that are very important in distributed-based systems. The paper models and analyses the performance value of data aggregation in the network in question. The results show that what ever the sources of cluster, either clustered or randomly, energy gains can happen with data aggregation. The energy gain is maximum when the number of sources is large and are located relatively close to each other. [8]

This paper analyses the three main phases of fault tolerance and fault detection models at four level of abstractions, namely, hardware, system software, middleware and applications. Four scopes, namely, components of individual node, individual node, network and the distributed system also encloses the fault model that is being analysed. A final conclusion is made that a brief survey of the future directions can widely affect the tolerance research in wireless sensor networks. [9]

This paper gives an analysis that defines the fault tolerance and the various terms related to it. Various aspects of data constraints such as redundancy and touched-upon fault tolerance has been explored and explained that are used in Wireless Sensor networks. Some of the techniques that has been covered in this paper are redundancy in hardware, NMR and N-version programming software. [10]

This paper describes the optimization problems that deal with wireless networks, its planning and design. Deployment and operation that give rise to formulation for multi-objective optimisation formulations. A list of constraints in the paper is also listed beautifully to give a reference of various constraints that has been taken into account while calculating the optimisation problems. This paper aims at opening up new research avenues in the path of wireless network optimisation.
The concern of this paper is data, topology and hierarchy. It examines the different types of wireless routing protocols, research done in this field as well as classifies the various methods that are said. It also helps one to know whether optimization of traffic and energy is done. Other benefits include knowing the node mobility, where the node is deployed etc [11].

In this paper, wireless sensor networks are classified based on their network type that is either proactive or reactive. Also, a protocol called TEEN (Threshold sensitive Energy Efficient sensor Network protocol) is introduced and used primarily for energy efficiency and applications with time constraints and performance is measured. It outruns the conventional sensor network protocols [12].

The main concern of this paper is providing security for routing sensors. It is more of an analysis of various types of sensor attacks such as sinkholes, etc. The scrutiny of all

major routing protocols is studied. Apart from investigation, model design and counteractions are also discussed. They have proved that current technologies are insecure and propose for the development of a system that is much better than the standard cryptographic techniques [13].

This paper proposes a cluster model called the LEACH (Low energy adaptive clustering hierarchy) protocol that works in a distributive environment. This is vital for reducing the energy usage by distributing the energy evenly to all the sensors. This provides large scale usability as well as toughness. There is also evidence of decrease in communication energy and lifespan [14].

This paper primarily focusses on energy efficiency of untended sensors or actuators. An algorithm, termed as GEAR also known as geographic and energy aware routing is introduced. Routing is decided based on the neighbour's selection. This also avoids the typical flooding scenario. Simulation is done to measure the performance. In terms of packet, it outperforms GPSR by 70 to 80% and also packets delivered are 25 to 30% more than a GPSR. Current work includes developing a working prototype of GEAR protocol [15].

The main objective of this paper is to create a hybrid protocol called APTEEN which primarily focusses on retrieval of information, reacting to periodic as well as time critical events. It enables a user to get the historical data in the form of queries and analyse it. This model works better in terms of energy efficiency as well as the lifetime of senor networks whose nodes are evenly distributed and could be extended to unevenly distributed nodes in the future [16].

This paper emphases on need for energy efficient wireless sensors because there is battery oriented and it's hard to track and replace them. Routing protocol called the Base-Station Controlled Dynamic Clustering Protocol (BCDCP) is introduced that aims in even distribution of energy among the sensor nodes and compares it to LEACH-C and PEGASIS.BCDCP provide a means of balancing the clusters as well as outperforming the computational tasks that demands more energy and provides a wide range of applicability for sensors [17].
This paper is a study of energy efficient routing for wireless sensors. It investigates the energy histogram and draws various methods to enhance routing. Packet streams are joined together in the first approach, next arguments on various energy efficient routing of sensors is done. Finally, conclusions are drawn that energy efficient routing is impossible in practical life. This paper stresses more on the need of practical implementations of energy efficient routing models [18].

3.METHODOLOGY:

System architecture: -

The key research questions in methods development are driven by the various monitoring needs tied to compliance with the national water quality standards, real time public information, and support for atmospheric and health research studies. The main aim is to design, develop and implement a mechanism to identify various contamination issues and assess the level of pollution in relation to the water quality standards as defined by Indian government. Analysing effluent data values with the predefined thresholds as stated by CPCB and generating alert information depending on the degree of pollution.

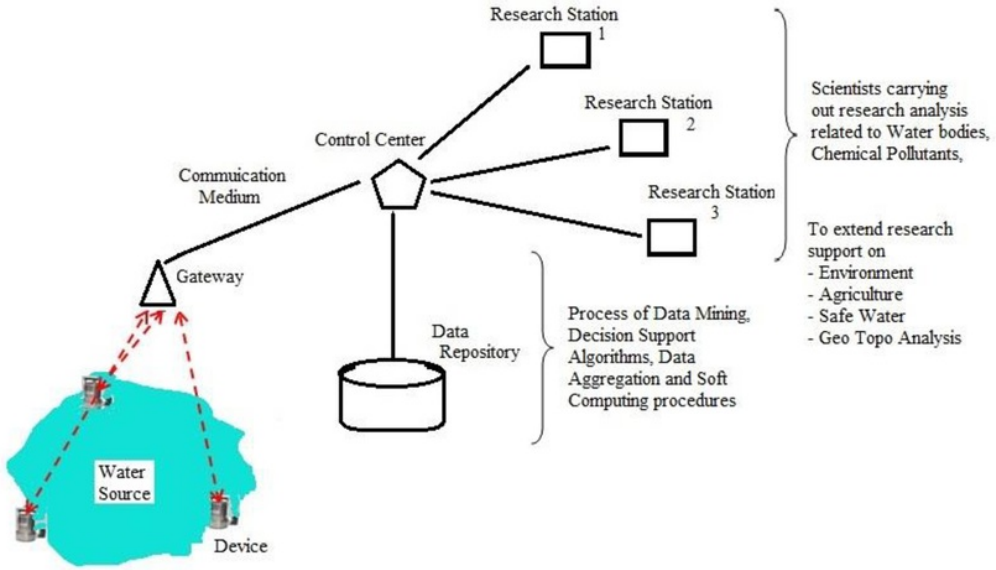


Figure 1: water quality system architecture

Functional modules

- Sensors data Aggregation
- Gateway functionality
- Server analysis (ML, Decision making)
- Network failure optimization

We acquired the dataset with various columns of sensor data from various places in India. we have the average readings of ambient air quality with respect to air quality parameters are collected from PH sensor, NITRATENAN N+ NITRITENANN Sensor, FECAL COLIFORM Sensor, B.O.D. Sensor, D.O. Sensor, TEMPERATURE Sensor. Data acquired from the source has noisier data since few of the data from the stations have been shifted or closed the period were marked as NAN or not available.so we have to pre-process the data in order to remove the outliers.

Sample dataset: -

	A	B	C	D	E	F	G	H	I	J	K	L
1	STATION C	LOCATIONS	STATE	Temp	D.O. (mg/l)	PH	CONDUCTIVITY (µmhos/cm)	B.O.D. (mg/l)	NITRATENAN N+ NITRITENANN	FECAL COLIFORM (MPN/100ml)	TOTAL COLIFORM (MPN/100m	year
2	1393	DAMANGANGA AT D/S OF MADHUBAN, DA	DAMAN & DIU	30.6	6.7	7.5	203	NAN	0.1	11	27	2014
3	1399	ZUARI AT D/S OF PT. WHERE KUMBARJRIA C	GOA	29.8	5.7	7.2	189	2	0.2	4953	8391	2014
4	1475	ZUARI AT PANCHAWADI	GOA	29.5	6.3	6.9	179	1.7	0.1	3243	5330	2014
5	3181	RIVER ZUARI AT BORIM BRIDGE	GOA	29.7	5.8	6.9	64	3.8	0.5	5382	8443	2014
6	3182	RIVER ZUARI AT MARCAIM JETTY	GOA	29.5	5.8	7.3	83	1.9	0.4	3428	5500	2014
7	1400	MANDOVI AT NEGHBOURHOOD OF PANAJI,	GOA	30	5.5	7.4	81	1.5	0.1	2853	4049	2014
8	1476	MANDOVI AT TONCA, MARCELA, GOA	GOA	29.2	6.1	6.7	308	1.4	0.3	3355	5672	2014
9	3185	RIVER MANDOVI AT AMONA BRIDGE	GOA	29.6	6.4	6.7	414	1	0.2	6073	9423	2014
10	3186	RIVER MANDOVI AT IFFI JETTY	GOA	30	6.4	7.6	305	2.2	0.1	3478	4990	2014
11	3187	RIVER MANDOVI NEAR HOTEL MARRIOT	GOA	30.1	6.3	7.6	77	2.3	0.1	2606	4301	2014
12	1543	RIVER KALNA AT CHANDELNAN PERNEM, G	GOA	27.8	7.1	7.1	176	1.2	0.1	4573	7817	2014
13	1548	RIVER ASSONORA AT ASSONORA, GOA	GOA	27.9	6.7	6.4	93	1.4	0.1	2147	3433	2014
14	2276	RIVER BICHOLIM VARAZAN NAGAR , BICHOI	GOA	29.3	7.4	6.8	121	1.7	0.4	11633	18125	2014
15	2275	RIVER CHAPORA NEAR ALORNA FORT ,PERN	GOA	29.2	6.9	7	620	1.1	0.1	3500	6300	2014
16	3189	RIVER CHAPORA AT SIOLIM	GOA	30	6	7.5	72	1.6	0.2	4995	9517	2014
17	1546	RIVER KHANDEPAR AT OPA NAN PONDA, G	GOA	29	7.3	7	247	1.5	0.2	1095	2453	2014
18	2270	RIVER KHANDEPAR AT CODLI NEAR BRIDGE	GOA	29.1	7.3	7	188	1	0.1	1286	3048	2014
19	2272	RIVER KUSHAWATI NEAR BUND AT KEVONA	GOA	28.7	7	6.9	224	1.2	0.3	3896	6742	2014
20	1545	RIVER MADAI AT DABOS NAN VALPOI, GOA	GOA	28.7	7.3	6.7	144	1.5	0.1	1940	3052	2014
21	2274	RIVER MAPUSA ON CULVERT ON HIGHWAY (	GOA	29.5	5.3	6.8	319	1.8	0.3	6458	10250	2014
22	2271	RIVER SAL PAZORKHONI,CUNCOLIM(NEAR (	GOA	29	6.3	6.4	79	1.6	1.4	7592	12842	2014
23	2273	RIVER SAL NEAR HOTEL LEELA MOBOR,CAVE	GOA	29.4	5.4	7.6	39	1.4	0.1	3176	6367	2014
24	3183	RIVER SAL AT KHAREBAND, MARGAO	GOA	28.3	2.2	6.5	322	4.7	1.2	11210	14920	2014
25	3184	RIVER SAL AT ORLIM BRIDGE, ORLIM	GOA	30.1	5.2	7.1	192	2.6	0.3	5073	8925	2014
26	3190	RIVER SINQUERIM (CANDOLIM SIDE NEAR B	GOA	30.3	5.6	7.5	282	1.8	0.1	3205	5082	2014
27	3191	RIVER SINQUERIM NEAR NERUL TEMPLE	GOA	30.5	5.5	7.4	275	1.5	0.1	4698	8625	2014
28	1547	RIVER TALPONA AT CANACONA, GOA	GOA	29.1	7.3	6.7	55	1.4	0.1	2638	4003	2014
29	3188	RIVER TIRACOL AT TIRACOL	GOA	30.1	6.5	7.5	415	2	0.1	864	1538	2014
30	1544	RIVER VALVANT AT SANKLI NAN BICHOLIM,	GOA	29.2	7.2	6.3	100	1.5	0.1	7942	13575	2014

Table 1: Water Quality Data

In this dataset we have the pollutant concentration levels occurring on each place. These parameters should be reduced show that our model learning and predicting rate will be better. So, we have calculated the water quality index (WQI) for all the available data points.to calculate the WQI we have to find the individual indexes of each pollutant. Each index of pollutant represents the level of damage caused by the pollutant. Each indexes which varies with its own scale.

4.WATER DATA ANALYSIS

From the obtained dataset, various from pollutant concentrations are obtained from PH sensor, NITRATENAN N+ NITRITENANN Sensor, FECAL COLIFORM Sensor, B.O.D. Sensor, D.O. Sensor, TEMPERATURE sensor with respect to the timestamp.

Pollutants list:

D.O. (mg/l)

B.O.D. (mg/l)

NITRATENAN N+ NITRITENANN (mg/l)

TOTAL COLIFORM (MPN/100ml)

	STATION CODE	LOCATIONS	STATE	Temp	D.O. (mg/l)	PH	CONDUCTIVITY (µmhos/cm)	B.O.D. (mg/l)	NITRATENAN N+ NITRITENANN (mg/l)	FECAL COLIFORM (MPN/100ml)	TOTAL COLIFORM (MPN/100ml)Mean	year
0	1393	DAMANGANGA AT D/S OF MADHUBAN, DAMAN	DAMAN & DIU	30.6	6.7	7.5	203.0	NaN	0.1	11	27.0	2014
1	1399	ZUARI AT D/S OF PT. WHERE KUMBARJRIA CANAL JOI...	GOA	29.8	5.7	7.2	189.0	2.0	0.2	4953	8391.0	2014
2	1475	ZUARI AT PANCHAWADI	GOA	29.5	6.3	6.9	179.0	1.7	0.1	3243	5330.0	2014
3	3181	RIVER ZUARI AT BORIM BRIDGE	GOA	29.7	5.8	6.9	64.0	3.8	0.5	5382	8443.0	2014
4	3182	RIVER ZUARI AT MARCAIM JETTY	GOA	29.5	5.8	7.3	83.0	1.9	0.4	3428	5500.0	2014

Table 2: Water quality Dataframe

Histogram plot of pollutant concentration:

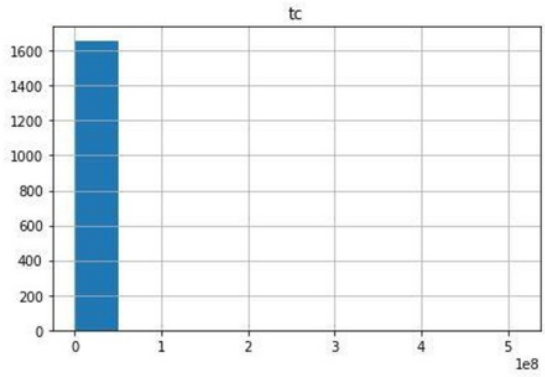


Figure 2: TOTAL COLIFORM (MPN/100ml)

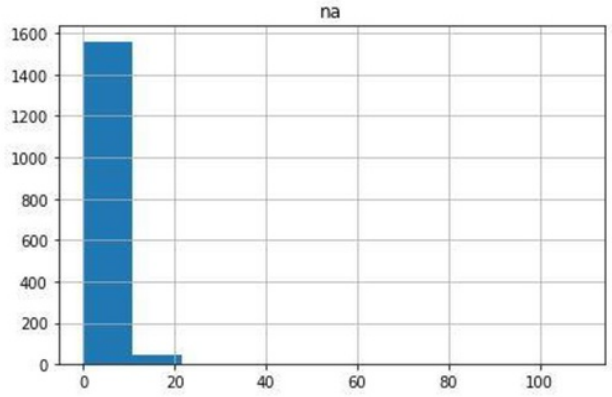


Figure 3: NITRATENAN N+ NITRITENANN (mg/l)

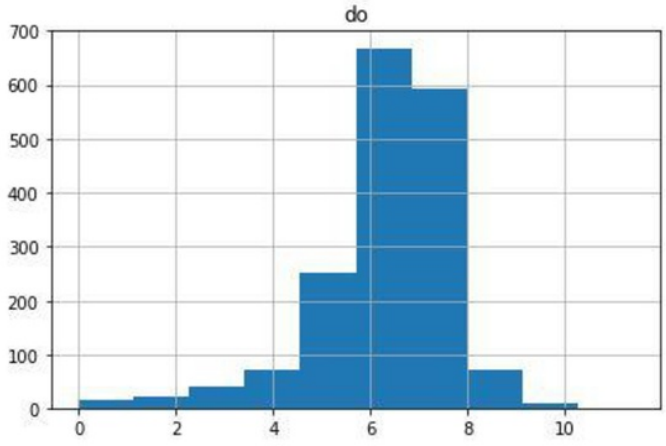


Figure 4: D.O. (mg/l)

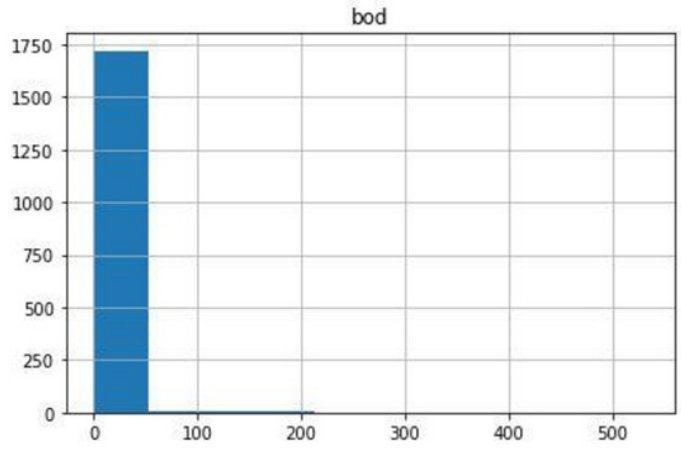


Figure 5: B.O.D. (mg/l)

Other features:

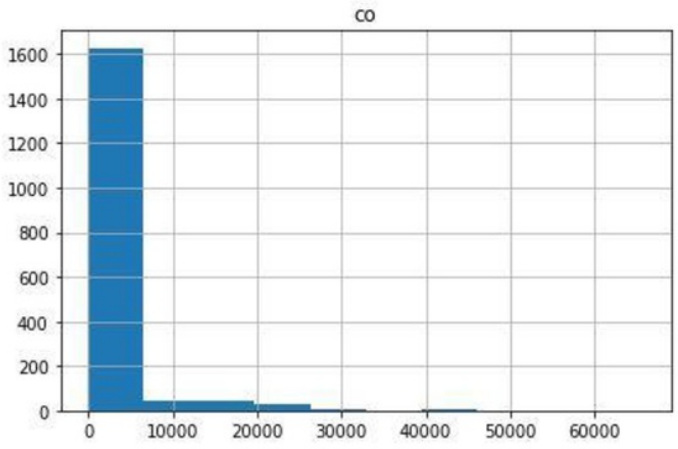


Figure 6:CONDUCTIVITY (*μ*mhos/cm)

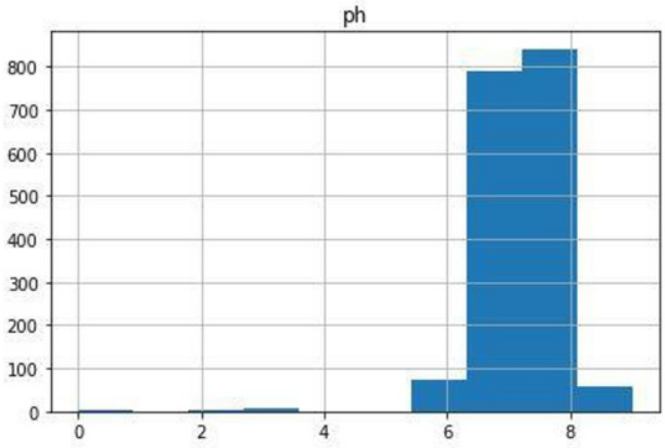


Figure 7:PH value

Seasonality and trend of various pollutants:

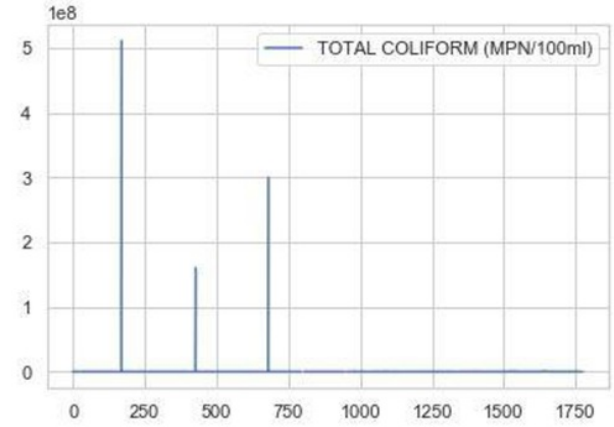


Figure 8: seasonality of coliform

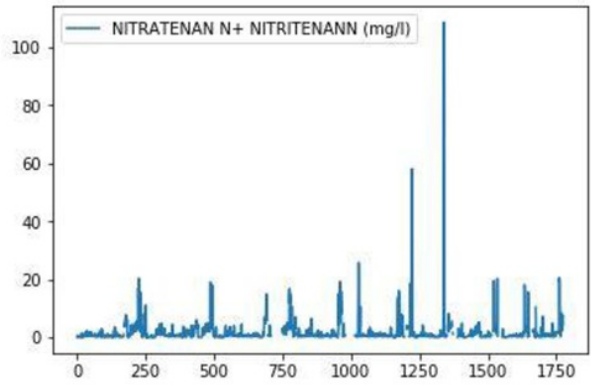


Figure 9: seasonality of nitratenan

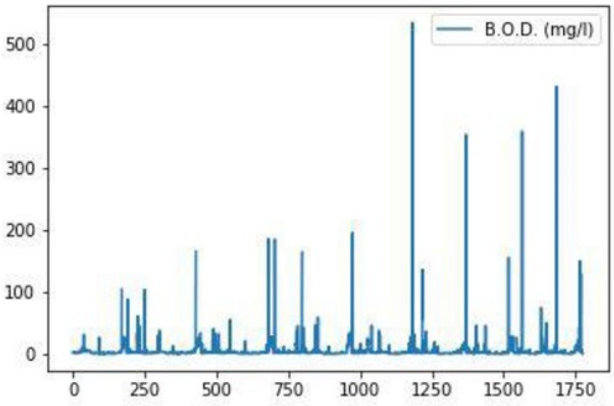


Figure 10: seasonality of B.O.D

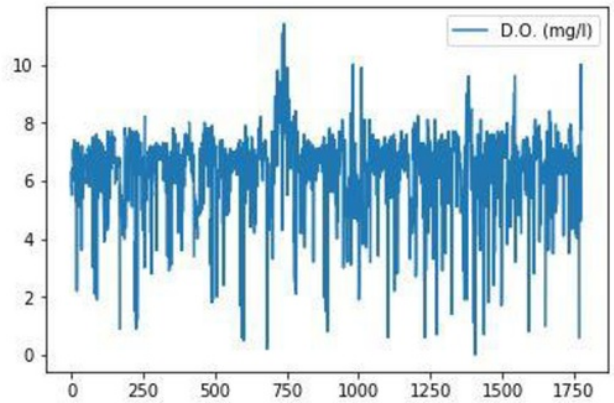


Figure 11: seasonality of D.O

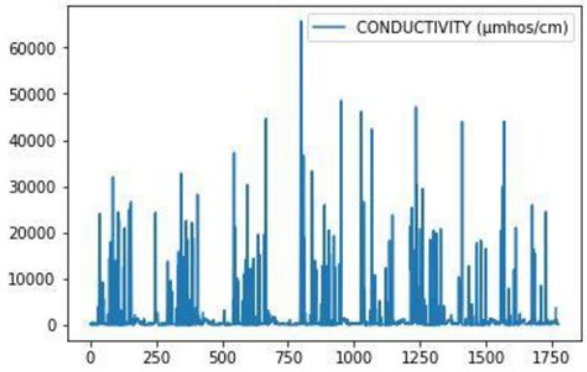


Figure 12: seasonality of conductivity

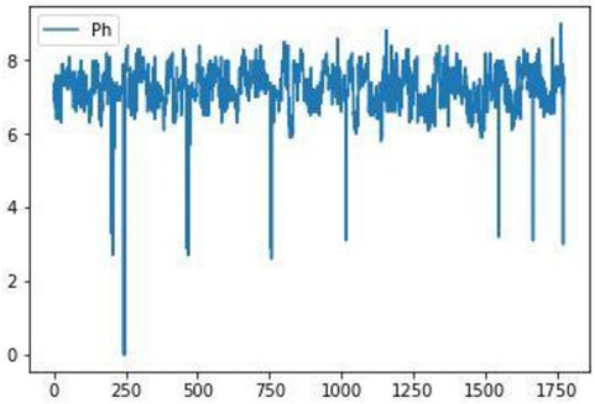


Figure 13: seasonality of PH value

As we see in these plots the pollutant concentrations tend to vary a lot, it depends on the location, season and other affecting factors. These graphs have no increasing or decreasing trends on their measure. So various data cleaning techniques will be used to clean the data.

Outlier analysis: -

In this problem there are various outlier on the pollution concentration from various sensor readings, so box plot outlier analysis is used to identify and remove the outliers from the Dataframe. The box plot consist of various quartiles Q1,Q2,Q3. Q1 and Q3 are the first and third quartile and Q2 is the median in the Dataframe. After the Q1 region the points are called smallest non outlier and Q3 region has largest outlier.

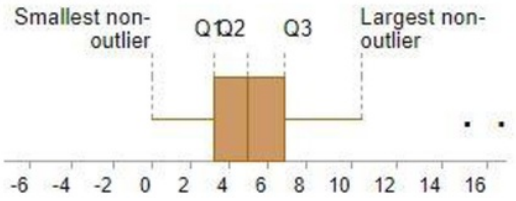


Figure 14: Box plot analysis

The datapoints which are away from the Q1 and Q3 regions are classified as outlier from the Dataframe. These outliers should be removed so the data will be cleaned, and various machine learning algorithms can be applied.

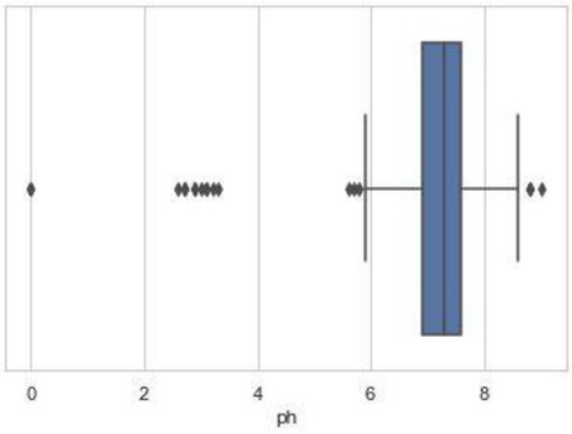


Figure 15: Boxplot analysis for PH values

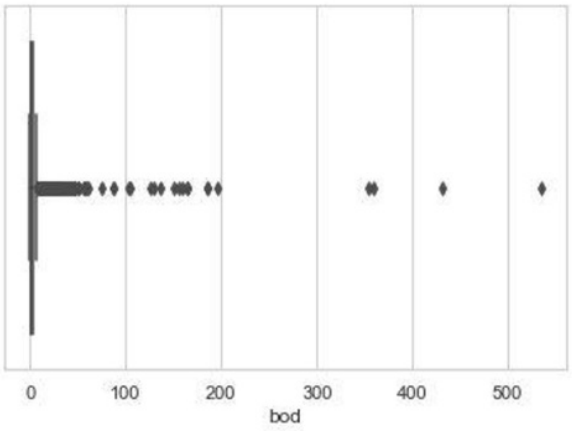


Figure 16:Boxplot analysis for B.O.D

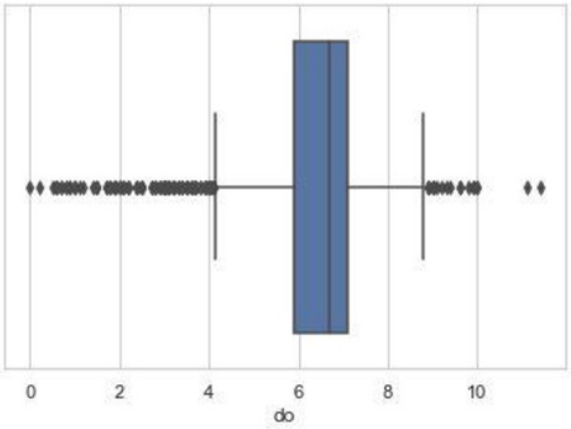


Figure 17:Boxplot analysis for D.O

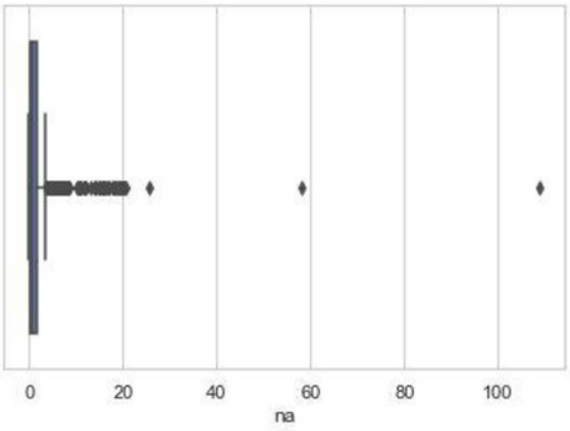


Figure 18: Boxplot analysis for NITRATE

Correlation matrix of features:

Correlation matrix is generated for the Dataframe to identify the dependency or relationship between the features. Various features like do, ph, co, bod, na, tc data values are taken and correlation matrix is generated. This also helps in feature selection of the choosen dataframe

	do	ph	co	bod	na	tc
do	1	0.051	-0.16	-0.31	-0.21	-0.15
ph	0.051	1	0.093	0.093	0.081	0.02
co	-0.16	0.093	1	0.13	0.058	0.0033
bod	-0.31	0.093	0.13	1	0.15	0.24
na	-0.21	0.081	0.058	0.15	1	-0.0021
tc	-0.15	0.02	0.0033	0.24	-0.0021	1

Figure 19: Correlation matrix of water quality data

Generated the heat maps for the correlation matrix to identify the level of dependency and relationship among the features.

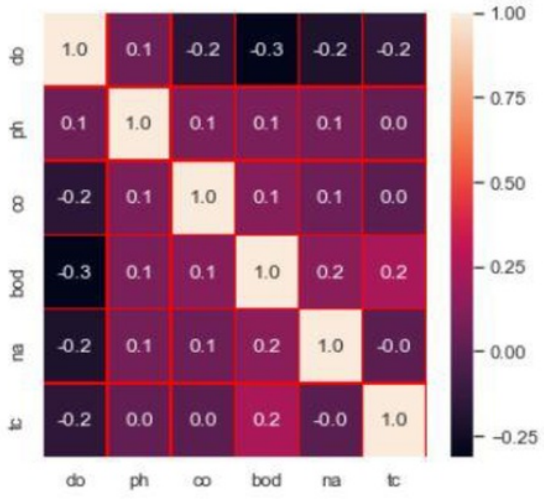


Figure 20:Heat maps of Correlation matrix

Calculating water quality index (WQI): -

Water quality index (WQI) is valuable and unique rating to depict the overall water quality status in a single term that is helpful for the selection of appropriate treatment technique to meet the concerned issues. However, WQI depicts the composite influence of different water quality parameters and communicates water quality information to the public and legislative decision makers. In spite of absence of a globally accepted composite index of water quality.

The water quality index of a particular data point is the aggregate of maximum indexed pollutant on that particular area. That pollutants max sub index is taken as the air quality index of that particular location. This maximum value of the pollutants is taken as water quality index because to backtrack the pollutant levels from the water quality index.

National Sanitation Foundation Water Quality Index (NSFWQI)	
WQI Value	**Rating of Water Quality**
91-100	Excellent water quality
71-90	Good water quality
51-70	Medium water quality
26-50	Bad water quality
0-25	Very bad water quality
Canadian Council of Ministers of the Environment Water Quality Index (CCME WQI)	
95-100	Excellent water quality
80-94	Good water quality
60-79	Fair water quality
45-59	Marginal water quality
0-44	Poor water quality
Oregon Water Quality Index (OWQI)	
90-100	Excellent water quality
85-89	Good water quality
80-84	Fair water quality
60-79	Poor water quality
0-59	Very poor water quality

Table 3: WQI range

This method for comparing the water quality of various water sources is based upon nine water quality parameters such as temperature, pH, turbidity, feral coliform, dissolved oxygen, biochemical oxygen demand, total phosphates, nitrates and total solids. The water quality data are recorded and transferred to a weighting curve chart, where a numerical value of Qi is obtained. The mathematical expression for WQI is given by as per Indian govt.

$$WQI = \sum_{i=1}^{n} QiWi$$

Where,

Qi = sub-index for ith water quality parameter;

Wi = weight associated with ith water quality parameter;

n = number of water quality parameters.

Calculating individual index of nitrate:

```
#Calulation of nitrate
data['nna']=data.na.apply(lambda x:(100 if (20>=x>=0)
                                    else(80 if  (50>=x>=20)
                                         else(60 if (100>=x>=50)
                                              else(40 if (200>=x>=100)
                                                   else 0)))))

data.head()
data.dtypes
```

Calculating individual index of connectivity:

```
#calculation of electrical conductivity
data['nec']=data.co.apply(lambda x:(100 if (75>=x>=0)
                                    else(80 if  (150>=x>=75)
                                         else(60 if (225>=x>=150)
                                              else(40 if (300>=x>=225)
                                                   else 0)))))
```

Calculating individual index of B.O.D:

```
#calc of B.D.O
data['nbdo']=data.bod.apply(lambda x:(100 if (3>=x>=0)
                                    else(80 if  (6>=x>=3)
                                         else(60 if (80>=x>=6)
                                              else(40 if (125>=x>=80)
                                                   else 0)))))
```

Calculating individual index of coliform:

```
#calculation of total coliform
data['nco']=data.tc.apply(lambda x:(100 if (5>=x>=0)
                                    else(80 if  (50>=x>=5)
                                         else(60 if (500>=x>=50)
                                              else(40 if (10000>=x>=500)
                                                   else 0)))))
```

Calculating individual index of dissolved oxygen:

```
#calculation of dissolved oxygen
data['ndo']=data.do.apply(lambda x:(100 if (x>=6)
                                    else(80 if  (6>=x>=5.1)
                                         else(60 if (5>=x>=4.1)
                                              else(40 if (4>=x>=3)
                                                   else 0)))))
```

Calculating individual index of Ph value:

```
#calulation of Ph
data['npH']=data.ph.apply(lambda x: (100 if (8.5>=x>=7)
                                  else(80 if  (8.6>=x>=8.5) or (6.9>=x>=6.8)
                                       else(60 if (8.8>=x>=8.6) or (6.8>=x>=6.7)
                                            else(40 if (9>=x>=8.8) or (6.7>=x>=6.5)
                                                 else 0)))))
```

Calculating Water quality index:

```
data['wph']=data.npH * 0.165
data['wdo']=data.ndo * 0.281
data['wbdo']=data.nbdo * 0.234
data['wec']=data.nec* 0.009
data['wna']=data.nna * 0.028
data['wco']=data.nco * 0.281
data['wqi']=data.wph+data.wdo+data.wbdo+data.wec+data.wna+data.wco
data
```

	station	location	state	do	ph	co	bod	na	tc	year	...
2	1475	ZUARI AT PANCHAWADI	GOA	6.300	6.900	179.0	1.700	0.100	5330.0	2014	...
3	3181	RIVER ZUARI AT BORIM BRIDGE	GOA	5.800	6.900	64.0	3.800	0.500	8443.0	2014	...
4	3182	RIVER ZUARI AT MARCAIM JETTY	GOA	5.800	7.300	83.0	1.900	0.400	5500.0	2014	...
5	1400	MANDOVI AT NEGHBOURHOOD OF PANAJI, GOA	GOA	5.500	7.400	81.0	1.500	0.100	4049.0	2014	...
6	1476	MANDOVI AT TONCA, MARCELA, GOA	GOA	6.100	6.700	308.0	1.400	0.300	5672.0	2014	...
7	3185	RIVER MANDOVI AT AMONA BRIDGE	GOA	6.400	6.700	414.0	1.000	0.200	9423.0	2014	...
8	3186	RIVER MANDOVI AT IFFI JETTY	GOA	6.400	7.600	305.0	2.200	0.100	4990.0	2014	...

Table 4.1: Resultant Dataframe after calculating individual index

nbdo	nec	nna	wph	wdo	wbdo	wec	wna	wco	wqi
100	60	100	13.2	28.10	23.40	0.54	2.80	11.24	79.28
80	100	100	13.2	22.48	18.72	0.90	2.80	11.24	69.34
100	80	100	16.5	22.48	23.40	0.72	2.80	11.24	77.14
100	80	100	16.5	22.48	23.40	0.72	2.80	11.24	77.14
100	0	100	9.9	28.10	23.40	0.00	2.80	11.24	75.44
100	0	100	9.9	28.10	23.40	0.00	2.80	11.24	75.44
100	0	100	16.5	28.10	23.40	0.00	2.80	11.24	82.04

Table 4.2: Resultant Dataframe after calculating individual index

Virtualizing water Quality index: -

```
#visualizing the filttered data

year=data['year'].values
AQI=data['wqi'].values
data['wqi']=pd.to_numeric(data['wqi'],errors='coerce')
data['year']=pd.to_numeric(data['year'],errors='coerce')

import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (20.0, 10.0)
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
ax = Axes3D(fig)
ax.scatter(year,AQI, color='red')
plt.show()
data
```

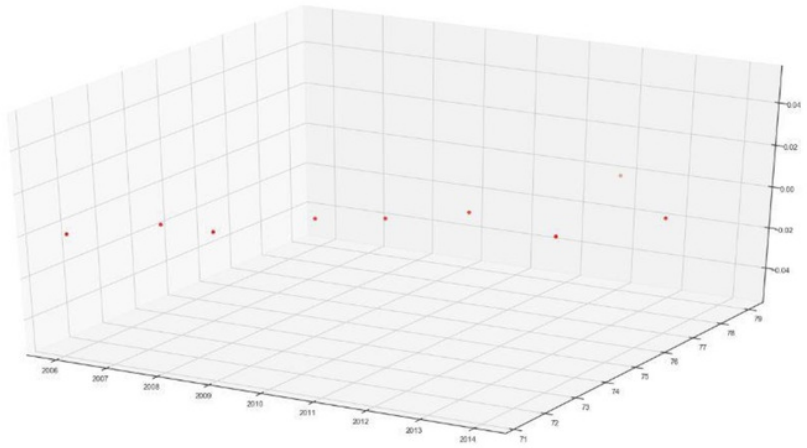


Figure 21: Scatter plot of data points

```
In [26]: x=data.set_index('year')

         import seaborn as sns
         fig = plt.figure(figsize=(12,12))
         r = sns.heatmap(x, cmap='BuPu')
         r.set_title("Heatmap of Flight Density from 1949 to 1961")

Out[26]: Text(0.5, 1.0, 'Heatmap of Flight Density from 1949 to 1961')
```

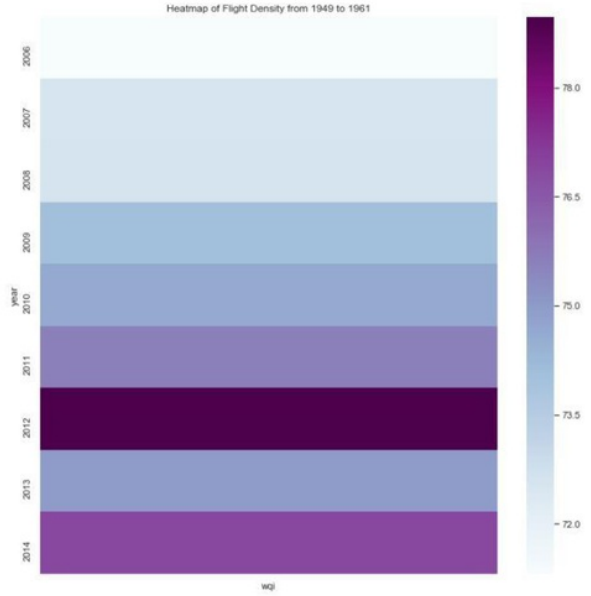


Figure 22: Heat map of WQI vs year

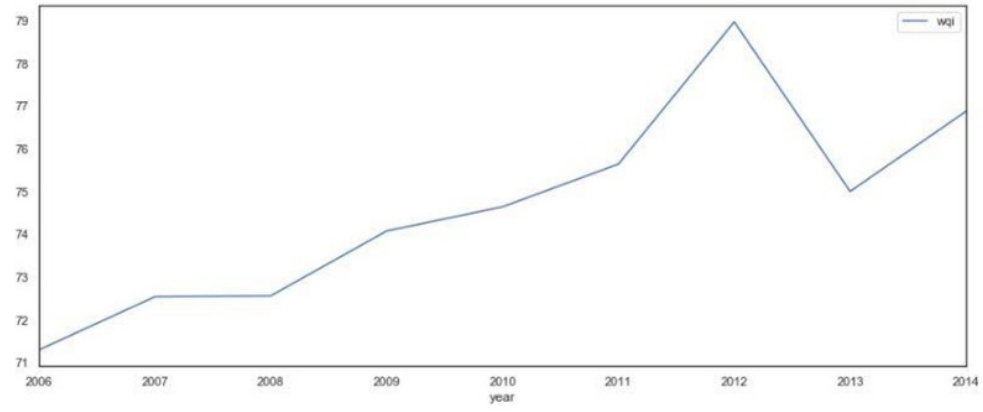


Figure 23: WQI vs year

Logistic regression:

Logistic regression dates back to twenith century .It was applied in various biological and social science research.Recently they are implemented for categorical problems .

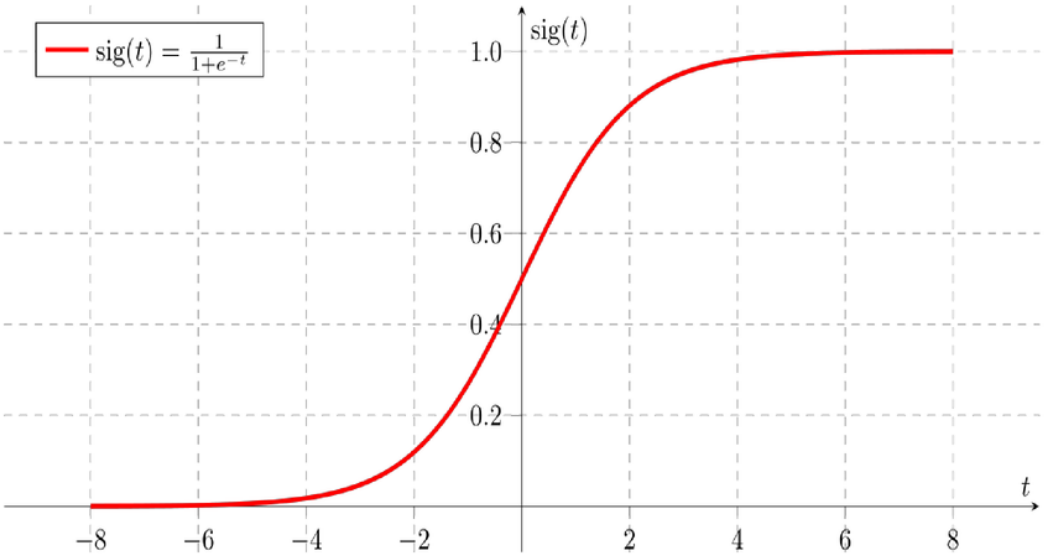


Figure 24: non-linear graph

Some of the examples include:

1) Predicting if the mail is spam or ham

2) Classifying whether the customers are genuine or not

LINEAR VS LOGISTIC REGRESSION

The reason for implementation of logistic regression instead of linear regression is because

1) In case of linear regression, we must fix an threshold for classification kind of problem since it is designed to predict continuous value .so linear regression cannot be used for this problem.

2)unbounded nature of linear regression makes it hard for classification problems while in logistic regression they vary from 0 to 1.

Linear regression threshold fixation

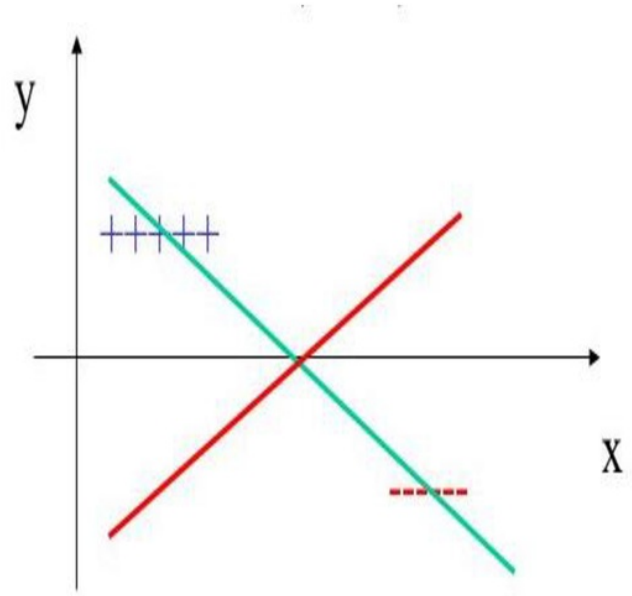


Figure 25: logistic regression for classification

Hence logistic regression are preferred for classification problems. It is named as logistic regression since it uses logit function to classify.

In classification problems, input variables are continuous and the outputs are in categorical form.

Simple logistic regression models use sigmoid function.standard logistic function is as follows.it takes values from 0 and 1.

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

Here t is a linear function

Model

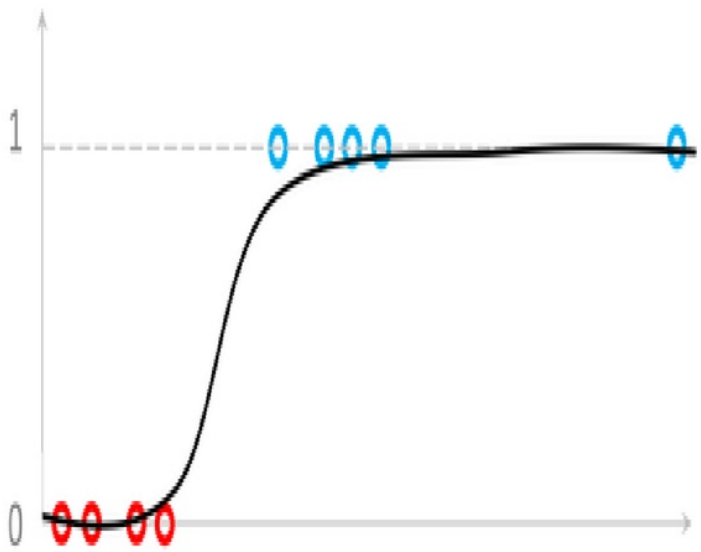


Figure 26: Here t is considered as a linear function.0 stands for negative and 1 stands for positive class

$$t = \beta_0 + \beta_1 x$$

Applying this to our model it will become

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

And one of the advantage of using logistic regression is that it will take care of the outliers.

LOGISTIC REGRESSION TYPES:

BINARY

MULTI CLASS

ORDINAL

BINARY –

Here we have two output classes 0 and 1.email,fraudent,tumor problems comes into this category.

1)SIGMOIDFUNCTION

$$S(z) = \frac{1}{1+e^{-z}}$$

Here s(z) represents the output that varies from 0 to 1

e is the natural logarithm base

z is the input function

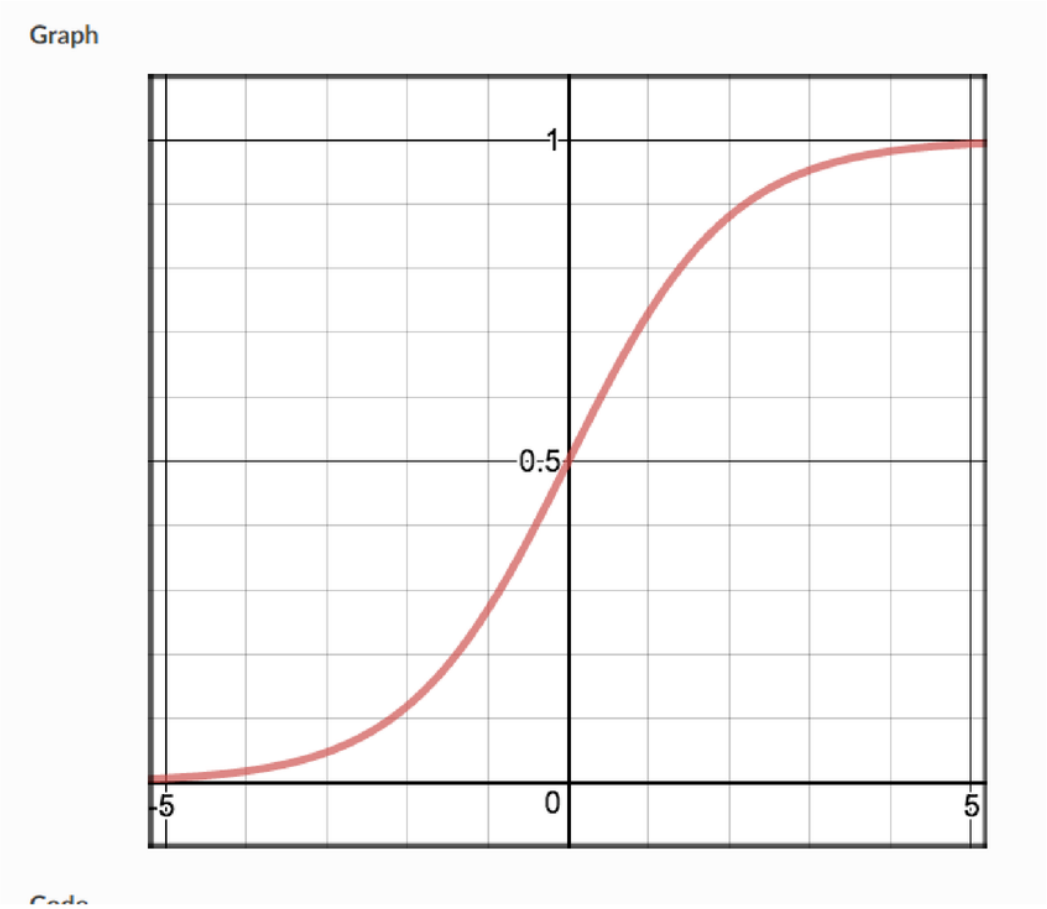


Figure 27: sigmoid function graph

It typically ranges from 0 to 1 and has

1)step function that is non linear in nature.

2)change in value of x influences the value of y .

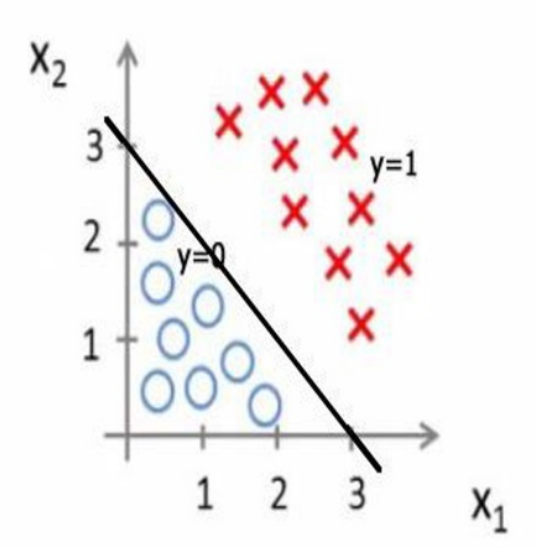
3)they are simple since output is between 0 and 1.

They can result in vanishing gradient problem since there is no learning at the end and also they are not zero centric.

They are preferred less now a days but work well with classification problems especially binary ones.

Decision boundary

A score between 0 and 1 is given as output using the current function(0/1 class),here we use a tipping point to classify them into the relevant classes. They can be linear or non linear in nature.



For y = 1, Equation of line would be $x_1 + x_2 >= 3$

For y = 0, Equation of line would be $x_1 + x_2 < 3$

Figure 28: graph showing the decision boundaries

Decision boundary can be considered similar to threshold. For example ,consider a decision boundary around 0.5,if there is a prediction of value 0.2 it will belong to class 0 or if its above 0.5 ,then to class 1.

To predict complex problems degree of the polynomial can be increased.

Non Linear Decision Boundary

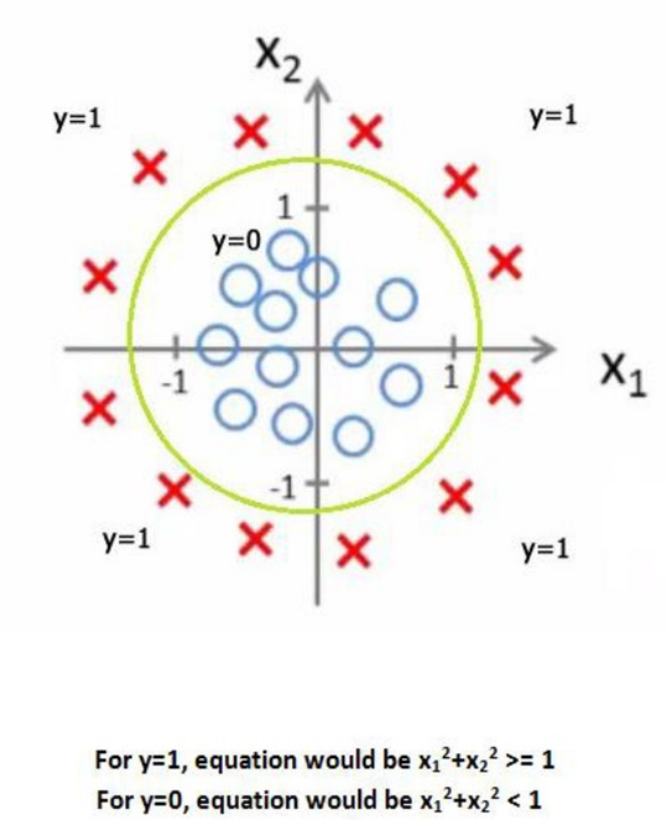


Figure 29: non linear decision boundary

Cost function-

The linear regression cost function does not work well here.we will end up with a non convex function by doing so.

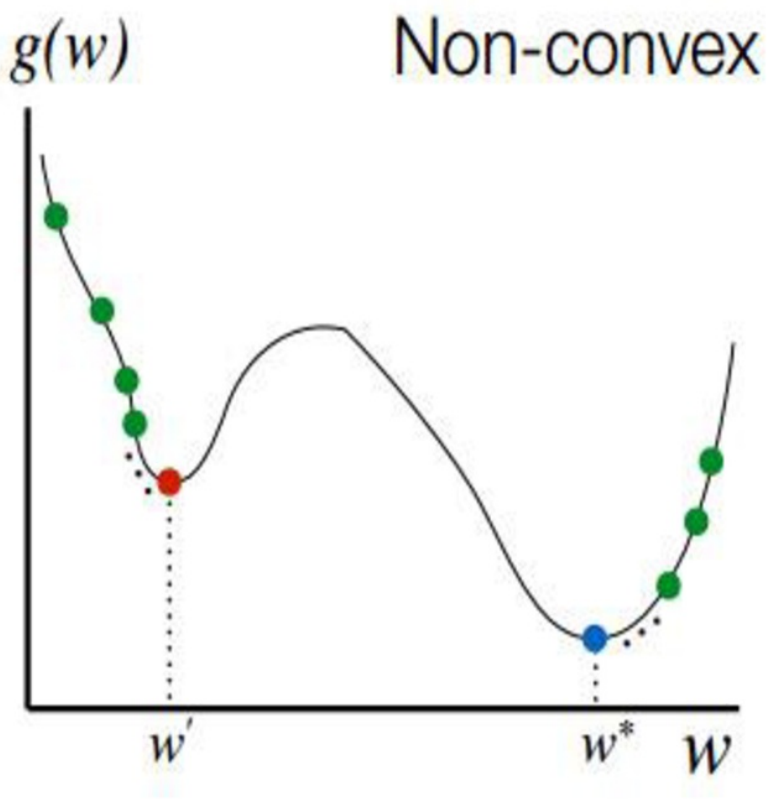


Figure 30: various local minima's in the same problem

Here it is difficult to find the local minima.this is due to the presence of non linear sigmoid function.there are chances of getting struck at these points also termed as local minima and leading to poor performance.

Simplified cost function

So we need a better function for logistic regresssion due to problem stated above.

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \text{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$\text{Cost}(h_\theta(x), y) = -\log(h_\theta(x)) \qquad \text{if } y = 1$$

$$\text{Cost}(h_\theta(x), y) = -\log(1 - h_\theta(x)) \qquad \text{if } y = 0$$

This is known as cross entropy.it seperates the cost function into two –one for class 0 and other for class 1.it helps in finding minimum cost easily

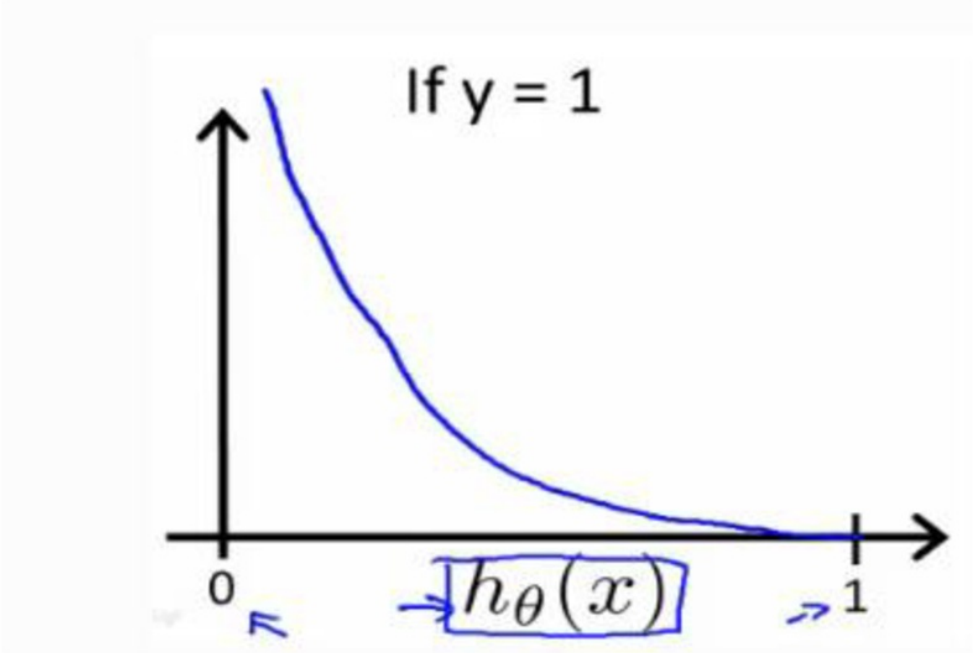


Figure 31: Representation of class 1

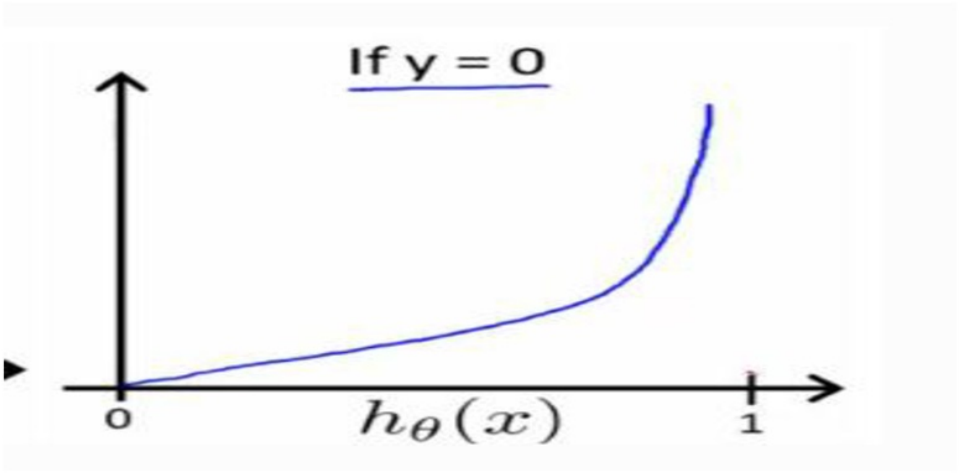


Figure 32: Representation of class 0

Combined cost function is as follows.

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))]$$

If y equals 1,second part after addition cancels out and if its zero the first part will become zero and provides better metrics.

Gradient descent:

It is an optimization algorithm used for classical problems.In real life scenario ,take an example where your at the mountain top and you want to proceed to the lake at the lowest point of the mountain.the best route is to look outside and check the path that descends down and advance.

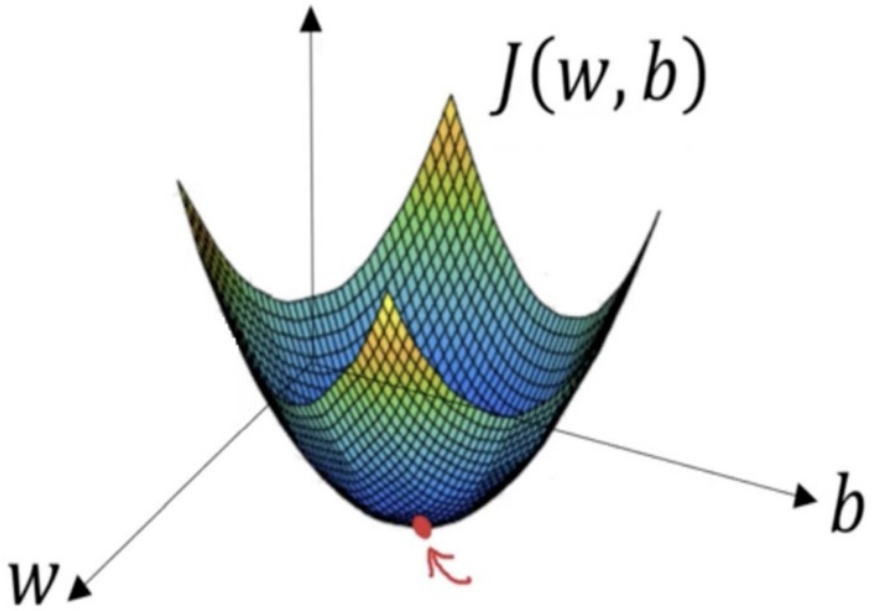


Figure 33: representation of gradient decent

This is used to find the best parameters for our algorithm. They are generally classified as

1)full batch

2)stochastic

Based on techniques for differentiation they are classified as

1)first order

2)second order

To represent this graphically, notice the below graph.

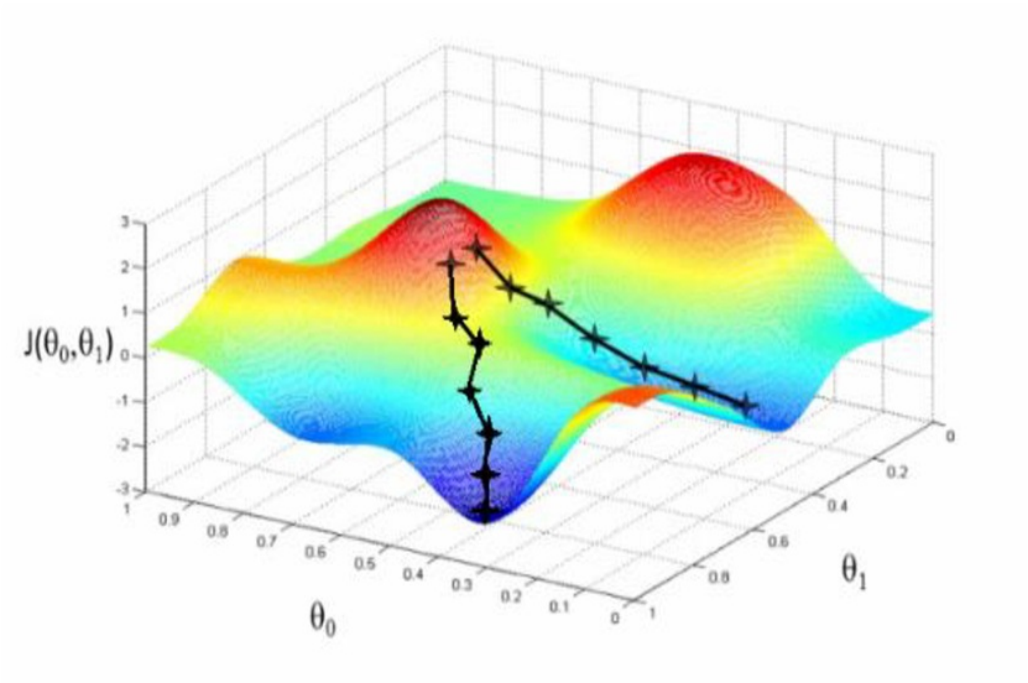


Figure 34: issues in gradient descent

There are various issues with the gradient descent that include

1) Data and input issues

2) Gradient issues

3) Deployment issues

DATA ISSUES

Non convex optimization problems can be inferred due to the way the data is arranged leading to local minima constriants.it works well with problems that have convex optimization problem.

Another issue is saddle point where the gradient becomes zero but it is not the optimal point.

ISSUES WITH GRADIENT

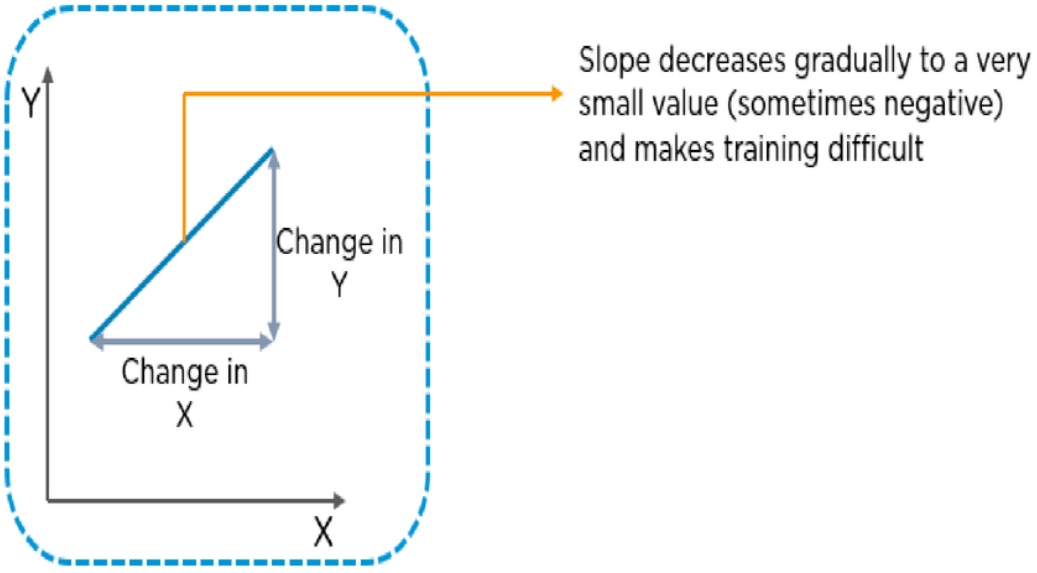


Figure 35: vanishing gradient problem

There is a possibility of vanishing gradient and exploding gradient problems and the model will not converge well.

DEPLOYMENT ISSUES

How well resources are used by networks must also be taken care .

GRADIENT DESCENT

It is an optimization algorithm that minimizes the cost function by choosing the values of parameters correctly.it is used when we cannot calculate parameters manually or when it is computationally expensive.

Values for coefficients are picked ,cost function is computed then new coefficients are estimated.

Repeating the method for a few times will lead to choosing the best values for parameters and reducing the cost function.

PROCEDURE

Initial values for the parameters are chosen they can be 0 or very small values.

They are then put into a function and cost of coeffients is computed.

Algorithm 2: Gradient Descent

input : $f : \mathbb{R}^n \to \mathbb{R}$ a differentiable function
$\mathbf{x}^{(0)}$ an initial solution

output: $\mathbf{x}^\star$, a local minimum of the cost function f.

```
begin
    k ← 0 ;
    while STOP-CRIT and (k < k_max) do
        x^(k+1) ← x^(k) − α^(k) ∇f(x) ;
        with α^(k) = arg min_{α∈ℝ+} f(x^(k) − α∇f(x)) ;
        k ← k + 1 ;
    return x^(k)
end
```

The calculus derivation is also computed for a functions slope at a particular point.we need to estimate the direction in which the slope moves .it should move downhill.this is done until the problem is optimized.

STOCHASTIC GRADIENT DESCENT

Gradient descent becomes slower when the dataset becomes larger.because in each iteration,one instance is predictied and we will have millions of data that are present.in such situations we can use this type of gradient descent.

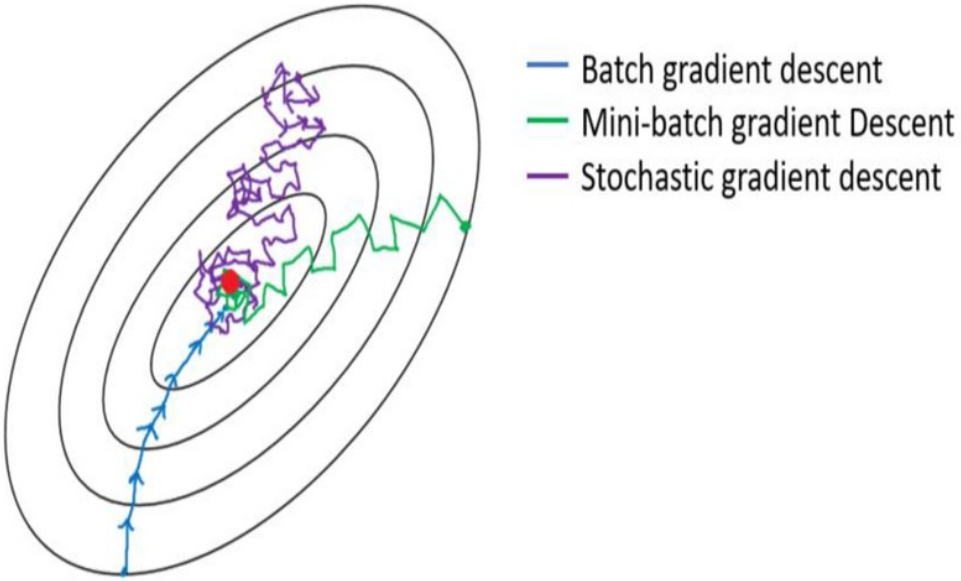


Figure 36: choosing gradient descent type

for each instance of training set, we update the coefficents.at each instance the weight are updated and this will result in more noise.it can result in getting stuck at any point and is not optimal ,so we move on to the next type.

BATCH GRADIENT DESCENT

Supervised algorithms maps the input variable to the output function.there are different algorithms and representations that are present,but requires different optimization techniques.the cost function is evaluated for all the outputs and calculates the sum of average.it is implemented for the entire dataset and is the commonly used gradient.

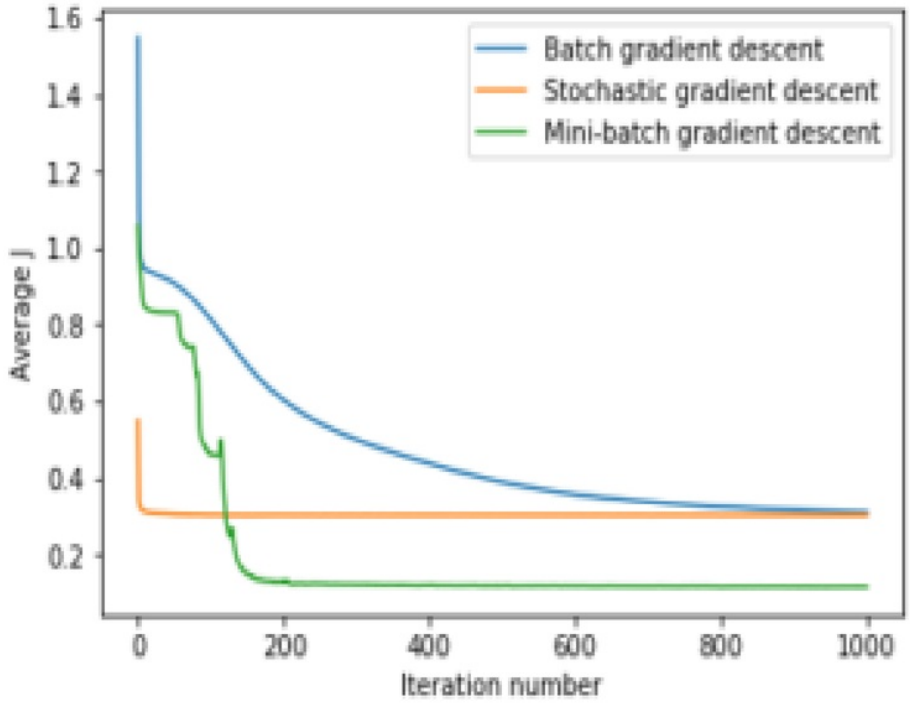


Figure 37: comparison of various gradient descent

VARIANTS OF GRADIENT DESCENT

VANNILA GRADIENT

It is a simple model and is pure and no adulteration takes place.updation of parameters is taken care.

```
update = learning_rate * gradient_of_parameters

parameters = parameters - update
```

Learning rate is then multiplied which represents a hyper parameter

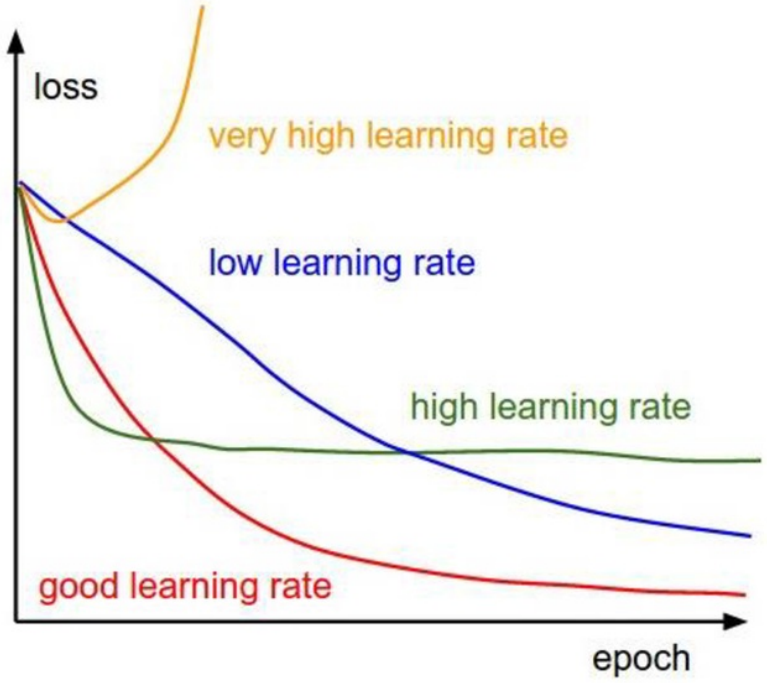


Figure 38: comparison of learning rate

GRADIENT DESCENT WITH MOMENTUM

```
update = learning_rate * gradient

velocity = previous_update * momentum

parameter = parameter + velocity - update
```

It is same as the previous one except they include a term called velocity that considers the momentum which is aconstant.

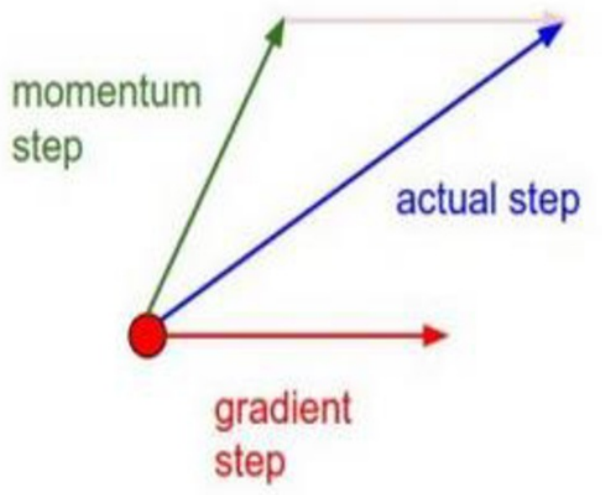


Figure 39: gradient step representation

ADAGRAD

Learning rate is updated using a technique that is adaptive.it depends upon the iterations .based on previous iterations the learning rate is changed.

Pseudo code is as follows

Epsilon keeps a check on learning rate change.

```
grad_component = previous_grad_component + (gradient * gradient)

rate_change = square_root(grad_component) + epsilon

adapted_learning_rate = learning_rate * rate_change
```

ADAM

It combines momentum and adagrad.

```
adapted_gradient = previous_gradient + ((gradient - previous_gradient) * (1 - beta
1))


gradient_component = (gradient_change - previous_learning_rate)
adapted_learning_rate =  previous_learning_rate + (gradient_component * (1 - beta
2))
```

```
update = adapted_learning_rate * adapted_gradient
parameter = parameter - update
```

APPLICATION OF GRADIENT DESCENT IN VARIOUS CASES

1)Used adam/adagrad in prototyping,we can get the results faster and there is no complexity involved.hyperparameters need not be used in this case.

2)by using vanilla descent or momentum it is possible to obtain the best results than using normal gradient descent since it is adoptive technique.

3)if the data size is very low ,second order techniques can be implemented.

Also the following should be checked.they include

1)error rates

We must check them regularly and see if they do down and does not lead to overfitting or underfitting problem

2)gradient flow

They should not result in vanishing or exploding descent problem

3)learning rate

They are very important for our model to perform and learn better

After removing the outlier's linear regression is applied to the filtered data and to fit the trent line on the data points gradient descent hyper parameters are used to optimize the model. We set random values to our slope equation and change according to the learning rate and position of data points.

Applying Logistic regression with Gradient decent: -

(y=mx+c)

y = B1 + B2 * x

B1 = 0.0

B2 = 0.0

y = 0.0 + 0.0 * x

error = p(j) – y(j)

x=1, y=1

p(j) = 0.0 + 0.0 * 1

p(j) = 0:

we calculated the error by: -

error = 0 – 1

error = -1

We can now use this error rate in our equation for gradient descent to update the its weights. Then we will start with updating the solpe intercept first.

B1(t+1) = B2(t) – alpha * error mean

B1(t+1) = B2(t) – alpha * error mean * x

B2(t+1) = 0.0 – 0.01 * -1 * 1

B2(t+1) = 0.01

We have just finished the first iteration of gradient descent and we have updated our weights to be B1=0.01 and B2=0.01. This process must be repeated for the remaining 'x' number of instances from our dataset. (x=3000)

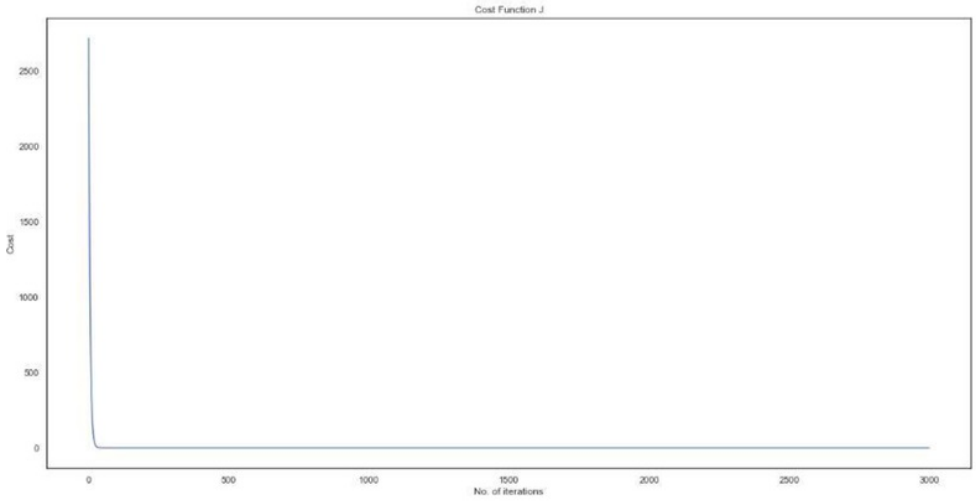


Figure 40: Plotting the cost function

Gradient descent for desired features:

```
alpha = 0.1 #Step size
iterations = 3000 #No. of iterations
m = y.size #No. of data points
np.random.seed(4) #Setting the seed
theta = np.random.rand(2) #Picking some random values to start with

def gradient_descent(x, y, theta, iterations, alpha):
    past_costs = []
    past_thetas = [theta]
    for i in range(iterations):
        prediction = np.dot(x, theta)
        error = prediction - y
        cost = 1/(2*m) * np.dot(error.T, error)
        past_costs.append(cost)
        theta = theta - (alpha * (1/m) * np.dot(x.T, error))
        past_thetas.append(theta)

    return past_thetas, past_costs

past_thetas, past_costs = gradient_descent(x, y, theta, iterations, alpha)
theta = past_thetas[-1]

#Print the results...
print("Gradient Descent: {:.2f}, {:.2f}".format(theta[0], theta[1]))
```

```
Gradient Descent: 74.63, 2.01
```

X1= 74.63

X2= 2.01

Scaling:

Converting out input data points into smaller values, to reduce the complexity of the algorithm. All the available datapoints are scaled to smaller values ranging from -1 to 1. This process is called one hot encoding

```
#using gradient descent to optimize it further
x = (x - x.mean()) / x.std()
x = np.c_[np.ones(x.shape[0]), x]
x
```

```
array([[ 1.        , -1.46059349],
       [ 1.        , -1.09544512],
       [ 1.        , -0.73029674],
       [ 1.        , -0.36514837],
       [ 1.        ,  0.        ],
       [ 1.        ,  0.36514837],
       [ 1.        ,  0.73029674],
       [ 1.        ,  1.09544512],
       [ 1.        ,  1.46059349]])
```

5.RESULTS:

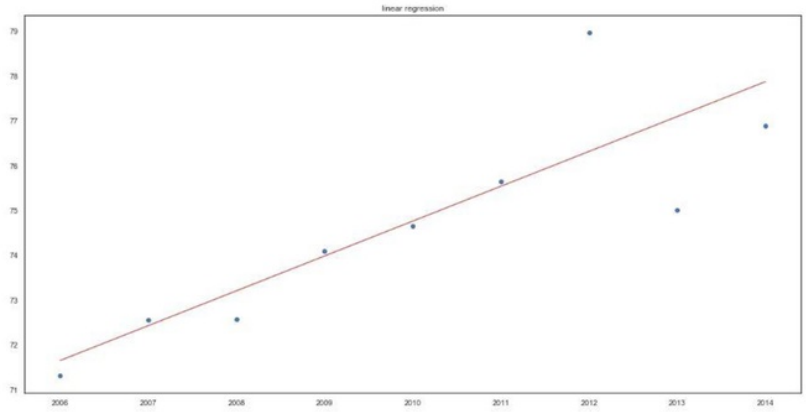


Figure 41: Plotting Logistic regression on WQI data

	year	wqi	Actual	Predicted
0	2006	71.308824	71.308824	71.648936
1	2007	72.549000	72.549000	72.426702
2	2008	72.570943	72.570943	73.204468
3	2009	74.085193	74.085193	73.982234
4	2010	74.648723	74.648723	74.760000
5	2011	75.647013	75.647013	75.537766
6	2012	78.969041	78.969041	76.315532
7	2013	75.009425	75.009425	77.093298
8	2014	76.879588	76.879588	77.871064

Table 8: table showing Predicted data of WQI

RMSE for actual vs predicted:

```
#testing the accuracy of the model

from sklearn import metrics
print(np.sqrt(metrics.mean_squared_error(y,y_pred)))
```

```
1.1987755149740886
```

2.5 References: -

1. Tan, Hüseyin Özgür, and Ibrahim Körpeoğlu. "Power efficient data gathering and aggregation in wireless sensor networks." *ACM Sigmod Record* 32.4 (2003): 66-71.
2. Dobslaw, Felix, Tingting Zhang, and Mikael Gidlund. "QoS assessment for mission-critical wireless sensor network applications." *2013 IEEE 38th Conference on Local Computer Networks (LCN 2013)*. IEEE, 2013.
3. Stojkoska, Biljana, and Kliment Mahoski. "Comparison of different data prediction methods for wireless sensor networks." *CIIT, Bitola* (2013).
4. Stojkoska, Biljana, and Kliment Mahoski. "Comparison of different data prediction methods for wireless sensor networks." *CIIT, Bitola* (2013).
5. Stojkoska, Biljana, and Kliment Mahoski. "Comparison of different data prediction methods for wireless sensor networks." *CIIT, Bitola* (2013).
6. Kumar, Dilip. "Performance analysis of energy efficient clustering protocols for maximising lifetime of wireless sensor networks." *IET Wireless Sensor Systems* 4.1 (2014): 9-16.
7. El-Sayed, Hamdy H. "Data Aggregation Energy and Probability Effects on the Performance of EDEEC and MODLEACH Protocol in WSN." *Appl. Math* 12.1 (2018): 171-177.
8. Gupta, Gaurav, and Mohamed Younis. "Fault-tolerant clustering of wireless sensor networks." *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*. Vol. 3. IEEE, 2003.
9. De Souza, Luciana Moreira Sá, Harald Vogt, and Michael Beigl. "A survey on fault tolerance in wireless sensor networks." *Interner Bericht. Fakultät für Informatik, Universität Karlsruhe* (2007).
10. Iqbal, Muhammad, et al. "Wireless sensor network optimization: multi-objective paradigm." *Sensors* 15.7 (2015): 17572-17620.
11. Akkaya, Kemal, and Mohamed Younis. "A survey on routing protocols for wireless sensor networks." Ad hoc networks 3.3 (2005): 325-349.
12. Manjeshwar, Arati, and Dharma P. Agrawal. "TEEN: a routing protocol for enhanced efficiency in wireless sensor networks." null. IEEE, 2001.
13. Karlof, Chris, and David Wagner. "Secure routing in wireless sensor networks: Attacks and countermeasures." Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, 2003.. IEEE, 2003.
14. Heinzelman, Wendi Rabiner, Anantha Chandrakasan, and Hari Balakrishnan. "Energy-efficient communication protocol for wireless microsensor networks." Proceedings of the 33rd annual Hawaii international conference on system sciences. IEEE, 2000.
15. Yu, Yan, Ramesh Govindan, and Deborah Estrin. "Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks." (2001).
16. Manjeshwar, Arati, and Dharma P. Agrawal. "APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks." ipdps. IEEE, 2002.

17. Muruganathan, Siva D., et al. "A centralized energy-efficient routing protocol for wireless sensor networks." IEEE Communications Magazine 43.3 (2005): S8-13.
18. Schurgers, Curt, and Mani B. Srivastava. "Energy efficient routing in wireless sensor networks." 2001 MILCOM Proceedings Communications for Network-Centric Operations: Creating the Information Force (Cat. No. 01CH37277). Vol. 1. IEEE, 2001.

INDIAN WATER QUALITY TRACKING AND PREDICTION USING WSN AND MACHINE LEARNING

ORIGINALITY REPORT

4%	4%	0%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	econstor.eu Internet Source	4%

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		