CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Safe and readily available water is important for public health, whether it is used for drinking, domestic use, food production or recreational purposes. Better water supplies and sanitation, as well as better management of water resources, can contribute greatly to poverty reduction and economic growth. It is known that contaminated water and inadequate sanitation facilitate the transmission of diseases such as cholera, diarrhoea, dysentery, hepatitis A, typhoid, and polio. Those without access to clean water and sanitation face preventable health risks. We are going to implement a water quality prediction using machine learning techniques. In this technique, our model predicts that the water is safe to drink or not using some parameters like Ph value, conductivity, hardness, etc

1.2 PURPOSE

The purpose of this research is to develop a reliable method for forecasting water quality with a proposed model as precisely as necessary.

The following are the suitable approaches:In this study, missing data is handled using the Random Forest approach, and the dataset is splitting using the min-max normalization technique.Describe and demonstrate the dataset's significant distribution and feature correlation.Based on prior research, select the most important features for WQC and categorize three distinct types of water quality based on WQI rate.SVM, NN, MLR, BTM, and RF algorithms are used to optimize model performance.The proposed model approaches: develop a software application that uses the MLR algorithm to predict water quality in real time for these three types of WQ.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEMS

S.NO	AUTHOR'S NAME	PROPOSED WORK
1	Daud et al	Gathered water samples from different areas of Pakistan and tested them against different parameters using a manual lab analysis and found a high presence of E. coli and fecal coliform due to industrial and sewerage waste.
2	Alamgir et al	Tested 46 different samples from Orangi town, Karachi, using manual lab analysis and found them tobe high in and total fecal coliform count.After getting familiar with the water quality research concerning Pakistan, we explored research Employing machine learning methodologies in the realm of water quality.
3	Shafi et al	Estimated water quality using classical machine learning algorithms namely, Support Vector Machines (SVM), Neural Networks (NN), Deep Neural Networks (Deep NN) and k Nearest Neighbors (kNN), with the highest accuracy of 93% with Deep NN
4	Ahmad et al	Employed single feed forward neural networks and a combination of multiple neural networks toestimate the WQI. They used 25 water quality parameters as the

		input. Using a combination of backward elimination and forward selection selective combination methods, they achieved an R2 and MSE of 0.9270, 0.9390 and 0.1200, 0.1158, respectively. The use of 25 parameters makes their solution a little immoderate in terms of an inexpensive real time system, given the price of the parameter sensors.
5	Sakizadeh	Predicted the WQI using 16 water quality parameters and ANN with Bayesian regularization.
6	Rankovic et al.	Predicted the dissolved oxygen (DO) using a feedforward neural network (FNN). They used 10 parameters to predict the DO, which again defeats the purpose if it has to be used for a Real time WQI estimation of an IOT system.

Table 2.1 - Existing Problems

2.2 REFERENCES

1.TH Aldhyani, M Al-Yaari, H Alkahtani, and M Maashi, Water quality prediction using artificial intelligence algorithms, Applied Bionics and Biomechanics, Vol. 2020, 2020, pp. 6659314.

2. U Ahmed, R Mumtaz, H Anwar, AA Shah, R Irfan, and J García-Nieto, Efficient water quality prediction using supervised machine learning, Water, Vol. 11, 2019, pp. 2210.

3. P Liu, J Wang, AK Sangaiah, Y Xie, and X Yin, Analysis and prediction of water qualit usingLSTM deep neural networks in IoT environment, Sustainability, Vol. 11, 2019, pp. 2058.

4. L Wang, Z Zhu, L Sassoubre, G Yu, C Liao, Q Hu, et al., Improving the robustness of beach water quality modeling using an ensemble machine learning approach, Science of The Total Environment, Vol. 765, 2021, pp. 142760.

2.3 PROBLEM STATEMENT DEFINITION

Safe and readily available water is important for public health, whether it is used for drinking, domestic use, food production or recreational purposes. Better water supplies and sanitation, as well as better management of water resources, can contribute greatly to poverty reduction and economic growth. It is known that contaminated water and inadequate sanitation facilitate the transmission of diseases such as cholera, diarrhoea, dysentery, hepatitis A, typhoid, and polio. Those without access to clean water and sanitation face preventable health risks.

We are going to implement a water quality prediction using machine learning techniques. In this technique, our model predicts that the water is safe to drink or not using some parameters like Ph value, conductivity, hardness, etc.

Who does the problem?	People who are drinking impure water.
What is the issue?	Poor quality of potable, domestic use , or even recreational water due to contamination can lead to human illness.
When does the issue occurs?	Contact with suspended materials and elements such as sand, boulders and biological matter in rivers ,streams and lakes causes water to become undrinkable a non-pure water.
Why is it important that we fix the problem?	By solving this issue, people who drinking impure water can drink pure water.

Table 2.2 - Problem Statement Definition

CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

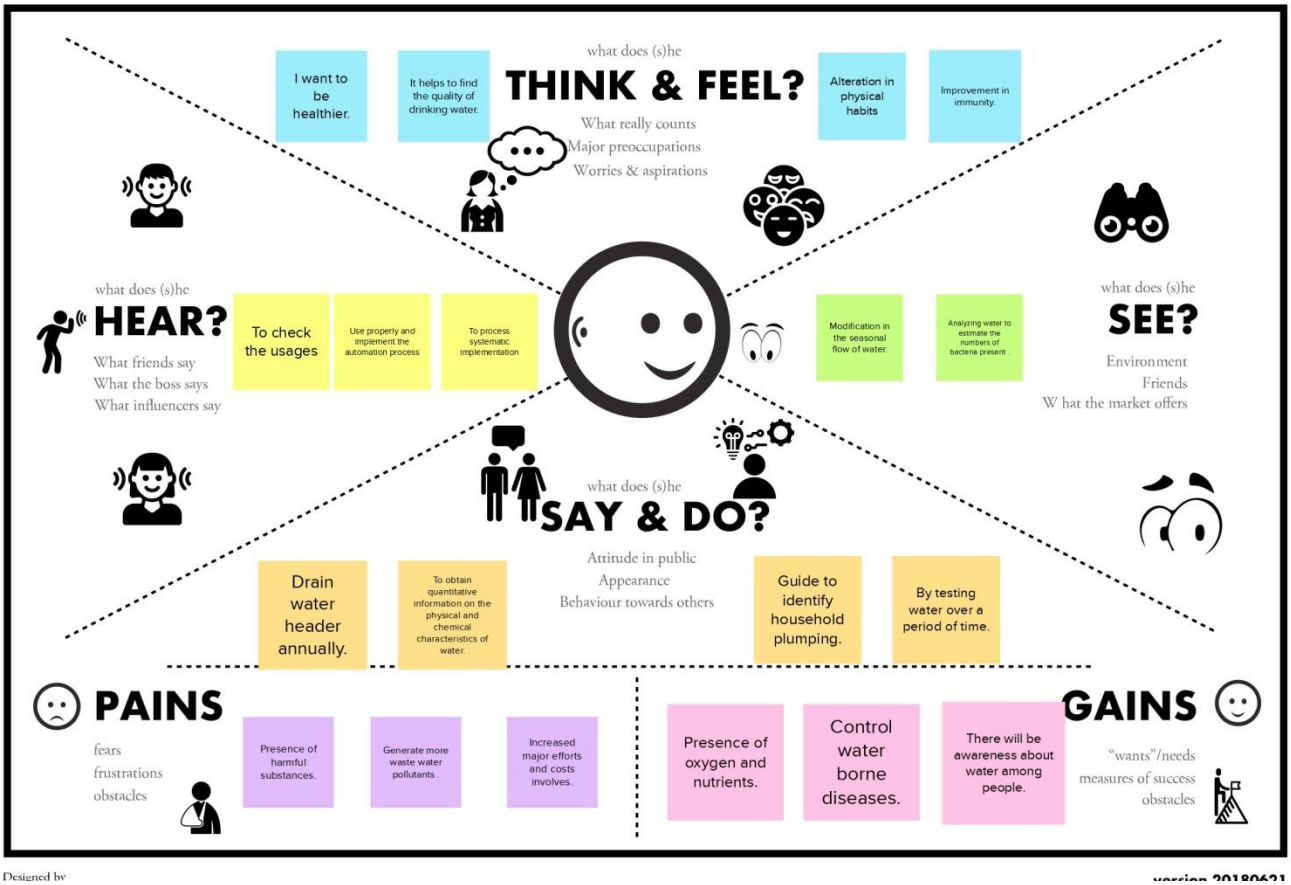


Figure 3.1 - Empathy Map

An empathy map is a widely-used visualization tool within the field. In relation to **empathetic** design, the primary purpose of an empathy map is to bridge the understanding of the end user.

3.2 IDEATION AND

BRAINSTORMINGStep-1:

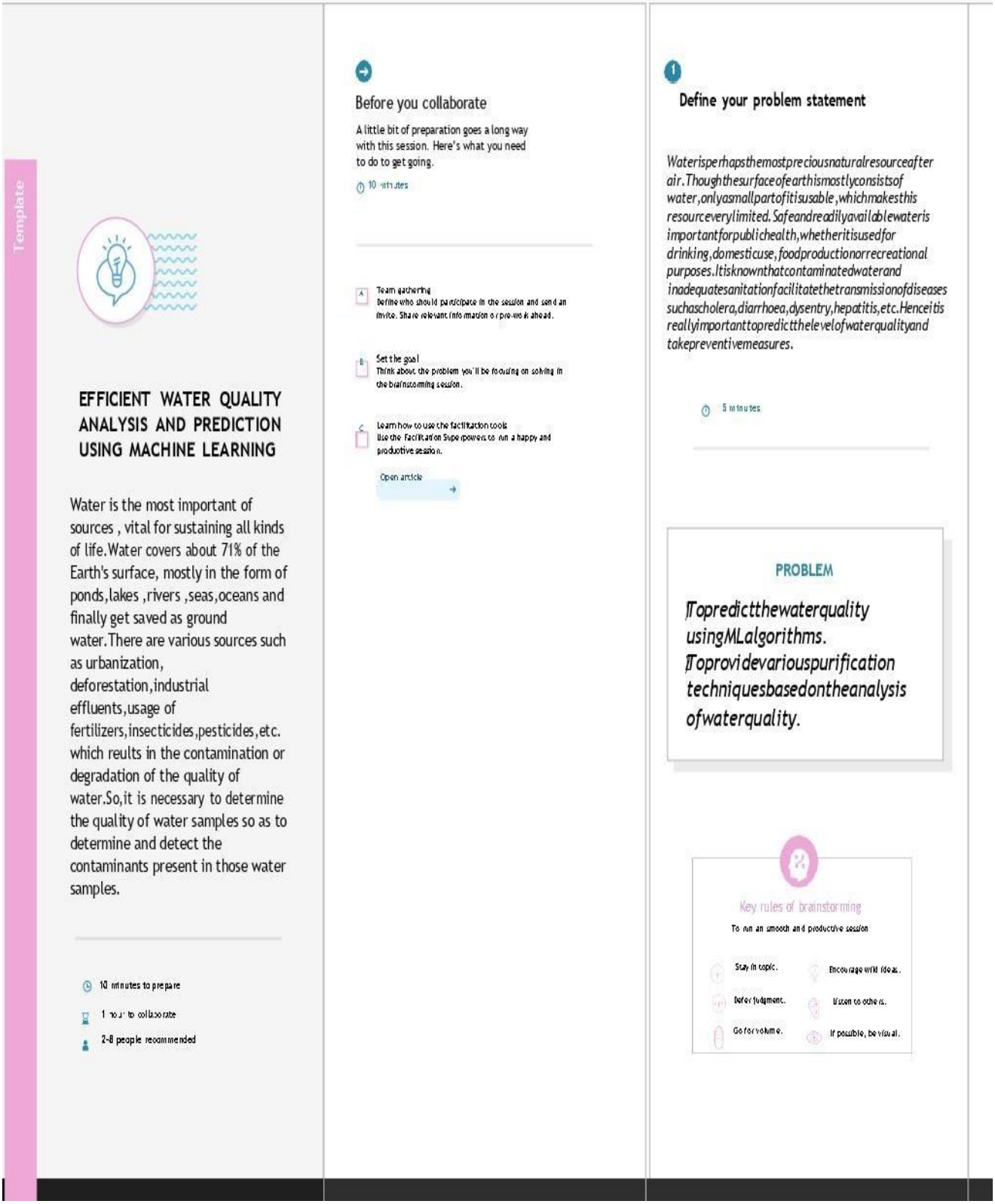
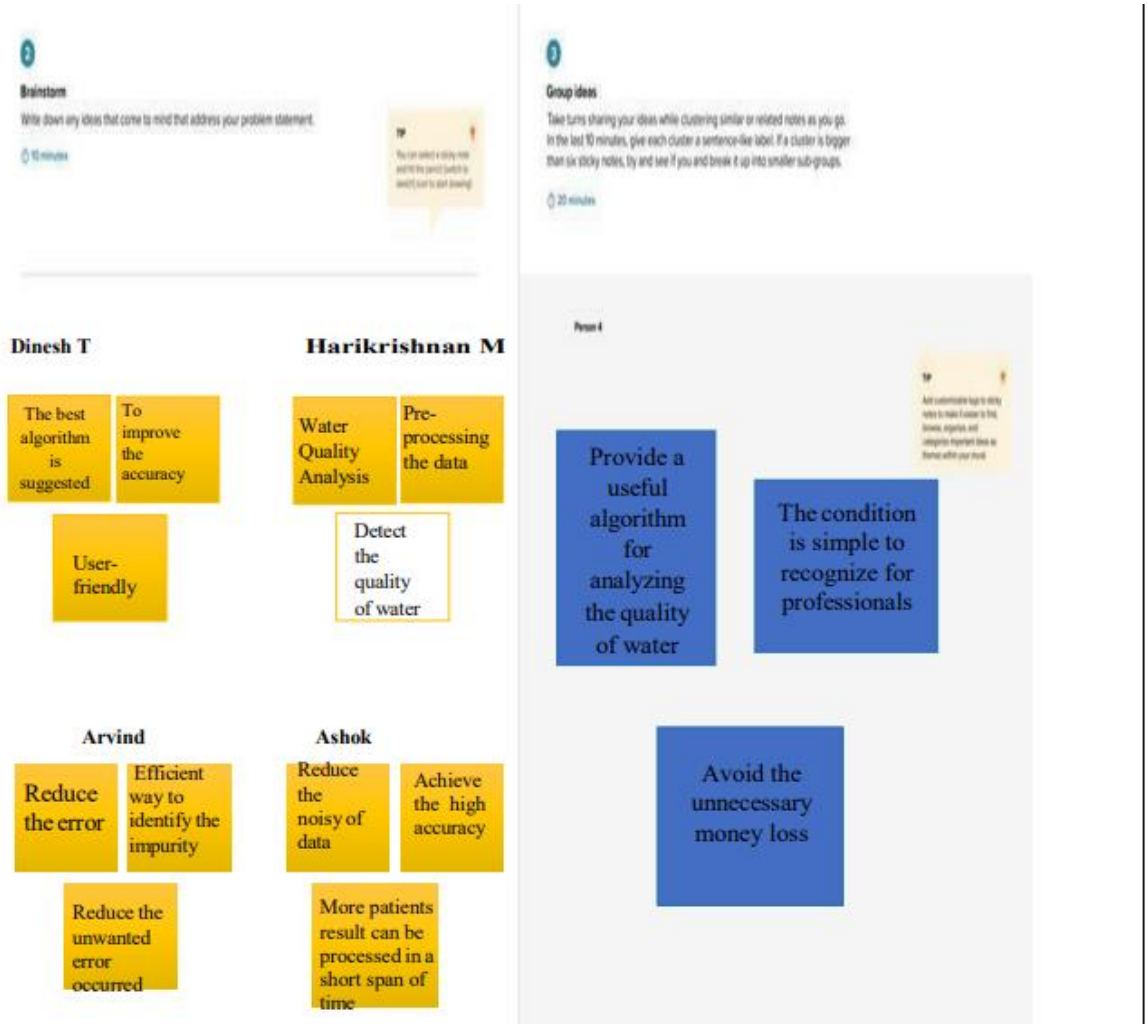


Figure 3.2 - Ideation and Brainstorming

A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity.

Step 2:

Figure 3.3 - Brainstorm, Idea Listing and Grouping



The idea listing and grouping is used to organize and analyse large numbers of ideas by categorising them. By organising and reorganising ideas, students gain a better appreciation of, and dialogue about, their ideas. As students create idea clusters, new contexts and connections among themes emerge.

Step 3:

Figure 3.4 - Idea Prioritization

Idea prioritization is just a part of the idea management process. Having a structured idea management process and a systematic way of gathering, evaluating and prioritizing new ideas takes time. To make it work, the entire idea management process should be integrated to the everyday ways of working.

3.3 PROPOSED SOLUTION

S.No.	PARAMETERS	DESCRIPTION
1.	Problem Statement (Problem to be solved)	At recent times water pollution have risen due to human activities and others. Due to this drinking water is not safe and there is a need to check for the water quality before using it for various purposes.
2.	Idea/Solution description	An analyzer allows to collect data about water and analyses the quality of the water and predicts the quality percentage of it and based on which we can utilize it.
3.	Novelty / Uniqueness	This analyzer is more efficient and allows user to get results in a better and more efficient way. This also utilizes needed data alone compared to others and provides more precise prediction.

4.	Social Impact / Customer Satisfaction	Since our project use efficient and low cost components it provides the product in very low price. It also has no harm towards the environment.
5.	Scalability of the Solution	Since our project use efficient and low cost components it provides the product in very low price. It also has no harm towards the environment.

Table 3.1 - Proposed Solution

3.4 PROBLEM SOLUTION FIT

Figure 3.5 - Solution fit of design with user requirements

This occurs when the user have evidence that customers care about certain jobs, pains, and gains. At this stage the user proved the existence of a problem and have designed a value proposition that addresses customers' jobs, pains and gains.

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 **Functional Requirements:**

Following are the functional requirements of the proposed solution.

FR No.	**Functional Requirement(Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	User Registration	Registration through Form. Registration through Gmail.
FR-2	User Confirmation	Confirmation via Email.Confirmation via OTP.
FR-3	User Login/ Authentication	Validation of Login ID and password.
FR-4	Machine Learning Model Deployment	Develop the Machine Learning Regression Modelto predict the Water Quality Index (WQI).Develop the Machine Learning ClassificationModel to predict theWater Quality Classification (WQC).
FR-5	Testing The Water Samples	Provides an option to test any kind of water sampleswith required parameters and to calculate the Water Quality Index and impurities present.
FR-6	Interface Function	Provides an interface to : 1. View the Water Quality Index value. 2. Display the Water Sample type. 3. Produce any purification techniquerecommended for the sample.
FR-7	Reporting	If any issues are faced by the customer or user itwill be directly notified to the developer.
FR-8	Compliance to Rules or Laws	Privacy Policy, Terms and Conditions and End UserAgreement.

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Customers can access the system moreefficiently and in a simpler way. The customers can have the opportunity to view a better interpretation of results. The customers are also recommended with the purification techniques based on the impurities.
NFR-2	**Security**	All the predicted information is accessed only bythe authenticated users.
NFR-3	**Reliability**	It should be reliable in producing effective and efficient water quality prediction results. It should ensure the trust and belief among people that this water quality prediction system produces correct results when used.
NFR-4	**Performance**	The system should be consistent in producing the prediction results of Water Quality Index (WQI) and also needs to ensure better throughput and response time compared to othersystems.
NFR-5	**Availability**	The system can be utilised by the customers 24/7 and it should be availed to test any kind ofwater samples anywhere.
NFR-6	**Scalability**	It can be used by wide variety of users like testing agencies, private and public laboratories, restaurants and hotels and people who wish to test the quality of water they consume. The system should also be compatible enough so as to be integrated with the future technologies also.

CHAPTER 5

PROJECT DESIGN

5.1 DATAFLOW DIAGRAMS

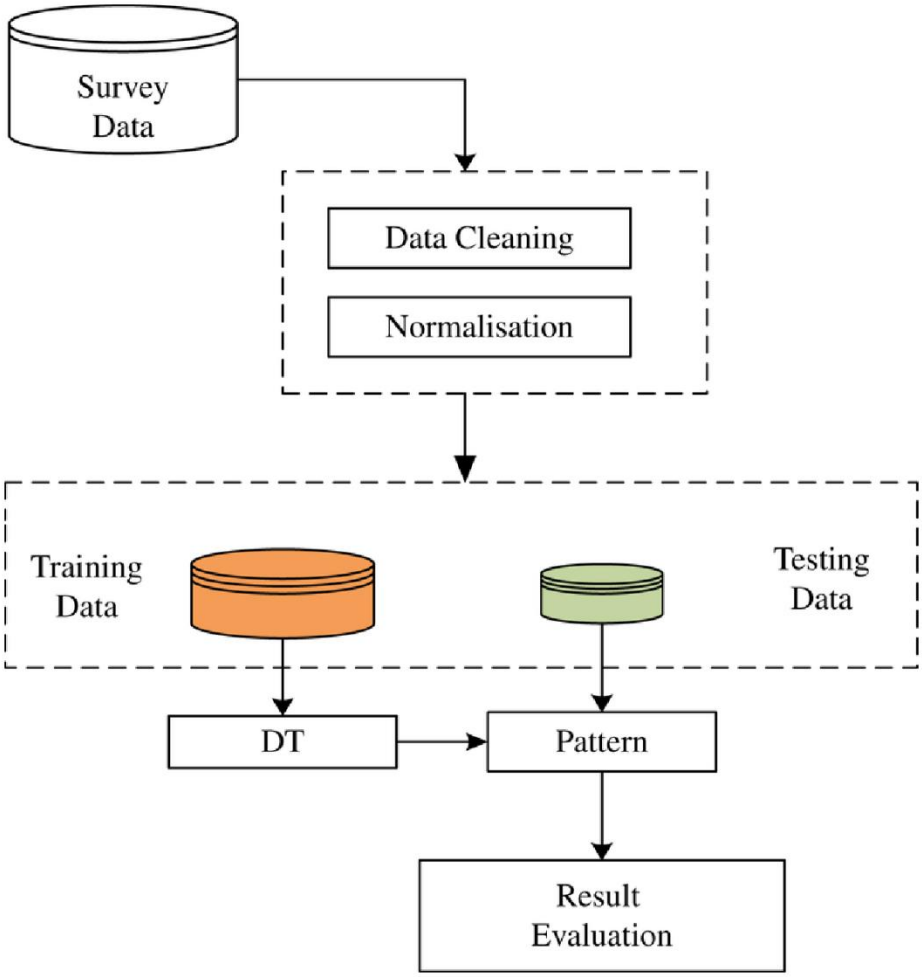


Figure 5.1 - Data flow Diagram

A data flow diagram is a graphical or visual representation using a standardized set of symbols and notations to describe a business's operations through data movement. They are often elements of a formal methodology such as Structured Systems Analysis and Design Method.

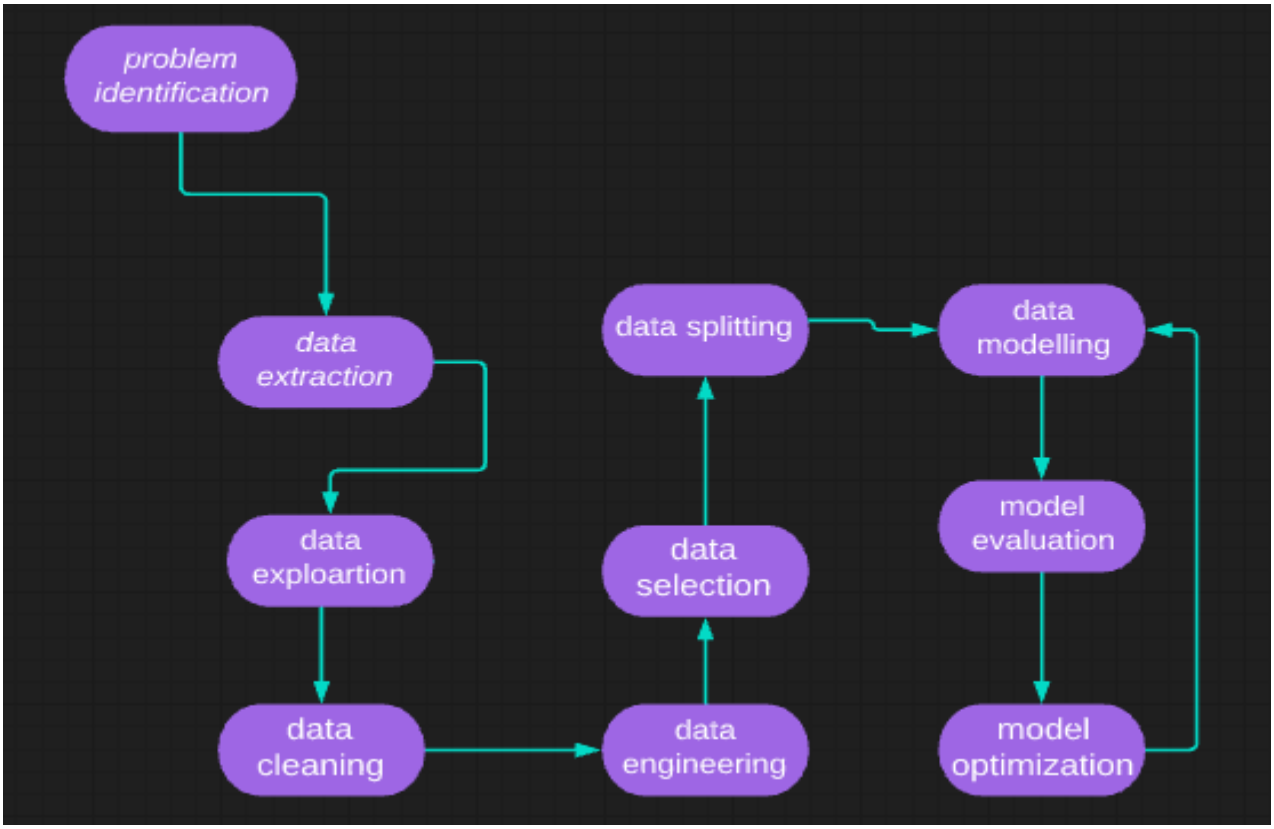


Figure 5.2 - Describes the flow of the project deployment

The flow through which applications, modules, updates, and patches are delivered from developers to users. The methods used by developers to build, test and deploy new code will impact how fast a product can respond to changesin customer preferences or requirements and the quality of each change.

5.2 SOLUTION AND TECHNICAL ARCHITECTURE

Solution Architects are most similar to project managers, ensuring that all parties, includingstakeholders, are on the same page and moving in the right direction at all stages. Technical architects manage all activities leading to the successful implementation of a new application. A solution architect must have a technical background with at least eight years of work experience inone or more IT areas including but not limited to: IT architecture, infrastructure, and cloud development.

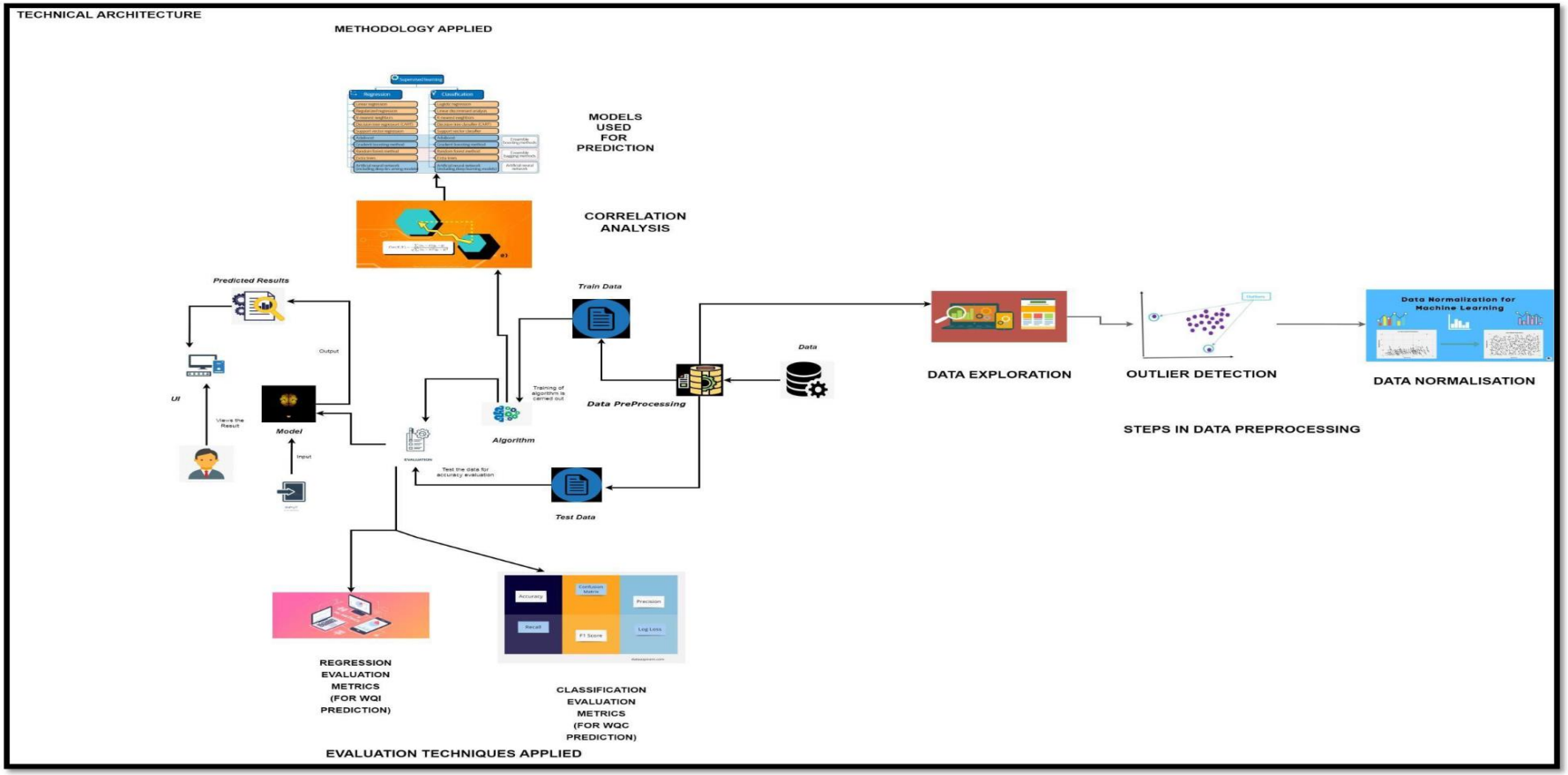


Figure 5.2 - The process of architecttural description

5.3 MODEL PERFORMANCE AND TESTING

S. No	Parameter	Values
1.	Metrics	**Regression Model:** MAE -0.9425563909774494 MSE: 5.63627572932331 RMSE: 2.374084187497004 **R2 score** : Training: 0.9948711603144075 Testing:0.9692766700278257
2.	Tune the Model	Hyper parameter Tuning - NIL Validation Method – Split Sample/ Data Validation.

Table 5.3 Model performance testing

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Collect the appropriate dataset for predicting the water quality.	10	High	Harikrishnan M Dinesh T
Sprint-1		USN-2	Data Preprocessing – Used to transform the data into useful format.	10	High	Harikrishnan M Arvind J Dinesh T
Sprint-2	Model Building	USN-3	Calculate the Water Quality Index (WQI) using Regression algorithm of MachineLearning.	10	High	Dinesh T Ashok N Harikrishnan M
Sprint-2		USN-4	Splitting the Model into Training for Testing from the overall dataset.	7	Medium	Dinesh T Harikrishnan M Ashok N Arvind J
Sprint-3	Training and Testing	USN-5	Train the Model using Regression algorithm and Testing the Performance of the model.	10	High	Harikrishnan M Ashok N Dinesh T
Sprint-4	Implementatio nofthe Application	USN-6	Predict the Water Quality Index (WQI) and recommend the appropriate purificationtechnique.	10	High	Dinesh T Ashok N Harikrishnan M
Sprint-4		USN-7	Deploy the Model on IBM Cloud.	7	Medium	Arvind J Ashok N

Table 6.1 - Sprint planning and estimation

6.1 SPRINT DELIVERY SCHEDULE

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	10	6 Days	24 Oct 2022	29 Oct 2022	8	29 Oct 2022
Sprint-2	10	6 Days	31 Oct 2022	05 Nov 2022	7	05 Nov 2022
Sprint-3	10	6 Days	07 Nov 2022	12 Nov 2022	8	12 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	19 Nov 2022	7	19 Nov 2022

Velocity:

Imagine we have a 6 -day sprint duration, and the velocity of the team is 10 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day).

$$AV = \frac{\text{sprint duration}}{\text{velocity}} = 6/10=0.6$$

Table 6.2 - Sprint Delivery Schedule

6.2 REPORTS FROM JIRA

Burdown Chart

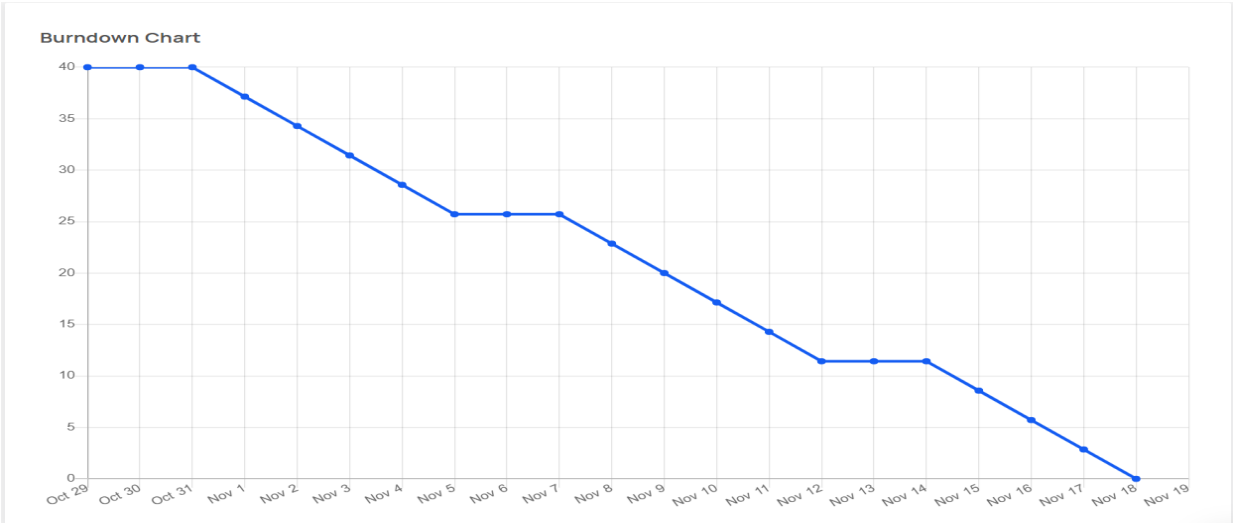


Figure 6.1 - Burndown Chart

A burndown chart shows the amount of work that has been completed in anepic or sprint, and the total work remaining. Burndown charts are used to predict your team's likelihood of completing their work in the time available.

Burnup Chart

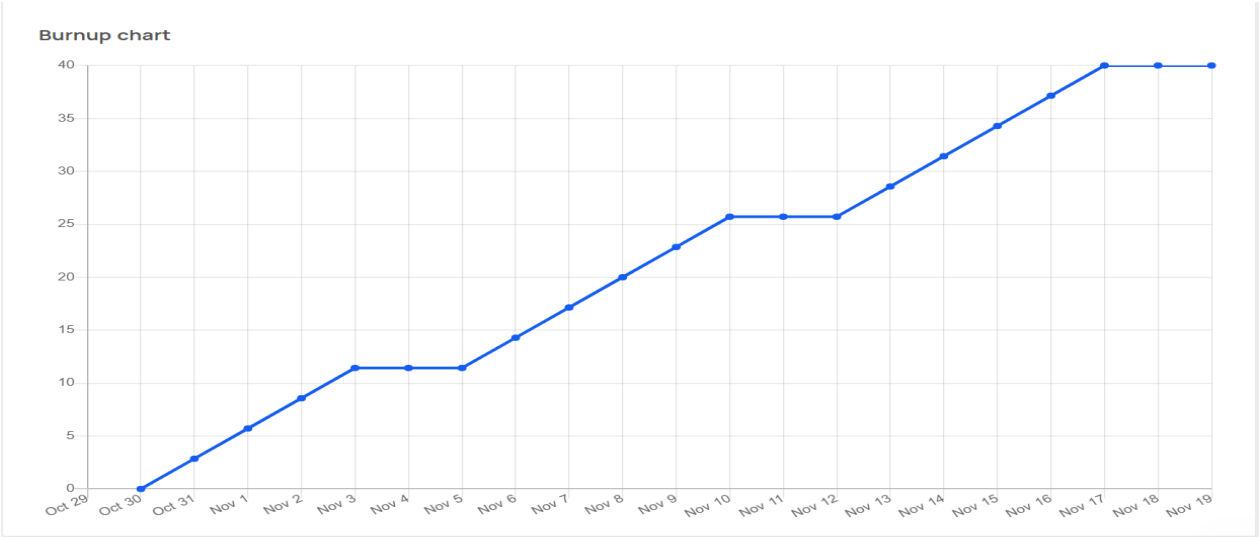


Figure 6.2 - Burnup Chart

A burnup chart highlights the work you've completed against your total project scope while a burn down chart highlights the amount of work remaining in a project. A burnup chart contains a work completed line and a project scope line.It displays the scope of a project and the work completed

CHAPTER 7
CODING AND SOLUTION

7.1 FEATURE 1

The first feature of the deployement is the process of Random Forest Classifier is used to train and test the model for detecting the water quality withthe help of collected and pre-processed dataset collections.

```
# Train Test Split:
from sklearn.model_selection import train_test_split
 X_train,X_test,y_train,y_test=train_test_split(X_smote,y_smote,
test_size=0.3, random_state=33)
```

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average incase of regression.

```
# RandomForestClassifier:
from  sklearn.ensemble
import RandomForestClassifier
RandomForest = RandomForestClassifier()
 RandomForest  =  RandomForest.fit(X_train,y_train)
# Predictions:
y_pred = RandomForest.predict(X_test)

# Performance:
print('Accuracy:', accuracy_score(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually used when doing gradient boosting.

```
# GradientBoostingClassifier:
from sklearn.ensemble import GradientBoostingClassifier
GradientBoost = GradientBoostingClassifier() GradientBoost =
GradientBoost.fit(X_train,y_train)

# Predictions:
y_pred = GradientBoost.predict(X_test)

# Performance:
print('Accuracy:', accuracy_score(y_test,y_pred))
print(confusion_matrix(y_test,y_pred)) print(classification_report(y_test,y_pred))
```

7.2 FEATURE 2

Python flask is the first feature that helps to complete this project. Itallows the user to create local server and host the website in a local machine.

```
from flask import Flask, render_template, request
import numpy as np
import pickle
```

Here we import all the necessary features of this project involving in Pythonflask.

```
import numpy as np
from flask import Flask,render_template,request
import pickle

app = Flask(__name__)
model = pickle.load(open('wqi.pkl','rb'))
@app.route('/about')
def about():
    return render_template("about.html")
@app.route('/')
```

```
def home() :
    return render_template("predict.html")
@app.route('/bod')
def bod():
    return render_template("bod.html")
@app.route('/col')
def col():

    return render_template("col.html")
@app.route('/do')
def do():
    return render_template("do.html")
@app.route('/nit')
def nit():
    return render_template("nit.html")
@app.route('/ph')
def ph():
    return render_template("ph.html")
@app.route('/cond')
def cond():
    return render_template("cond.html")
@app.route('/analysis')
def analysis():
    return render_template("analysis.html")
@app.route('/login',methods = ['POST'])
def login() :
    year = request.form["year"]
    do = request.form["do"]
    ph = request.form["ph"]
    co = request.form["co"]
    bod = request.form["bod"]
    na = request.form["na"]
    tc = request.form["tc"]
    total = [[int(year),float(do),float(ph),float(co),float(bod),float(na),float(tc)]]
```

```
    y_pred = model.predict(total)
    print(y_pred)
    y_pred =y_pred[[0]]
    if(y_pred >= 95 and y_pred <= 100) :
        return render_template("predict.html",showcase = 'Excellent,The predicted value is '+
str(y_pred)+' No Purification or Treatment of Water is needed.')
    elif(y_pred >= 89 and y_pred <= 94) :
        return render_template("predict.html",showcase = 'Very good,The predicted value is
'+str(y_pred)+' Minor Purification or Treatment of Water is needed.')
    elif(y_pred >= 80 and y_pred <= 88) :
        return render_template("predict.html",showcase = 'Good,The predicted value
is'+str(y_pred)+' Conventional Purification or Treatment of Water is needed.')
    elif(y_pred >= 65 and y_pred <= 79) :
        return render_template("predict.html",showcase = 'Fair,The predicted value is
'+str(y_pred)+' Extensive Purification or Treatment of Water is needed.')
    elif(y_pred >= 45 and y_pred <= 64) :
        return render_template("predict.html",showcase = 'Marginal,The predicted value is
'+str(y_pred)+' Doubtful in purifying and treating the water so as to get Pure Water.')
    else :
        return render_template("predict.html",showcase = 'Poor,The predicted value is
'+str(y_pred)+' The Water is not fit for to be used for Drinking.')


    if__name__== '__main__' :
    app.run()
```

Here we use the inputs from the html pages which has to be get by using request method in Python Flask. By validating the values from the database, weallow the user to access the home page. render_template: Used for rendering html pages on browser. url_for: Passing the control of the program to another function. session: Creates a separate session for the individual use.

```
    <html>
<title>water quality prediction</title>
<style>
*{
        margin:0;
        padding:0;
        text-decoration:none;
        font-family:montserrat;
}
.pd{
padding-bottom:100%;}
body
{
background-image:url('https://tenor.com/view/fallwater-gif-8021595.gif');
background-position: center;
font-family:sans-serif;
background-size:cover;
margin-top:40px;
}

.main{
        background-color:rgb(0,0,0,0.6);
        width:800px;
        height:500px;
        margin:auto;
        position:center;
        border-top-left-radius:100px;
        border-bottom-right-radius:100px;

}
```

```
.main input[type="text"],.main input[type="text"],.main input[type="text"],.main input[type="text"],.main input[type="text"],.main input[type="text"]{
	border:0;
	background:none;
	display:block;
	margin:20px auto;
	text-align:center;
	border:2px solid #3498db;
	padding:10px 3px;
	width:200px;
	outline:none;
	color:white;
	border-radius:24px;
	transition:0.25s;

}
.main input[type="text"]:focus,.main input[type="text"]:focus,.main input[type="text"]:focus,.main input[type="text"]:focus,.main input[type="text"]:focus,.main input[type="text"]:focus{
	width:280px;
	border-color:#8e44ad;
}
.logbtn{
	display:block;
	width:35%;
	height:50px;
	border:none;
	border-radius:24px;
	background:linear-gradient(120deg,#3498db,#8e44ad,#3498db);
	background-size:200%;
	color:#fff;
```

```
        outline:none;
        cursor:pointer;
        transition:.5s;
}
.logbtn:hover{
        background-position:right;
}


.predict{
  color:black;
  text-align:center;
}
</style>



<body>
<center><img
src="https://i.pinimg.com/236x/4b/bb/7d/4bbb7dae1470c8cdc3821b7c5b2f10fb.jpg"
height="100"><b class="pd"><font color="white" size="15" font-family="Comic Sans MS"
>Water Quality Prediction</font></b></center>
        <a href="/about" target="_blank"><font color="white" size="12" font-family="Comic
Sans MS" >Why To Find Water Quality</font></a>
        <form action="y_predict" class="main" method="post">
        <br>
        <center><input type="text" name="Station" placeholder="Station" required="required"
/>
            <input type="text" name="Dissolved Oxygen" placeholder="Dissolved Oxygen"
required="required" />
            <input type="text" name="PH" placeholder="PH" required="required" />
```

```
            <input type="text" name="Carbon Monoxide" placeholder="Carbon Monoxide"
required="required" />
          <input type="text" name="Biochemical Oxygen Demand" placeholder="Biochemical
oxygen Demand" required="required" />
        <input type="text" name="Sodium" placeholder="Sodium" required="required" />
                  <input type="text" name="Technetium" placeholder="Technetium"
required="required" />
        <button type="submit" class="logbtn">Predict</button></center>

    </form>



          <br>
          <br>
           <h1 class=predict> {{ prediction_text }}</h1>


</body>
</html>
```

CHAPTER 8

TESTING

8.1 TEST CASES

S.No.	PREREQUISITES	TEST DATA
1	Access to Chrome Browser	By clicking the website link
2	Entering the details required	Details should be in a integer format
3	Check for correct values	Data sholud be filled
4	Application to train the model	Provide the datasets for model training

Table 8.1 - Test Details

Test Scenario :

Step	Step Details	Expected Results	Actual Results	Pass/Fail/Not/ Executed/ Suspended
1	Navigate to website link	Site should open	As Expected	pass
2	Enter the details	Details should be entered	As Expected	pass
3	Click Submit	Check the result	As Expected	Pass
4	Output results	Result are generated	As Expected	Pass

Table 8.2 - Test Cases

ER ACCEPTANCE TESTING

Figure 8.1 - User Acceptance Test 1

In this the user will be entering tested values which is moved for prediction.

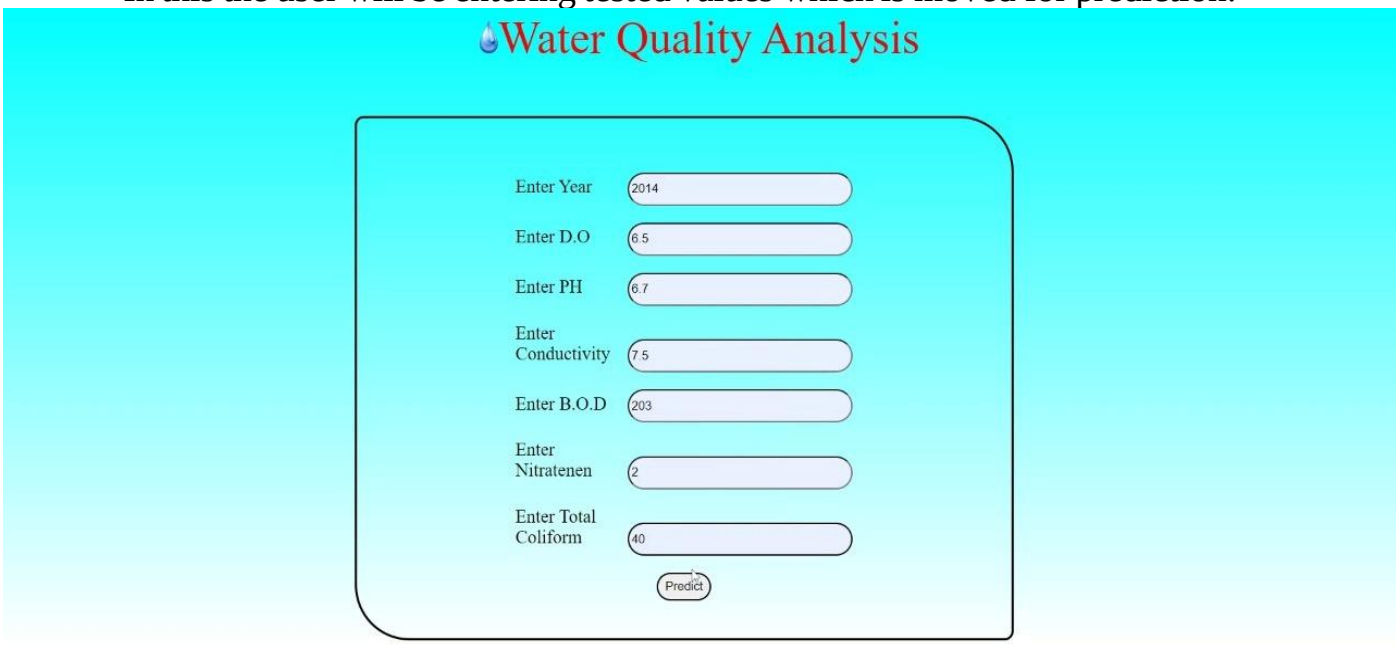


Figure 8.2 - User Acceptance Test 2

In this the data which was entered by the user will be analyzed

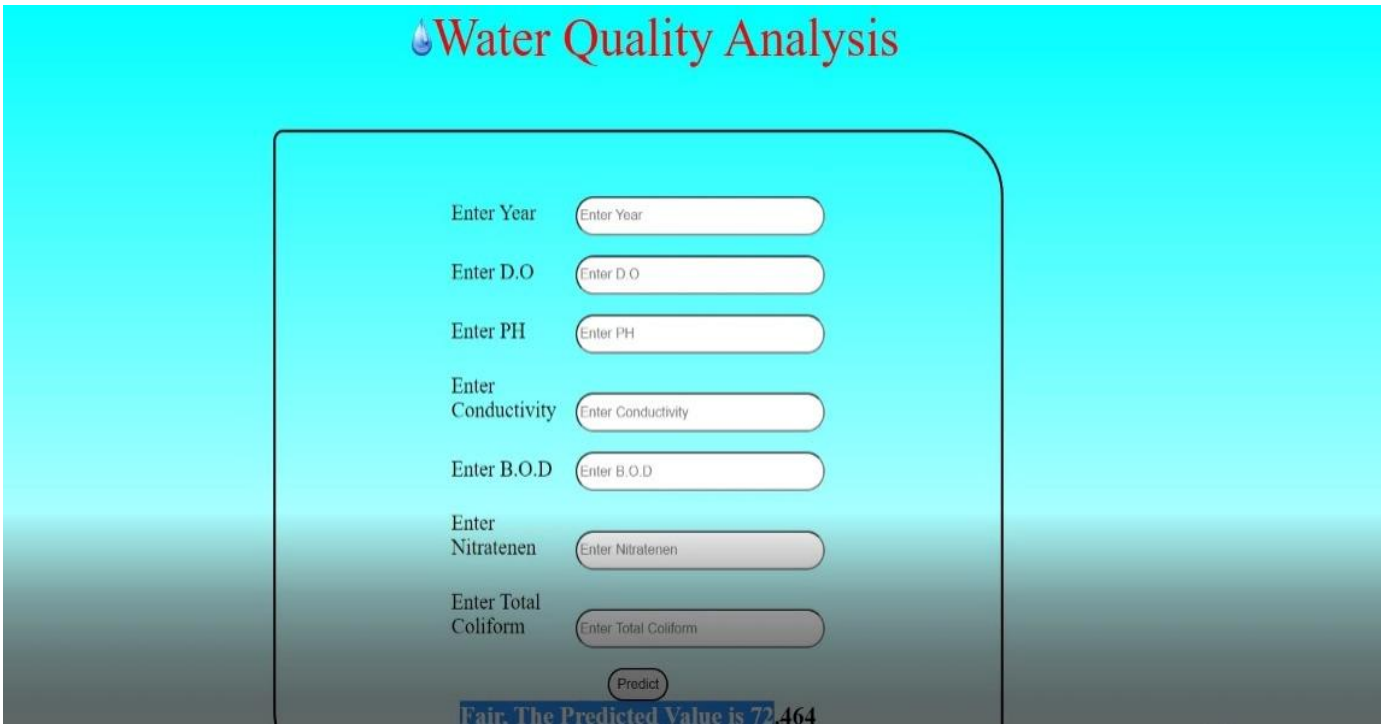


Figure 8.3 - User Acceptance Test 3

In this the data which was entered by the user will be analyzed

CHAPTER 9

RESULTS

9.1 PERFORMANCE METRICS

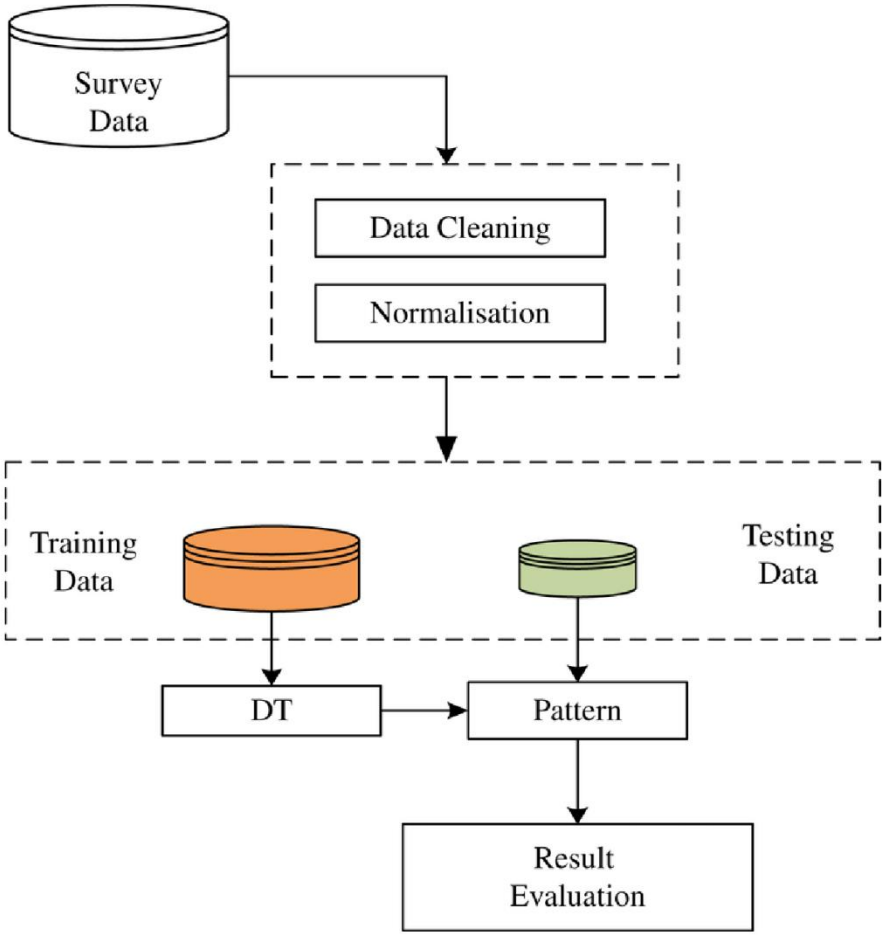


Figure 9.1 - Flow of the Performance

Performance Metrics is performed along for all the above assigned tasks. In each and every tasks various metrics is performed inorder to provide the optimum outcome.Results gives the best outcome as expected in project.

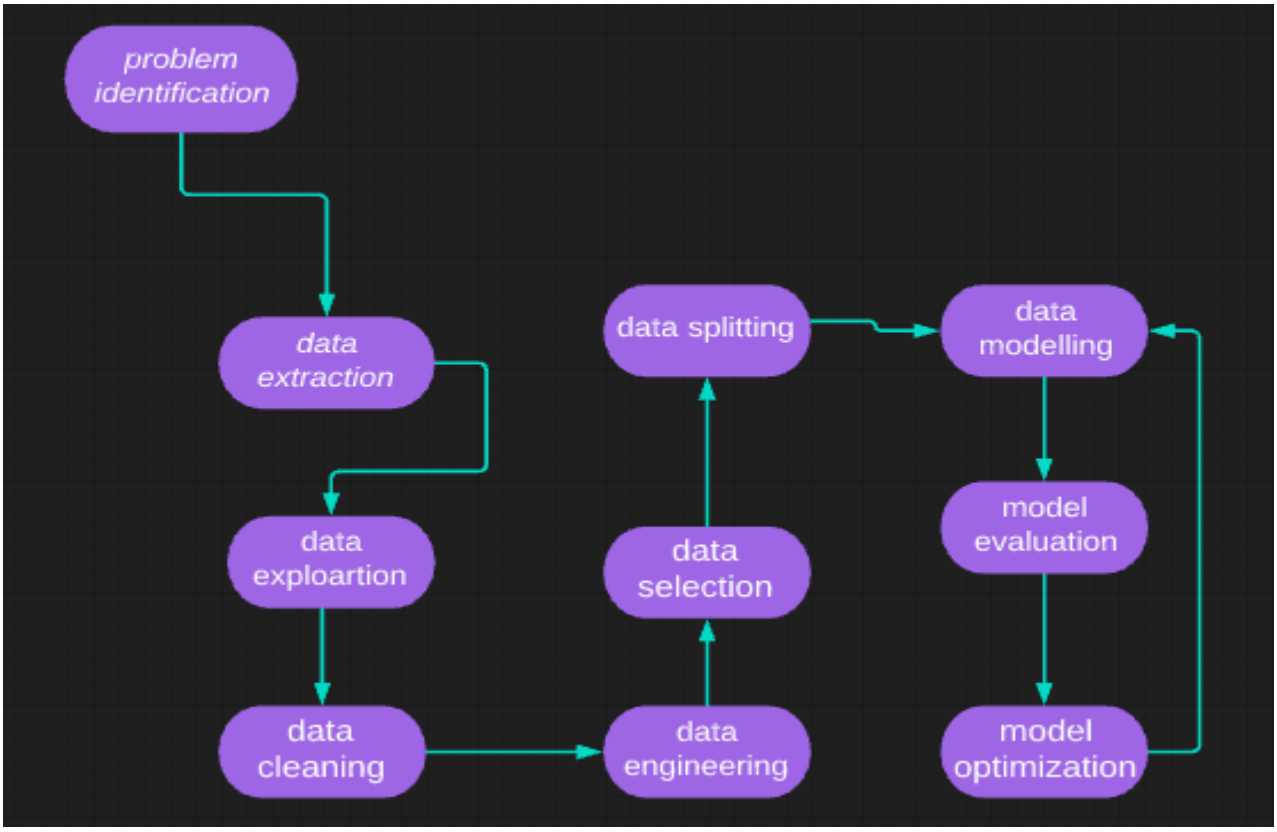


Figure 9.2 - Performance Matrix

CHAPTER 10

ADVANTAGES AND DISADVANTAGES

ADVANTAGES

Predicting the quality of drinkable water may reduce the incidence of water-related diseases.

The latest machine learning approach has shown promising predictive accuracy for water quality.

The evaluation and prediction of source water quality based on machine learning can assist in the early warning and control of pollution.

CHAPTER 11

CONCLUSION

The performance of machine learning techniques such as RF, NN, MLR, SVM, and BTM to predict the water quality components of an Indian water quality dataset was evaluated in this work. The most well-known dataset variables, such as BOD, DO, TC, Nitrate, pH, and Temp, were obtained for this purpose. The findings revealed that the applied models performed well in forecasting water quality parameters; however, the greatest performance was linked with the MLR with Accuracy Upper. Further research will be done to build models that combine the proposed method with other techniques and deep learning approaches to improve the efficacy of the selection process.

CHAPTER 12

FUTURE SCOPE

The study provides a frugal technique for eliminating water pollution in residential overhead tanks. They are used toassess water quality, and machine learning algorithms are used to forecast potential water pollution problems. To gather water parameters, the proposed system comprises multiple sensors interfaced with Node McU. The user is notified before the water turns contaminated. The implemented solution protects the water from pollution while still being cost efficient. The projects future scope includes detecting illnesses caused by numerous factors and devising the most effective plan to clean the tank.

CHAPTER 13

APPENDIX

SOURCE CODE:

Algorithm :

```
#Importing The Libraries
 import pandas as pd
 import numpy as np
 import matplotlib.pyplot as plt
 import seaborn as sns


#Reading the Dataset data=pd.read_csv(path)

#Exploratory Data Analysis
data.head()
data.info()
data.describe()

#Checking For Null Values And Handling Null Values
 data.isnull().any()
 data.isnull().sum()
data.isnull().sum()
 #Data Visualization
```

1. Univariate Analysis

```
for col in df.columns:
if df.dtypes[col]=='float64'or df.dtypes[col]=='int64':
sns.boxplot(x=df[col]).set(xlabel=col)
plt.show()
```

2. Bivariate Analysis

```
for col1 in df.columns:
 if df.dtypes[col1]=='float64':
 for col2 in ['wqi']:
  if df.dtypes[col2]=='float64':
    sns.boxplot(x=df[col1],y=df[col2]).set(xlabel=[col1],ylabel=[col2])
    plt.show()

   #Splitting The Dataset Into Dependent And Independent Variable
    x=data.iloc[:,0:-1]
    y=data.iloc[:,-1]

    #Split The Dependent And Independent Features Into Train Set And TestSetfrom
    sklearn.model_selction import train_test_split
    xtrain.xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2)

    #Check the shape of both xtrain and xtest.
    xtrain.shape
    xtest.shape
    #importing the machine learning model
    from sklearn.svm import svc
    from sklearn.ensemble import RandomForestClassifierfrom
    sklearn.neighbors import KNeighborsClassifier #initiailizing
    the machine learning models
    svm=SVC()
    RFmodel=RandomForestClassifier()
    KNNmodel=KNeighborsClassifiers()
    svm=SVC()
    #Train the data with SVM model
    svm.fit(xtrain,ytrain)
```

Github and Project Video Demo Link

Github Link:

https://github.com/IBM-EPBL/IBM-Project-46027-1660735022

Project Video Demo Link:

https://drive.google.com/drive/folders/1ZrJZmTvWnFwHkj0iOZusnm6DFsQpAgzZ?usp=share_link