A

Project Report on

PREDICTING URBAN WATER QUALITY WITH UBIQUITOUS DATA

Submitted for partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

B. RAJESH	**20K81A0508**
G. SAI KUMAR	**20K81A0520**
R. DEEKSHITH GOUD	**20K81A0549**
A. SOUMITH REDDY	**20K81A0503**

Under the Guidance of

P. AKHIL

ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

St. MARTIN'S ENGINEERING COLLEGE

UGC Autonomous

Affiliated to JNTUH, Approved by AICTE,
Accredited by NBA & NAAC A+, ISO 9001:2008 Certified
Dhulapally, Secunderabad - 500 100
www.smec.ac.in

MARCH - 2024

St. MARTIN'S ENGINEERING COLLEGE

UGC Autonomous
NBA& NAAC A+ Accredited
Dhulapally, Secunderabad - 500 100
www.smec.ac.in

Certificate

This is to certify that the project entitled **"Predicting Urban Water Quality With Ubiquitous Data"** is being submitted by **B. RAJESH (20K81A0508), G. SAI KUMAR (20K81A0520), A. SOUMITH REDDY (20K81A0503), R. DEEKSHITH GOUD (20K81A0549)**, in fulfilment of the requirement for the award of degree of **BACHELOROF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING** is recorded of bonafide work carried out by them. The result embodied in this report have been verified and found satisfactory.

Guide
Mr. P. AKHIL
Assistant Professor
Department of CSE

Head of the Department
Dr. R. SANTHOSHKUMAR
Associate Professor & Head
Department of CSE

Internal Examiner **External Examiner**

Date:

Place:

St. MARTIN'S ENGINEERING COLLEGE

UGC Autonomous
NBA& NAAC A+ Accredited
Dhulapally, Secunderabad - 500 100
www.smec.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, the students of '**Bachelor of Technology in Department of Computer Science and Engineering**', session: 2020 - 2024**, St. Martin's Engineering College, Dhulapally, Kompally, Secunderabad,** hereby declare that the work presented in this Project Work entitled "**Predicting Urban Water Quality With Ubiquitous Data"** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. This result embodied in this project report has not been submitted in any university for award of any degree.

B. Rajesh	**20K81A0508**
G. Sai Kumar	**20K81A0520**
A. Soumith Reddy	**20K81A0503**
R. Deekshith Goud	**20K81A0549**

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowded our efforts with success.

We extend our deep sense of gratitude to Group Director **Dr. P. SANTOSH KUMAR PATRA**, St. Martin‟s Engineering College Dhulapally, for permitting us to undertake this project.

We wish to record our profound gratitude to **Dr. M. SREENIVAS RAO**, Principal, St. Martin‟s Engineering College, for his motivation and encouragement.

We are also thankful to **Dr. R. SANTHOSHKUMAR**, Head of the Department, Department of Computer Science and Engineering, St. Martin's Engineering College, Dhulapally, Secunderabad. For his support and guidance throughout our project as well as Project Coordinators **Dr. G. JAWAHERLALNEHRU,** Associate Professor and **Mr. S. BAVANKUMAR**, Assistant Professor, Computer Science and Engineering department for their valuable support.

We would like to express our sincere gratitude and indebtedness to our projectsupervisor **Mr. P. AKHIL**, Assistant Professor, Department of Computer Science and Engineering, St. Martins Engineering College, Dhulapally, for his support and guidance throughout our project.

Finally, we express thanks to all those who have helped us successfully completing this project. Furthermore, we would like to thank our family and friends for their moral support and encouragement. We express thanks to all those who have helped us in successfully completing the project.

B. Rajesh **20K81A0508**
G. Sai Kumar **20K81A0520**
A. Soumith Reddy **20K81A0503**
R. Deekshith Goud **20K81A0549**

CONTENTS PAGE NO

CERTIFICATE II

DECLARATION III

ACKNOWLEDGEMENT IV

ABSTRACT VII

LIST OF FIGURES VIII

LIST OF TABLES X

LIST OF ACRONYMS AND DEFINITIONS XI

CHAPTER 1 INTRODUCTION 01

CHAPTER 2 LITERATURE SURVEY 04

CHAPTER 3 SYSTEM ANALYSIS AND DESIGN 07

3.1 Existing Systems 07

3.1.1 Disadvantages 08

3.2 Proposed Systems 08

3.2.1 Advantages 09

3.3 Project Architecture 09

3.4 Algorithms 10

3.5 UML Diagrams 14

3.5.1 Use Case Diagram 15

3.5.2 Class Diagram 16

3.5.3 Sequence Diagram 17

3.5.4 Data Flow Diagram 17

3.5.5 Package Diagram 19

3.5.6 Profile Diagram 19

CHAPTER 4 MODULES 20

4.1 Data Collection Module 20

4.2 Feature Engineering Module 20

4.3 Machine Learning Module 20

4.4 Real Time Data Processing Module 21

CHAPTER 5 TECHNOLOGY DESCRIPTION 22

CHAPTER 6 EXPERIMENTAL RESULTS 35

CHAPTER 7 SYSTEM TESTING **41**

CHAPTER 8 CODE IMPLEMENTATION **45**

CHAPTER 9 CONCLUSION AND FUTURE ENHANCEMENT **48**

REFERENCES **49**

ABSTRACT

Urban water quality is of great importance to our daily lives. Prediction of urban water quality help control water pollution and protect human health. However, predicting the urban water quality is a challenging task since the water quality varies in urban spaces non-linearly and depends on multiple factors, such as meteorology, water usage patterns, and land uses. In this work, we forecast the water quality of a station over the next few hours from a data-driven perspective, using the water quality data and water hydraulic data reported by existing monitor stations and a variety of data sources we observed in the city, such as meteorology, pipe networks, structure of road networks, and point of interests (POIs). First, we identify the influential factors that affect the urban water quality via extensive experiments. Second, we present a multi-task multi-view learning method to fuse those multiple datasets from different domains into an unified learning model. We evaluate our method with real-world datasets, and the extensive experiments verify the advantages of our method over other baselines and demonstrate the effectiveness of our approach.

LIST OF FIGURES

Figure No.	Figure Title	Page No.
3.1	Project Architecture	10
3.2	Use Case Diagram	15
3.3	Class Diagram	16
3.4	Sequence Diagram	17
3.5	Data Flow Diagram	18
3.6	Package Diagram	19
3.7	Profile Diagram	19
5.1	official python website	25
5.2	Latest version of python	26
5.3	Release version	26
5.4	Files	27
5.5	Applications	27
5.6	Installation	28
5.7	Installation successful	28
5.8	Command Prompt	29
5.9	Save the file	30
5.10	Program	30
6.1	User login	35
6.2	Login service provider	35
6.3	Registration Details	36
6.4	Viewing Profile	36
6.5	Predicting water quality type	37
6.6	View all remote users	37
6.7	Water quality data sets trained and tested results	38

6.8	Bar chart	38
6.9	Line chart	39
6.10	Pie chart	39
6.11	Water quality type found ratio details	40

LIST OF TABLES

Table No.	Table Name	Page No.
7.1	Test Cases	43

CHAPTER 1
INTRODUCTION

Urban water is a vital resource that affects various aspects of human, health and urban lives. People living in major cities are increasingly concerned about the urban water quality, calling for technology that can monitor and predict the water quality in real time throughout the city. Urban water quality, which serves as "a powerful environmental determinant" and "a foundation for the prevention and control of waterborne diseases" [1], refers to the physical, chemical and biological characteristics of a water body, and several chemical indexes (such as residual chlorine, turbidity and pH) can be used as effective measurements for the water quality in current urban water distribution systems [2].

With the increasing demand for water quality information, several water quality monitoring stations have been deployed throughout the city's water distribution system to provide the real-time water quality reports in a city. Figure 1 illustrates the water quality monitor stations that have been deployed in Shenzhen, China. Besides water quality monitoring, predicting the urban water quality plays an essential role in many urban aquatic projects, such as informing waterworks' decision making (e.g., pre-adjustment of chlorine from the waterworks), affecting governments' policy making (e.g., issuing pollution alerts or performing a pollution control), and providing maintenance suggestions (e.g., suggestions for replacements of certain pipelines).

Predicting urban water quality, however, is very challenging due to the following reasons.

First, urban water quality varies by locations non-linearly and depends on multiple factors, such as meteorology, water usage patterns, land use, and urban structures. As depicted in Figure 1, the water quality indexes (RC) reported by the three stations demonstrate different patterns. Exiting hydraulic model-based approaches try to model water quality from physical and chemical perspective, but such hydraulic model can hardly capture all of those complex factors. Moreover, the parameters I model are hard to get, which make it difficult to extend to other water distribution systems. Second, as all the stations are connected through the pipeline system, the water quality among different stations are mutually correlated by several complex factors, such as attributes in pipe networks and distribution of POIs. Traditional hydraulic model-based approaches build hydraulic model for each station and ignore their spatial correlations, and thus their performance is far from satisfactory. Hence, besides identifying the influential factors, how to efficiently characterize and incorporate such relatedness poses another challenge.

Fortunately, in the era of big data [3] [4] [5], unprecedented data in urban areas (e.g., meteorology, POIs, and road networks) can provide complementary information to help predict the urban water quality. For example, temperature can be an indicator of water quality, with higher temperature indicating better water quality. The possible reason is that the water consumption tends to grow when temperature is high since most people may choose to take a shower, and the increased water consumption is one major cause that prevents the water quality's deterioration in the distribution systems.

To benefit from the unprecedented data in urban areas, in this paper, we predict the water quality of a station through a data-driven perspective using a variety of data sets, including water quality data, hydraulic data, meteorology data, pipe networks data, road networks data, and POIs. First, we perform extensive experiments and data analytics between the water quality and multiple potential factors, and identify the most influential ones that have an effect on the urban water quality. Second, we present a novel spatio-temporal multi-task multi-view learning (stMTMV) framework to fuse the heterogeneous data from multiple domains and jointly capture each station's local information as well as their global information into an unified learning model [6].

1.2 Problem Statement:

Ensuring the quality of urban water sources is a critical aspect of maintaining public health and environmental sustainability. The increasing urbanization and expansion of cities bring about challenges in effectively monitoring and predicting urban water quality. Traditional water quality monitoring systems often rely on sporadic and discrete sampling, leading to delayed detection of pollutants and inadequate understanding of dynamic changes in water quality parameters. The need for real-time, continuous monitoring and prediction of urban water quality remains largely unmet.

Ubiquitous data sources, including sensor networks, Internet of Things (IoT) devices, and various digital platforms, offer an opportunity to collect extensive and real-time data on various aspects of urban water systems. However, integrating and leveraging this ubiquitous data for accurate and timely prediction of water quality is a complex problem. Challenges include data heterogeneity, reliability, and the development of robust data-driven models capable of capturing the intricate relationships between diverse environmental factors and water quality parameters.

Therefore, the existing problem lies in the inefficiency of current approaches to predict urban water quality, particularly the reliance on infrequent sampling and the lack of comprehensive data-driven models that harness the potential of ubiquitous data sources. Addressing this challenge is essential for proactively managing and maintaining the quality of urban water sources, ensuring the well-being of urban populations and sustainable water resource management.

1.3 Objectives:

1. Data Integration and Standardization:
Develop methods for integrating heterogeneous data from ubiquitous sources, including sensor networks, IoT devices, and digital platforms, ensuring standardized formats for comprehensive analysis.

2. Feature Selection and Dimensionality Reduction:
Implement techniques for effective feature selection and dimensionality reduction to identify and prioritize relevant parameters influencing urban water quality, reducing computational complexity and enhancing model interpretability.

3. Data Quality Assurance:Establish protocols for quality assurance of ubiquitous data, addressing issues such
as accuracy, reliability, and calibration of sensors to ensure the reliability and trustworthiness of the input data used for prediction.

4. Real-time Data Acquisition and Processing:
Develop mechanisms for real-time acquisition, processing, and integration of ubiquitous data, enabling timely responses to dynamic changes in urban water quality and facilitating proactive management strategies.

5. Machine Learning Model Development:
Design and implement data-driven machine learning models capable of predicting urban water quality parameters based on the integrated and processed ubiquitous data, considering the complex relationships between environmental factors and water quality.

CHAPTER 2
LITERATURE OVERVIEW

“Modeling chlorine residuals in drinking-water distribution systems,”

A mass transfer-based model is developed for predicting chlorine decay in drinking-water distribution networks. The model considers first-order reactions of chlorine to occur both in the bulk flow and at the pipe wall. The overall rate of the wall reaction is a function of the rate of mass transfer of chlorine to the wall and is therefore dependent on pipe geometry and flow regime. The model can thus explain field observations that show higher chlorine decay rates associated with smaller pipe sizes and higher flow velocities. It has been incorporated into a computer program called EPANET that can perform dynamic water-quality simulations on complex pipe networks. The model is applied to chlorine measurements taken at nine locations over 53 h from a portion of the South Central Connecticut Regional Water Authority's service area. Good agreement with observed chlorine levels is obtained at locations where the hydraulics are well characterized. The model should prove to be a valuable tool for managing chlorine-disinfection practices in drinking-water distribution systems.

“Methodologies for cross-domain data fusion: An overview,”

Traditional data mining usually deals with data from a single domain. In the big data era, we face a diversity of datasets from different sources in different domains. These datasets consist of multiple modalities, each of which has a different representation, distribution, scale, and density. How to unlock the power of knowledge from multiple disparate (but potentially connected) datasets is paramount in big data research, essentially distinguishing big data from traditional data mining tasks. This calls for advanced techniques that can fuse knowledge from various datasets organically in a machine learning and data mining task. This paper summarizes the data fusion methodologies, classifying them into three categories: stage-based, feature level-based, and semantic meaning-based data fusion methods. The last category of data fusion methods is further divided into four groups: multi-view learning-based, similarity-based, probabilistic dependency-based, and transfer learning-based methods. These methods focus on knowledge fusion rather than schema mapping and data merging, significantly distinguishing between cross-domain data fusion and traditional data fusion studied in the database community. This paper does not only introduce high-level principles of each category of methods, but also give

examples in which these techniques are used to handle real big data problems. In addition, this paper positions existing works in a framework, exploring the relationship and difference between different data fusion methods. This paper will help a wide range of communities find a solution for data fusion in big data projects.

"Urban computing: Concepts, methodologies, and applications,"

Urbanization's rapid progress has modernized many people's lives but also engendered big issues, such as traffic congestion, energy consumption, and pollution. Urban computing aims to tackle these issues by using the data that has been generated in cities (e.g., traffic flow, human mobility, and geographical data). Urban computing connects urban sensing, data management, data analytics, and service providing into a recurrent process for an unobtrusive and continuous improvement of people's lives, city operation systems, and the environment. Urban computing is an interdisciplinary field where computer sciences meet conventional city-related fields, like transportation, civil engineering, environment, economy, ecology, and sociology in the context of urban spaces. This article first introduces the concept of urban computing, discussing its general framework and key challenges from the perspective of computer sciences. Second, we classify the applications of urban computing into seven categories, consisting of urban planning, transportation, the environment, energy, social, economy, and public safety and security, presenting representative scenarios in each category. Third, we summarize the typical technologies that are needed in urban computing into four folds, which are about urban sensing, urban data management, knowledge fusion across heterogeneous data, and urban data visualization. Finally, we give an outlook on the future of urban computing, suggesting a few research topics that are somehow missing in the community.

"Detecting collective anomalies from multiple spatio- temporal datasets across different domains,"

The collective anomaly denotes a collection of nearby locations that are anomalous during a few consecutive time intervals in terms of phenomena collectively witnessed by multiple datasets. The collective anomalies suggest there are underlying problems that may not be identified based on a single data source or in a single location. It also associates individual locations and time intervals, formulating a panoramic view of an event. To detect a collective anomaly is very

challenging, however, as different datasets have different densities, distributions, and scales. Additionally, to find the spatio-temporal scope of a collective anomaly is time consuming as there are many ways to combine regions and time slots. Our method consists of three components: Multiple-Source Latent-Topic (MSLT) model, Spatio-Temporal Likelihood Ratio Test (ST_LRT) model, and a candidate generation algorithm. MSLT combines multiple datasets to infer the latent functions of a geographic region in the framework of a topic model. In turn, a region's latent functions help estimate the underlying distribution of a sparse dataset generated in the region. ST_LRT learns a proper underlying distribution for different datasets, and calculates an anomalous degree for each dataset based on a likelihood ratio test (LRT). It then aggregates the anomalous degrees of different datasets, using a skyline detection algorithm. We evaluate our method using five datasets related to New York City (NYC): 311 complaints, taxicab data, bike rental data, points of interest, and road network data, finding the anomalies that cannot be identified (or earlier than those detected) by a single dataset. Results show the advantages beyond six baseline methods.

"Urban water quality prediction based on multi-task multi-view learning,"

Urban water quality is of great importance to our daily lives. Prediction of urban water quality help control water pollution and protect human health. In this work, we forecast the water quality of a station over the next few hours, using a multitask multi-view learning method to fuse multiple datasets from different domains. In particular, our learning model comprises two alignments. The first alignment is the spaio-temporal view alignment, which combines local spatial and temporal information of each station. The second alignment is the prediction alignment among stations, which captures their spatial correlations and performs copredictions by incorporating these correlations. Extensive experiments on real-world datasets demonstrate the effectiveness of our approach.

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

3.1 EXISTING SYSTEM

Several studies in the environmental science have been tried to analyze the water quality problems via data-driven based approaches, and those studies covers a range of topics, from the physical process analysis in the river basin, to the analysis of concurrent input and output time series. The approaches adopted in these studies include instance-based learning models (e.g., kNN) as well as neural network models (e.g., ANN). In general, those data-driven approaches in the environmental science can fall into the following three major categories: Instance-based Learning models (IBL), Artificial Neural Network models (ANN) and Support Vector Machine models (SVM).

Instance-based learning models (IBL) is a family of learning algorithms that model a decision problem with instances or examples of training data that are deemed important to test model . As a typical example of IBL, k-Nearest Neighbors
(k-NN) is widely used due to its simplicity and incredibly good performance in practice.

For example, the work introduced by Karlsson et al. addressed the classical rainfall-runoff forecasting problem by k-NN algorithm, and demonstrated promising results. Toth et al. used k-NN to predict the rainfall depths from the history data, and showed the persistent outperformance of k-NN over other time series prediction methods.

As another example, Ostfeld et al. developed a hybrid genetic k-Nearest Neighbor algorithm to calibrate the two-dimensional surface quantity and water quality model. Artificial Neural Network (ANN) is a network inspired by biological neural networks (in particular the human brain), which consists of multiple layers of nodes (neurons) in a directed graph with each ayer fully connected to the next one. Neural networks have been widely employed to solve a wide variety of tasks, and can acheive good results. For instance, Moradkhani et al. proposed an hourly streamflow forecasting method based on a radial-basis function (RBF) network and demonstrated its advantages over other numerical prediction methods. Also, the work introduced by Kalin predicted the water quality indexes in watersheds through ANN. Support

Vector Machines (SVMs) are typical supervised learning models that analyze data used for classification and regression.

In aquatic studies, it was also extended to solving prediction problems. For instance, Liong et al. addressed the issue of flood forecasting using Support Vector Regression (SVR) which is an extension of SVM. Another work by Xiang et al. utilized a LS-SVM model to deal with the water quality prediction problem in Liuxi River in Guangzhou.

However, none of these approaches is applied into urban scenarios, which is quite different from our applications. Moreover, those existing approaches process the data from a single source, and can hardly integrate the data from different sources. Thus, their applications in the urban scenarios are restricted.

3.1.1 DISADVANTAGES:

- The system is implemented only Multi-task Multi-view Learning Approaches.
- Instance-based learning models (IBL) is a family of learning algorithms that model a decision problem with instances or examples of training data that are deemed important to the model.

3.2 PROPOSED SYSTEM:

Data-driven Perspective: We present a novel data-driven approach to co-predict the future water quality among different stations with data from multiple domains. Additionally, the approach is not restricted to urban water quality prediction, but also can be applied to other multi-locations based coprediction problem in many other urban applications.

Influential Factor Identification: We identify spatially-related (such as POIs, pipe networks, and road networks) and temporally-related features (e.g., time of day, meteorology and water hydraulics), contributing to not only our application but also the general problem of water quality prediction.

Unified Learning Model: We present a novel spatio-temporal multi-view multi-task learning framework (stMTMV) to integrate multiple sources of spatio-temporal urban data, which provides a general framework of combining heterogeneous spatio-temporal properties for prediction, and can also be applied to other spatio-temporal based applications.

Real evaluation: We evaluate our method by extensive experiments that use real-world datasets in Shenzhen, China. The results demonstrate the advantages of our method beyond other baselines, such as ARMA, Kalman filter, and ANN, and reveal interesting discoveries that can bring social good to urban life.

3.2.1 ADVANTAGES:

1) Water quality data: We collect water quality data every five minutes from 15 water quality monitoring stations in Shenzhen City. It comprises residual chlorine (RC), turbidity (TU) and pH. In this paper, we only use RC as the indexfor water quality, since RC is the most important and effective measurement for water quality in current urban water distribution system.

2) Hydraulic data: Hydraulic data consists of flow and pressure, which are collected every five minutes from 13 flow sites and 14 pressure sites, respectively.

HARDWARE REQUIREMENTS:

- Processor - Pentium –IV
- RAM - 16 GB
- Hard Disk - 1TB

SOFTWARE REQUIREMENTS:

- **Operating system** : Windows 11 Home
- **Coding Language** : Python.
- **Front-End** : Python.
- **Back-End** : Django-ORM
- **Designing** : Html, css, javascript.
- **Data Base** : MySQL (WAMP Server).

3.3 PROJECT ARCHITECTURE

"The project architecture for predicting urban water quality leveraging ubiquitous data involves a systematic approach encompassing data collection from diverse sources such as IoT sensors, satellite imagery, weather stations, and municipal databases. Following data preprocessing to ensure accuracy and consistency, feature engineering extracts pertinent environmental and water quality indicators. Machine learning models, including regression, decision trees, and ensemble methods, are trained on historical data and validated for accuracy. Real-time data integration allows continuous monitoring and adaptive learning. Visualization through interactive dashboards aids in comprehending water quality trends and predictions. Deployment

on scalable cloud infrastructure with API access ensures accessibility and security. Continuous evaluation, stakeholder engagement, and iterative improvements solidify the project's effectiveness in addressing urban water quality challenges."

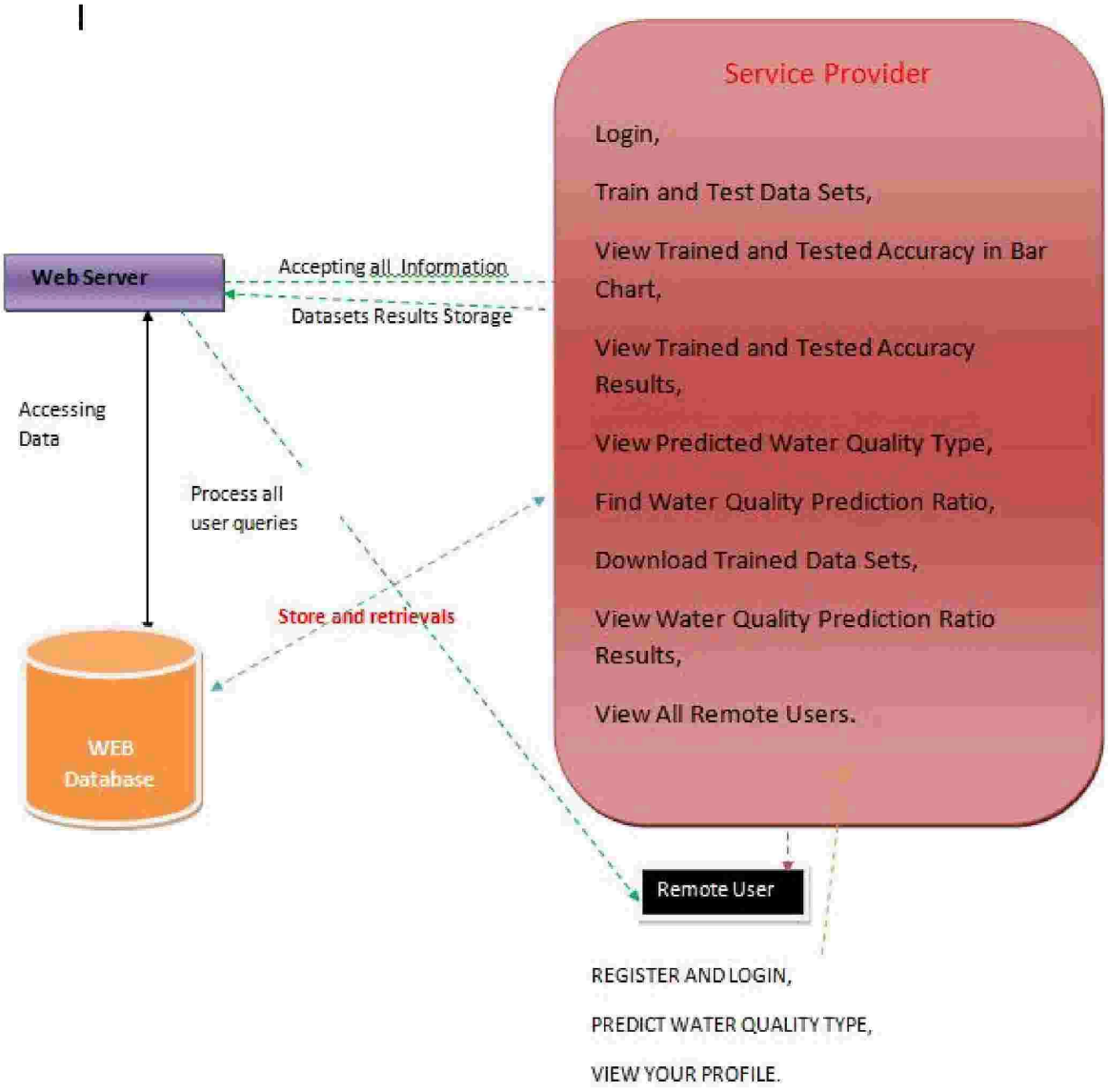

Figure 3.1 Project Architecture

3.4 ALGORITHMS:

1. Decision Tree Classification Algorithm
2. K-Nearest Neighbours(KNN)
3. Logistic Regression Algorithm
4. Naïve Bayes

5. Random Forest
6. SVM Algorithm

DECISION TREE CLASSIFIERS:

Decision tree classifiers are used successfully in many diverse areas. Their most important feature is the capability of capturing descriptive decision making knowledge from the supplied data. Decision tree can be generated from training sets. The procedure for such generation based on the set of objects (S), each belonging to one of the classes C1, C2, …, Ck is as follows:

Step 1. If all the objects in S belong to the same class, for example Ci, the decision tree for S consists of a leaf labelled with this class

Step 2. Otherwise, let T be some test with possible outcomes O1, O2,…, On. Each object in S has one outcome for T so the test partitions S into subsets S1, S2,… Sn where each object in Si has outcome Oi for T. T becomes the root of the decision tree and for each outcome Oi we build a subsidiary decision tree by invoking the same procedure recursively on the set Si.

Gradient boosting:

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.[1][2] When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest.A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

K-Nearest Neighbors (KNN):

- Simple, but a very powerful classification algorithm
- Classifies based on a similarity measure
- Non-parametric
- Lazy learning
- Does not “learn” until the test example is given
- Whenever we have a new data to classify, we find its K-nearest neighbors from the training data

Example :

Training dataset consists of k-closest examples in feature space. Feature space means, space with categorization variables (non-metric variables). Learning based on instances, and thus also works lazily because instance close to the input vector for test or prediction may take time to occur in the training dataset.

LOGISTIC REGREESION CLASSIFIERS :

Logistic regression analysis studies the association between a categorical dependent variable and a set of independent (explanatory) variables. The name logistic regression is used when the dependent variable has only two values, such as 0 and 1 or Yes and No. The name multinomial logistic regression is usually reserved for the case when the dependent variable has three or more unique values, such as Married, Single, Divorced, or Widowed. Although the type of data used for the dependent variable is different from that of multiple regression, the practical use of the procedure is similar.

This program computes binary logistic regression and multinomial logistic regression on both numeric and categorical independent variables. It reports on the regression equation as well as the goodness of fit, odds ratios, confidence limits, likelihood, and deviance. It performs a comprehensive residual analysis including diagnostic residual reports and plots. It can perform an independent variable subset selection search, looking for the best regression model with the fewest independent variables. It provides confidence intervals on predicted values and provides ROC curves to help determine the best cutoff point for classification. It allows you to validate your results by automatically classifying rows that are not used during the analysis

Naïve Bayes:

The naive bayes approach is a supervised learning method which is based on a simplistic hypothesis: it assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature .

Yet, despite this, it appears robust and efficient. Its performance is comparable to other supervised learning techniques. Various reasons have been advanced in the literature. In this tutorial, we highlight an explanation based on the representation bias. The naive bayes classifier is a linear classifier, as well as linear discriminant analysis, logistic regression or linear SVM (support vector machine). The difference lies on the method of estimating the parameters of the classifier (the learning bias).

While the Naive Bayes classifier is widely used in the research world, it is not widespread among practitioners which want to obtain usable results. On the one hand, the researchers found especially it is very easy to program and implement it, its parameters are easy to estimate, learning is very fast even on very large databases, its accuracy is reasonably good in comparison to the other approaches. On the other hand, the final users do not obtain a model easy to interpret and deploy, they does not understand the interest of such a technique.

Thus, we introduce in a new presentation of the results of the learning process. The classifier is easier to understand, and its deployment is also made easier. In the first part of this tutorial, we present some theoretical aspects of the naive bayes classifier. Then, we implement the approach on a dataset with Tanagra. We compare the obtained results (the parameters of the model) to those obtained with other linear approaches such as the logistic regression, the linear discriminant analysis and the linear SVM. We note that the results are highly consistent. This largely explains the good performance of the method in comparison to others. In the second part, we use various tools on the same dataset (Weka 3.6.0, R 2.9.2, Knime 2.1.1, Orange 2.0b and RapidMiner 4.6.0). We try above all to understand the obtained results.

Random Forest :

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

The first algorithm for random decision forests was created in 1995 by Tin Kam Ho[1] using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

An extension of the algorithm was developed by Leo Breiman and Adele Cutler, who registered "Random Forests" as a trademark in 2006 (as of 2019, owned by Minitab, Inc.).The extension combines Breiman's "bagging" idea and random selection of features, introduced first by Ho[1] and later independently by Amit and Geman[13] in order to construct a collection of decision trees with controlled variance.

SVM :

In classification tasks a discriminant machine learning technique aims at finding, based on an independent and identically distributed (iid) training dataset, a discriminant function that can correctly predict labels for newly acquired instances. Unlike generative machine learning approaches, which require computations of conditional probability distributions, a discriminant classification function takes a data point x and assigns it to one of the different classes that are a part of the classification task. Less powerful than generative approaches, which are mostly used when prediction involves outlier detection, discriminant approaches require fewer computational resources and less training data, especially for a multidimensional feature space and when only posterior probabilities are needed. From a geometric perspective, learning a classifier is equivalent to finding the equation for a multidimensional surface that best separates the different classes in the feature space.

SVM is a discriminant technique, and, because it solves the convex optimization problem analytically, it always returns the same optimal hyperplane parameter—in contrast to genetic algorithms (GAs) or perceptrons, both of which are widely used for classification in machine learning. For perceptrons, solutions are highly dependent on the initialization and termination criteria. For a specific kernel that transforms the data from the input space to the feature space, training returns uniquely defined SVM model parameters for a given training set, whereas the perceptron and GA classifier models are different each time training is initialized. The aim of GAs and perceptrons is only to minimize error during training, which will translate into several hyperplanes' meeting this requirement.

3.5 UML DIAGRAMS :

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.

2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.

3.5.1 USE CASE :

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

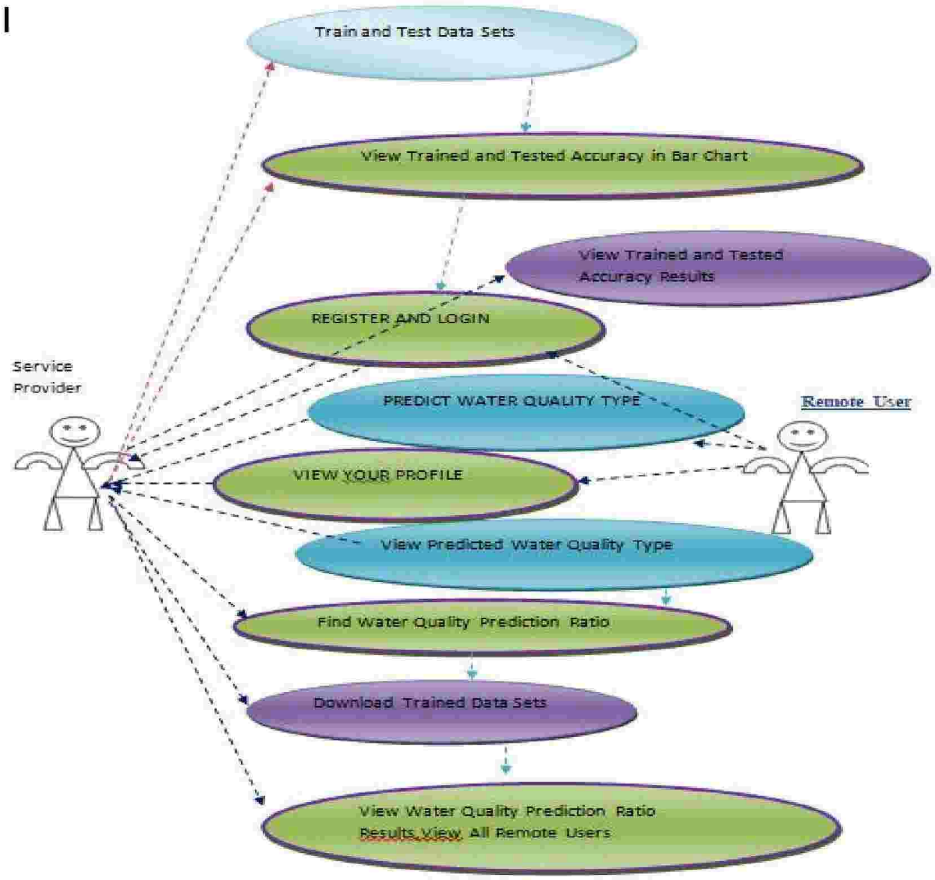

Figure 3.2 Use Case Diagram

3.5.2 CLASS DIAGRAM :

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

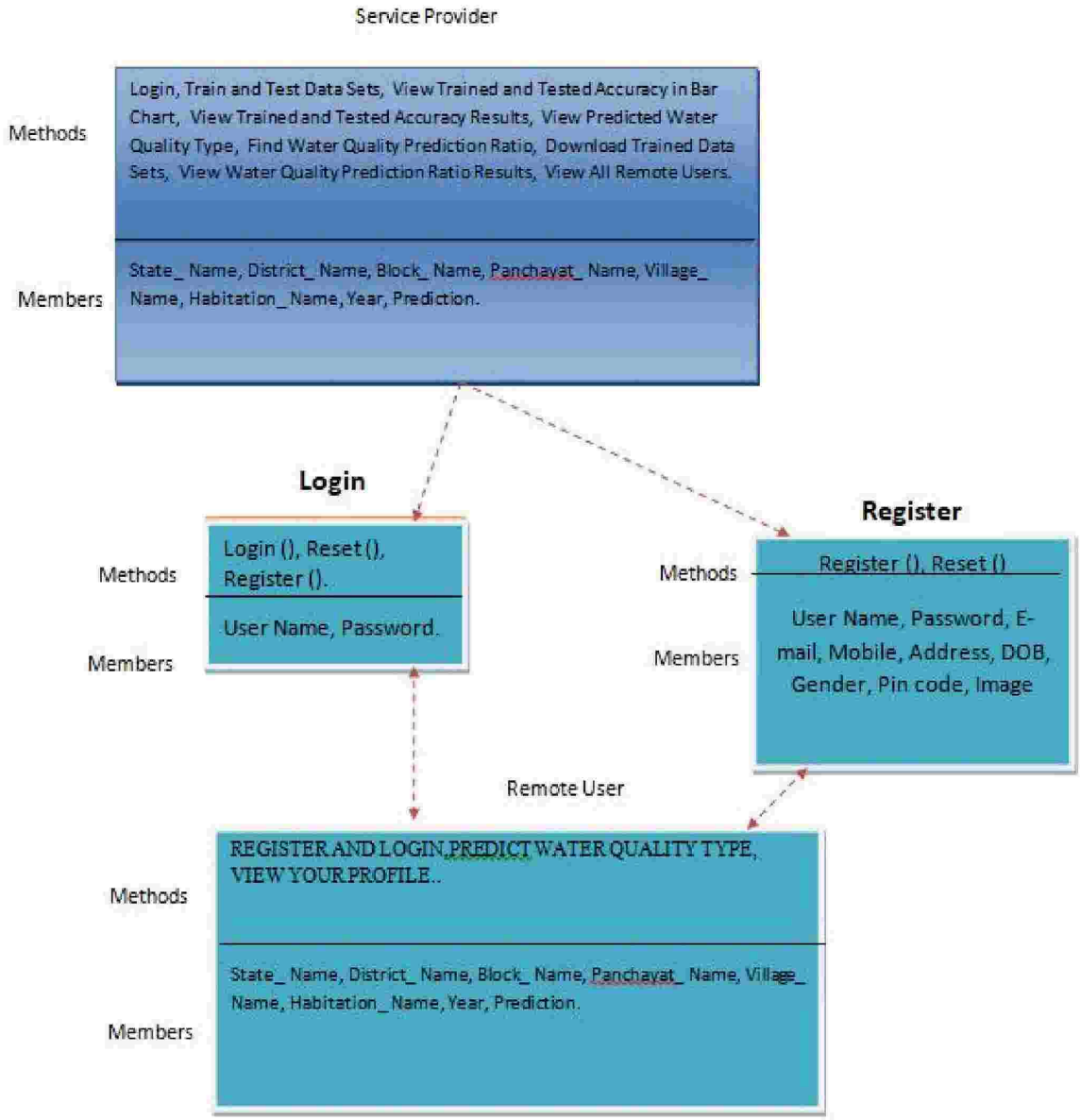

Figure 3.3 Class Diagram

3.5.3 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

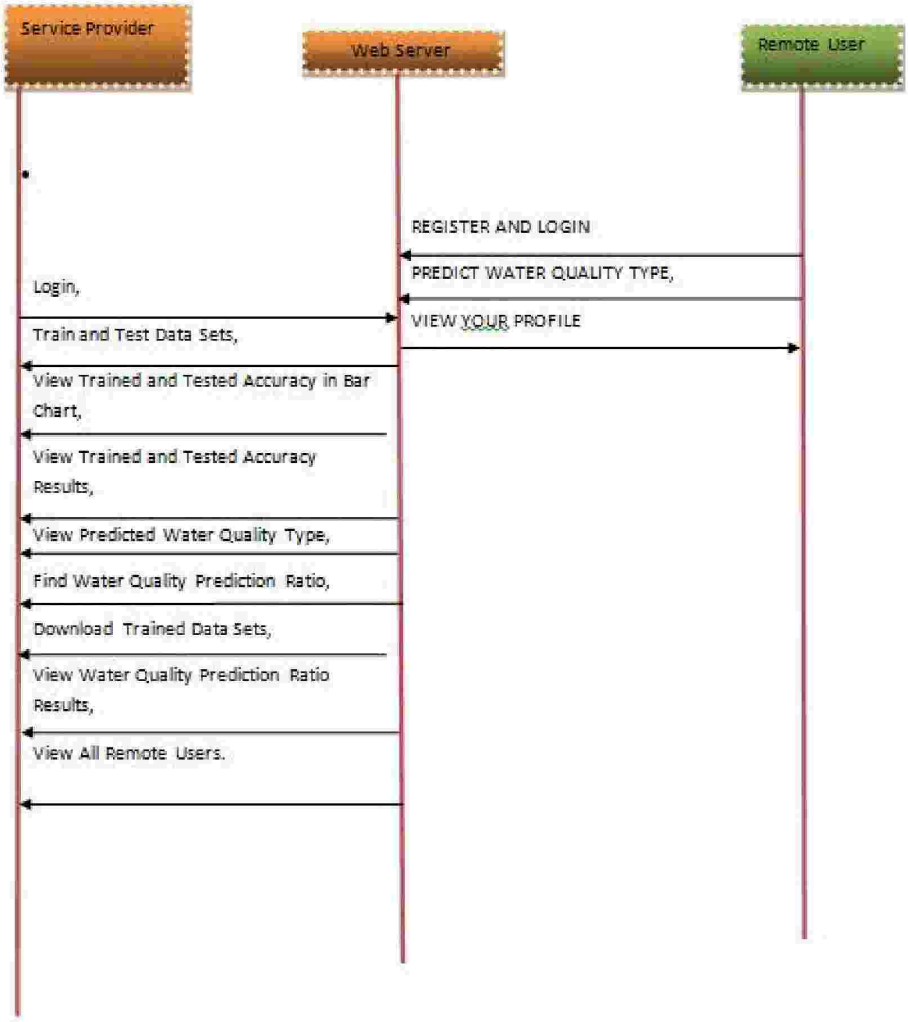

Figure 3.4 Sequence Diagram

3.5.4 DATA FLOW DIAGRAM :

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The

technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required. The technique exploits a method called top down expansion to conduct the analysis in a targeted way

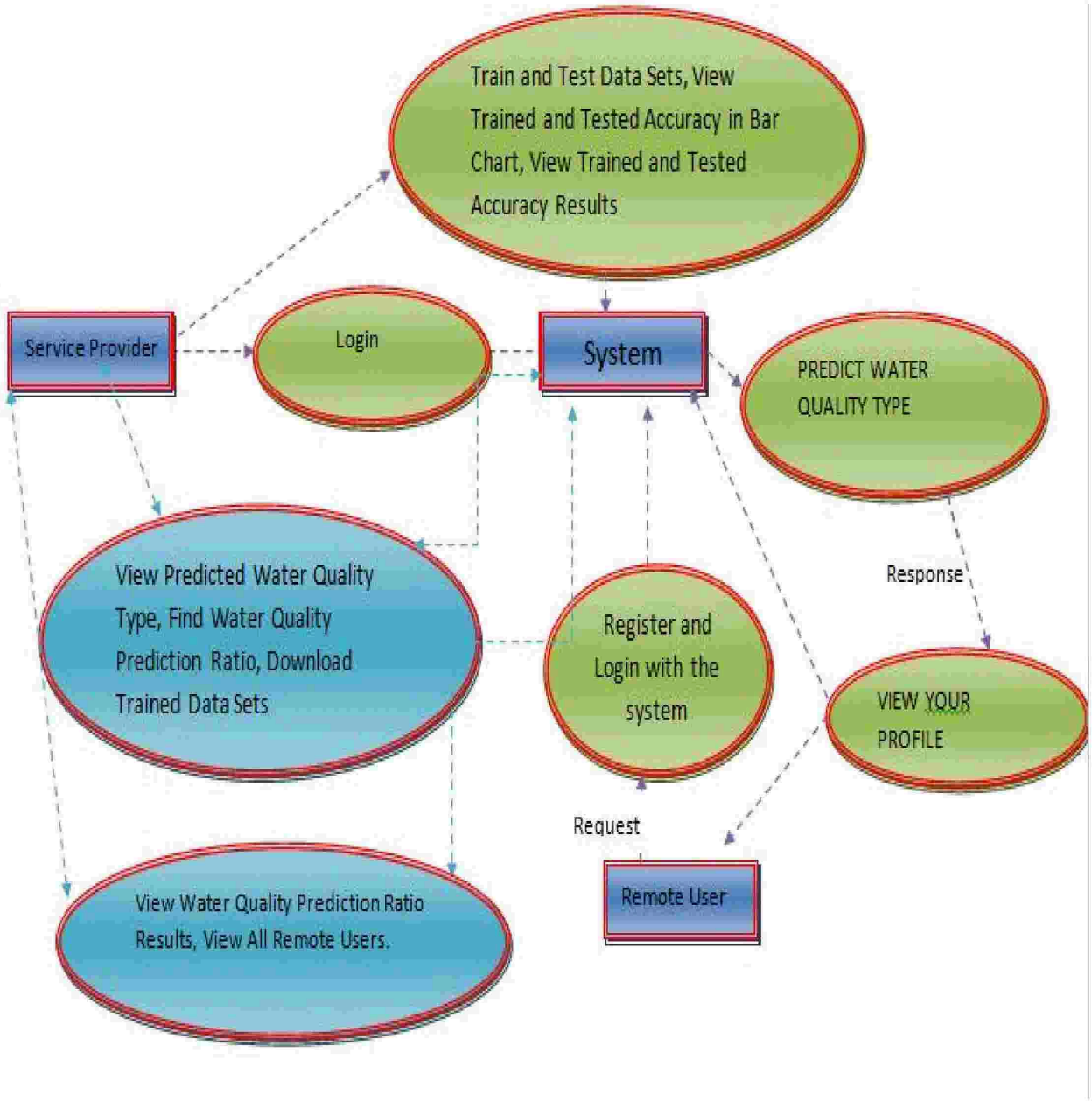

Figure 3.5 Data Flow Diagram

3.5.5 PACKAGE DIAGRAM:

Package diagram is UML structure diagram which shows structure of the designed system at the level of packages. The following elements are typically drawn in a package diagram: package, packageable element, dependency, element import, package import, package merge.

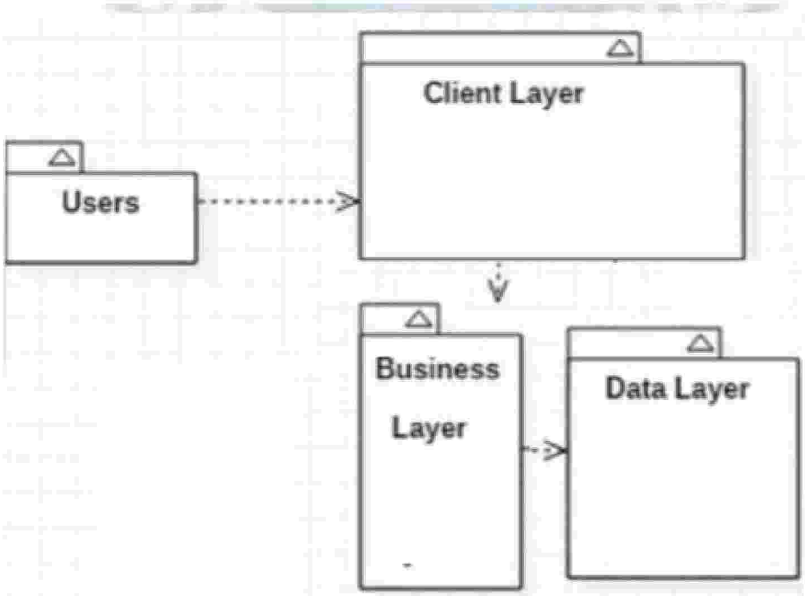

Figure 3.6 Package Diagram

3.5.6 PROFILE DIAGRAM:

A Profile diagram is any diagram created in a «profile» Package. Profiles provide a means of extending the UML. They are based on additional stereotypes and Tagged Values that are applied to UML elements, connectors and their components.

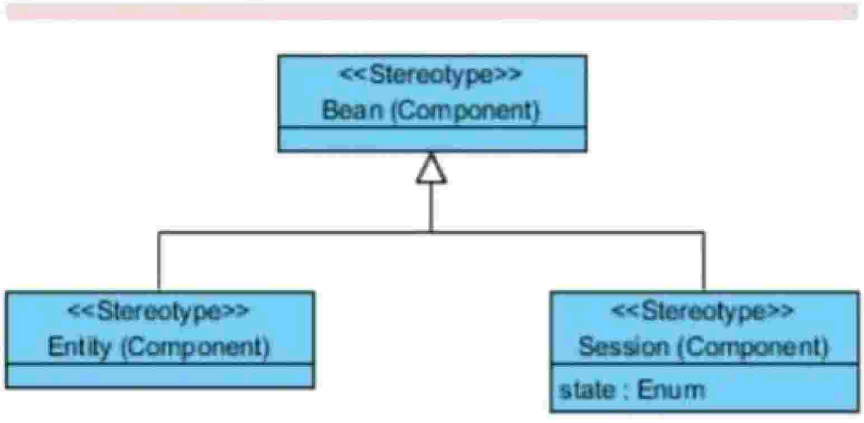

Figure 3.7 Profile Diagram

CHAPTER 4

MODULES

4.1 Data Collection Module:

This component involves interfacing with various IoT sensors deployed across the urban area. It includes protocols for retrieving data from these sensors, managing different data formats, handling connectivity issues, and ensuring a consistent stream of real-time data.

API Integration:

Integration with external APIs involves accessing diverse data sources such as weather APIs for meteorological data, satellite imagery APIs for land use information, municipal databases for historical water quality records, and potentially social media or citizen-contributed data sources for additional context. This module manages authentication, data retrieval, and periodic updates.

4.2 Feature Engineering Module:

Environmental Feature Extraction:

This module identifies and extracts relevant features from the collected data. It involves techniques such as feature selection, dimensionality reduction, and extracting environmental factors like temperature, humidity, precipitation, and proximity to industrial areas or water bodies.

Water Quality Parameter Calculation:

This component computes essential water quality indicators from the collected data, including pH levels, dissolved oxygen content, turbidity, presence of pollutants, microbial content, and other relevant parameters crucial in determining water quality.

4.3 Machine Learning Module:

Model Selection and Training:

This module involves experimenting with various machine learning algorithms such as regression (linear, logistic), decision trees, random forests, and potentially more advanced techniques like neural networks or gradient boosting to build predictive models. Models are trained using historical water quality data and environmental features.

Model Validation and Optimization:

Validation techniques such as cross-validation, hyperparameter tuning, and model evaluation metrics (RMSE, MAE, R^2 for regression; accuracy, precision, recall for classification) are employed to ensure the models generalize well to new data and are optimized for accuracy and performance.

4.4 Real-Time Data Processing Module:

Stream Processing:

This module involves setting up mechanisms to handle incoming real-time data streams. Techniques like stream processing frameworks or event-driven architectures are used to continuously process and analyze data as it arrives.

Adaptive Learning:

Implementing algorithms that allow the models to adapt and learn from incoming real-time data is crucial. This involves techniques such as online learning or incremental learning to update models with new data and evolving trends.

CHAPTER – 5
TECHNOLOGY DESCRIPTION

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++or Java. It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

What is Python

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

What can Python do

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).

- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way.

Good to know

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Introduction

Python applications will often use packages and modules that don't come as part of the standard library. Applications will sometimes need a specific version of a library, because the application may require that a particular bug has been fixed or the application may be written using an obsolete version of the library's interface.

This means it may not be possible for one Python installation to meet the requirements of every application. If application A needs version 1.0 of a particular module but application B needs version 2.0, then the requirements are in conflict and installing either version 1.0 or 2.0 will leave one application unable to run.

Different applications can then use different virtual environments. To resolve the earlier example of conflicting requirements, application A can have its own virtual environment with version 1.0 installed while application B has another virtual environment with version 2.0. If

application B requires a library be upgraded to version 3.0, this will not affect application A's environment. Python is the most widely used multi-purpose, high-level programming language at the moment. Python supports both Object-Oriented and Procedural programming paradigms. Python programmes are typically smaller than those written in other programming languages such as Java. Programmers must type relatively little, and the language's indentation requirement ensures that their code is always readable.Python is used by almost all tech giants, including Google, Amazon, Facebook, Instagram, Dropbox, Uber, and others.Python's greatest strength is its vast collection of standard libraries, which can be used for the following.

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

History of Python : -

What are the similarities between Python and the alphabet? Yes, both begin with ABC. It is abundantly clear that the programming language ABC is being referred to when we talk about ABC in the context of Python. ABC is a broadly useful programming language and programming climate, which had been created in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde and Informatica). Influencing the development of Python was ABC's greatest accomplishment. Python was first thought of at the end of the 1980s. During that time, Guido van Rossum worked on a distributed operating system called Amoeba at the CWI. Guido van Rossum stated in an interview with Bill Venners 1: At Centrum voor Wiskunde en

Informatica (CWI), I worked as an implementer on a team developing a language called ABC in the early 1980s. I don't have any idea how well individuals know ABC's effect on Python. I attempt to make reference to ABC's impact since I'm obliged to all that I mastered during that undertaking and to individuals who chipped away at it." Guido van Rossum went on later in the same interview: " I recalled all my experience and a portion of my dissatisfaction with ABC.

I decided to try to create a straightforward scripting language with some of ABC's best features without its drawbacks. So I began composing. I made a straightforward virtual machine, a basic parser, and a basic runtime.

How to Install Python on Windows and Mac :

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your System Requirements. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a Windows 64-bit operating system. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheatsheet here.The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

Download the Correct version into the system

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: https://www.python.org

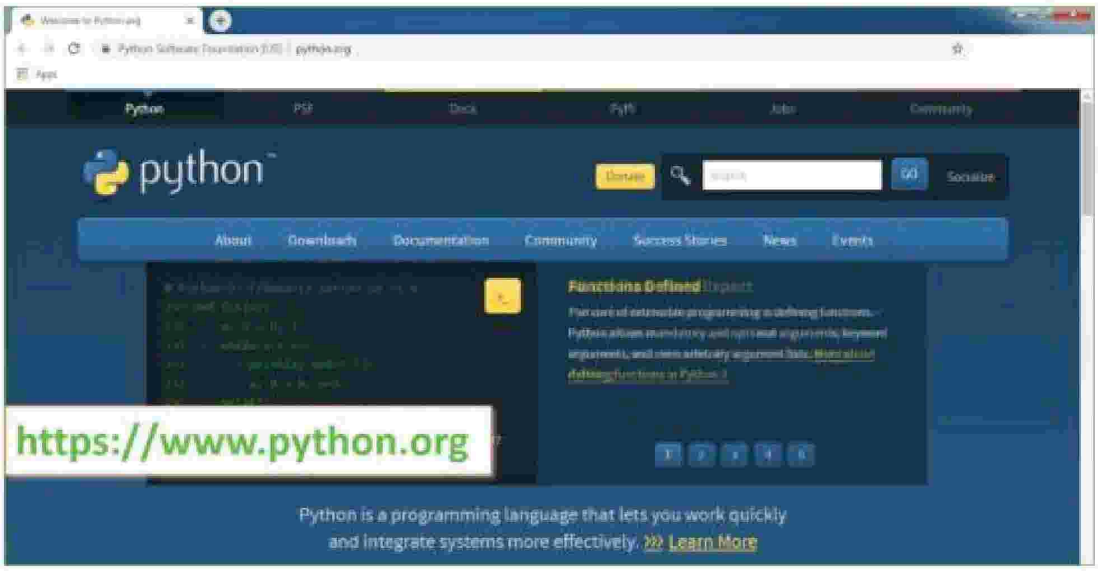

Figure 5.1 official python website

Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.

Figure 5.2 Latest version of python

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

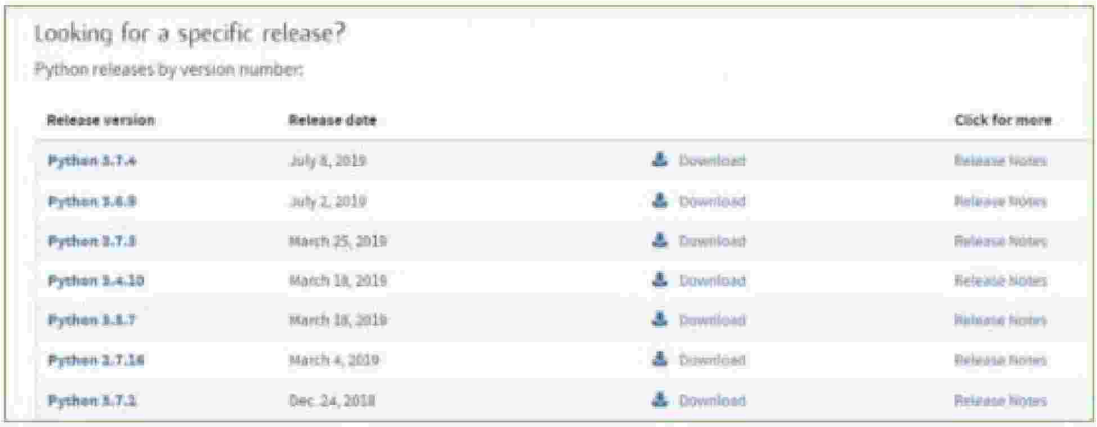

Figure 5.3 Release version

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.

Figure 5.4 Files

- To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

- To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.

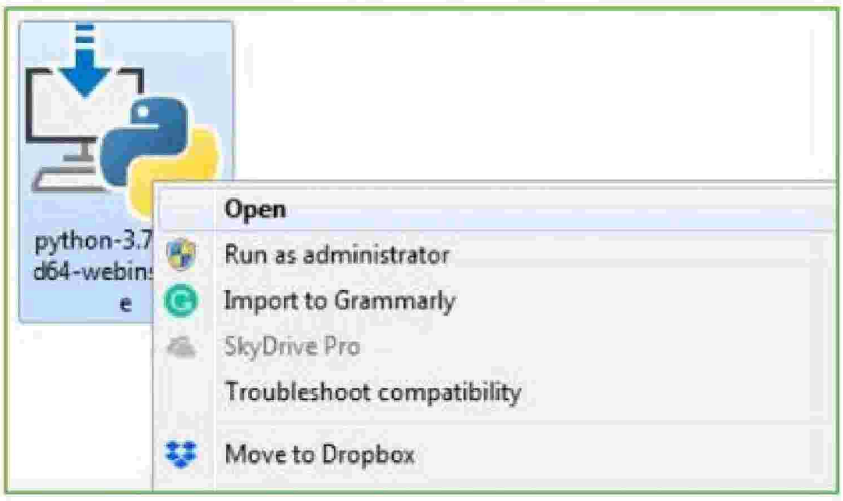

Figure 5.5 Application

Step 2: Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.

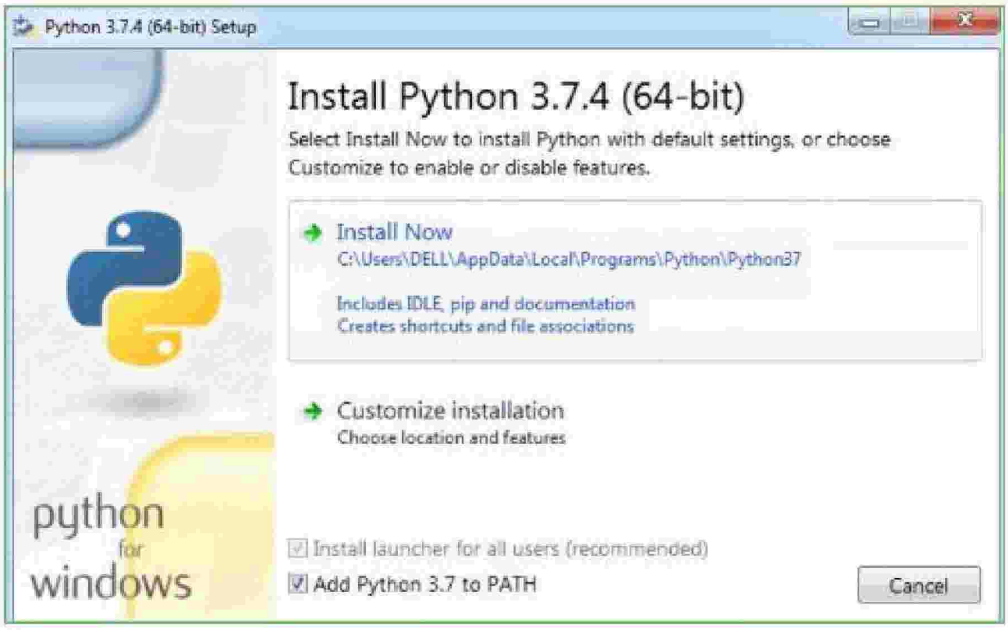

Figure 5.6 Installation

Step 3: Click on Install NOW After the installation is successful. Click on Close.

Figure 5.7 Installation successful

With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

Verify the Python Installation

Step 1: Click on Start

Step 2: In the Windows Run Command, type “cmd”

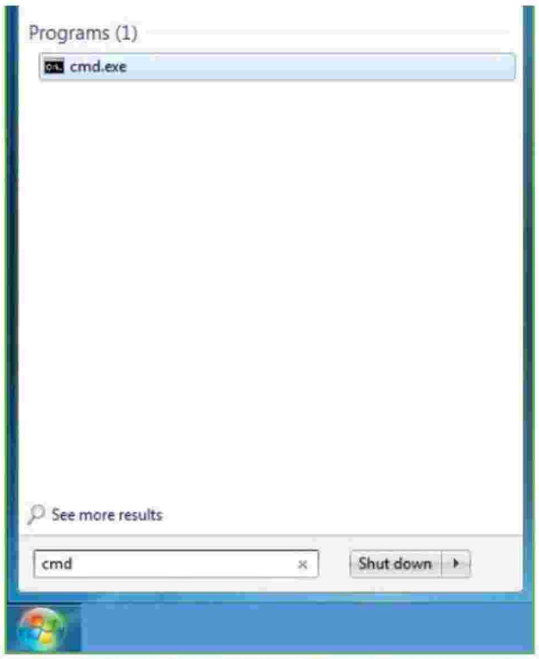

Figure 5.8 Command Prompt

Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type python –V and press Enter.

Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works

Step 1: Click on Start

Step 2: In the Windows Run command, type “python idle”

Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

Step 4: To go ahead with working in IDLE you must first save the file. Click on File > Click on Save

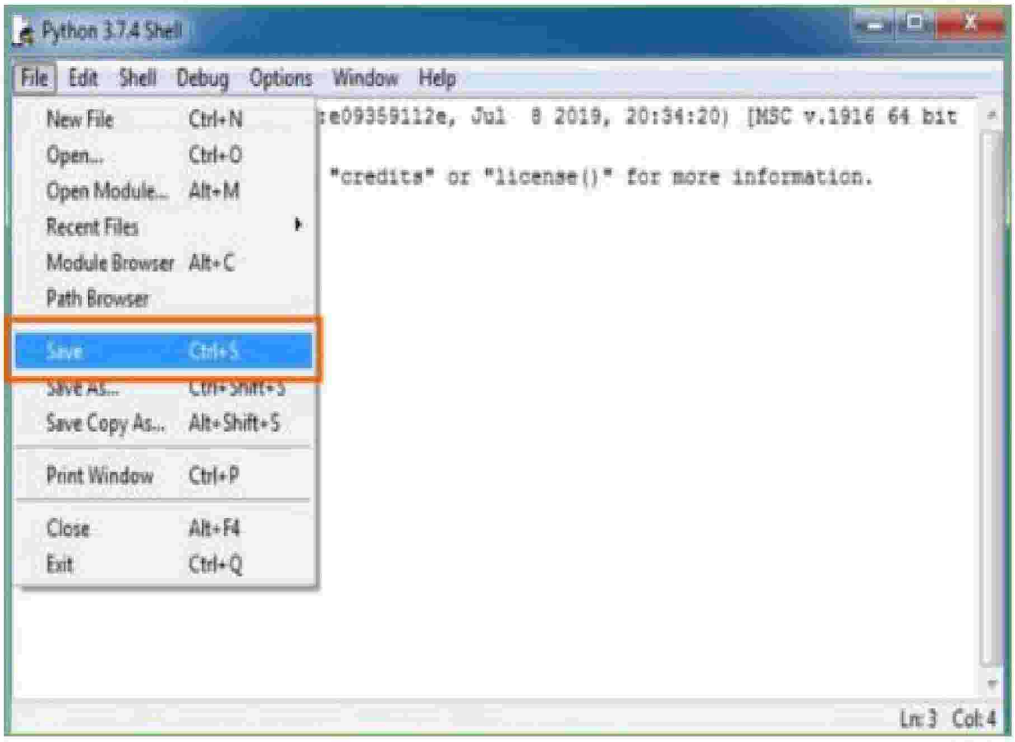

Figure 5.9 Save the file

Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step 6: Now for e.g. enter print ("Hey World") and Press Enter.

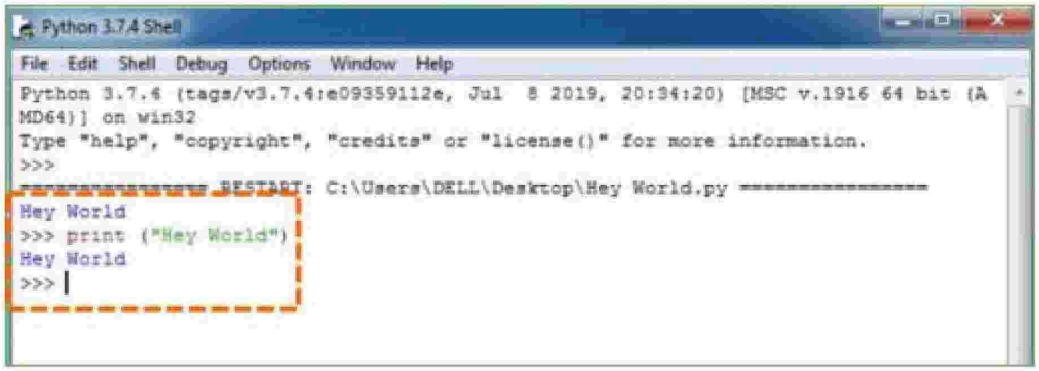

Figure 5.10 Program

MACHINE LEARNING : -

Before we investigate the subtleties of different AI techniques, how about we start by seeing what AI is, and what it isn't. Although machine learning is frequently referred to as a subfield of artificial intelligence, I believe that categorization can frequently be initially misleading. The investigation of AI positively emerged from research in this specific circumstance, however in

the information science use of AI techniques, it's more useful to consider AI for the purpose of building models of information. In its most basic form, machine learning is the creation of mathematical models that aid in data comprehension. Learning" enters the conflict when we give these models tunable boundaries that can be adjusted to noticed information; The program can thus be regarded as "learning" from the data. These models can be used to predict and comprehend aspects of newly observed data once they have been fitted to data that has already been observed. The more philosophical discussion regarding the degree to which this kind of mathematical, model-based "learning" is comparable to the "learning" exhibited by the human brain will be left up to the reader. Understanding the issue setting in AI is crucial for utilizing these devices successfully, thus we will begin for certain general classifications of the kinds of approaches we'll examine here.

Classes Of Machine Inclining :-

Machine learning can be broken down into two main categories at the most fundamental level: learning under supervision and unsupervised learning.

Modeling the relationship between the data's measured features and some label is one aspect of supervised learning. New, unidentified data can be labeled with the help of this model once it has been established. Regression and classification tasks are two more subcategories of this: The labels in classification are discrete categories, whereas the labels in regression are continuous quantities. In the following section, we will see examples of both kinds of supervised learning.

Unsupervised learning is often referred to as "letting the dataset speak for itself," and it involves modeling the features of a dataset without using any labels. These models incorporate undertakings like bunching and dimensionality decrease. Bunching calculations distinguish unmistakable gatherings of information, while dimensionality decrease calculations look for additional brief portrayals of the information. In the following section, we will see examples of both kinds of unsupervised learning.

Need for AI

Individuals, as of now, are the most keen and high level species on earth since they can think, assess and tackle complex issues. On the opposite side, computer based intelligence is still in its underlying stage and haven't outperformed human knowledge in numerous viewpoints. Then the inquiry that is the need to make machine learn? The most appropriate justification behind doing this is, "to decide, in view of information, with proficiency and scale".

Recently, associations are putting vigorously in fresher advancements like Man-made brainpower, AI and Profound Figuring out how to get the critical data from information to play out a few true errands and tackle issues. We can call it information driven choices taken by machines, especially to mechanize the cycle. Problems that cannot naturally be programmed can benefit from these data-driven decisions rather than programming logic. The truth of the matter is that we can't manage without human insight, however other perspective is that we as a whole need to take care of genuine issues with proficiency at a gigantic scope. Because of this, machine learning is required..

Applications of Machines Learning :-

Researchers assert that we are in the golden age of AI and machine learning, with machine learning being the technology with the fastest growth rate. It is used to solve a lot of real-world complex problems that can't be solved using the traditional method. Emotion analysis, sentiment analysis, error detection and prevention, weather forecasting, stock market analysis and forecasting, speech synthesis, speech recognition, customer segmentation, object recognition, fraud detection and prevention, and product recommendation for online shoppers are just a few examples of ML's real-world applications. a) Terminologies of Machine Learning A model is a particular representation that is learned from data by using a machine learning algorithm. A model is likewise called a speculation.

• Highlight - A component is an individual quantifiable property of the information. A feature vector makes it simple to describe a collection of numerical features. The model receives input in the form of feature vectors. For instance, there may be characteristics like color, smell, taste, and so on that can be used to predict a fruit.

• Target (Label): The value that will be predicted by our model is referred to as a target variable or label. For the organic product model talked about in the component segment, the mark with each arrangement of information would be the name of the organic product like apple, orange, banana, and so on.

• Training: The idea is to provide a set of inputs (features) and expected outputs (labels), so that after training, a model (hypothesis) will map new data to one of the categories trained on.

• Prediction: When our model is ready, it can be given a set of inputs and a predicted output (a label) will be generated.

(b) Types of Machine Learning • Supervised Learning: This involves using classification and regression models to learn from a training dataset with labeled data. Until the required level of performance is reached, this learning process continues.

• Unsupervised Learning: Using factor and cluster analysis models, this method uses unlabeled data to learn more about the data itself by locating its underlying structure.

• Semi-supervised Learning: This method uses a small amount of labeled data and unlabeled data, similar to unsupervised learning. Utilizing named information tremendously expands the learning precision and is likewise more savvy than Managed Learning.

• Reinforcement Learning: This method entails figuring out the best course of action through trial and error. Learning behaviors that are based on the current state and will maximize reward in the future determine the subsequent action.

Packages and Versions

astunparse ==1.6.3

certifi ==2022.12.7

charset-normalizer ==3.0.1

contourpy ==1.0.7

cycler ==0.11.0

Django ==3.0.4

et-xmlfile ==1.1.0

fonttools ==4.38.0

gast ==0.3.3

google-pasta ==0.2.0

grpcio ==1.51.1

h5py ==2.10.0

idna ==3.4

importlib-metadata ==6.0.0

joblib ==1.2.0

Keras ==2.3.1

Keras-Applications ==1.0.8

Keras-Preprocessing ==1.1.2

kiwisolver ==1.4.4

Markdown ==3.4.1

matplotlib ==3.6.3

mysql-connector-python ==8.0.32

mysqlclient ==2.1.1

numpy ==1.21.4

oauthlib ==3.2.2

openpyxl ==3.1.0

opt-einsum ==3.3.0

packaging ==23.0

pandas ==1.3.5

Pillow ==9.4.0

pip ==20.1.1

protobuf ==3.20.3

pyparsing ==3.0.9

opencv-contrib-python 4.5.2.52

CHAPTER – 6

EXPERIMENTAL RESULTS

6.1 Screenshots

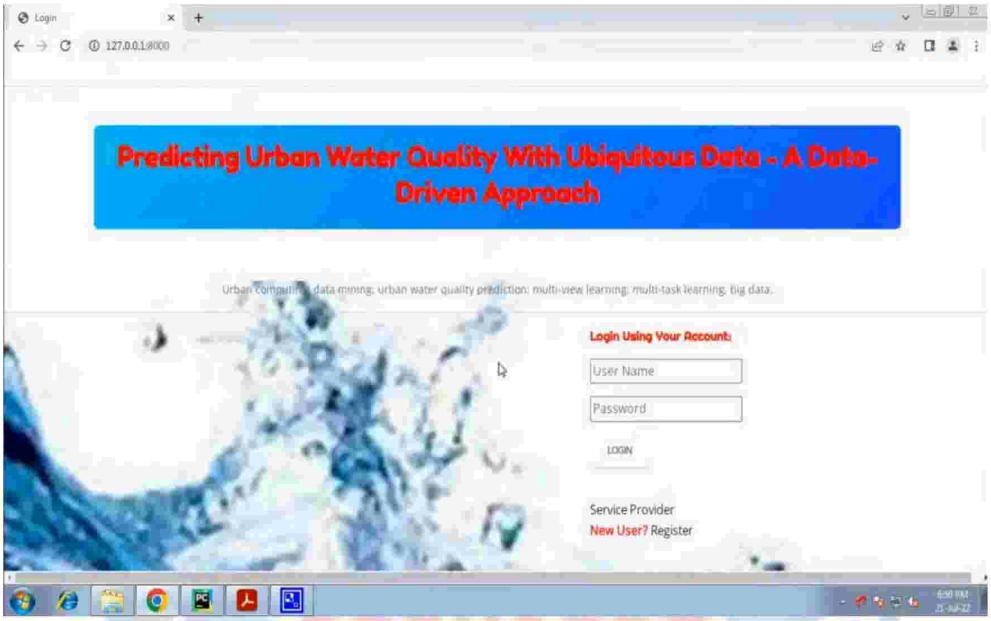

Figure 6.1 user login

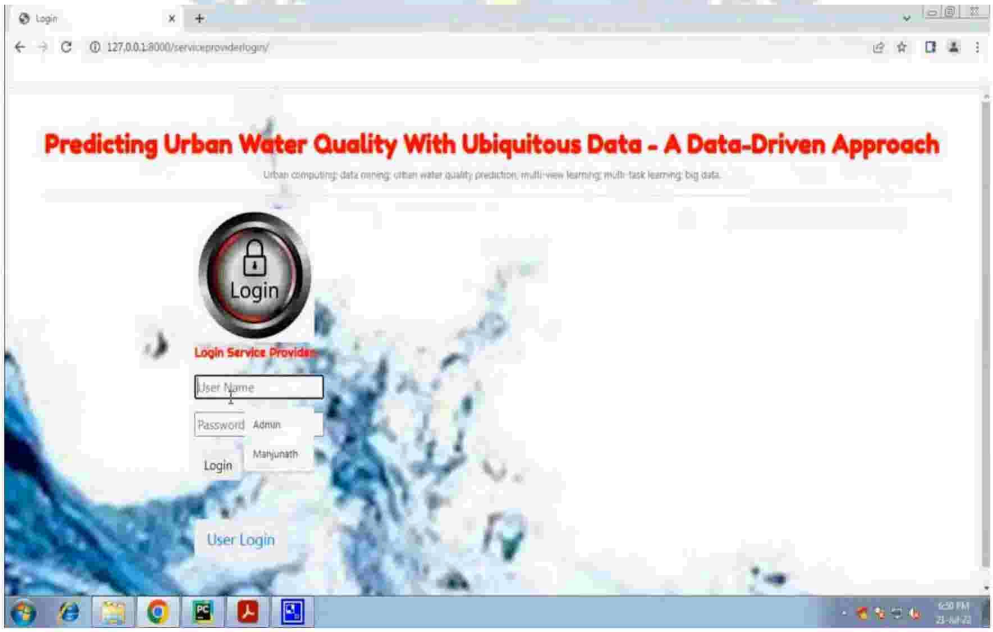

Figure 6.2 Login service provider

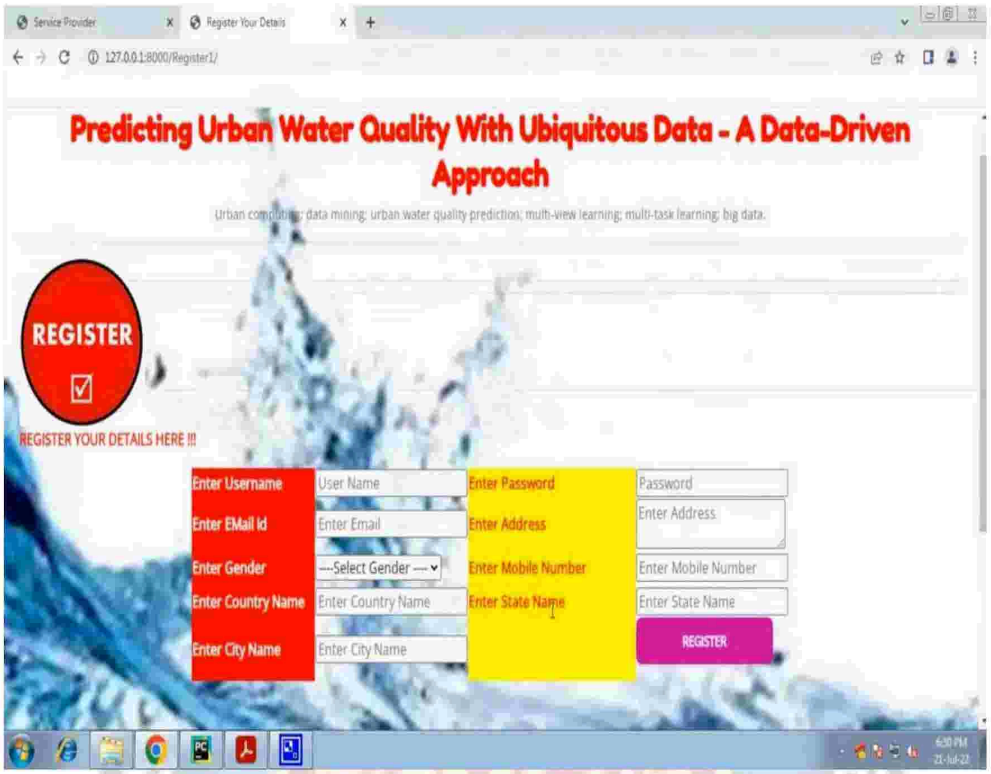

Figure 6.3 Registering Details

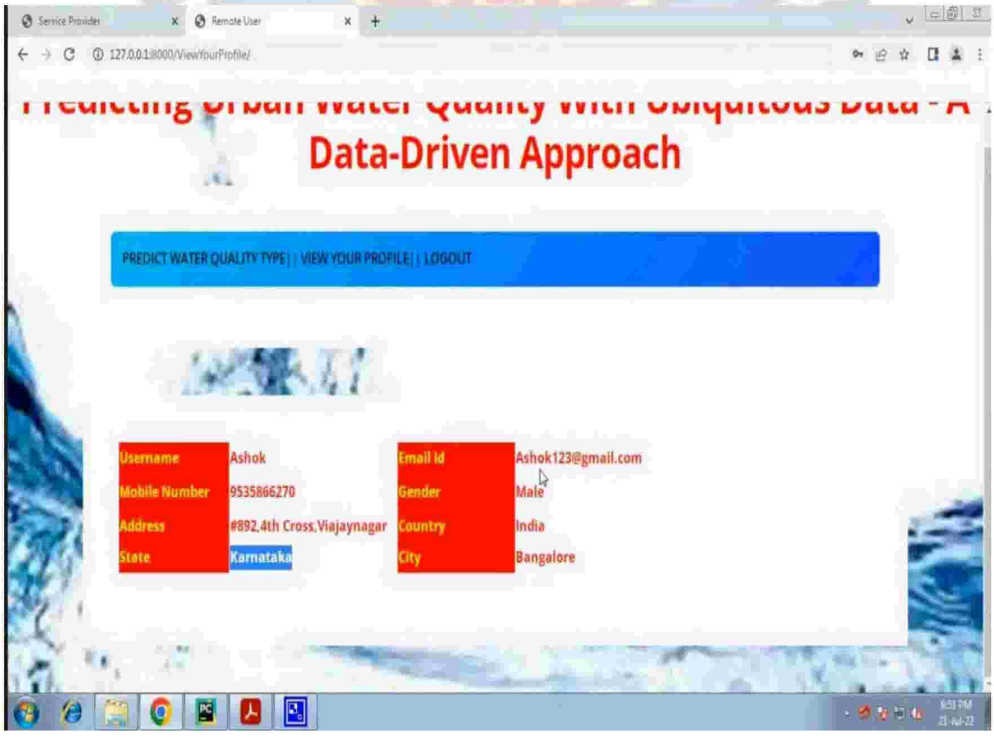

Figure 6.4 Viewing Profile

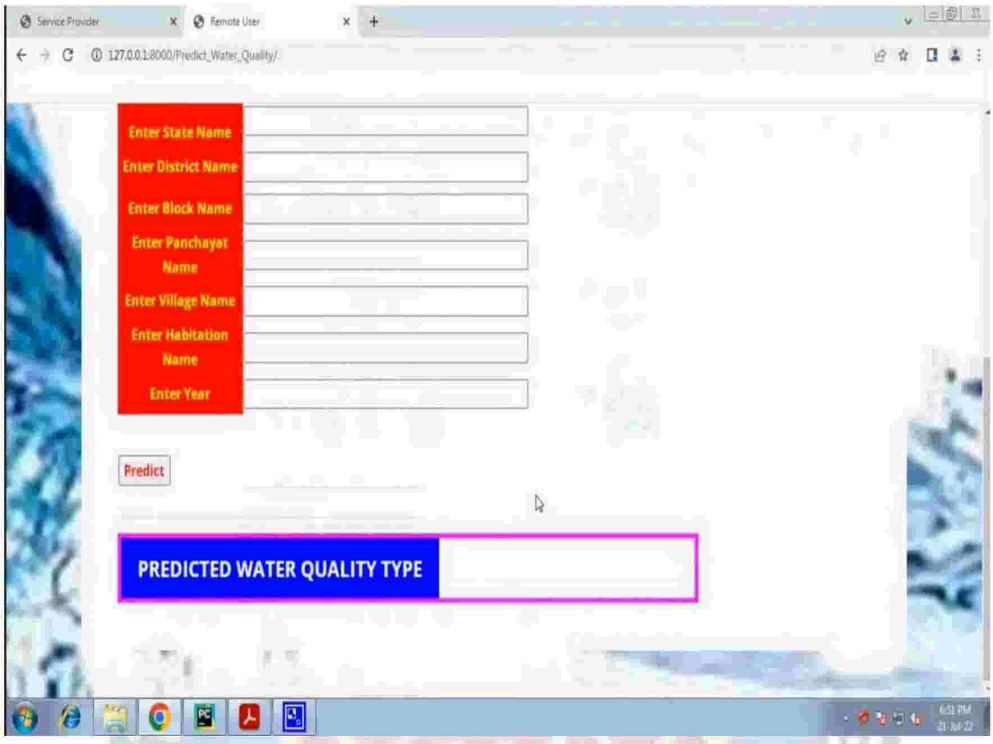

Figure 6.5 Predicting Water Quality Type

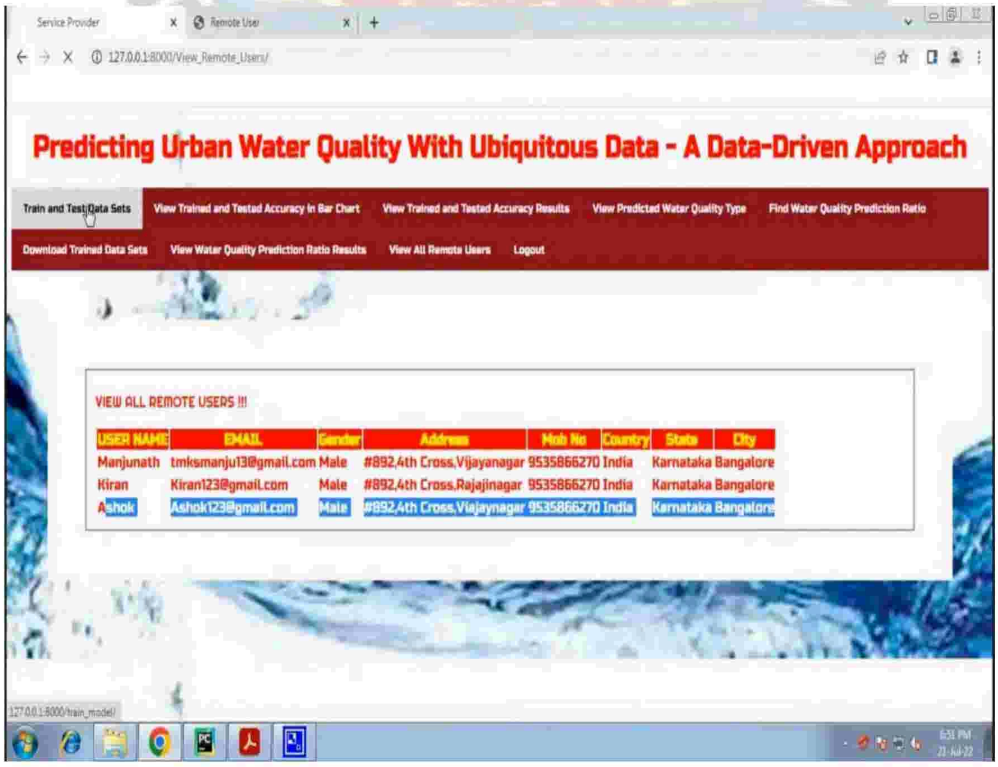

Figure 6.6 View all remote users

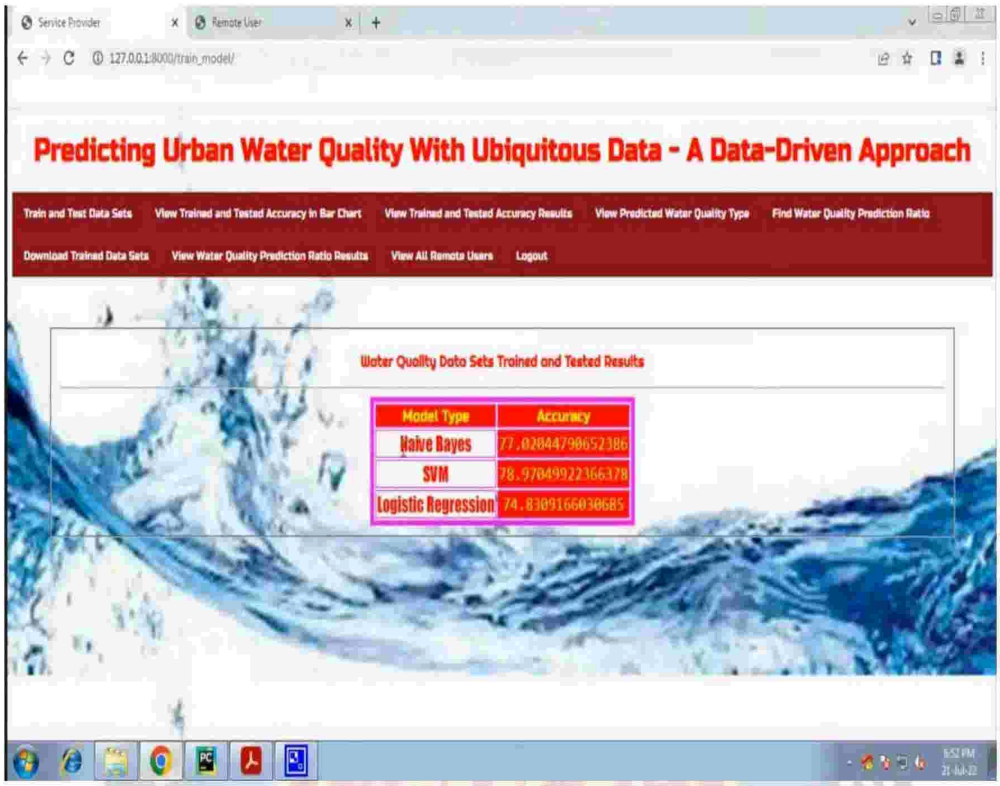

Figure 6.7 Water quality data sets trained and tested results

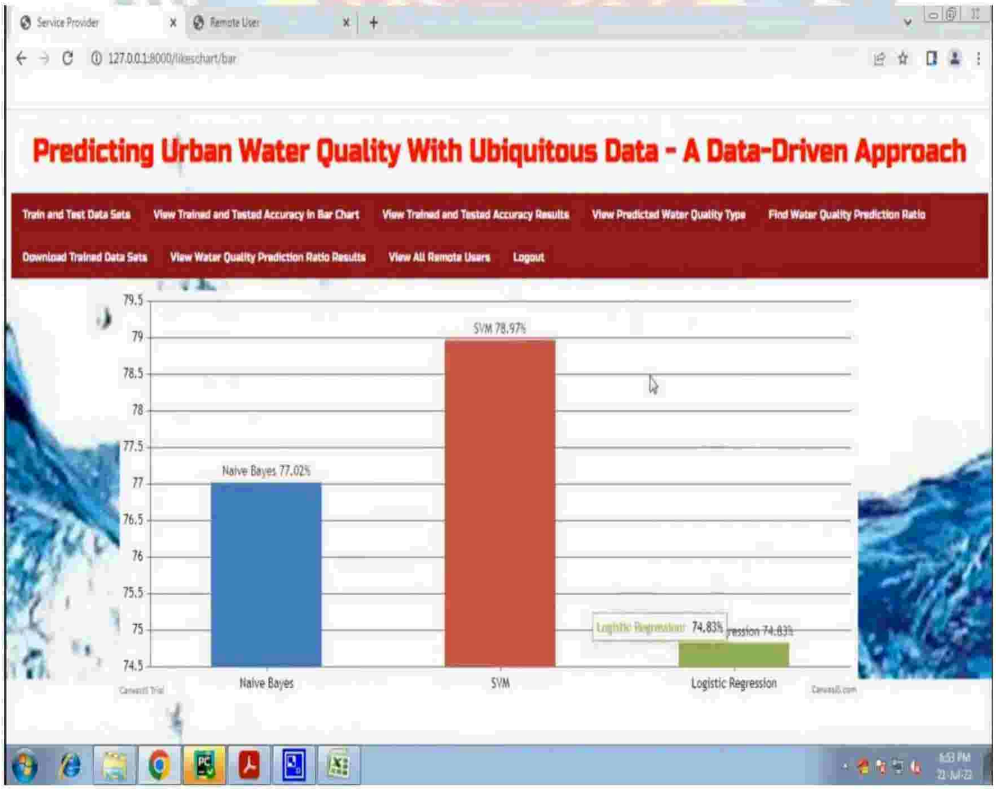

Figure 6.8 Bar chart

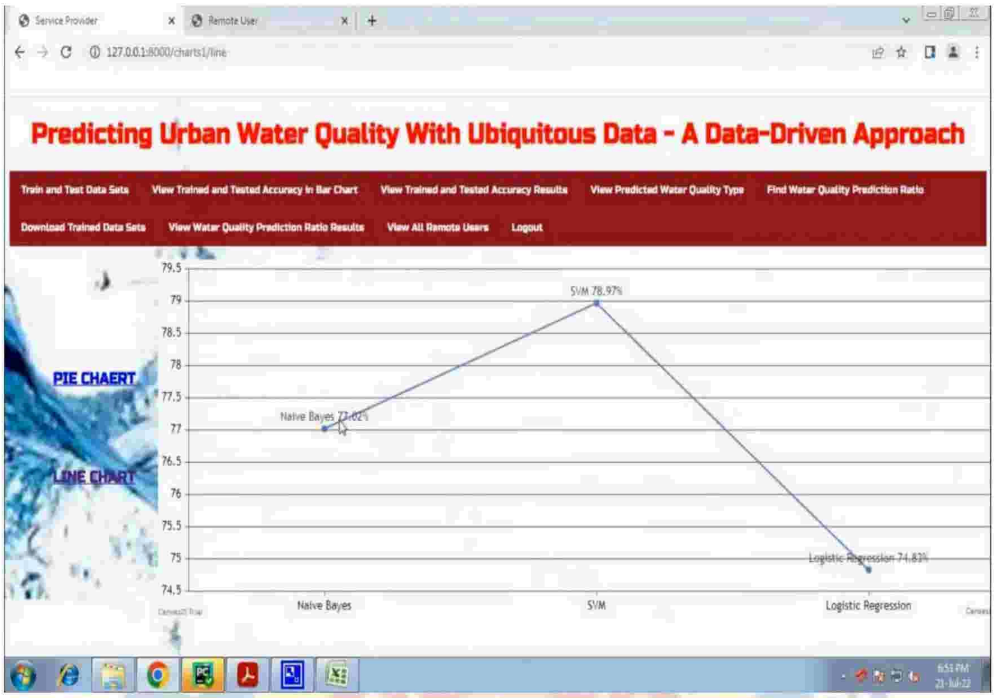

Figure 6.9 Line chart

Figure 6.10 Pie chart

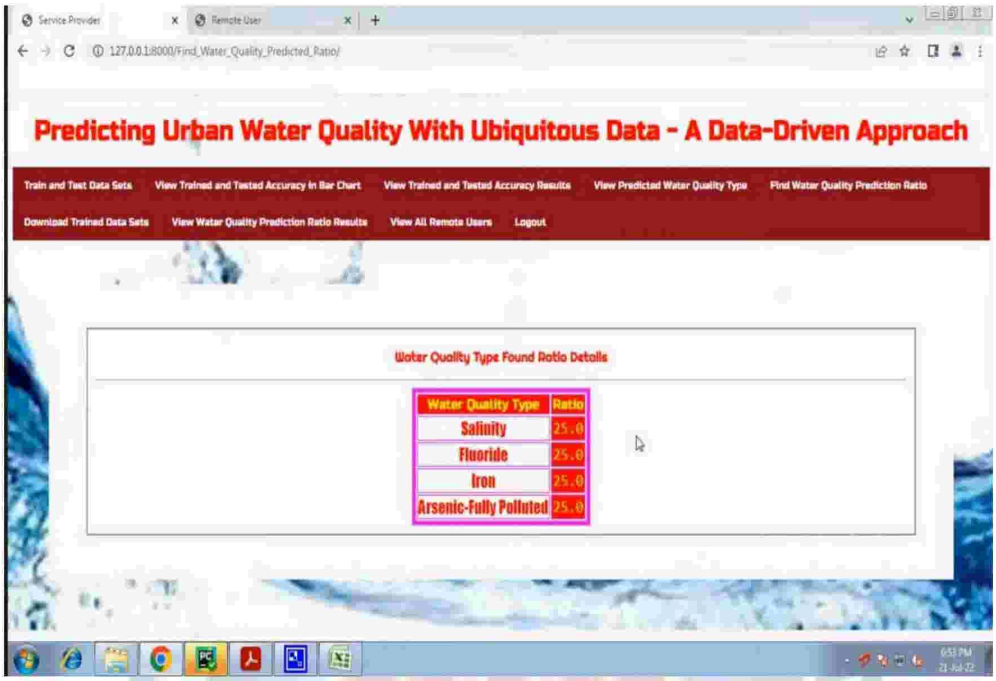

Figure 6.11 water quality type found ratio details

CHAPTER – 7

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable miner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cKNNot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cKNNot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.1 TEST CASES:

S.no	Test Case	Excepted Result	Result
1	User register	User register successfully	Pass
2	User login	User login successfully	Pass
3	Add Data	Add Data Successfully	Pass
4	Data View	Data View Successfully	Pass
5	Data Descriptions	Data Descriptions Successfully	Pass
6	Correlation Matrix	Correlation Matrix Successfully	Pass
7	Box plot	Box plot Successfully	Pass
8	Heat Frequency graph	Successfully get Heat Frequency graph	Pass
9	Distribution on Age graph	Successfully get Distribution on Age graph	Pass

10	Calculated Frequency	Successfully get Calculated Frequency	Pass
11	Scatter matrix	Successfully get Scatter matrix	Pass
12	Age Wise Disease Scatter graph	Successfully get Age Wise Disease Scatter graph	Pass
13	Cleaned data	Successfully get Cleaned data	Pass
14	Random Forest Confusion matrix	Successfully get Random Forest Confusion matrix	Pass
15	Decision Tree Confusion matrix	Successfully get Decision Tree Confusion matrix	Pass
16	Random Forest Roc	Successfully get Random Forest Roc	Pass
17	SVC Curve	Successfully get Successfully get	Pass
18	Algorithm Accuracy	Algorithm Accuracy Successfully	Pass

CHAPTER – 8

CODE IMPLEMENTATION

CODE:

```
from django.shortcuts import render, redirect, get_object_or_404
import pandas as pd
from sklearn.metrics import classification_report,accuracy_score,confusion_matrix
# Create your views here.
From Remote_User.models import
ClientRegister_Model,water_quality_type,detection_ratio,detection_accuracy
def login(request):
    if request.method == "POST" and 'submit1' in request.POST:
        username = request.POST.get('username')
        password = request.POST.get('password')
        try:
            enter = ClientRegister_Model.objects.get(username=username,password=password)
            request.session["userid"] = enter.id
            return redirect('ViewYourProfile')
        except:
            pass
    return render(request,'RUser/login.html')
def Register1(request):
    if request.method == "POST":
        username = request.POST.get('username')
        email = request.POST.get('email')
        password = request.POST.get('password')
        phoneno = request.POST.get('phoneno')
        country = request.POST.get('country')
        state = request.POST.get('state')
        city = request.POST.get('city')
        address = request.POST.get('address')
        gender = request.POST.get('gender')
```

```
        ClientRegister_Model.objects.create(username=username, email=email,
password=password, phoneno=phoneno, country=country, state=state, city=city,
address=address, gender=gender)
        obj = "Registered Successfully"
        return render(request, 'RUser/Register1.html',{'object':obj})
    else:
        return render(request,'RUser/Register1.html')

def ViewYourProfile(request):
    userid = request.session['userid']
    obj = ClientRegister_Model.objects.get(id= userid)
    return render(request,'RUser/ViewYourProfile.html',{'object':obj})


def Predict_Water_Quality(request):
    if request.method == "POST":
      if request.method == "POST":

            State_Name = request.POST.get('State_Name')
            District_Name = request.POST.get('District_Name')
            Block_Name = request.POST.get('Block_Name')
            Panchayat_Name = request.POST.get('Panchayat_Name')
            Village_Name = request.POST.get('Village_Name')
            Habitation_Name = request.POST.get('Habitation_Name')
            Year = request.POST.get('Year')

        df = pd.read_csv('Water_Quality_Datasets.csv', encoding='latin-1')

        def apply_results(results):
            if (results == 'Salinity'):
                return 0
            elif (results == 'Fluoride'):
                return 1
            elif (results == 'Iron'):
                return 2
```

```
elif (results == 'Arsenic'):
    return 3
elif (results == 'Nitrate'):
    return 4
```

CHAPTER – 9

CONCLUSION AND FUTURE ENHANCEMENT

This paper presents a novel data-driven approach to forecast the water quality of a station by fusing multiple sources of urban data. We evaluate our approach based on Shenzhen's water quality and various urban data. The experimental results demonstrate the effectiveness and efficiency of our approach. Specifically, our approach outperforms the traditional RC decay model and other classical time series predictive models (ARMA, Kalman) in terms of RMSE metric. Meanwhile, as our approach consists of two components, each of the components demonstrates its effectiveness through extensive experiments and analysis. In particular, the first component is the influential factors identification, which explores the factors that affect the urban water quality via extensive experiments and analysis in Section 3 and 4. The second one is a spatiotemporal multi-view multi-task learning (STMTMV) framework that consists of multi-view learning and multi-task learning. The experiments have shown that STMTMV has a predictive accuracy of around 85% for forecasting next 1-4 hours, which outperforms the single-task methods (LR) by approximately 11% and the single-view methods (t-view and s-view) by approximately 11% and 12%, respectively. The code has been released at: https://www.microsoft.com/en-us/research/publication/urbanwater- quality-prediction-based-multi-task-multi-view-learning-2/ In future, we plan to deal with the water quality inference problems in the urban water distribution systems through a limited number of water quality monitor stations.

REFERENCES

[1] W. H. Organization, Guidelines for drinking-water quality, 2004, vol. 3.

[2] L. A. Rossman, R. M. Clark, and W. M. Grayman, "Modeling chlorine residuals in drinking-water distribution systems," Journal of environmental engineering, vol. 120, no. 4, pp. 803–820, 1994.

[3] Y. Zheng, "Methodologies for cross-domain data fusion: An overview," IEEE Transactions on Big Data, vol. 1, no. 1, pp. 16–34, 2015.

[4] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: Concepts, methodologies, and applications," ACM Transactions on Intelligent Systems and Technology, vol. 5, no. 3, pp. 38:1–38:55, 2014.

[5] Y. Zheng, H. Zhang, and Y. Yu, "Detecting collective anomalies from multiple spatio-temporal datasets across different domains," 2015.

[6] Y. Liu, Y. Zheng, Y. Liang, S. Liu, and D. S. Rosenblum, "Urban water quality prediction based on multi-task multi-view learning," in Proceedings of the International Joint Conference on Artificial
Intelligence, 2016.

[7] H. Cohen, "Free chlorine testing," http://www.cdc.gov/safewater/chlorineresidual-testing.html, 2014, accessed on 5 August 2016.

[8] B. D. Barkdoll and H. Didigam, "Effect of user demand on water quality and hydraulics of distribution systems," in Proceedings of the World Water and Environmental Resources Congress, 2003.

[9] P. Castro and M. Neves, "Chlorine decay in water distribution systems case study–lousada network," Electronic Journal of Environmental, Agricultural and Food Chemistry, vol. 2, no. 2, pp. 261–266, 2003.

[10] L. W. Mays, Water distribution system handbook, 1999.

[11] L. A. Rossman and P. F. Boulos, "Numerical methods for modeling water quality in distribution systems: A comparison," Journal of Water Resources planning and management, vol. 122, no. 2, pp. 137–146, 1996.

[12] W. M. Grayman, R. M. Clark, and R. M. Males, "Modeling distributionsystem water quality: dynamic approach," Journal of Water Resources Planning and Management, vol. 114, no. 3, pp. 295–312, 1988.

[13] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in Proceedings of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 2003, pp. 2–11.

[14] G. Luo, K. Yi, S.-W. Cheng, Z. Li, W. Fan, C. He, and Y. Mu, “Piecewise linear approximation of streaming time series data with max-error
guarantees,” in Proceedings of the IEEE International Conference on Data Engineering, 2015, pp. 173–184.

[15] E. O. Brigham and E. O. Brigham, The fast Fourier transform. Prentice-Hall Englewood Cliffs, NJ, 1974, vol. 7.

[16] C. S. Burrus, R. A. Gopinath, and H. Guo, “Introduction to wavelets and wavelet transforms: a primer,” 1997.

[17] M. W. LeChevallier, T. Evans, and R. J. Seidler, “Effect of turbidity on chlorination efficiency and bacterial persistence in drinking water.” Applied and environmental microbiology, vol. 42, no. 1, pp. 159–167,
1981.

[18] L. Monteiro, D. Figueiredo, S. Dias, R. Freitas, D. Covas, J. Menaia, and S. Coelho, “Modeling of chlorine decay in drinking water supply systems using epanet msx,” Procedia Engineering, vol. 70, pp. 1192–1200, 2014.

[19] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables,” Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 68, no. 1, pp. 49–67, 2006.

[20] W. Zhang, K. Zhang, P. Gu, and X. Xue, “Multi-view embedding learning for incompletely labeled data,” in Proceedings of the International Joint Conference on Artificial Intelligence, 2013, pp. 1910–1916.

[21] Y. Liu, L. Zhang, L. Nie, Y. Yan, and D. S. Rosenblum, “Fortune teller: Predicting your career path,” in Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 201–207.

[22] S. Sun, “A survey of multi-view machine learning,” Neural Computing and Applications, vol. 23, no. 7-8, pp. 2031–2038, 2013.

[23] C. Xu, D. Tao, and C. Xu, “A survey on multi-view learning,” arXiv preprint arXiv:1304.5634, 2013.

[24] J. Zhang and J. Huan, “Inductive multi-task learning with multiple view data,” in Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining, 2012, pp. 543–551.

B. RAJESH

20K81A0508

I am Bandar Rajesh, currently immersed in the pursuit of a bachelor's degree in technology at St. Martin's Engineering College, specializing in the field of computer science. My academic journey thus far has been characterized by an unwavering commitment to knowledge acquisition and skill refinement, as evidenced by my current CGPA of 7.41. Having completed my formative years of education at Vijay High School, I achieved an exemplary CGPA of 9.5, a testament to my relentless pursuit of scholastic excellence. Subsequently, I embarked upon my higher education at SR Junior College, where I triumphantly concluded my 12th grade, attaining an impressive percentage of 97.8. Fuelled by an ardent fascination for data science, I have garnered proficiency in diverse programming languages, including C, Java, and Python. With a sound understanding of data structures, I possess the aptitude to meticulously dissect and manipulate complex information. Furthermore, I have honed my skills in web development, proficiently navigating the realms of HTML, CSS, and JavaScript. Embracing a collaborative mindset, I relish engaging in synergistic endeavours that foster team cohesion and propel us towards collective triumph. During leisurely interludes, I derive immense gratification from immersing myself in the world of sports, with a particular penchant for cricket and basketball. The camaraderie, virtuosity, and strategic acumen exhibited in these athletic pursuits captivate my senses and serve as a wellspring of inspiration. Armed with a robust academic foundation, an unwavering ardour for data science, and an unwavering dedication to collaboration, I stand poised to confront novel challenges and make a profound impact in the realm of computer science.

G. SAI KUMAR

20K81A0520

I am Gadusandula Sai Kumar, a motivated student pursuing a bachelor's degree in computer science and engineering. My academic journey began at High Tech Modern Residential High School, where I completed my 10th grade education. Building upon that foundation, I enrolled in Sri Gayatri Junior College to pursue my intermediate studies. Programming languages have always fascinated me, and I have a particular interest in HTML, CSS, C, Java, and Python. Exploring these languages and their applications energizes me, and I am constantly expanding my knowledge and skills in these areas. I consider myself a quick learner, always seeking new challenges and opportunities to enhance my understanding of these programming languages. In addition to my academic pursuits, I am passionate about sports, with a special affinity for badminton. Engaging in sports not only helps me stay physically active but also teaches me the values of discipline, teamwork, and perseverance. I believe in the power of sports to foster personal growth and well-being. As I progress in my studies, my aim is to continually strive for excellence and stay updated with the latest advancements in the field of computer science and engineering. I am driven by a thirst for knowledge and a desire to make a positive impact in the world through my technological contributions.

A. SOUMITH REDDY

20K81A0503

I am Almawar Soumith Reddy, currently pursuing a bachelor's degree in computer science and engineering at St. Martin's Engineering College. My educational foundation was laid during my time at Vijaya High School. This early success motivated me to strive for excellence in my subsequent endeavours. Continuing my educational pursuit, I enrolled in Narayana Junior College for my intermediate studies. These educational experiences have equipped me with a well-rounded understanding of various subjects within the field of computer science and engineering. My true passion lies in the realm of programming. I have developed a deep interest in languages such as C, Java, Python, HTML, CSS and SQL. What excites me the most is the opportunity to utilize my expertise to invent and create new solutions. As a quick learner, I thrive on challenges and enjoy pushing the boundaries of what is possible. Beyond my academic pursuits, I am an avid musician. I am proficient in playing the drums and have also acquired skills in playing the guitar. Music serves as a creative outlet for me, allowing me to express myself and find solace in the harmonies I create. Engaging in music not only provides a balance to my life but also fuels my creativity and problem-solving abilities. Looking ahead, my ultimate goal is to make a significant impact in the field of computer science and engineering. In conclusion, I am poised to embark on a fulfilling and impactful career. I am eager to continue my learning journey, explore new horizons, and make a positive difference in the world through my pursuits.

R. DEEKSHITH GOUD

20K81A0549

I am Rangu Deekshith Goud, pursuing a bachelor's degree in computer science and engineering. My educational journey began at SR Prime School, where I completed my 10th grade education. I then moved on to SR Junior College to complete my intermediate studies. My primary areas of interest lie in programming languages such as C, Java, Python, HTML, and SQL. I find immense joy in exploring these languages and continuously expanding my knowledge and skills. Being a quick learner, I thrive on challenging myself and staying updated with the latest developments in the field. The ever-evolving nature of technology excites me, and I am driven to stay at the forefront of advancements in the industry. Aside from my academic pursuits, I am deeply passionate about bodybuilding and fitness. I believe in the importance of maintaining a healthy and balanced lifestyle. Through bodybuilding, I have not only honed my physical strength but also developed discipline, perseverance, and goal-setting abilities. The dedication I pour into my workouts mirrors my commitment to achieving excellence in all aspects of my life. With a solid foundation in computer science and engineering, combined with my passion for fitness, I consider myself a well-rounded individual. I possess the determination, drive, and hunger for knowledge required to succeed in the field. As I continue to pursue my studies, I am eager to make meaningful contributions to the world of technology and utilize my skills to create innovative solutions. I look forward to embracing new challenges, staying abreast of emerging technologies, and making a positive impact in the ever-evolving realm of technology.