CSE3999: Technical Answers for Real world Problems (TARP)

TG1 SLOT

Group 4

**Water related problems in and around VIT and their solutions**

**TARP FINAL REPORT**



|  |  |
| --- | --- |
| Pulkit Khurana | 15BCE0322 |
| M.S. Sanjay | 15BCE0517 |
| Venkat Ratnam Sabbavarapu | 15BCE0664 |
| Abharika Nayyer | 15BCE0779 |
| Adarsh Anand | 15BCE0896 |
| Shashwat Mishra | 15BCE0957 |

Lab Section: CSE3999

Workstation: VIT University

**TARP**

Project Management Form

1. ***Title of the Project***

Water related problems in and around VIT and its solutions

1. ***Group ID :*** *Group 4*
2. ***Team Leader:*** M.S. Sanjay (15BCE0517)

|  |  |
| --- | --- |
| 1. ***Team Members*** |  |
| M.S. Sanjay | 15BCE0517 |
| Pulkit Khurana | 15BCE0322 |
| Venkat Ratnam Sabbavarapu | 15BCE0664 |
| Abharika Nayyer | 15BCE0779 |
| Adarsh Anand | 15BCE0896 |
| Shashwat Mishra | 15BCE0957 |

***5. Scope Statement***

To analyse past rainfall patterns and predict future rainfall patterns using Machine Learning and suggest approximate water conservation measures. Also periodical checking of hardness of water is done by placing pH sensors at various locations and any anomaly is reported.

***6. -Roles and Responsibilities***

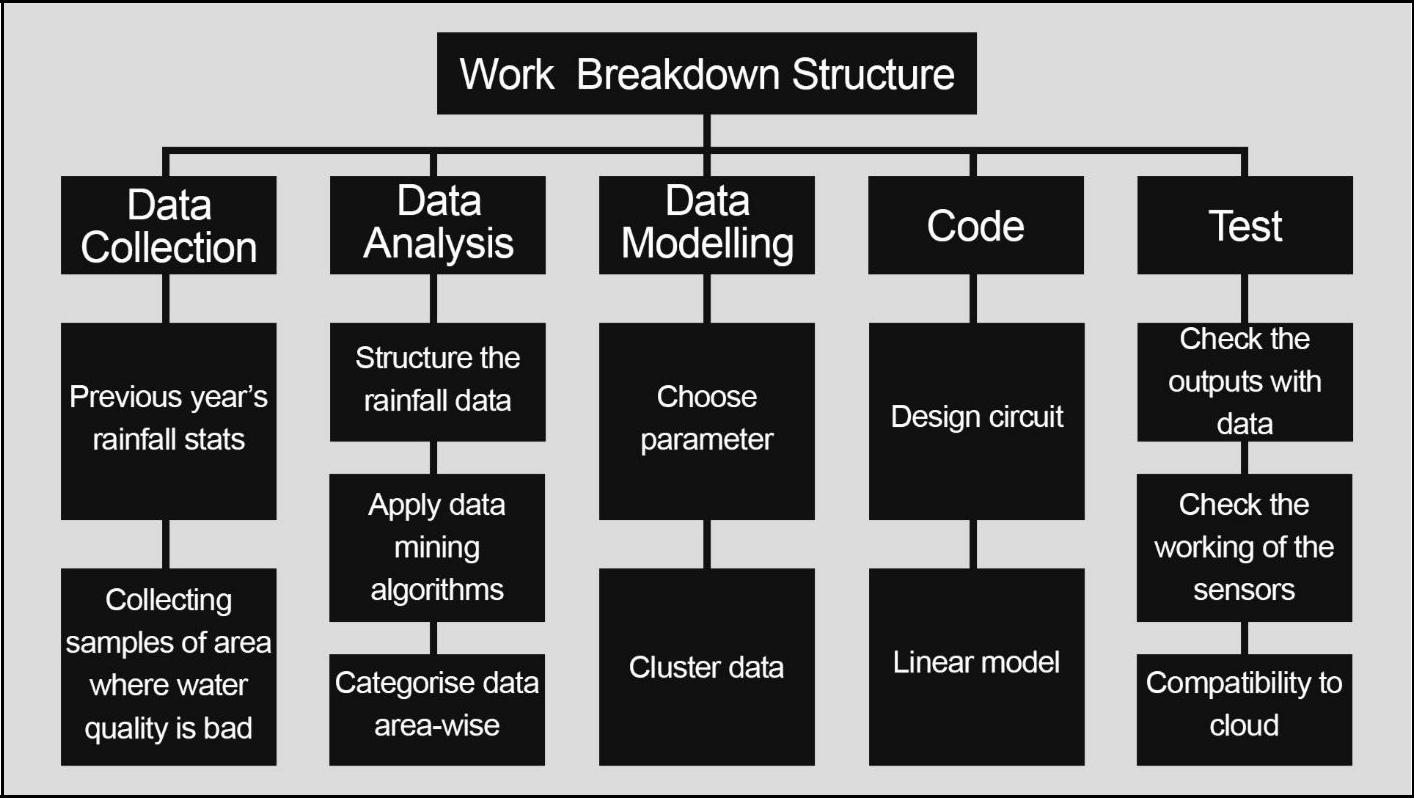
A. M.S. Sanjay, Pulkit and Venkat Ratnam:

Analyse and predict rainfall patterns and build and test different models.

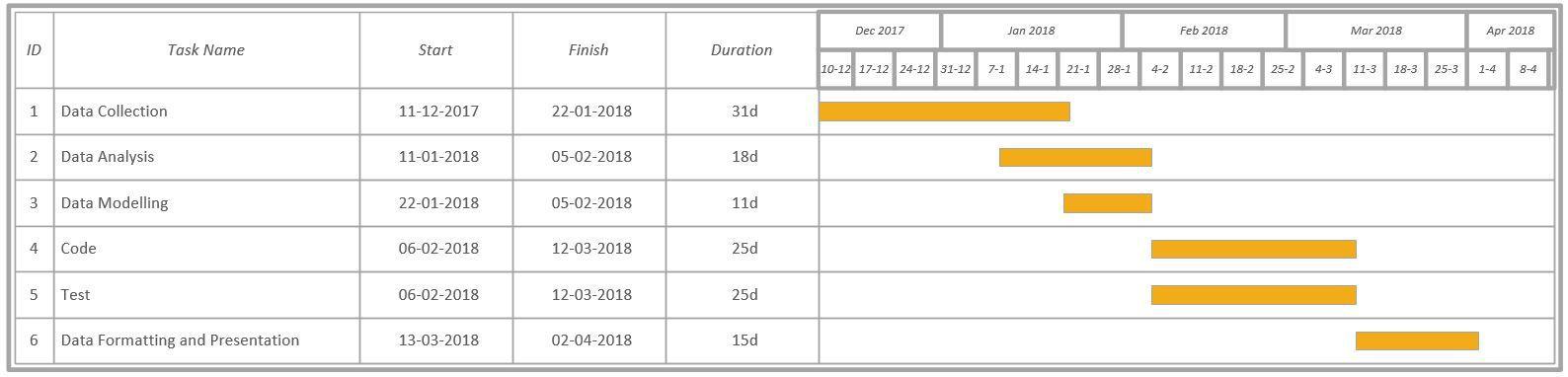
B. Abharika, Adarsh and Shashwat:

Analyse rainfall pattern and suggest suitable rainwater harvesting technique with focus on rural areas.

***7. Work Plan and Deliverables: [Work Breakdown Structure]***

******

8.GANTT CHART



CSE3999: Technical Answers for Real world Problems (TARP)

TG1 SLOT

Group 4

**Water related problems in and around VIT and their solutions**

**REQUIREMENTS ANALYSIS DOCUMENT**



|  |  |
| --- | --- |
| Pulkit Khurana | 15BCE0322 |
| M.S. Sanjay | 15BCE0517 |
| Venkat Ratnam Sabbavarapu | 15BCE0664 |
| Abharika Nayyer | 15BCE0779 |
| Adarsh Anand | 15BCE0896 |
| Shashwat Mishra | 15BCE0957 |

Lab Section: CSE3999

Workstation: VIT University

**INDEX**

**1.Introduction** 1

1.1 Purpose of the System 1

1.2 Scope of the System 1

1.3 Objectives and Success Criteria of the Project 1

1.4 Definitions, acronyms and abbreviations 1

1.5 References 2

1.6 Overview 2

**2. Current System** 2

**3.Proposed System** 4

3.1 Overview 4

3.2 Functional Requirements: 5

3.3 Non Functional Requirements: 6

**4. Glossary** 7

**1.Introduction**

**1.1 Purpose of the System**

The purpose of the system is to predict rainfall in the future using previous rainfall data and recommend appropriate water conservation techniques .

**1.2 Scope of the System**

The system aims to deliver accurate predictions about future rainfall with an accuracy of 95 percent. The region of Vellore, Tamil Nadu will be the main focus of the system. The rainfall data is collected for Vellore and will be used to predict rainfall accurately. Machine Learning Techniques will be using to predict rainfall for a given time period and with that data, appropriate conservation technique can be used.

**1.3 Objectives and Success Criteria of the Project**

The success criteria of the system is to achieve 95 % accuracy.

**1.4 Definitions, acronyms and abbreviations**

Machine learning:

Machine learning is a field of computer science that gives computer systems the ability to "learn" (i.e., progressively improve performance on a specific task) with data, without being explicitly programmed

Supervised learning:

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples

Unsupervised learning:

Unsupervised machine learning is the machine learning task of inferring a function to describe hidden structure from "unlabeled" data (a classification or categorization is not included in the observations). Since the examples given to the learner are unlabeled, there is no evaluation of the accuracy of the structure that is output by the relevant algorithm

Perceptron:

A computer model or computerized machine devised to represent or simulate the ability of the brain to recognize and discriminate.

**1.5 References**

[1] Indrabayu, Nadjamuddin Harun, M.Saleh Pallu and Andani Achmad,” A New Approach of Expert System for Rainfall Prediction Based on Data Series”, International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622, Vol. 3, Issue 2, March -April 2013, pp.1805-1809

[2] Somia A. Asklany , Khaled Elhelow , I.K. Youssef and M. Abd El-wahab ,” Rainfall events prediction using rule-based fuzzy inference system”,Elsevier, Atmospheric Research 101 (2011) 228–236

[3] Jyothis Joseph and Ratheesh T K,” Rainfall Prediction using Data Mining Techniques”, International Journal of Computer Applications (0975 –8887) Volume 83 – No 8, December 2013

[4] Jignesh Patel and Dr.Falguni Parekh,” Forecasting Rainfall Using Adaptive Neuro-Fuzzy Inference System (ANFIS)”, International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 3, Issue 6, June 2014 ,ISSN 2319 – 4847

[5] Akash D Dubey,” Artificial Neural Network Models for Rainfall Prediction in Pondicherry”, International Journal of Computer Applications (0975 – 8887) Volume 120 – No.3, June 2015

**1.6 Overview**

The main reason the system is built because the existing systems use a limited number of techniques to predict rainfall. While rainfall has a trend, one cannot be so sure what machine learning model predicts what trend .So we need to take into account different machine learning models and consider the average or most frequent rating to make sure the correct pattern is found.

**2. Current System**

1) Rainfall forecasting using Data mining - analysis of the rainfall forecasting is done by using the methods artificial intelligence, neural network, fuzzy sets and data mining in some journals. Artificial intelligence and neural network are more difficult compared to data mining because artificial intelligence involves some algorithms.

In Data Mining, some of the functionalities are used i.e. classification, clustering, regression or prediction etc. Using the classification we classify what is the reason for rainfall fall in the ground level. Using clustering technique, we grouping the element that is particular area occupied by the rainfall region.

In Prediction one has to predict the rainfall occurs in the particular region. Finally, we take the prediction methods in data mining because rainfall occupied in the region done by the regression approach. In regression we use Karl Pearson correlation Coefficient for finding how many centimeters rainfall fall in the particular region. We have to predict the rainfall fall in the future years by using the multiple linear regression approach.

2) Short-term rainfall forecasting based on GNSS-derived PWV - Water vapor is an important component of the atmosphere, which affects the radiation balance, energy transportation, and the formation of cloud and precipitation1. Water vapor has a very low content but a complex spatial distribution in atmosphere, and it plays an important role in atmospheric processes such as rain, adverse weather, and global climate variation.

Traditional atmosphere sensing techniques such as radiosonde and microwave radiometer (MWR) have shortcomings in reflecting the continuous transformations of atmospheric water vapor because of their low spatiotemporal resolution4.

Global navigation satellite system (GNSS) has several advantageous characteristics (i.e., all weather applications, high accuracy, and low cost), and it has been recognized as an efficient approach to estimate precipitable water vaper (PWV).

Global Navigation Satellite System (GNSS) can effectively retrieve precipitable water vapor (PWV) with high precision and high-temporal resolution. GNSS-derived PWV can be used to reflect water vapor variation in the process of strong convection weather. By studying the relationship between time-varying PWV and rainfall, it can be found that PWV contents increase sharply before raining. Therefore, a short-term rainfall forecasting method is proposed based on GNSS-derived PWV.

3) Forecasting Rainfall Using Adaptive NeoFuzzy Inference - It is a combination of two intelligence systems, namely ANN system and FIS system in such a way that the ANN learning algorithm is used to determine the parameters of the FIS. ANN is a non-linear statistical data-modelling tool, which can capture and model any input-output relationship (or can learn detect complex patterns in data). FIS (involves membership function (mf), fuzzy logic operator and if-then-rules) is the process of formulating the mapping from a given input to an output using fuzzy logic.

ANFIS is the fuzzy-logic based paradigm that grasps the learning abilities of ANN to enhance the intelligent system’s performance using knowledge gained after learning. Using a given input-output data set, ANFIS constructs a fuzzy inference system whose membership function parameters are tuned or adjusted using hybrid type of neural algorithms. The Rainfall forecasting is nonlinear system so ANFIS model has been developed with a view to predict the rainfall based on previous year data. Model has been designed, trained and tested with different membership functions and different number of members. After Defining ANFIS Model, it is run with various FIS Algorithm, error tolerance and number of epochs to analyse the effect of all these parameters on RMSE and predict rainfall. The parameter optimization is done in such a way during training session that the error between the target and the actual output is minimized.

4) Rainfall Prediction based on data series - The system combines the SVM and Fuzzy methods to achieve high accuracy of the rainfall prediction. A Support Vector Machine (SVM) is a computer algorithm that learns by example to find the best function of classifier /hyperplane to separate the two classes in the inputs pace. The SVM analysed two kinds of data, i.e. linearly and nonlinearly separable data. Fuzzy Logic is a type of reasoning based on the recognition that logical statements are not only true or false (white or black areas of probability) but can also range from “almost certain” to “very unlikely” (gray areas of probability). Fuzzy logic has proven to be particularly useful in expert system and other artificial intelligence applications.

**3.Proposed System**

**3.1 Overview**

The proposed system uses a combination of regression tree, linear regression, neural network, support vector machine and K nearest neighbor.

Regression tree:

A regression tree is built through a process known as binary recursive partitioning, which is an iterative process that splits the data into partitions or branches, and then continues splitting each partition into smaller groups as the method moves up each branch.

Initially, all records in the Training Set (pre-classified records that are used to determine the structure of the tree) are grouped into the same partition. The algorithm then begins allocating the data into the first two partitions or branches, using every possible binary split on every field. The algorithm selects the split that minimizes the sum of the squared deviations from the mean in the two separate partitions. This splitting rule is then applied to each of the new `nal node.

Linear regression:

In [statistics](https://en.wikipedia.org/wiki/Statistics), linear regression is a [linear](https://en.wikipedia.org/wiki/Linear) approach for modelling the relationship between a scalar [dependent variable](https://en.wikipedia.org/wiki/Dependent_variable) y and one or more [explanatory variables](https://en.wikipedia.org/wiki/Explanatory_variable) (or independent variables) denoted X. The case of one explanatory variable is called [simple linear regression](https://en.wikipedia.org/wiki/Simple_linear_regression). For more than one explanatory variable, the process is called multiple linear regression

Neural network

Artificial neural networks (ANNs) or connectionist systems are computing systems vaguely inspired by the biological neural networks that constitute animal brains. Such systems "learn" (i.e. progressively improve performance on) tasks by considering examples, generally without task-specific programming.

An ANN is based on a collection of connected units or nodes called artificial neurons (a simplified version of biological neurons in an animal brain). Each connection (a simplified version of a synapse) between artificial neurons can transmit a signal from one to another. The artificial neuron that receives the signal can process it and then signal artificial neurons connected to it.

Support vector machine

Support vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

K nearest neighbor:

k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression .In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression.

k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms.

Both for classification and regression, a useful technique can be to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones.

The system takes input a series of rainfall data and outputs a predicted rainfall value.

**3.2 Functional Requirements:**

1. The system should be able to take any number of argument from the user.

2. The system should produce the output in a definite time.

3. The system should give the user the freedom of choice for input.i.e. user can choose whatever date he wants.

4. The system should have the appropriate dataset.

5. The system should output the apt water conservation technique based on the predicted rainfall.

6. The system should prevent overfiting of data

7. The system should not tolerate high error rates.

**3.3 Non Functional Requirements:**

3.3.1Usability

As the system is easy to handle and navigates in the most expected way with no delays. In that case the system program reacts accordingly and transverses quickly between its states.

3.3.2 Reliability

As the system provide the right tools for discussion, problem solving it must be made sure that the system is reliable in its operations and for securing the sensitive details.

3.3.3 Performance

The system must be interactive and the delays involved must be less .So in every action-response of the system, there are no immediate delays. In case of opening windows forms, of popping error messages there is delay much below 2 seconds, In case of opening databases, evaluation there are no delays and the operation is performed in less than 2 seconds for opening ,sorting > 95% of the files.

3.3.4 Supportability

The system should support python 3.4 or above and libraries like scikitlearn, numpy and pandas.Any operating system is suitable.

3.3.5 Security

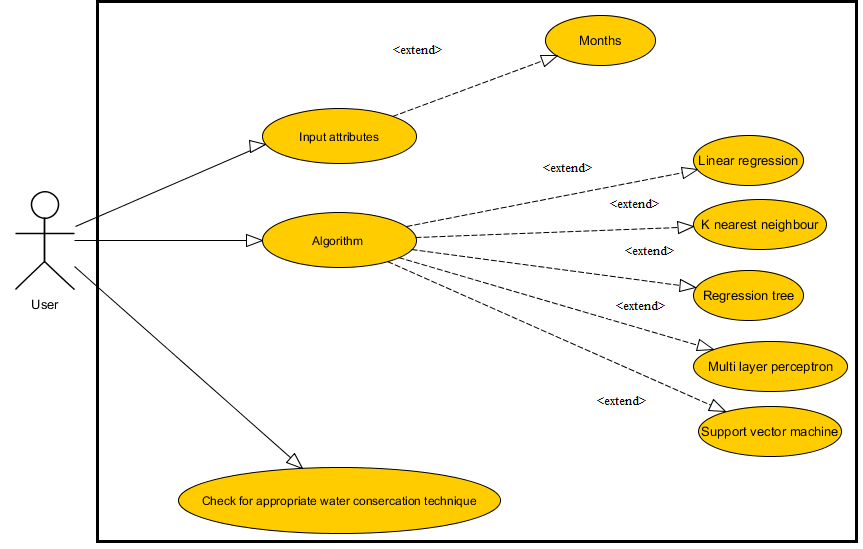
The main security concern is for users account hence proper login mechanism should be used to avoid hacking into database. Hence, security is provided from unwanted use of recognition software.

3.3.6 Availability

If the program halts in the middle or the Operating System hangs in the middle, then the program can be restarted.Even if the database is corrupted, then also it can be downloaded anytime, thus being available most of the time.

3.4 System models

**Use case diagram**



**4. Glossary**

* Linear Regression - In statistics, linear regression is a linear approach for modelling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted X.
* Neural Network - a computer system modelled on the human brain and nervous system.
* SVM - A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane.
* Regression tree-A regression tree is built through a process known as binary recursive partitioning, which is an iterative process that splits the data into partitions or branches, and then continues splitting each partition into smaller groups as the method moves up each branch.

CSE3999: Technical Answers for Real world Problems (TARP)

TG1 SLOT

Group 4

**Water related problems in and around VIT and their solutions**

**SOFTWARE DESIGN DOCUMENT**



|  |  |
| --- | --- |
| Pulkit Khurana | 15BCE0322 |
| M.S. Sanjay | 15BCE0517 |
| Venkat Ratnam Sabbavarapu | 15BCE0664 |
| Abharika Nayyer | 15BCE0779 |
| Adarsh Anand | 15BCE0896 |
| Shashwat Mishra | 15BCE0957 |

Lab Section: CSE3999

Workstation: VIT University

|  |  |  |  |
| --- | --- | --- | --- |
| **TABLE OF CONTENTS** | |  | |
| **1.I**[**NTRODUCTION**](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page4) | **1** | |
| 1.1 [Purpose](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page4) | 1 | |
| 1.2 [Scope](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page4) | 1 | |
| 1.3 Overview | 1 | |
| 1.4 Reference Material | 1 | |
| 1.5 [Definitions and Acronyms](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page4) | 2 | |
| **2.S**[**YSTEM**](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page4) **O**[**VERVIEW**](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page4) | **2** | |
| **3.S**[**YSTEM**](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page4) **A**[**RCHITECTURE**](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page4) | **3** | |
| 3.1 [Architectural Design](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page4) | 3 | |
| 3.2 [Decomposition Description](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page5) | 4 | |
| 3.3 [Design Rationale](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page5) | 4 | |
| **4. D**[**ATA**](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page5) **D**[**ESIGN**](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page5) | **4** | |
| 4.1 [Data Description](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page5) | 4 | |
| 4.2 [Data Dictionary](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page5) | 5 | |
| **5. C**[**OMPONENT**](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page5) **D**[**ESIGN**](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page5) | **5** | |
| **6. H**[**UMAN**](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page6) **I**[**NTERFACE**](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page6) **D**[**ESIGN**](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page6) | **8** | |
| 6.1 Overview of User Interface | 8 | |
| 6.2 [Screen Images](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page6) | [8](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page6) | |
| 6.3 [Screen Objects and Actions](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page6) | [9](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page6) | |
| **7.** [**REQUIREMENTS MATRIX**](file:///C:\Users\Lenovo\Downloads\sdd_template%20(1).doc#page6) | **9** | |

**1.INTRODUCTION**

**1.1Purpose**

The purpose of the system is to predict rainfall in the future using previous rainfall data and recommend appropriate water conservation techniques .

**1.2 Scope**

The system aims to deliver accurate predictions about future rainfall with an accuracy of 95 percent. The region of Vellore, Tamil Nadu will be the main focus of the system. The rainfall data is collected for Vellore and will be used to predict rainfall accurately. Machine Learning Techniques will be using to

**1.3. Overview**

This document will give a detailed description of the project, starting from the basic design to the methodology that is implemented. We have included all the necessary data required to understand the basic functioning of the topics with relevant illustrations.

**1.4 Reference Material**

[1] Indrabayu, Nadjamuddin Harun, M.Saleh Pallu and Andani Achmad,” A New Approach of Expert System for Rainfall Prediction Based on Data Series”, International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622, Vol. 3, Issue 2, March -April 2013, pp.1805-1809

[2] Somia A. Asklany , Khaled Elhelow , I.K. Youssef and M. Abd El-wahab ,” Rainfall events prediction using rule-based fuzzy inference system”,Elsevier, Atmospheric Research 101 (2011) 228–236

[3] Jyothis Joseph and Ratheesh T K,” Rainfall Prediction using Data Mining Techniques”, International Journal of Computer Applications (0975 –8887) Volume 83 – No 8, December 2013

[4] Jignesh Patel and Dr.Falguni Parekh,” Forecasting Rainfall Using Adaptive Neuro-Fuzzy Inference System (ANFIS)”, International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 3, Issue 6, June 2014 ,ISSN 2319 – 4847

[5] Akash D Dubey,” Artificial Neural Network Models for Rainfall Prediction in Pondicherry”, International Journal of Computer Applications (0975 – 8887) Volume 120 – No.3, June 2015

**1.5 Definitions and Acronyms**

* Linear Regression - In statistics, linear regression is a linear approach for modelling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted X.
* Neural Network - a computer system modelled on the human brain and nervous system.
* SVM - A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane.
* Regression tree-A regression tree is built through a process known as binary recursive partitioning, which is an iterative process that splits the data into partitions or branches, and then continues splitting each partition into smaller groups as the method moves up each branch.

1. **SYSTEM OVERVIEW**

The proposed system uses a combination of regression tree, linear regression, neural network, support vector machine and K nearest neighbor.

Regression tree:

A regression tree is built through a process known as binary recursive partitioning, which is an iterative process that splits the data into partitions or branches, and then continues splitting each partition into smaller groups as the method moves up each branch.

Initially, all records in the Training Set (pre-classified records that are used to determine the structure of the tree) are grouped into the same partition. The algorithm then begins allocating the data into the first two partitions or branches, using every possible binary split on every field. The algorithm selects the split that minimizes the sum of the squared deviations from the mean in the two separate partitions. This splitting rule is then applied to each of the new `nal node.

Linear regression:

In [statistics](https://en.wikipedia.org/wiki/Statistics), linear regression is a [linear](https://en.wikipedia.org/wiki/Linear) approach for modelling the relationship between a scalar [dependent variable](https://en.wikipedia.org/wiki/Dependent_variable) y and one or more [explanatory variables](https://en.wikipedia.org/wiki/Explanatory_variable) (or independent variables) denoted X. The case of one explanatory variable is called [simple linear regression](https://en.wikipedia.org/wiki/Simple_linear_regression). For more than one explanatory variable, the process is called multiple linear regression

Neural network

Artificial neural networks (ANNs) or connectionist systems are computing systems vaguely inspired by the biological neural networks that constitute animal brains. Such systems "learn" (i.e. progressively improve performance on) tasks by considering examples, generally without task-specific programming.

An ANN is based on a collection of connected units or nodes called artificial neurons (a simplified version of biological neurons in an animal brain). Each connection (a simplified version of a synapse) between artificial neurons can transmit a signal from one to another. The artificial neuron that receives the signal can process it and then signal artificial neurons connected to it.

Support vector machine

Support vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

K nearest neighbor:

k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression .In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression.

k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms.

Both for classification and regression, a useful technique can be to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones.

The system takes input a series of rainfall data and outputs a predicted rainfall value.

1. **SYSTEM ARCHITECTURE**

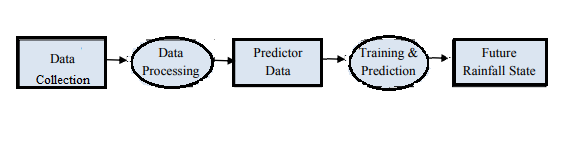
**3.1 Architectural Design**

PIPE AND FILTER ARCHITECTURE:

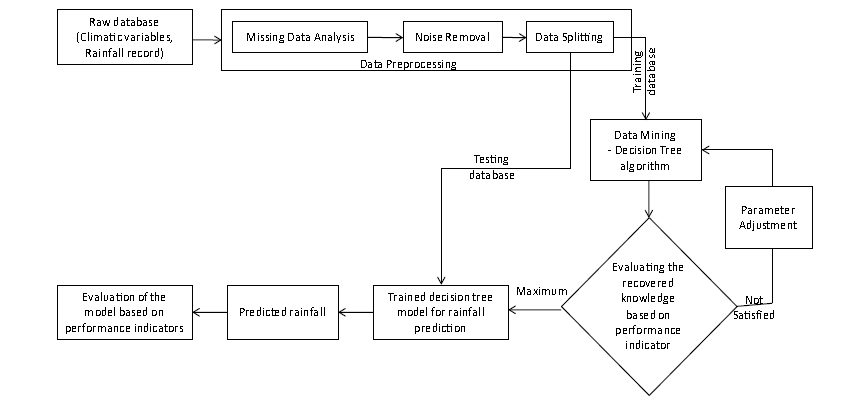
This is a model of the run-time organization of a system where functional transformations

process their inputs and produce outputs. Data flows from one to another and is transformed

as it moves through the sequence. Each processing step is implemented as a transform. Input data flows through these transforms until converted to output. The transformations may execute sequentially or in parallel. The data can be processed by each transform item by item or in a single batch.



**3.2 Decomposition Description**



**3.3 Design Rationale**

The pipe and filter model is chosen because in data processing, the data is transformed in each step,and the model is driven based on the manipulation of data.Thus the architecture model is appropriate.

1. **DATA DESIGN**

**4.1 Data Description**

The database consists of rainfall data collected over 100 years for each month of the year. Also weather data are grouped according to season for easy access to users and logical patterns.The initial data is loaded and preprocessed to feed into one of the five machine learning algorithms and depending on the ML algorithm, the data are transformed differently.

Ex. In a decision tree, data is classified into gropus based on entropy measures.In Multi Layer Perceptron, the data is transformed in each neuron to a Sigmoid value/tangent value.

In K nearest neighbour, the data is not at all transformed at all.

**4.2 Data Dictionary**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| YEAR | JAN | FEB | MAR | | APR | | MAY | | JUN | | JUL | | AUG | SEP | OCT |
| 1901 | 24.5 | 39.1 | 21.7 | | 36 | | 74 | | 41.8 | | 49.3 | | 67.9 | 191.1 | 122.3 |
| 1902 | 67.2 | 9.8 | 25.1 | | 21.9 | | 84.7 | | 39.3 | | 55.1 | | 113.8 | 98.6 | 282.2 |
| 1903 | 19.3 | 7.8 | 1.7 | | 18.2 | | 128.5 | | 58.5 | | 72.6 | | 115 | 210.4 | 128.1 |
| 1904 | 35.2 | 0.1 | 0.7 | | 19.5 | | 121.9 | | 34.9 | | 89 | | 40.4 | 85.7 | 163.2 |
| 1905 | 6.5 | 7.5 | 17.2 | | 64.8 | | 83.7 | | 49.8 | | 39 | | 101.8 | 73.5 | 250.4 |
| 1906 | 52.4 | 12.9 | 17 | | 8.5 | | 39.6 | | 43.6 | | 76 | | 195.2 | 65.3 | 162.8 |
| 1907 | 8.4 | 1.2 | 25 | | 78.9 | | 56.6 | | 46.1 | | 70.4 | | 58.2 | 117.2 | 160.4 |
| 1908 | 16.8 | 36.9 | 25 | | 24.5 | | 70.3 | | 34.2 | | 37.5 | | 48.4 | 154.2 | 261 |
| 1909 | 116.5 | 11.2 | 7.7 | | 68.6 | | 117.4 | | 30.8 | | 36.1 | | 206.6 | 106.1 | 158 |
| 1910 | 9.2 | 21.2 | 2.4 | | 26.8 | | 64.7 | | 51.5 | | 148.1 | | 142 | 52.4 | 278 |
| 1911 | 2.8 | 1.1 | 6 | | 30.8 | | 72.1 | | 70.7 | | 50.3 | | 28.9 | 120.5 | 125.6 |
| 1912 | 5.9 | 2.8 | 2.9 | | 18.5 | | 64.2 | | 46.7 | | 34.8 | | 74.2 | 126.7 | 244.2 |
| NOV | DEC | ANNUAL | | Jan-Feb | | Mar-May | | Jun-Sep | | Oct-Dec | |
| 212.3 | 80.4 | 960.3 | | 63.6 | | 131.6 | | 350.1 | | 415 | |
| 174.9 | 165.8 | 1138.2 | | 77 | | 131.7 | | 306.7 | | 622.9 | |
| 200.5 | 203.2 | 1163.9 | | 27.1 | | 148.4 | | 456.5 | | 531.9 | |
| 23.6 | 49.1 | 663.1 | | 35.3 | | 142.1 | | 249.9 | | 235.8 | |
| 123.7 | 3.2 | 821.1 | | 14 | | 165.7 | | 264.1 | | 377.2 | |
| 189.1 | 127.7 | 990.2 | | 65.4 | | 65.1 | | 380 | | 479.6 | |
| 200 | 63.2 | 885.6 | | 9.6 | | 160.5 | | 291.9 | | 423.6 | |
| 36.1 | 20.1 | 765 | | 53.7 | | 119.8 | | 274.3 | | 317.2 | |
| 74.7 | 20.8 | 954.6 | | 127.7 | | 193.8 | | 379.7 | | 253.6 | |
| 146.7 | 0.6 | 943.7 | | 30.4 | | 93.8 | | 394.2 | | 425.3 | |
| 176.2 | 130.1 | 814.9 | | 3.9 | | 108.9 | | 270.3 | | 431.9 | |
| 259.3 | 23.9 | 904.1 | | 8.7 | | 85.6 | | 282.4 | | 527.5 | |

Data type: All data are numerical in value.

Description: Along with monthly rainfall data, annual and seasonal data in constructed.The season are grouped into 4 categories: spring(jan-feb),summer(mar-may),autumn(jun-sep),winter(oct-dec)

1. **COMPONENT DESIGN**

**Regressiontree:**

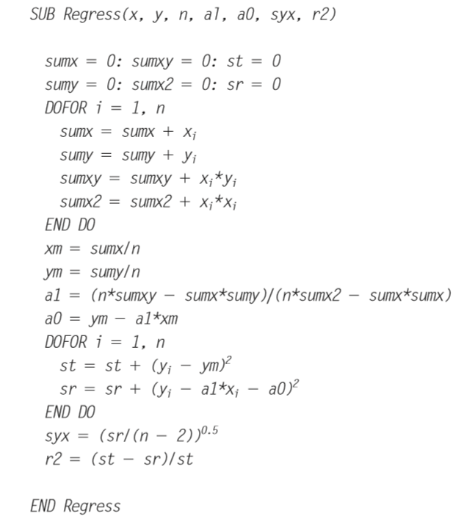
The basic regression-tree-growing algorithm then is as follows:

1. Start with a single node containing all points. Calculate mc(predictor for leaf) and S(sum of square).

2. If all the points in the node have the same value for all the independent variables, stop. Otherwise, search over all binary splits of all variables for the one which will reduce S as much as possible. If the largest decrease in S would be less than some threshold δ, or one of the resulting nodes would contain less than q points, stop. Otherwise, take that split, creating two new nodes.

3. In each new node, go back to step 1.

**Linear regression:**



**Multi Layer Perceptron:**

Given training set {(x(1),y(1))⋯(x(m),y(m))}

* Set Δ(l)i,j := 0 for all (l,i,j), (hence you end up having a matrix full of zeros)

For training example t =1 to m:

1. Set a(1):=x(t)
2. Perform forward propagation to compute a(l) for l=2,3,…,L

3. Using y(t), compute δ(L)=a(L)−y(t)

Where L is our total number of layers and a(L) is the vector of outputs of the activation units for the last layer. So our "error values" for the last layer are simply the differences of our actual results in the last layer and the correct outputs in y. To get the delta values of the layers before the last layer, we can use an equation that steps us back from right to left:

4. Compute δ(L−1),δ(L−2),…,δ(2) using δ(l)=((Θ(l))Tδ(l+1)) .∗ a(l) .∗ (1−a(l))

The delta values of layer l are calculated by multiplying the delta values in the next layer with the theta matrix of layer l. We then element-wise multiply that with a function called g', or g-prime, which is the derivative of the activation function g evaluated with the input values given by z(l).

The g-prime derivative terms can also be written out as:

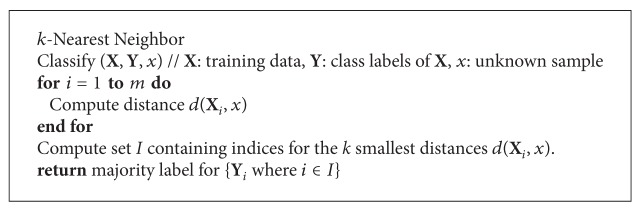
|  |
| --- |
| g′(z(l))=a(l) .∗ (1−a(l)) |

5. Δ(l)i,j:=Δ(l)i,j+a(l)jδ(l+1)i or with vectorization, Δ(l):=Δ(l)+δ(l+1)(a(l))T

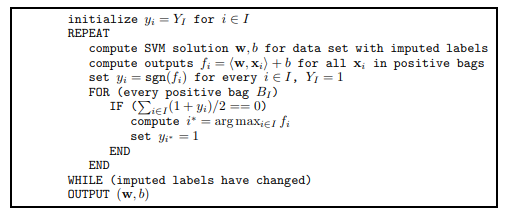
Hence we update our new Δ matrix.

* D(l)i,j:=1m(Δ(l)i,j+λΘ(l)i,j), if j≠0.
* D(l)i,j:=1mΔ(l)i,j If j=0

**K nearest neighbour:**



**Support vector machine:**



1. **HUMAN INTERFACE DESIGN**

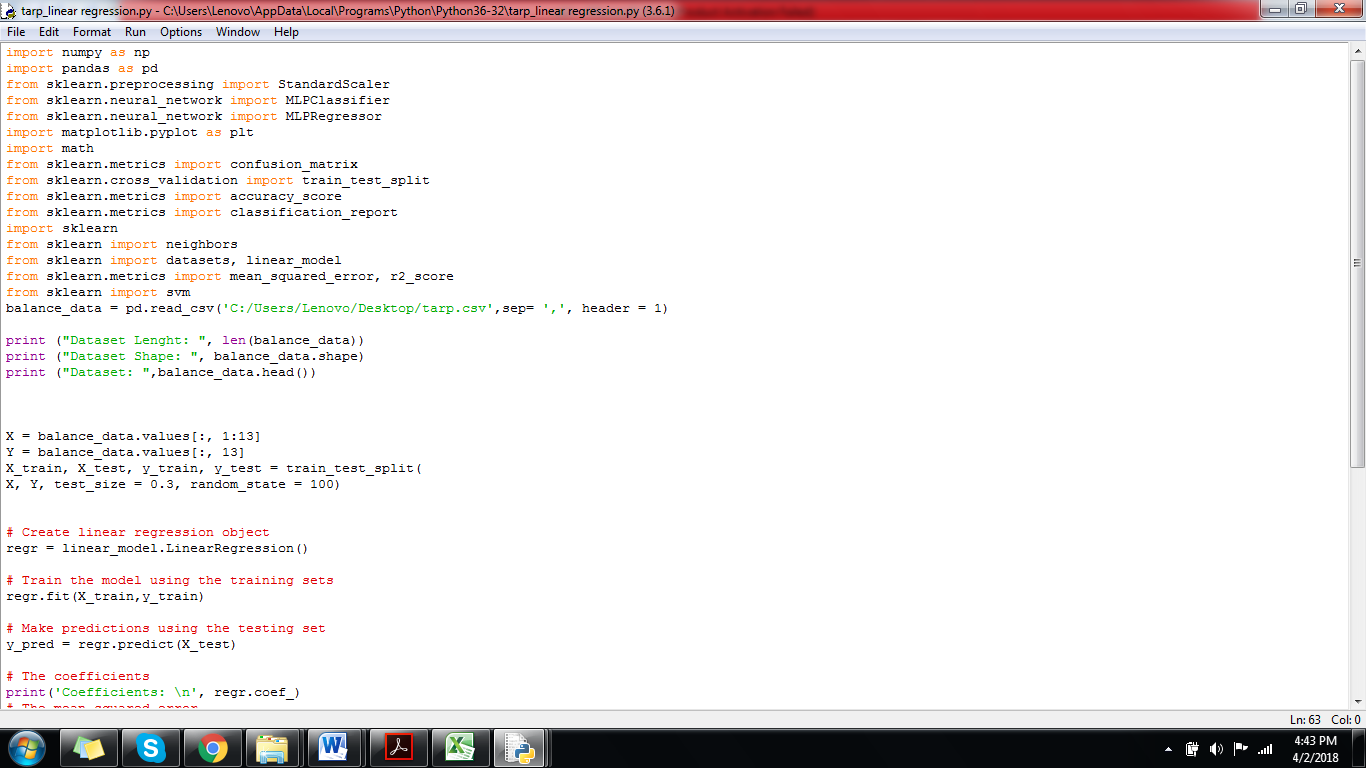
**6.1 Overview of User Interface**

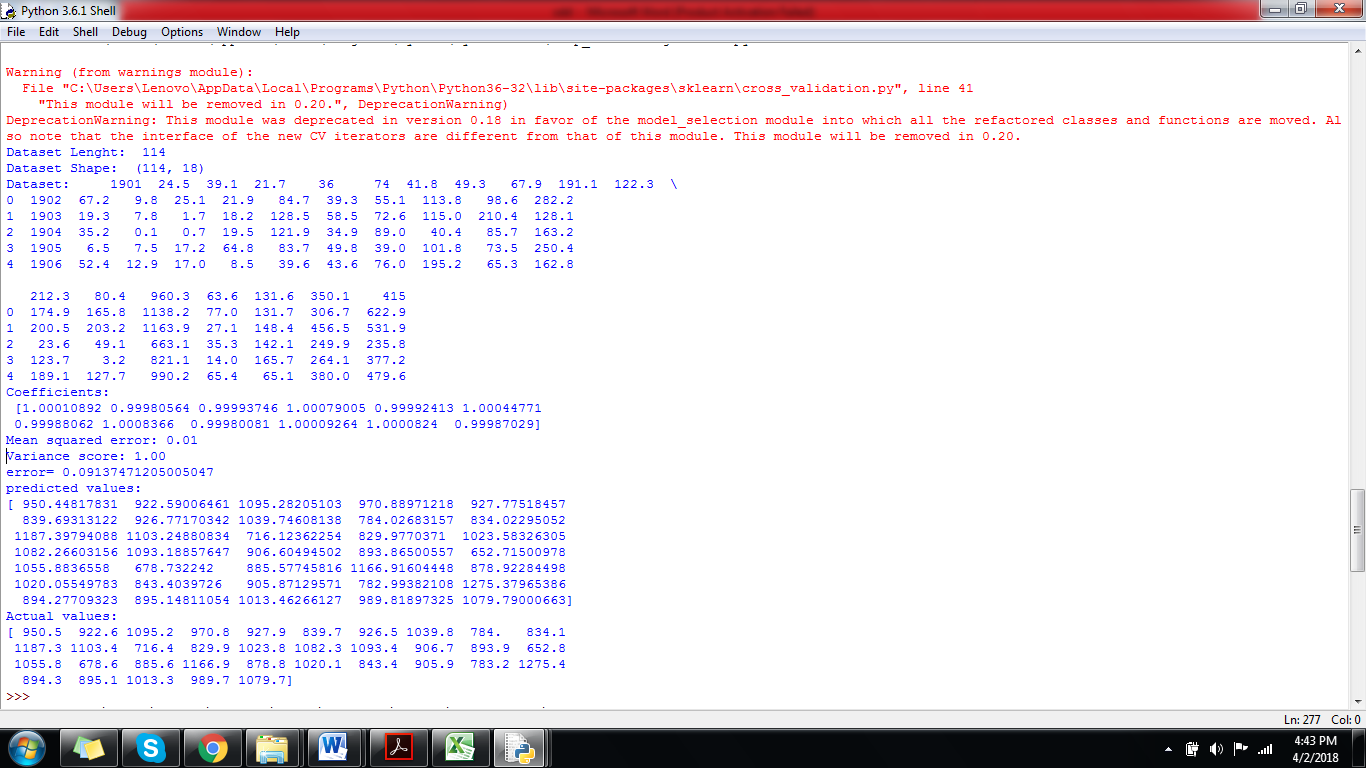
The user interface is very simple. The user just needs to choose the required fields for input and output and the algorithms will train based on these selected features.

The user can simply select the desired months of his choice just by typing the month number he wants.

The output is almost instantaneous because only small time is taken to train and fir the model parameters.

**6.2 Screen Images**





**6.3 Screen Objects and Actions**

The objects in the screen are python code only.The user just needs ot change the input and output arguments by just selecting the month number accordingly.Ex:If the user wants tor ain data of January to March month and wants to predict the output for April month, the input argument is 1:4 (1:4 returns 1,2,3 as output.Here 1 denotes January,2 denotes February and 3 denotes march) and output argument is 4(as 4 denotes April)

1. **REQUIREMENTS MATRIX**

|  |  |
| --- | --- |
| **ID** | **Functional Requirements** |
| R01 | The system should be able to take any number of arguments from the user |
| R02 | The system should produce the output in a definite time |
| R03 | The system should have the appropriate dataset |
| R04 | The system should output the apt water conservation technique based on the predicted rainfall |
| R05 | The system should prevent overfitting of data |
| R06 | The system should not tolerate high error rates |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Cases** | **R01** | **R02** | **R03** | **R04** | **R05** | **R06** |
| **T01** | X | X | X | X | X | X |
| **T02** | X | X |  |  |  |  |
| **T03** |  |  |  | X | X |  |
| **T04** |  |  |  |  | X | X |

T01: verify the user is able to interact with the system.

T02: Verify whether input and output are accessible.

T03:Verify whether data is available and usable.

T04:Verify whether system is accurate enough to provide a considerable output.

CSE3999: Technical Answers for Real world Problems (TARP)

TG1 SLOT

Group 4

**Water related problems in and around VIT and their solutions**

**IEEE TEST DOCUMENT**



|  |  |
| --- | --- |
| Pulkit Khurana | 15BCE0322 |
| M.S. Sanjay | 15BCE0517 |
| Venkat Ratnam Sabbavarapu | 15BCE0664 |
| Abharika Nayyer | 15BCE0779 |
| Adarsh Anand | 15BCE0896 |
| Shashwat Mishra | 15BCE0957 |

Lab Section: CSE3999

Workstation: VIT University

**INDEX**

**Test plan identifier: 1**

**Introduction: 1**

**Items To Be Tested: 1**

**Items Not To Be Tested: 1**

**Acceptance Test Approach: 1**

**Acceptance Test Process: 2**

**Item Pass/Fail Criteria 2**

**Suspension criteria and resumption requirements: 3**

**Test deliverables: 3**

**Testing tasks: 4**

**Environmental needs: 4**

**Staffing And Training Needs: 5**

**Schedule: 5**

**Risks And Contingencies: 7**

**Approvals: 9**

**Test plan identifier:**

**Introduction:**

This test document is for rainfall predictor software. The acceptance test verifies that the system works as required and validates that the correct functionality has been delivered. The ATP establishes the acceptance test framework used by the acceptance test team to plan, execute, and document acceptance testing. It describes the scope of the work performed and the approach taken to execute the tests created to validate that the system performs as required. The details of the ATP are developed according to the requirements specifications, and must show traceability back to those specifications.

The main scope of the system is to predict rainfall and suggest appropriate water conservation techniques.The system allows user to select any algorithm for predicting rainfall data.ALos the suer can train with the data of his choice and can choose what rainfall pattern to predict, whether seasonal or monthly etc.Based on the predicted values, different water conservation techniques can be appropriately suggested and used.

**ITEMS TO BE TESTED:**

Fitting of trained data: Chancs are there that the data that was trained was biased or did not fit properly.IN such cases a separate test data is used to check if there are any error when the algorithmsees new data.

Overfitting: Sometimes it is possible that the data is overtrained and the model becomes too perfect for the trained data and as a result the accuracy on trained dataset is low,but on testing the model on a new dataset, it is possible that the erro rate is high as the model is too much biased to the trained data and therefore unsuitable for practical purpose.To check for overfitting, we need cross validation set.

**ITEMS NOT TO BE TESTED:**

Algorithms:All the algorithms which are used Regression tree,linear regression,KNN,MLP and SVM need not be tested as it is clear that they are perfectly functioning.

Dataset: Since the dataset is obtained from Kaggle, we need not waste time in testing whether the data is original or not.

# ACCEPTANCE TEST APPROACH

There will be two stages of testing. The first will involve rerun of 75% of the developer's system tests using data supplied using the developer's data before client observers. Additional functional testing will be performed by end-users. Data will be supplied for testing the interface, and parallel processing will be performed for the legacy systems being replaced. Oversight group will observe security testing.

This approach is chosen because

Advantages:

* Ensure system software requrement based on requirements document which is developer’s own understanding of the requirements is actually what the client needs from the software.
* Requirements changes during the course of the project is communicated effectively to the developers.

# ACCEPTANCE TEST PROCESS

Activities:

* Analysis of Business Requirements
* Creation of UAT test plan
* Identify Test Scenarios
* Create UAT Test Cases
* Preparation of Test Data(Production like Data)
* Run the Test cases
* Record the Results
* Confirm business objectives

**ITEM PASS/FAIL CRITERIA**

T01: verify the user is able to interact with the system.

T02: Verify whether input and output are accessible.

T03:Verify whether data is available and usable.

T04:Verify whether system is accurate enough to provide a considerable output.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sno | Test case | Input | Expected output | Actual output | Pass/fail |
| 1 | T01 | Any user commands | Appropriate response from system | System responded correctly | Pass |
| 2 | T02 | Months that user wants to give as training parameter | Trained and tested output as well as tested data and error | System trained the data and responeded with predicted output,actual output and error | Pass |
| 3 | T03 | Dataset | Any processing with dataset | Data was opened and processed | Pass |
| 4 | T04 | Any ML algorithm(MLP) and any papameter | Low error rate defined by user | Error rate higher than what user epected | Fail |

**Suspension criteria and resumption requirements:**

The system is suspended when unexpected error like file not found,unable to open file,main memory unavailable, segmentation fault,unexpected indent or spacing,etc.. occurs.

The system is resumed when the unexpected errors ar fixed i.e. given correct input path file, change the file to correct format, clear main/cache memory, remove extra tabs and spaces and rearrange code properly.

**Test deliverables:**

The following is a list of deliverables produced during the acceptance test phase:

* Acceptance Test Plan
* Acceptance Test Schedule
* Traceability
* Acceptance Test Environment Inventory
* Acceptance Test Summary Report
* Acceptance Test Final Report

**Testing tasks:**

A successful acceptance test effort requires plannning. The acceptance test team identifies the tasks that need to be accomplished, including milestones. The functional requirements and SRS documents are the primary drivers for identifying those tasks.

The acceptance test schedule is the timeline of acceptance testing activities and deliverable due dates. For each acceptance testing effort, a test schedule is developed identifying the major test preparation, test execution, and test reporting activities, as well as providing interim checkpoints to measure the progress of acceptance testing. The client monitors the acceptance test effort.

Comprehensive acceptance test materials are a critical component of a successful acceptance test program. The acceptance test team uses a requirements-driven, structured approach to identify acceptance test data.

The test cases are grouped according to their functions.Each test case tests for a specific function. The test cases are all interface based and functional based. The Investment test case is functional based. The Balance test case is interface based. The transaction test case is both interface and functional based.

**Environmental needs:**

Hardware:

Pc

Software:

Pyhton 3.6 and above with installed packages of scikit learn,pandas,numpy

Any operating system

**Responsibilities:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Deliverable Name** | **Author** | **Reviewer** |
| 1. | Test Plan | Test Lead | Project Manager/ Business Analyst’s |
| 2. | Functional Test Cases | Test Team | Business Analyst’s Sign off |
| 3. | Logging Defects | Test Team | Test Lead/ Programming Lead |
| (4. | Daily/weekly status report | Test Team/ Test Lead | Test Lead/ Project Manager |
| 5. | Test Closure report | Test Lead | Project Manager |

**Staff and training needs:**

Staff:

No staff required team members enough.

Training:

Team members needs to be trained in software engineering first before proceeding to test documents.

**Schedule:**

| ***Activity*** | ***Planned***  ***Completion***  ***Date*** | ***Actual Completion Date*** | ***Deliverable/ Checkpoint*** |
| --- | --- | --- | --- |
| *Plan Acceptance Testing for CURRENCY TRADING* |  |  | *Preliminary Acceptance Test Schedule* |
| *Plan acceptance testing for Investment* |  |  |  |
| *Identify Test Materials* |  |  | *Preliminary Acceptance Test Matrix* |
| *Establish Acceptance Test Environment* | *15/12/17* | *20/12/17* | *Acceptance Test Environment Inventory* |
| *Execute Test* | *21/12/17* | *31/12/17* | *Draft Acceptance Test Plan Matrix*  *Completed Test Readiness Review Checklist* |
| *Plan acceptance testing for Balance* | *1/1/18* | *15/1/18* | *Acceptance Test Progress* |
| *Identify Test Materials* | *16/1/18* | *31/1/18* | *Preliminary Acceptance Test Matrix* |
| *Establish Acceptance Test Environment* | *1/2/18* | *18/2/18* | *Acceptance Test Environment Inventory* |
| *Conduct Acceptance Test Readiness Review* | *19/2/18* | *28/2/18* | *Draft Acceptance Test Plan Matrix*  *Completed Test Readiness Review Checklist* |
| *Execute Tests* | *1/3/18* | *15/3/18* | *Acceptance Test Progress* |
| *Complete Acceptance Testing* | *16/3/18* | *31/3/18* | *Acceptance Test Summary Report* |
| *Document Acceptance Testing* | *1/4/18* | *5/4/18* | *Final Acceptance Test Report* |

**Risks and contingencies:**

The following are typical, general overall acceptance test risks:

1. Insufficient test time –

* Risk: If the amount of time available is too short, the acceptance test team may not have enough time to complete acceptance testing or perform regression testing.
* Mitigation: Develop a critical path of tests, prioritized by importance.

1. Incomplete requirements -

* Risk: May result in insufficient testing of the system.
* Mitigation: Use the traceability matrices to track the testing/requirements relationship.

1. A test environment that is not the same as the production environment -

* Risk: May prevent the detection of some defects and issues.
* Mitigation: Note the differences and work to have them as close as possible.

| Risk | Prob. | Impact | Mitigation Plan |
| --- | --- | --- | --- |
| SCHEDULE Testing schedule is tight. If the start of the testing is delayed due to design tasks, the test cannot be extended beyond the UAT scheduled start date. | High | High | • The testing team can control the preparation tasks (in advance) and the early communication with involved parties.  • Some buffer has been added to the schedule for contingencies, although not as much as best practices advise. |
| RESOURCES Not enough resources, resources on boarding too late (process takes around 15 days. | Medium | High | Holidays and vacation have been estimated and built into the schedule; deviations from the estimation could derive in delays in the testing. |
| DEFECTS Defects are found at a late stage of the cycle or at a late cycle; defects discovered late are most likely be due to unclear specifications and are time consuming to resolve. | Medium | High | Defect management plan is in place to ensure prompt communication and fixing of issues. |
| SCOPE Scope completely defined | Medium | Medium | Scope is well defined but the changes are in the functionality are not yet finalized or keep on changing. |
| Natural disasters | Low | Medium | Teams and responsibilities have been spread to two different geographic areas. In a catastrophic event in one of the areas, there will resources in the other areas needed to continue (although at a slower pace) the testing activities. |
| Non-availability of Independent Test environment and accessibility | Medium | High | Due to non availability of the environment, the schedule gets impacted and will lead to delayed start of Test execution. |
| Delayed Testing Due To new Issues | Medium | High | During testing, there is a good chance that some “new” defects may be identified and may become an issue that will take time to resolve.  There are defects that can be raised during testing because of unclear document specification. These defects can yield to an issue that will need time to be resolved.  If these issues become showstoppers, it will greatly impact on the overall project schedule.  If new defects are discovered, the defect management and issue management procedures are in place to immediately provide a resolution. |

**Approvals:**

The test plan is to be approved by concerned team manager and faculty