VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgaum-590018

A PROJECT REPORT **(15CSP85)** ON

"Personalized Real Time Weather Prediction with recommendation"

Submitted in Partial fulfillment of the Requirements for the Degree of

Bachelor of Engineering in Computer Science & Engineering

By

Niveditha T(1CR16CS419)

Under the Guidance of,

ANJALI GUPTA

Assistant Professor, Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled **"Personalized Real Time Weather Prediction with recommendations"** carried out by **Ms. Niveditha T**, USN **1CR16CS419**, bonafide students of CMR Institute of Technology, in partial fulfillment for the award of **Bachelor of Engineering** in Computer Science and Engineering of the Visveswaraiah Technological University, Belgaum during the year 2019-2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Anjali Gupta	**Dr. Prem Kumar Ramesh**	**Dr. Sanjay Jain**
Assistant Professor	**Professor & Head**	**Principal**
Dept. of CSE, CMRIT	**Dept. of CSE, CMRIT**	**CMRIT**

External Viva

Name of the examiners — Signature with date

1. _____

2. _____

DECLARATION

I, the student of Computer Science and Engineering, CMR Institute of Technology, Bangalore declare that the work entitled **"Personalized Real Time Weather Prediction with recommendations"** has been successfully completed under the guidance of **Prof Anjali Gupta**, Computer Science and Engineering Department, CMR Institute of technology, Bangalore. This dissertation work is submitted in partial fulfillment of the requirements for the award of Degree of Bachelor of Engineering in Computer Science and Engineering during the academic year 2019 - 2020. Further the matter embodied in the project report has not been submitted previously by anybody for the award of any degree or diploma to any university.

Place:

Date:

Team members:

NIVEDITHA T (1CR16CS419) _____

ABSTRACT

Temperature forecasting and rain forecasting in today's environment is playing a major role in many fields like transportation, tour planning and agriculture. The purpose of this project is to provide a real time forecasting to the user according to their current position and requirement. The simplest method of forecasting the weather, persistence, relies upon today's conditions to forecast the conditions tomorrow i.e. analyzing historical data for predicting future weather conditions. The weather data used for the DM (Disaster Mitigation) research include daily temperature, daily pressure and monthly rainfall.

This system is a web application with effective graphical user interface. User will enter current temperature; humidity and wind, System will take this parameter and will predict weather from previous data in database. The role of the admin is to add previous weather data in database, so that system will calculate weather based on these data. Weather forecasting system takes parameters such as temperature, humidity, and wind and will forecast weather based on previous record therefore this prediction will prove reliable. This system can be used in Air Traffic, Marine, Agriculture, Forestry, Military, and Navy etc.

ACKNOWLEDGEMENT

I take this opportunity to express my sincere gratitude and respect to **CMR Institute of Technology, Bengaluru** for providing me a platform to pursue my studies and carry out my final year project

I have a great pleasure in expressing my deep sense of gratitude to **Dr. Sanjay Jain,** Principal, CMRIT, Bangalore, for his constant encouragement.

I would like to thank **Dr. Prem Kumar Ramesh,** Professor and Head, Department of Computer Science and Engineering, CMRIT, Bangalore, who has been a constant support and encouragement throughout the course of this project.

I consider it a privilege and honor to express my sincere gratitude to my guide

Anjali Gupta Assistant Professor, Department of Computer Science and Engineering, for the valuable guidance throughout the tenure of this review.

I also extend my thanks to all the faculty of Computer Science and Engineering who directly or indirectly encouraged me.

Finally, I would like to thank my parents and friends for all their moral support they have given me during the completion of this work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Relevance of the Project

Weather forecasting is the application of science and technology to predict the state of the atmosphere for a given location. Ancient weather forecasting methods usually relied on observed patterns of events, also termed pattern recognition. For example, it might be observed that if the sunset was particularly red, the following day often brought fair weather. However, not all of these predictions prove reliable. Here this system will predict weather based on parameters such as temperature, humidity and wind. This system is a web application with effective graphical user interface. User will login to the system using his user ID and password. User will enter current temperature; humidity and wind, System will take this parameter and will predict weather from previous data in database.

The role of the admin is to add previous weather data in database, so that system will calculate weather based on these data. Weather forecasting system takes parameters such as temperature, humidity, and wind and will forecast weather based on previous record therefore this prediction will prove reliable. This system can be used in Air Traffic, Marine, Agriculture, Forestry, Military, and Navy etc.

The project mainly focuses on forecasting weather conditions using historical data. This can be done by extracting knowledge from this given data by using techniques such as association, pattern recognition, nearest neighbour etc.

Disaster Mitigation: Predicting storms, floods, droughts

Helping those sectors which are most dependent on whether such as agriculture, aviation also depends on weather conditions. Forecasting the temperature and rain on a particular day and date is the main aim of this project. In the project we forecast rain and temperature for Europe; year up to 2051 and also, we forecast temperature of world; year up to 2100. Our project is aimed to provide real time weather forecast service at finest granularity level with recommendations. We grab user's location (longitude, latitude) using GPS data service whenever user requests for our services. Our system will process the users query and will mine the data from our repository to

draw appropriate results. Users will be provided with recommendations also and that is the key facility of our service.

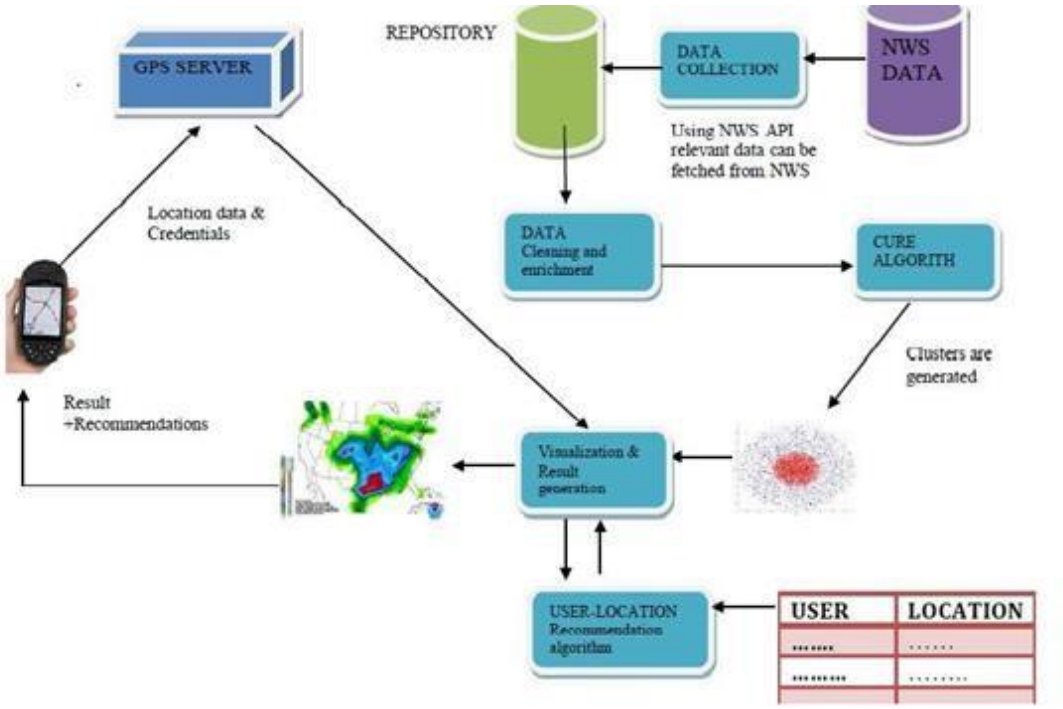


Fig 1.1 Personalized forecast is generated for each individual user based on their location.

1.2 DOMAIN INTRODUCTION

Weather forecasting is the application of science and technology to predict the state of the atmosphere for a given location. Ancient weather forecasting methods usually relied on observed patterns of events, also termed pattern recognition. For example, it might be observed that if the sunset was particularly red, the following day often brought fair weather. However, not all of these predictions prove reliable. Here this system will predict weather based on parameters such as temperature, humidity and wind. This system is a web application with effective graphical user interface. User will login to the system using his user ID and password. User will enter current temperature; humidity and wind, System will take this parameter and will predict weather from previous data in database. The role of the admin is to add previous weather data in database, so that system will calculate weather based on these data.

Weather forecasting system takes parameters such as temperature, humidity, and wind and will forecast weather based on previous record therefore this prediction will prove reliable. This system can be used in Air Traffic, Marine, Agriculture, Forestry, Military, and Navy etc. Forecasting the temperature and rain on a particular day and date is the main aim of this paper. In the paper we forecast rain and temperature for Europe; year up to 2051 and also we forecast temperature of world; year up to 2100.Our paper is aimed to provide real time weather forecast service at finest granularity level with recommendations. We grab user's location (longitude, latitude) using GPS data service whenever user requests for our services. Our system will process the users query and will mine the data from our repository to draw appropriate results. Users will be provided with recommendations also and that is the key facility of our service. Personalized forecast is generated for each individual user based on their location.

1.3 PROBLEM DEFINITION

System will take current parameters entered by the user and will compare the parameters with the data in database and will predict the Weather. The output is shown in the maps and also a bar graph that is plotted to show the predicted results. Advantages are User can easily find out Weather condition by using this system. The primary advantage of forecasting is that it provides the business with valuable information that the business can use to make decisions about the future of the organization. Disadvantages are Weather forecast by the system is not very accurate. Previous data is required by the system to forecast weather.

1.4 OBJECTIVES

Forecasting the temperature and rain on a particular day and date is the main aim of this paper. In the paper we forecast rain and temperature for Europe; year up to 2051 and also we forecast temperature of world; year up to 2100.Our paper is aimed to provide real time weather forecast service at finest granularity level with recommendations. We grab user's location (longitude, latitude) using GPS data service whenever user requests for our services. Our system will process the users query and will mine the data from our repository to draw appropriate results. Users will be provided with recommendations also and that is the key facility of our service. Personalized forecast is generated for each individual user based on their location.

1.5 SCOPE OF THE PROJECT

The project mainly focuses on forecasting weather conditions using historical data. This can be done by extracting knowledge from this given data by using techniques such as association, pattern recognition, nearest neighbour etc.

Disaster Mitigation: Predicting storms, floods, droughts

Helping those sectors which are most dependent on weather such as agriculture, aviation also depends on weather conditions.

1.6 Agile Methodology

Story ID	Requirement description	User stories/Task	Description
Requirement	Gathering project ideas to work on.	1. Find if there already exist implementations of the chosen project idea. 2. If implementation exists, study the existing implementation. 3. Based on the study, arrive at the missing necessary features we can build.	Collect data and ideas from various sources to find a potential project
Planning	Prepare feature list. Technologies to be used. Estimate effort.	Decide on the important functionalities we can add to our implementation	Planning is process which embraces a no of steps to be taken

Development	High level API class design.	Decide and arrive at an overview of modules and classes.	Coding basic classes and class structures.
Test Cases	Further coding and testing.	Implement different features for the UI and make sure they are working properly.	Information and research on the algorithms Used for clustering and classification of data

1.7 LIMITATIONS

Our project has the following limitations: -

1. In this system the customer will not select seat number. The authorized people in this process will be the employee who works at the cinema.
2. In this system the customer can book the tickets only for the present day. Advanced booking for the upcoming movies (2 or 3 months before the movie is released) is not included but can be included later.
3. Also, a printing system will not be in this project. In future, a ticket printing system can be integrated.
4. The customers who have booked the tickets must reach the multiplex 30 minutes prior to the show time or else ticket will get cancelled.

CHAPTER 2

LITERATURE SURVEY

Customer churn management, as a part of CRM, has become a major concern. In mobile telecommunications, the term “churn” refers to the loss of subscribers who switch from one provider to another during a given period. Based on an earlier study [3], the estimated average churn rate for mobile telecommunications is about 2.2% per month. This means that one in fifty subscribers of a given company discontinues their services every month. As it is more profitable to retain existing customers than to constantly attract new customers [4][6], it is crucial to build an accurate churn prediction model for identifying those customers who are most prone to churn. Established literature on customer churn uses various data mining technologies, such as neural networks [7], clustering [8], decision tree [7], [9], regression [10], [11], support vector machine [4], [12], and ensemble of hybrid methods [13], to provide more accurate predictions. According to a review on customer churn prediction modelling [14], Regression is the most commonly adopted technique, probably because of its high reported accuracy and interpretability for understanding key drivers, as well as for providing information to set up retention actions.

As the churner usually takes only a fraction of the customer base, the problem of customer churn prediction is always combined with the problem of highly skewed class distribution or lack of churner data. One of the most common techniques for dealing with rarity is sampling [15].Methods that adopt the sampling technique alter the distribution of training examples and generate balanced training set(s) for building churn prediction model(s) [8], [9], [12], [13]. However, a recent study [11] on the class imbalance issue in churn prediction reveals that advanced sampling technique does not increase predictive performance.

Although the weighted Random Forests technique is suggested [11], tree ensembles, such as Random Forests, are often criticized for being hard to interpret [16],i.e., it is difficult to identify risk factors which can be addressed by the retention process to prevent a customer from leaving, thus they are not the preferred methods in this study.

Other research explores the power of new features for churn prediction, such as social network and text information of customer complaints, which is beyond the scope of this paper.

The established literature only uses boosting as a general method to boost accuracy, and few researchers have ever tried to take advantage of the weight assigned by boosting algorithms. The weight also provides important information, specifically, outliers.

2.1 Data mining

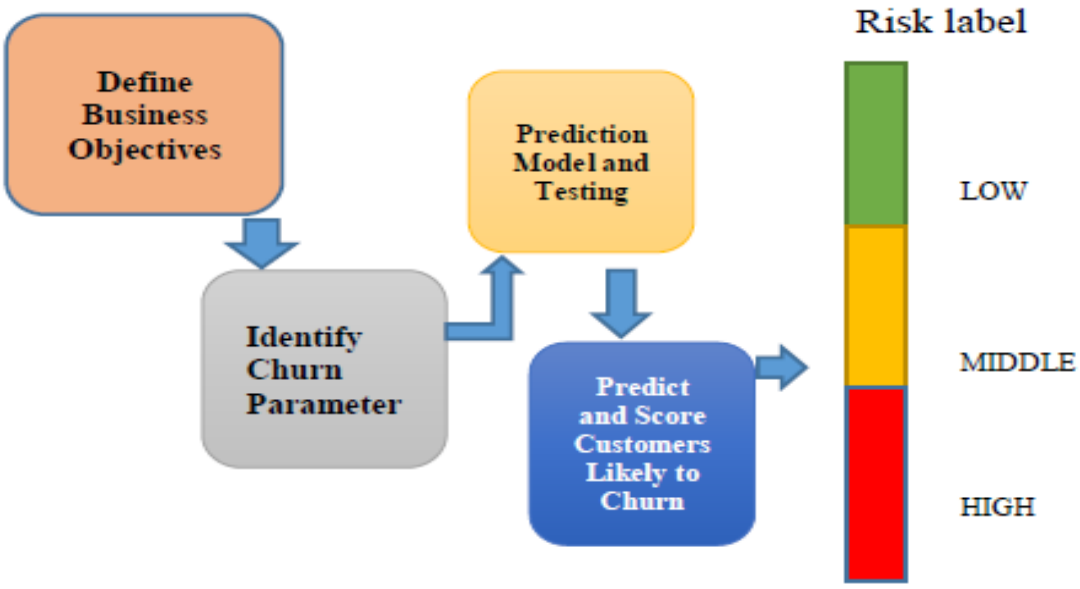


Fig 1.1: Data mining Diagram

This section explains how the Ensemble based Classifiers and well-known Base Classifiers were used in the Churn Prediction Model.

A. Decision Tree

Decision Tree was developed to overcome the drawbacks of ID3 algorithm. C4.5 utilizes the benefits of greedy approach and uses a series of rules for classification. Although this approach gives a high classification accuracy rate it fails to respond to

noisy data. Gain is the main metric used in the decision tree to decide the root node attribute.

B. Naive Bayes

Naïve Bayes is a brute-force method for training the model. The underlying principle behind Naïve Bayesian classifier is Bayes Theorem. For the classification problem, each predictor attribute was considered separately with class label for model construction using training dataset.

Predictor attribute includes the area, service calls, evening calls night calls etc. Apply the conditional probability for each attribute belongs to all the predictor attributes given that class label represents churn. The disadvantage of this methods is, it is not suited for the large dataset.

C. Support Vector Machine

SVM algorithm was proposed by **Boser, Guyon, and Vapnik.** It was very well used for both classification and regression problem. SVM maps all the data points to a higher dimensional plane to make the data points linear separable. The plane which divides data points is known as hyper plane. It can be used for small dataset to give an optimal solution. SVM cannot be more effective for noisy data. SVM model tries to find out the churn and non-churn customer. In order to divide the dataset into churner and non-churner group, first it will take all the data points in n d*imensional* plane and divide the data points into churner and non-churner group based on maximum marginal hyper plane. Based on the maximum marginal hyper plane it will divide the data points into churner and non-churner group. Here *n* represents the number of predictor variable associated with the dataset.

D. Bagging

Bagging (or Bootstrap aggregation) is one of the Ensemble based Classifiers which consist of bag of similar type or dissimilar type base classifiers. Bagging algorithm helps to reduce the variance of the classifier used for the Churn Prediction Model in order to increase the performance. The steps in designing of Churn Prediction Model

using Bagging algorithm are as follows, First, it is required to divide the input dataset into *k* subset with replacement, then it requires to train the model by using the *(k-1)* subset and test the model using the dataset which has not been used for training model. The experimental results showed that, Bagging is effective, because it predicts the test instances using the classifier which has more accuracy from the bag of classifier, Bagging requires heavier computational resource for the Model construction.

E. Boosting

Boosting Ensemble technique is designed in such a way that it will maintain a weight for each training tuple. After a classifier is learned from the training tuple, weights are updated for the subsequent classifier. The final Boosted Classifier combines the vote of each individual classifier for prediction to improve the performance of the classifier. In similar to SVM, this model is also not suited for noisy data. The key idea for the customer Churn Prediction using Boosting algorithm is to train a series of classifier simultaneously and keep updating the model accuracy for improving the performance of the classifier.

F. Random Forest

Random forest (**Breiman 2001)** works based on the random subspace method. The designed strategy used in Random Forest is divide and conquer. It forms number of Decision Trees and each Decision Tree is trained by selecting any random subset of attribute from the whole predictor attribute set. Each tree will grow up to maximum extent based on the attribute present in the subset. Then after, based on average or weighted average method, the final Decision Tree will be constructed for the prediction of the test dataset. Random forest runs efficiently in large dataset.

2.2 EXISTING SYSTEM

Preparation of All India weather bulletins that includes forecast & warning with graphics along with text up to 3 days for 36 sub-divisions and outlook for subsequent 4 days. It has been issued four times in a day and disseminated to various users like Door Darshan, All India Radio, Press Information Bureau, National Disaster Management Authority, National Disaster Management, Ministry of Home affairs, Ministry of Agriculture etc. This is also post in IMD Website for public in general.

a. Preparation of 5 days forecast of different cities at All India Level, to preparation of this forecast current synoptic observations, satellite imageries and numerical Weather Prediction model outputs taken into consideration.

b. Preparation of Press Releases/Reports on severe weather events. During monsoon season, weekly press release with detail weather of the week and forecast & warning for one week.

c. Monitoring of southwest and northeast monsoon onset, advance, performance and its withdrawal.

d. Co-ordination with sub offices of IMD regarding forecast & warning through video/audio conferences.

e. Collection of meteorological parameters like temperature, rainfall, snowfall, humidity etc. of major cities for internal use and for media briefing etc.

f. To provide the operational forecasting training to national and international forecasters. Also impart internship training to various colleges and university students. In addition, also provide briefing to school/college/university students, disaster managers and other various national and international delegates.

2.3 PROPOSED SYSTEM

The system is built on windows 2007 operating system. The system uses advanced java technology along with machine learning concepts. MySQL is used for storing data. This system uses three-tier architecture. The web service layer provides the android user to rate movies, view similar recommendations given by the system and comment on it. The proposed system is a better system than any other existing systems. This system has added the positive features of existing systems and has overcome the drawbacks of existing systems. The system uses all the existing algorithms i.e. content based, context based and collaborative based algorithms. All these algorithms are combined to give more precise result. The following modules are developed as: A. Admin The system admin will add movie in a database, view movies and update it. B. Recommendation Engine This recommendation engine will calculate the similarities between the different users. On the basis of that similarities calculated, this engine will

recommend movie to a user. C. Movie Web Service This will allow user to rate movies, comments on movies. This service will also show the movie recommendation to the users. D. Android User The android user can rate a movie, can comment on any movie, and can see similar movies recommended by other users who are similar to this user.

2.4 METHODOLOGY

The process we have briefed in earlier section can be depicted pictorially and which is self-explanatory.

We can divide our process in two modules namely:

1) Weather Mining

2) Recommendation

B. Weather Mining:

Data collection: We have collected weather data from WORLD DATA CENTER for climate, Hamburg. We have decided to use NWS API for data collection in future.

Data formatting and cleaning: We have converted our data from .NC (netcdf) format to .CSV (comma-separated values) format because WEKA supports .CSV format.

Clustering: Using WEKA, we have performed clustering on weather data to draw inferences.

Recommendation: We have planned to use recommendation algorithm as user to location collaborative algorithm similar to user to item collaborative algorithm. This algorithm uses user location (N*M) metrics.

2.5. MODULES

Login: - Admin will login to the system using ID and Password.

User Subscription: - User can access the system using his user ID and password. User can register himself by filling registration form.

Data Collection: - Admin will add previous weather data in database. which is used for clustering the data based on which the prediction is done

Weather Forecast: - In this module, system will take current parameters entered by the user and will compare the parameters with the data in database and will predict the weather.

Visualization of predicted result: This gives a picturized representation of the predicted weather in the form of maps and graphs.

2.6. PRODUCT PERCEPTIVE

User will login to the system using his user ID and password. User will choose a location for which the weather must predicted System will take this parameter and will predict weather from previous data in database**.**

The project mainly focuses on forecasting weather conditions using historical data (Dynamic data). This can be done by extracting knowledge from this given data by using techniques such as association, pattern recognition, nearest neighbour etc.

Disaster Mitigation: Predicting storms, floods, droughts helping those sectors which are most dependent on weather such as agriculture, etc.

2.7. DESIGN DESCRIPTION

Admin: Is a super user who has unlimited privileges. He/she can add a place, delete a place, view a place, view predicted place. He/she has control over what an average user can physically see on their screens therefore admins can also make changes to the website layouts.

User: Is an account type who will have limited privileges. He/she can search for a place, check the weather, and view the place for which we want the weather prediction.

Weather Forecasting system: This is a advanced system that uses various machine learning techniques and the WEKA tool (which will be mainly used for data mining) to predict the weather for the next day.

2.8. DESIGN APPROACH

Model-View-Controller

MVC is an application design model comprised of three interconnected parts. They include the

- Model (data),
- View (user interface).
- Controller (processes that handle input).
- Commonly used for developing modern user interfaces.
- It is provides the fundamental pieces for designing a programs for web applications

It works well with object-oriented programming, since the different models, views, and controllers can be treated as objects and reused within an application.

2.8.1 Advantages Of MVC

- Faster Web Application Development Process: .
- MVC Web Application Supports Asynchronous Technique
- Offers the Multiple Views:
- Ideal for developing large size web application: ...
- MVC Model Returns the Data Without the Need of Formattingg.

2.8.2 Disadvantage of MVC

- Difficulty of using MVC with modern user interface
- Knowledge on multiple technologies is required.
- Developer have knowledge of client-side code and html code
- Need multiple programmers.

2.8.3 RISKS

Design Constraints, Assumptions and Dependencies

The working of the application has a limitation to take inputs as few previous days' data with the current day and predict the next day's climate.

User constraints: Should have to be a registered user. Access to redirect and allows to use the location that is available in out server only. Internet connection to use the application.

Usage Limitations: The location can be chosen from the list of places the application is bound to. The specifications to the range of each location is set by the administrator not the user.

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATION.

3.1 Functional Requirements

In software engineering, a functional requirement defines a function of the software system or its components. A function is described as a set of inputs, the behavior and outputs. Functional requirements maybe calculations, technical details, data manipulation and processing and other specific functionality that define what a system must accomplish. A functional requirement defines a function of a software system or its components. It captures the intended behavior of the system. This behavior maybe expressed in terms of services, tasks or functions that the system has to perform.

3.2 Non-Functional Requirements

Non-Functional Requirements specify the criteria that can be used to judge the operation of a system, rather than specific behaviours. They are the metrics that are considered to measure the performance of the developed system. The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates.

Three key considerations involved in the feasibility analysis are

- **Usability:** Simple is the key here. The system must be simple that people like to use it, but not so complex that people avoid using it. The user must be familiar with the user interfaces and should not have problems in migrating to a new system with a new environment. The menus, buttons and dialog boxes should be named in a manner that they provide clear understanding of the functionality.
- **Reliability:** The system should be trustworthy and reliable in providing the functionalities.
- **Performance:** The system is going to be used by many people simultaneously. Performance becomes a major concern. The system should not succumb when many users would be using it simultaneously. It should allow fast accessibility to all of its users.

- **Scalability:** The system should be scalable enough to add new functionalities at a later stage. There should be a common channel, which can accommodate the new functionalities.
- **Maintainability:** The system monitoring and maintenance should be simple and objective in its approach.
- **Portability:** The system should be easily portable to another system.
- **Reusability:** The system should be divided into such modules that it could be used as a part of another system without requiring much of work.
- **Security**: Security is a major concern .This system must not allow unauthorized users to access the information of other users.

3.3 DOMAIN AND UI REQUIREMENTS

- **Admin Login: -** Admin will login to the system using Admin ID and Password.
- **Add previous weather data: -** Admin will add previous weather data in database.
- **User Login: -** User can access the system using his user ID and password.
- **User Registration: -** User can register himself by filling registration form.
- **Input Parameters: -** In this module user will enter parameters such as temperature, humidity and wind.
- **Weather Forecast: -** In this module, system will take current parameters entered by the user and will compare the parameters with the data in database and will predict the weather.

3.4. HARDWARE REQUIREMENTS

The hardware requirements listed below are almost in a significantly higher level which represents the ideal situations to run the system. Following are the system hardware requirements used :

Processor	-	Pentium –III
Speed	-	1.1 GHz
RAM	-	256 MB (min)
Hard Disk	-	20 GB
Floppy Drive	-	1.44 MB
Key Board	-	Standard Windows Keyboard
Mouse	-	Two or Three Button Mouse
Monitor	-	SVGA

3.5. SOFTWARE REQUIREMENTS

A major element in building a system is a section of compatible software since the software in the market is experiencing in geometric progression. Selected software should be acceptable by the firm and the user as well as it should be feasible for the system. This document gives the detailed description of the software requirements specification. The study of requirement specification is focused specially on the functioning of the system. It allows the analyst to understand the system, functions to be carried out and the performance level which has to be maintained including the interfaces established.

Operating System	-	Windows
Application Server	-	Tomcat5.0/6.X
Front End	-	HTML, Java, Jsp
Scripts	-	JavaScript.

Server-side Script	-	Java Server Pages.
Database	-	MySQL
Database Connectivity	-	JDBC.

3.6. DATA REQUIREMENTS

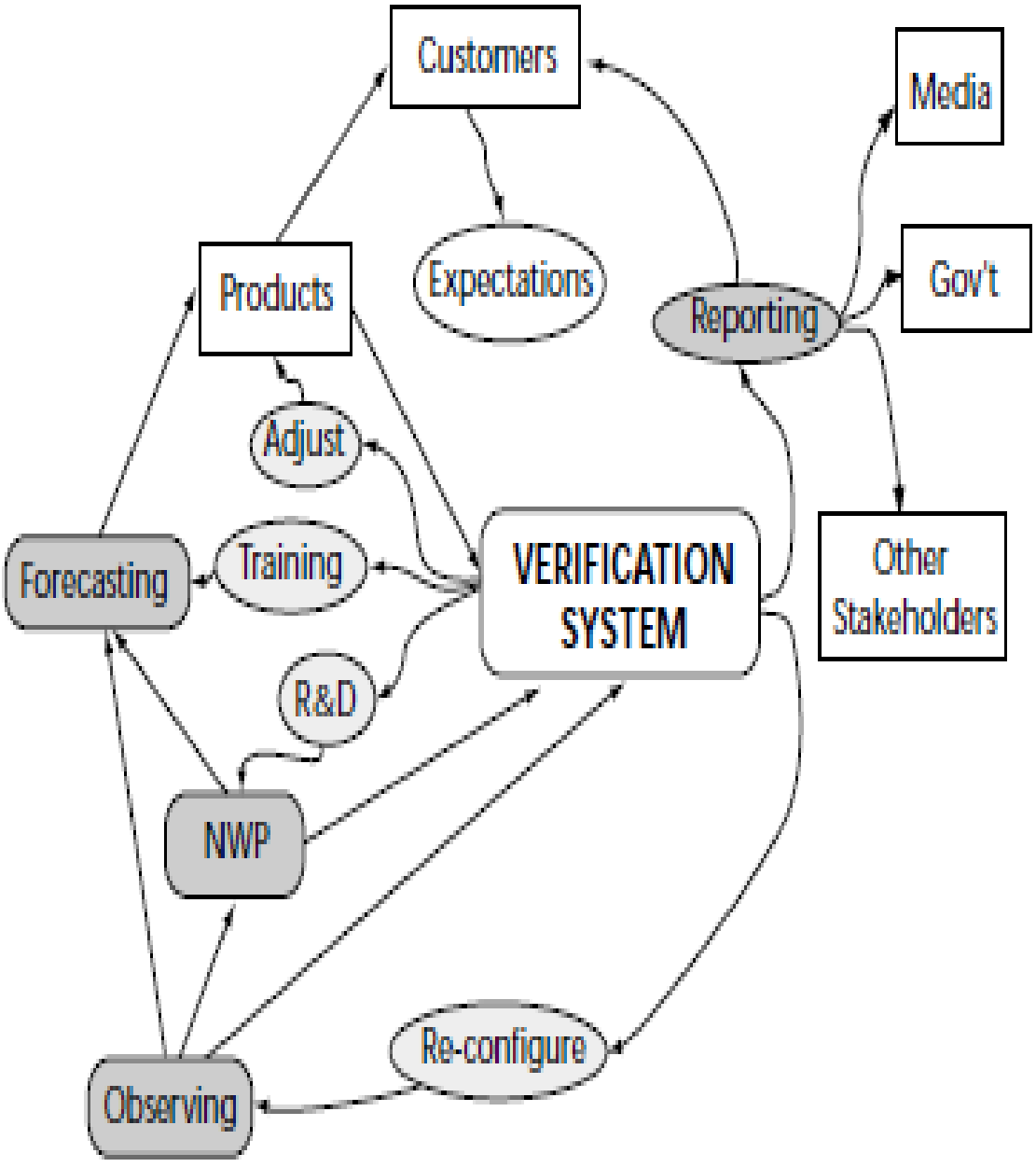
Data collection: We have collected weather data from WORLD DATA CENTER for climate, Hamburg. We have decided to use NWS API for data collection in future.

Data formatting and cleaning: We have converted our data from .NC (netcdf) format to .CSV (comma-separated values) format because WEKA supports .CSV formats.

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

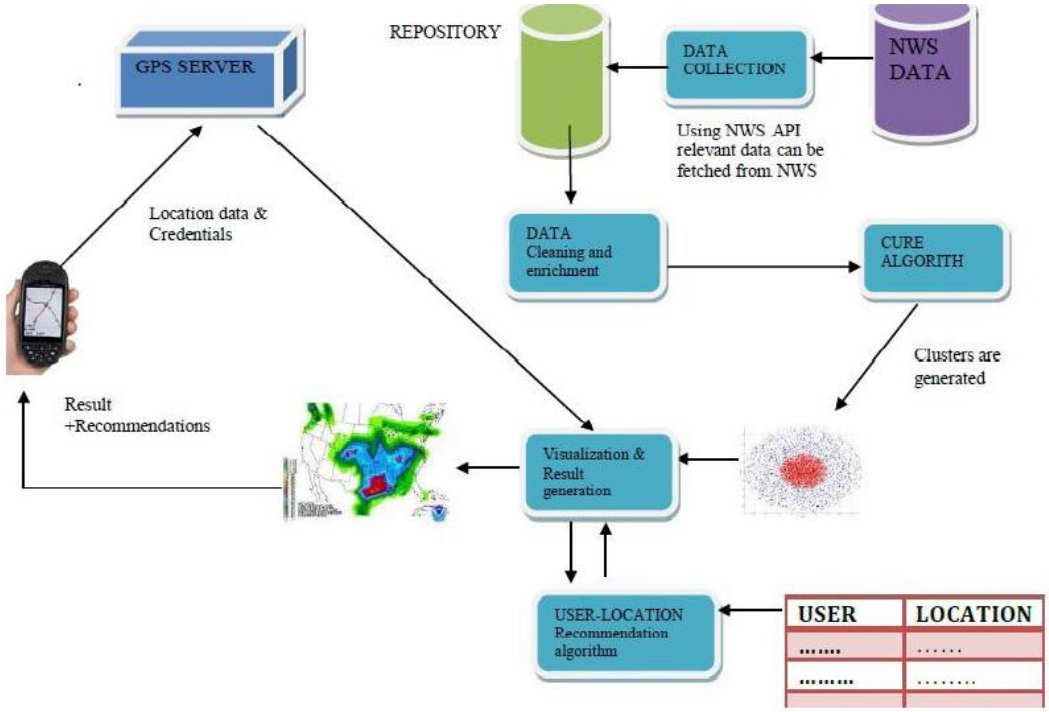
4.1 DESIGN GOALS



The Fig 1.2 shows the basic design model of how the forecasting takes place in real world.

Fig 1.2 Design Goals

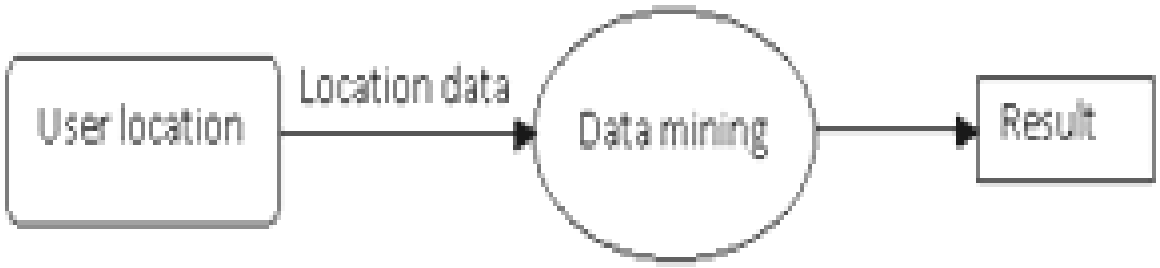
4.2 OVERALL SYSTEM ARCHITECTURE



The Fig 1.3 shows the overall Architecture of the forecasting Mechanism.

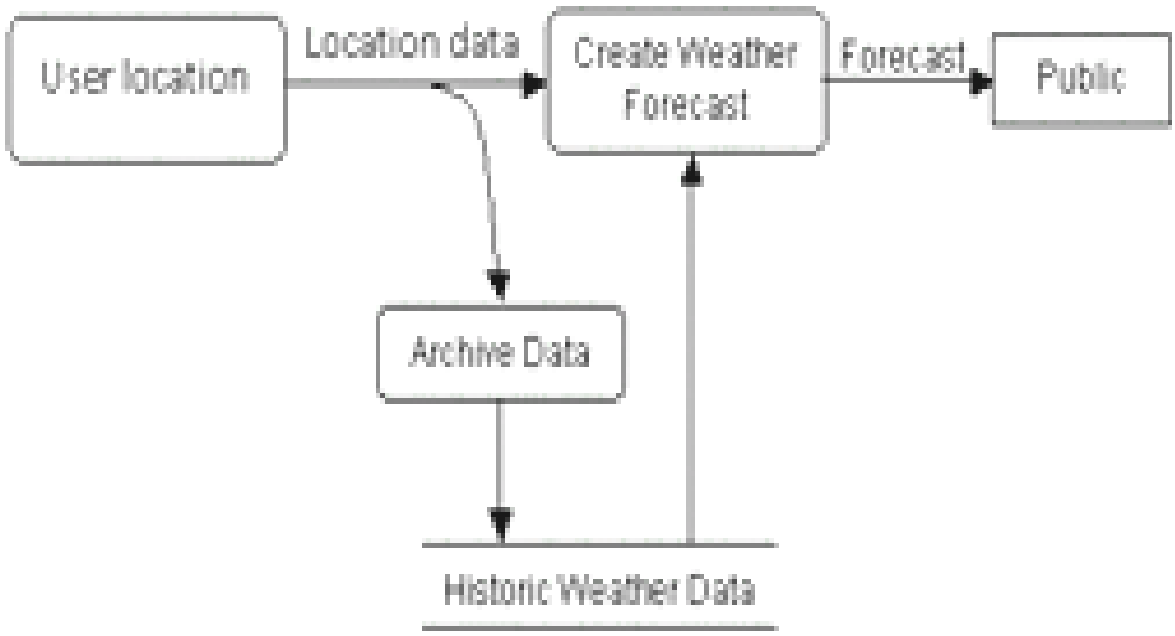
Fig 1.3. Overall System Architecture

4.3 DATA FLOW DIAGRAM/ ACTIVITY DIAGRAM



The Fig 1.4 shows the module flowchart of predicting weather in user location using data mining.

Fig 1.4 Level 0



The Fig 1.5 shows the how the data mining is done using Historical data.

Level 1

Fig 1.5. Data Flow Diagram

A **data flow diagram (DFD)** is a graphical representation of the flow of data through an information system. A data flow diagram can also be used for the visualization of data processing (structured design). It is common practice for a designer to draw a context-level DFD first which shows the interaction between the system and outside entities. This context-level DFD is then exploded to show more detail of the system being modelled.

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system. The DFD is simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data and the output data generated by the system.

A DFD model uses a very limited number of primitive symbols to represent the functions performed by a system and the data flow among the functions. The main reason why the DFD technique is so popular is probably because of the fact that DFD is a very simple formalism- It is simple to understand and use. Starting with the set of high level functions that a system performs, a DFD model hierarchically represents various sub functions. In fact, any hierarchical model is simple to understand. The human mind is such that it can easily understand any hierarchical model of a system because in a hierarchical model, starting with a very simple and abstract model of system, different details of a system are slowly introduced through the different hierarchies. The data flow diagramming technique also follows a simple set of intuitive concepts and rules.

DFD is an elegant modeling technique that turns out to be useful not only to represent the results of a structured analysis of a software problem but also for several other

applications such as showing the flow of documents or items in an organization. A data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design). On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process.

Level 0: A context-level or level 0 data flow diagram shows the interaction between the system and external agents which act as data sources and data sinks. On the context diagram (also known as the Level 0 DFD) the system's interactions with the outside world are modeled purely in terms of data flows across the system boundary. The context diagram shows the entire system as a single process, and gives no clues as to its internal organization

Level 1: The Level 1 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, Level 1 Data Flow diagram shows an in depth explanation of overall process of the data flow.

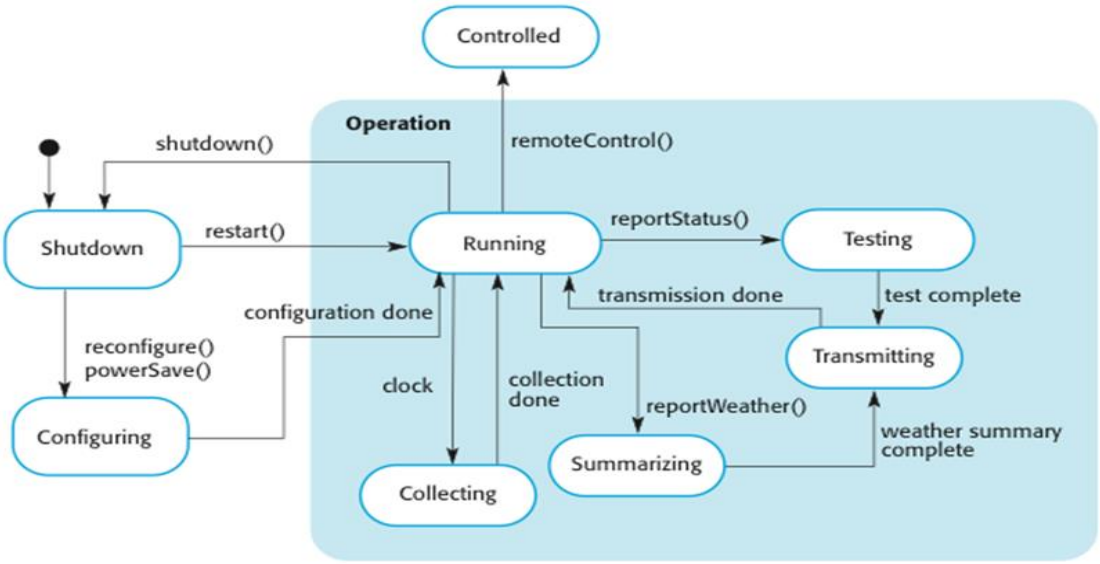
Level 2: Complete details of every function performed by Admin. Every step is explained in detail.

The four components of a data flow diagram (DFD) are:

External Entities/Terminators are outside of the system being modelled. Terminators represent where information comes from and where it goes. In designing a system, we have no idea about what these terminators do or how they do it. Processes modify the inputs in the process of generating the outputs

Data Stores represent a place in the process where data comes to rest. A DFD does not say anything about the relative timing of the processes, so a data store might be a place to accumulate data over a year for the annual accounting process. Data Flows shows how data moves between terminators, processes, and data stores (those that cross the system boundary are known as IO or Input Output Descriptions).

4.4 STATE MACHINE UML DIAGRAM

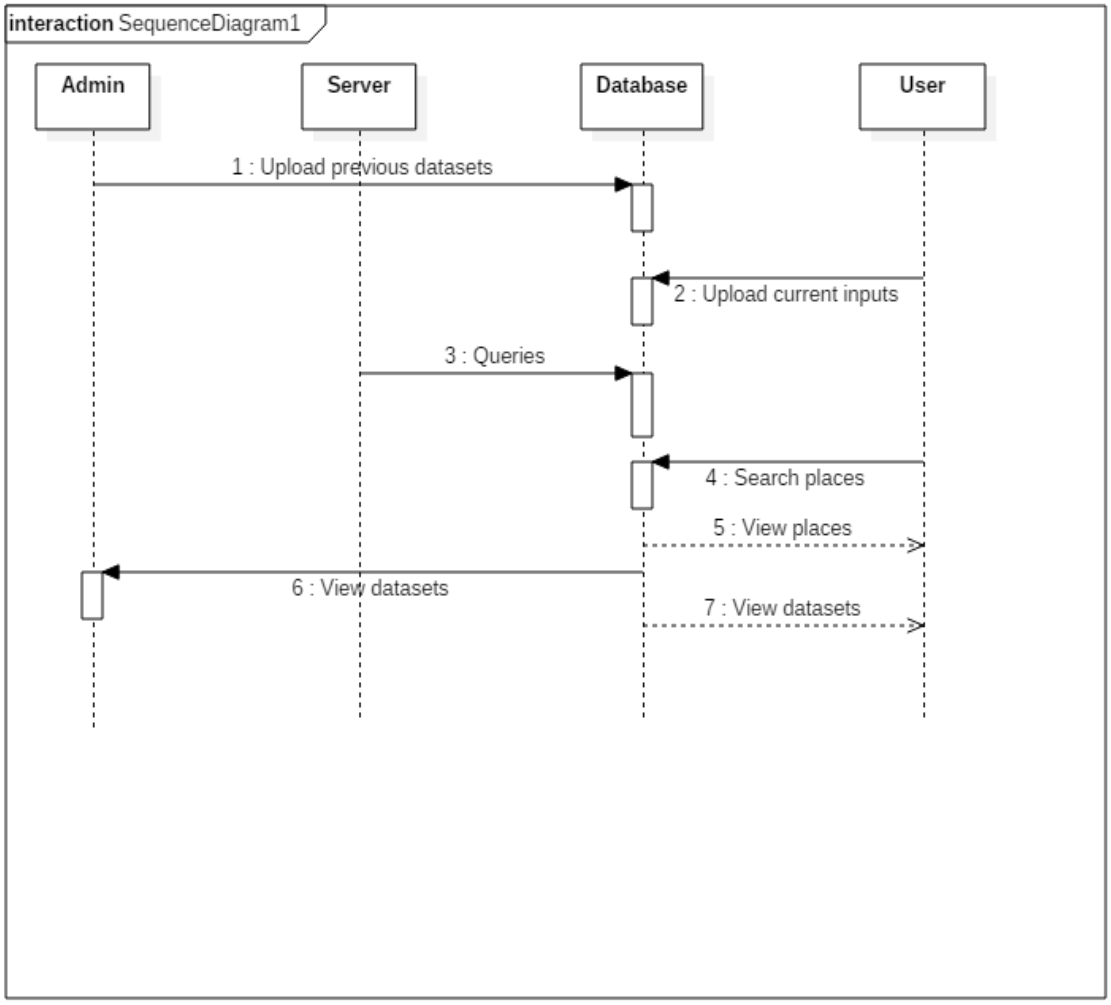


The Fig 1.6 shows the flow of UML from start to end with each process

Fig 1.6. State Machine UML Diagram

4.5 SEQUENCE DIAGRAM

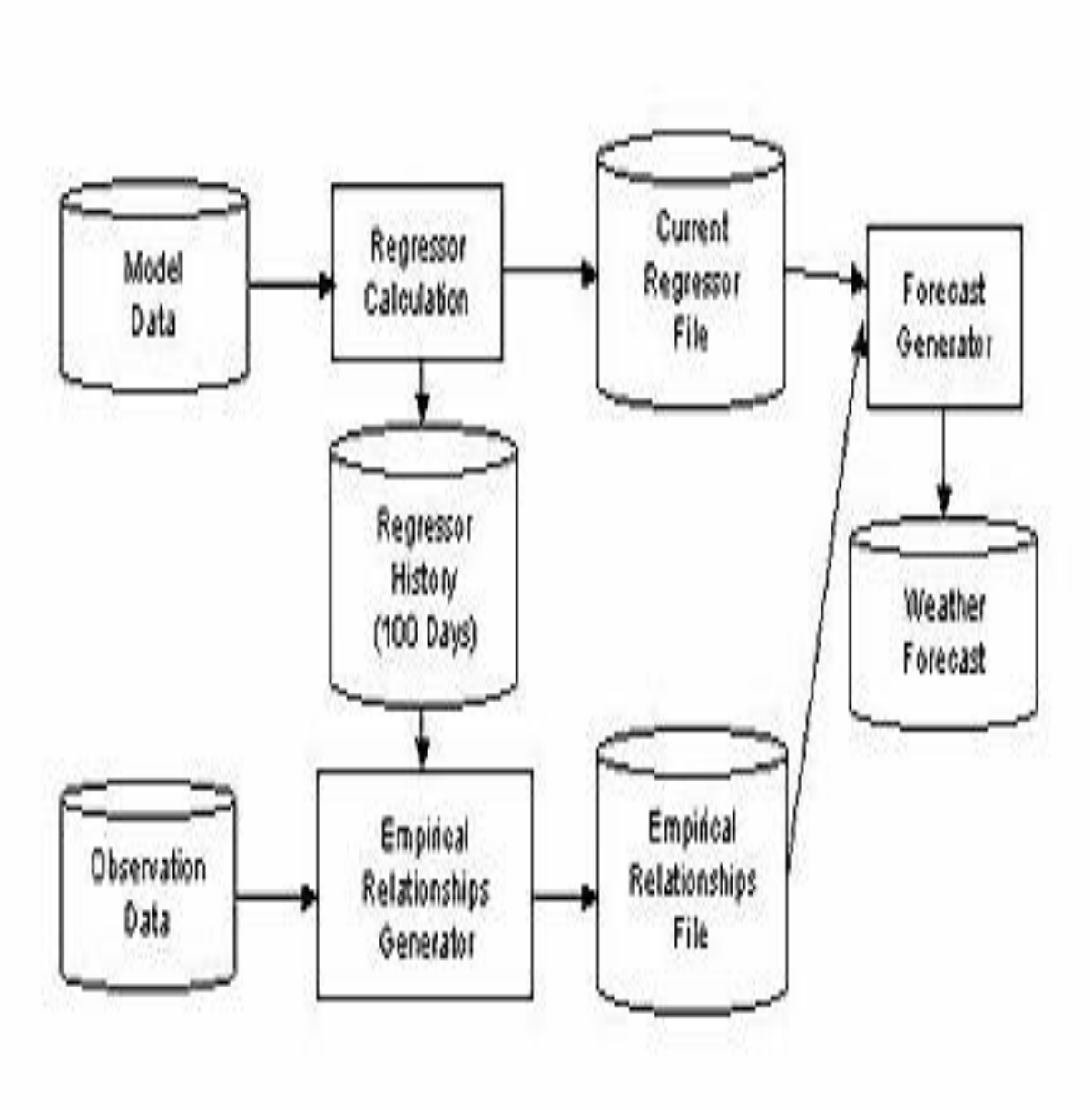
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Chart. Sequence diagrams are sometimes called event diagrams, event scenarios and the sequence diagram for this project is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



The fig 1.7 shows the datasets and the output given to the desired output for the queries.

Fig 1.7. Sequence Diagram

4.6 INTERACTION OVERVIEW DIAGRAM



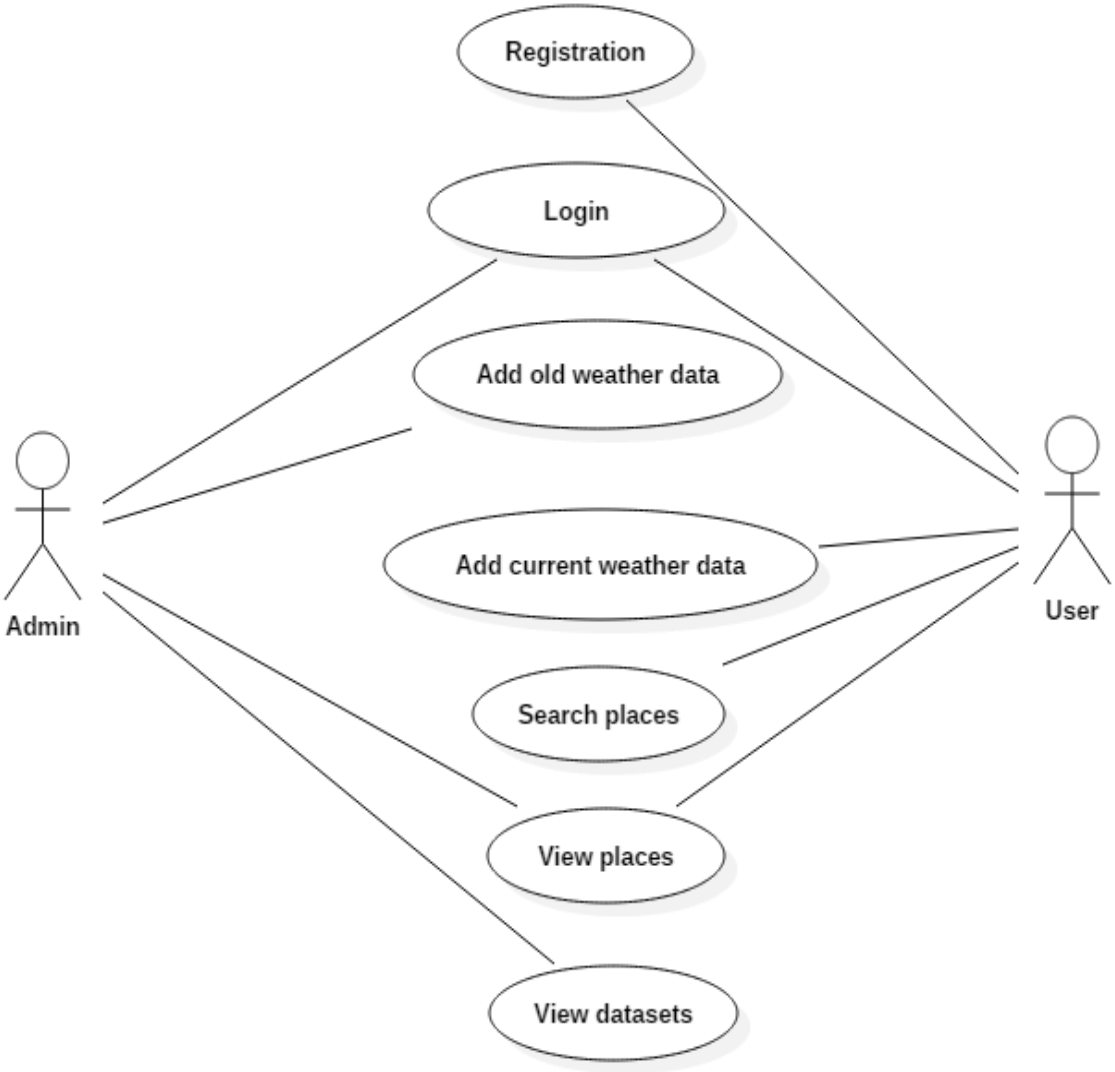
The Fig 1.8 shows the entire forecasting model of our project

Fig 1.8. Interaction Overview Diagram

4.7 USE CASE DIAGRAM

A use case diagram is a type of behavioural diagram 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. Use case diagram provides us the information about how that users and use cases are related

with the system.



The fig 1.9 shows the interaction to the server from the admin and user

Fig 1.9. Use Case Diagram

CHAPTER 5

IMPLEMENTATION

5.1 ALGORITHM

Understanding Artificial Neural Networks Theory

In the last article I described the process of building a linear regression model, a venerable machine learning technique that underlies many others, to predict the mean daily temperature in Lincoln, Nebraska. Linear regression models are extremely powerful and have been used to make numerical, as well as categorical, predictions since well before the term "machine learning" was ever coined. However, the technique has some criticisms, mostly around its ridged assumption of a linear relationship between the dependent variable and the independent variable(s).

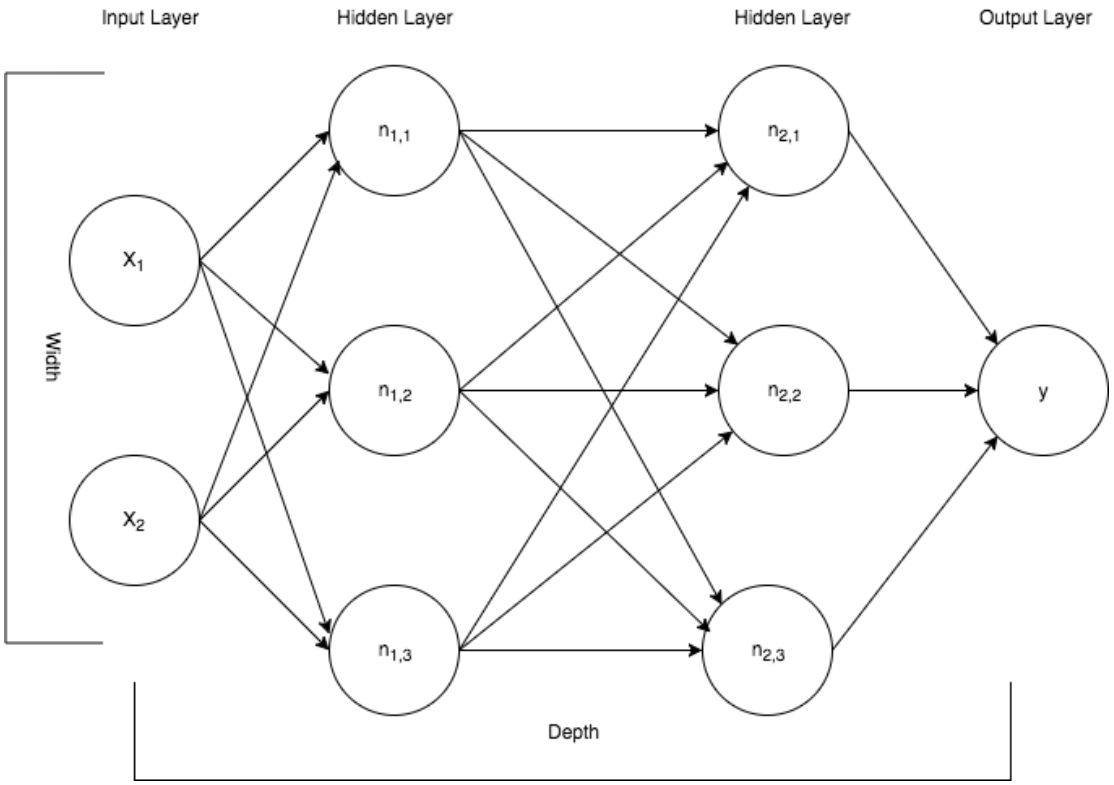
An uncountable number of other algorithms exist in the data science and machine learning industry which overcome this assumption of linearity. One of the more popular areas of focus in recent years has been to apply neural networks to a vast array of machine learning problems. Neural networks have a powerful way of utilizing learning techniques based on both linear and non-linear operations.

Neural networks are inspired by biological neurons in the brain which work in a complex network of interactions to transmit, collect, and learn information based off a history of the information that has already been collected. The computational neural networks we are interested in are similar to the neurons of the brain in that they are a collection of neurons (nodes) that receive input signals (numerical quantities), process the input, and transmits the processed signals to other downstream agents in the network. The processing of signals as numerical quantities that pass through the neural network is a very powerful feature that is not limited to linear relationships.

In this series I have been focusing on a specific type of machine learning called supervised learning, which simply means that the models being trained are built using data that has known target outcomes that the model is trying to learn to predict.

Furthermore, the type of predictions being made are numerical real values, which means we are dealing with *regressor* prediction algorithms.

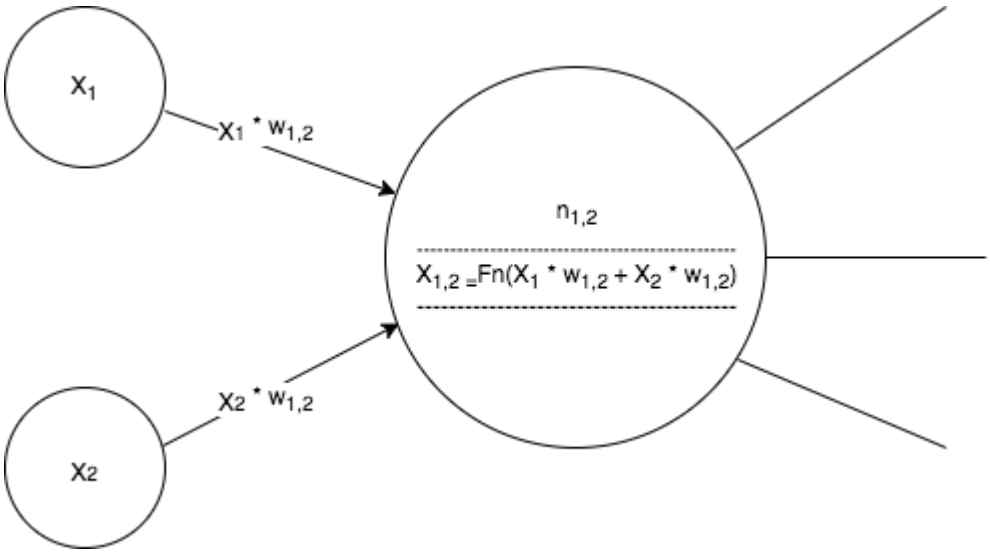
Graphically, a neural network similar to the one being described in this article is shown in the image below.



The neural network depicted above contains an input layer on the far left representing two features, x1 and x2, that are feeding the neural network. Those two features are fed into the neural network, which are processed and transmitted through two layers of neurons, which are referred to as hidden layers. This depiction shows two hidden layers with each layer containing three neurons (nodes). The signal then exits the neural network and is aggregated at the output layer as a single numerical predicted value.

Let me take a moment to explain the meaning behind the arrows signifying data being processed from node to node across the layers. Each arrow represents a mathematical transformation of a value, beginning at the arrow's base, which is then multiplied by a weight specific to that path. Each node within a layer will be fed a value in this way.

Then all the values converging at the node are summed. It is this aggregate of multiplying by weights and summing the products that define the linear operations of a neural network I mentioned earlier.



After summation is carried out at each node a special, non-linear, function is applied to the sum, which is depicted in the image above as **Fn(...)**. This special function that introduces non-linear characteristics into a neural network is called an activation function. It is this non-linear characteristic brought about by activation functions that give multi-layer neural networks their power. If it was not for the non-linearity added to the process then all layers would effectively just algebraically combine into one constant operation consisting of multiplying the inputs by some flat coefficient value (ie, a linear model).

Alright, so that is all fine and dandy, but I hope you are wondering in the back of your mind... ok, Adam, but how does this translate into a learning algorithm? Well the most straight forward answer to that is to evaluate the predictions being made, the output of the model "y", to the actual expected values (the targets) and make a series of adjustments to the weights in a manner that improves the overall prediction accuracy.

In the world of regressor machine learning algorithms one evaluates the accuracy by using a cost (aka "loss", or "objective") function, namely the sum of squared errors (SSE). Notice that I generalized that statement to the whole continuum of

machine learning, not just neural networks. In the prior article the Ordinary Least Squares algorithm accomplished just that, it found the combinations of coefficients that minimized the sum of the squared errors (ie, least squares).

Our neural network regressor will do the exact same thing. It will iterate over the training data feeding in feature values, calculate the cost function (using SSE) and make adjustments to the weights in a way that minimizes the cost function. This process of iteratively pushing features through the algorithm and evaluating how to adjust the weights based off the cost function is, in essence, what is known as model optimization.

Model optimization algorithms are very important in building robust neural networks. As examples are fed through the network's architecture (i.e., the width and depth) then evaluated against the cost function, the weights are adjusted. The models is said to be "learning" when the optimizer function identifies that a weight adjustment was made in a way that does not improve (lower) the cost function, which is registered with the optimizer so that it does not adjust the weights in that direction again.

TensorFlow's High Level Estimator API

Google's TensorFlow library consists a few API's, with the most popular being the Core API, which gives the user a low-level set of tools to define and train essentially any machine learning algorithm using symbolic operations. This is referred to as TensorFlow Core. While TensorFlow Core is an amazing API with vast application capability, I will be focusing on a newer, higher level, API the TensorFlow team developed that is collectively referred to as the Estimator API.

The TensorFlow team developed the Estimator API to make the library more accessible to the everyday developer. This high-level API provides a common interface to `train(...)` models, `evaluate(...)` models, and `predict(...)` outcomes of unknown cases similar to (and influenced by) the popular Sci-Kit Learn library, which is accomplished by implementing a common interface for various algorithms. Also, built into the high-level API are a load of machine learning best practices, abstractions, and ability for scalability.

All of this machine learning goodness brings about a set of tools implemented in the base Estimator class as well as multiple pre-canned model types that lowers the barrier to entry for using TensorFlow so it can be applied to a host of everyday problems (or opportunities). By abstracting away much of the mundane and manual aspects of things like writing training loops or dealing with sessions, the developer is able to focus on more important things like rapidly trying multiple models and model architectures to find the one that best fits their need.

5.2 PSEUDOCODE

Data collection and processing

```
def extract_weather_data(url, api_key, target_date, days):
    records = []
    for _ in range(days):
        request = BASE_URL.format(API_KEY, target_date.strftime('%Y%m%d'))
        response = requests.get(request)
        if response.status_code == 200:
            data = response.json()['history']['dailysummary'][0]
            records.append(DailySummary(
                date=target_date,
                meantempm=data['meantempm'],
                meandewptm=data['meandewptm'],
                meanpressurem=data['meanpressurem'],
                maxhumidity=data['maxhumidity'],
                minhumidity=data['minhumidity'],
                maxtempm=data['maxtempm'],
```

```
                mintempm=data['mintempm'],
                maxdewptm=data['maxdewptm'],
                mindewptm=data['mindewptm'],
                maxpressurem=data['maxpressurem'],
                minpressurem=data['minpressurem'],
                precipm=data['precipm']))
        time.sleep(6)
        target_date += timedelta(days=1)
    return records
if not station_id:
            raise Exception("'station_id' is required.")
        if 'recordId' in params and 'current' in params:
            raise Exception("Cannot have both 'current' and 'recordId'")
        if 'start' in params:
            start = params['start']
            self.parse_param_timestamp(start)
            if len(start) < 19:
                start = '{}T00:00:00Z'.format(start[:10])
            elif len(params['start']) < 20:
                start = start.replace(' ', 'T')
                start = '{}Z'.format(start)
            params['start'] = start
        if 'end' in params:
```

```
        end = params['end']
        self.parse_param_timestamp(end)
        if len(end) < 19:
            end = '{}T23:59:59Z'.format(end[:10])
        elif len(params['end']) < 20:
            end = end.replace(' ', 'T')
            end = '{}Z'.format(end)
        params['end'] = end

    request_uri = "/stations/{stationId}/observations".format(
        stationId=station_id)

    if len(params) > 0:
        if 'recordId' in params:
            return self.make_get_request(
                '{old_request_uri}/{recordId}'.format(
                    old_request_uri=request_uri,
                    recordId=params['recordId']),
                end_point=self.DEFAULT_END_POINT)
        if 'current' in params:
            return self.make_get_request(
                '{old_request_uri}/current'.format(
                    old_request_uri=request_uri),
```

```
                end_point=self.DEFAULT_END_POINT)

            if len(params) > 1:
                request_uri = '{old_request_uri}?{query_string}'.format(
                    old_request_uri=request_uri,
                    query_string=urlencode(params))

            observations = self.make_get_request(
                request_uri, end_point=self.DEFAULT_END_POINT)
            if 'features' not in observations:
                raise Exception(observations)
            return observations['features']

        return self.make_get_request(
"/stations/{stationId}/observations".format(stationId=station_id),
            end_point=self.DEFAULT_END_POINT)

    def products(self, id):
        functional tests @todo(paulokuong) later on.
    Args:
            id (str): product id.
        Returns:
            json: json response from api.
```

```
"""
        return self.make_get_request(
"/products/{productId}".format(productId=id),
            end_point=self.DEFAULT_END_POINT)


    def products_types(self, **params),
      Response in this method should not be modified.
        In this way, we can keep track of changes made by NOAA through
        functional tests @todo(paulokuong) later on.
 Args:
            type_id (str): an id of a valid product type
            locations (boolean[optional]): True to get a list of
                locations that have issues products for a type.
            location_id (str): location id.
        Returns:
            json: json response from api.
        if 'type_id' in params and 'locations' not in params:
            return self.make_get_request(
"/products/types/{type_id}".format(type_id=params['type_id']),
                end_point=self.DEFAULT_END_POINT)
        elif 'locations' in params:
            if 'type_id' not in params:
                raise Exception('Error: Missing type id (type_id=None)')
```

```
        if 'location_id' in params:
            return self.make_get_request(
                ('/products/types/{type_id}/locations/'
                 '{location_id}').format(
                    type_id=params['type_id'],
                    location_id=params['location_id']),
                end_point=self.DEFAULT_END_POINT)
        else:
            return self.make_get_request(
"/products/types/{type_id}/locations".format(
                    type_id=params['type_id']),
                end_point=self.DEFAULT_END_POINT)
    return self.make_get_request(
"/products/types",
        end_point=self.DEFAULT_END_POINT)
 def products_locations(self, **params)
    functional tests @todo(paulokuong) later on.
Args:
        location_id (str): location id.
    Returns:
        json: json response from api.
    if 'location_id' in params:
        return self.make_get_request(
```

```
"/products/locations/{locationId}/types".format(
                locationId=params['location_id']),
            end_point=self.DEFAULT_END_POINT)
        return self.make_get_request(
"/products/locations",
            end_point=self.DEFAULT_END_POINT)
```

Front-End Code

```
 <!DOCTYPE html>
<html lang="en" >
<head>
   <meta charset="UTF-8">
   <title>Weather.IO</title>
   <link rel="shortcut icon" type="ico" href="favicon.ico"/>
   <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.7.5/angular.min.js"></script>
   <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
   <link rel="stylesheet" href="images/animate.min.css">
   <style>
      @import
url('https://fonts.googleapis.com/css?family=Roboto|Krub|ABeeZee|Comfortaa');
      @import
url('https://fonts.googleapis.com/css?family=Andada|Permanent+Marker|Raleway:30
0');
      * {
```

```
        box-sizing: border-box;
    }
    input:-webkit-autofill,
    input:-webkit-autofill:hover,
    input:-webkit-autofill:focus,
    input:-webkit-autofill:active {
        transition: background-color 5000s ease-in-out 0s;
        -webkit-text-fill-color: #fff;
    }
html {
    -webkit-tap-highlight-color: rgba(0,0,0,0);
    -webkit-tap-highlight-color: transparent;
min-height: 100%;
-webkit-background-size: cover;
-moz-background-size: cover;
-o-background-size: cover;
background-size: cover;
background: -webkit-linear-gradient(to bottom,#004e92, #000428);
background: linear-gradient(to bottom,#004e92, #000428);
background-repeat:no-repeat;
background: -webkit-linear-gradient( to bottom #004e92, #000428);
background: -moz-linear-gradient( to bottom #004e92, #000428);
background: -ms-linear-gradient( to bottom #004e92, #000428);
background: -o-linear-gradient( to bottom #004e92, #000428);
```

```
    background: linear-gradient( to bottom #004e92, #000428);
}



    h1 {
        display: inline;
    }

    body {
        background-attachment: fixed;
        margin: 0;

        position: relative;
        font-family: 'Roboto', sans-serif;
        color: #fff;
    }
    #namer {
        position: relative;
        width: 400px;
        margin: 149px auto 0;
    }
    #namer input {
        border: 0px;
        border-bottom: 2px solid #1BB1DC;
```

```
    width: 55%;
    /*float: right;*/
    font-size: 30px;
    line-height: 35px;
    height: 70px;
    text-align: center;
    padding-right: 10px;
    background: transparent;
    color: #fff;
    font-family: 'ABeeZee',sans-serif;

}
#namer input:focus {
    outline: 0;
    color: #fff;
}
#namer input::-webkit-input-placeholder {
    color: #fff;
}
#namer input:-ms-input-placeholder {
    color: #fff;
}
#namer input::-ms-input-placeholder {
    color: #fff;
```

```
}
#namer input::placeholder {
    color: #fff;
}
.namer-controls {
    position: relative;
    display: block;
    height: 30px;
    margin: 20px 0;
    text-align: center;
    opacity: 1;
    cursor: pointer;
}
.namer-controls div {
    float: left;
    width: 50%;
}
button {
    box-shadow: 0px 0px 30px 0px rgba(0, 0, 0, 0.25);
    background: #004e92;
    padding: 15px 40px;
    width:170px;
    display: inline-block;
    border-radius: 30px;
```

```
    margin-top: 7px;
    /*background-color: transparent;*/
    outline: none;
    border: 0px;
    color: #fff;
    cursor: pointer;
}
.namer-controls div span:last-child {
    margin-right: 0;
}
#box {
    width:50%;
    height:50%;
    margin: 0 auto;
    border-radius: 100px;
}
#Header {
    font-family: 'Comfortaa', cursive;
    /*margin-top: 0px;*/
    /*text-align: center;*/
    font-size: 60px;
    color: rgba(250,250,250,1);
    cursor: pointer;
    /*padding-top: 5px;*/
```

```
    /*text-align: center;*/
    /*float: right;*/
    /*margin-right: 41%;*/
    margin-bottom: 10%;
    text-align: center;
    /*padding-top: 5px;
    /*padding-bottom: 5px;*/
    /*background: #00c4cc linear-gradient(139deg, #00c4cc 0, #7d2ae8 100%);*/
    /*background-color: red;*/
}
#one {
    /*margin-left: 80px;*/
    float: left;
    display: inline-block;
    margin-top: 15%;
}
#two {
    /*margin-left: 80px;*/
    display: inline-block;
    float: right;
    margin-top: 15%;
}
/*.hide {*/
/*visibility: hidden;*/
```

```
/*}*/
#form {
    margin-right: 400px;
    text-align: center;
    width : 600px;
    /*padding-left: 100px;*/
}
#this {
    float: right;
}
#back {
    box-shadow: 0px 0px 30px 0px rgba(0, 0, 0, 0.25);
    display: inline-block;
    /*margin-top: 7px;*/
    /*height: ;*/
    border-radius: 50%;
    width: 48px;
    height: 48px;
    /*font-size: 20px;*/
    /*text-transform: uppercase;*/
    /*letter-spacing: 0.3px;*/
    margin-top: 18px;
    background-color: transparent;
    outline: none;
```

```
    border: 0px;
    color: #fff;
    /*font-size: 20px;*/
    cursor: pointer;
    float: left;
    /*padding: 20px 50px;*/
    /*width: 100%;*/
    /*height:100%;*/
}

/*#logo {*/
/*margin-top: 100px;*/
/*display:none;*/
/*}*
.mar {
    margin-top: -100px;
}
#one, #two {
    z-index: 1;
    background: #131521;  /* fallback for old browsers */
    /*background: -webkit-linear-gradient(to right, #A6FFCB, #12D8FA, #1FA2FF);  !* Chrome 10-25, Safari 5.1-6 *!*/
    /*background: linear-gradient(to right, #A6FFCB, #12D8FA, #1FA2FF); !* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+ *!*/
    /*position: relative;*/
```

```
        display: inline-block;
        border-radius: 20px;
        width: 48%;
        height: 150px;
        box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2), 0 6px 20px 0 rgba(0,0,0,0.19);
        /*padding: 20px 38px 100px 38px;*/
        margin: 10% auto ;
        /*margin: 10px;*/
    }
</style>
<script type="text/javascript">
    function secondClick() {
        setTimeout(function(){
            document.getElementById("secondClick").click(); }, 0);        }
    function getLocation() {
        if (navigator.geolocation) {
            navigator.geolocation.getCurrentPosition(showPosition);
        } else {
            x.innerHTML = "Geolocation is not supported by this browser.";
        }
    }
    function showPosition(position) {
        x = position.coords.latitude;
        y = position.coords.longitude;
```

```
    var a = document.getElementById("input_lat");
    a.value = x;
    var b = document.getElementById("input_lng");
    b.value = y;
    secondClick();
}

$(document).ready(function(){
    $("#back").css(
        "display" , "none"
    );
    $("#namer-input").css(
        "display" , "none"
    );
    $("#this").css(
        "display" , "none"
    );
    $("body").fadeIn("slow");
    $("#one").fadeIn("slow");
    $("#two").fadeIn("slow");
    $("#start").fadeIn("slow");
    $("#last").fadeIn("slow");
    $("#logo").fadeIn("slow");
    $("#Header").fadeIn("slow");
```

```
$("#one").click(function(e){
    e.preventDefault();

    $("#one").css(
        "display" , "none"
    );

    $("#two").css(
        "display" , "none"
    );
    $("#namer-input").css(
        "display" , "block"
    );
    $("#this").css(
        "display" , "block"
    );
    $("#back").css(
        "display" , "block"
    );
});

$("#back").click(function(){
    // $("#back").fadeOut('slow');
    // $("#this").fadeOut('slow');
```

```
// $("#namer-input").fadeOut('slow');


// $("#back").css(
//     "display" , "none"
// );
// $("#this").fadeOut('slow').animate({
//     opacity : 0
// }, 'slow', 'linear');
// $("#namer-input").fadeOut('slow').animate({
//     opacity : 0
// }, 'slow', 'linear');


$("#namer-input").css(
    "display" , "none"
);
$("#this").css(
    "display" , "none"
);
$("#back").css(
    "display" , "none"
);


// $("#start").animate({
//         opacity:0,
```

```
            //         marginLeft: '33.20vw',
            //         marginRight: '36.40vw'
            //     },'slow','linear'
            // );
            //

            $("#one").css(
                "display" , "block"
            );

            $("#two").css(
                "display" , "block"
            );
        })
    });
</script>
</head>
<body>
<div ng-app="app">
    <div ng-controller="emp" style="display: inline;">

        <img id="logo" src="images/logo.svg" class="animated fadeIn" height="10.5%"
width="5%" style=" float: left !important; position: absolute;margin: 0 0 0 2%; ">
```

```
<p id="Header" class="animated fadeIn" style="padding: 1%; text-align: center; position: relative; margin-top: 1%"><b>Weather.IO</b></p>

<div id="box" style="text-align: center;margin-top: 10%;">

<div id="namer" style="text-align: center;">

<div id="main" style="text-align: center;">

<div class="namer-controls" style="text-align: center;" >

<form id="form" action='/res' style="text-align: center;">

<span id="namer-input" class="hide" style="text-align: center;">

<input type="text" class="animated flipInX" name="location" placeholder="Search here" style="z-index:0;margin-left:-70%;position: absolute;width:300px;margin-top: 10%">

</span>

<button class="hide animated flipInX" id="this" type="submit" style="margin-left:-45%;z-index:3;margin-right: 10%;margin-top: 9%"> <img src="images/search.svg" width="26px" height="26px"></button>

<button type="button" class="animated flipInX" id="back" style="z-index: 1; padding: 0px; display:none;margin-left:-10%;margin-top: 10%"><span > <img width="48px" height="48px" src="images/back.svg"></span></button> </form><button id="one" class="animated fadeInLeft" type="button" style="display:none;text-align: center;"> <img src="images/search.svg" width="32px" height="32px" style="margin-top: 5px; margin-bottom: 10px;"><br> Search</button> <button id="two" class="animated fadeInRight" type="button" onclick="getLocation()" style="display:none;z-index:4;text-align: center;"><img src="images/location1.svg" width="32px" height="32px" style="margin-top: 5px; margin-bottom: 10px;" ><br>Tap to get weather of current location</button> </div><!-- <p style="font-family: 'ABeeZee', sans-serif; font-size: 22px; text-align: center; margin-top: 190px;">Tap GO to get the weather of current location or search for any place around the world</p> -->

</div> </div></div> </div> </div>

<div id="last" style="text-align: center; font-size: 15px; font-family: 'ABeeZee', sans-serif;margin-top: 45vh; color: white; text-decoration: none;display: none;">

<h3>&amp;lt;&amp;gt; with &amp;#10084;</h3>

<p style="text-align: center; font-size: 15px; font-family: 'ABeeZee', sans-serif; color: white; text-decoration: none;"> by </p>
```

```
    <a style="text-align: center; font-size: 15px; font-family: 'ABeeZee', sans-serif; color: white; text-decoration: none;" href="https://github.com/nikhils4"></a>

    <a style="text-align: center; font-size: 15px; font-family: 'ABeeZee', sans-serif; color: white; text-decoration: none;" href="https://github.com/pm-sharma"></a>

    <a style="text-align: center; font-size: 15px; font-family: 'ABeeZee', sans-serif; color: white; text-decoration: none;" href="https://github.com/satin-04"></a>

</div>

<form action='/locate' style="display:none;">

    <input type="text" name='name_lat' id="input_lat">

    <input type="text" name='name_lng' id="input_lng">

    <button type="submit" id="secondClick"></button>

</form>

<script
src='https://cdnjs.cloudflare.com/ajax/libs/jquery/3.1.1/jquery.min.js'></script>

</body>

</html>
```

Server side scripting

```
const request = require("request");

var getWeather = (lat, lan, callback) => {

    request ({

        url : 'https://api.darksky.net/forecast/61043ce7510ce565a6616401169c0fba/' + lat + ',' + lan,

        json : true

    }, (error, response, body) => {

        if(error){

            callback('Unable to connect to server try again later !');
```

```
        }
        else if (response.statusCode===400){
            callback('Unable to fetch weather try again later !')
        }
        else if(response.statusCode===200){
            callback(undefined, {
                temperature : body.currently.temperature,
                icon : body.currently.icon,
                prediction : body.hourly.summary,
                summary : body.currently.summary,
                wind : body.currently.windSpeed,
                humidity : body.currently.humidity,
                pressure : body.currently.pressure
            })
        }
    });
};
module.exports.getWeather = getWeather;
```

Reserve code

```
const request = require("request");
var reverse = (lat, lan, callback) => request ({
        url : 'https://api.opencagedata.com/geocode/v1/json?q=' + lat + '+' + lan + '&pretty=1&key=5db5a96f174d46e389853c0ec12d44a2',
        json : true
    }, (error, response, body) => {
```

```
        if(error){
            callback('Unable to connect to server try again later !');
        } else{
            callback(undefined, {
                address : body.results[0].formatted
            });
        }
    });
};
module.exports.reverse = reverse;
```

CHAPTER 6

RESULTS AND DISCUSSION

This system is a mobile app application with an effective graphical user interface, that is capable of predicting weather based on parameters such as temperature, pressure and rain. The user enters the temperature, pressure and rain and must enter the next day to get the accurate prediction. The location and current weather is mapped and shown in the google maps using an API and also a bar graph is plotted to show the predicted results.

Two graphs are humidity and temperature tomorrow. The left side graph shows the humidity and the right-side graph shows the temperature. The user can understand the graph by seeing the difference of increase and decrease in temperature and humidity compared to previous day.

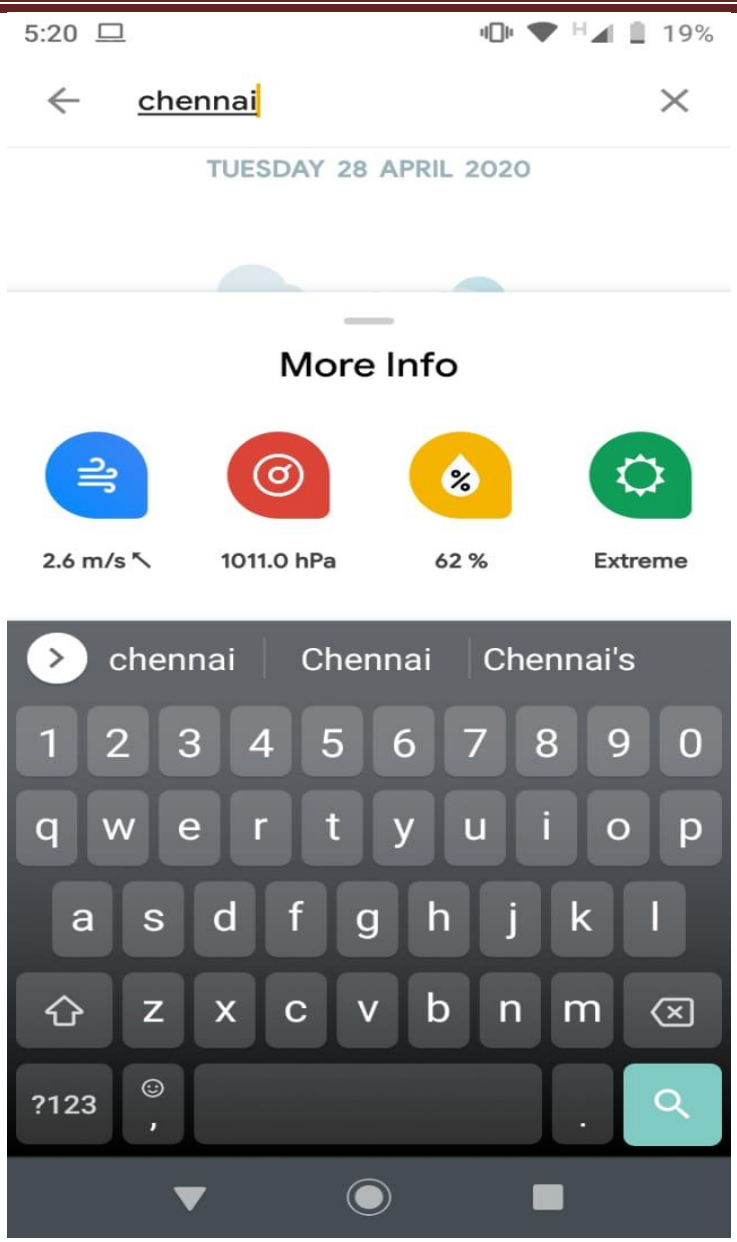


Fig 2.1. Search for particular places

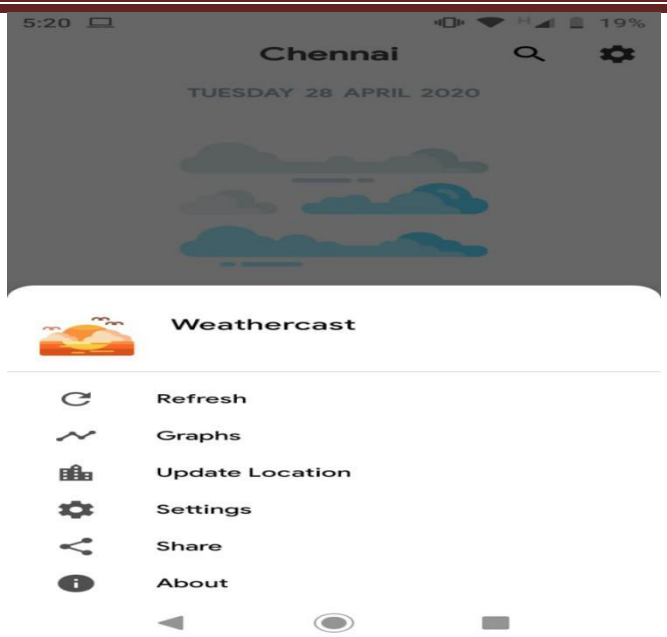


Fig 2.2. Options Used In Weather Forecasting

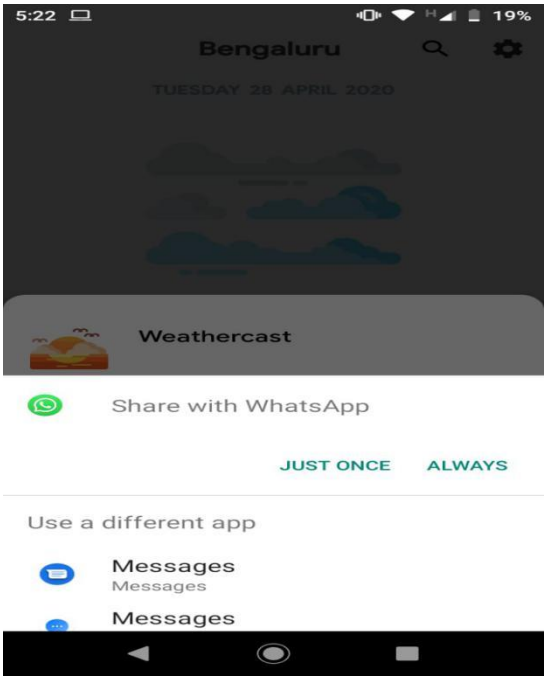


Fig 2.3. Sharing Weather Forecasting Details

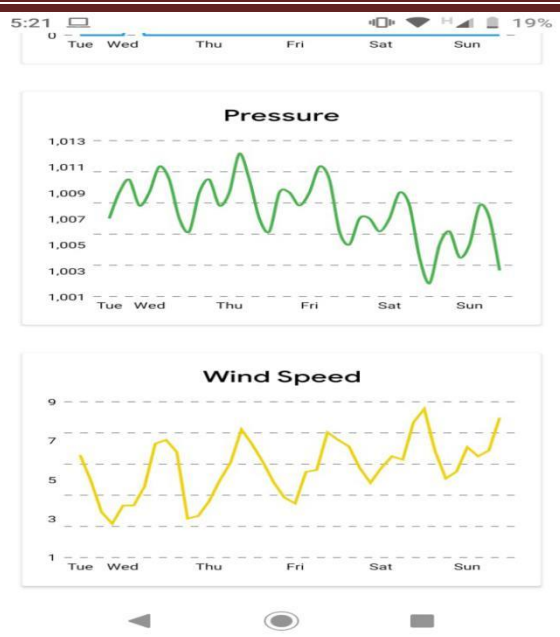


Fig 2.4. Predicting Pressure and Wind Speed

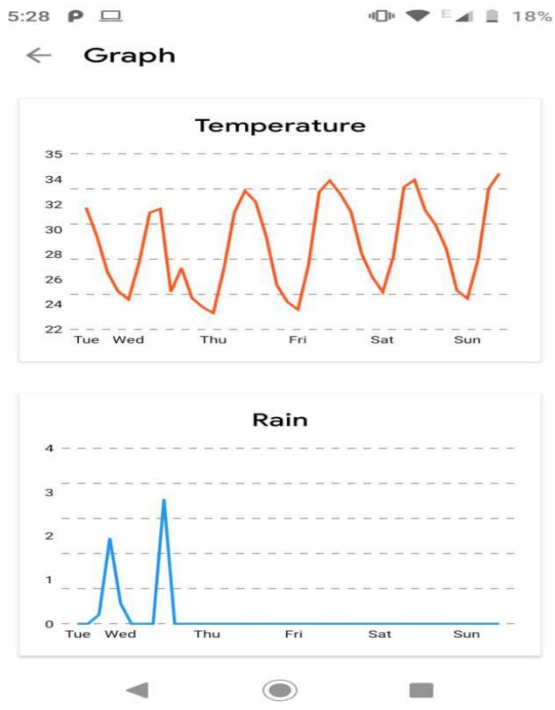


Fig 2.5. Predicting Temperature and Rain

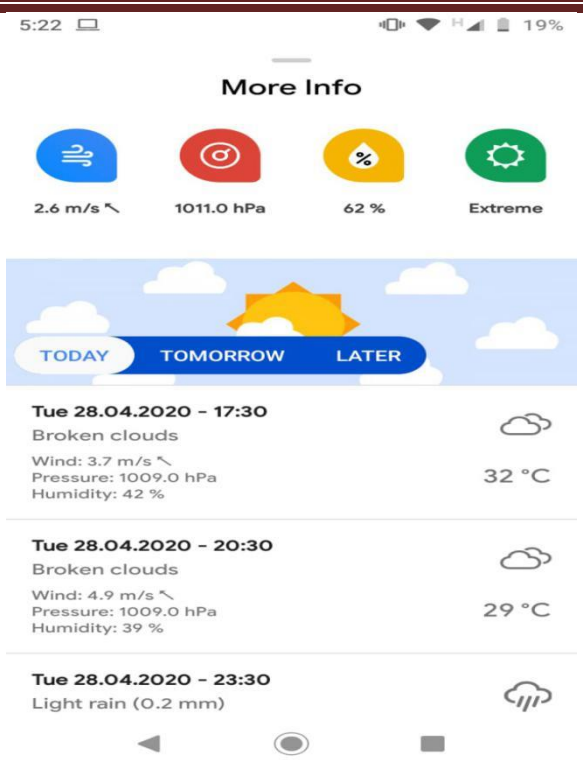


Fig 2.6. Predicting Today's Weather

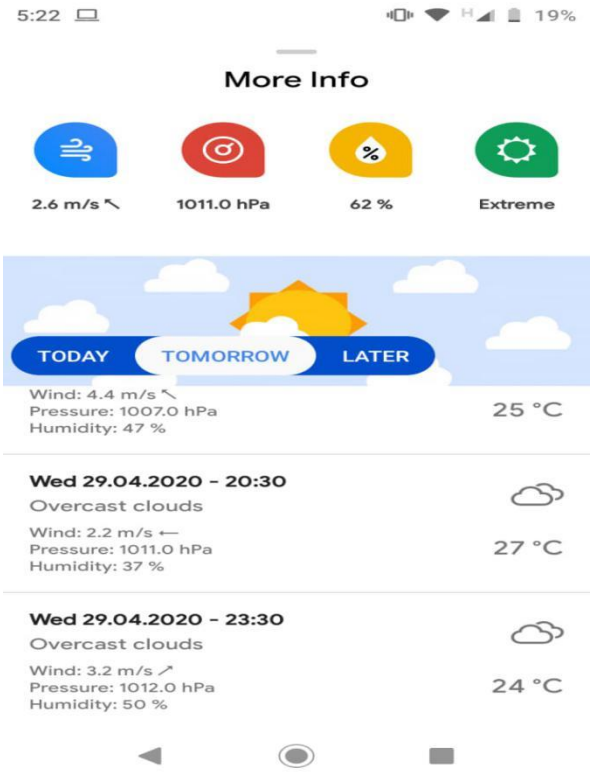


Fig 2.7. Predicting tomorrow's Weather

Fig 2.8. Predicting Later Weather

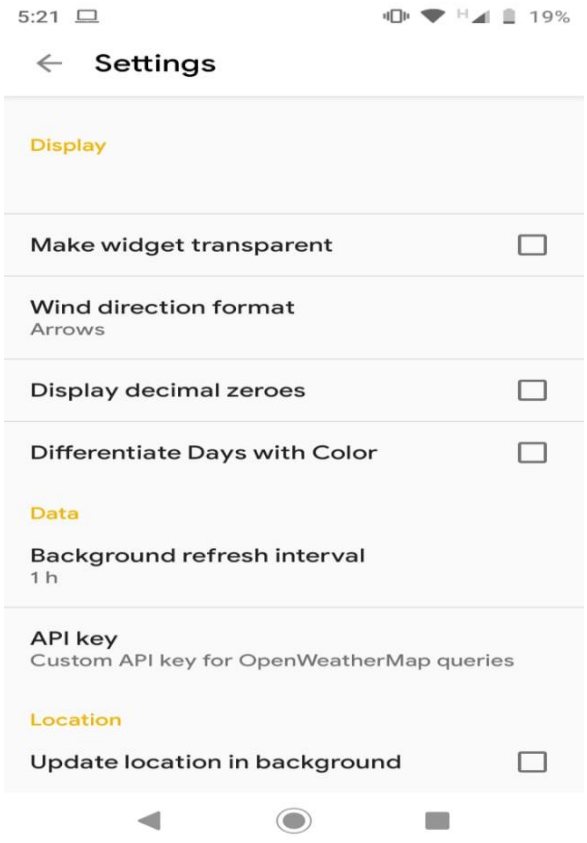


Fig 2.9. Settings Menu

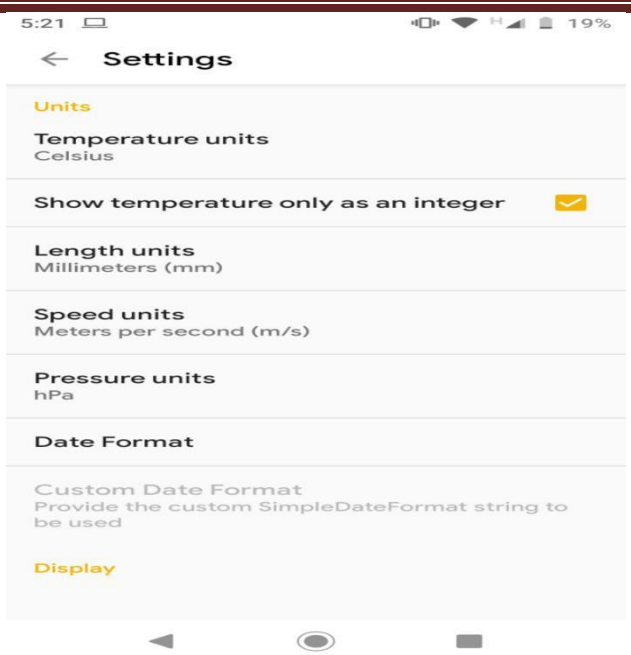


Fig 2.10. Settings Menu Showing Different Units

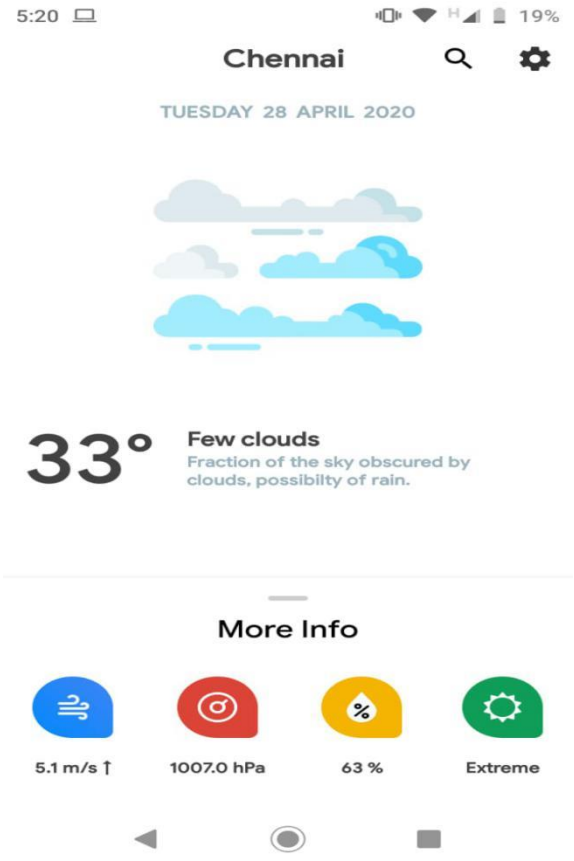


Fig 2.11 Weather Data of Chennai

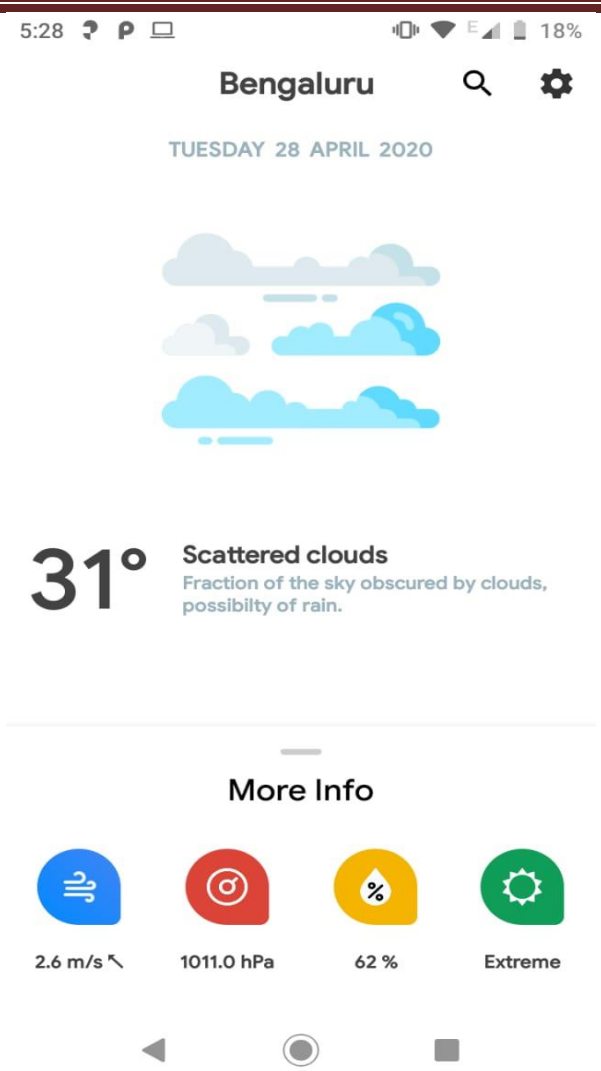


Fig 2.12. Weather Data of Bangalore

CHAPTER 7

TESTING

7.1. TEST STRATEGY

The Die bold-Mariano test for predictive accuracy has been used widely and adapted for economic forecasts, but has not seen much activity in weather forecast verification. The technique is applied to both simulated verification sets as well as weather data at eight stations in Utah, and a loss function based on dynamic time warping (DTW) is used. Results of the simulation experiment show that the DTW technique can be useful if timing errors are the concern.

The project mainly focuses on forecasting weather conditions using historical data. This can be done by extracting knowledge from this given data by using techniques such as association, pattern recognition, nearest neighbor etc.

The testing is completely based on the how accurate the weather conditions are retrieved. The other parameters that the testing is considered over functional testing are:

Admin

- To check for proper locations to be added with their latitude and longitude.
- To check if there is unique generated id for each location.
- To check the database as and when the place is added or deleted.
- To check weather the previous predicted places are shown Etc.

User

- Check for the user and password to be valid.
- Check the location should be selected from the drop down that is updated by the admin.
- Check user location once loaded should show the current weather.
- Check predicted weather should be graphically representation.

7.2. PERFORMANCE CRITERIA

There are three key purposes for carrying out an public weather services criteria check. They are:

(1) Ensuring that public weather services are responding to user requirements

(2) Ensuring the effectiveness and efficiency of the overall public weather services system

(3) Ensuring the overall credibility and proven value of public weather services.

Another way of looking at this, is that the three purposes are about:

(1) Making sure that you are providing the right products

(2) Making sure that you have a good system for making them

Test Environment

We have used NOAA weather and climate tool kit. For the part of the implementation, on which your project focused most, which algorithms you implemented or used and if any modifications were needed to those algorithms or if you did some initial pre-processing, discuss here For the other phases of data mining, discuss briefly. E.g., if you focused most on visualization, you can talk about: which data (Example: downloaded from some website put the URL here; did some survey, then talk about how you did the survey etc) collection approach was used in the project?

Recommendation

- *Extract the location of the user*
- Extract the destination of the user And then recommend the best path according to the conditions.

Test Tools Used

Forecasting a Continuum of Environmental Threats (FACETs)

A Tool to proposed next-generation severe weather watch and warning framework that is modern, flexible, and designed to communicate clear and simple hazardous weather information to serve the public. FACETs supports NOAA's Weather-Ready Nation

initiative to build community resilience in the face of increasing vulnerability to extreme weather and water events.

7.3. RISK IDENTIFICATION AND CONTIGENCY PLANNING

This section identifies the risks that are associated during the testing

Risk #	Risk	Nature of Impact	Contingency Plan
1	Mapping of day to day data	No output generated	Proper data to be checked and updated on the current weather field
2	Data Collection with specific parameters	Unexpected termination	Creating a new dataset with the required parameters

Tab 1.1 risks during testing

If any of the risks occur during testing, following shall be documented

Risk #	Date of Occurrence	Impact of Risk	Action Taken
1	24/03/2019	Wrong prediction	Proper data to be checked and updated on the current weather field

2	25/03/2019	Termination	Creating a new dataset with the required parameters

Tab 1.2 risk occurrence

7.4. TEST SCHEDULE

Test on	Start date	End date
Admin login	20/03/2019	22/03/2019
Admin View place	20/03/2019	22/03/2019
Admin view predict ed place	20/03/2019	22/03/2019
Admin view place	22/03/2019	23/03/2019
User login	24/03/2019	24/03/2019
User predict	24/03/2019	25/03/2019
User previous predicti on	25/03/2019	26/03/2019
Api for current locatio n	26/03/2019	28/03/2019

Prediction check on admin	27/03/2019	29/03/2019
Prediction check on admin	28/03/2019	29/03/2019

Tab 1.3 test schedule

6.5. ACCEPTANCE CRITERIA

This application must serve the purpose by predicting the next day weather which should be shown as a graph with has to be generated with the increase in humidity and degree of the location requested by the user.

Test Case List

This section shall clearly define the test cases that are planned for testing. The following information shall be mentioned: -

TestCase Number	Test Case	Required Output
1	Enters the latitude and longitude of each place	Successful update of the table place
2	Generation of User ID.	Unique id generated for each user
3	Auto generation of Place ID.	Each place has its own id .
4	The predict details page for previous weather Condition.	Previous weather data updated
5	View place on the places table	Check for the added new place

6	Predict for next day	Should provide a graph with humidity and degree generated
7	Predict check for previous days	Should match almost the values that has been stored on the data set

Tab 1.4 acceptance criteria

CHAPTER 8

CONCLUSION

Traditionally, weather forecasting has always been performed by physically simulating the atmosphere as a fluid and then the current state of the atmosphere would be sampled. In the previous system the future state of the atmosphere is computed by solving numerical equations of thermodynamics. But this model is sometimes unstable under disturbances and uncertainties while measuring the initial conditions of the atmosphere. This leads to an incomplete understanding of the atmospheric processes, so it restricts weather predictionn.

Our proposed solution of using Machine learning for weather predicting is relatively robust to most atmospheric disturbances when compared to traditional methods. Another advantage of using machine learning is that it is not dependent on the physical laws of atmospheric processes. In the long run weather prediction using Machine Learning has a lot of advantages and thus it should be used globallyy.

FUTURE SCOPE

Weather forecasts are becoming more detailed, more accurate and are providing the information needed to make sound decisions to protect life and property. Technological advances, such as apps, are making weather information more accessible and immediately alerting those in harm's way. The current version of Weather Prediction that we have developed is still premature. This implies that there are still many limitations that can be resolved and improved. One of the biggest limitation right now is, that the location has to be chosen from the list of places the application is bound to This can be improved if we use web scraping tools to automatically get the weather data, for various locations, from the internet and then input it into the database. Another enhancement that can be done is the automatic validation of longitude and latitude coordinates. Another improvement that can be done is to beautify the UI to make it more appealing to the younger generation. Future scope will make our Weather Prediction more flexible, user friendly and thus it will be more appealing to a wide range of audiencee.

REFERENCES

[1] A. Payne and P. Frow, "A strategic framework for customer relationship management," *J. Marketing*, vol. 69, no. 4, pp. 167–176, Oct. 2005.

[2] L. D. Xu, "Enterprise systems: State-of-the-art and future trends," *IEEE Trans. Ind. Inf.*, vol. 7, no. 4, pp. 630–640, Nov. 2011.

[3] A. Berson, K. Thearling, and S. Smith, *Building Data Mining Applications for CRM*. New York: McGraw-Hill, 1999

[4] K. Coussement and D. V. Poel, "Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques," *Expert Syst. Appl.*, vol. 34, no. 1, pp. 313–327, Jan. 2008.

[5] W. Verbeke, K. Dejaeger, D. Martens, J. Hur, and B. Baesens, "New insights into churn prediction in the telecommunication sector: A profit driven data mining approach," *Eur. J. Oper. Res.*, vol. 218, no. 1, pp. 211–229, Apr. 2012.

[6] W. J. Reinartz and V. Kumar, "The impact of customer relationship characteristics on profitable lifetime duration," *J. Marketing*, vol. 67, no. 1, pp. 77–99, Jan. 2003.

[7] P. Datta, B. Masand, D. R. Mani, and B. Li, "Automated cellular modeling and prediction on a large scale," *Artif. Intell. Rev.*, vol. 14, no. 6, pp. 485–502, Dec. 2000.

[8] D. Popović and B.D. Bašić, "Churn prediction model in retail banking using fuzzy C-means algorithm," *Informatica*, vol. 33, no. 2, pp. 235–239, May 2009.

[9] C.-P. Wei and I. T. Chiu, "Turning telecommunications call details to churn prediction: A data mining approach," *Expert Syst. Appl.*, vol. 23, no. 2, pp. 103–112, Aug. 2002.

[10] M.Owczarczuk, "Churn models for prepaid customers in the cellular telecommunication industry using large datamarts," *Expert Syst. Appl.*, vol. 37, no. 6, pp. 4710–4712, Jun. 2010.

[11] Weather Forecasting Using Deep Learning Techniques - IEEEConference Publication, ieeexplore.ieee.org/document/7415154.

[12] "Weather." Kaggle, www.kaggle.com/tags/weather.

[13] "Weather Forecasting." Kagg le, www.kaggle.com/questions-and-answers/27537.

Trivedi, Shivam. "What Is the Weather Prediction Algorithm? How It Works? What Is the Future?" Medium, Medium, 23 Aug. 2018, medium.com/@shivamtrivedi25/what-is-the- weather-prediction-algorithm-how-it-works-what-is-the-future-a159040dd269.

[14] Vutha, Amar. "Could Machine Learning Mean the End of Understanding in Science?" The Conversation, 25 Oct. 2018, theconversation.com/could-machine-learning-mean-the-end-of- understanding-in-science-98995.

[15] Weather Forecasting Using Deep Learning Techniques - IEEE Conference Publication, ieeexplore.ieee.org/document/7415154.

[16] "Weather Forecasting Using Machine Learning Models and Model Accuracy Assessment." LinkedIn, www.linkedin.com/pulse/weather-forecasting-using-machine-learning-models- model-kinjal-ami.