

**MACHINE LEARNING METHODS OF FLOOD PREDICTION IN WEATHER
DERIVATIVES**

Submitted in partial fulfilment of the requirements for the award of

Bachelor of Engineering Degree

in

Computer Science and Engineering

By

37110365 KORIPELLA SAIMRUNAL

37110363 Koushik



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

Institute of Science and Technology
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | 12B

Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 6000119

MARCH 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with “A” grade by NAAC 12B Status by UGC
Approved by AICTE

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai-600119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **37110365 KORIPELLA SAIMRUNAL** and **37110363 Koushik** who carried out the project entitled “**Machine learning methods of flood prediction in weather derivatives**” under my supervision from November 2020 to March 2021.

Internal Guide

Ms. Yogitha. R, M.TECH., (Ph.D).,

Head of the Department

Submitted for Vivavoce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **37110365 KORIPELLA SAIMRUNAL** and **37110363 Koushik** hereby declare that the Project Report entitled “**Machine Learning Methods of Flood Prediction in Weather Derivatives**” done by me under the guidance of **Ms. Yogitha. R, M.TECH., (Ph.D.,)** at Sathyabama Institute of Science and Technology is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr.SASIKALA. T,M.E., (Ph.D.)** Dean, School of Computing and **Dr.L.Lakshmanan, M.E., (Ph.D.)** and **S. VIGNESHWARI,M.E., Ph.D.**, Head of the Department, Dept. of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mr.S.DHAMODARAN, M.TECH., (Ph.D.)** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

Abstract

Weather forecasting has traditionally been done by physical models of the atmosphere, which are unstable to perturbations, and thus are inaccurate for large periods of time. Since machine learning techniques are more robust to perturbations, in this implementation we explore their application to weather forecasting to potentially generate more accurate weather forecasts for large periods of time. The scope of this project was restricted to forecasting the maximum temperature and the minimum temperature for seven days, given weather data for the past two days. A linear regression model and a variation on a functional regression model were used, with the latter able to capture trends in the weather. The main purpose behind this project was to predict rainfall using data mining techniques. The objective of this concept was to make the farmers aware about the rainfall possibility long before it takes place. As a result, they will be able to face the consequences and the damages will be. Mainly this concept will contribute to the agricultural sector. Each year several areas are submerged underwater due to heavy rainfall. As a consequence, farmland is severely damaged and many cattle farmers suffer from this. Several housings are also deteriorated in storms and floods affecting daily lives. Most of them who suffer are the people who are living in the river side. The sufferers are not only the people of those areas, but also the rest of the country as such situations lead to food crisis. Moreover, this project has been done using machine learning algorithms.

TABLE OF CONTENTS

ABSTRACT

LIST OF FIGURES

Chapter No	Title	Page no.
1.	INTRODUCTION	1
	1.1 INTRODUCTION TO FLOOD FORECASTING	1
	1.2 COMPUTERIZED FLOOD FORECASTING	2
	1.3 MACHINE LEARNING IN FLOOD PREDICTION	2
2.	LITERATURE SURVEY	3
	2.1 LITERATURE SURVEY 1	4
	2.2 LITERATURE SURVEY 2	5
3.	AIM AND SCOPE OF THE INVESTIGATION	6
	3.1 AIM OF THE PROJECT	6
	3.2 PROBLEM STATEMENT	6
	3.2 SCOPE OF THE PROJECT	7
4.	SYSTEM DESIGN AND METHODOLOGY	9
	4.1 EXISTING SYSTEM	8
	4.1.1 DRAWBACKS OF EXISTING SYSTEM	8
	4.2 PROPOSED SYSTEM	8
	4.2.1 ADVANTAGES OF PROPOSED SYSTEM	8

4.3	SYSTEM REQUIREMENTS	9
4.3.1	HARDWARE REQUIREMENTS	9
4.3.2	SOFTWARE REQUIREMENTS	9
4.4	ARCHITECTURAL DESIGN	9
4.5	MODULES	10
4.5.1	DATA COLLECTION	10
4.5.2	DATA SELECTION	10
4.5.3	DATA MINING STAGE	11
4.5.4	WEATHER PREDICTION/ FORECAST	11
4.5.5	METEOROLOGISTS IN WEATHER FORECAST	11
4.6	DATA FLOW DIAGRAMS	12
4.6.1	LEVEL 0	12
4.6.2	LEVEL 1	13
4.6.3	LEVEL 2	13
4.6.4	LEVEL 3	14
5.	EXECUTION AND RESULTS	15
5.1	EXECUTING THE PROJECT	15
5.2	THE RESULT	15
6	SUMMARY AND CONCLUSION	23
6.1	SUMMARY	23
6.2	CONCLUSION	23
	APPENDIX	24
	(A) SOURCE CODE	

LIST OF FIGURES

FIGURE NUMBER	FIGURE	PAGE NUMBER
4.4	Architecture Diagram	8
4.6.1	Level 0 Diagram	10
4.6.2	Level 1 Diagram	11
4.6.3	Level 2 Diagram	11
4.6.4	Level 3 Diagram	12
5.1	Sequence Diagram	13
5.2.1	Dataset of Maharashtra	14
5.2.2	Data Head and Description Maharashtra	14
5.2.3	Overview of the dataset	15
5.3.1	Areas where flood is predicted	16
5.3.2	Current precipitation data across the nation	16
5.3.3	Damage analysis of the flood	17
5.3.4	The cost and damage analysis, based of the flood risk prediction	17

5.3.5	ML powered prediction of where a flood is most likely to occur	18
5.3.6	Satellite image of Mumbai	19
5.3.7	Weather Forecast Home Screen	19
5.3.8	Result of Tambaram Weather	20

CHAPTER 1

INTRODUCTION

1.1 Introduction to Flood Forecasting

Natural flood is one of the most recurrent disasters. Unlike stagnant water discharge, occasionally experienced in poorly planned cities, major flood incidents always cause considerable damages to properties and loss of lives. Furthermore, overflow from main rivers along shore sides to surrounding basins can greatly spread the damages. Although being located further away from a river, an area with inappropriate land uses are unable to efficiently discharge accumulated precipitation, and hence are inevitably prone to even more frequent floods. Regardless of causes, however, a flood is generally sudden and thus almost formidable for the general public and relevant organization to be adequately prepared for the incident. Despite the recent extensive development of computerized flood forecasting systems, they remained based primarily on present precipitation, monitored by rain stations or rain gauges. These facilities are normally owned by a meteorology department or similar organizations. Besides, they are scantily located in a few areas due to costly installation and maintenance. Hence, it is difficult to determine precipitation or predict flood accurately, especially in areas with no such facility. To remedy this issue, precipitation in these areas were typically estimated either by inter or extrapolation from those with rain stations present.

1.2 Computerized Flood Forecasting

Conventional meteorological readings, e.g., precipitation, temperature, and humidity, etc., took a long time to measure, process, record, and transfer to relevant organizations. Analyses based on past precipitation were known to be associated with several shortcomings. For instance, they contribute to inaccurate and often outdated flood prediction. Limited sample size, inadequate computing capability, and inefficient prediction methods were all undermining the real potential of this scheme. Nonetheless, with the recent advances in distributed computing and especially modern

machine learning (ML), resembling human intelligence, computerized flood forecasting, based on thematic factors has widely been investigated. In addition, as the number of both open and proprietary data providers escalates, Big Data has now become a central source of information in such pursuits.

1.3 Machine Learning In Flood Prediction

In addition, there have been recent developments in flood forecasting systems based on ML. These systems embedded both attributes and crowdsourcing data into their ML frameworks. However, most existing systems operate by analysing these data offline on premise before presenting their prediction results on various platforms. A typical practice was proposed in, where an ML was trained with real-time rainfalls, streamflow, and other data. It was unclear, nonetheless, how a prediction result was verified against an actual event, which was obtained from crowdsourcing. Therefore, this project proposes a novel flood forecasting system based on fusing meteorological, hydrological, geospatial, as well as crowdsourcing data, and integrating them into an ML framework. These data were compiled from various big data platforms, by using online application programming interfaces (API). The forecasting mechanism was driven by a machine learning strategy. To determine the most suitable one for the task, several state-of-the-art MLs, i.e., decision tree, random forest, naïve Bayes, artificial neural networks, support vector machine, and fuzzy logic, were compared. It is worth emphasizing that the novelty of this paper was not only to use different data in an ML, but also to enhance and verify its predictions based on crowdsourcing ones. It will be later shown in the experiments that the developed system was able to elevate known limitations and to enhance the effectiveness and efficiency of computerized flood forecasting .

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY 1

Indian Summer Monsoon Rainfall (ISMR) Forecasting using Time Series Data Authors Pritpal Singh Published Year 2015 .

A Fuzzy-Entropy-Neuro based Expert System In authors presented a model to forecast Indian Summer Monsoon Rainfall on the basis of monthly and seasonal timescales. To forecast, time series dataset was used, spanning from 1871 till 2014. The dataset was classified in two parts (1) 1960-2014 used as training data, and (2) 2015-2020 used as test data. Statistical analysis reported the dynamic nature of rainfall in monsoon, which could not be predicted effectively with mathematical and statistical models. So, the authors in this research recommended to use three techniques for this type of prediction: Fuzzy Set, Entropy and Artificial Neural Network. By using these three techniques, a forecasting model is developed to deal with the dynamic nature of the ISMR. In proposed model, fuzzy set theory is used to handle uncertainties which are inherited in dataset. The entropy computational concept was modified in this model and used to provide the input as a degree of membership in the entropy function. That entropy function was referred as Fuzzy Information-Gain (FIG). Then, each fuzzified rule was defuzzied using the ANN. The value of FIG of each fuzzy- set was then used as input into ANN. The proposed model was named as “Fuzzy-Entropy-Neuro Based Expert System for ISMR Forecasting” because it is the integration of fuzzy set, entropy and ANN. To evaluate the performance of proposed model following accuracy measures were used: Standard Deviations (SDs), Correlation Coefficient (CC), Root Mean Square Error (RMSE) and Performance Parameter (PP). According to results the proposed model is effective and efficient in comparison with other existing models.

2.2 LITERATURE SURVEY 2

An Extensive Evaluation of Seven Machine Learning Methods for Rainfall Prediction in Weather Derivatives: Authors Sam Cramer, MichealKampouridis, Alex A. Freitas, Antonis K. Alexandridis Published Year 2017.

The researchers in compared the predictive performance of latest and state of the art method named “Markov chain extended with rainfall prediction” with the other widely used machine learning techniques: Support Vector Regression, Genetic Programming, M5 Rules, M5 Model trees, Radial Basis Neural Networks, and k-Nearest Neighbours. Daily rainfall datasets were collected from 42 cities of two continents, with very diverse climatic features. 20 cities were selected from around the Europe and 22 from around the USA. There were two reasons of choosing two continents for data extraction, first is to perform the experiment on different climates having diverse weather and second was the geographical locations as the selected cities were very far apart from each other. The ultimate goal was to not bias the experiment to particular climate type or for particular geographic location. According to results the accumulating rainfall amounts can bring good results as compared to prediction using daily rainy data. While using the accumulated data, Support Vector Regression, Radial Basis Functions, and Genetic Programming overall performed well however Radial Basis Functions performed better then modern technique of “Markov chain”. For all selected datasets, each technique used the same parameters so it was not guaranteed that the best possible set of parameters was used for all the techniques. During the experiment, the researchers have noted a relationship between predictive accuracy and climatic attributes such as: volatile nature of rainfall, amount of maximum rainfall and the interquartile range of rainfall. Moreover no significant difference was noted in algorithms’ prediction error among the cities of both the continents (USA and Europe). Issue regarding the discontinuity in rainfall data was solved with the help of accumulated rainfall amounts.

CHAPTER 3

AIM AND SCOPE OF THE INVESTIGATION

3.1 AIM OF THE PROJECT

Ours is a developing country. One of the key factors behind the economic growth of our country is agriculture. Since agriculture directly depends on rainfall, it is crucial to anticipate it earlier and take necessary measures so that it does not spread any adverse effects. With an intention of helping our agriculture, we have done our study in this area and came up with this research work consequently. Another significant reason for this research to be conducted is flood. For the past four decades, flood has been a major thorn in the path of our economic development. Almost every year, several parts of this country have to stay underwater for a long time which damages the crops and washes away all the hard work of the farmers. Its prevention can only be done by anticipating it earlier and taking measures accordingly.

3.2 PROBLEM STATEMENT

In our project, nearly thousand data instances containing values like rainfall, temperature, air pressure, year etc. have been used and analysed with the help of different algorithms for getting a proper prediction. Collecting all the data was quite difficult as we had revised the data from Bangladesh Meteorological Department (BMD) for rainfall data.. After that we collected other data from different reliable sites. Getting more accurate data was important for us. Finding out the site where we could get the actual data was difficult. Moreover, none of us was so expert in programming. As a result we had to go through tough times to understand the codes well and work based on that. Giving input in an accurate way and knowing what is the expected output and what we are getting was also an important task for us. We had to go through many trials and errors in this regard during working with those algorithms. Extensive amount of research was done to understand the basic of the algorithms. Furthermore, learning how to work with big data in excel was a bit new experience.

3.3SCOPE OF THE PROJECT

Flood is important for food production plan, water resource management and all activity plans in the nature. The occurrence of prolonged dry period or heavy rain at the critical stages of the crop growth and development may lead to significant reduce crop yield. India is an agricultural country and its economy is largely based upon crop productivity. Thus rainfall prediction becomes a significant factor in agricultural countries like India. Flood forecasting has been one of the most scientifically and technologically challenging problems around the world in the last century.

CHAPTER 4

SYSTEM DESIGN & METHODOLOGY

4.1 EXISTING SYSTEM

Agriculture is the strength of our Indian economy. . Farmer only depends upon monsoon to be their cultivation. The good crop productivity needs good soil, fertilizer and also good climate. Weather forecasting is the very important requirement of the each farmer. Due to the sudden changes in climate/weather, The people are suffered economically and physically. Weather prediction is one of the challenging problems in current state. The main motivation of this paper to predict the weather using various data mining techniques. Such as classification, clustering, decision tree and also neural networks. Weather related information is also called the meteorological data. In this project the mostly used weather parameters are rainfall, wind speed, temperature and cold.

4.1.1 DRAWBACKS OF EXISTING SYSTEM

- Classification
- Clustering
- Decision tree

4.2 PROPOSED SYSTEM

Flood is important for food production plan, water resource management and all activity plans in the nature. The occurrence of prolonged dry period or heavy rain at the critical stages of the crop growth and development may lead to significant reduce crop yield. India is an agricultural country and its economy is largely based upon crop productivity. Thus rainfall prediction becomes a significant factor in agricultural countries like India. Flood forecasting has been one of the most scientifically and technologically challenging problems around the world in the last century.

4.2.1 ADVANTAGES OF PROPOSED SYSTEM

4.3 SYSTEM REQUIREMENTS

4.3.1 HARDWARE REQUIREMENTS:

- System - Windows 7/10
- Speed - 2.4GHZ
- Hard disk - 40GB
- Monitor - 15VGA color
- RAM - 4GB

4.3.2 SOFTWARE REQUIREMENTS:

- Operating System - Windows XP
- Coding language - PYTHON
- IDE - Jupyter Notebook(Anaconda)

4.4 ARCHITECTURAL DESIGN

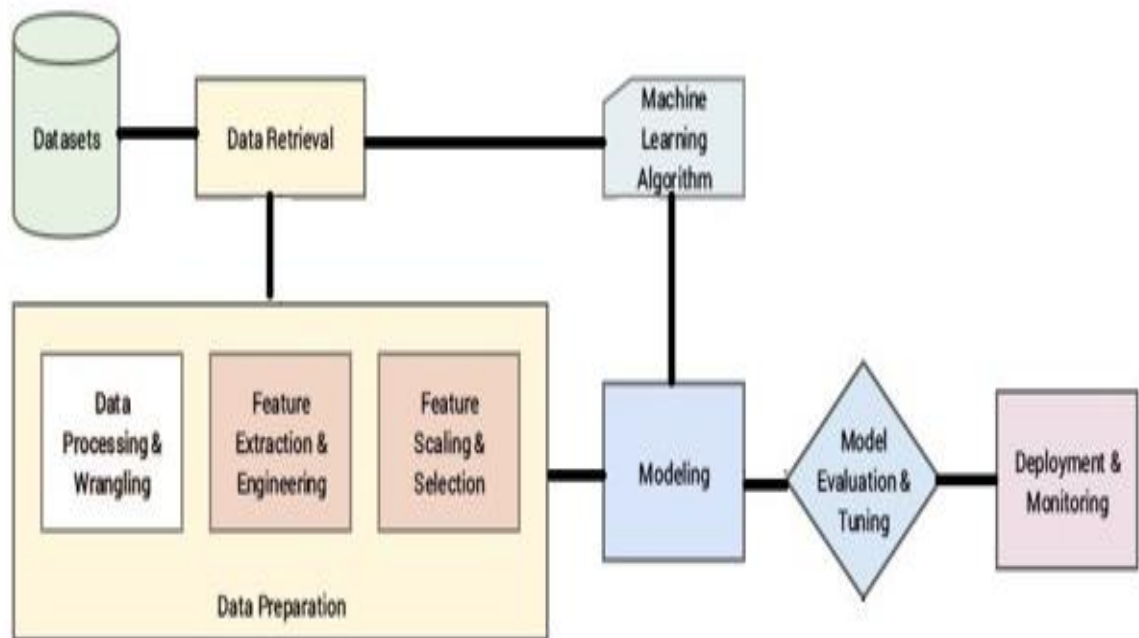


Fig :4.4 architecture diagram

4.5 MODULES

4.5.1 Data Collection

Data collection is the process of gathering and measuring information from countless different sources. In order to use the data we collect to develop practical artificial intelligence (AI) and machine learning solutions, it must be collected and stored in a way that makes sense for the business problem at hand.

4.5.2 Data Selection

At this stage, data relevant to the analysis like decision tree was decided on and retrieved from the dataset. The Meteorological dataset had ten attributes in that were using two attributes for future prediction. Due to the nature of the Cloud Form data where all the values are the same and the high percentage of missing values in the sunshine data both were not used in the analysis.

4.5.3 Data Mining Stage

The data mining stage was divided into three phases. At each phase all the algorithms were used to analyse the meteorological datasets. The testing method adopted for this research was percentage split that train on a percentage of the dataset, cross validate on it and test on the remaining percentage. There after interesting patterns representing knowledge were identified.

4.5.4 Weather prediction /Forecast

Weather Forecasting serves many purposes and needs. It can help people and organizations to plan for the future and to make rational decisions. Goal of atmospheric research Weather prediction is said to be the ultimate goal of atmospheric research. It is the most advanced area in the subject and application of meteorology. First, to make an accurate forecast, a meteorologist must first understand what processes are occurring in the atmosphere to produce the current weather at the location for which the meteorologist is forecasting.

4.5.5 Meteorologists in weather Forecasting

If meteorologists can understand how the atmosphere changes over time in response to various factors, they can write the governing mathematical equations to express these changes. These equations are developed into numerical models to find out how the atmosphere is changing and will appear in the output from these models

can be used as an aid to forecasters in preparing the forecasts both in short range or long range.

4.6 DATA FLOW DIAGRAM

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 input data to the system, various processing carried out on this data, and the output data is generated by this system. The data flow diagram (DFD) is one of the most important modelling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

4.6.1 Level 0:

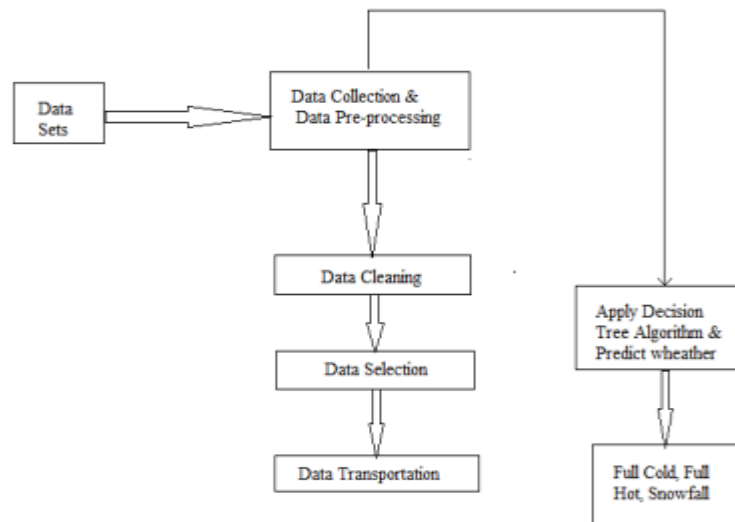


Fig: 4.6.1 Level0 diagram

4.6.2 Level 1:

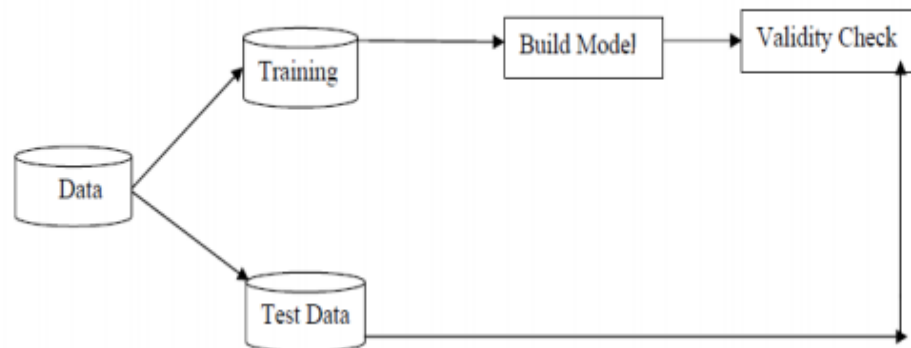


Fig: 4.6.2 Level 1 diagram

4.6.3 Level 2:

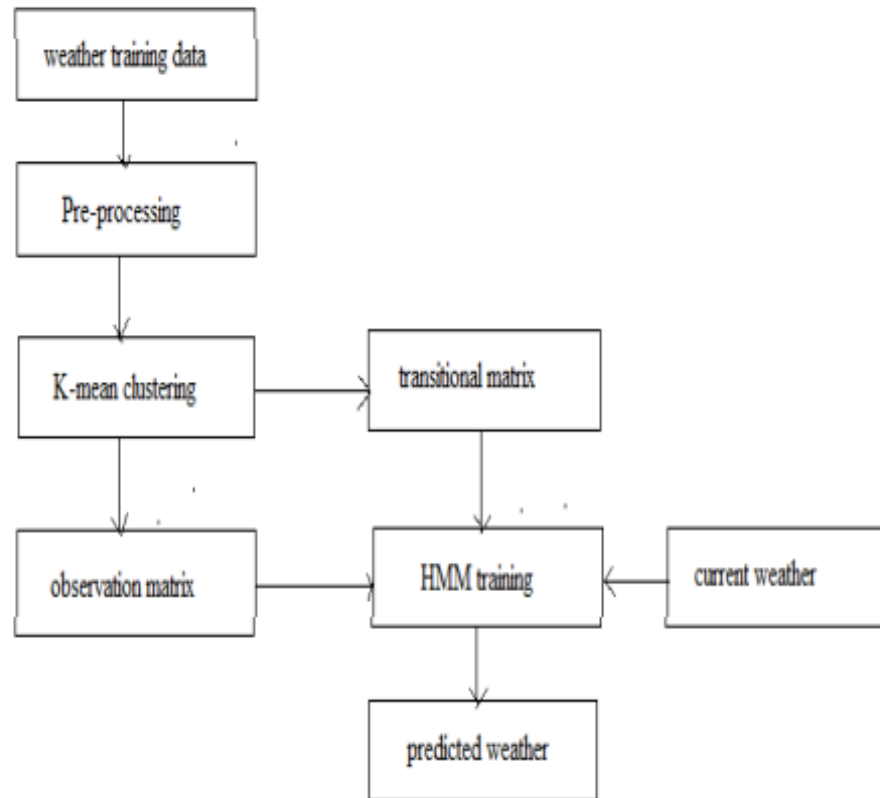


Fig:4.6.3 Level 2 Diagram

4.6.4 Level 3:

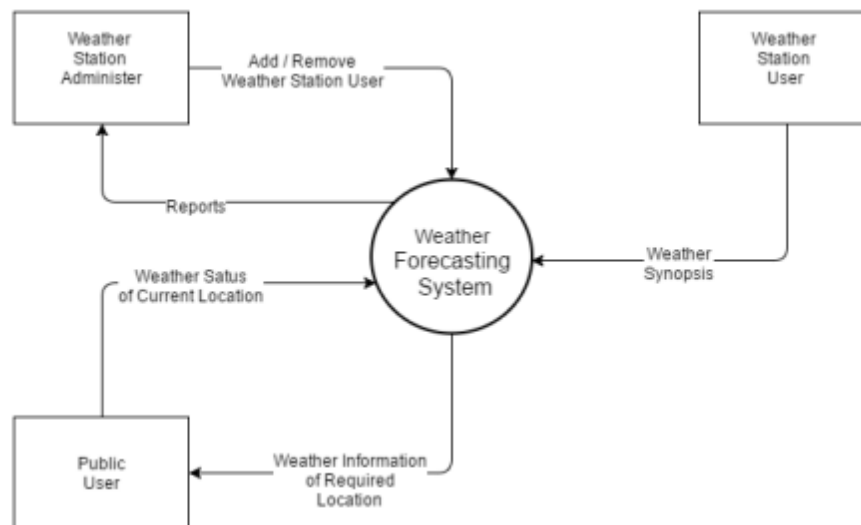


Fig:4.6.4 Level 3 Diagram

CHAPTER 5

EXECUTION AND RESULTS

5.1 EXECUTION OF THE PROJECT

The end project is run in the Jupyter notebook, For the inputs the program will collect the type of the data sets they are asked from the datasets to show the output from it. It will validate in the layer wise format to get the result. As shown in below figure

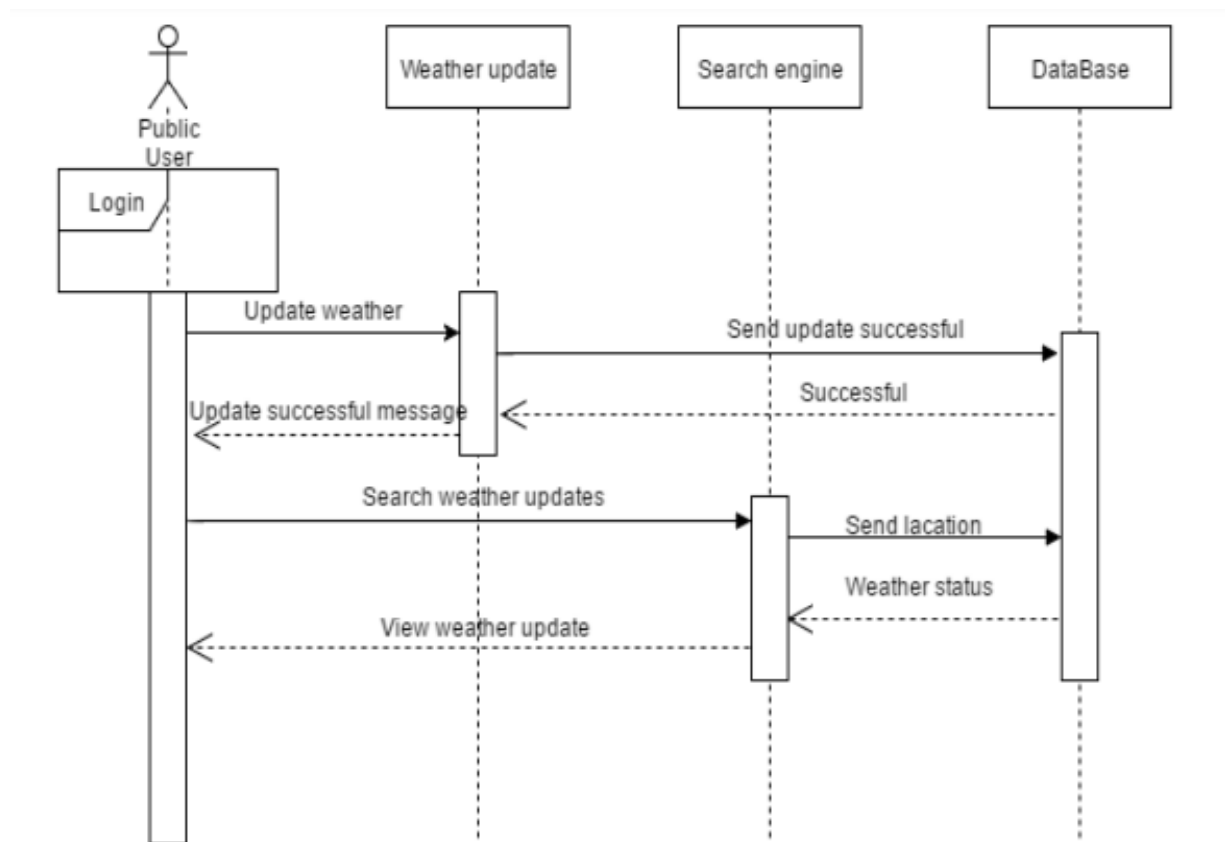


Fig:5.1Sequence diagram

5.2 RESULTS

The screenshot shows a Jupyter Notebook interface with the following code and output:

```
In [2]: fn = 'MADHYA MAHARASHTRA'
```

```
In [3]: data = pd.read_csv('Data/'+fn+'.csv')
data = data.fillna(data.mean())
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 115 entries, 0 to 114
Data columns (total 20 columns):
Unnamed: 0      115 non-null int64
SUBDIVISION     115 non-null object
YEAR            115 non-null int64
JAN             115 non-null float64
FEB             115 non-null float64
MAR             115 non-null float64
APR             115 non-null float64
MAY             115 non-null float64
JUN             115 non-null float64
JUL             115 non-null float64
AUG             115 non-null float64
SEP             115 non-null float64
OCT             115 non-null float64
NOV             115 non-null float64
DEC             115 non-null float64
ANNUAL          115 non-null float64
Jan-Feb         115 non-null float64
Mar-May         115 non-null float64
Jun-Sep         115 non-null float64
Oct-Dec         115 non-null float64
dtypes: float64(17), int64(2), object(1)
memory usage: 18.1+ KB
```

Fig:5.2.1 Dataset of coastal Maharashtra

The screenshot shows a Jupyter Notebook interface with the following code and output:

```
In [5]: data.describe()
```

Out[5]:

	Unnamed: 0	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
count	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000
mean	2679.000000	1958.000000	3.054783	1.467826	3.596522	9.146957	22.943478	147.426087	248.980000	184.397391	157.221739	70.194783
std	33.341666	33.341666	6.666652	2.915282	6.411625	9.402000	22.368097	55.088562	73.776111	67.497544	64.154052	47.520595
min	2622.000000	1901.000000	0.000000	0.000000	0.000000	0.000000	0.300000	30.700000	68.400000	59.500000	29.900000	3.400000
25%	2650.500000	1929.500000	0.000000	0.000000	0.200000	3.200000	7.350000	109.650000	198.800000	139.050000	102.900000	36.700000
50%	2679.000000	1958.000000	0.700000	0.200000	1.500000	6.300000	15.200000	146.400000	247.000000	186.600000	161.300000	57.700000
75%	2707.500000	1986.500000	2.600000	1.550000	4.200000	12.100000	33.000000	179.900000	304.600000	230.900000	200.700000	95.200000
max	2736.000000	2015.000000	41.500000	20.000000	41.200000	54.500000	104.200000	293.800000	421.700000	421.700000	324.800000	196.000000

Fig: 5.2.2 Data Head And Distribution of Maharashtra

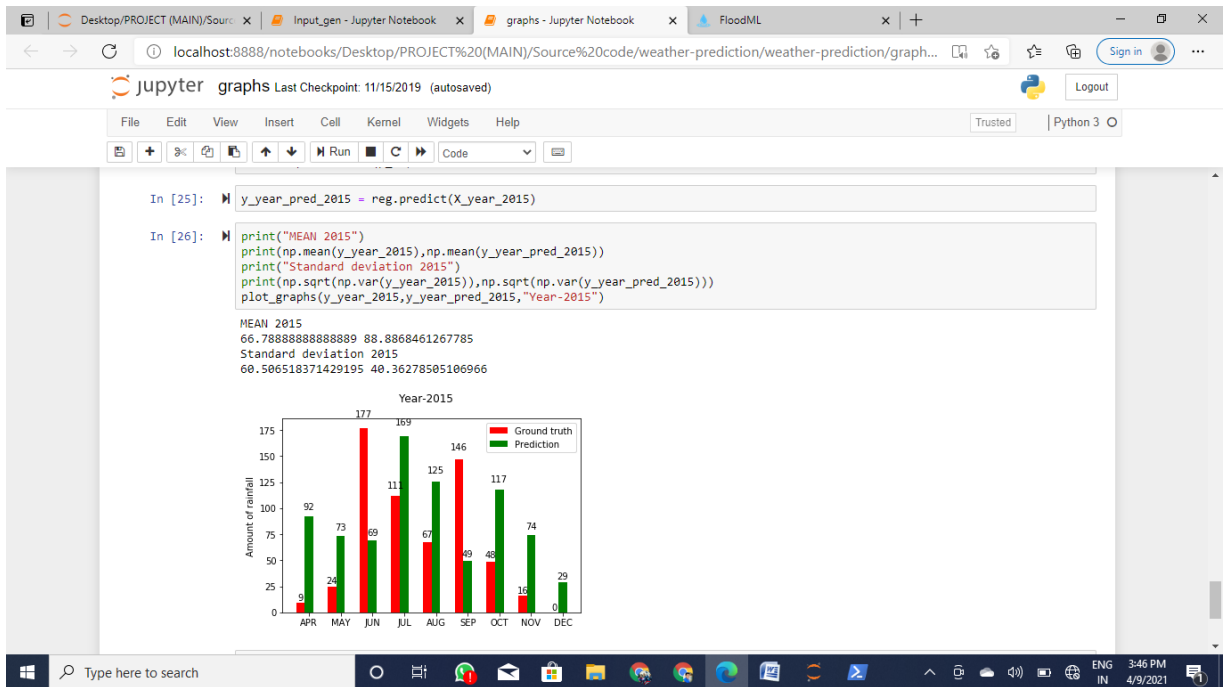


Fig:5.2.6 Overview Of The Dataset

5.3 Outputs

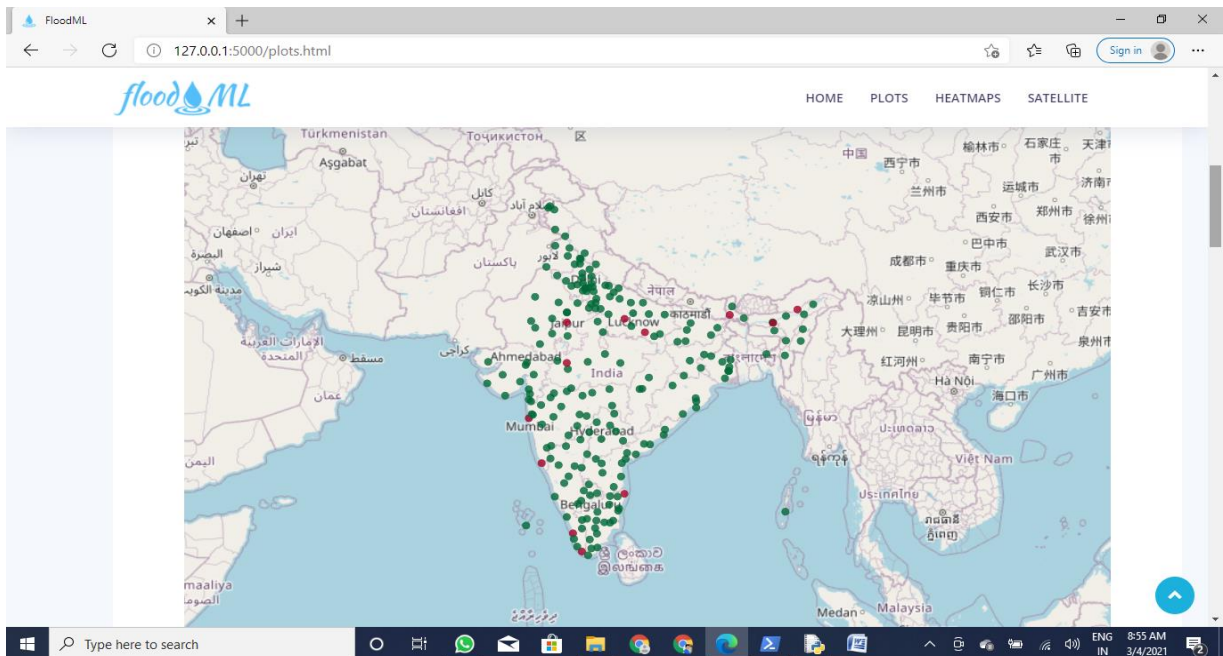


Fig:5.3.1 Areas Where the flood is predicted

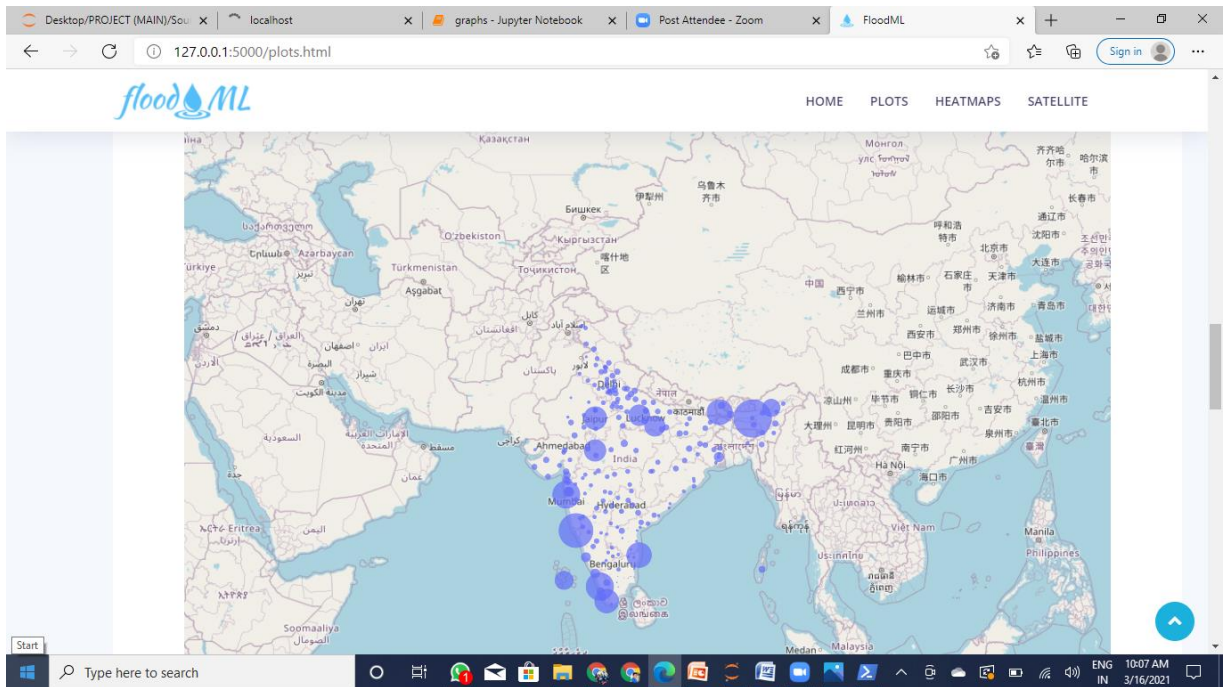


Fig:5.3.2Current precipitation data across the nation

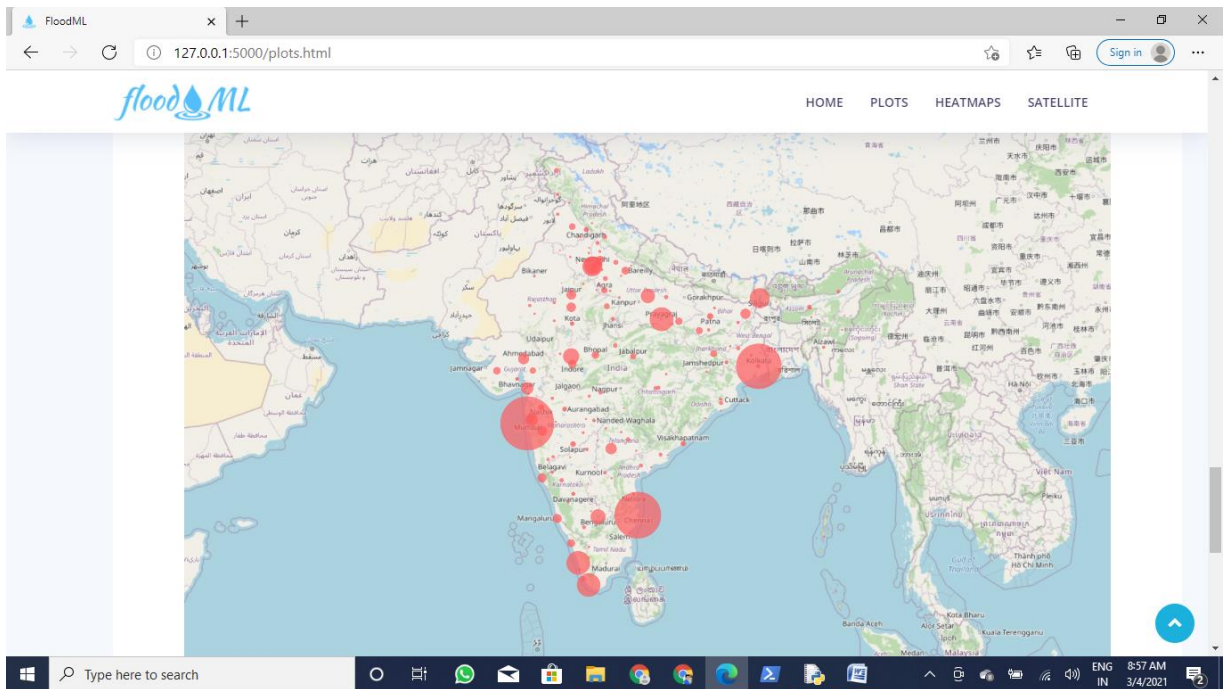


Fig:5.3.3 Damage Analysis Of The Flood

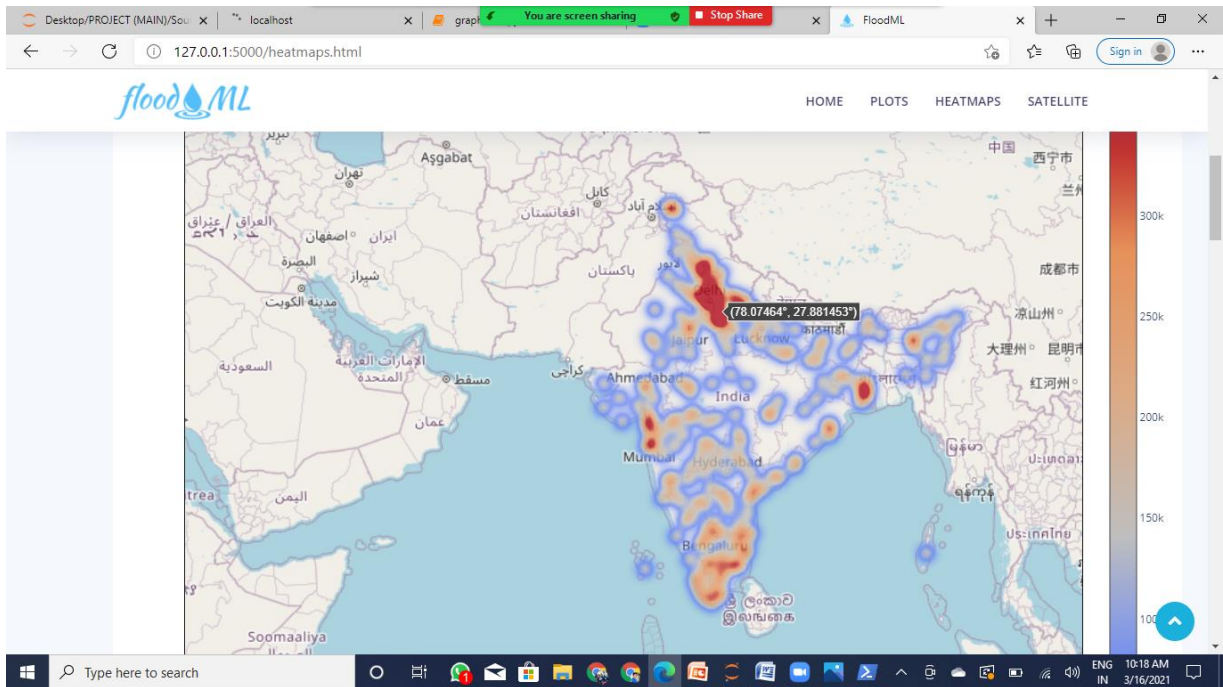


Fig:5.3.4 The cost and damage analysis, based on the flood risk prediction.

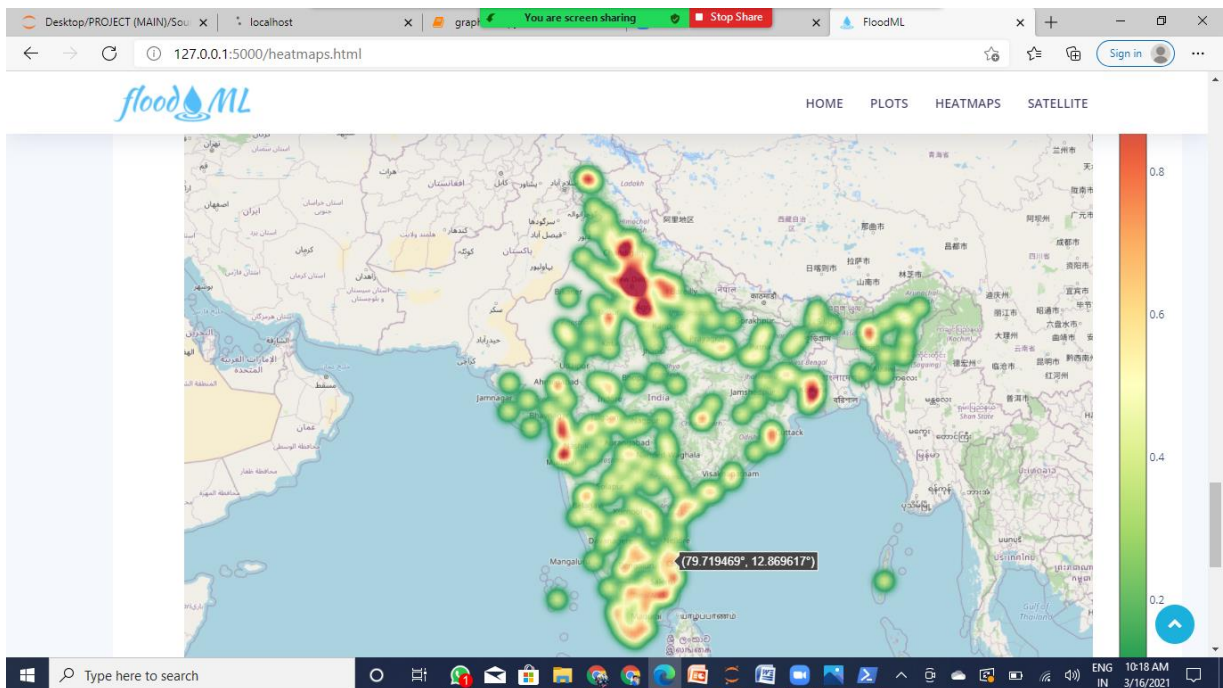


Fig:5.3.5 ML powered prediction of where a flood is most likely to occur.

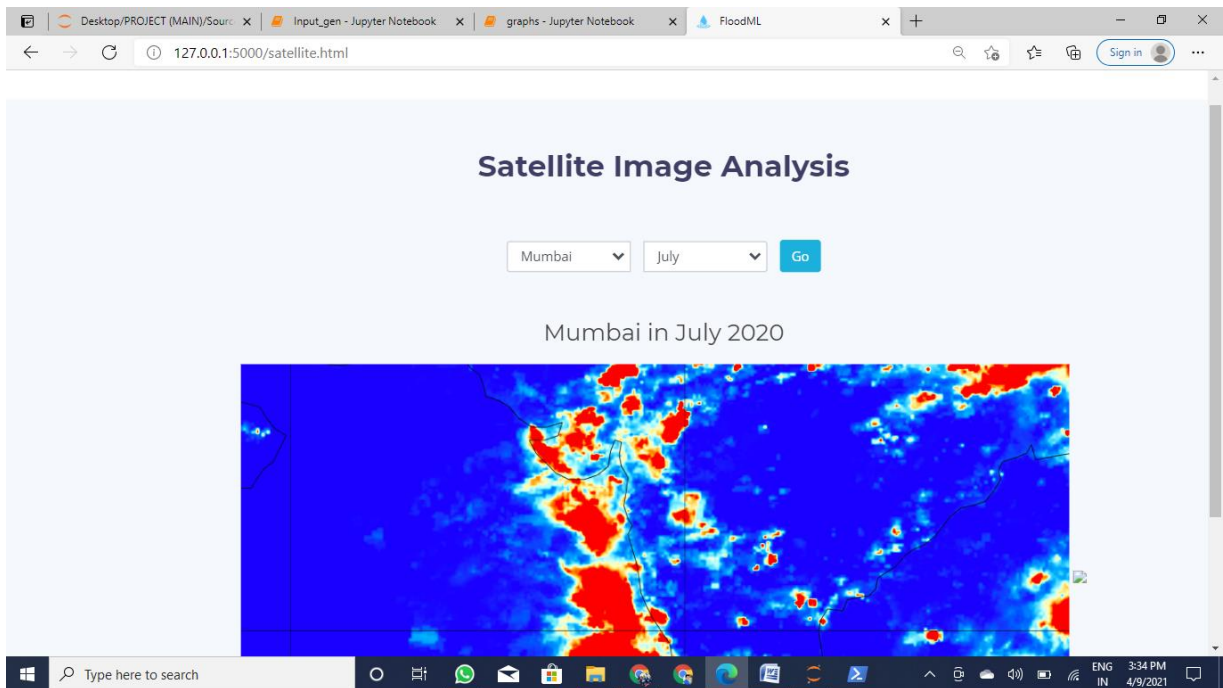


Fig:5.3.6 Satellite Image Of Mumbai

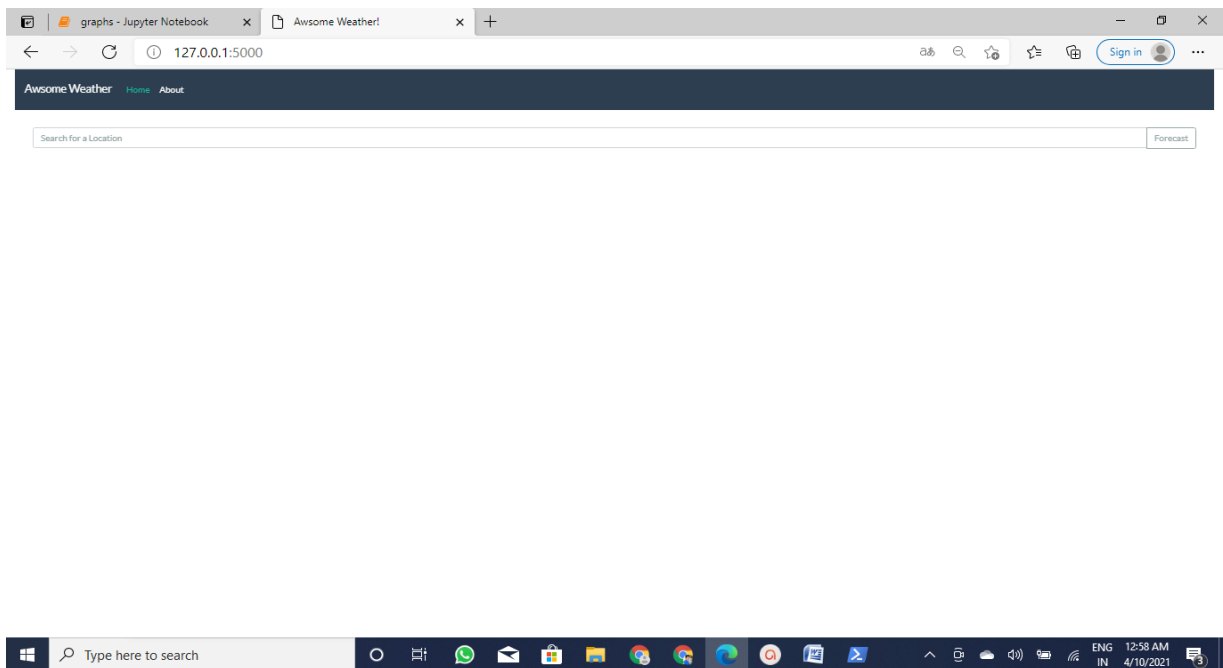


Fig:5.3.7 Weather Forecasting Home Screen

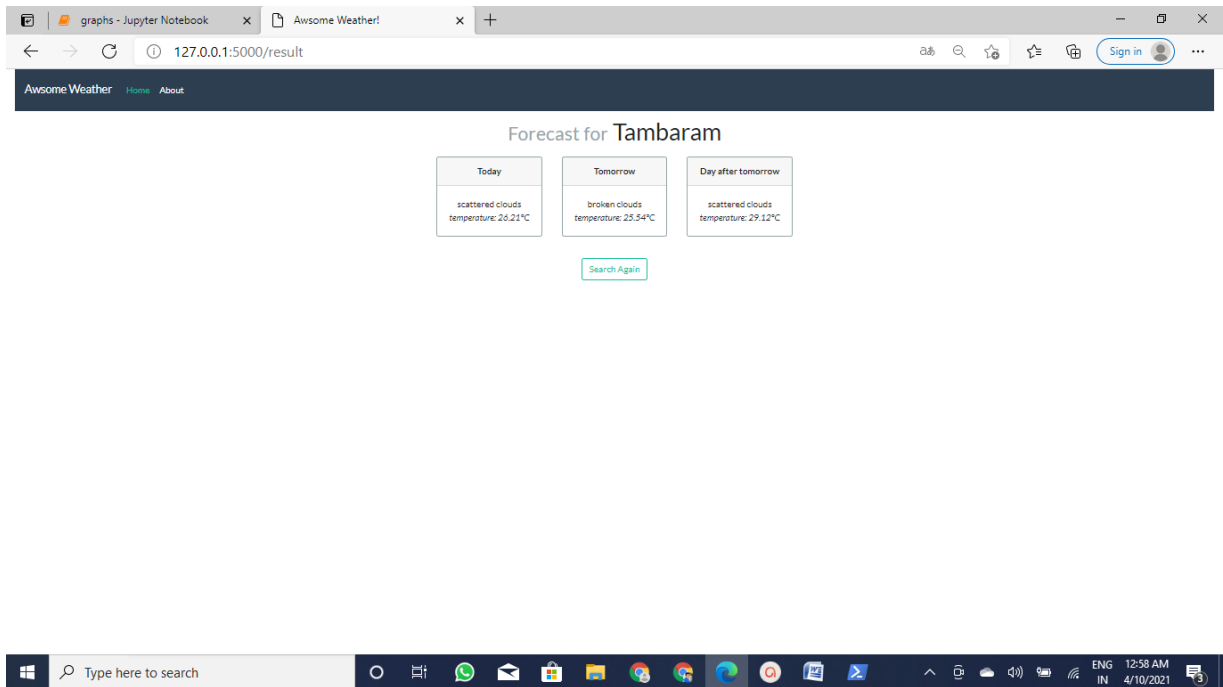


Fig:5.3.8 Result of Tambaram Weather

CHAPTER 6

SUMMARY AND CONCLUSION

6.1 SUMMARY

As it was discussed earlier, this application was made to restore damaged images, remove unwanted objects on the image with Pytorch and Contextual Attention Technique.

6.2 CONCLUSION

Weather forecasting is a meteorological work that easy to modify researcher work by applying the numerical weather prediction method. weather forecasted by using various data mining techniques especially classification clustering and neural network, decision tree. the key aim for improving the classification and prediction performance for the traditional; weather prediction model is designed and developed in this work. but some limitation of the model is also observed, thus in near future need to be review before use of the proposed technique. and also soil there are some issues and challenges in which better implement of data mining technique should be implemented in field of weather forecasting.

APPENDIX

(A) SOURCE CODE

MAIN PAGE

```
"""Web app."""
import flask
from flask import Flask, render_template, request, redirect, url_for

import pickle
import base64
from training import prediction
import requests
app = flask.Flask(__name__)

data = [{'name':'Delhi', "sel": "selected"}, {'name':'Mumbai', "sel": ""}, {'name':'Kolkata',
"sel": ""}, {'name':'Bangalore', "sel": ""}, {'name':'Chennai', "sel": ""}]

# data = [{'name':'India', "sel": ""}]

months = [{"name":"May", "sel": ""}, {"name":"June", "sel": ""}, {"name":"July", "sel":
"selected"}]

cities = [{'name':'Delhi', "sel": "selected"}, {'name':'Mumbai', "sel": ""}, {'name':'Kolkata',
"sel": ""}, {'name':'Bangalore', "sel": ""}, {'name':'Chennai', "sel": ""}, {'name':'New
York', "sel": ""}, {'name':'Los Angeles', "sel": ""}, {'name':'London', "sel": ""},
{'name':'Paris', "sel": ""}, {'name':'Sydney', "sel": ""}, {'name':'Beijing', "sel": ""}]

model = pickle.load(open("model.pickle", 'rb'))

@app.route("/")
@app.route('/index.html')
```

```

def index() -> str:
    """Base page."""
    return flask.render_template("index.html")

@app.route('/plots.html')
def plots():
    return render_template('plots.html')

@app.route('/heatmaps.html')
def heatmaps():
    return render_template('heatmaps.html')

@app.route('/satellite.html')
def satellite():
    direc = "processed_satellite_images/Delhi_July.png"
    with open(direc, "rb") as image_file:
        image = base64.b64encode(image_file.read())
        image = image.decode('utf-8')
    return render_template('satellite.html', data=data, image_file=image,
months=months, text="Delhi in January 2020")

@app.route('/satellite.html', methods=['GET', 'POST'])
def satelliteimages():
    place = request.form.get('place')
    date = request.form.get('date')
    data = [{'name':'Delhi', "sel": ""}, {'name':'Mumbai', "sel": ""}, {'name':'Kolkata', "sel": ""}, {'name':'Bangalore', "sel": ""}, {'name':'Chennai', "sel": ""}]
    months = [{"name":"May", "sel": ""}, {"name":"June", "sel": ""}, {"name":"July", "sel": ""}]

```

```

for item in data:
    if item["name"] == place:
        item["sel"] = "selected"

for item in months:
    if item["name"] == date:
        item["sel"] = "selected"

text = place + " in " + date + " 2020"

direc = "processed_satellite_images/{_}.png".format(place, date)
with open(direc, "rb") as image_file:
    image = base64.b64encode(image_file.read())
image = image.decode('utf-8')
return render_template('satellite.html', data=data, image_file=image,
months=months, text=text)

@app.route('/predicts.html')
def predicts():
    return render_template('predicts.html', cities=cities, cityname="Information about
the city")

@app.route('/predicts.html', methods=["GET", "POST"])
def get_predicts():
    try:
cityname = request.form["city"]

        cities = [{'name':'Delhi', "sel": ""}, {'name':'Mumbai', "sel": ""}, {'name':'Kolkata',
"sel": ""}, {'name':'Bangalore', "sel": ""}, {'name':'Chennai', "sel": ""}, {'name':'New
York', "sel": ""}, {'name':'Los Angeles', "sel": ""}, {'name':'London', "sel": ""},
{'name':'Paris', "sel": ""}, {'name':'Sydney', "sel": ""}, {'name':'Beijing', "sel": ""}]

```

```

for item in cities:
    if item['name'] == cityname:
        item['sel'] = 'selected'
print(cityname)
URL = "https://geocode.search.hereapi.com/v1/geocode"
location = cityname
api_key = 'LHtLOXHyYWrAYEFNbXIQg8BM6PEY8Jryp88mmHYvAGw' # Acquire
from developer.here.com

PARAMS = {'apikey':api_key,'q':location}
# sending get request and saving the response as response object
r = requests.get(url = URL, params = PARAMS)
data = r.json()
latitude = data['items'][0]['position']['lat']
longitude = data['items'][0]['position']['lng']
final = prediction.get_data(latitude, longitude)

final[4] *= 15
if str(model.predict([final])[0]) == "0":
    pred = "Safe"
else:
    pred = "Unsafe"

    return render_template('predicts.html', cityname="Information about " +
cityname, cities=cities, temp=round(final[0], 2), maxt=round(final[1], 2),
wspd=round(final[2], 2), cloudcover=round(final[3], 2), percip=round(final[4], 2),
humidity=round(final[5], 2), pred = pred)
except:
    return render_template('predicts.html', cities=cities, cityname="Oops, we weren't
able to retrieve data for that city.")

```

```
if __name__ == "__main__":  
    app.run(debug=True)
```

WEATHER

```
import requests
```

```
class Weather():
```

```
    def __init__(self, config):  
self.location = None  
self.config = config
```

```
    def set_location(self, location):  
self.location = location
```

```
    def get_location(self):  
        return self.location
```

```
    def download_weather_data(self):  
        params = {  
            'q': self.location,  
            'appid': self.config['API_KEY'],  
            'units': 'metric'  
        }  
        try:
```

```

        response = requests.get(self.config['API_URL'], params=params)
    response.raise_for_status()
    return response.json()
except requests.exceptions.HTTPError:
    raise WeatherException("Weather for {} not found".format(self.location))

def get_forecast_data(self):
    weather = self.download_weather_data()
    w = weather['list']
    current = w[0]['weather'][0]['description'], w[0]['main']['temp']
    tomorrow = w[1]['weather'][0]['description'], w[1]['main']['temp']
    dayafter = w[2]['weather'][0]['description'], w[2]['main']['temp']

    return weather['city']['name'], current, tomorrow, dayafter

```

```

class WeatherException(Exception):

```

```

    def __init__(self, *args):
        if args:
            self.message = args[0]
        else:
            self.message = None

```

```

    def __str__(self):
        if self.message:
            return self.message
        else:

```

```
return "Weather not found!"
```

WEATHER-APP

```
from flask import Flask, render_template, request, redirect, url_for
from weather import Weather, WeatherException
```

```
app = Flask(__name__)
app.config.from_pyfile('config/config.cfg')
w = Weather(app.config)
```

```
@app.route('/')
def homepage():
    return render_template('homepage.html')
```

```
@app.route('/result', methods=['POST', 'GET'])
def result_page():
    if request.method == 'POST':
        location = request.form
    w.set_location(location.get('location'))
```

```
    try:
        return render_template('result.html', data=w.get_forecast_data())
    except WeatherException:
app.log_exception(WeatherException)
        return render_template('error.html')
    else:
```

```
        return redirect(url_for('homepage'))

@app.route('/about', methods=['GET'])
def about_page():
    return render_template('about.html')

if __name__ == '__main__':
    app.run(debug=True)
```

```
import json
import pandas as pd
import time
import numpy as np
import itertools
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
import scipy
```

```

from scipy.stats import spearmanr
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import scale
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
import pickle
from flask import Flask,render_template,url_for,request

app = Flask(__name__)

def preprocess(state, year):
    data = pd.read_csv('Data/'+state+'.csv')
    temp = data[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
        'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[data['YEAR'] == year]

    data_year = np.asarray(temp[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
        'AUG', 'SEP', 'OCT', 'NOV', 'DEC']])

    X_year = None; y_year = None
    for i in range(data_year.shape[1]-3):
        if X_year is None:
            X_year = data_year[:, i:i+3]
            y_year = data_year[:, i+3]
        else:
            X_year = np.concatenate((X_year,data_year[:, i:i+3]), axis=0)
            y_year = np.concatenate((y_year, data_year[:, i+3]), axis=0)

```

```

    return (X_year, y_year)

@app.route('/predict',methods=['POST'])
def predict():
    if request.method == 'POST':
        data = request.get_json()
        year = int(data['Year'])
        fname = data['State']
        p_in = open("trained/"+fname+".pkl","rb")
        model = pickle.load(p_in)
        x = pd.read_csv("test/"+fname+"_input.csv")
        a, b, c = float(x[x['YEAR'] == year]['Mar-May']), float(x[x['YEAR'] ==
year]['avgjune']),float(x[x['YEAR'] == year]['sub'])
        l = [[a, b, c]]
        my_prediction = model.predict(l)
        my_prediction = my_prediction[0]
        result = {}
        if my_prediction == '0':
            result['data'] = "Very Low Rainfall"
        elif my_prediction == '-1':
            result['data'] = "Light to Medium Rainfall"
        elif my_prediction == '1':
            result['data'] = "Heavy Rainfall"
        print(result)
        return result

@app.route('/rainfall', methods=['POST'])
def rainfall():
    if request.method == 'POST':

```

```
data = request.get_json()
year = int(data['Year'])
fname = data['State']
x_test, y_test = preprocess(fname, year)
p_in = open("graphs/"+fname+".pkl", "rb")
    model = pickle.load(p_in)
y_pred = model.predict(x_test)
    result = {}
    result['actual'] = list(y_test)
    result['prediction'] = list(y_pred)
print(result)
    return result

if __name__ == '__main__':
    app.run(debug=True)
```

REFERENCE

- [8] Charaniya, Nizar Ali, and Sanjay V. Dudul. "Committee of artificial neural networks for monthly rainfall prediction using wavelet transform." Business, Engineering and Industrial Applications (ICBEIA), 2011 International Conference on. IEEE, 2011.
- [7] French, Mark N., Witold F. Krajewski, and Robert R. Cuykendall. "Rainfall forecasting in space and time using a neural network." Journal of hydrology 137.1-4 (1992): 1-31.
- [4] Guhathakurta, P. "Long-range monsoon rainfall prediction of 2005 for the districts and sub-division Kerala with artificial neural network." Current Science 90.6 (2006): 773-779.
- [6] Lee, Sunyoung, Sungzoon Cho, and Patrick M. Wong. "Rainfall prediction using artificial neural networks." journal of geographic information and Decision Analysis 2.2 (1998): 233-242..
- [5] Pilgrim, D. H., T. G. Chapman, and D. G. Doran. "Problems of rainfall-runoff modelling in arid and semiarid regions." Hydrological Sciences Journal 33.4 (1988): 379-400.
- [3] Riordan, Denis, and Bjarne K. Hansen. "A fuzzy casebased system for weather prediction." Engineering Intelligent Systems for Electrical Engineering and Communications 10.3 (2002): 139-146.
- [2] Schmitz, G. H., and J. Cullmann. "PAI-OFF: A new proposal for online flood forecasting in flash flood prone catchments." Journal of hydrology 360.1-4 (2008): 1-14.
- [1] Xiong, Lihua, and Kieran M. OConnor. "An empirical method to improve the prediction limits of the GLUE methodology in rainfallrunoffmodeling." Journal of Hydrology 349.1-2 (2008): 115-124.

