EXPLORATORY ANALYSIS OF RAINFALL DATA IN INDIA FOR AGRICULTURE

1.INTRODUCTION:

Rainfall remains one of the most influential meteorological parameters in many aspects of our daily lives. With effects ranging from damage to infrastructure in the event of a flood to disruptions in the transport network, the socio-economic impacts of rainfall are not. Floods and similar extreme events are consequences of climate change that are expected to occur more frequently and have catastrophic effects in years to come. More interestingly, recent studies have highlighted that weather conditions can potentially increase air pollution (another major topic of discourse alongside climate change in recent times) in winter and summer periods. It is pertinent to reiterate that increased air pollution results in health conditions such as asthma and similar problems related to the lungs. Therefore, as a mitigation approach, many studies have investigated and proposed rainfall forecasting techniques in preparation for any eventuality. However, in order to enhance human mobility activities and enhance agriculture and industrial development, these approaches must provide efficient and timely predictions.

1.1.Project Overview:

Predicting the amount of daily rainfall improves agricultural productivity and secures food and water supply to keep citizens healthy. To predict rainfall, several types of research have been conducted using data mining and machine learning techniques of different countries' environmental datasets. An erratic rainfall distribution in the country affects the agriculture on which the economy of the country depends on. Wise use of rainfall water should be planned and practiced in the country to minimize the problem of the drought and flood occurred in the country. The main objective of this study is to identify the relevant atmospheric features that cause rainfall and predict the intensity of daily rainfall using machine learning techniques.

The Pearson correlation technique was used to select relevant environmental variables which were used as an input for the machine learning model. The dataset was collected from the local meteorological office at Bahir Dar City, Ethiopia to measure the performance of three machine learning techniques (Multivariate Linear Regression, Random Forest, and Extreme Gradient Boost).

Root mean squared error and Mean absolute Error methods were used to measure the performance of the machine learning model. The result of the study revealed that the Extreme Gradient Boosting machine learning algorithm performed better than others.

1.2.Purpose:

To choose the better machine learning algorithms to study the daily rainfall amount prediction, various papers have been reviewed concerning rainfall prediction.

To predict the daily rainfall intensity using the real-time environmental data, three algorithms such as MLP, RF, and XGBoost gradient descent were chosen for the experiment. Hence, the three machine learning algorithms were experimented with and compared to report the better algorithms to predict the daily rainfall amount.

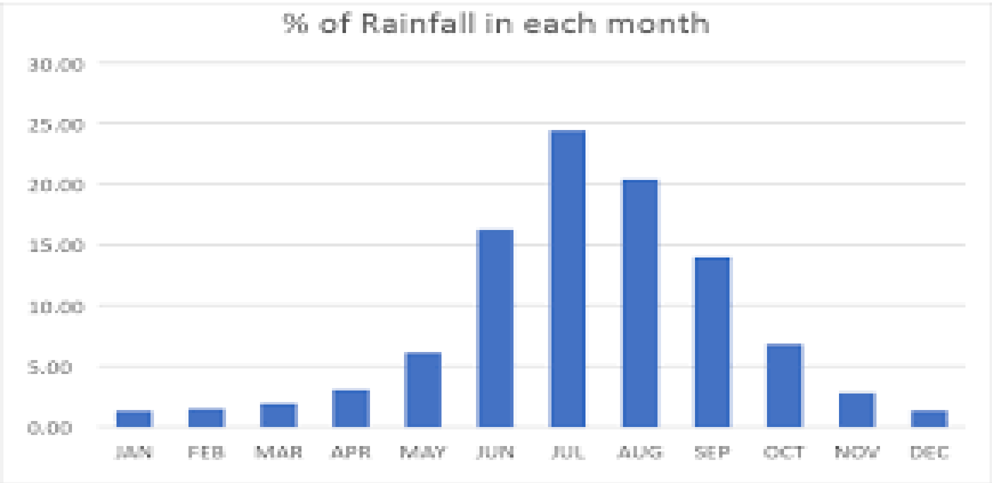
2. LITERATURE SURVEY:

Exploratory Data Analysis of Indian Rainfall Data for Agriculture

India is an agricultural country and secondary agro based market will be steady with a good monsoon. The economic growth of each year depends on the amount of duration of monsoon rain, bad monsoon can lead to destruction of some crops, which may result in scarcity of some agricultural products which in turn can cause food inflation, insecurity and public unrest. In our analysis we are trying to understand the behavior of rainfall in India over the years, by months and different subdivisions.

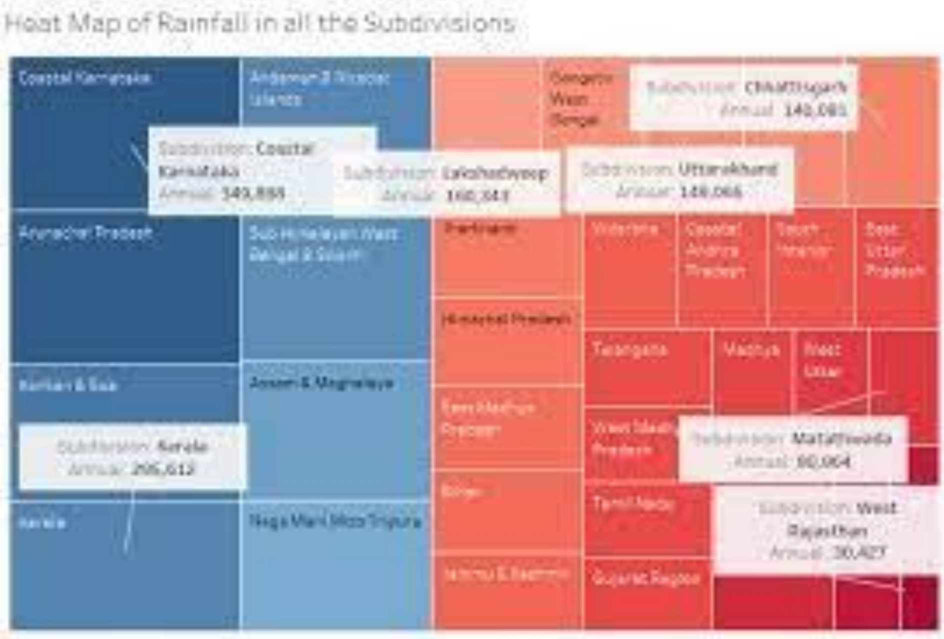
Annual rainfall by months:

The below graph shows the percentage of rainfall each month receives when we consider India as a whole. The rainfall in the months of June, July, August and September together contribute to almost 80% of the annual rainfall.

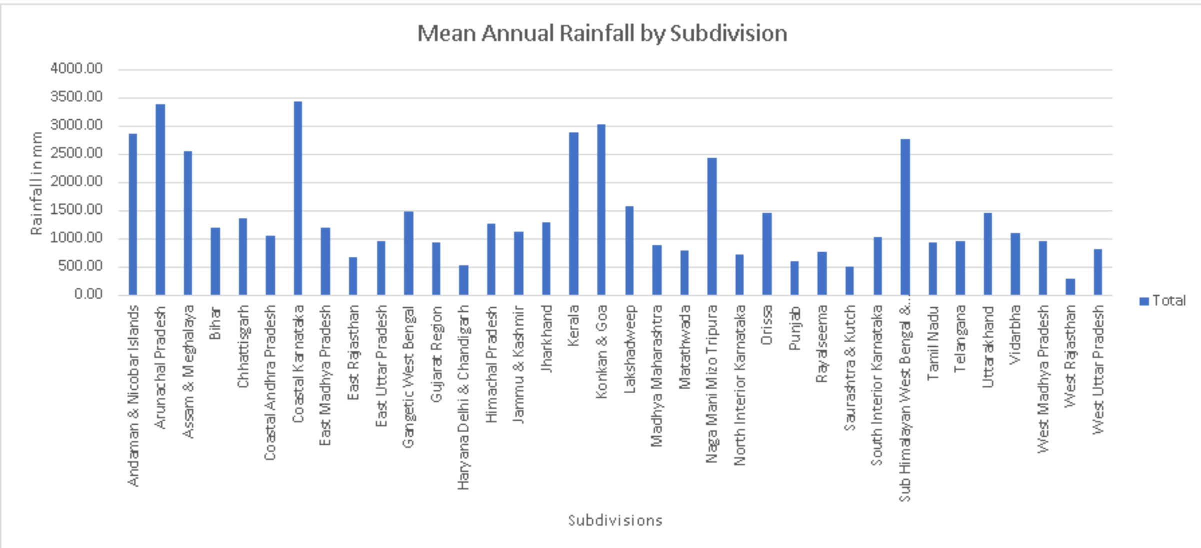


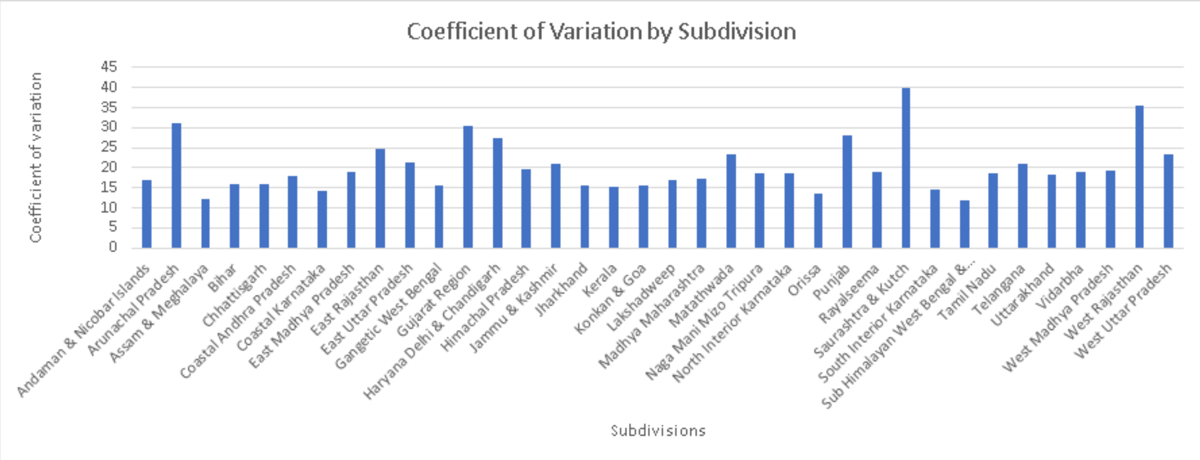
Annual rainfall by subdivision:

The following is a heat map plotted based on sum of rainfall received by each subdivision for all these years. The subdivisions with large area represents high rainfall and with small boxes represent less rainfall. We can see that the subdivision located at Southwest and Northeast part of India have received more rainfall compared to central India.



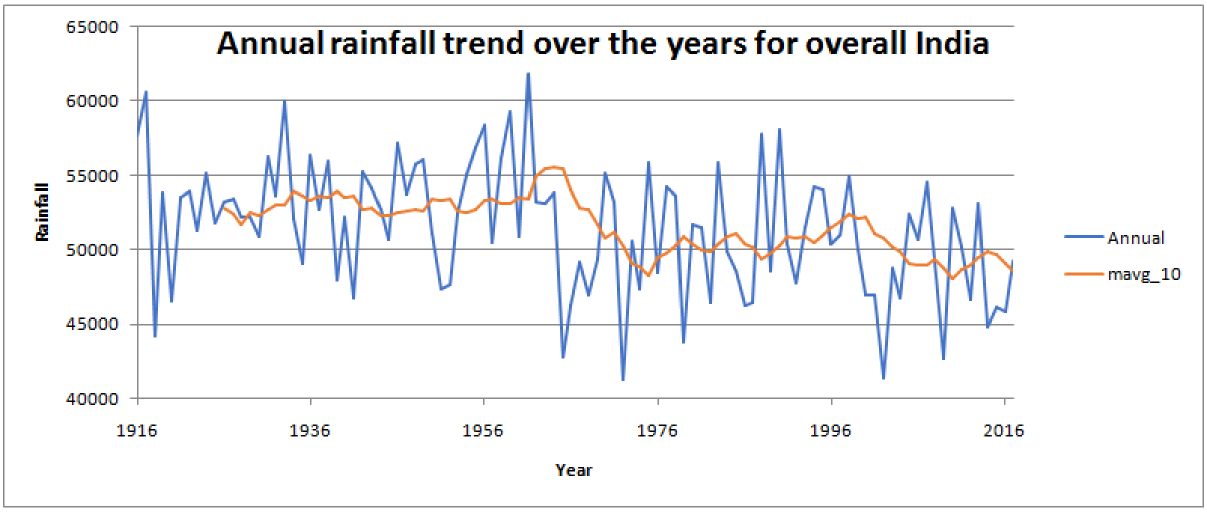
The average rainfall and variation values are plotted for each subdivision on different graphs which are given below. We can see that the subdivision which receive High rainfall have less variation seen over years whereas the subdivisions receiving low rainfall showed more variation over the years.





Annual Rainfall trend over the years for whole India

10 years moving average was plotted, we can see that there is a decreasing trend in rainfall in the recent years.



2.1 Existing problem:

Over the previous decade, academic and commercialized databases have been extending at exceptional rates. Capture advanced perception from such databases is hard, expansive and time-consuming if done manually. It is hopeless when data exceeds definite limits of size and complexity. For this reason, during the previous years the automated analysis and visualization of huge multi-dimensional datasets has been the center of attention on scientific research. The fundamental aim is to observe rules and relationships in the data, thereby gaining attain to invisible and potentially valuable knowledge. Artificial Neural Networks are a hopeful part of this broad field. Motivated by advances in biomedical research, they shape a class of algorithms that goal to reproduce the neural structures of the brain. The reason is that ANN (Artificial Neural Network) model is based on 'prediction' by smartly 'analyzing' the trend from an already existing voluminous historical set of data.

2.2 References:

1. Aftab, S.; Ahmad, M.; Hameed, N.; Salman, M.; Ali, I.; Nawaz, Z. Rainfall Prediction in Lahore City using Data Mining Techniques. Int. J. Adv. Comput. Sci. Appl. 2018, 9, 254–260.

2. Aftab, S.; Ahmad, M.; Hameed, N.; Salman, M.; Ali, I.; Nawaz, Z. Rainfall Prediction using Data Mining Techniques: A Systematic Literature Review. Int. J. Adv. Comput. Sci. Appl. 2018, 9, 143–150.

3. Nayak, M.A.; Ghosh, S. Prediction of extreme rainfall event using weather pattern recognition and support vector machine classifier. Arch. Meteorol. Geophys. Bioclimatol. Ser. B 2013, 114, 583–603.

4. Yue, T.; Zhang, S.; Zhang, J.; Zhang, B.; Li, R. Variation of representative rainfall time series length for rainwater harvesting modelling in different climatic zones. J. Environ. Manag. 2020, 269, 110731.

5. Mishra, N.; Soni, H.K.; Sharma, S.; Upadhyay, A. A Comprehensive Survey of Data Mining Techniques on Time Series Data for Rainfall Prediction. J. ICT Res. Appl. 2017, 11, 168.

6. Gupta, D.; Ghose, U. A comparative study of classification algorithms for forecasting rainfall. In Proceedings of the 2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) Trends and Future Directions, Noida, India, 2–4 September 2015; pp. 1–6.

7. Wu, C.L.; Chau, K.W. Prediction of Rainfall Time Series Using Modular Soft Computing Methods. Eng. Appl. Artif. Intell. 2013, 26, 997–1007.

8. Chau, K.W.; Wu, C.L. A hybrid model coupled with singular spectrum analysis for daily rainfall prediction. J. Hydroinformatics 2010, 12, 458–473.

9. Wu, J.; Long, J.; Liu, M. Evolving RBF neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm. Neurocomputing 2015, 148, 136–142.
10. Sawale, G.J.; Gupta, S.R. Use of Artificial Neural Network in Data Mining For Weather Forecasting. Int. J. Comput. Sci. Appl. 2013, 6, 383–387.

2.3 Problem Statement Definition

Rainfall forecasting has been around for years using traditional methods that employ statistical techniques to assess the correlation between rainfall, geographic coordinates (such as latitude and longitude), and other atmospheric factors (like pressure, temperature, wind speed, and humidity). However, the complexity of rainfall such as its non-linearity makes it difficult to predict. Consequently, attempts have been made to reduce this non-linearity by using Singular Spectrum Analysis, Empirical Mode Decomposition, Wavelet analysis, among others. Nevertheless, the mathematical and statistical models employed require complex computing power and can be time-consuming with minimal effects.

India is an agricultural country and secondary agro based market will be steady with a good monsoon. The economic growth of each year depends on the amount of duration of monsoon rain, bad monsoon can lead to destruction of some crops, which may result in scarcity of some agricultural products which in turn can cause food inflation, insecurity and public unrest. In our analysis we are trying to understand the behavior of rainfall in India over the years, by months and different subdivisions.

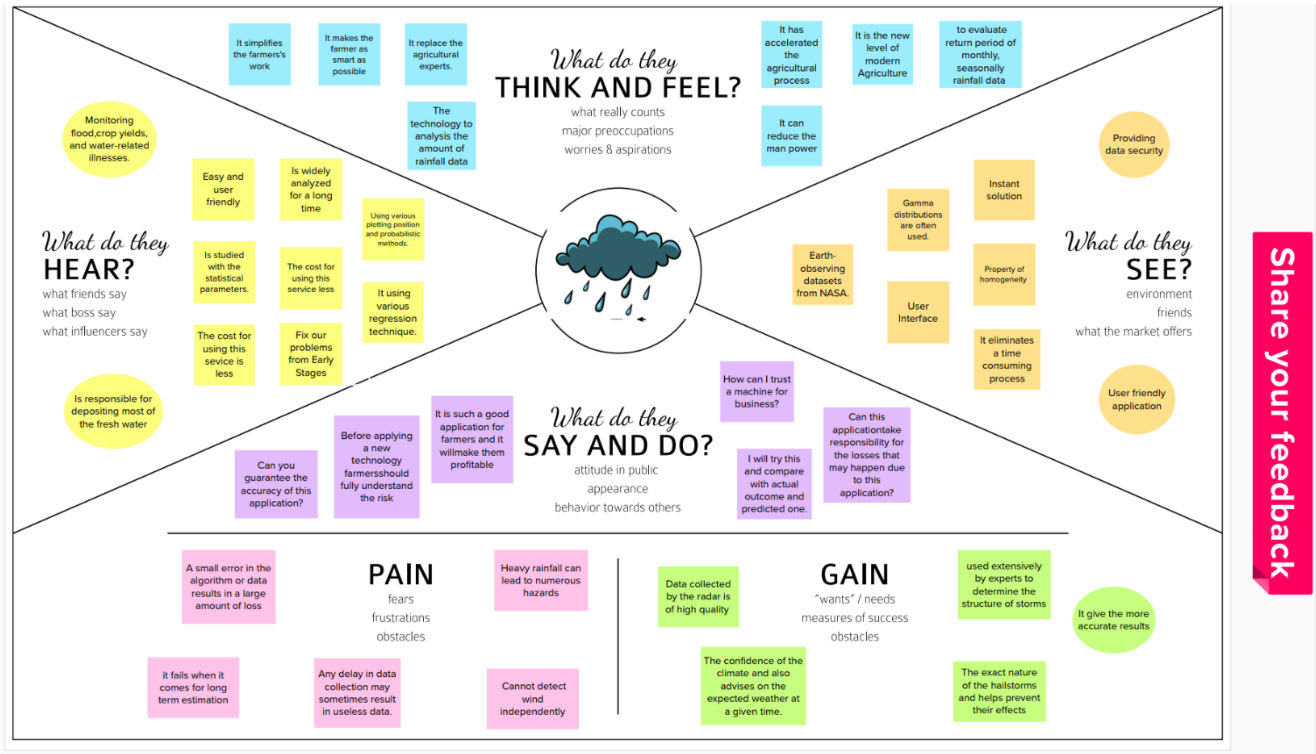
This comparative study is conducted concentrating on the following aspects: modeling inputs, Visualizing the data, modeling methods, and pre-processing techniques. The results provide a comparison of various evaluation metrics of these machine learning techniques and their reliability to predict rainfall by analyzing the weather data.We will be using classification algorithms such as Decision tree, Random forest, KNN, and xgboost. We will train and test the data with these algorithms. From this best model is selected.

3. IDEATION & PROPOSED SOLUTION:

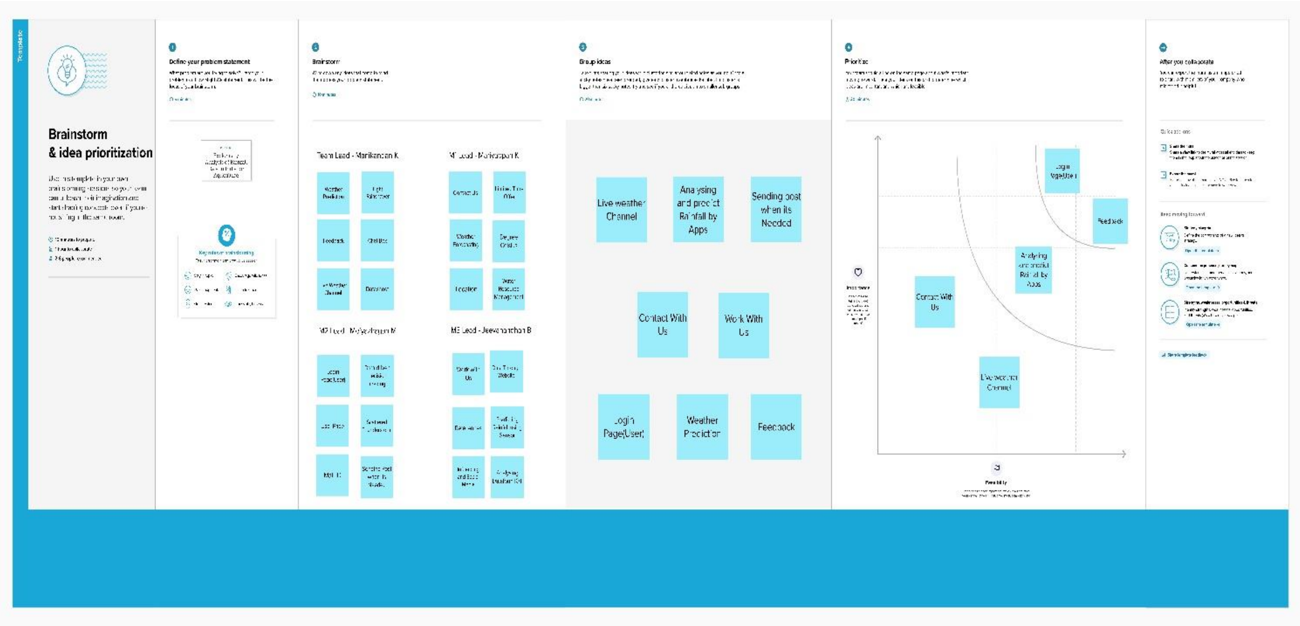
3.1 Empathy Map Canvas:

Exploratory Analysis Of RainFall Data In India For Agriculture:

Rainfall has been a major concern these days. Weather conditions have been changing for time being. Rainfall forecasting is important otherwise, it may lead to many disasters. Irregular heavy rainfall may lead to the destruction of crops, heavy foods that can cause harm to human life. It is important to exactly determine the rainfall for effective use of water resources, crop productivity, and pre-planning of water structures.



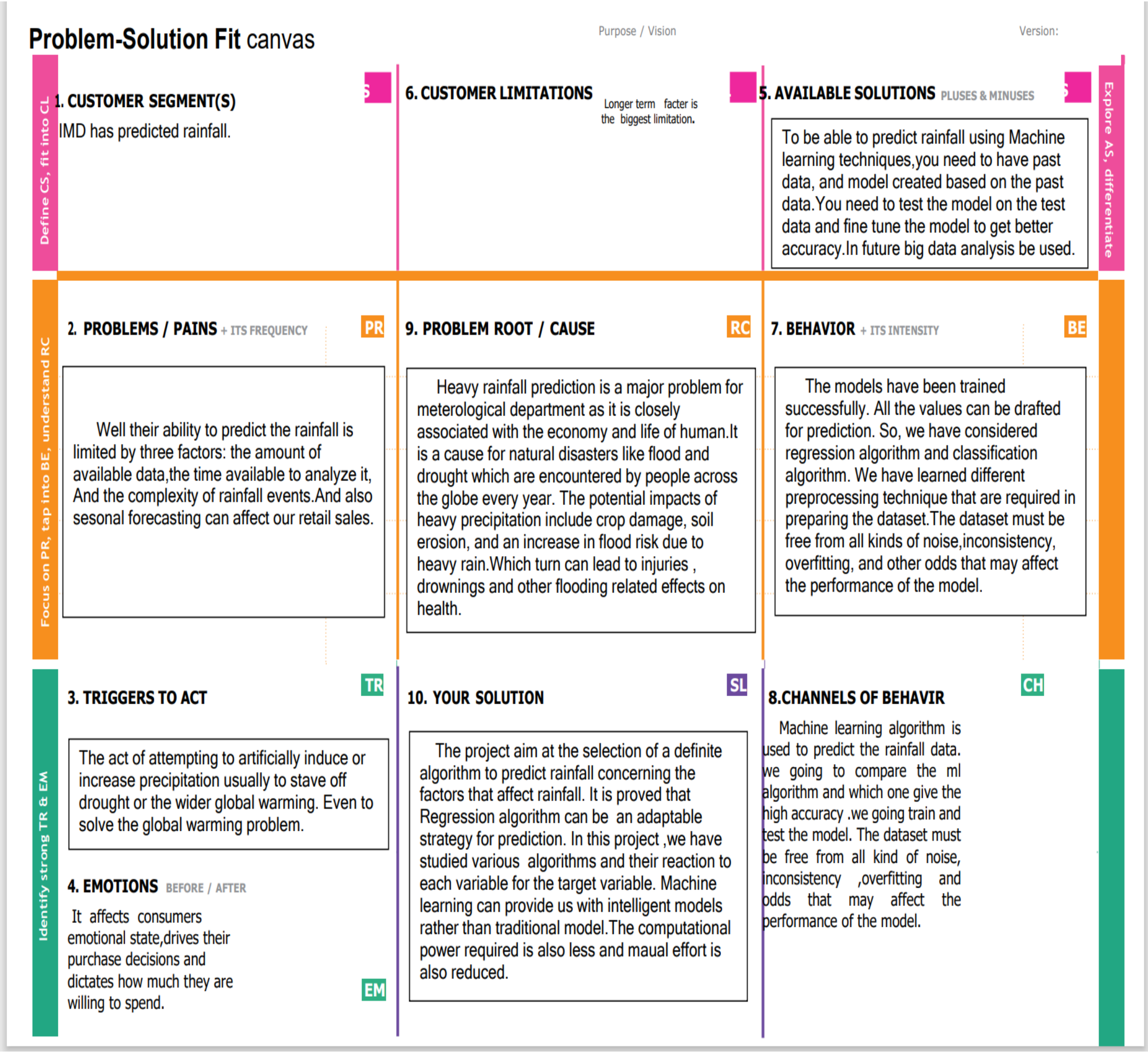
3.2 Ideation & Brainstorming:



3.3 Proposed Solution:

A detailed survey on rainfall predictions using Artificial Neural Network architecture over twenty-five years is done. From the survey it has been found that most of the researchers used different models for rainfall prediction, but keras model of ANN gives significant results. ANN is the model with least mean squared error and accurate prediction. The survey also gives a conclusion that the forecasting techniques like Decision Tree, Random Forest of XGBoost,KNN are suitable to predict rainfall than other forecasting techniques such as statistical and numerical methods. However, some limitation of those methods has been found. The extensive references in support of the different developments of ANN research provided should be of great help to ANN researchers to accurately predict rainfall in the future.

3.4 Problem Solution Fit:



4. REQUIREMENT ANALYSIS:

4.1 Functional Requirement:

Following are the functional requirements of the proposed solution

FR No:	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR - 1	User Authentication	The users must be registered first and can be only able to access the web application . This is to ensure that the web application is used for a good reason.
FR - 2	Web Service Management Process	Web Service Management process by Web Portal admin in registering web client to do SSO or member data communication. The web page is hosted in cloud.
FR - 3	Data Management	The Web server and Portal manager can have access to data to edit and update again to server.
FR - 4	Testing	Applying the algorithms on the test data.
FR - 5	Confirmation	Display the result with the description of having Rainfall or no.

4.2 Non-functional Requirements:

Following are the non-functional requirements of the proposed solution

FR No:	Non-Functional Requirement	Description
NFR – 1	**Usability**	The webpage loading for users submitting their image input details at the web application must be loaded fast than rendering more time.
NFR – 2	**Security**	Authorization access scenarios and definitions, hand-over procedures for patient

		records. The image and other inputs of patients must be highly secured and can't be accessible to others.
NFR – 3	**Reliability**	The prediction of the system must be with higher accuracy so that the output from the application can be trusted by the users without any doubts and can be used for further dragonising process with Researchers.
NFR – 4	**Performance**	The landing page supporting 5,000 users per hour must provide 6 second or less response time in a Chrome desktop browser, including the rendering of text and images and over an LTE connection and the uploading of Data (image) must also should be fast and the output page should be rendered within seconds
NFR – 5	**Availability**	The web application should be available to all Research across the globe and can be implemented in every hospital so that the people can use it effectively .

NFR – 6	**Scalability**	The System must function using Cloud and during a down process also it must satisfy the maximum number of clients. . The system must use higher RAM and CPU processing in Server to handle multiple request at same time

5. PROJECT DESIGN:

5.1 Data Flow Diagrams:

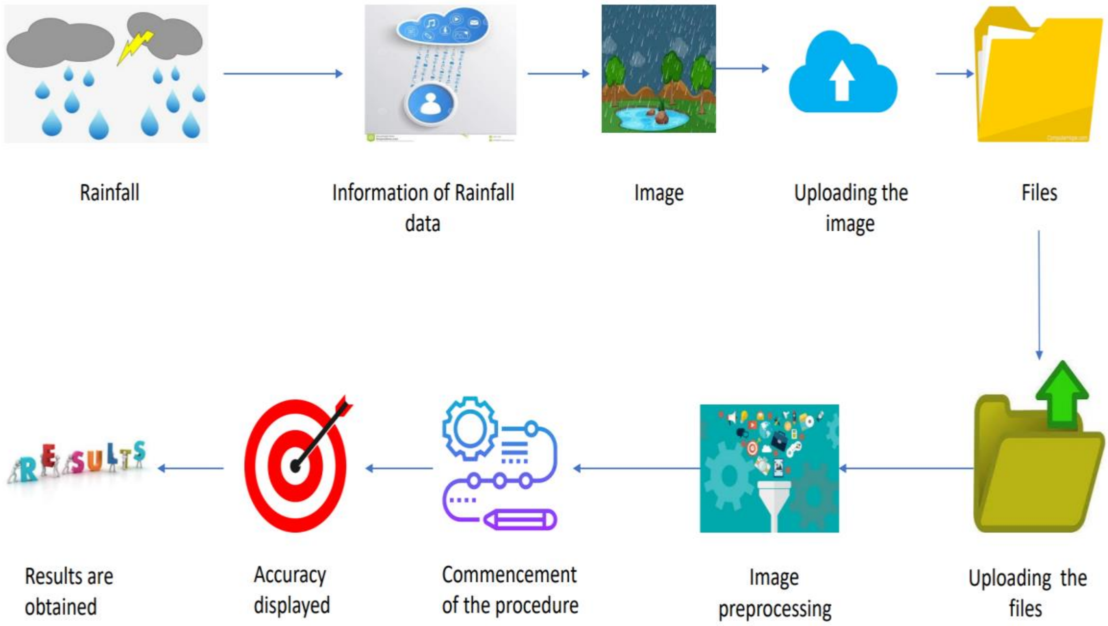
DFD is the abbreviation for Data Flow Diagram

The flow of data of a system or a process is represented by DFD.

It also gives insight into the inputs and outputs of each entity and the process itself

DFD does not have control flow and no loop or decision rules are present

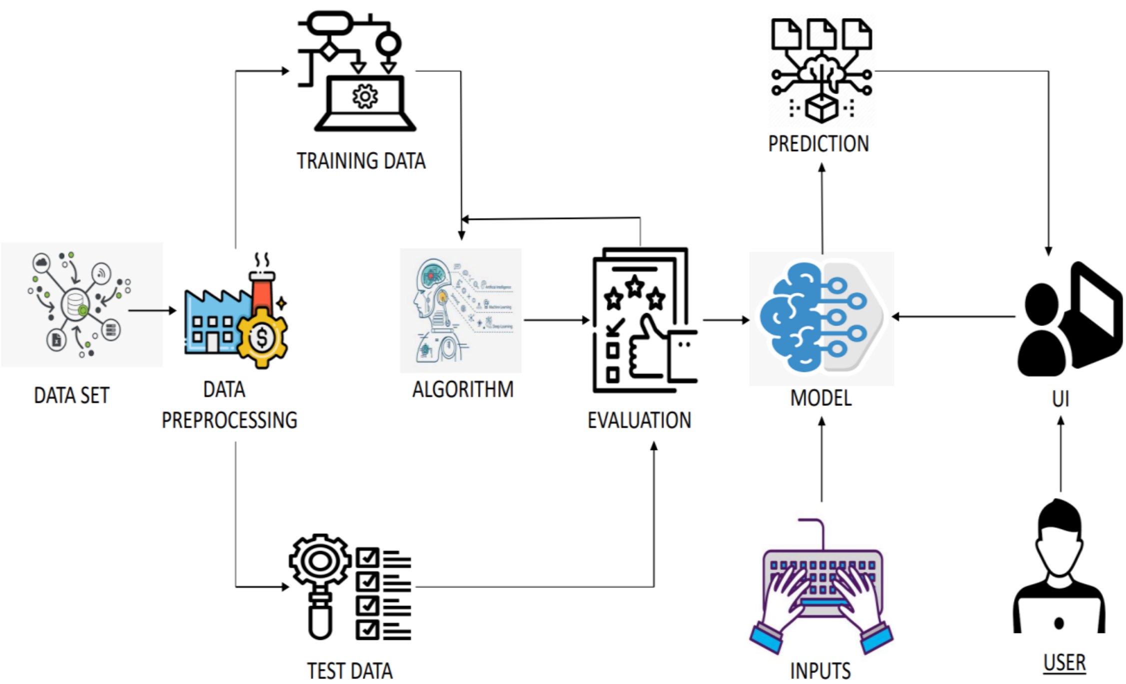
Specific operation depending on the type of data can be explained by a flowchart



User Type	Functional requirement	User story number	User story/task	Acceptance criteria	Priority	Release
User	Account creation	USN-1	User can connect to the application	User can access the account being created	High	Sprint-1
Input data	Adding data	USN-2	Input can be given to the system for its learning purposes	Data entered could be verified by the user	High	Sprint-1
Data validation	Checking accuracy	USN-3	Ability and accuracy of the model can be checked by the user	On logging in to account the capability could be checked	Medium	Sprint-2
Classification	Data classification	USN-4	Data can be viewed by the user	Verify the user data with real data	Medium	Sprint-2
App work	Work flow	USN-5	Working action of the application model could be viewed	Application working and responses to the	Medium	Sprint-2

Image classification	Checking for the rain	USN-6	With the help of trained and test data user can verify with application that the image is identified with the actual image	User can confirm that the data shows accrate results	Low	Sprint-3
User interaction	AI-powered chatbot	USN-7	User can interact with the automated chatbot to engage my time till the application processed the accurate result	Result could be viewed from the interaction from the chatbot	Low	Sprint-3
Agriculture assistance	Agriculture suggestion	USN-8	User can get agriculture advises	Enough assistance could be obtained	High	Sprint-3
Data extraction	Obtaining the data	USN-9	User can retrieve the result data from the application for data storage	Result could be downloaded in the form data to be shown to the agriculture teams	Medium	Sprint-4

5.2 Solution & Technical Architecture:



5.3 User Stories:

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by Filling the form	I can receive confirmation via OTP	High	Sprint - 1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint - 1

6. PROJECT PLANNING & SCHEDULING:

6.1 Sprint Planning & Estimation:

Sprint	Functional Requirement [Epic]	User story Number	User story/Task/Activity	Story points	Priority	Team Members
Sprint-1	Registration	USN-1	User can register for the application by entering his or her email,password,and confirming the password.	5	High	Rithikaa, Aishwarya, Gokul Prasath, Sounder

Sprint-1		USN-2	User will receive conformation email or message once registered for the application.	3	High	Rithikaa, Aishwarya, Gokul Prasath, Sounder
Sprint-1	Login	USN-3	Enter the username and password to login to the application	2	High	Rithikaa, Aishwarya, Gokul Prasath, Sounder
Sprint-2	Dashboard	USN-4	User can visualization of the rainfall data for a specific time period	3	Medium	Rithikaa, Aishwarya
Sprint-2		USN-5	User can change his/her password and can view the account details and search history	5	High	Gokul Prasath, Sounder
Sprint-3	Support	USN-6	User can give the feedback on the accuracy of the prediction and on the user interface	5	High	Rithikaa, Aishwarya
Sprint-3		USN-7	Responds to user queries via email	2	Medium	Gokul Prasath, Sounder
Sprint-3		USN-8	The team must respond immediately to the queries based on the priority	5	High	Rithikaa, Aishwarya
Sprint-4	Core Function	USN-9	User can enter the temperature condition of the environment	8	High	Rithikaa, Aishwarya, Gokul Prasath, Sounder
Sprint-4		USN-10	Prediction of rainfall and displaying of result	2	Medium	Rithikaa, Aishwarya, Gokul Prasath, Sounder
Sprint-4		USN-11	The website is response on all the device and the screen sizes	5	High	Rithikaa, Aishwarya, Gokul Prasath, Sounder

6.2 Sprint Delivery Schedule:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed(as on Planned End Date)	Sprit Release Date(Actual)
Sprint-1	10	6 Days	30 Oct 2022	04 Nov2022	-----	05 Nov 2022
Sprint-2	07	5 Days	03 Nov 2022	07Nov 2022	-----	08 Nov 2022
Sprint-3	12	6 Days	08 Nov 2022	13Nov 2022	-----	14 Nov 2022
Sprint-4	15	5 Days	14 Nov 2022	18Nov 2022	-----	19 Nov 2022

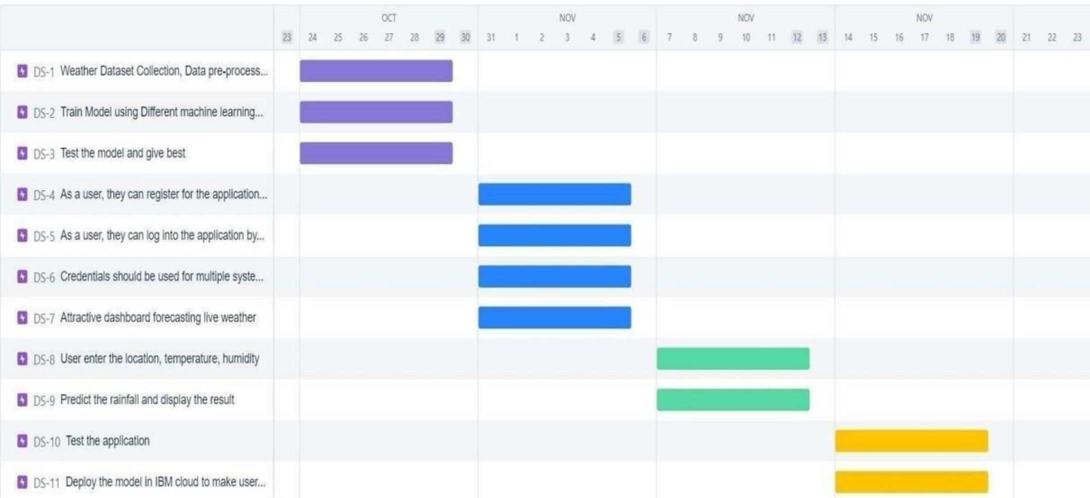
Velocity:

Average Sprint Velocity [Estimated To Be Idea] = Story points to be completed out of all User stories / Total number of sprint

=44\4= 11

Therefore, The amount of work to be done on each Sprint in an average of 11 story points.

6.3 Reports from JIRA :



7.CODING & SOLUTIONING

(Explain the features added in the project along with code)

7.1 Feature 1

```
import pandas as pd
import numpy as np
import seaborn as sns
import  matplotlib.pyplot as plt
from sklearn import preprocessing
from sklearn import model_selection
from sklearn import metrics
from sklearn import linear_model
from sklearn import ensemble
from sklearn import tree
from sklearn import svm
import xgboost
import sklearn

data = pd.read_csv("/content/weatherAUS.csv - weatherAUS.csv.csv")
data.head()
```

```
         Date Location  MinTemp  MaxTemp  Rainfall  Evaporation
Sunshine  \
0  2008-12-01   Albury     13.4     22.9       0.6          NaN
NaN
1  2008-12-02   Albury      7.4     25.1       0.0          NaN
NaN
2  2008-12-03   Albury     12.9     25.7       0.0          NaN
NaN
3  2008-12-04   Albury      9.2     28.0       0.0          NaN
NaN
4  2008-12-05   Albury     17.5     32.3       1.0          NaN
NaN

  WindGustDir  WindGustSpeed WindDir9am  ... Humidity9am  Humidity3pm
\
0           W           44.0          W  ...        71.0         22.0

1         WNW           44.0        NNW  ...        44.0         25.0

2         WSW           46.0          W  ...        38.0         30.0

3          NE           24.0         SE  ...        45.0         16.0

4           W           41.0        ENE  ...        82.0         33.0


   Pressure9am  Pressure3pm  Cloud9am  Cloud3pm  Temp9am  Temp3pm
RainToday  \
0       1007.7       1007.1       8.0       NaN     16.9     21.8
No
1       1010.6       1007.8       NaN       NaN     17.2     24.3
No
2       1007.6       1008.7       NaN       2.0     21.0     23.2
```

```
No
3       1017.6       1012.8       NaN       NaN      18.1      26.5
No
4       1010.8       1006.0       7.0       8.0      17.8      29.7
No

   RainTomorrow
0            No
1            No
2            No
3            No
4            No

[5 rows x 23 columns]
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 145460 entries, 0 to 145459
Data columns (total 23 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   Date           145460 non-null  object
 1   Location       145460 non-null  object
 2   MinTemp        143975 non-null  float64
 3   MaxTemp        144199 non-null  float64
 4   Rainfall       142199 non-null  float64
 5   Evaporation    82670 non-null   float64
 6   Sunshine       75625 non-null   float64
 7   WindGustDir    135134 non-null  object
 8   WindGustSpeed  135197 non-null  float64
 9   WindDir9am     134894 non-null  object
 10  WindDir3pm     141232 non-null  object
 11  WindSpeed9am   143693 non-null  float64
 12  WindSpeed3pm   142398 non-null  float64
 13  Humidity9am    142806 non-null  float64
 14  Humidity3pm    140953 non-null  float64
 15  Pressure9am    130395 non-null  float64
 16  Pressure3pm    130432 non-null  float64
 17  Cloud9am       89572 non-null   float64
 18  Cloud3pm       86102 non-null   float64
 19  Temp9am        143693 non-null  float64
 20  Temp3pm        141851 non-null  float64
 21  RainToday      142199 non-null  object
 22  RainTomorrow   142193 non-null  object
dtypes: float64(16), object(7)
memory usage: 25.5+ MB
```

```
data.shape
```

```
(145460, 23)
```

```
print('\nUnique Values: ',data.nunique())
```

```
Unique Values:  Date             3436
Location         49
MinTemp         389
MaxTemp         505
Rainfall        681
Evaporation     358
Sunshine        145
WindGustDir      16
WindGustSpeed    67
WindDir9am       16
WindDir3pm       16
WindSpeed9am     43
WindSpeed3pm     44
Humidity9am     101
Humidity3pm     101
Pressure9am     546
Pressure3pm     549
Cloud9am         10
Cloud3pm         10
Temp9am         441
Temp3pm         502
RainToday         2
RainTomorrow      2
dtype: int64
```

```
print('\nMissing Values: ',data.isna().sum())
```

```
Missing Values:  Date                0
Location            0
MinTemp          1485
MaxTemp          1261
Rainfall         3261
Evaporation     62790
Sunshine        69835
WindGustDir     10326
WindGustSpeed   10263
WindDir9am      10566
WindDir3pm       4228
WindSpeed9am     1767
WindSpeed3pm     3062
Humidity9am      2654
Humidity3pm      4507
Pressure9am     15065
Pressure3pm     15028
Cloud9am        55888
Cloud3pm        59358
Temp9am          1767
```

```
Temp3pm          3609
RainToday        3261
RainTomorrow     3267
dtype: int64
```

```
data.describe()
```

```
             MinTemp        MaxTemp       Rainfall   Evaporation  \
count  143975.000000  144199.000000  142199.000000  82670.000000
mean       12.194034      23.221348       2.360918      5.468232
std         6.398495       7.119049       8.478060      4.193704
min        -8.500000      -4.800000       0.000000      0.000000
25%         7.600000      17.900000       0.000000      2.600000
50%        12.000000      22.600000       0.000000      4.800000
75%        16.900000      28.200000       0.800000      7.400000
max        33.900000      48.100000     371.000000    145.000000

           Sunshine  WindGustSpeed   WindSpeed9am   WindSpeed3pm  \
count  75625.000000  135197.000000  143693.000000  142398.000000
mean       7.611178      40.035230      14.043426      18.662657
std        3.785483      13.607062       8.915375       8.809800
min        0.000000       6.000000       0.000000       0.000000
25%        4.800000      31.000000       7.000000      13.000000
50%        8.400000      39.000000      13.000000      19.000000
75%       10.600000      48.000000      19.000000      24.000000
max       14.500000     135.000000     130.000000      87.000000

         Humidity9am    Humidity3pm   Pressure9am    Pressure3pm  \
count  142806.000000  140953.000000  130395.00000  130432.000000
mean       68.880831      51.539116    1017.64994    1015.255889
std        19.029164      20.795902       7.10653       7.037414
min         0.000000       0.000000     980.50000     977.100000
25%        57.000000      37.000000    1012.90000    1010.400000
50%        70.000000      52.000000    1017.60000    1015.200000
75%        83.000000      66.000000    1022.40000    1020.000000
max       100.000000     100.000000    1041.00000    1039.600000

           Cloud9am      Cloud3pm        Temp9am       Temp3pm
count  89572.000000  86102.000000  143693.000000  141851.00000
mean       4.447461      4.509930      16.990631      21.68339
std        2.887159      2.720357       6.488753       6.93665
min        0.000000      0.000000      -7.200000      -5.40000
25%        1.000000      2.000000      12.300000      16.60000
50%        5.000000      5.000000      16.700000      21.10000
75%        7.000000      7.000000      21.600000      26.40000
max        9.000000      9.000000      40.200000      46.70000
```

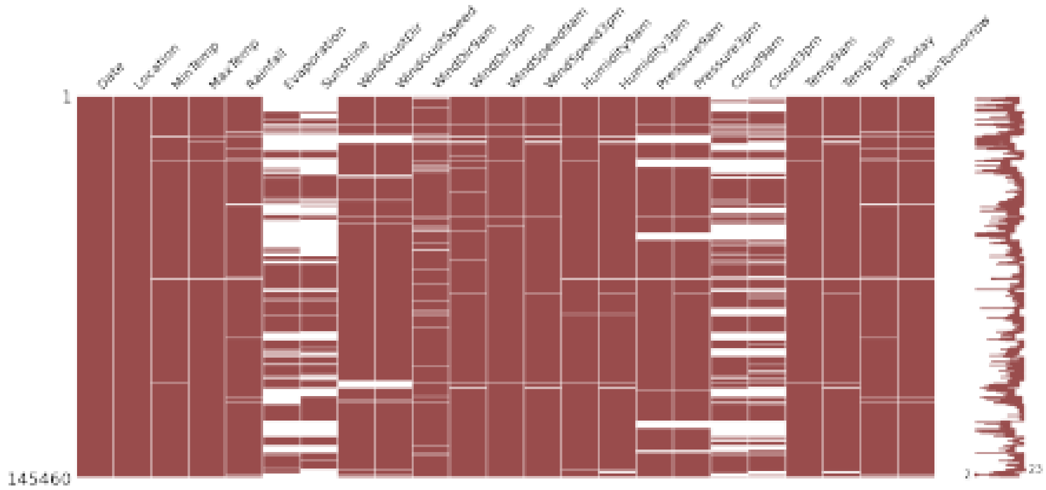
```
data.isnull().sum()
```

```
Date                0
Location            0
```

```
MinTemp           1485
MaxTemp           1261
Rainfall          3261
Evaporation      62790
Sunshine         69835
WindGustDir      10326
WindGustSpeed    10263
WindDir9am       10566
WindDir3pm        4228
WindSpeed9am      1767
WindSpeed3pm      3062
Humidity9am       2654
Humidity3pm       4507
Pressure9am      15065
Pressure3pm      15028
Cloud9am         55888
Cloud3pm         59358
Temp9am           1767
Temp3pm           3609
RainToday         3261
RainTomorrow      3267
dtype: int64
```

```
import missingno as msno
msno.matrix(data,color=(0.60,0.300,0.300),fontsize=20)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff0c1783bd0>
```



```
data_cat = data[['RainToday', 'WindGustDir', 'WindDir9am',
'WindDir3pm']]
data.drop(columns=['Evaporation', 'Sunshine', 'Cloud9am',
'Cloud3pm'],axis=1,inplace=True)
data.drop(columns=['RainToday', 'WindGustDir', 'WindDir9am',
'WindDir3pm'],axis=1,inplace=True)
```

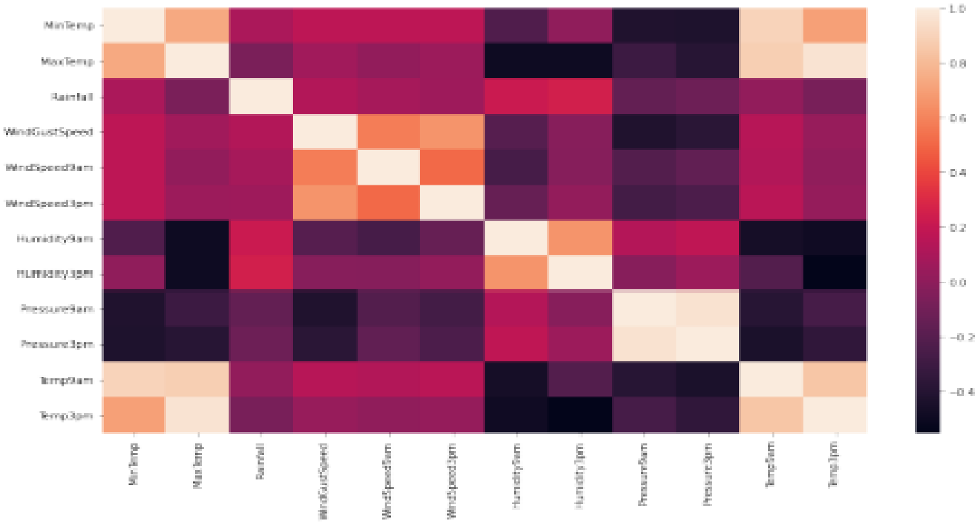
```
data['MinTemp'].fillna(data['MinTemp'].mean(), inplace=True)

data['MaxTemp'].fillna (data['MaxTemp'].mean(), inplace=True)

data['Rainfall'].fillna (data['Rainfall'].mean(), inplace=True)

data['WindGustSpeed'].fillna (data['WindGustSpeed'].mean(),
inplace=True)

data['WindSpeed9am'].fillna (data['WindSpeed9am'].mean(),
inplace=True)

data['WindSpeed3pm'].fillna (data['WindSpeed3pm'].mean(),
inplace=True)

data['Humidity9am'].fillna (data[ 'Humidity9am'].mean(), inplace=True)

data['Humidity3pm'].fillna (data['Humidity3pm'].mean(), inplace=True)

data['Pressure9am'].fillna (data[ 'Pressure9am'].mean(), inplace=True)

data['Pressure3pm'].fillna (data['Pressure3pm'].mean(), inplace=True)

data['Temp9am'].fillna (data['Temp9am'].mean(),inplace=True)

data['Temp3pm'].fillna(data['Temp3pm'].mean(),inplace=True)


cat_names=data_cat.columns

import numpy as np


from sklearn.impute import SimpleImputer
imp_mode= SimpleImputer (missing_values=np.nan, strategy =
'most_frequent')

data_cat= imp_mode.fit_transform(data_cat)

data_cat = pd.DataFrame(data_cat,columns=cat_names)

data = pd.concat([data, data_cat],axis=1)

data.corr()

                MinTemp   MaxTemp  Rainfall  WindGustSpeed
WindSpeed9am  \
MinTemp        1.000000  0.733400  0.102706       0.172553
0.173404
MaxTemp        0.733400  1.000000 -0.074040       0.065895
```

```
0.014294
Rainfall        0.102706 -0.074040  1.000000       0.126446
0.085925
WindGustSpeed   0.172553  0.065895  0.126446       1.000000
0.577319
WindSpeed9am    0.173404  0.014294  0.085925       0.577319
1.000000
WindSpeed3pm    0.173058  0.049717  0.056527       0.657243
0.512427
Humidity9am    -0.230970 -0.497927  0.221380      -0.207964     -
0.268271
Humidity3pm     0.005995 -0.498760  0.248905      -0.025355     -
0.030887
Pressure9am    -0.423584 -0.308309 -0.159055      -0.425760     -
0.215339
Pressure3pm    -0.433147 -0.396622 -0.119541      -0.383938     -
0.165388
Temp9am         0.897692  0.879170  0.011069       0.145904
0.127592
Temp3pm         0.699211  0.968713 -0.077684       0.031884
0.004476

               WindSpeed3pm  Humidity9am  Humidity3pm  Pressure9am  \
MinTemp            0.173058    -0.230970     0.005995    -0.423584
MaxTemp            0.049717    -0.497927    -0.498760    -0.308309
Rainfall           0.056527     0.221380     0.248905    -0.159055
WindGustSpeed      0.657243    -0.207964    -0.025355    -0.425760
WindSpeed9am       0.512427    -0.268271    -0.030887    -0.215339
WindSpeed3pm       1.000000    -0.143458     0.016275    -0.277604
Humidity9am       -0.143458     1.000000     0.659072     0.131503
Humidity3pm        0.016275     0.659072     1.000000    -0.025848
Pressure9am       -0.277604     0.131503    -0.025848     1.000000
Pressure3pm       -0.239659     0.176009     0.048695     0.959662
Temp9am            0.161060    -0.469641    -0.216964    -0.397131
Temp3pm            0.027587    -0.490709    -0.555608    -0.265532

               Pressure3pm   Temp9am   Temp3pm
MinTemp          -0.433147  0.897692  0.699211
MaxTemp          -0.396622  0.879170  0.968713
Rainfall         -0.119541  0.011069 -0.077684
WindGustSpeed    -0.383938  0.145904  0.031884
WindSpeed9am     -0.165388  0.127592  0.004476
WindSpeed3pm     -0.239659  0.161060  0.027587
Humidity9am       0.176009 -0.469641 -0.490709
Humidity3pm       0.048695 -0.216964 -0.555608
Pressure9am       0.959662 -0.397131 -0.265532
Pressure3pm       1.000000 -0.441459 -0.360707
Temp9am          -0.441459  1.000000  0.846141
Temp3pm          -0.360707  0.846141  1.000000
```

```
cor=data.corr()
```

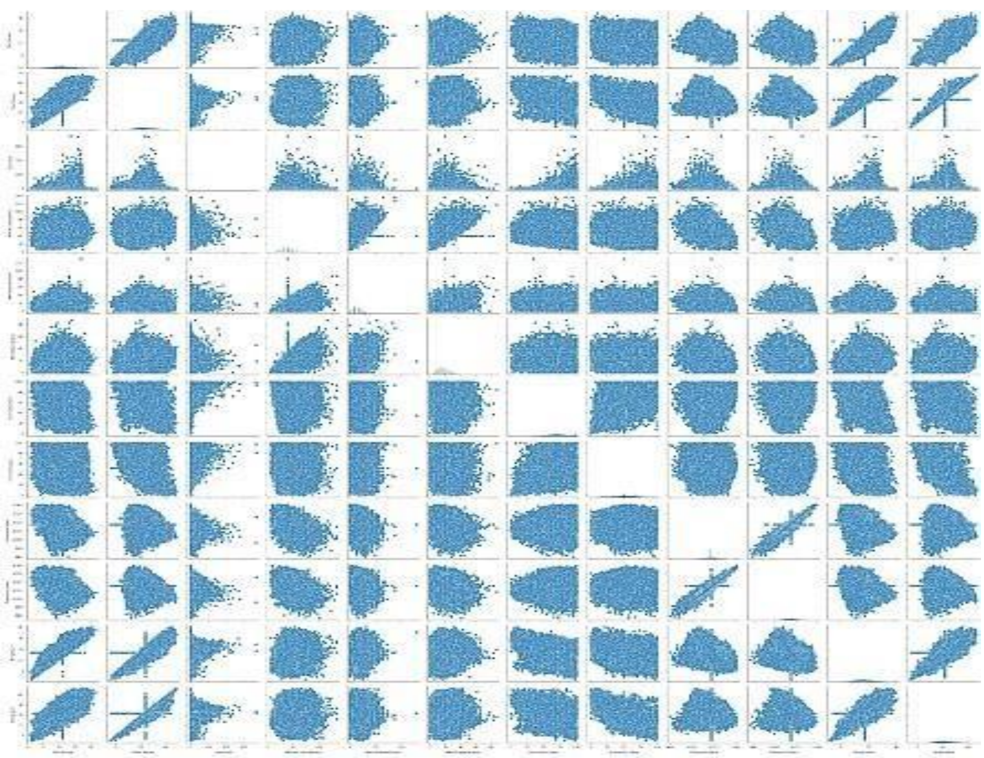
```
plt.figure(figsize=(15,8))
sns.heatmap(data=cor,xticklabels=cor.columns.values,yticklabels=cor.co
lumns.values)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb321e2bc10>
```




```
sns.pairplot(data)
```

```
<seaborn.axisgrid.PairGrid at 0x7fb31479d610>
```



```
plt.figure(figsize=(15,8))
data.boxplot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb30f56ec50>
```

```
0          W        WNW
1        NNW        WSW
2          W        WSW
3         SE          E
4        ENE         NW
```

```
df.shape
```

```
(142193, 19)
```

```
x=df.drop('RainTomorrow',axis=1)
y=df['RainTomorrow']
```

```
x.head()
```

```
         Date Location  MinTemp  MaxTemp  Rainfall  WindGustSpeed  \
0  2008-12-01   Albury     13.4     22.9       0.6           44.0
1  2008-12-02   Albury      7.4     25.1       0.0           44.0
2  2008-12-03   Albury     12.9     25.7       0.0           46.0
3  2008-12-04   Albury      9.2     28.0       0.0           24.0
4  2008-12-05   Albury     17.5     32.3       1.0           41.0

   WindSpeed9am  WindSpeed3pm  Humidity9am  Humidity3pm
Pressure9am  \
0          20.0          24.0         71.0         22.0       1007.7

1           4.0          22.0         44.0         25.0       1010.6

2          19.0          26.0         38.0         30.0       1007.6

3          11.0           9.0         45.0         16.0       1017.6

4           7.0          20.0         82.0         33.0       1010.8


   Pressure3pm  Temp9am  Temp3pm RainToday WindGustDir WindDir9am
WindDir3pm
0       1007.1     16.9     21.8        No           W          W
WNW
1       1007.8     17.2     24.3        No         WNW        NNW
WSW
2       1008.7     21.0     23.2        No         WSW          W
WSW
3       1012.8     18.1     26.5        No          NE         SE
E
4       1006.0     17.8     29.7        No           W        ENE
NW
```

```
x_main=x.drop(['Date','Location','WindGustDir','WindDir9am','WindDir3p
m'],axis=1)
x_main.head()
```

```
4           7.0           20.0          82.0          33.0        1010.8


   Pressure3pm  Temp9am  Temp3pm RainTomorrow RainToday WindGustDir  \
0       1007.1     16.9     21.8           No        No           W
1       1007.8     17.2     24.3           No        No         WNW
2       1008.7     21.0     23.2           No        No         WSW
3       1012.8     18.1     26.5           No        No          NE
4       1006.0     17.8     29.7           No        No           W

  WindDir9am WindDir3pm
0          W        WNW
1        NNW        WSW
2          W        WSW
3         SE          E
4        ENE         NW
```

```
df=data.dropna()
df.head()
```

```
         Date Location  MinTemp  MaxTemp  Rainfall  WindGustSpeed   \
0  2008-12-01   Albury     13.4     22.9       0.6           44.0
1  2008-12-02   Albury      7.4     25.1       0.0           44.0
2  2008-12-03   Albury     12.9     25.7       0.0           46.0
3  2008-12-04   Albury      9.2     28.0       0.0           24.0
4  2008-12-05   Albury     17.5     32.3       1.0           41.0

   WindSpeed9am  WindSpeed3pm  Humidity9am  Humidity3pm
Pressure9am  \
0          20.0          24.0          71.0          22.0        1007.7

1           4.0          22.0          44.0          25.0        1010.6

2          19.0          26.0          38.0          30.0        1007.6

3          11.0           9.0          45.0          16.0        1017.6

4           7.0          20.0          82.0          33.0        1010.8


   Pressure3pm  Temp9am  Temp3pm RainTomorrow RainToday WindGustDir  \
0       1007.1     16.9     21.8           No        No           W
1       1007.8     17.2     24.3           No        No         WNW
2       1008.7     21.0     23.2           No        No         WSW
3       1012.8     18.1     26.5           No        No          NE
4       1006.0     17.8     29.7           No        No           W

  WindDir9am WindDir3pm
```

```
   MinTemp  MaxTemp  Rainfall  WindGustSpeed  WindSpeed9am
WindSpeed3pm  \
0     13.4     22.9       0.6           44.0          20.0
24.0
1      7.4     25.1       0.0           44.0           4.0
22.0
2     12.9     25.7       0.0           46.0          19.0
26.0
3      9.2     28.0       0.0           24.0          11.0
9.0
4     17.5     32.3       1.0           41.0           7.0
20.0

   Humidity9am  Humidity3pm  Pressure9am  Pressure3pm  Temp9am
Temp3pm  \
0         71.0         22.0       1007.7       1007.1     16.9
21.8
1         44.0         25.0       1010.6       1007.8     17.2
24.3
2         38.0         30.0       1007.6       1008.7     21.0
23.2
3         45.0         16.0       1017.6       1012.8     18.1
26.5
4         82.0         33.0       1010.8       1006.0     17.8
29.7

  RainToday
0        No
1        No
2        No
3        No
4        No
```

```
x_p=pd.get_dummies(x_main,columns=['RainToday'])
x_p.head()
```

```
   MinTemp  MaxTemp  Rainfall  WindGustSpeed  WindSpeed9am
WindSpeed3pm  \
0     13.4     22.9       0.6           44.0          20.0
24.0
1      7.4     25.1       0.0           44.0           4.0
22.0
2     12.9     25.7       0.0           46.0          19.0
26.0
3      9.2     28.0       0.0           24.0          11.0
9.0
4     17.5     32.3       1.0           41.0           7.0
20.0

   Humidity9am  Humidity3pm  Pressure9am  Pressure3pm  Temp9am
```

```
Temp3pm  \
0          71.0         22.0        1007.7       1007.1      16.9
21.8
1          44.0         25.0        1010.6       1007.8      17.2
24.3
2          38.0         30.0        1007.6       1008.7      21.0
23.2
3          45.0         16.0        1017.6       1012.8      18.1
26.5
4          82.0         33.0        1010.8       1006.0      17.8
29.7

   RainToday_No  RainToday_Yes
0             1              0
1             1              0
2             1              0
3             1              0
4             1              0
```

```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
y_main=pd.DataFrame(lb.fit_transform(y),columns=['RainTomorrow'])
y_main.head()
```

```
   RainTomorrow
0             0
1             0
2             0
3             0
4             0
```

```
from sklearn.preprocessing import StandardScaler

names = x.columns

names
```

```
Index(['Date', 'Location', 'MinTemp', 'MaxTemp', 'Rainfall',
'WindGustSpeed',
       'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am', 'Humidity3pm',
       'Pressure9am', 'Pressure3pm', 'Temp9am', 'Temp3pm',
'RainToday',
       'WindGustDir', 'WindDir9am', 'WindDir3pm'],
      dtype='object')
```

```
sc=StandardScaler()

x_scaled=pd.DataFrame(sc.fit_transform(x_p),columns=x_p.columns)
x_scaled.head()
```

7.2 Feature 2

```
    MinTemp   MaxTemp  Rainfall  WindGustSpeed  WindSpeed9am
WindSpeed3pm  \
0  0.189949 -0.045963 -0.207770       0.305395      0.677617
0.614796
1 -0.749180  0.263481 -0.279002       0.305395     -1.130078
0.385479
2  0.111688  0.347875 -0.279002       0.457621      0.564636
0.844114
3 -0.467441  0.671385 -0.279002      -1.216867     -0.339212     -
1.105087
4  0.831687  1.276207 -0.160282       0.077056     -0.791135
0.156161

   Humidity9am  Humidity3pm  Pressure9am  Pressure3pm   Temp9am
Temp3pm  \
0     0.113867    -1.436005    -1.475400    -1.220931 -0.013524
0.016423
1    -1.312289    -1.289891    -1.045530    -1.116169  0.032829
0.380285
2    -1.629213    -1.046369    -1.490223    -0.981474  0.619960
0.220185
3    -1.259469    -1.728231    -0.007913    -0.367863  0.171886
0.700483
4     0.694893    -0.900255    -1.015884    -1.385559  0.125534
1.166225

   RainToday_No  RainToday_Yes
0      0.532962      -0.532962
1      0.532962      -0.532962
2      0.532962      -0.532962
3      0.532962      -0.532962
4      0.532962      -0.532962
```

```
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test =
train_test_split(x_scaled,y_main,test_size=0.2,random_state=0)
```

7.3.MODEL BUILDING

MODEL BULIDING

Training And Testing The Model

```
XGBoost=xgboost.XGBRFClassifier()
Rand_forest=sklearn.ensemble.RandomForestClassifier()
svm=sklearn.svm.SVC()
Dtree=sklearn.tree.DecisionTreeClassifier()
GBM=sklearn.ensemble.GradientBoostingClassifier()
log=sklearn.linear_model.LogisticRegression()

# Training the every model with Train data
model1=XGBoost.fit(x_train,y_train)
model2=Rand_forest.fit(x_train,y_train)
model3=svm.fit(x_train,y_train)
model4=Dtree.fit(x_train,y_train)
model5=GBM.fit(x_train,y_train)
model6=log.fit(x_train,y_train)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/preprocessing/
_label.py:98: DataConversionWarning: A column-vector y was passed when
a 1d array was expected. Please change the shape of y to
(n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.7/dist-packages/sklearn/preprocessing/_label.py
:133: DataConversionWarning: A column-vector y was passed when a 1d
array was expected. Please change the shape of y to (n_samples, ), for
example using ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3:
DataConversionWarning: A column-vector y was passed when a 1d array
was expected. Please change the shape of y to (n_samples,), for
example using ravel().
  This is separate from the ipykernel package so we can avoid doing
imports until
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:993
: DataConversionWarning: A column-vector y was passed when a 1d array
was expected. Please change the shape of y to (n_samples, ), for
example using ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.7/dist-packages/sklearn/ensemble/_gb.py:494:
DataConversionWarning: A column-vector y was passed when a 1d array
was expected. Please change the shape of y to (n_samples, ), for
example using ravel().
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:993
: DataConversionWarning: A column-vector y was passed when a 1d array
was expected. Please change the shape of y to (n_samples, ), for
```

8.TESTING

Testing Report

Testing of an individual software component or module is termed as Unit Testing. It is typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code.

The Code was developed in 3 separate parts-

1.AI Model developed using Jupyter Notebook

2.Web Front end was developed using VS Code

3.Backend Database was developed using MongoDB

PROJECT NAME	Exploratory Analysis of RainFall Data in India for Agriculture
PROJECT TYPE	APPLIED DATA SCIENCE
DEVELOPER	RITHIKAA, AISHWARYA, GOKUL PRASATH, SOUNDER.
LANGUAGE	PYTHON,HTML,CSS,JAVA SCRIPT
TOTAL NUMBER OF TEST CASES	25
NUMBER OF TEST CASES EXECUTED	23
NUMBER OF TEST CASES PASSED	20
NUMBER OF TEST CASES FAILED	2-DUE TO TECHNICAL ISSUES

UNIT TESTING

Unit testing is carried out screen-wise, each screen being identified as an object. Attention is diverted to individual modules, independently to one another to locate errors .This has enabled the detection of errors in coding and logic .This is the first level of testing. In this, codes are written such that from one module, we can move on to the next module according tothe choice winter.

SYSTEM TESTING

In this, the entire system was tested as a whole with all forms, code, modules and class modules .System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences.

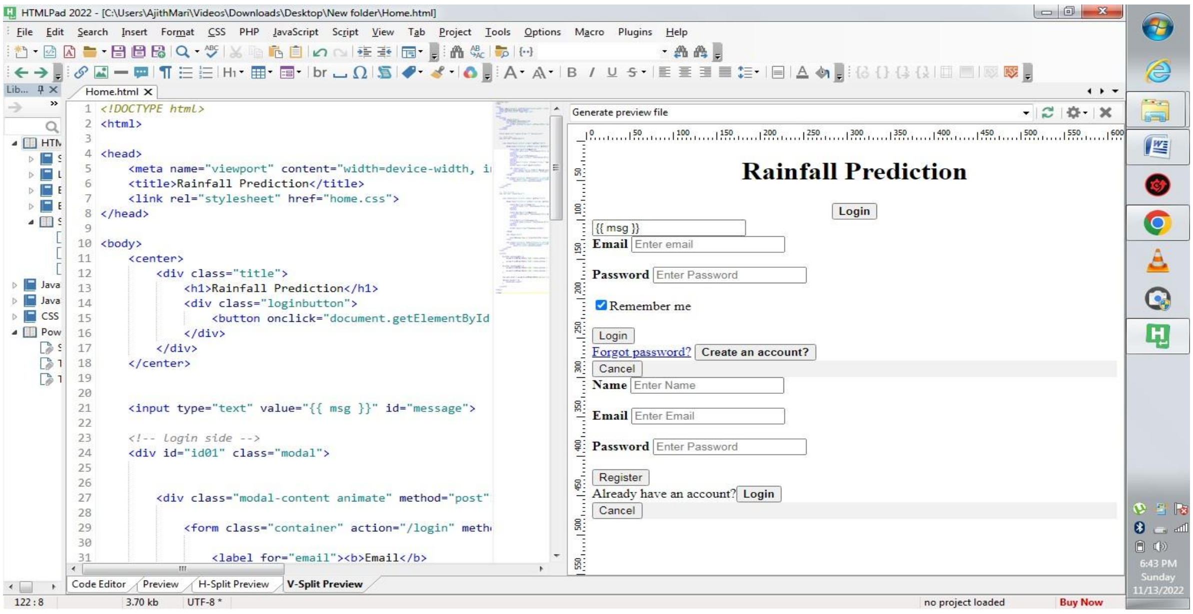
It is a series of different tests that verifies that all system elements have been properly integrated and perform allocated functions.

System testing makes logical assumptions that if all parts of the system are correct, the goalwill be successfully achieved. Testing is the process of executing the program with the intentof finding errors.

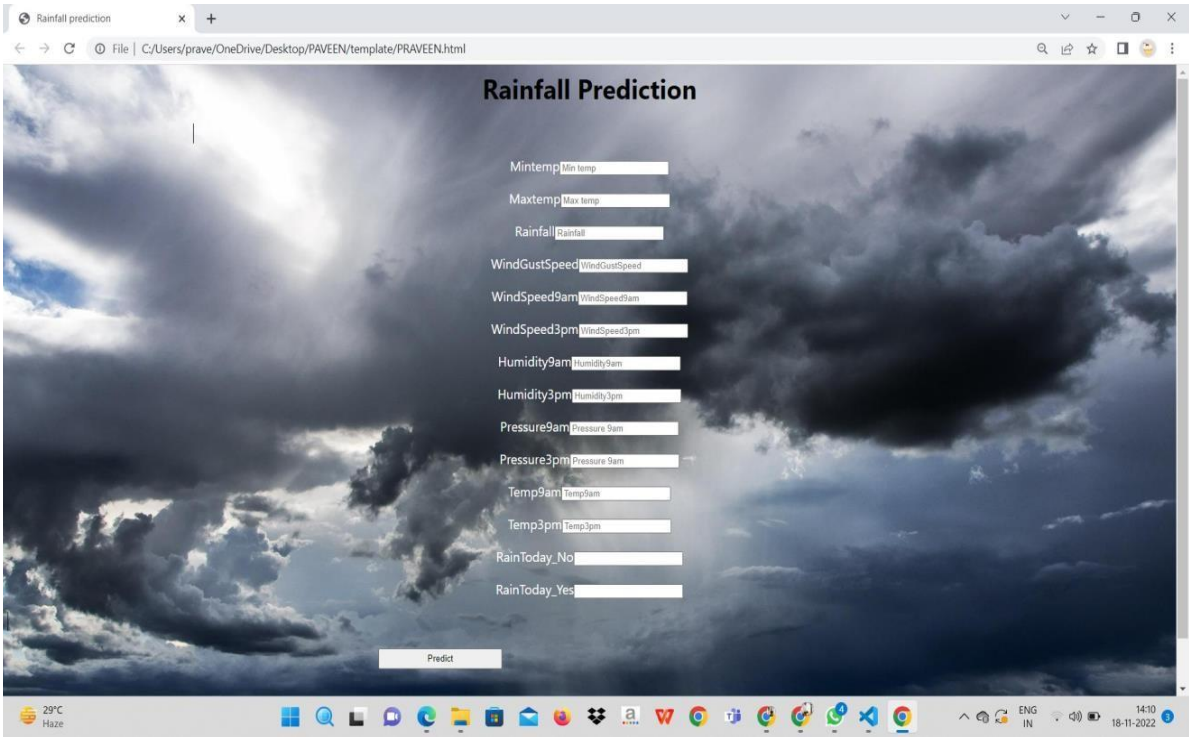
Testing cannot show the absence of defects, it can only show that software errors are present.

9.RESULTS

MODEL-1



MODEL-2



10.ADVANTAGES & DISADVANTAGES

- High prediction accuracy.
- Hold perfectly good for large scale datasets with large number of variables.
- Integral variable selection based on importance and variable interaction.
- Deals efficiently with data having missing values.
- Computation of relation between variables and classification.
- Proximity calculation between cases.
- Can be used for unsupervised learning and outlier detection.
- Internal unbiased estimation of the generalization error.

11.CONCLUSION

A detailed survey on rainfall predictions using Artificial Neural Network architecture over twenty-five years is done. From the survey it has been found that most of the researchers used different models for rainfall prediction, but keras model of ANN gives significant results. ANN is the model with least mean squared error and accurate prediction. The survey also gives a conclusion that the forecasting techniques like Decision Tree, Random Forest, KNN and XGBoost are suitable to predict rainfall than other forecasting techniques such as statistical and numerical methods. However, some limitation of those methods has been found. The extensive references in support of the different developments of ANN research provided should be of great help to ANN researchers to accurately predict rainfall in the future.

12.FUTURE SCOPE

Predicting the rainfall of a specific geographic location would be a challenge. Improvising the prediction model to predict the weather conditions and even predicting the loses of rainfall. Coping with the changing parameter values and making the oode compatible for the changes in the parameter values. Improvising the ANN algorithm to further reduce the mean squared error.