Exploratory Analysis of RainFall Data in India for Agriculture

A PROJECT REPORT

TEAM ID: PNT2022TMID30456

Submitted by

SAMAYA.S(611419106049)
ANITHA.M(611419106007)
DHANALAKSHMI.S(611419106021)
ARTHI.D(61141910612)

In partial fulfillment for the award of the degree

Of

DEPARTMENT OF ELECTRONICS AND COMMUNICATION

ENGINEERING

MAHENDRA ENGINEERING COLLEGE FOR WOMEN

KUMARAMANGALAM

CONTENTS

1. INTRODUCTION

1.1. Project Overview

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.

Agriculture is the backbone of the Indian economy. For agriculture, the most important thing iswater source, i.e., rainfall. The prediction of the amount of rainfall gives alertness to farmers byknowing early they can protect their crops from rain. So, it is important to predict the rainfallaccurately as much as possible. Exploration and analysis of data on rainfall over various regions of India and especially the regions where agricultural works have been done persistently in a widerange. With the help of analysis and the resultant data, future rainfall prediction for those regionsusing various machine learning techniques such as Logistic Regression, Linear Regression, CatboostClassifier etc.

PRE-REQUISTIES

Anaconda Installation:

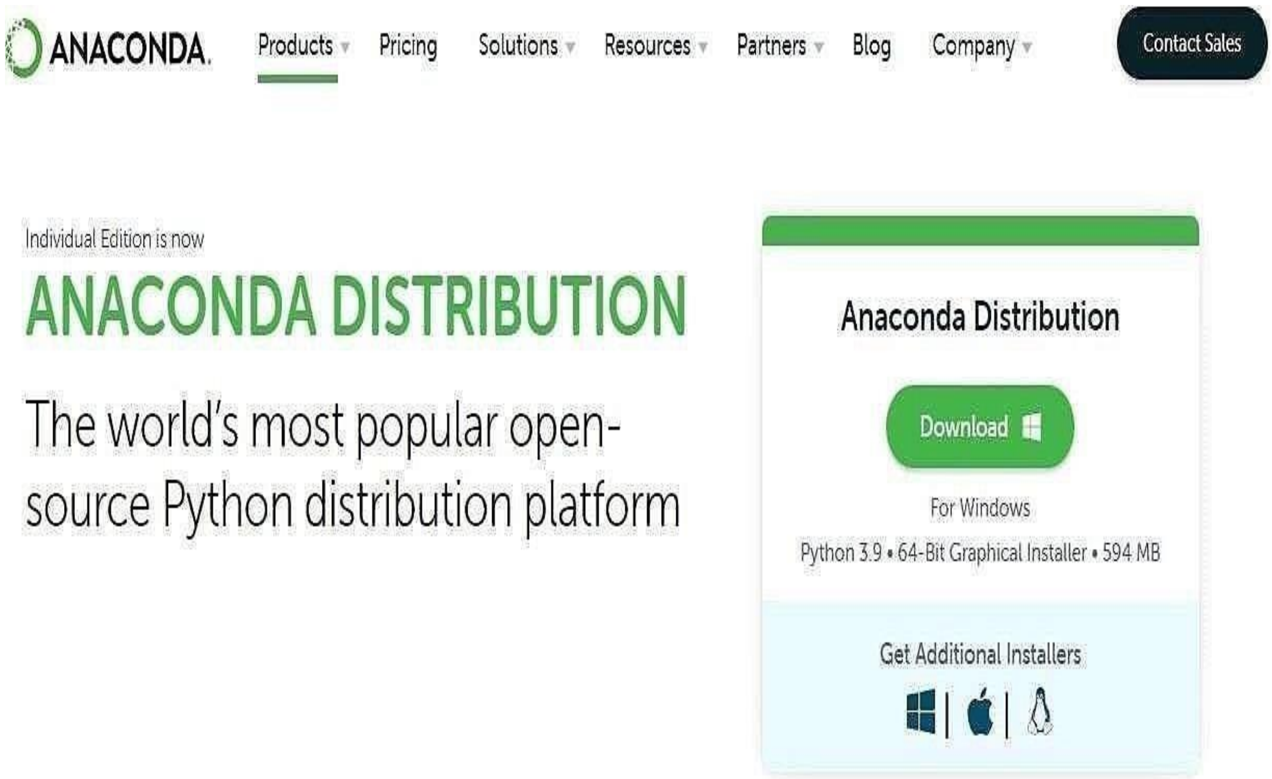
Anaconda is a distribution of the Python and R programming languages for scientific computing that aims to simplify package management and deployment. The distribution includes datascience packages suitable for Windows, Linux, and macOS. Developed and maintained byAnaconda.

Founded

in 2012 by Peter Wang and Travis Olyphant. As Anaconda, also known as Anaconda Distribution or Anaconda Individual Edition, the company's other products include hisAnaconda Team Edition and Anaconda Enterprise Edition, neither of which are free.

WAY TO INSTALL ANACONDA:

STEP 1: Download and Anaconda



Individual Edition is now

ANACONDA DISTRIBUTION

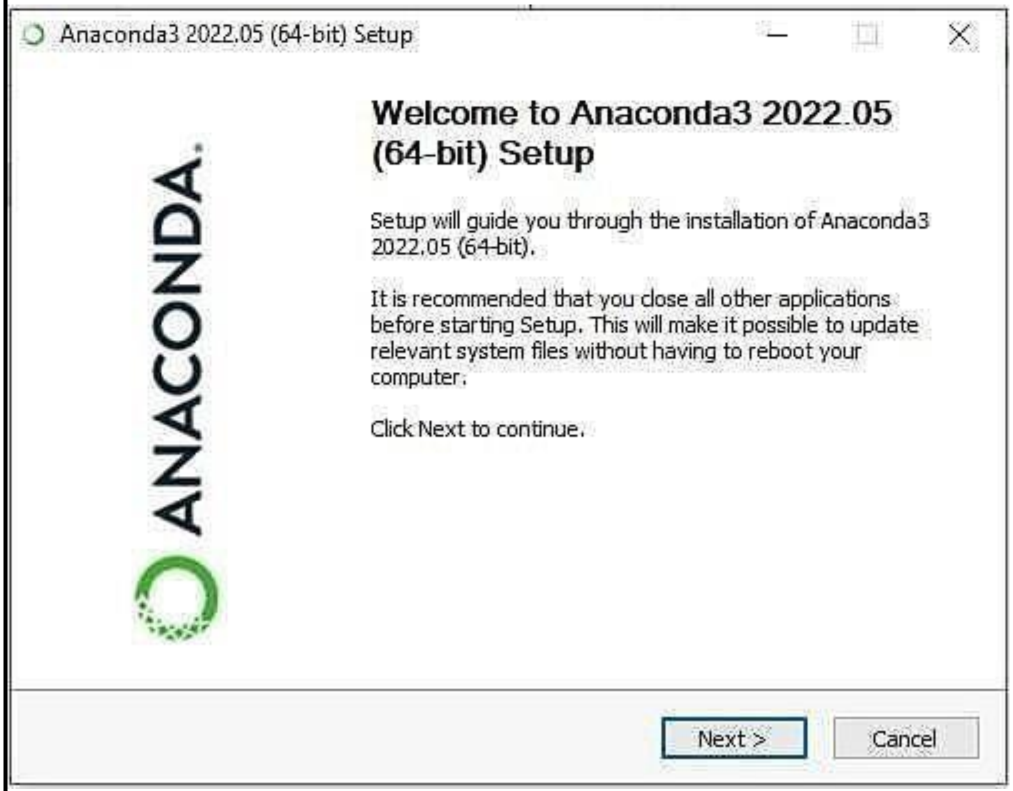
The world's most popular open-source Python distribution platform

Anaconda Distribution

For Windows

Python 3.9 • 64-Bit Graphical Installer • 594 MB

STEP 2: Install the Anaconda



Anaconda3 2022.05 (64-bit) Setup

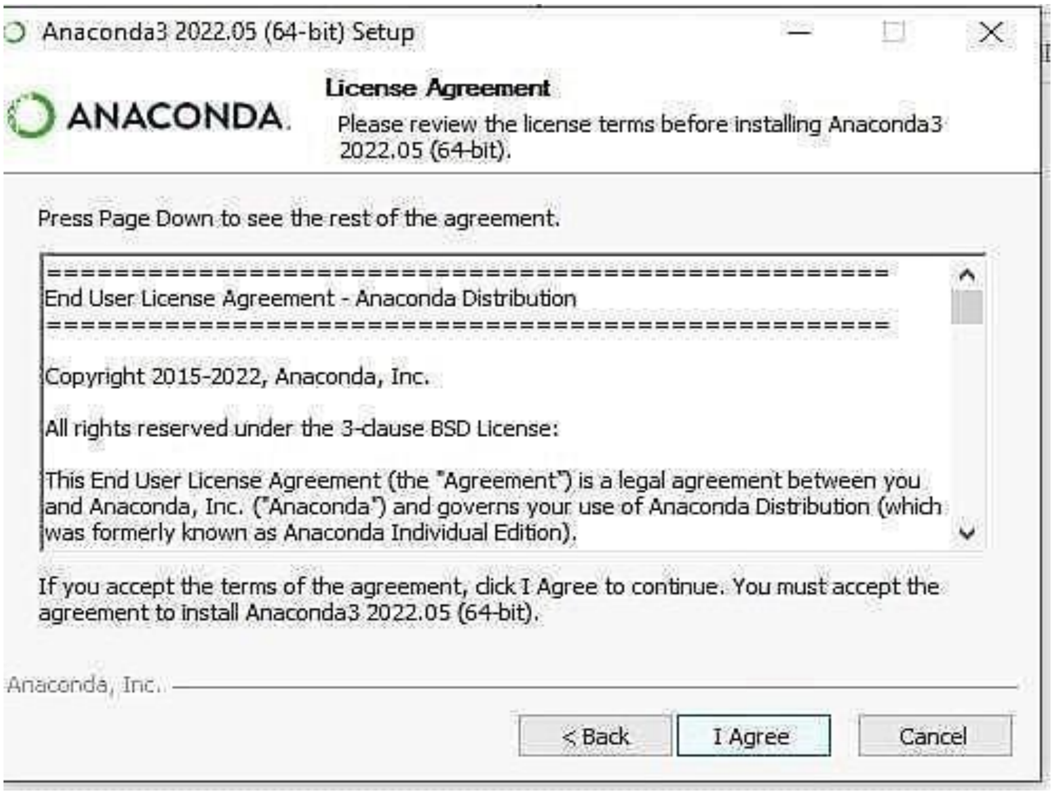
Welcome to Anaconda3 2022.05 (64-bit) Setup

Setup will guide you through the installation of Anaconda3 2022.05 (64-bit).

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

STEP 3: Click I Agree



Anaconda3 2022.05 (64-bit) Setup

License Agreement

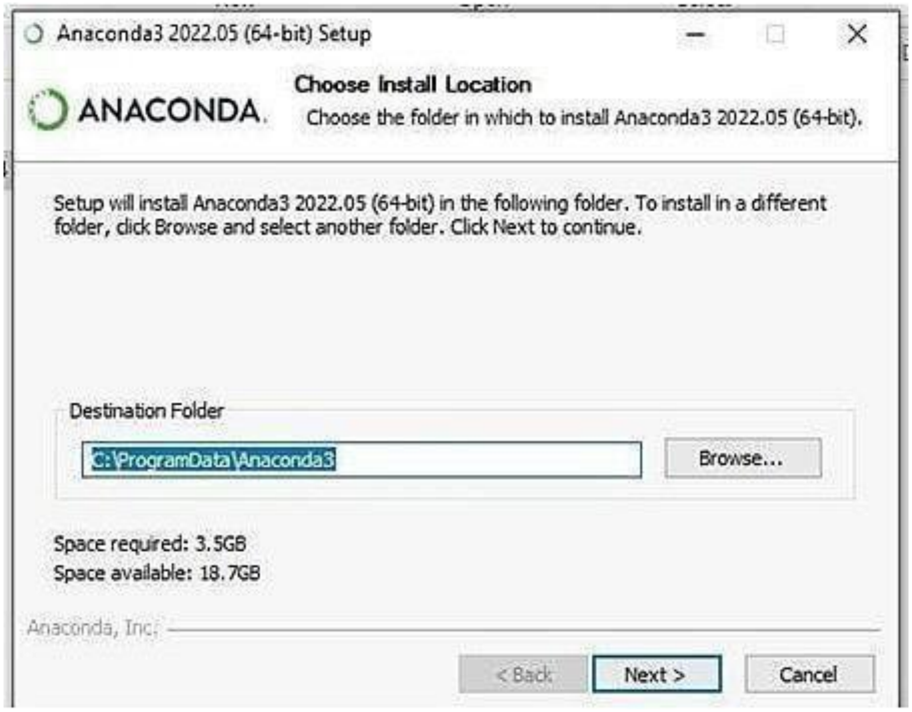
Please review the license terms before installing Anaconda3 2022.05 (64-bit).

Press Page Down to see the rest of the agreement.

```
=====================================================
End User License Agreement - Anaconda Distribution
=====================================================

Copyright 2015-2022, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

This End User License Agreement (the "Agreement") is a legal agreement between you
and Anaconda, Inc. ("Anaconda") and governs your use of Anaconda Distribution (which
was formerly known as Anaconda Individual Edition).
```

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Anaconda3 2022.05 (64-bit).

STEP 4: Choose the Installation Location



Anaconda3 2022.05 (64-bit) Setup

Choose Install Location

Choose the folder in which to install Anaconda3 2022.05 (64-bit).

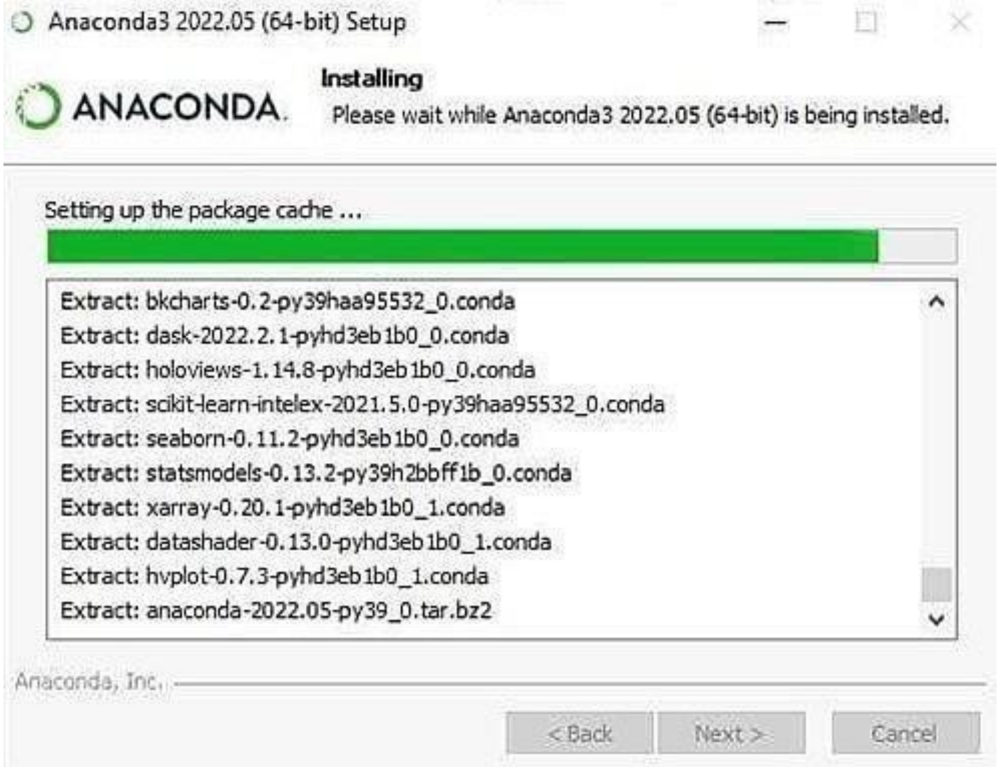
Setup will install Anaconda3 2022.05 (64-bit) in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder

C:\ProgramData\Anaconda3

Space required: 3.5GB
Space available: 18.7GB

STEP 5: Installing the Requiring packages



Anaconda3 2022.05 (64-bit) Setup

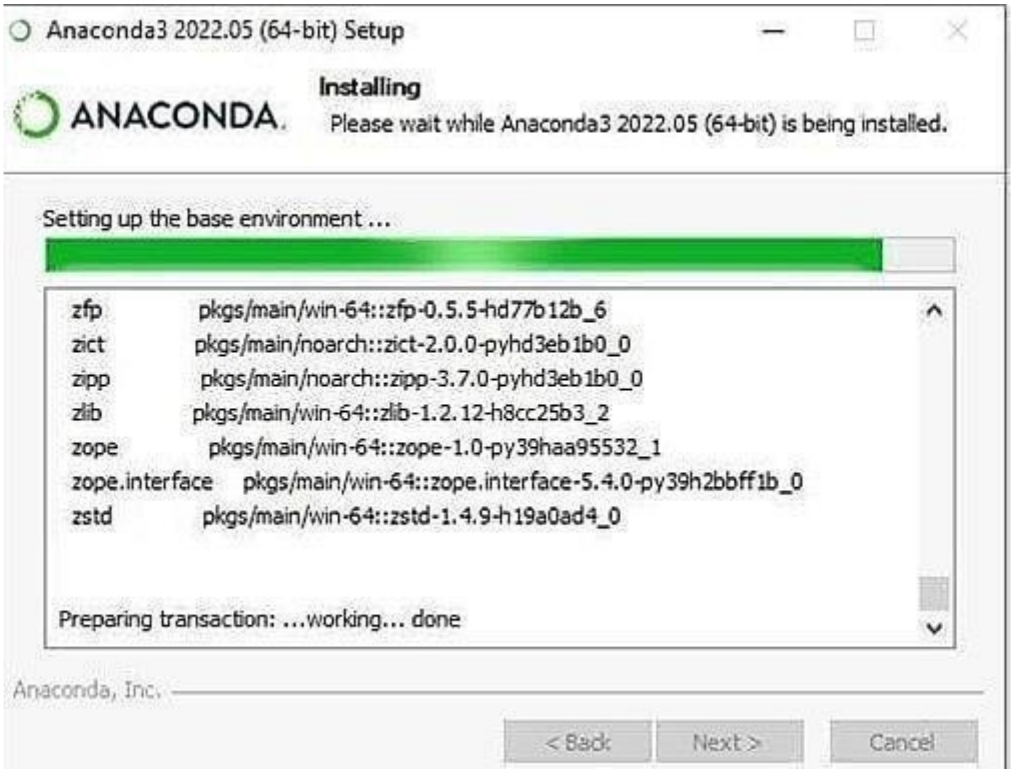
Installing

Please wait while Anaconda3 2022.05 (64-bit) is being installed.

Setting up the package cache ...

```
Extract: bkcharts-0.2-py39haa95532_0.conda
Extract: dask-2022.2.1-pyhd3eb1b0_0.conda
Extract: holoviews-1.14.8-pyhd3eb1b0_0.conda
Extract: scikit-learn-intelex-2021.5.0-py39haa95532_0.conda
Extract: seaborn-0.11.2-pyhd3eb1b0_0.conda
Extract: statsmodels-0.13.2-py39h2bbff1b_0.conda
Extract: xarray-0.20.1-pyhd3eb1b0_1.conda
Extract: datashader-0.13.0-pyhd3eb1b0_1.conda
Extract: hvplot-0.7.3-pyhd3eb1b0_1.conda
Extract: anaconda-2022.05-py39_0.tar.bz2
```

STEP 6: Setting up the base environment



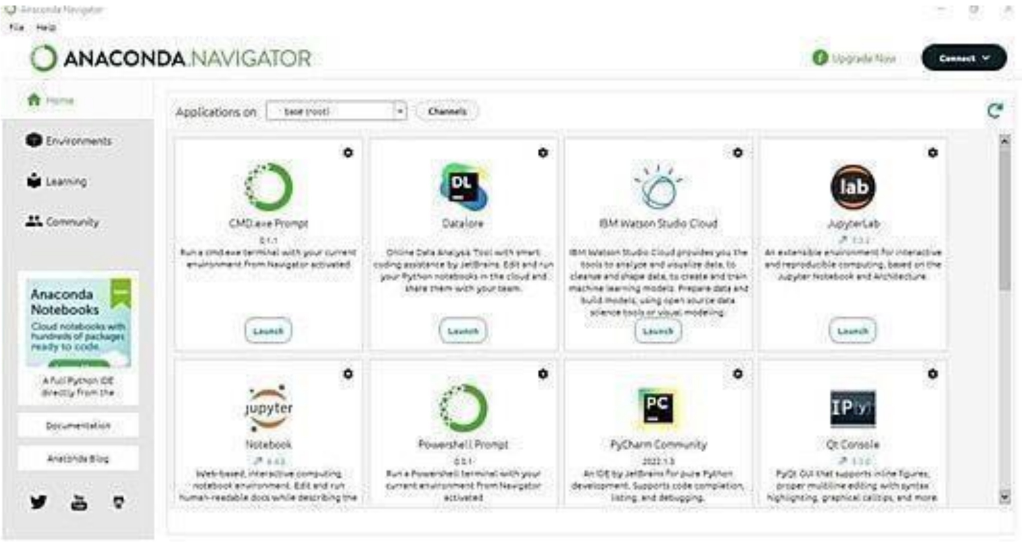
Installing

Please wait while Anaconda3 2022.05 (64-bit) is being installed.

Setting up the base environment ...

```
zfp             pkgs/main/win-64::zfp-0.5.5-hd77b12b_6
zict            pkgs/main/noarch::zict-2.0.0-pyhd3eb1b0_0
zipp            pkgs/main/noarch::zipp-3.7.0-pyhd3eb1b0_0
zlib            pkgs/main/win-64::zlib-1.2.12-h8cc25b3_2
zope            pkgs/main/win-64::zope-1.0-py39haa95532_1
zope.interface  pkgs/main/win-64::zope.interface-5.4.0-py39h2bbff1b_0
zstd            pkgs/main/win-64::zstd-1.4.9-h19a0ad4_0


Preparing transaction: ...working... done
```

STEP 7: Successfully Installed and check the Anaconda Navigator working or not



Run a cmd.exe terminal with your current environment from Navigator activated

Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the

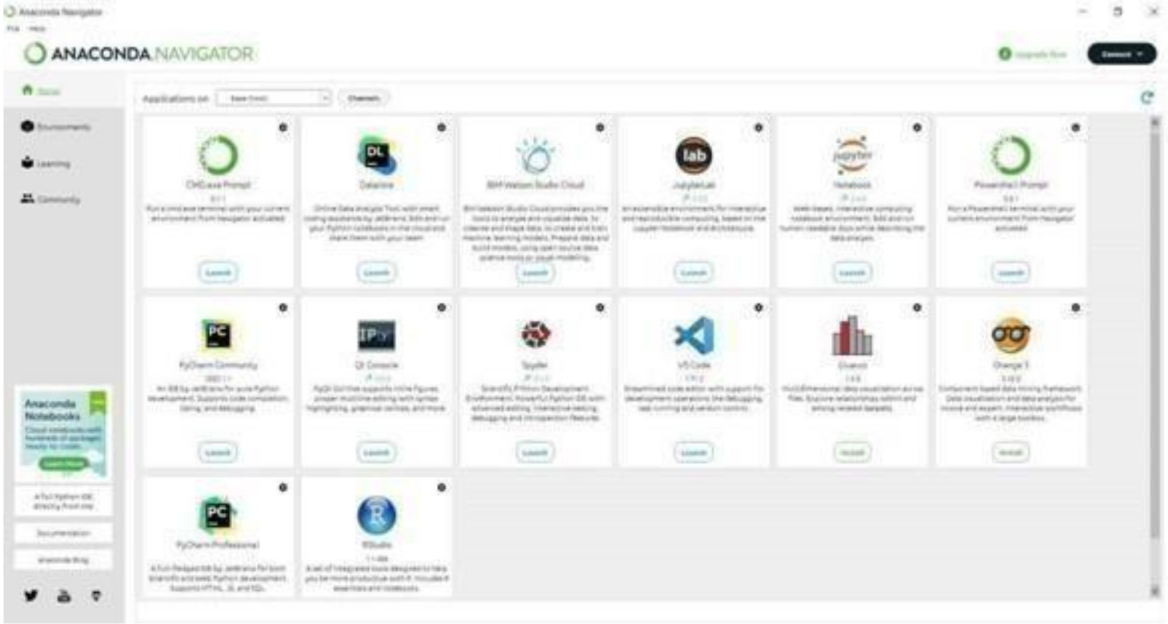
Run a Powershell terminal with your current environment from Navigator activated

An IDE by JetBrains for pure Python development. Supports code completion, listing, and debugging.

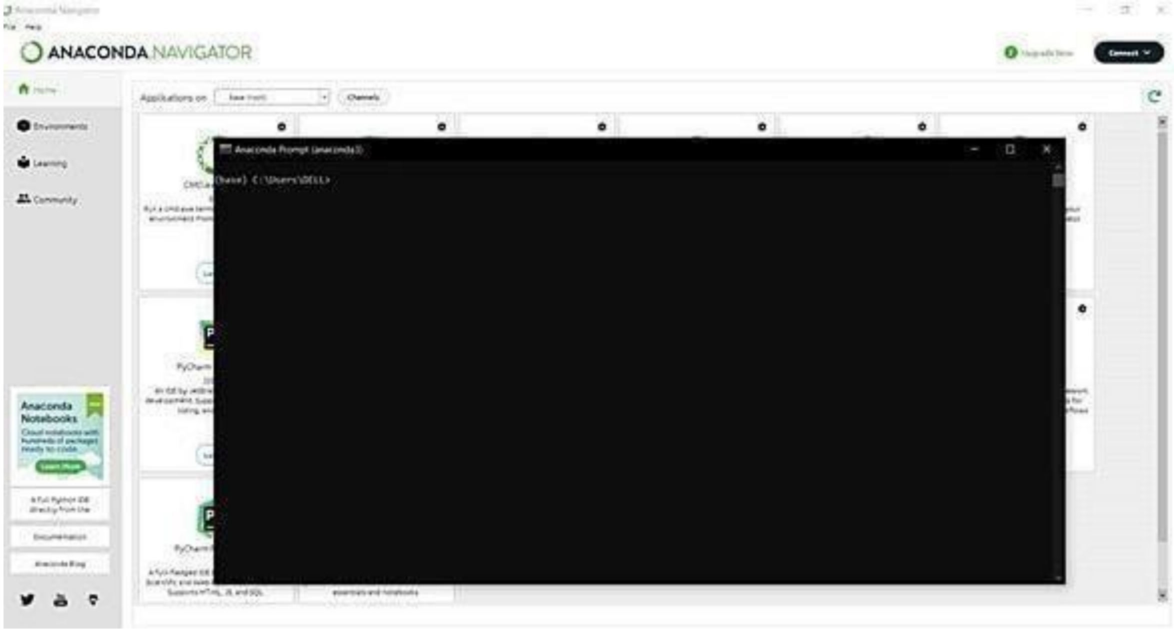
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Python packages installation:

Step 1: Open the anaconda navigator in the start menu



Step 2: Open the CMD.exe prompt



```
(base) C:\Users\DELL>
```

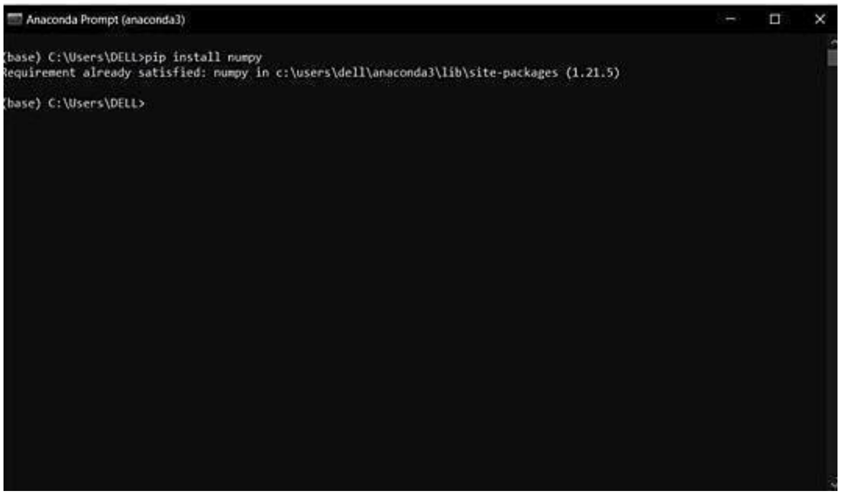
Step 3: Install the NUMPY package

To enter the numpy package enter the command in the

CMD.exeCommand: **Pip install numpy**

Numpy:

This package is used to perform numerical computations. This package comes pre-installed withAnaconda. NumPy is used for manipulating arrays. NumPy stands for Numerical Python.



```
(base) C:\Users\DELL>pip install numpy
Requirement already satisfied: numpy in c:\users\dell\anaconda3\lib\site-packages (1.21.5)

(base) C:\Users\DELL>
```

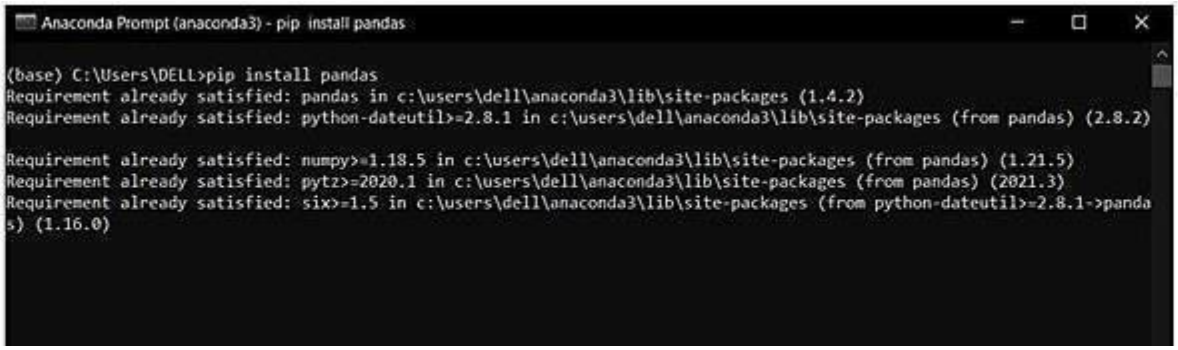
Step 4: Install the pandas package.

To enter the pandas package enter the command in the

CMD.exeCommand: **Pip install pandas**

Pandas:

Pandas is one of the most widely used Python libraries for data science. It provides powerful and easy-to-use structure and data analysis tools. This package comes pre- installed with Anaconda. An open source library built on top of the NumPy library. A Python package that provides various data structures and operations for working with numerical data and time series. Mainly, it's common for data to be imported and analyzed much easier. Pandas is fast, providing users with high performance and productivity.



```
(base) C:\Users\DELL>pip install pandas
Requirement already satisfied: pandas in c:\users\dell\anaconda3\lib\site-packages (1.4.2)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\dell\anaconda3\lib\site-packages (from pandas) (2.8.2)

Requirement already satisfied: numpy>=1.18.5 in c:\users\dell\anaconda3\lib\site-packages (from pandas) (1.21.5)
Requirement already satisfied: pytz>=2020.1 in c:\users\dell\anaconda3\lib\site-packages (from pandas) (2021.3)
Requirement already satisfied: six>=1.5 in c:\users\dell\anaconda3\lib\site-packages (from python-dateutil>=2.8.1->panda
s) (1.16.0)
```

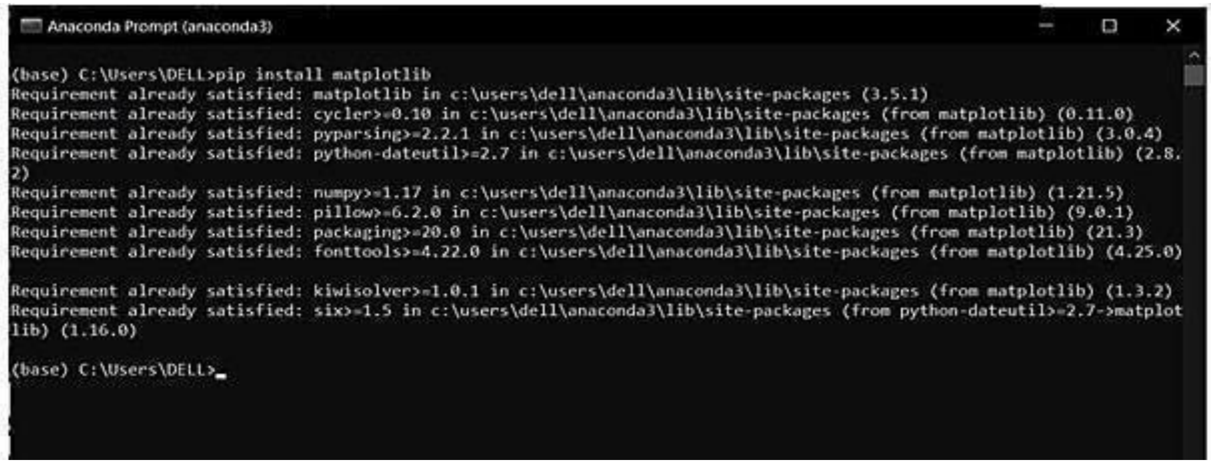
Step 5: Install the Matplotlib package.

To enter the Matplotlib package enter the command in the

CMD.exeCommand: **Pip install Matplotlib**

Matplotlib:

Matplotlib is a comprehensive library for creating static, animated and interactive visualizations in Python. This package comes pre-installed with Anaconda. Matplotlib is a nice visualization library in Python for 2D plotting of arrays. Matplotlib is a cross-platform data visualization librarybased on NumPy arrays and designed to work with the wider SciPy stack. Introduced by John Hunter in 2002.



```
(base) C:\Users\DELL>pip install matplotlib
Requirement already satisfied: matplotlib in c:\users\dell\anaconda3\lib\site-packages (3.5.1)
Requirement already satisfied: cycler>=0.10 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (3.0.4)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (2.8.
2)
Requirement already satisfied: numpy>=1.17 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (1.21.5)
Requirement already satisfied: pillow>=6.2.0 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (9.0.1)
Requirement already satisfied: packaging>=20.0 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (21.3)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (4.25.0)

Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\dell\anaconda3\lib\site-packages (from matplotlib) (1.3.2)
Requirement already satisfied: six>=1.5 in c:\users\dell\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplot
lib) (1.16.0)

(base) C:\Users\DELL>_
```

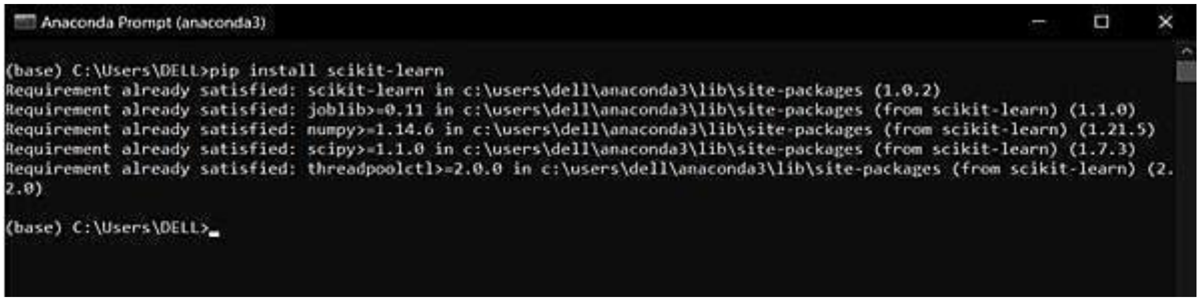
Step 6: Install the Scikit-learn package.

To enter the Scikit- learn package enter the command in the

CMD.exeCommand: **Pip install Scikit-learn**

Scikit-learn:

This is a machine learning library for the Python programming language. This package comes pre-installed with Anaconda. Scikit Learn in Python is primarily used to focus on modeling in Python. It was only focused on modeling, not loading data.



```
(base) C:\Users\DELL>pip install scikit-learn
Requirement already satisfied: scikit-learn in c:\users\dell\anaconda3\lib\site-packages (1.0.2)
Requirement already satisfied: joblib>=0.11 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn) (1.1.0)
Requirement already satisfied: numpy>=1.14.6 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn) (1.21.5)
Requirement already satisfied: scipy>=1.1.0 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn) (1.7.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn) (2.
2.0)

(base) C:\Users\DELL>_
```

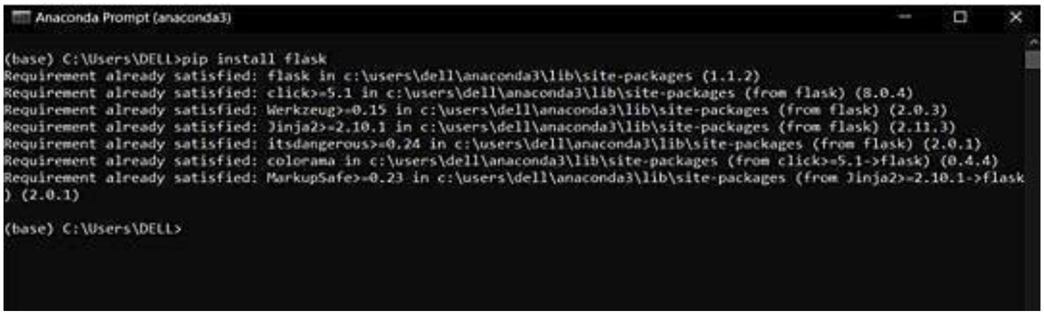
Step 7: Install the Flask package.

To enter the Flask package enter the command in

the CMD.exeCommand: **Pip install Flask**

Flask:

Flask is a lightweight WSGI web application framework Flask is a web application framework written in Python. It is developed by Armin Ronacher, who leads an internationalgroup of Python enthusiasts called Pocco. Flask is based on the WSGI toolkit tools and the Jinja2 template engine. Both are Pocco projects.



```
(base) C:\Users\DELL>pip install flask
Requirement already satisfied: flask in c:\users\dell\anaconda3\lib\site-packages (1.1.2)
Requirement already satisfied: click>=5.1 in c:\users\dell\anaconda3\lib\site-packages (from flask) (8.0.4)
Requirement already satisfied: Werkzeug>=0.15 in c:\users\dell\anaconda3\lib\site-packages (from flask) (2.0.3)
Requirement already satisfied: Jinja2>=2.10.1 in c:\users\dell\anaconda3\lib\site-packages (from flask) (2.11.3)
Requirement already satisfied: itsdangerous>=0.24 in c:\users\dell\anaconda3\lib\site-packages (from flask) (2.0.1)
Requirement already satisfied: colorama in c:\users\dell\anaconda3\lib\site-packages (from click>=5.1->flask) (0.4.4)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\dell\anaconda3\lib\site-packages (from Jinja2>=2.10.1->flask
) (2.0.1)

(base) C:\Users\DELL>
```

1.2 Purpose

The main aim of objective is to find the

- Rainfall Prediction is the application of science and technology to predict the amount of rainfall over a region.
- It is important to exactly determine the rainfall for effective use of water resources, crop productivity and pre-planning of water structures.

LITERATURE SURVEY

1.2. Existing Problem

Climate is important aspect of human life. So, the Prediction should accurate as much as possible. In this paper we try to deal with the prediction of the rainfall which is also a major aspect of human life, and which provide the major resource of human life which is Fresh Water. Fresh water is always a crucial resource of human survival – not only for the drinking purposes but also for farming, washing and many other purposes. Making a good prediction of climate is always a major task because of the climate change.

Now climate change is the biggest issue all over the world. Peoples are working on to detect the patterns in climate change as it affects the economy in production to infrastructure. So as in rainfall also making prediction of rainfall is a challenging task with a good accuracy rate. Making prediction on rainfall cannot be done by the traditional way, so scientist is using machine learning and deep learning to find out the pattern for rainfall prediction.

A bad rainfall prediction can affect the agriculture mostly framers as their whole crop is dependent on the rainfall and agriculture. It is always an important part of every economy. So, making an accurate prediction on the rainfall. There are number of techniques are used of machine learning, but accuracy is always a matter of concern in prediction made in rainfall.

There are number of causes made by rainfall affecting the world ex. Drought, Flood, and intense summer heat etc. And it will also affect water resources around the world.

1.3. References

PROJECT TITLE	AUTHOR	OBJECTIVE/OUTCOME
Spatialanalysis of Indian Summer monsoon Rainfall (Mar 26,2014)	Markan Oza d C.M.Kishtawal	Understanding the variability in rainfall, analysis of IndianSummer monsoon rainfall using Spatial resolution.
Climate impacts on Indian Agriculture. (16 June,2004)	K.Krish kumar na K.Rupa Kumar R.G.Ashrit N.R.Deshpande J.W.Hansen	Presents about the analysis of Crop-climate relationships for India, using historical predictions.
Exploratory data Analysis of Indian Rainfall Data	Anusha Gajinkar	This Study shows that, India has two monsoon rainfall season one is northwest monsoon and second one is southeast monsoon.

1.4. Problem Statement Definition

❖ Climate is a important aspect of human life. So, the Prediction should accurate as much as possible. In this paper we try to deal with the prediction of the rainfall which is also a major aspect of human life and which provide the major resource of human life which is Fresh Water. Fresh water is always a crucial resource of human survival – not only for the drinking purposes but also for farming,

❖ Making a good prediction of climate is always a major task now a day because of the climate change.

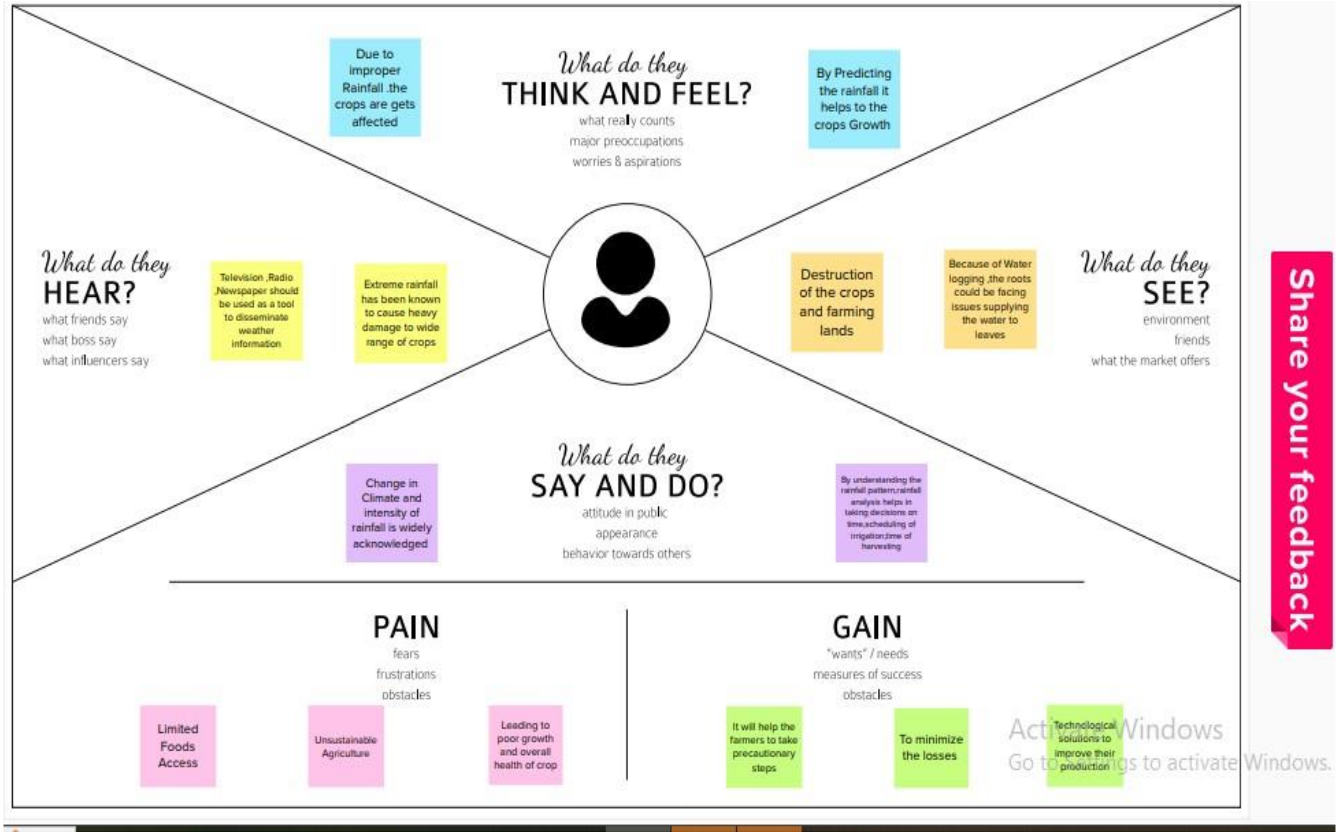
❖ Now climate change is the biggest issue all over the world. Peoples are working on to detect the patterns

in climate change as it affects the economy in production to infrastructure. So as in rainfall also making prediction of rainfall is a challenging task with a good accuracy rate. Making prediction on
rainfall cannot be done by the traditional way, so scientist is using machine learning and deep learning to
find out the pattern for rainfall prediction.

- A bad rainfall prediction can affect the agriculture mostly framers as their whole crop is depend on
 the rainfall and agriculture is always an important part of every economy. So, making an accurate prediction
 of the rainfall.somewhat good

2. IDEATION AND PROPOSED SOLUTION

2.1. Empathy Map Canvas



What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations

Due to improper Rainfall .the crops are gets affected

By Predicting the rainfall it helps to the crops Growth

What do they
HEAR?
what friends say
what boss say
what influencers say

Television ,Radio ,Newspaper should be used as a tool to disseminate weather information

Extreme rainfall has been known to cause heavy damage to wide range of crops

What do they
SEE?
environment
friends
what the market offers

Destruction of the crops and farming lands

Because of Water logging ,the roots could be facing issues supplying the water to leaves

What do they
SAY AND DO?
attitude in public
appearance
behavior towards others

Change in Climate and intensity of rainfall is widely acknowledged

By understanding the rainfall pattern,rainfall analysis helps in taking decisions on time,scheduling of irrigation,time of harvesting

PAIN
fears
frustrations
obstacles

Limited Foods Access

Unsustainable Agriculture

Leading to poor growth and overall health of crop

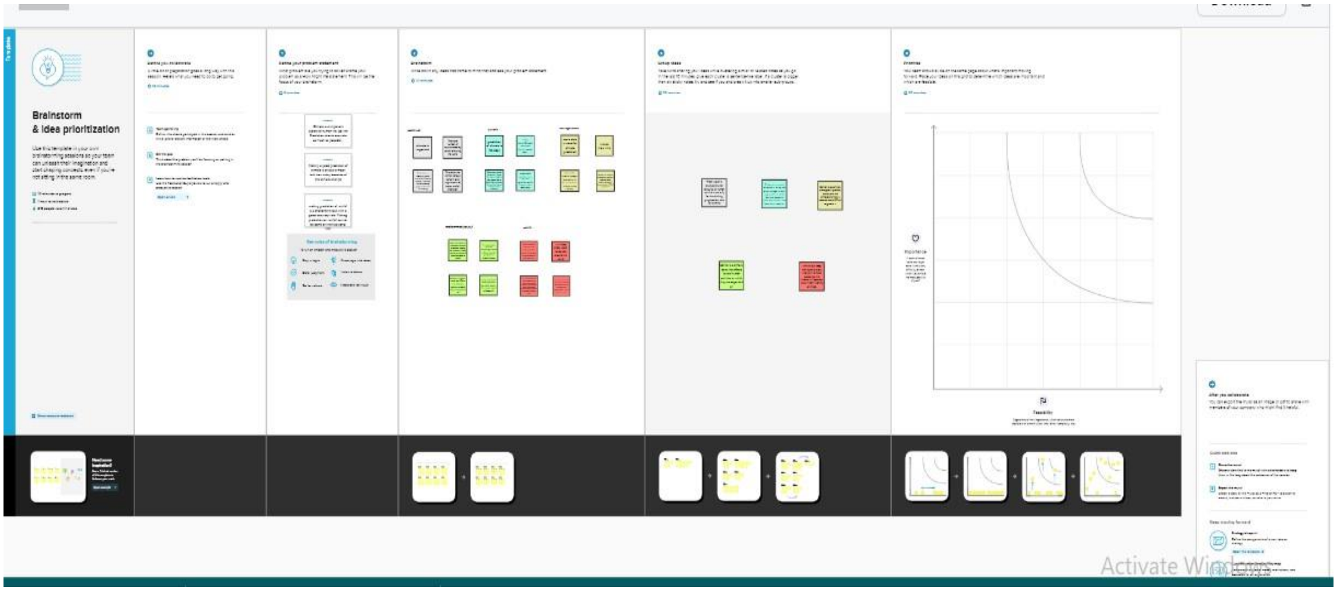
GAIN
"wants" / needs
measures of success
obstacles

It will help the farmers to take precautionary steps

To minimize the losses

Technological solutions to improve their production

2.2. Ideation and Brainstorming

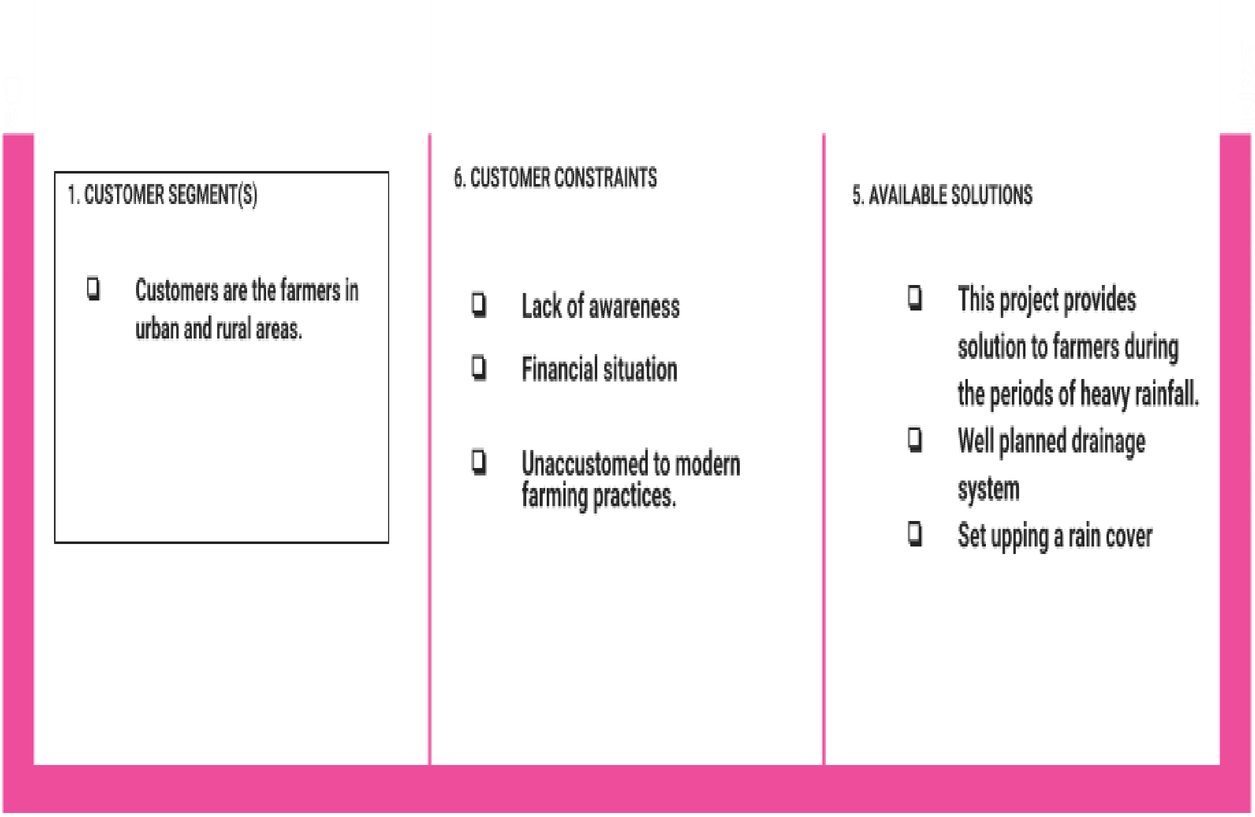


Brainstorm
& idea prioritization

2.3. Proposed Solution

S.No.	**Parameter**	**Description**
1.	Problem Statement	Climate is a important aspect of human life. So, the Prediction should accurate as much as possible. In this paper we try to deal with the prediction of the rainfall which is also a major aspect of human life and which provide the major resource of human life which is Fresh Water. • Now climate change is the biggest issue all over the world. Peoples are working on to detect the patterns in climate change as it affects the economy in production to infrastructure.
2.	Proposed Solution	Analyzing the previous 10 years data can give us a rough idea about Rainfall pattern. Using Data Science, we can predict the Rainfall up to some good extent.
3.	Uniqueness	• This application is useful for the beginners in agriculture. • Seed maturity selection features are available.
4.	Social Impact	• Different types of crops can be planted for good health. • Helps in producing healthy crops and good fields.
5.	Business Model	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
6.	Scalability	• When we predict rainfall correctly,it helps growth of crop and yielding will be better.

2.4. Proposed Solution Fit



1. CUSTOMER SEGMENT(S)

❑ Customers are the farmers in urban and rural areas.

6. CUSTOMER CONSTRAINTS

❑ Lack of awareness

❑ Financial situation

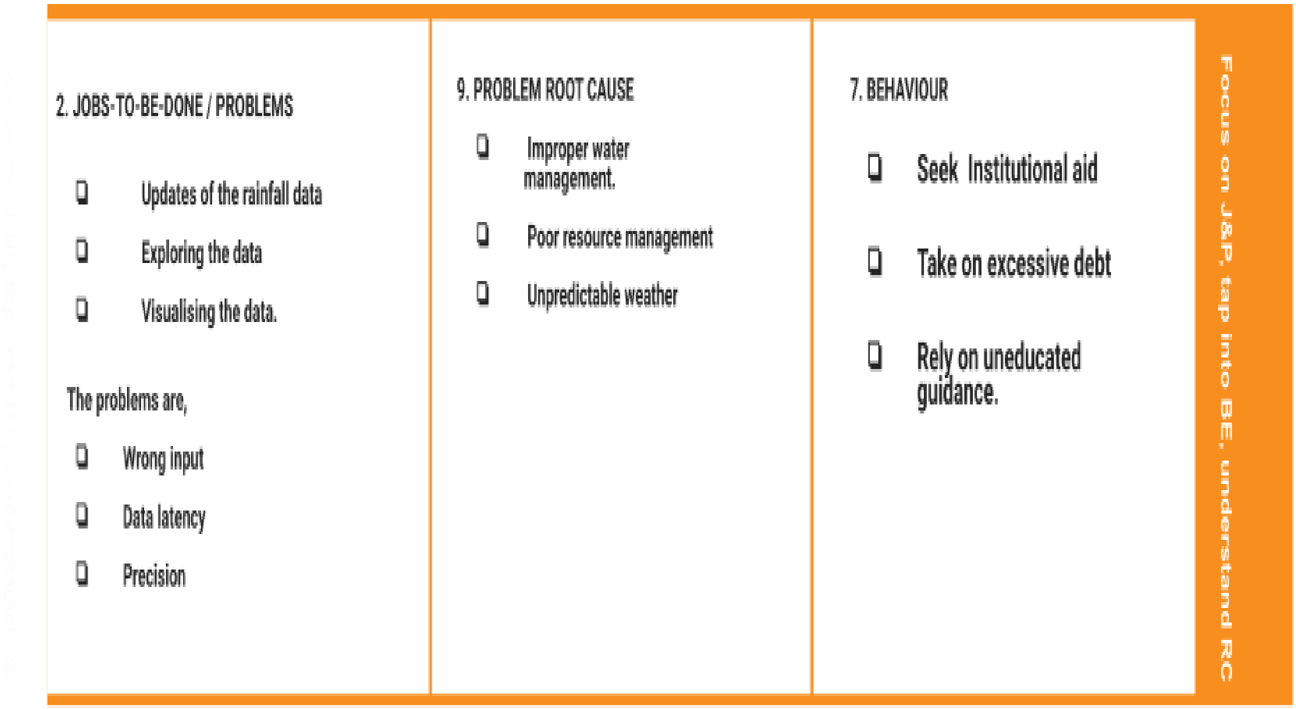
❑ Unaccustomed to modern farming practices.

5. AVAILABLE SOLUTIONS

❑ This project provides solution to farmers during the periods of heavy rainfall.

❑ Well planned drainage system

❑ Set upping a rain cover



2. JOBS-TO-BE-DONE / PROBLEMS

❑ Updates of the rainfall data

❑ Exploring the data

❑ Visualising the data.

The problems are,

❑ Wrong input

❑ Data latency

❑ Precision

9. PROBLEM ROOT CAUSE

❑ Improper water management.

❑ Poor resource management

❑ Unpredictable weather

7. BEHAVIOUR

❑ Seek Institutional aid

❑ Take on excessive debt

❑ Rely on uneducated guidance.

Focus on J&P, tap into BE, understand RC

3. REQUIREMENT ANALYSIS

3.1. Functional Requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Import necessary packages	Import necessary packages Importing packages like NumPy, pandas,seaborn, etc
FR-2	Download and load dataset	Download the dataset Load the Appropriate dataset
FR-3	Pre-processing of data	Making data suitable for building a good model
FR-4	Building Machine learning model	Choose the best algorithm. Check for the best optimised result.
FR-5	Train the data	Train the model using training data.
FR-6	Test the mode	Test the model for the best evaluation and analysing..

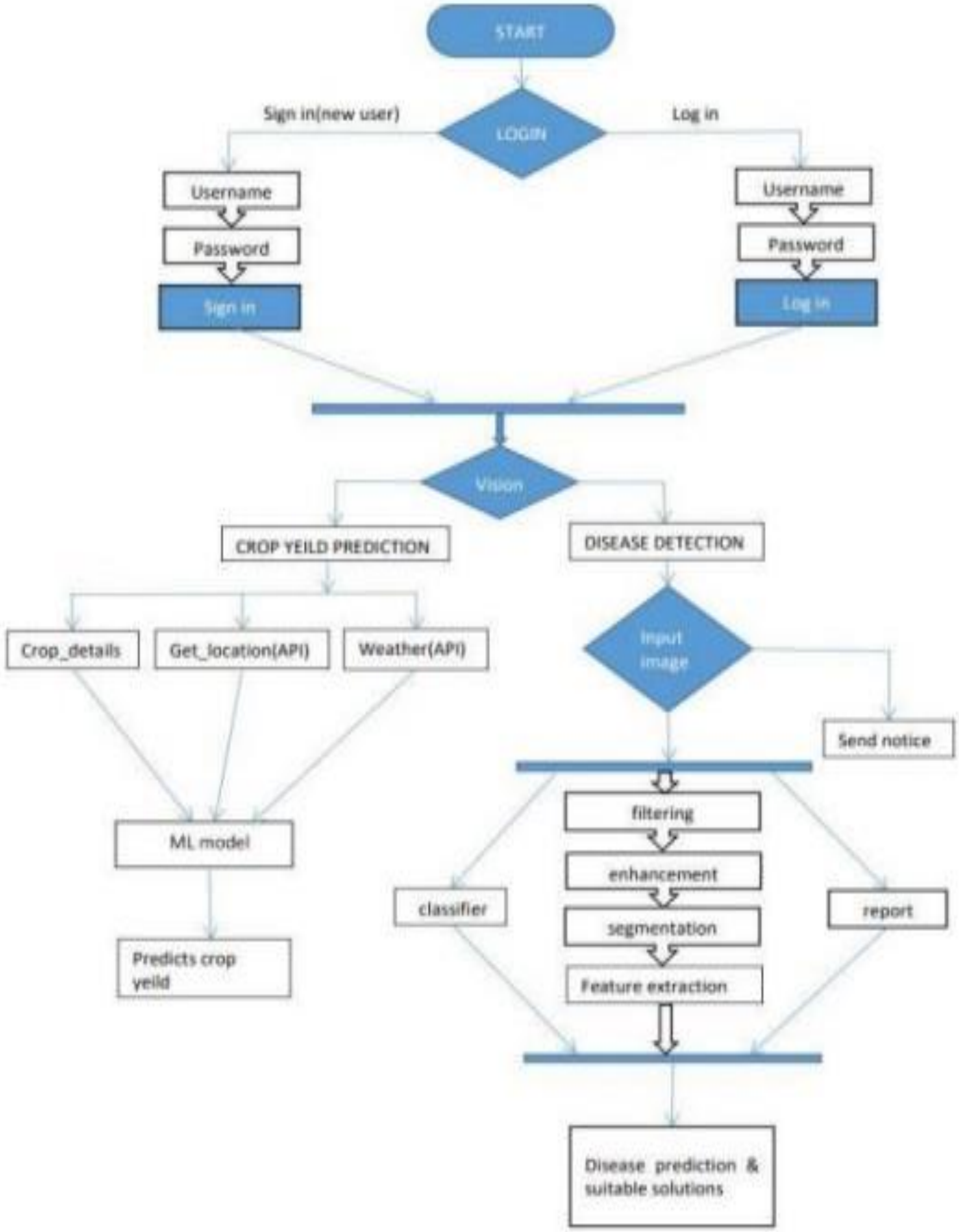
3.2. Non-Functional Requirements

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The usability of the website is to make all users willbe satisfied with our requirements of the product. The user should reach the summarized text or resultwith one button press if possible
NFR-2	**Security**	The security of the project is to develop the website that prevents SQL injection attack, XSS attack and DOS attack
NFR-3	**Reliability**	The reliability of the system is to make sure the websitedoes not go offline. The users can be reach and use program at any time,so maintenance should not be big issue.
NFR-4	**Performance**	The performance of the website isto provide data to allusers without unnecessary delay and provide 24*7 availability.

NFR-5	**Availability**	The availability of the website is that the website will be active on The Internet and people will be able to browse to it.
NFR-6	**Scalability**	The scalability of the system is we have limited our project to Indian cities We have plans to scale it to continent's level in coming updates.

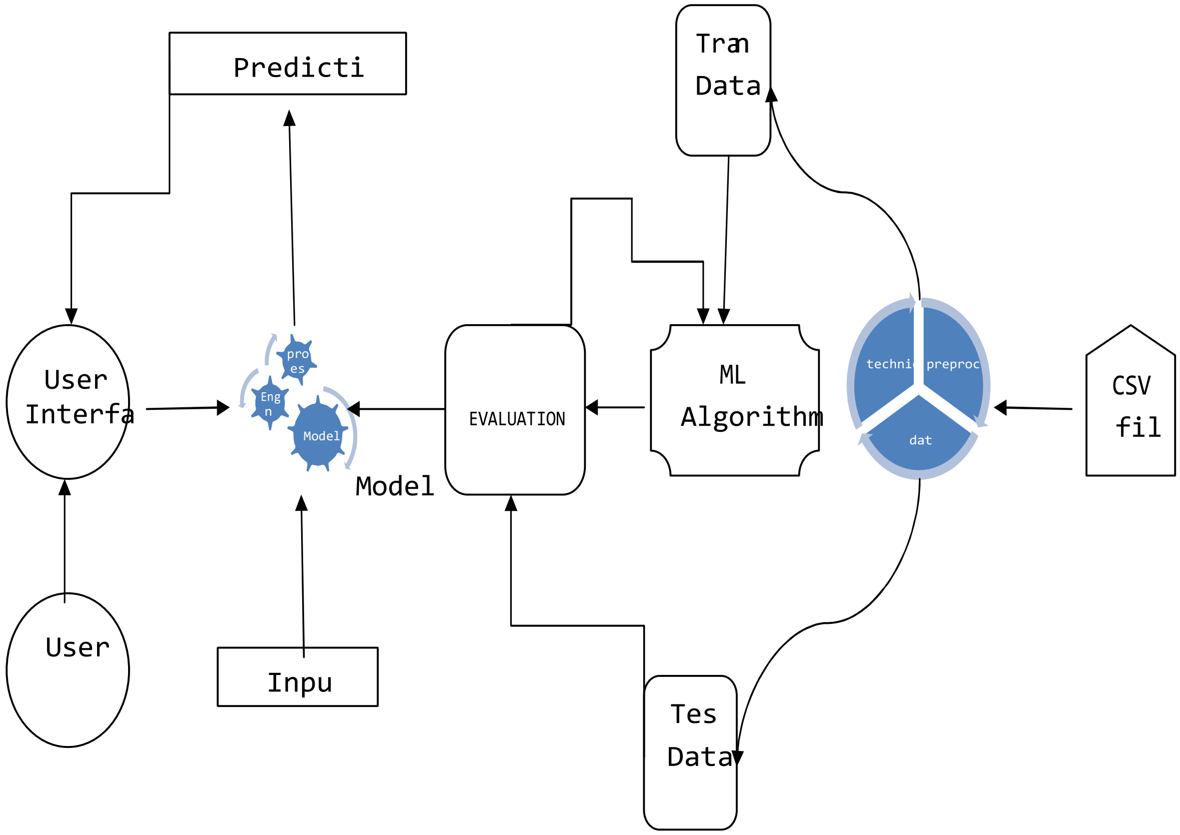
4. PROJECT DESIGN

4.1. Data Flow Diagrams


START
LOGIN
Sign in(new user)
Log in
Username
Password
Sign in
Username
Password
Log in
Vision
CROP YEILD PREDICTION
DISEASE DETECTION
Crop_details
Get_location(API)
Weather(API)
ML model
Predicts crop yeild
Input image
Send notice
filtering
enhancement
segmentation
Feature extraction
classifier
report
Disease prediction & suitable solutions

4.2. Solution and Technical Architecture

SOLUTION ARCHITECTURE

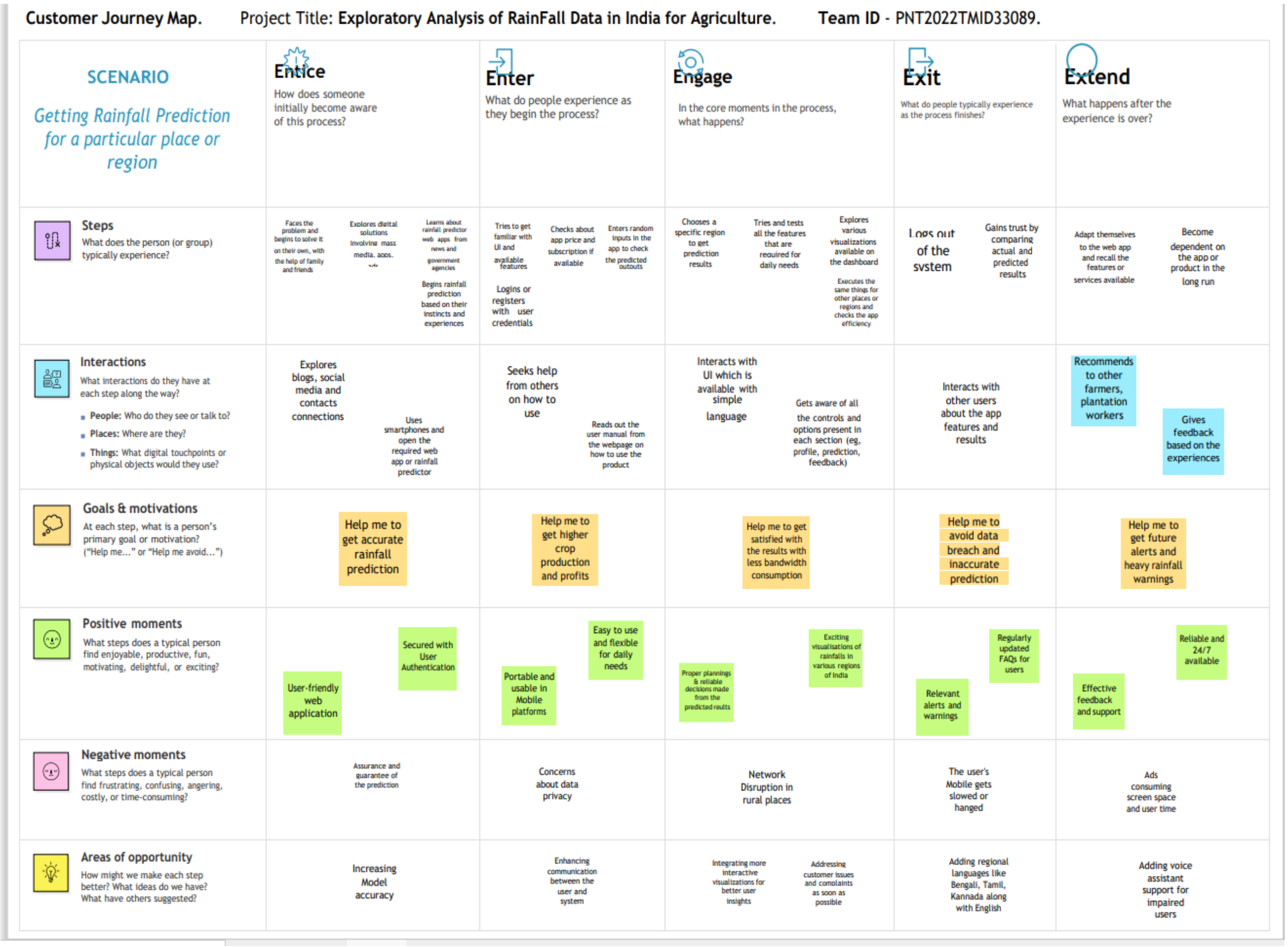


TECHNICAL ARCHITECTURE

S.No	Component	Description	Technology
1.	Website	User interacts with the prediction model through website to predict the rainfall data	HTML, CSS, JavaScript

2.	Cloud Database	The model is provided with data from IBM cloud database	IBM Cloud DB, ibm_db(python package)
3.	API	Used to extend the service to other applications	Flask Application
4.	JWT & Sessions	It is used for Handling JSON web tokens (signing, verifying, decoding)	PyJWT, Flask-Sessions
5.	Machine Learni ng Model	This model is developed to predict the rainfall using ML algorithms.	Sklearn, Algorithms - DT & MLR
6.	Data processing	Data is pre-processed and then used for prediction.	Pandas, Numpy, Matplotlib
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem

4.3. User Stories



Customer Journey Map. Project Title: **Exploratory Analysis of RainFall Data in India for Agriculture.** **Team ID** - PNT2022TMID33089.

SCENARIO *Getting Rainfall Prediction for a particular place or region*	**Entice** How does someone initially become aware of this process?	**Enter** What do people experience as they begin the process?	**Engage** In the core moments in the process, what happens?	**Exit** What do people typically experience as the process finishes?	**Extend** What happens after the experience is over?
Steps What does the person (or group) typically experience?	Faces the problem and begins to solve it on their own, with the help of family and friends Explores digital solutions involving mass media, apps, ads Learns about rainfall predictor web apps from news and government agencies Begins rainfall prediction based on their instincts and experiences	Tries to get familiar with UI and available features Checks about app price and subscription if available Enters random inputs in the app to check the predicted outputs Logins or registers with user credentials	Chooses a specific region to get prediction results Tries and tests all the features that are required for daily needs Explores various visualizations available on the dashboard Executes the same things for other places or regions and checks the app efficiency	Logs out of the system Gains trust by comparing actual and predicted results	Adapt themselves to the web app and recall the features or services available Become dependent on the app or product in the long run
Interactions What interactions do they have at each step along the way? - **People:** Who do they see or talk to? - **Places:** Where are they? - **Things:** What digital touchpoints or physical objects would they use?	Explores blogs, social media and contacts connections Uses smartphones and open the required web app or rainfall predictor	Seeks help from others on how to use Reads out the user manual from the webpage on how to use the product	Interacts with UI which is available with simple language Gets aware of all the controls and options present in each section (eg, profile, prediction, feedback)	Interacts with other users about the app features and results	Recommends to other farmers, plantation workers Gives feedback based on the experiences
Goals & motivations At each step, what is a person's primary goal or motivation? ("Help me..." or "Help me avoid...")	Help me to get accurate rainfall prediction	Help me to get higher crop production and profits	Help me to get satisfied with the results with less bandwidth consumption	Help me to avoid data breach and inaccurate prediction	Help me to get future alerts and heavy rainfall warnings
Positive moments What steps does a typical person find enjoyable, productive, fun, motivating, delightful, or exciting?	Secured with User Authentication User-friendly web application	Easy to use and flexible for daily needs Portable and usable in Mobile platforms	Exciting visualisations of rainfalls in various regions of India Proper plannings & reliable decisions made from the predicted reults	Regularly updated FAQs for users Relevant alerts and warnings	Reliable and 24/7 available Effective feedback and support
Negative moments What steps does a typical person find frustrating, confusing, angering, costly, or time-consuming?	Assurance and guarantee of the prediction	Concerns about data privacy	Network Disruption in rural places	The user's Mobile gets slowed or hanged	Ads consuming screen space and user time
Areas of opportunity How might we make each step better? What ideas do we have? What have others suggested?	Increasing Model accuracy	Enhancing communication between the user and system	Integrating more interactive visualizations for better user insights Addressing customer issues and complaints as soon as possible	Adding regional languages like Bengali, Tamil, Kannada along with English	Adding voice assistant support for impaired users

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Rainfall PredictionML Model (Dataset)	USN-1	Weather Dataset Collection, Data preprocessing, Data Visualization.	5	High	J.Murugavasan , B.Rohith
Sprint-1		USN-2	Train Model using Different machine learning Algorithms	5	High	S.Sakthivel , M.Suresh
Sprint-1		USN-3	Test the model and give best	10	High	J.Murugavasan , R.Mohamed Yousuf
Sprint-2	Registration	USN-4	As a user, they can register for the application through Gmail. Password is set up.	5	Medium	S.Sakthivel , R.Mohamed Yousuf
Sprint-2	Login	USN-5	As a user, they can log into the application by entering email & password	5	Medium	B.Rohith , M.Suresh
Sprint-2		USN-6	Credentials should be used for multiple systems and verified	4	Medium	S.Sakthivel , J.Murugavasan
Sprint-2	Dashboard	USN-7	Attractive dashboard forecasting live weather	6	Low	R.Mohamed Yousuf , B.Rohith
Sprint-3	Rainfall Prediction	USN-8	User enter the location, temperature,	10	High	M.Suresh , R.Mohamed Yousuf

			humidity			
Sprint-3		USN-9	Predict the rainfall and display the result	10	High	J.Murugavasan , B.Rohith

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)**	**Sprint Release Date(Actual)**
Sprint-1	20	6 Days	31Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-2	20	6 Days	05 Nov 2022	10 Nov 2022	20	10 Nov 2022
Sprint-3	20	6 Days	10 Nov 2022	15 Nov 2022	20	15 Nov 2022
Sprint-4	20	6 Days	15 Nov 2022	21 Nov 2022	20	21 Nov 2022

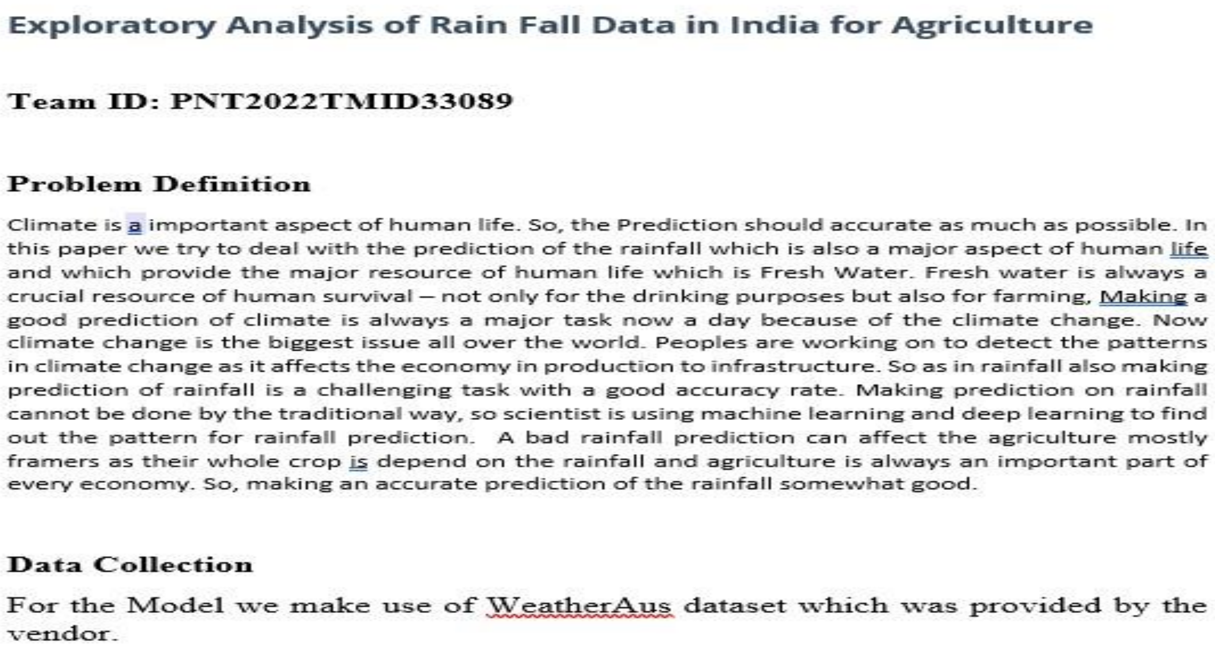
7.CODING AND SOLUTIONING

7.1Feature-1: Model Building

For this feature we have made use of Jupyter notebook which uses Python programming language. To use Jupyter Notebook install Anaconda, which is a desktop graphical user interface (GUI)

included in Anaconda® Distribution that allows you to launch applications and manage conda packages, environments, and channels without using command line interface (CLI) commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository. It is available for Windows, macOS, and Linux. It provides all basic necessary python libraries which are needed for Data Analysis and Visualizations.

Below images are source code for this feature:



Exploratory Analysis of Rain Fall Data in India for Agriculture

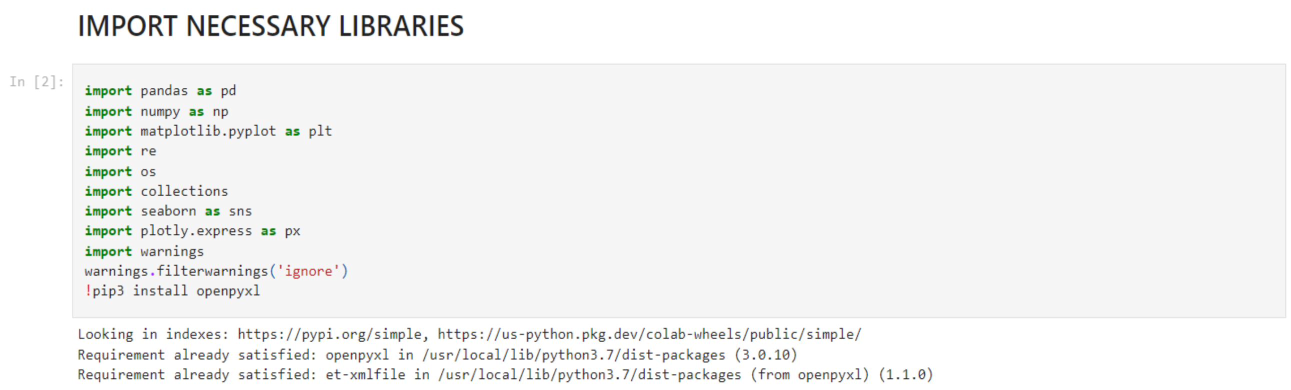
Team ID: PNT2022TMID33089

Problem Definition

Climate is a important aspect of human life. So, the Prediction should accurate as much as possible. In this paper we try to deal with the prediction of the rainfall which is also a major aspect of human life and which provide the major resource of human life which is Fresh Water. Fresh water is always a crucial resource of human survival – not only for the drinking purposes but also for farming, Making a good prediction of climate is always a major task now a day because of the climate change. Now climate change is the biggest issue all over the world. Peoples are working on to detect the patterns in climate change as it affects the economy in production to infrastructure. So as in rainfall also making prediction of rainfall is a challenging task with a good accuracy rate. Making prediction on rainfall cannot be done by the traditional way, so scientist is using machine learning and deep learning to find out the pattern for rainfall prediction. A bad rainfall prediction can affect the agriculture mostly framers as their whole crop is depend on the rainfall and agriculture is always an important part of every economy. So, making an accurate prediction of the rainfall somewhat good.

Data Collection

For the Model we make use of WeatherAus dataset which was provided by the vendor.

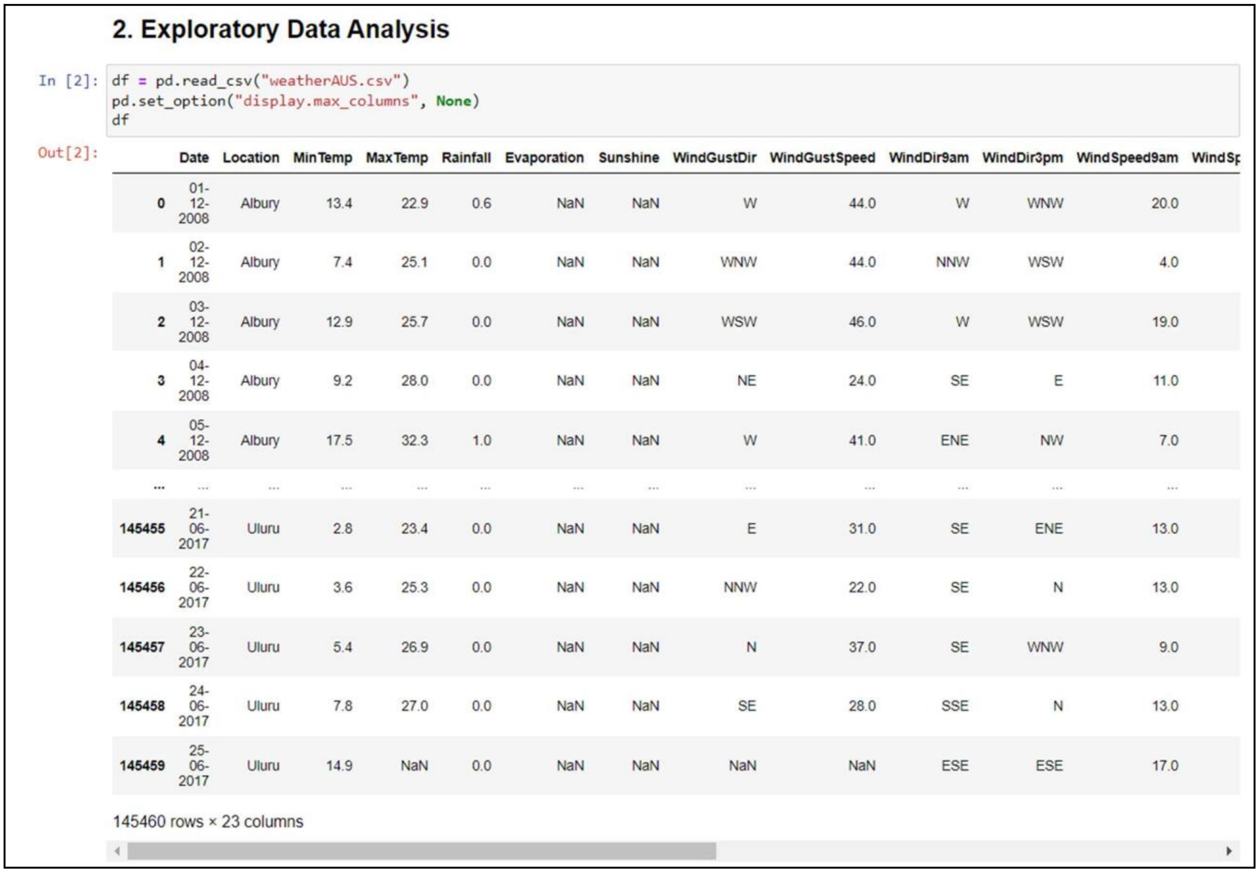


IMPORT NECESSARY LIBRARIES

```
In [2]:
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import re
import os
import collections
import seaborn as sns
import plotly.express as px
import warnings
warnings.filterwarnings('ignore')
!pip3 install openpyxl
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: openpyxl in /usr/local/lib/python3.7/dist-packages (3.0.10)
Requirement already satisfied: et-xmlfile in /usr/local/lib/python3.7/dist-packages (from openpyxl) (1.1.0)
```

In the above image, we import all necessary libraries needed for data exploration, preprocessing, model building and saving it. The below image specifies the values present in the dataset.



2. Exploratory Data Analysis

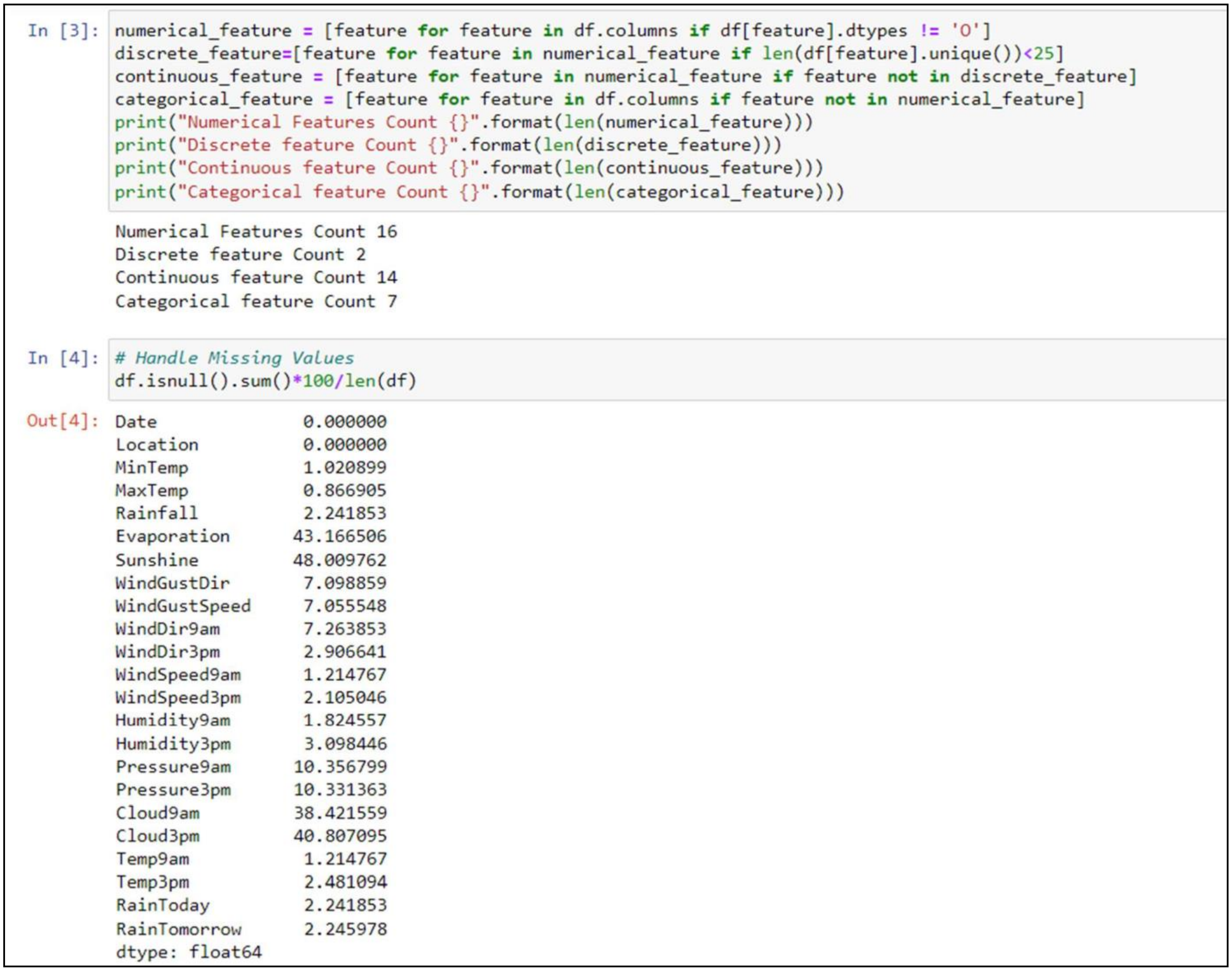
```
In [2]: df = pd.read_csv("weatherAUS.csv")
        pd.set_option("display.max_columns", None)
        df
```

Out[2]:

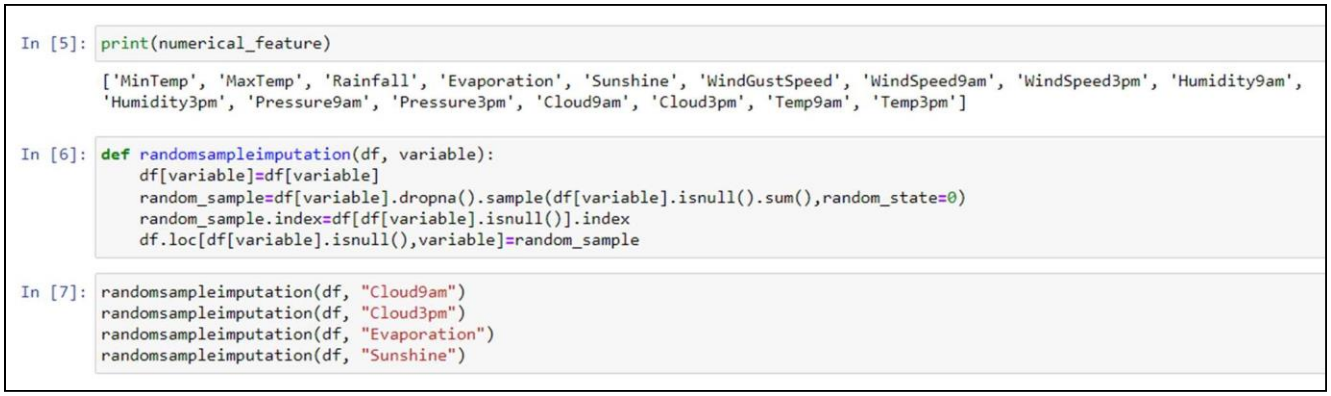
	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	WindSpeed9am	WindSp
0	01-12-2008	Albury	13.4	22.9	0.6	NaN	NaN	W	44.0	W	WNW	20.0	
1	02-12-2008	Albury	7.4	25.1	0.0	NaN	NaN	WNW	44.0	NNW	WSW	4.0	
2	03-12-2008	Albury	12.9	25.7	0.0	NaN	NaN	WSW	46.0	W	WSW	19.0	
3	04-12-2008	Albury	9.2	28.0	0.0	NaN	NaN	NE	24.0	SE	E	11.0	
4	05-12-2008	Albury	17.5	32.3	1.0	NaN	NaN	W	41.0	ENE	NW	7.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
145455	21-06-2017	Uluru	2.8	23.4	0.0	NaN	NaN	E	31.0	SE	ENE	13.0	
145456	22-06-2017	Uluru	3.6	25.3	0.0	NaN	NaN	NNW	22.0	SE	N	13.0	
145457	23-06-2017	Uluru	5.4	26.9	0.0	NaN	NaN	N	37.0	SE	WNW	9.0	
145458	24-06-2017	Uluru	7.8	27.0	0.0	NaN	NaN	SE	28.0	SSE	N	13.0	
145459	25-06-2017	Uluru	14.9	NaN	0.0	NaN	NaN	NaN	NaN	ESE	ESE	17.0	

145460 rows × 23 columns

The below image specifies types of features and its count along with number of missing values in the dataset.

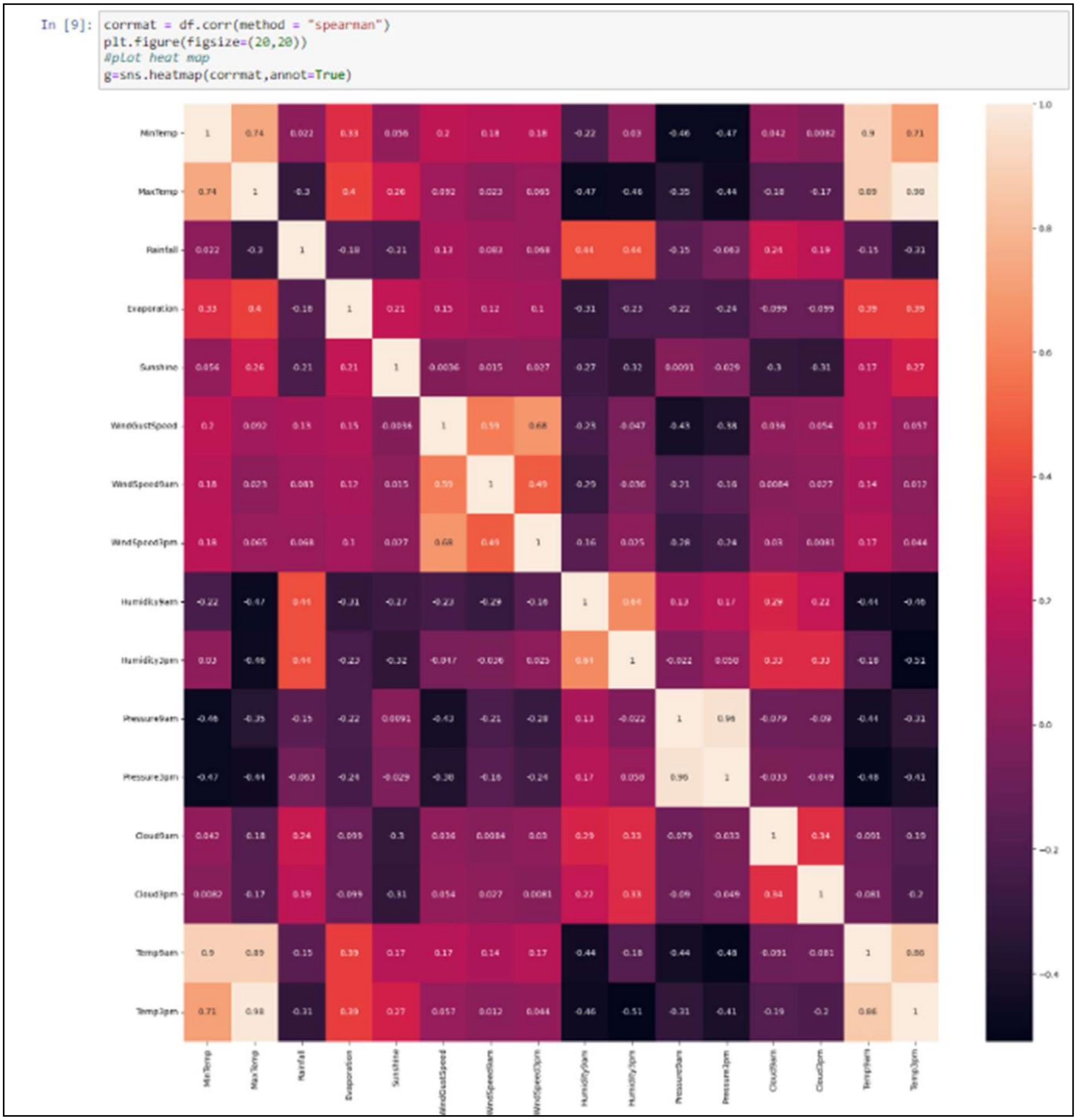


```
In [3]: numerical_feature = [feature for feature in df.columns if df[feature].dtypes != 'O']
        discrete_feature=[feature for feature in numerical_feature if len(df[feature].unique())<25]
        continuous_feature = [feature for feature in numerical_feature if feature not in discrete_feature]
        categorical_feature = [feature for feature in df.columns if feature not in numerical_feature]
        print("Numerical Features Count {}".format(len(numerical_feature)))
        print("Discrete feature Count {}".format(len(discrete_feature)))
        print("Continuous feature Count {}".format(len(continuous_feature)))
        print("Categorical feature Count {}".format(len(categorical_feature)))

        Numerical Features Count 16
        Discrete feature Count 2
        Continuous feature Count 14
        Categorical feature Count 7

In [4]: # Handle Missing Values
        df.isnull().sum()*100/len(df)

Out[4]: Date              0.000000
        Location          0.000000
        MinTemp           1.020899
        MaxTemp           0.866905
        Rainfall          2.241853
        Evaporation      43.166506
        Sunshine         48.009762
        WindGustDir       7.098859
        WindGustSpeed     7.055548
        WindDir9am        7.263853
        WindDir3pm        2.906641
        WindSpeed9am      1.214767
        WindSpeed3pm      2.105046
        Humidity9am       1.824557
        Humidity3pm       3.098446
        Pressure9am      10.356799
        Pressure3pm      10.331363
        Cloud9am         38.421559
        Cloud3pm         40.807095
        Temp9am           1.214767
        Temp3pm           2.481094
        RainToday         2.241853
        RainTomorrow      2.245978
        dtype: float64
```



```
In [5]: print(numerical_feature)

        ['MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation', 'Sunshine', 'WindGustSpeed', 'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am',
        'Humidity3pm', 'Pressure9am', 'Pressure3pm', 'Cloud9am', 'Cloud3pm', 'Temp9am', 'Temp3pm']

In [6]: def randomsampleimputation(df, variable):
            df[variable]=df[variable]
            random_sample=df[variable].dropna().sample(df[variable].isnull().sum(),random_state=0)
            random_sample.index=df[df[variable].isnull()].index
            df.loc[df[variable].isnull(),variable]=random_sample

In [7]: randomsampleimputation(df, "Cloud9am")
        randomsampleimputation(df, "Cloud3pm")
        randomsampleimputation(df, "Evaporation")
        randomsampleimputation(df, "Sunshine")
```

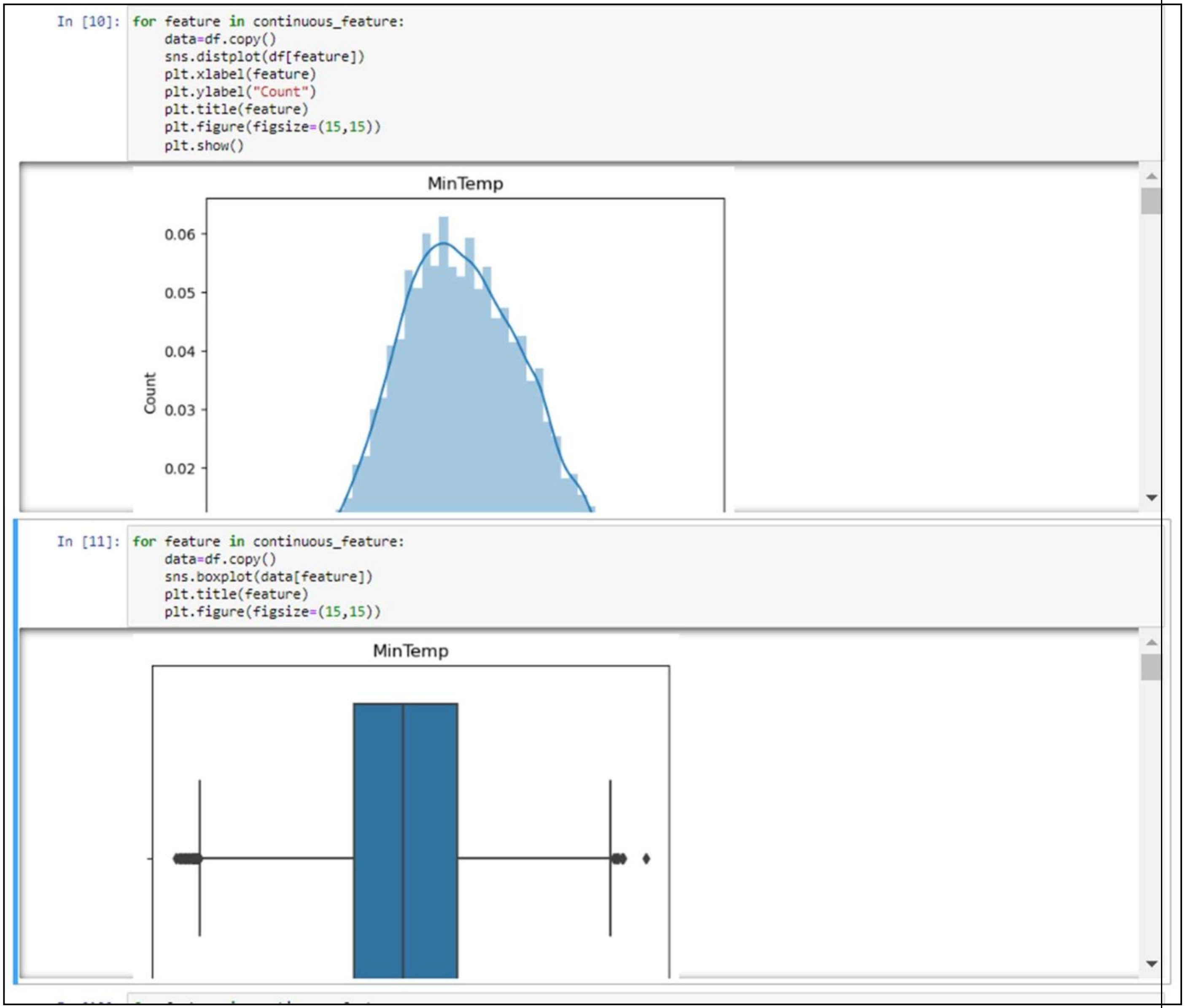
The lines 6 is used to drop rows which have high count missing values.



```
In [9]: corrmat = df.corr(method = "spearman")
        plt.figure(figsize=(20,20))
        #plot heat map
        g=sns.heatmap(corrmat,annot=True)
```

MinTemp 1 0.74 0.022 0.33 0.056 0.2 0.18 0.18 -0.22 0.03 -0.46 -0.47 0.042 0.0082 0.9 0.71
MaxTemp 0.74 1 -0.3 0.4 0.26 0.092 0.023 0.065 -0.47 -0.46 -0.35 -0.44 -0.18 -0.17 0.89 0.98
Rainfall 0.022 -0.3 1 -0.18 -0.21 0.13 0.083 0.068 0.44 0.44 -0.15 -0.063 0.24 0.19 -0.15 -0.31
Evaporation 0.33 0.4 -0.18 1 0.21 0.15 0.12 0.1 -0.31 -0.23 -0.22 -0.24 -0.099 -0.099 0.39 0.39
Sunshine 0.056 0.26 -0.21 0.21 1 -0.0036 0.015 0.027 -0.27 -0.32 0.0091 -0.029 -0.3 -0.31 0.17 0.27
WindGustSpeed 0.2 0.092 0.13 0.15 -0.0036 1 0.59 0.68 -0.23 -0.047 -0.43 -0.38 0.036 0.054 0.17 0.057
WindSpeed9am 0.18 0.023 0.083 0.12 0.015 0.59 1 0.49 -0.29 -0.036 -0.21 -0.16 0.0084 0.027 0.14 0.012
WindSpeed3pm 0.18 0.065 0.068 0.1 0.027 0.68 0.49 1 -0.16 0.025 -0.28 -0.24 0.03 0.0081 0.17 0.044
Humidity9am -0.22 -0.47 0.44 -0.31 -0.27 -0.23 -0.29 -0.16 1 0.64 0.13 0.17 0.29 0.22 -0.44 -0.46
Humidity3pm 0.03 -0.46 0.44 -0.23 -0.32 -0.047 -0.036 0.025 0.64 1 -0.022 0.058 0.33 0.33 -0.18 -0.51
Pressure9am -0.46 -0.35 -0.15 -0.22 0.0091 -0.43 -0.21 -0.28 0.13 -0.022 1 0.96 -0.079 -0.09 -0.44 -0.31
Pressure3pm -0.47 -0.44 -0.063 -0.24 -0.029 -0.38 -0.16 -0.24 0.17 0.058 0.96 1 -0.033 -0.049 -0.48 -0.41
Cloud9am 0.042 -0.18 0.24 -0.099 -0.3 0.036 0.0084 0.03 0.29 0.33 -0.079 -0.033 1 0.34 -0.091 -0.19
Cloud3pm 0.0082 -0.17 0.19 -0.099 -0.31 0.054 0.027 0.0081 0.22 0.33 -0.09 -0.049 0.34 1 -0.081 -0.2
Temp9am 0.9 0.89 -0.15 0.39 0.17 0.17 0.14 0.17 -0.44 -0.18 -0.44 -0.48 -0.091 -0.081 1 0.86
Temp3pm 0.71 0.98 -0.31 0.39 0.27 0.057 0.012 0.044 -0.46 -0.51 -0.31 -0.41 -0.19 -0.2 0.86 1
MinTemp
MaxTemp
Rainfall
Evaporation
Sunshine
WindGustSpeed
WindSpeed9am
WindSpeed3pm
Humidity9am
Humidity3pm
Pressure9am
Pressure3pm
Cloud9am
Cloud3pm
Temp9am
Temp3pm
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

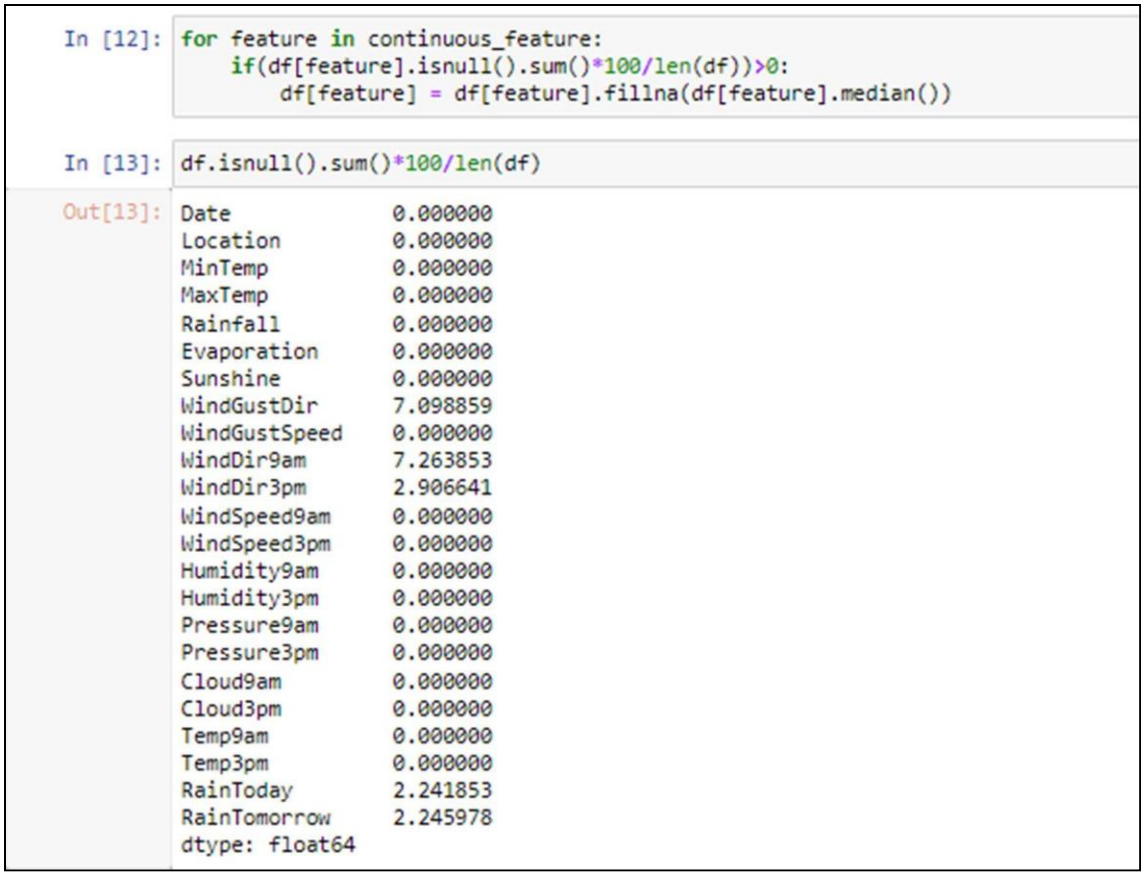
The above code displays the correlation between the columns present in the dataset.



```
In [10]: for feature in continuous_feature:
             data=df.copy()
             sns.distplot(df[feature])
             plt.xlabel(feature)
             plt.ylabel("Count")
             plt.title(feature)
             plt.figure(figsize=(15,15))
             plt.show()
```

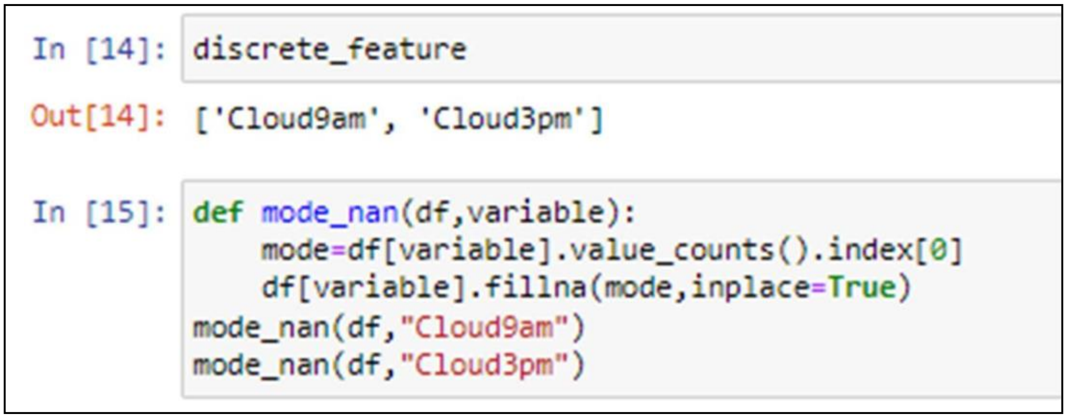
```
In [11]: for feature in continuous_feature:
             data=df.copy()
             sns.boxplot(data[feature])
             plt.title(feature)
             plt.figure(figsize=(15,15))
```

The above code shows the distance plot and box plot of continuous features.



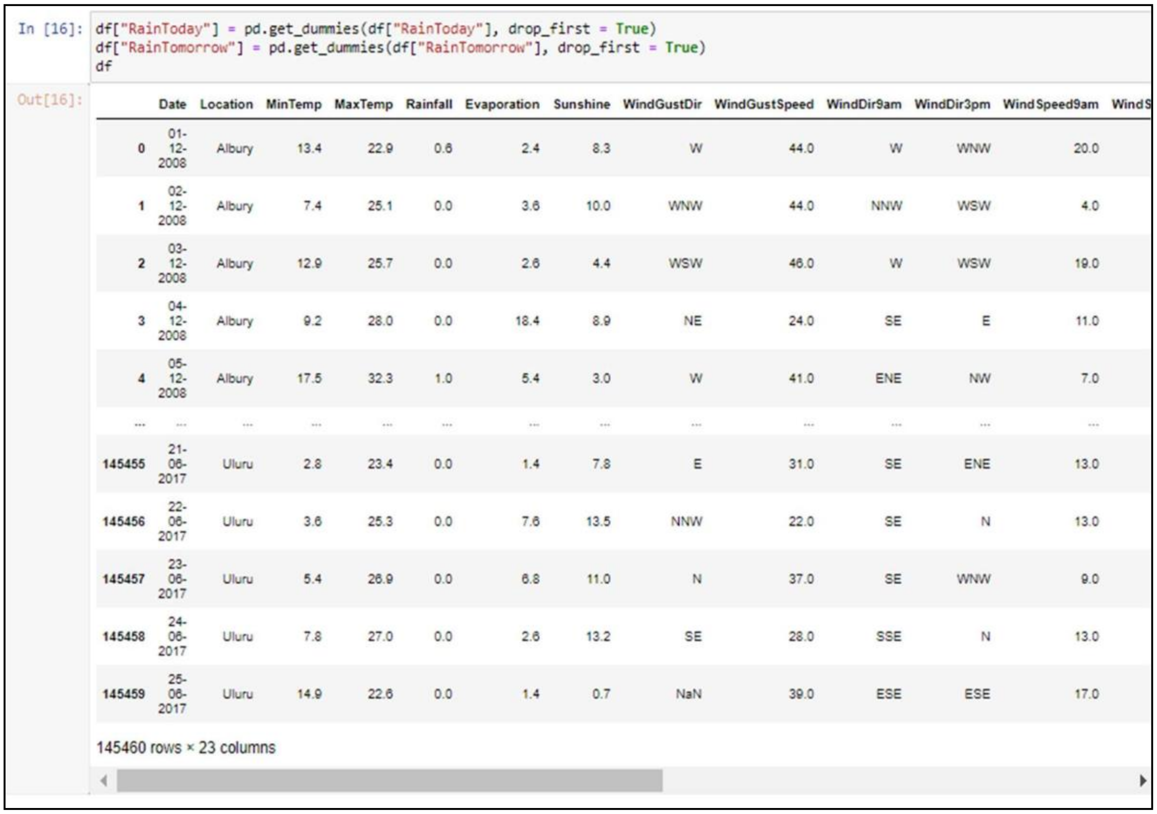
```
In [12]: for feature in continuous_feature:
             if(df[feature].isnull().sum()*100/len(df))>0:
                 df[feature] = df[feature].fillna(df[feature].median())

In [13]: df.isnull().sum()*100/len(df)

Out[13]: Date             0.000000
         Location         0.000000
         MinTemp          0.000000
         MaxTemp          0.000000
         Rainfall         0.000000
         Evaporation      0.000000
         Sunshine         0.000000
         WindGustDir      7.098859
         WindGustSpeed    0.000000
         WindDir9am       7.263853
         WindDir3pm       2.906641
         WindSpeed9am     0.000000
         WindSpeed3pm     0.000000
         Humidity9am      0.000000
         Humidity3pm      0.000000
         Pressure9am      0.000000
         Pressure3pm      0.000000
         Cloud9am         0.000000
         Cloud3pm         0.000000
         Temp9am          0.000000
         Temp3pm          0.000000
         RainToday        2.241853
         RainTomorrow     2.245978
         dtype: float64
```

The above code removes null values from continuous features.



```
In [14]: discrete_feature

Out[14]: ['Cloud9am', 'Cloud3pm']

In [15]: def mode_nan(df,variable):
             mode=df[variable].value_counts().index[0]
             df[variable].fillna(mode,inplace=True)
         mode_nan(df,"Cloud9am")
         mode_nan(df,"Cloud3pm")
```

The above code removes null values by replacing it with Mode value.



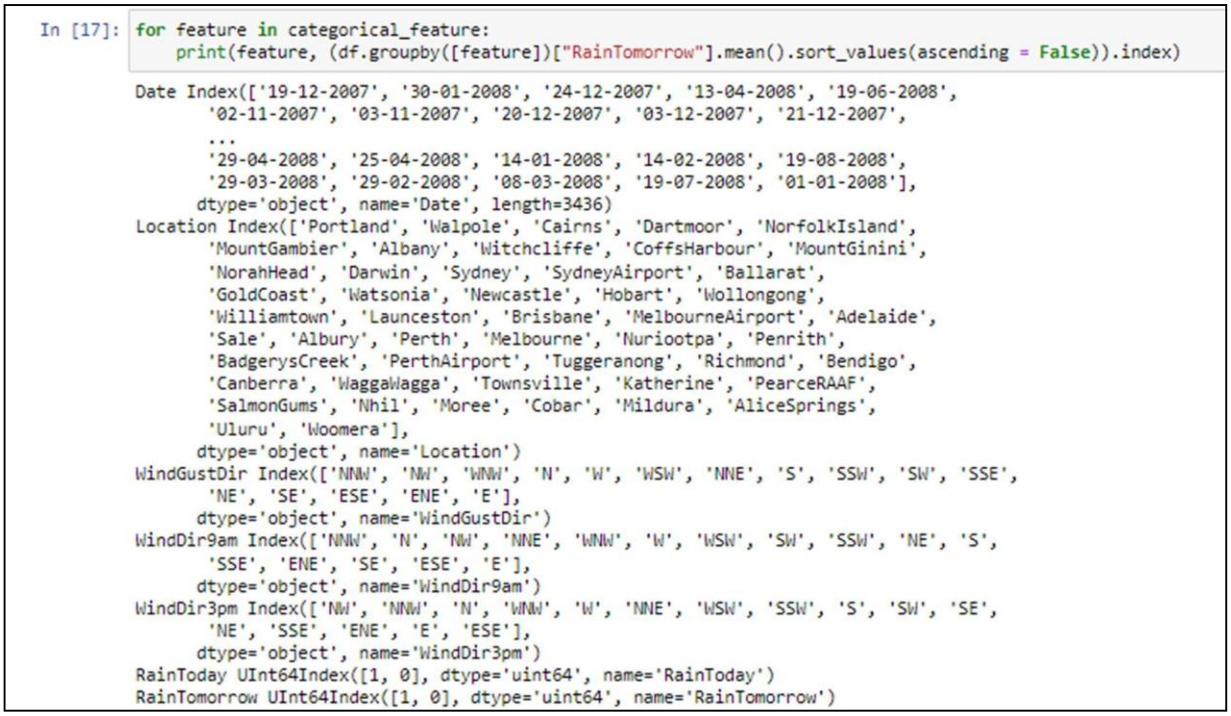
```
In [16]: df["RainToday"] = pd.get_dummies(df["RainToday"], drop_first = True)
         df["RainTomorrow"] = pd.get_dummies(df["RainTomorrow"], drop_first = True)
         df
```

Out[16]:

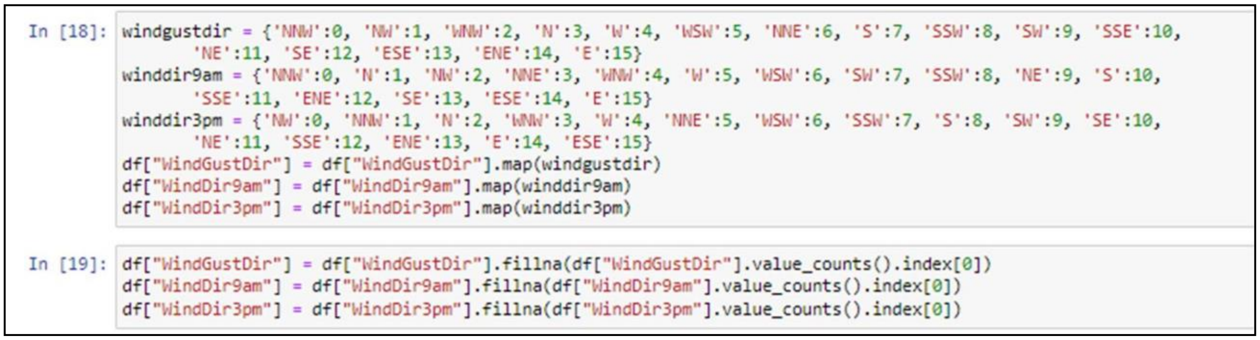
	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	WindSpeed9am	WindS
0	01-12-2008	Albury	13.4	22.9	0.6	2.4	8.3	W	44.0	W	WNW	20.0	
1	02-12-2008	Albury	7.4	25.1	0.0	3.6	10.0	WNW	44.0	NNW	WSW	4.0	
2	03-12-2008	Albury	12.9	25.7	0.0	2.6	4.4	WSW	46.0	W	WSW	19.0	
3	04-12-2008	Albury	9.2	28.0	0.0	18.4	8.9	NE	24.0	SE	E	11.0	
4	05-12-2008	Albury	17.5	32.3	1.0	5.4	3.0	W	41.0	ENE	NW	7.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
145455	21-06-2017	Uluru	2.8	23.4	0.0	1.4	7.8	E	31.0	SE	ENE	13.0	
145456	22-06-2017	Uluru	3.6	25.3	0.0	7.6	13.5	NNW	22.0	SE	N	13.0	
145457	23-06-2017	Uluru	5.4	26.9	0.0	6.8	11.0	N	37.0	SE	WNW	9.0	
145458	24-06-2017	Uluru	7.8	27.0	0.0	2.6	13.2	SE	28.0	SSE	N	13.0	
145459	25-06-2017	Uluru	14.9	22.6	0.0	1.4	0.7	NaN	39.0	ESE	ESE	17.0	

145460 rows × 23 columns

The above code makes use of Label Encoding technique, which is used to convert labels into machinereadable numeric values.

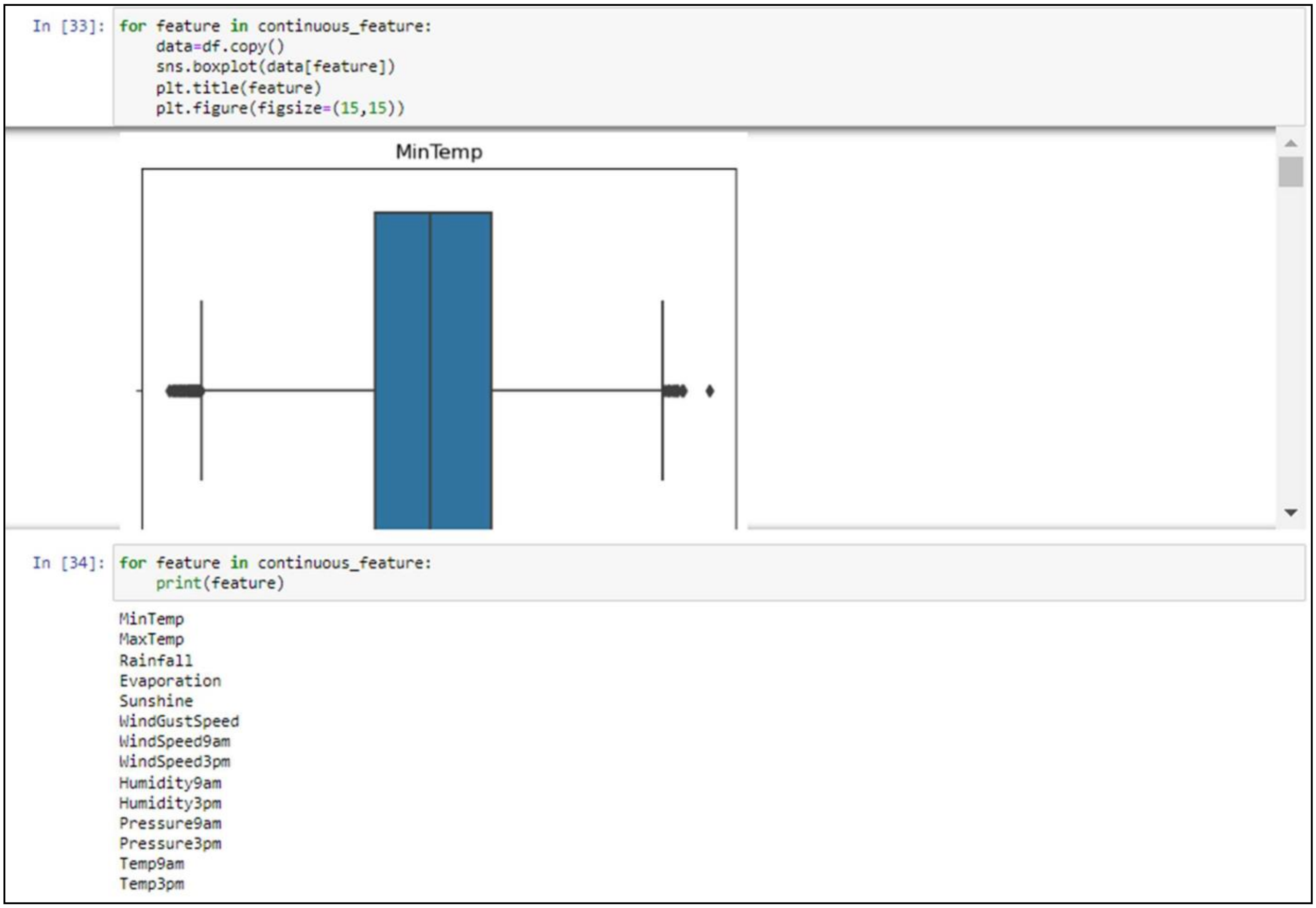


```
In [17]: for feature in categorical_feature:
             print(feature, (df.groupby([feature])["RainTomorrow"].mean().sort_values(ascending = False)).index)

         Date Index(['19-12-2007', '30-01-2008', '24-12-2007', '13-04-2008', '19-06-2008',
                '02-11-2007', '03-11-2007', '20-12-2007', '03-12-2007', '21-12-2007',
                ...
                '29-04-2008', '25-04-2008', '14-01-2008', '14-02-2008', '19-08-2008',
                '29-03-2008', '29-02-2008', '08-03-2008', '19-07-2008', '01-01-2008'],
               dtype='object', name='Date', length=3436)
         Location Index(['Portland', 'Walpole', 'Cairns', 'Dartmoor', 'NorfolkIsland',
                'MountGambier', 'Albany', 'Witchcliffe', 'CoffsHarbour', 'MountGinini',
                'NorahHead', 'Darwin', 'Sydney', 'SydneyAirport', 'Ballarat',
                'GoldCoast', 'Watsonia', 'Newcastle', 'Hobart', 'Wollongong',
                'Williamtown', 'Launceston', 'Brisbane', 'MelbourneAirport', 'Adelaide',
                'Sale', 'Albury', 'Perth', 'Melbourne', 'Nuriootpa', 'Penrith',
                'BadgerysCreek', 'PerthAirport', 'Tuggeranong', 'Richmond', 'Bendigo',
                'Canberra', 'WaggaWagga', 'Townsville', 'Katherine', 'PearceRAAF',
                'SalmonGums', 'Nhil', 'Moree', 'Cobar', 'Mildura', 'AliceSprings',
                'Uluru', 'Woomera'],
               dtype='object', name='Location')
         WindGustDir Index(['NNW', 'NW', 'WNW', 'N', 'W', 'WSW', 'NNE', 'S', 'SSW', 'SW', 'SSE',
                'NE', 'SE', 'ESE', 'ENE', 'E'],
               dtype='object', name='WindGustDir')
         WindDir9am Index(['NNW', 'N', 'NW', 'NNE', 'WNW', 'W', 'WSW', 'SW', 'SSW', 'NE', 'S',
                'SSE', 'ENE', 'SE', 'ESE', 'E'],
               dtype='object', name='WindDir9am')
         WindDir3pm Index(['NW', 'NNW', 'N', 'WNW', 'W', 'NNE', 'WSW', 'SSW', 'S', 'SW', 'SE',
                'NE', 'SSE', 'ENE', 'E', 'ESE'],
               dtype='object', name='WindDir3pm')
         RainToday UInt64Index([1, 0], dtype='uint64', name='RainToday')
         RainTomorrow UInt64Index([1, 0], dtype='uint64', name='RainTomorrow')
```



```
In [18]: windgustdir = {'NNW':0, 'NW':1, 'WNW':2, 'N':3, 'W':4, 'WSW':5, 'NNE':6, 'S':7, 'SSW':8, 'SW':9, 'SSE':10,
                'NE':11, 'SE':12, 'ESE':13, 'ENE':14, 'E':15}
         winddir9am = {'NNW':0, 'N':1, 'NW':2, 'NNE':3, 'WNW':4, 'W':5, 'WSW':6, 'SW':7, 'SSW':8, 'NE':9, 'S':10,
                'SSE':11, 'ENE':12, 'SE':13, 'ESE':14, 'E':15}
         winddir3pm = {'NW':0, 'NNW':1, 'N':2, 'WNW':3, 'W':4, 'NNE':5, 'WSW':6, 'SSW':7, 'S':8, 'SW':9, 'SE':10,
                'NE':11, 'SSE':12, 'ENE':13, 'E':14, 'ESE':15}
         df["WindGustDir"] = df["WindGustDir"].map(windgustdir)
         df["WindDir9am"] = df["WindDir9am"].map(winddir9am)
         df["WindDir3pm"] = df["WindDir3pm"].map(winddir3pm)

In [19]: df["WindGustDir"] = df["WindGustDir"].fillna(df["WindGustDir"].value_counts().index[0])
         df["WindDir9am"] = df["WindDir9am"].fillna(df["WindDir9am"].value_counts().index[0])
         df["WindDir3pm"] = df["WindDir3pm"].fillna(df["WindDir3pm"].value_counts().index[0])
```

The above image is used to remove the remaining null values.

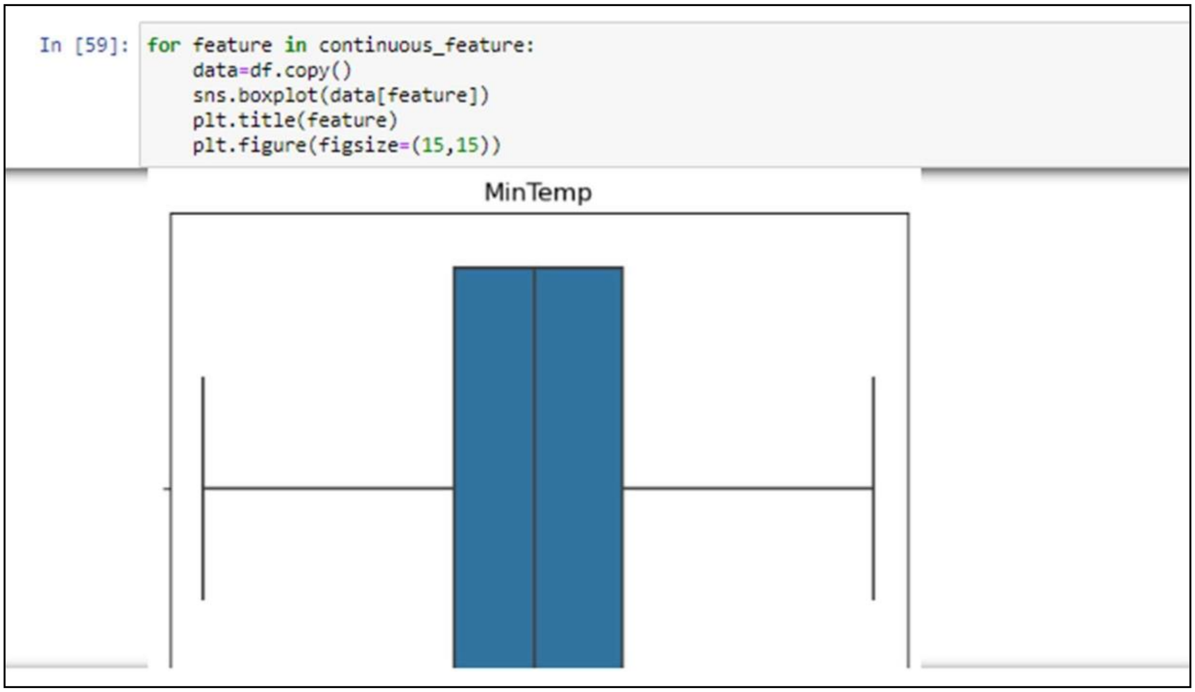


```
In [33]: for feature in continuous_feature:
             data=df.copy()
             sns.boxplot(data[feature])
             plt.title(feature)
             plt.figure(figsize=(15,15))
```

```
In [34]: for feature in continuous_feature:
             print(feature)
```

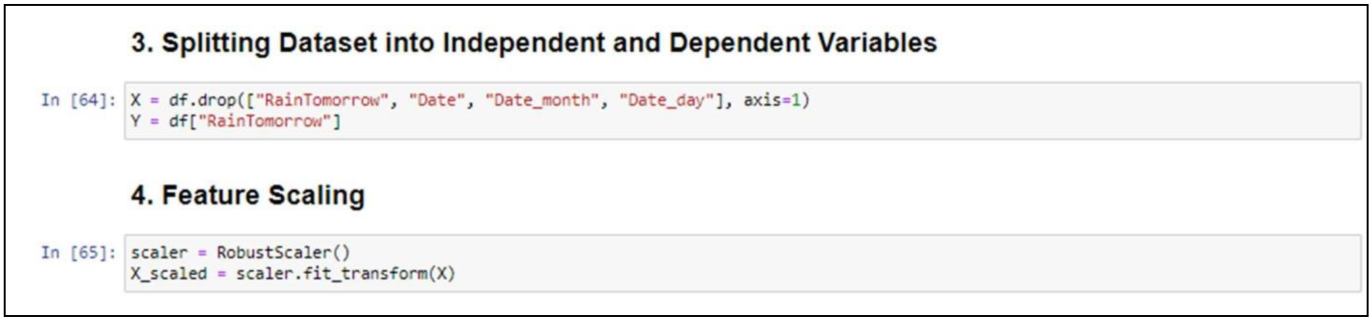
```
MinTemp
MaxTemp
Rainfall
Evaporation
Sunshine
WindGustSpeed
WindSpeed9am
WindSpeed3pm
Humidity9am
Humidity3pm
Pressure9am
Pressure3pm
Temp9am
Temp3pm
```

The above image is used to find values which lies outside the Inter-Quartile Range of each continuous feature. After finding the lower and higher bound, we remove the outliers from each continuous feature.



```
In [59]: for feature in continuous_feature:
             data=df.copy()
             sns.boxplot(data[feature])
             plt.title(feature)
             plt.figure(figsize=(15,15))
```

The above image shows the boxplot of each continuous feature after removing the outliers.



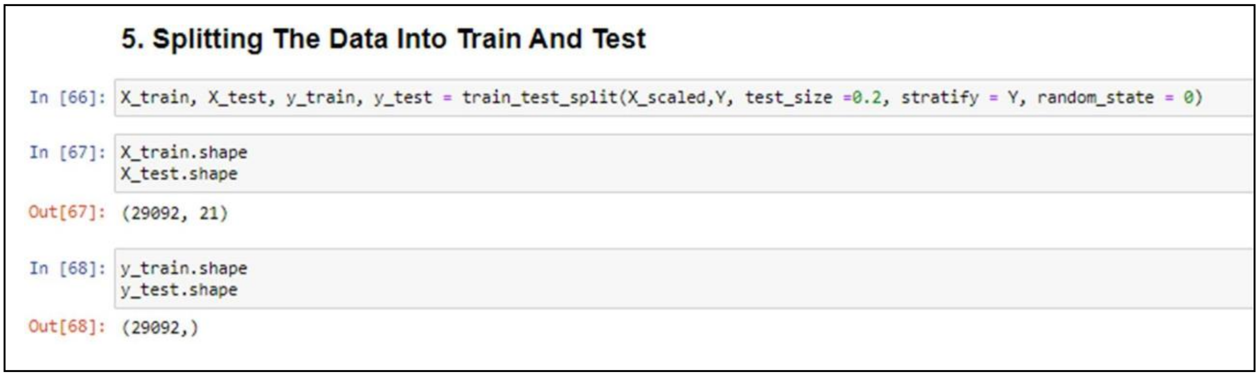
3. Splitting Dataset into Independent and Dependent Variables

```
In [64]: X = df.drop(["RainTomorrow", "Date", "Date_month", "Date_day"], axis=1)
         Y = df["RainTomorrow"]
```

4. Feature Scaling

```
In [65]: scaler = RobustScaler()
         X_scaled = scaler.fit_transform(X)
```

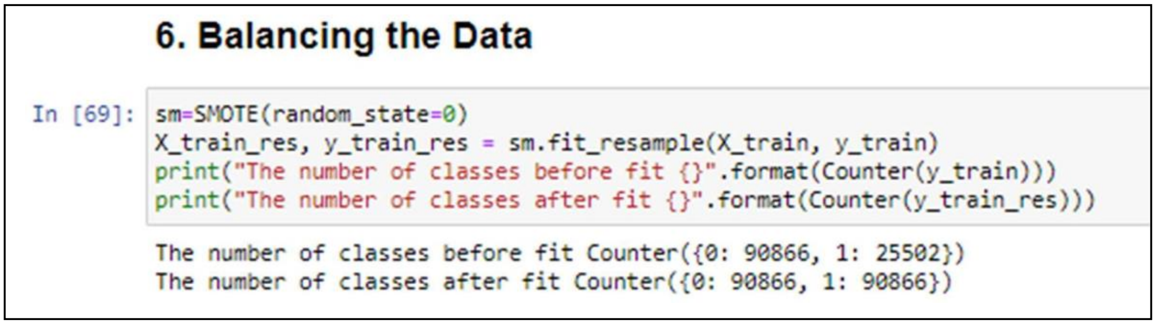
We split the dataset into independent and dependent variables. Here we must predict ‘RainTomorrow’, hence it will be the dependent variable and Date columns are unnecessary columns hence we drop it. And all other columns are independent variables. Using RobustScaler, we perform feature scaling to normalize the independent variables such that the standard distribution results to zero and standard deviation to one. This also removes remaining outliers in the independent variables.



5. Splitting The Data Into Train And Test

```
In [66]: X_train, X_test, y_train, y_test = train_test_split(X_scaled,Y, test_size =0.2, stratify = Y, random_state = 0)

In [67]: X_train.shape
         X_test.shape

Out[67]: (29092, 21)

In [68]: y_train.shape
         y_test.shape

Out[68]: (29092,)
```

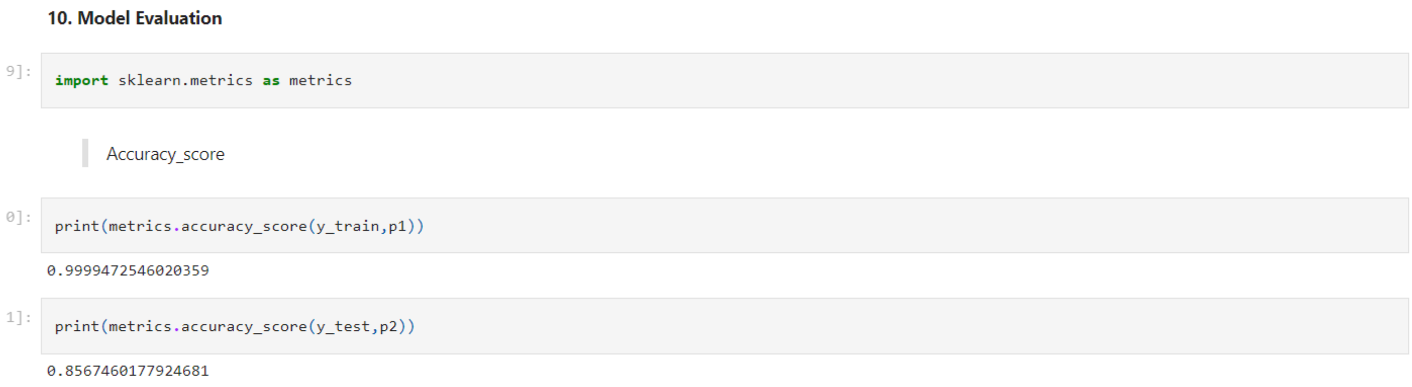
Now using ‘train_test_split’, we split the variables into train and test variables for each variable.



6. Balancing the Data

```
In [69]: sm=SMOTE(random_state=0)
         X_train_res, y_train_res = sm.fit_resample(X_train, y_train)
         print("The number of classes before fit {}".format(Counter(y_train)))
         print("The number of classes after fit {}".format(Counter(y_train_res)))

         The number of classes before fit Counter({0: 90866, 1: 25502})
         The number of classes after fit Counter({0: 90866, 1: 90866})
```

SMOTE (Synthetic Minority Oversampling Technique) is used to increase the number of test cases in abalanced way to avoid overfit cases.



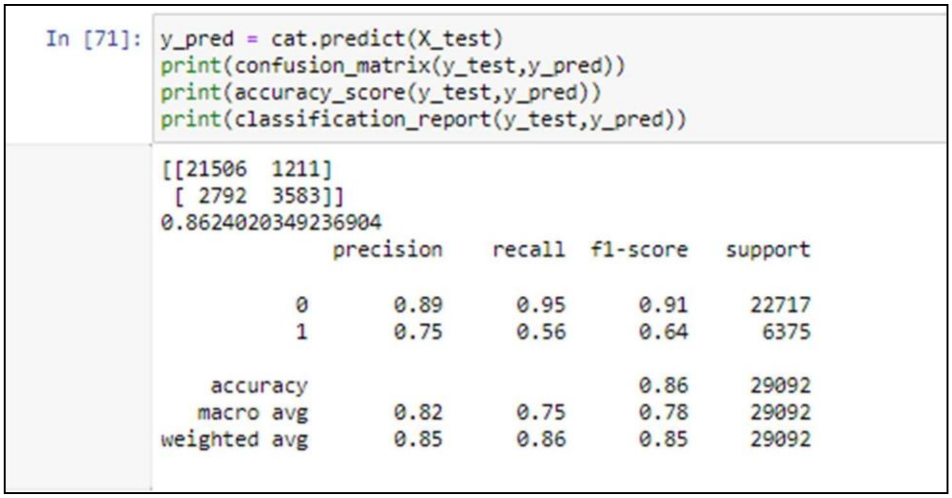
10. Model Evaluation

```
9]:  import sklearn.metrics as metrics
```

Accuracy_score

```
0]:  print(metrics.accuracy_score(y_train,p1))

     0.9999472546020359

1]:  print(metrics.accuracy_score(y_test,p2))

     0.8567460177924681
```

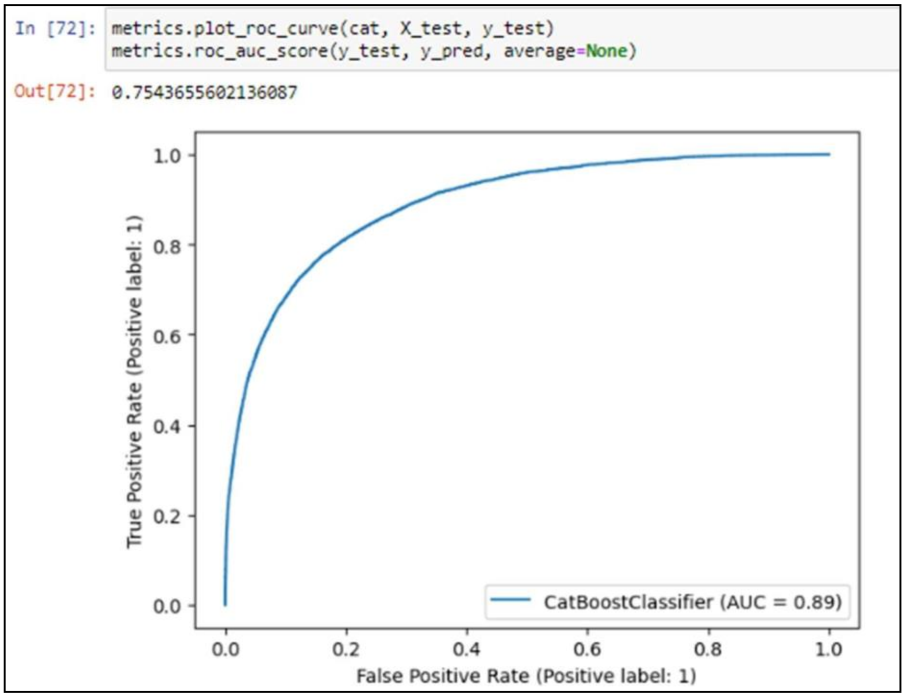
The algorithm chosen here to build the model is CatBoostClassifier. CatBoost is based on gradient boosted decision trees. During training, a set of decision trees is built consecutively. Each successive tree is built with reduced loss compared to the previous trees. The number of trees is controlled by the starting parameters.



```
In [71]: y_pred = cat.predict(X_test)
         print(confusion_matrix(y_test,y_pred))
         print(accuracy_score(y_test,y_pred))
         print(classification_report(y_test,y_pred))
```

```
[[21506  1211]
 [ 2792  3583]]
0.8624020349236904
              precision    recall  f1-score   support

           0       0.89      0.95      0.91     22717
           1       0.75      0.56      0.64      6375

    accuracy                           0.86     29092
   macro avg       0.82      0.75      0.78     29092
weighted avg       0.85      0.86      0.85     29092
```

The above image shows the Confusion Matrix, Accuracy Score and Classification report.

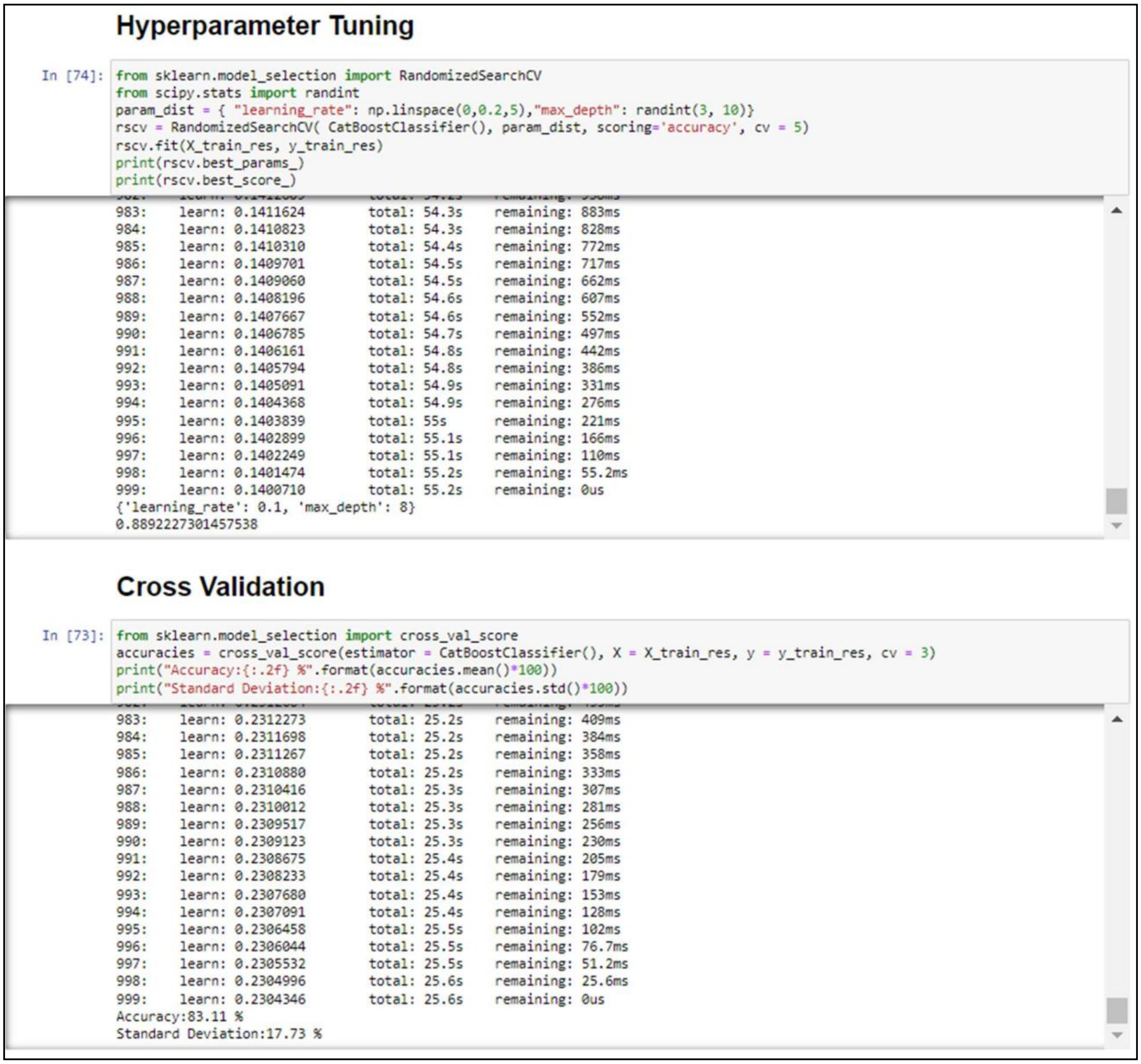


```
In [72]: metrics.plot_roc_curve(cat, X_test, y_test)
         metrics.roc_auc_score(y_test, y_pred, average=None)
```

```
Out[72]: 0.7543655602136087
```

1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate (Positive label: 1)
CatBoostClassifier (AUC = 0.89)
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate (Positive label: 1)

The above image shows the roc curve and roc accuracy score for the built model.



Hyperparameter Tuning

```
In [74]: from sklearn.model_selection import RandomizedSearchCV
         from scipy.stats import randint
         param_dist = { "learning_rate": np.linspace(0,0.2,5),"max_depth": randint(3, 10)}
         rscv = RandomizedSearchCV( CatBoostClassifier(), param_dist, scoring='accuracy', cv = 5)
         rscv.fit(X_train_res, y_train_res)
         print(rscv.best_params_)
         print(rscv.best_score_)
```

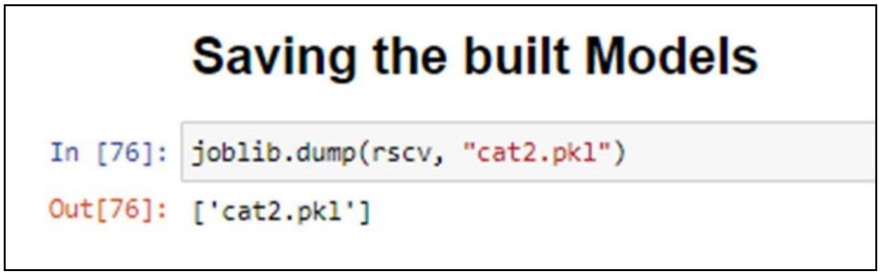
```
983:    learn: 0.1411624        total: 54.3s    remaining: 883ms
984:    learn: 0.1410823        total: 54.3s    remaining: 828ms
985:    learn: 0.1410310        total: 54.4s    remaining: 772ms
986:    learn: 0.1409701        total: 54.5s    remaining: 717ms
987:    learn: 0.1409060        total: 54.5s    remaining: 662ms
988:    learn: 0.1408196        total: 54.6s    remaining: 607ms
989:    learn: 0.1407667        total: 54.6s    remaining: 552ms
990:    learn: 0.1406785        total: 54.7s    remaining: 497ms
991:    learn: 0.1406161        total: 54.8s    remaining: 442ms
992:    learn: 0.1405794        total: 54.8s    remaining: 386ms
993:    learn: 0.1405091        total: 54.9s    remaining: 331ms
994:    learn: 0.1404368        total: 54.9s    remaining: 276ms
995:    learn: 0.1403839        total: 55s      remaining: 221ms
996:    learn: 0.1402899        total: 55.1s    remaining: 166ms
997:    learn: 0.1402249        total: 55.1s    remaining: 110ms
998:    learn: 0.1401474        total: 55.2s    remaining: 55.2ms
999:    learn: 0.1400710        total: 55.2s    remaining: 0us
{'learning_rate': 0.1, 'max_depth': 8}
0.8892227301457538
```

Cross Validation

```
In [73]: from sklearn.model_selection import cross_val_score
         accuracies = cross_val_score(estimator = CatBoostClassifier(), X = X_train_res, y = y_train_res, cv = 3)
         print("Accuracy:{:.2f} %".format(accuracies.mean()*100))
         print("Standard Deviation:{:.2f} %".format(accuracies.std()*100))
```

```
983:    learn: 0.2312273        total: 25.2s    remaining: 409ms
984:    learn: 0.2311698        total: 25.2s    remaining: 384ms
985:    learn: 0.2311267        total: 25.2s    remaining: 358ms
986:    learn: 0.2310880        total: 25.2s    remaining: 333ms
987:    learn: 0.2310416        total: 25.3s    remaining: 307ms
988:    learn: 0.2310012        total: 25.3s    remaining: 281ms
989:    learn: 0.2309517        total: 25.3s    remaining: 256ms
990:    learn: 0.2309123        total: 25.3s    remaining: 230ms
991:    learn: 0.2308675        total: 25.4s    remaining: 205ms
992:    learn: 0.2308233        total: 25.4s    remaining: 179ms
993:    learn: 0.2307680        total: 25.4s    remaining: 153ms
994:    learn: 0.2307091        total: 25.4s    remaining: 128ms
995:    learn: 0.2306458        total: 25.5s    remaining: 102ms
996:    learn: 0.2306044        total: 25.5s    remaining: 76.7ms
997:    learn: 0.2305532        total: 25.5s    remaining: 51.2ms
998:    learn: 0.2304996        total: 25.6s    remaining: 25.6ms
999:    learn: 0.2304346        total: 25.6s    remaining: 0us
Accuracy:83.11 %
Standard Deviation:17.73 %
```

The above image shows the Hyperparameter and Cross Validation score of the model.



Saving the built Models

```
In [76]: joblib.dump(rscv, "cat2.pkl")
Out[76]: ['cat2.pkl']
```

Finally save the model using joblib library.

4.4. Feature-2:

4.5. User Interface

4.6. Index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<title>Weather App using Flask in Python</title>
 <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">
      <style>
      body {
  background-image: url('https://www.worldatlas.com/r/w768/upload/7e/2e/5a/untitled-design-79.jpg');
   background-repeat: no-repeat;
 background-attachment: fixed;
  background-size: cover;


      }
</style>
</head>
<body>
      <div class="container">
          <br><br><br>
                <div class="row"><h2 style="color:Blue;">Weather Prediction App</h2></div>
                <br>
                <div class="row">
                          <b style="color:Tomato;">Get weather details of any city around the world.</b>
                </div>
```

```
        <div class="row">
            {% block content %}
                <form action="{{ url_for("index")}}" method="post">
                <div class="form-group">
                    <label style="color:Red;" for="Email">Email:</label><br>
                <input type="email" id="Email" name="Email" value="{{Email}}"
placeholder="Email" required><br>
                    <label style="color:blue;"
for="cityName"><b>Password:</b></label><br>
                <input type="password" id="password" name="password"
value="{{password}}" placeholder="password" required><br>
                <label for="cityName"><b style="color:Yellow;">City
Name:</b></label><br>
                <input type="text" id="cityName" name="cityName"
value="{{cityName}}" placeholder="City Name" required><br>
                    <br>
                <button class="submit">Find</button>
                {% if error is defined and error %}
                    <br><br><span class="alert alert-danger">Error: Please enter
valid city name.</span></br>
                {% endif %}
                </div>
            {% endblock %}
            {% if data is defined and data %}
            <table class="table table-bordered">
                <thead>
                    <tr>
                        <th>Country Code</th>
                        <th>Coordinate</th>
                        <th>temperature</th>
                        <th>Pressure</th>
                        <th>Humidity</th>
                    </tr>
                </thead>
                <tbody>
                    <tr>
                        <td class="bg-success">{{ data.sys.country }}</td>
```

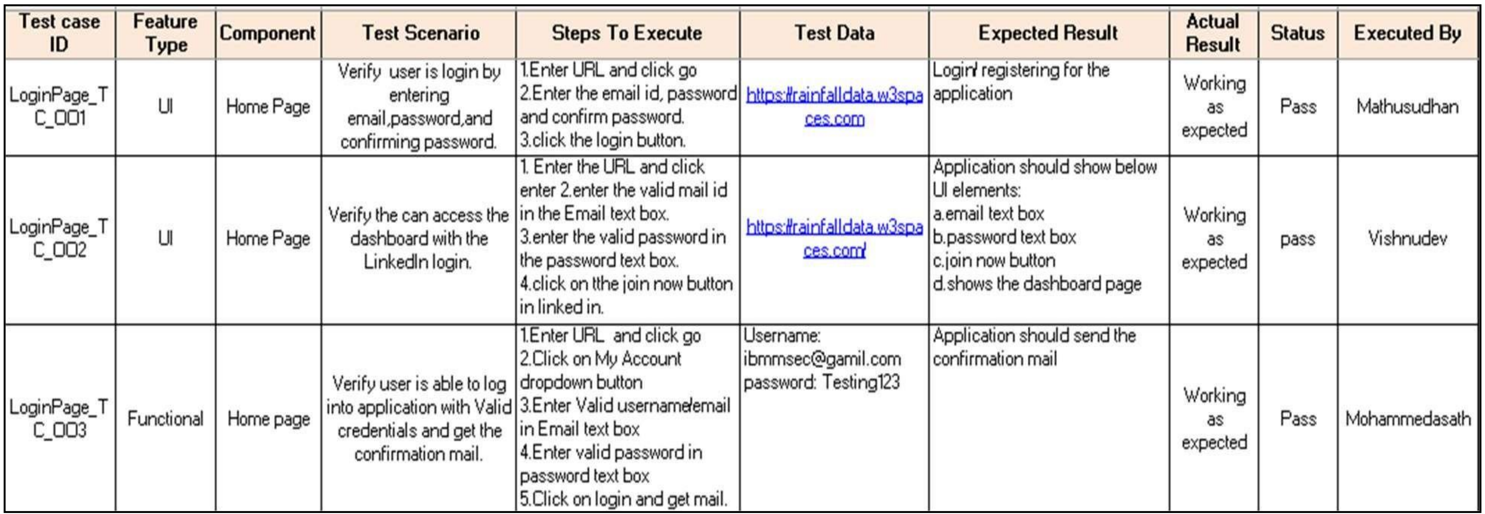
```
                                        <td class="bg-info">{{data.coord.lon }}
{{data.coord.lat}}</td>
                                        <td class="bg-danger">{{data.main.temp }} k</td>
                                        <td class="bg-warning">{{data.main.pressure}}</td>
                                        <td class="bg-primary">{{data.main.humidity}}</td>
                                   </tr>
                              </tbody>
                         </table>
                         {% endif %}
                    </div>
               </div>
</body>
</html>
```

App.py

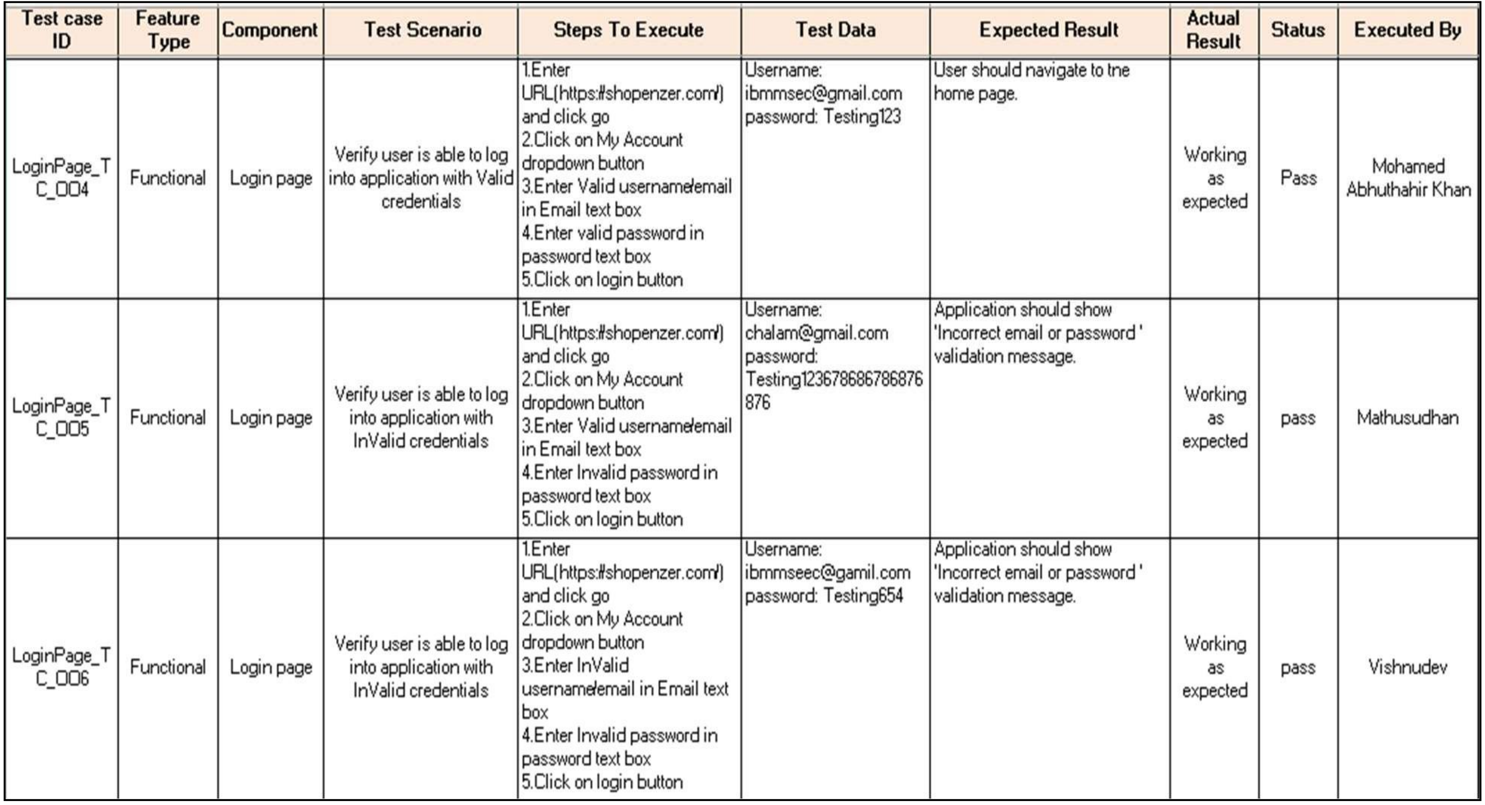
```
from flask import Flask, request, render_template
import requests
from flask import Flask, request, render_template
import requests

app = Flask(_name_)


@app.route('/', methods=["GET", "POST"])
def index():
    weatherData = ''
    error = 0
    cityName = ''
    if request.method == "POST":
        cityName = request.form.get("cityName")
        if cityName:
            weatherApiKey = '3f5d38932ad9ae0caa0302a35fbc8496'
            url = "https://api.openweathermap.org/data/2.5/weather?q=" + cityName + "&appid=" +
weatherApiKey
            weatherData = requests.get(url).json()
        else:
            error = 1
    return render_template('index.html', data=weatherData, cityName=cityName, error=error)


if _name_ == "_main_":
    app.run()

app = Flask(_name_)


@app.route('/', methods=["GET", "POST"])
def index():
    weatherData = ''
    error = 0
    cityName = ''
    if request.method == "POST":
        cityName = request.form.get("cityName")
        if cityName:
            weatherApiKey = '3f5d38932ad9ae0caa0302a35fbc8496'
            url = "https://api.openweathermap.org/data/2.5/weather?q=" + cityName + "&appid=" +
weatherApiKey
            weatherData = requests.get(url).json()
        else:
            error = 1
    return render_template('index.html', data=weatherData, cityName=cityName, error=error)


if _name_ == "_main_":
    app.run()
```

TESTING

4.7. Test Cases



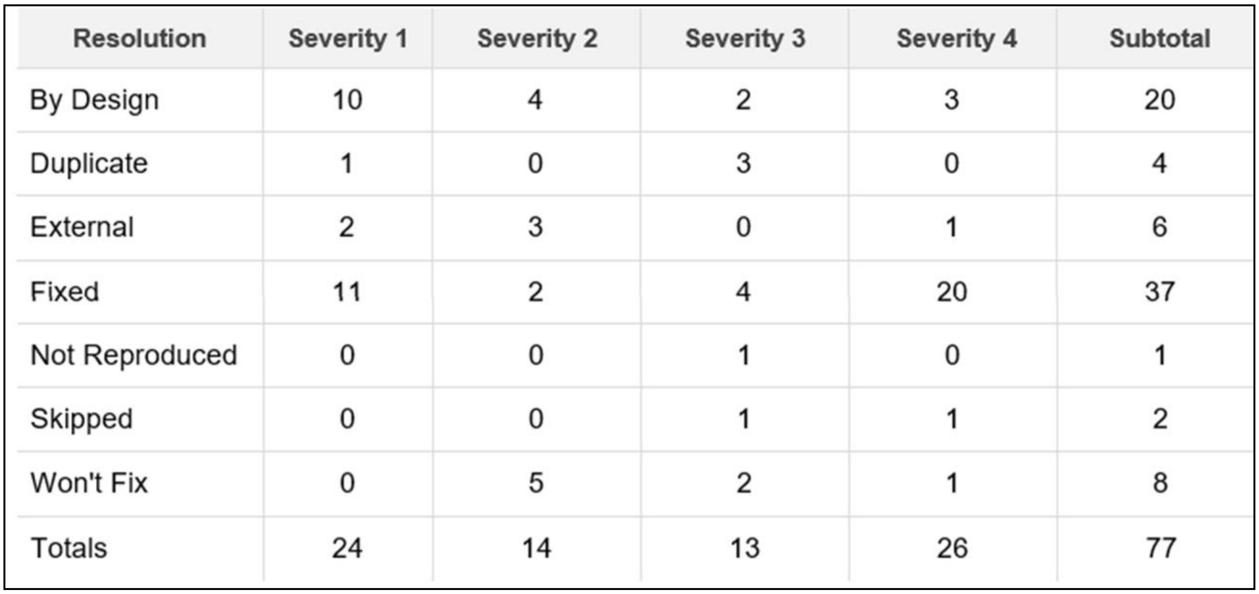
Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Executed By
LoginPage_TC_001	UI	Home Page	Verify user is login by entering email,password,and confirming password.	1.Enter URL and click go 2.Enter the email id, password and confirm password. 3.click the login button.	https://rainfalldata.w3spaces.com	Login/ registering for the application	Working as expected	Pass	Mathusudhan
LoginPage_TC_002	UI	Home Page	Verify the can access the dashboard with the LinkedIn login.	1. Enter the URL and click enter 2.enter the valid mail id in the Email text box. 3.enter the valid password in the password text box. 4.click on tthe join now button in linked in.	https://rainfalldata.w3spaces.com/	Application should show below UI elements: a.email text box b.password text box c.join now button d.shows the dashboard page	Working as expected	pass	Vishnudev
LoginPage_TC_003	Functional	Home page	Verify user is able to log into application with Valid credentials and get the confirmation mail.	1.Enter URL and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login and get mail.	Username: ibmmsec@gamil.com password: Testing123	Application should send the confirmation mail	Working as expected	Pass	Mohammedasath



Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Executed By
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with Valid credentials	1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: ibmmsec@gmail.com password: Testing123	User should navigate to tne home page.	Working as expected	Pass	Mohamed Abhuthahir Khan
LoginPage_TC_005	Functional	Login page	Verify user is able to log into application with InValid credentials	1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: chalam@gmail.com password: Testing123678686786876876	Application should show 'Incorrect email or password ' validation message.	Working as expected	pass	Mathusudhan
LoginPage_TC_006	Functional	Login page	Verify user is able to log into application with InValid credentials	1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: ibmmseec@gamil.com password: Testing654	Application should show 'Incorrect email or password ' validation message.	Working as expected	pass	Vishnudev

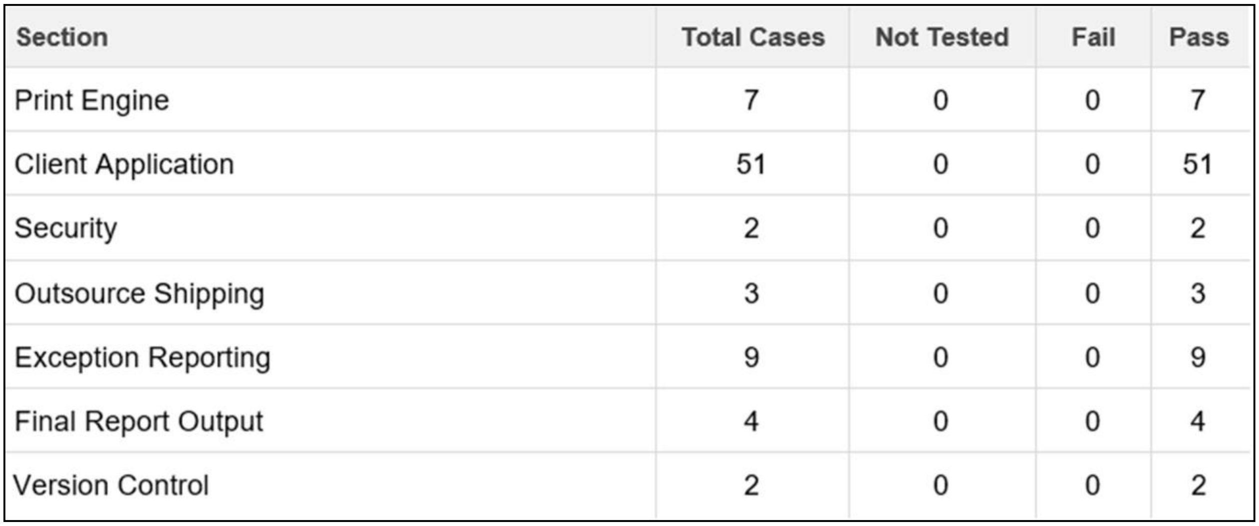
4.8. User Acceptance Testing

8.2.1. Defect Analysis



Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

8.2.2. Testcase Analysis



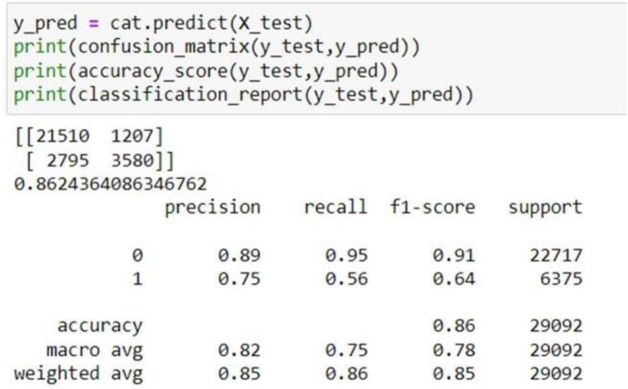
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

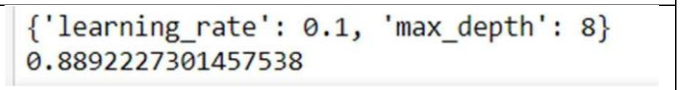
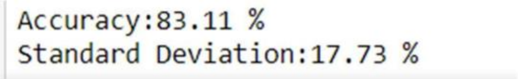
5. RESULTS

5.1. Performance Metrics

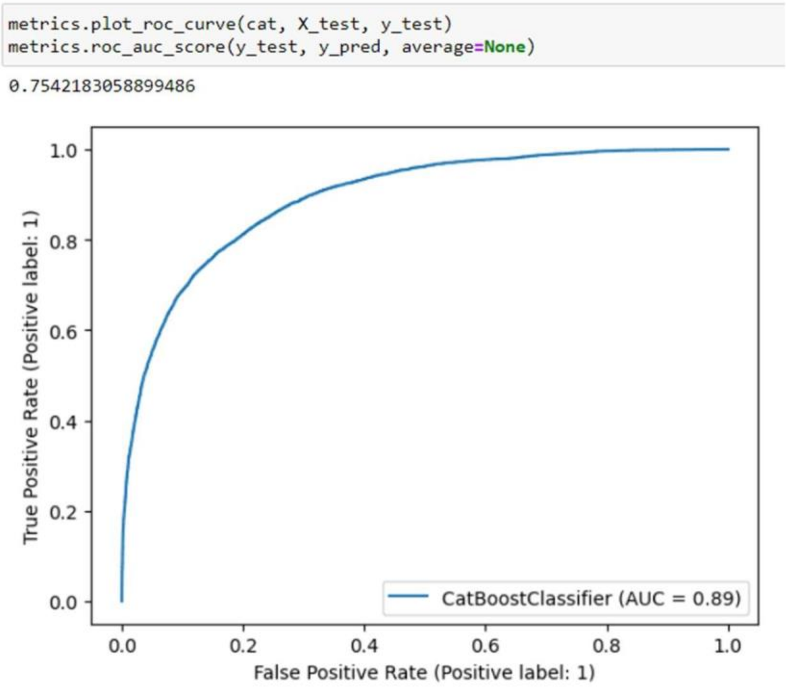
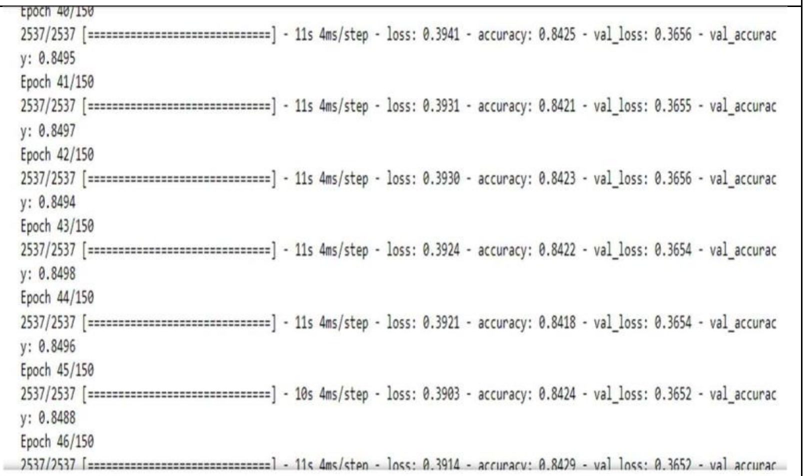
9.1.1. Machine Learning

S.No.	Parameter	Values	Screenshot

1.	Metrics	**Classification Model:** Confusion Matrix - Accuracy Score- Classification Report -	
			`y_pred = cat.predict(X_test)`
			`print(confusion_matrix(y_test,y_pred))`
			`print(accuracy_score(y_test,y_pred))`
			`print(classification_report(y_test,y_pred))`
			`[[21510  1207]`
			`[ 2795  3580]]`
			`0.8624364086346762`
			`precision    recall  f1-score   support`
			`0       0.89      0.95      0.91     22717`
			`1       0.75      0.56      0.64      6375`
			`accuracy                           0.86     29092`
			`macro avg       0.82      0.75      0.78     29092`
			`weighted avg       0.85      0.86      0.85     29092`

2.	Tune the Model	Hyperparameter Tuning –
		Validation Method -
			
			`{'learning_rate': 0.1, 'max_depth': 8}`
			`0.8892227301457538`
			
			`Accuracy:83.11 %`
			`Standard Deviation:17.73 %`

9.1.2. Artificial Intelligence

S.No.	Parameter	Values	Screenshot
1.	Model Summary	-	metrics.plot_roc_curve(cat, X_test, y_test) metrics.roc_auc_score(y_test, y_pred, average=None) 0.7542183058899486
2.	Accuracy	Training Accuracy - Validation Accuracy -	Epoch 40/150 2537/2537 [==============================] - 11s 4ms/step - loss: 0.3941 - accuracy: 0.8425 - val_loss: 0.3656 - val_accurac y: 0.8495 Epoch 41/150 2537/2537 [==============================] - 11s 4ms/step - loss: 0.3931 - accuracy: 0.8421 - val_loss: 0.3655 - val_accurac y: 0.8497 Epoch 42/150 2537/2537 [==============================] - 11s 4ms/step - loss: 0.3930 - accuracy: 0.8423 - val_loss: 0.3656 - val_accurac y: 0.8494 Epoch 43/150 2537/2537 [==============================] - 11s 4ms/step - loss: 0.3924 - accuracy: 0.8422 - val_loss: 0.3654 - val_accurac y: 0.8498 Epoch 44/150 2537/2537 [==============================] - 11s 4ms/step - loss: 0.3921 - accuracy: 0.8418 - val_loss: 0.3654 - val_accurac y: 0.8496 Epoch 45/150 2537/2537 [==============================] - 10s 4ms/step - loss: 0.3903 - accuracy: 0.8424 - val_loss: 0.3652 - val_accurac y: 0.8488 Epoch 46/150 2537/2537 [==============================] - 11s 4ms/step - loss: 0.3914 - accuracy: 0.8429 - val_loss: 0.3652 - val_accurac

6. ADVANTAGES AND DISADVANTAGES

6.1. Advantages

- Farmers can know when to plant or harvest their crops
- People can choose where and when to take their holidays to take advantages of good weather
- Surfers known when large waves are expected
- Regions can be evacuated if hurricanes or floods are expected
- Aircraft and shipping rely heavily on accurate weather forecasting
- It will help the farmers to take precautionary steps
- Technological solutions to improve their production

6.2. Disadvantages

- Weather is extremely difficult to forecast correctly
- It is expensive to monitor so many variables from so many sources
- The computers needed to perform the millions of calculations necessary are expensive
- The weather forecasters get blamed if the weather is different from the forecast
- Leading to poor growth and overall health of crop
- Limited Foods Access

7. CONCLUSION

The weather prediction has become one of the most essential entities now a days. To improve the risk management systems and to know the weather in coming days in an automatic and in scientific way, many models have been emerging to assist in weather Prediction. In this paper, we have seen building a Weather Prediction Web Application from scratch by making use of 6 different ML algorithms namely CatBoost Classifier, RandomForset Classifier, Logistic Regression, GaussianNB, KNN and XGB Classifier. In the result section, the results from the all the six models and its results such as Accuracy, Error rate, mean absolute error, Root mean squared error, Relative squared error, Root relative squared error and time taken to build the model are tabulated. The results show that the CatBoost Classifier and XGB Classifier has output the results of high accuracy than all the other classifiers that were used. When coming to the time taken to build the model, The CatBoost Classifier outperforms all the other classifiers in solving the Problem under scrutiny.

8. FUTURE SCOPE

In upcoming future updates, the WEATHER FORECASTING application will have additional features suchas:

- Live Location tracking
- News on Live Disasters
- Weather Forecast for next one week
- Will deploy as android app
- Help in predicting which crop will be best suited according to weather conditions

13.APPENDIX

1.1. Source Code

13.1.1. Ipynb file Link: RAINFALL PREDICTION

13.1.2. UI Link: FILE

1.2. Links

13.2.1. GITHUB

13.2.2. DEMO VIDEO