Exploratory Analysis Of RainFall Data In India For Agriculture.

Team ID:PNT2022TMID10910

TEAM MEMBERS:

TEAM LEADER : PRIYA VISHNU A S [811519104087]
TEAM MEMBER 1: MAHESH KUMAR R [811519104066]
TEAM MEMBER 2:SURYA PRAKASH P [811519104109]
TEAM MEMBER 3:VIJAYKUMAR D [811519104119]

Index

1. INTRODUCTION
    - 1.1 Project Overview
    - 1.2 Purpose
2. LITERATURE SURVEY
    - 2.1 Existing problem
    - 2.2 References
    - 2.3 Problem Statement Definition
3. IDEATION & PROPOSED SOLUTION
    - 3.1 Empathy Map Canvas
    - 3.2 Ideation & Brainstorming
    - 3.3 Proposed Solution
    - 3.4 Problem Solution fit
4. REQUIREMENT ANALYSIS
    - 4.1 Functional requirement
    - 4.2 Non-Functional requirements
5. PROJECT DESIGN
    - 5.1 Data Flow Diagrams
    - 5.2 Solution & Technical Architecture
    - 5.3 User Stories
6. PROJECT PLANNING & SCHEDULING
    - 6.1 Sprint Planning & Estimation
    - 6.2 Sprint Delivery Schedule
    - 6.3 Reports from JIRA
7. CODING & SOLUTIONING (Explain the features added in the project along with code)
    - 7.1 Feature 1
8. TESTING
    - 8.1 User Acceptance Testing
9. RESULTS
    - 9.1 Performance Metrics
10. ADVANTAGES & DISADVANTAGES
11. CONCLUSION
12. FUTURE SCOPE
13. APPENDIX
    - Source Code GitHub & Project
    - Demo Link

1.Introduction:

1.1 project overview

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 floods 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.

This comparative study is conducted concentrating on the following aspects: modelling inputs, Visualizing the data, modelling 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 analysing 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 and saved in pkl format. Once the model is saved, we integrate it with flask application and also deploy the model in IBM.

1.2 purpose

1.Develop a model that can predict the rainfall that will help the agriculture sector, so that rainfall doesn't become a barrier for the agricultural production.

2. Additionally ,This prediction can also be helpful to many sectors like forest plantation and tourism.

2. LITERATURE SURVEY :

Exploratory Data Analysis of Indian Rainfall Data : Author - Anusha Gajinkar

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.

Understanding the Monsoon in India:

We will see what exactly is monsoon, different types of monsoon winds in India, which subdivisions of India receives rainfall from which monsoon winds and why only particular subdivisions receive highest rainfall during this monsoon season.
The Southwest Monsoon usually starts in the first week of June and ends by first week of September and monsoon usually starts retreating from the Indian Subcontinent by the start of September and leaves the subcontinent completely by the end of November. And as we have seen in the previous graphs that Southwest monsoon provides almost 80% of the rainfall in India. This Southwest Monsoon has two branches, namely Arabian Sea branch and Bay of Bengal Branch.

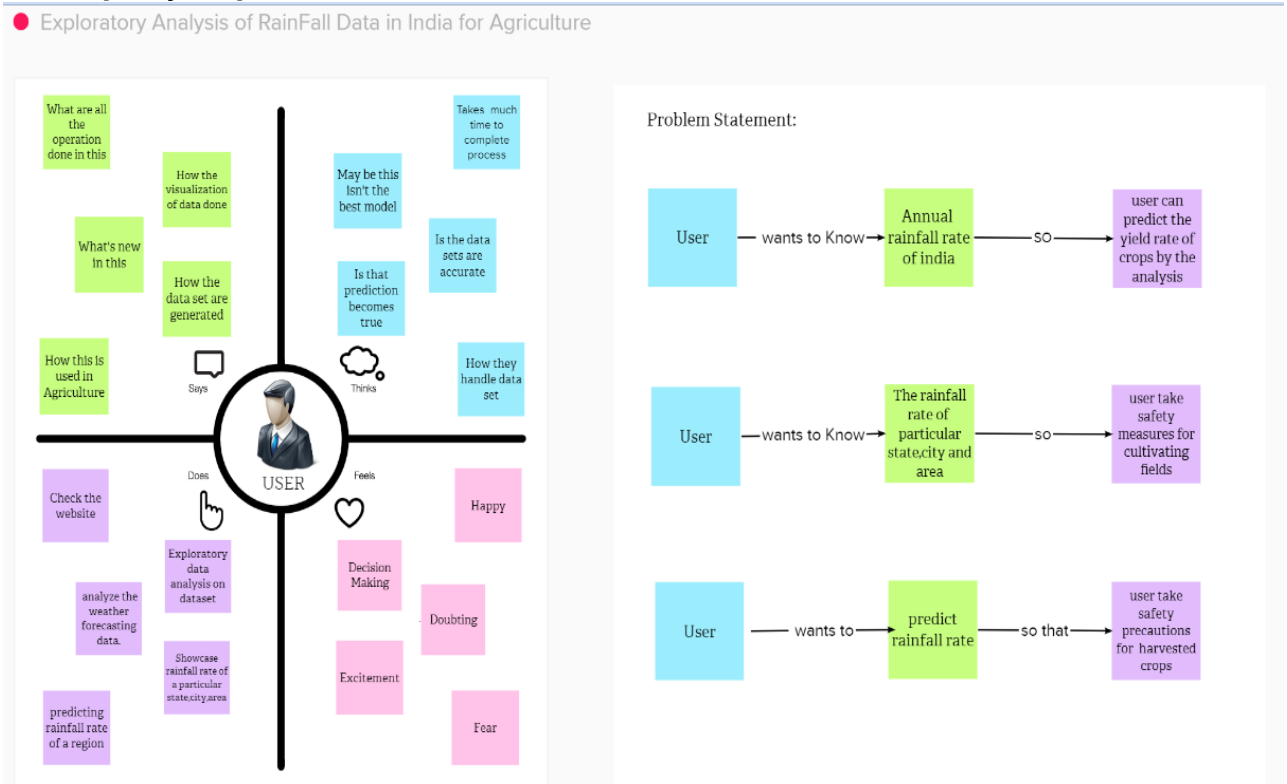
Now, due to the presence of high rising Western Ghats which runs along the South West coast of India in the states of Kerala, Karnataka, Goa and Maharashtra, they block the Arabian Sea branch of southwest monsoon and hence these regions receive very high rainfall during monsoon season. This is shown in the graph below. (For all the graphs we have considered the last 3 decades i.e. last 30 years data (1987–2017)).

Annual Rainfall trend over the years for whole India:

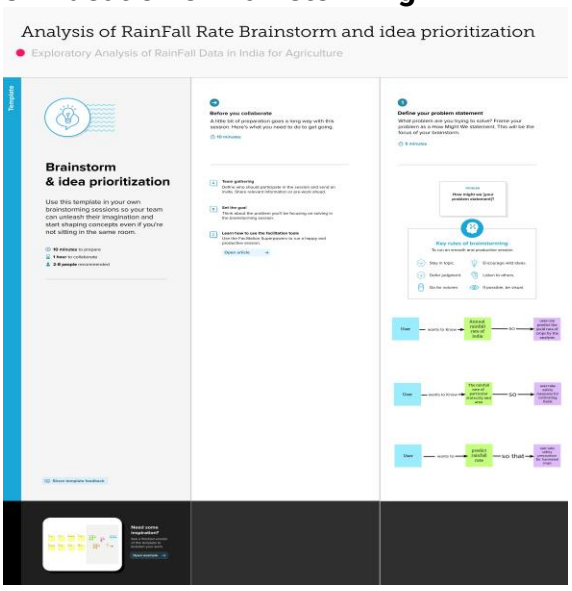
10 years moving average was plotted, we can see that there is a decreasing trend in rainfall in the recent years. The trend analysis of Annual rainfall considering India as whole show decreasing trend however when trend is analysed for all subdivision individually we can see some division showing increasing trend and some showing decreasing trend. It showed that is is import to study subdivision for better forecasting.

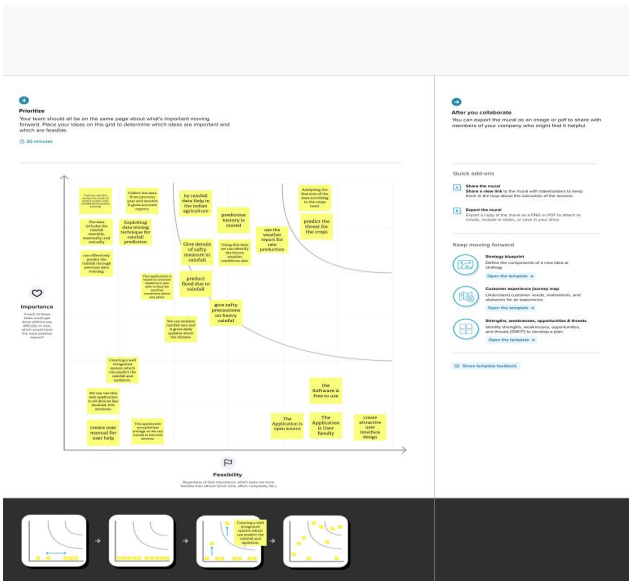
3. IDEATION & PROPOSED SOLUTION

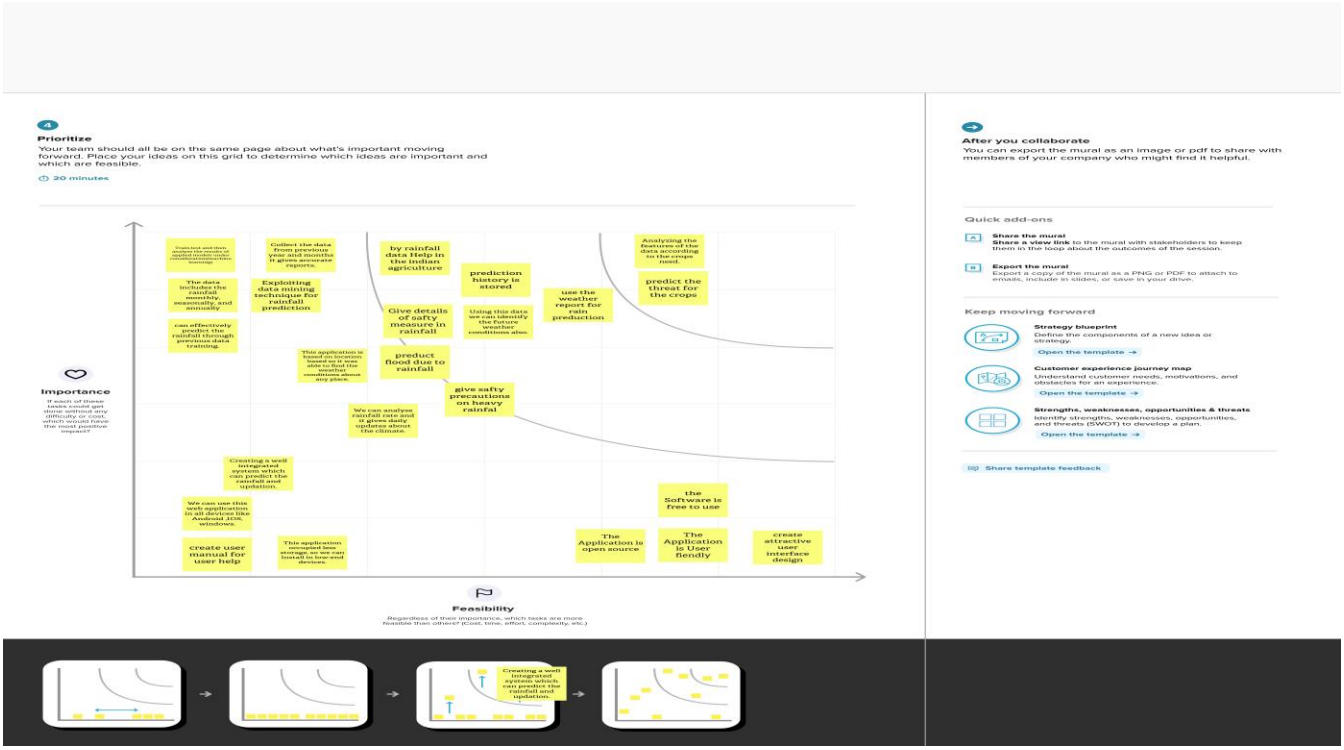
3.1 Empathy Map Canvas :



3.2 Ideation & Brainstorming







3.3 Proposed Solution

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To develop and train the machine learning model and then verify the outcomes of Exploratory Analysis of Rainfall in India for Agriculture.
2.	Idea / Solution description	Primarily ,data cleaning and other future Engineering process are carried out then Applying the classification/Regression algorithms by different models, which can help in predicting the rainfall. Dimensionality reduction(PCA)technique should be used. To achieve such results, will be using five various regression or classification models and select the best one among - Multiple linear regression, KNN regression, SVM(Support Vector Machine) regression, DTR(Decision tree

		regression), RFE(Random forest regression).
3.	Novelty / Uniqueness	1.To create such progressive scenarios we need to implement conventional or machine learning models with high accuracy that are important to achieve the results. 2. After applying the models in the data, after that comparison will be done , it is to testify that; selected model can do prediction of rainfall with or above 70 percent of correctness that is R-squared score and RMSE(Root Mean Square Error) less than 20 percent of compared to the target variable's mean value. 3. Creating, optimizing and cross validating the algorithms of the new model.
4.	Social Impact / Customer Satisfaction	1.The prediction enhanced the average profit done by the farmers and can reduce the migration of the labour from the village, due to the losses in agriculture. 2. The farmers produce the crops in the three major seasons with the insight of prediction. 3. The prediction can also help the farmer to take necessary measures for the future aspect.

5.	Business Model (Revenue Model)	1.Develop a model that can predict the rainfall that will help the agriculture sector, so that rainfall doesn't become a barrier for the agricultural production. 2. Additionally ,This prediction can also be helpful to many sectors like forest plantation and tourism.
6.	Scalability of the Solution	1.The prediction enhances various upgradable features for its scalability and efficiently in various fields. 1.Accurate prediction can assist farmers in determining when they should work most efficiently in their day-to-day operations. Weather forecast helps in controlling the pests and other crop diseases to spread over the field. 2. A good balance of rain and proper irrigation can lead to faster-growing plants, which can cut down on germination time and the length between seeding and harvest. 3. Besides disease, rainfall can also determine how fast a crop will grow from seed, including when it will be ready for harvesting.

3.4 Problem Solution fit :

Project Title: Exploratory Analysis of Rain Fall Data in India for Agriculture.

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID10910

1. CUSTOMER SEGMENT(S) — CS

Farmers are the primary customers

6. CUSTOMER CONSTRAINTS — CC

Lack of awareness

No Knowledge in scientific methodology

Financial crises

5. AVAILABLE SOLUTIONS — AS

This project provides solution to farmers during all form of season

Well planned irrigation and drainage system

Setting up preliminary protection and rain cover

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

2. JOBS-TO-BE-DONE — J&P

Updating rainfall data

Cleaning and exploring data

Visualizing the data

PROBLEMS

Wrong input and missing data

Data latency

Precision

Type conversion (numerical, categorical)

9. PROBLEM ROOT CAUSE — RC

Poor resource management

Unpredictable weather

Improper water and stagnant management for crops

7. BEHAVIOUR — BE

Rely on uneducated guidance

Seek institutional aid

Take on excessive debt

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

Identify strong TR & EM

3. TRIGGERS — TR

repeated financial and time loss

poor yielding technique

adapting to climatical changes

4. EMOTIONS: BEFORE / AFTER — EM

BEFORE: panic in case excess rainfall occurs, damage of crops during the pre-mature stage and financial hurdles

AFTER: can easily know the amount of rain falls in advance thus increasing the crop productivity

10. YOUR SOLUTION — SL

Our historical rainfall data predict an approximation of the future

Can predict the rainfall pattern for a specific time period

Categorize and analyse the intensity of rain

develop a web page and provide the necessary guidelines to formers based on the result

8. CHANNELS of BEHAVIOUR — CH

Channels that support behaviour are

Choosing optimal data

Proper visualization of data (actual vs predicted)

Promoting and marketing

compatibility for all devices

Identify strong TR & EM

4.Requirement analysis:

Date	03 October 2022
Team ID	PNT2022TMID10910
Project Name	Project - Exploratory Analysis of Rain Fall Data in India for Agriculture. Exploratory Analysis of Rain Fall Data in India for Agriculture.
Maximum Marks	4 Marks

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Forecasting Accuracy	Retrieve the forecasted weather conditions and measure the accuracy
FR-4	Prediction details	User should enter the current location to get the predicted result
FR-5	Forecast	Forecasted flood probability from the rainfall amount is displayed on the webpage.
FR-6	result	The web page will display the condition as a report, visualization plot, graphs and images

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

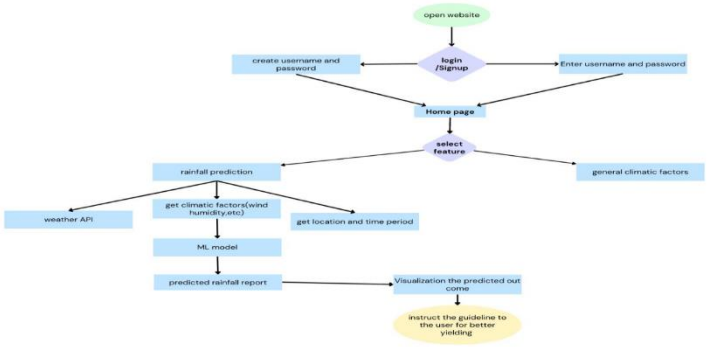
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The usability of the website is to make all users will be satisfied with the our requirements of the product. The user should reach the summarized text or result with one button press if possible
NFR-2	**Security**	The security of the 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 website does not go offline. The users can be reach and use program at any time, so maintenance should not be a big issue
NFR-4	**Performance**	The performance of the website is to provide data to all users without unnecessary delay and provide 24*7 availability

5. Project design:

5.1 Data flow diagrams

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 Solution and Technical Architecture:

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

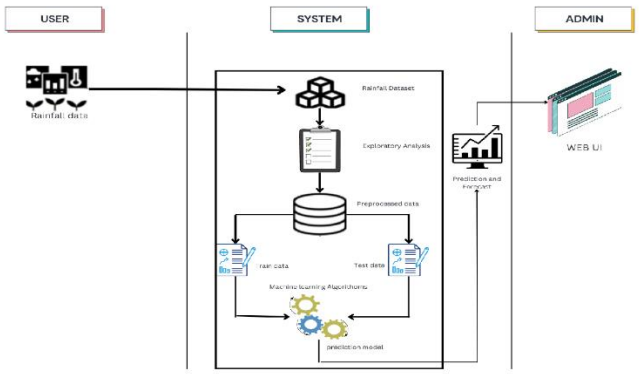
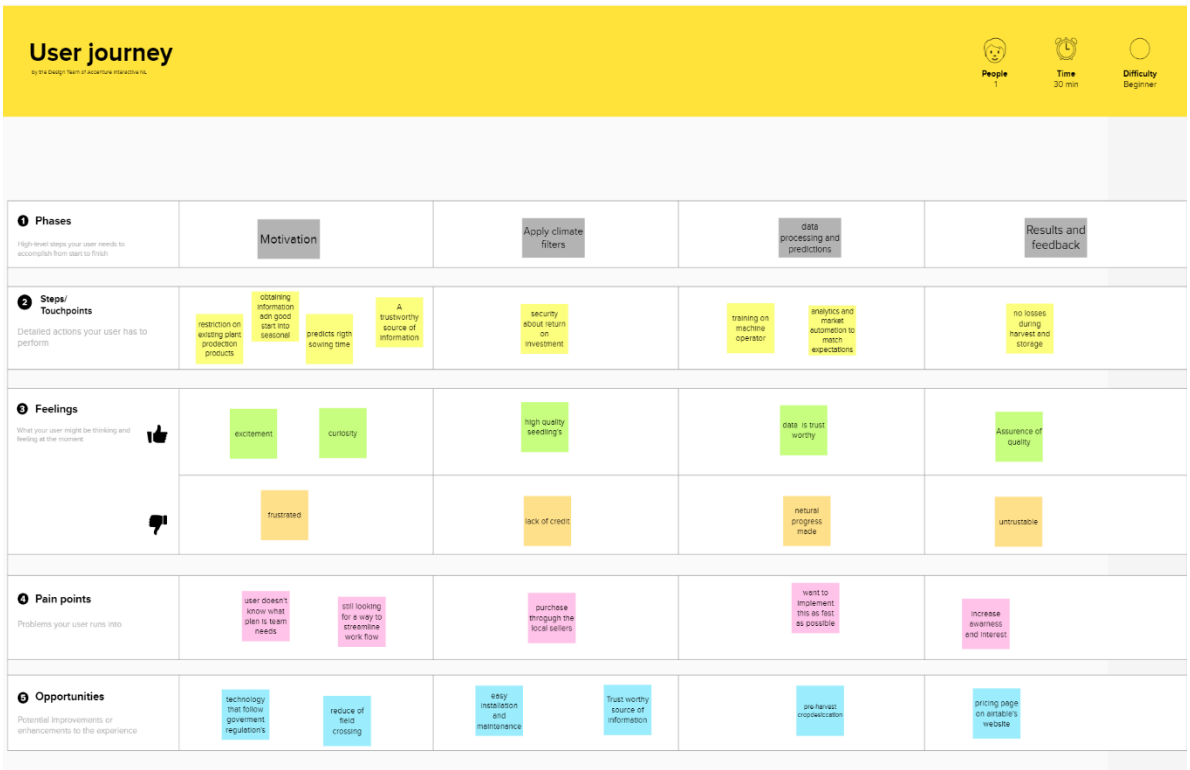


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	The user interacts with the application through a web UI.	HTML, CSS, python, Flask
2.	Application Logic-1	Logic for registration.	Python
3.	Application Logic-2	Logic for login to the application.	Python
4.	Application Logic-3	Integrating machine learning model and the webpage	Flask
5.	Database	Numeric data.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	MySQL
7.	File Storage	To store files such as prediction report	Local Filesystem
8.	External API-1	Allows developers access to critical forecasts, alerts, and observations, along with other weather data	IBM Weather API.
9.	Machine Learning Model	Predictive modeling is a statistical technique using machine learning and data mining to predict and forecast likely future outcomes with the aid of historical and existing data	Predictive modeling
10.	Infrastructure (Server / Cloud)	Application Deployment on Local System Local Server Configuration: built-in flask web server	Flask web Server.

5.3 user-journey map:



6.Project Planning and Scheduling:

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 (web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
	confirmation	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
	Registration	USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
	Registration	USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-6	As a user, I can view the details about the system and can navigate through the pages	Successfully redirected	High	Sprint-3
	Dashboard	USN-7	Create the dashboard of the web page	This page contains my database	High	Sprint-3
	Rainfall prediction	USN-8	To enter the climatic factor and location	Valid information	High	Sprint-4
	Algorithm	USN-9	Execution of ML Algorithm	Report generated	High	Sprint -4
	Visualization	USN-10	The reports are generated as predicted plot, graphs and images	Visual representation	High	Sprint -4
	guidelines	USN-11	By the report itself generates the instruction guideline for crop yielding	Guidelines generated	High	Sprint-4
Core development team	Update data	USN-12	change the current updated data set	Updated dataset	Medium	Sprint -3
	Core function	USN-13	This web sit is available on all devices and screen sizes, user friendly webpage		medium	Sprint -3

7. CODING & SOLUTIONING (Explain the features added in the project along with code) .

7.1 Future engineering and model fitting:

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:

#For Data loading and preprocessing
import pandas as pd

#For matrix operations
import numpy as np

#For plotting
import matplotlib.pyplot as plt
get_ipython().run_line_magic('matplotlib', 'inline')
import seaborn as sns

#For splitting the data
from sklearn.model_selection import train_test_split

#For data preprocessing
from sklearn.preprocessing import StandardScaler

#For hyperparameter tuning
from sklearn.model_selection import RandomizedSearchCV,GridSearchCV
#For appling LogisticRegression
from sklearn.linear_model import LogisticRegression

#For model/vatiable persistence
#from sklearn.externals import joblib
#For math operations
import math

#To see the progress of the iterations
#from tqdm import tqdm

#Performance metrices
from sklearn.metrics import roc_auc_score,roc_curve,auc,log_loss,confusion_matrix

#For encoding the features
from sklearn.preprocessing import LabelEncoder,LabelBinarizer

#For ignoring warnings
import warnings
warnings.filterwarnings("ignore")
```

In[3]:

```
df=pd.read_csv("https://raw.githubusercontent.com/IBM-EPBL/IBM-Project-8468-1658920292/main/Data%20Collection/weatherAUS.csv")
```

In[4]:

```
df.drop_duplicates(inplace=True)
```

In[5]:

```
df.isnull().any()
```

In[6]:

```
sns.set(style="whitegrid")
sns.countplot(df.RainTomorrow)
plt.title("Target labels")
plt.show()
```

In[7]:

```
#Separating the data based on its class label.
data_yes = df[df['RainTomorrow']=='Yes']
data_no = df[df['RainTomorrow']=='No']
```

In[8]:

```
#Observing the mode for all columns when RainTomorrow = Yes
mode_values_for_yes = data_yes.mode()
mode_values_for_yes
```

In[9]:

```
data_no['MaxTemp'].median()
```

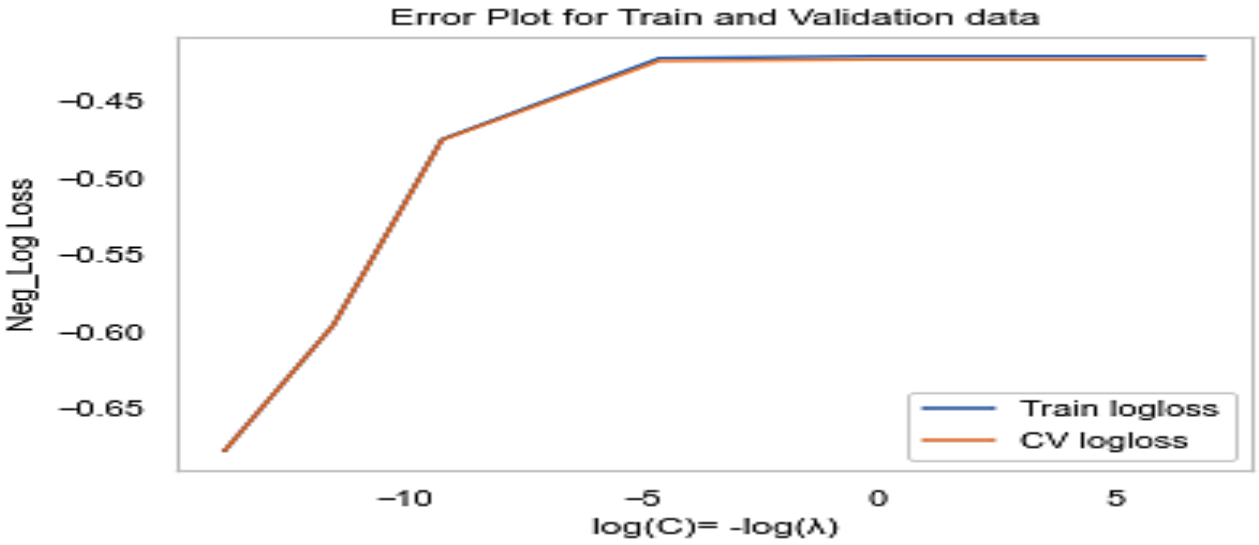
In[10]:

```
#For Temparatures we cannot replace NaN values with 0, hence replacing NaN with its respective mode value
data_yes['MinTemp'].fillna(value=data_yes['MinTemp'].mode()[0],inplace=True )
data_no['MinTemp'].fillna(value=data_no['MinTemp'].mode()[0],inplace=True )

data_yes['MaxTemp'].fillna(value=data_yes['MaxTemp'].mode()[0],inplace=True )
data_no['MaxTemp'].fillna(value=data_no['MaxTemp'].mode()[0],inplace=True )

data_yes['Temp9am'].fillna(value=data_yes['Temp9am'].mode()[0],inplace=True )
data_no['Temp9am'].fillna(value=data_no['Temp9am'].mode()[0],inplace=True )

data_yes['Temp3pm'].fillna(value=data_yes['Temp3pm'].mode()[0],inplace=True )
```

```
data_no['Temp3pm'].fillna(value=data_no['Temp3pm'].mode()[0],inplace=True )

# For humidity also
data_yes['Humidity9am'].fillna(value=data_yes['Humidity9am'].mode()[0],inplace=True )
data_no['Humidity9am'].fillna(value=data_no['Humidity9am'].mode()[0],inplace=True )

data_yes['Humidity3pm'].fillna(value=data_yes['Humidity3pm'].mode()[0],inplace=True )
data_no['Humidity3pm'].fillna(value=data_no['Humidity3pm'].mode()[0],inplace=True )

# For the rain fall feature we can replace NaN with 0.0 which says there is no rain fall
data_yes['Rainfall'].fillna(value=0.0,inplace=True)
data_no['Rainfall'].fillna(value=0.0,inplace=True)
data_yes['Pressure9am'].fillna(value=data_yes['Pressure9am'].median(),inplace=True )
data_no['Pressure9am'].fillna(value=data_no['Pressure9am'].median(),inplace=True )

data_yes['Pressure3pm'].fillna(value=data_yes['Pressure3pm'].median(),inplace=True )
data_no['Pressure3pm'].fillna(value=data_no['Pressure3pm'].median(),inplace=True )

data_yes['WindSpeed9am'].fillna(value=data_yes['WindSpeed9am'].median(),inplace=True )
data_no['WindSpeed9am'].fillna(value=data_no['WindSpeed9am'].median(),inplace=True )

data_yes['WindSpeed3pm'].fillna(value=data_yes['WindSpeed3pm'].median(),inplace=True )
data_no['WindSpeed3pm'].fillna(value=data_no['WindSpeed3pm'].median(),inplace=True )

#WindGustSpeed -- replacing with median
data_yes['WindGustSpeed'].fillna(value=data_yes['WindGustSpeed'].median(),inplace=True)
data_no['WindGustSpeed'].fillna(value=data_no['WindGustSpeed'].median(),inplace=True)

# In[11]:

# For RainToday feature we cannot fill any value, so better to remove the NaN values
data_yes.dropna(inplace=True)
data_no.dropna(inplace=True)

# In[12]:

data_filled= data_yes.append(data_no, ignore_index=True)

# In[13]:

data_filled.isnull().any()


# In[14]:

# sorting the data based on data (Time based splitting)
data_filled=data_filled.sort_values(by='Date')

# In[15]:
```

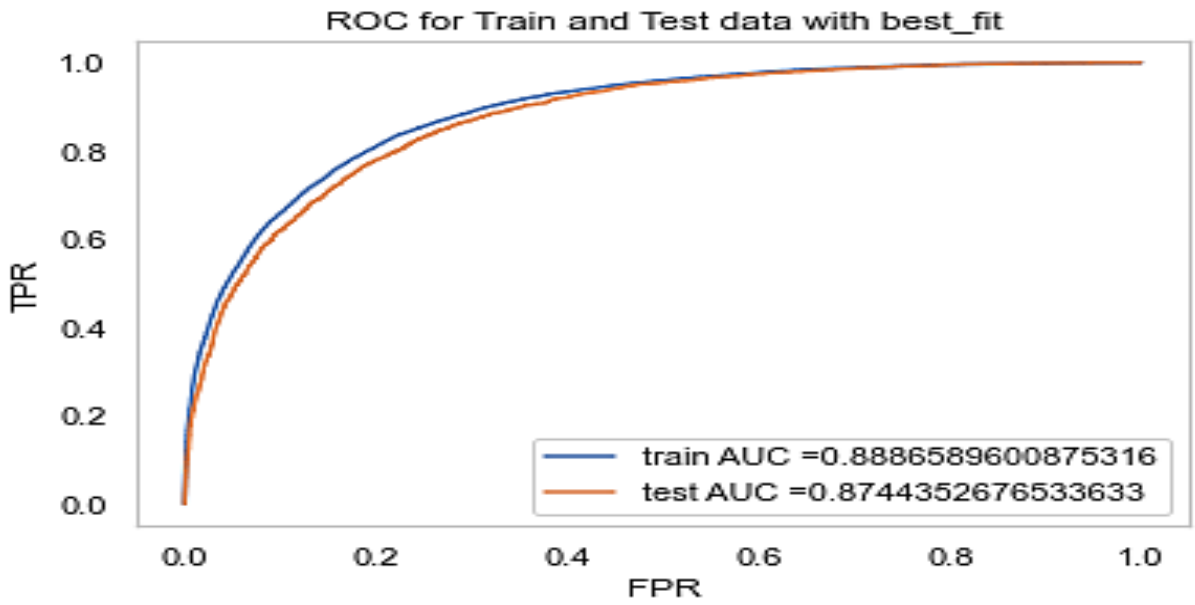
```
#Removing unwanted features, RISK_MM is same as target label hence removing with data and
loaction
data_final = data_filled.drop(['Date', 'Location'], axis=1)
data_final

# In[16]:

#Outliers we are checking only for numerical features
sns.set(style="whitegrid")
plt.figure(figsize=(10, 6))
sns.boxplot(data=data_final[['MinTemp','MaxTemp','Temp9am','Temp3pm']])


# In[17]:

data_final= data_final[data_final['Humidity3pm']!=0.0]
data_final= data_final[data_final['Humidity9am']!=0.0]

# In[18]:

WindGustDir_encode = LabelEncoder()
data_final['WindGustDir']=WindGustDir_encode.fit_transform(data_final['WindGustDir'])

WindDir9am_encode = LabelEncoder()
data_final['WindDir9am']=WindDir9am_encode.fit_transform(data_final['WindDir9am'])

WindDir3pm_encode = LabelEncoder()
data_final['WindDir3pm']=WindDir3pm_encode.fit_transform(data_final['WindDir3pm'])

RainToday_encode = LabelEncoder()
data_final['RainToday']=RainToday_encode.fit_transform(data_final['RainToday'])

RainTomorrow_encode = LabelEncoder()
data_final['RainTomorrow']=RainTomorrow_encode.fit_transform(data_final["RainTomorrow"])

# In[19]:

Y= data_final['RainTomorrow']
X = data_final.drop(['RainTomorrow'],axis=1)

# In[20]:
column_names=X.columns.tolist()
column_names
# In[21]:

X_train, X_test, y_train, y_test = train_test_split(X, Y, train_size=0.80,shuffle=False)

# In[22]:
```

```
scaler= StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)


# In[23]:

X_test

# In[24]:

def plotErrors(k,train,cv):

    plt.plot(k, train, label='Train logloss')
    plt.plot(k, cv, label='CV logloss')
    plt.legend()
    plt.xlabel("log(C)= -log(λ)")
    plt.ylabel("Neg_Log Loss")
    plt.title("Error Plot for Train and Validation data")
    plt.grid()
    plt.show()
```




```
#
# ###  Modeling the data using Logisitic Regression
# #### Hyper-parameter tuning

# In[25]:

parameters={'C':[10**-6,10**-5,10**-4, 10**-2, 10**0, 10**2, 10**3] }
log_c = list(map(lambda x : float(math.log(x)),parameters['C']))
```

```
clf_log = LogisticRegression(penalty='l2',class_weight='balanced')

clf = GridSearchCV(clf_log, parameters, cv=5, scoring='neg_log_loss',return_train_score =True)
clf.fit(X_train, y_train)

train_loss= clf.cv_results_['mean_train_score']
cv_loss = clf.cv_results_['mean_test_score']

plotErrors(k=log_c,train=train_loss,cv=cv_loss)


# In[26]:

clf = clf.best_estimator_
clf

# In[27]:

#Trainig with the best value of C
clf.fit(X_train, y_train)

# In[28]:

#Printing the log-loss for both trian and test data
train_loss = log_loss(y_train, clf.predict_proba(X_train)[:,1])
test_loss  =log_loss(y_test, clf.predict_proba(X_test)[:,1])

print("Log_loss on train data is :{}".format(train_loss))
print("Log_loss on test data is :{}".format(test_loss))

# In[29]:

#Plotting AUC
train_fpr, train_tpr, thresholds = roc_curve(y_train, clf.predict_proba(X_train)[:,1])
test_fpr, test_tpr, thresholds = roc_curve(y_test, clf.predict_proba(X_test)[:,1])
plt.plot(train_fpr, train_tpr, label="train AUC ="+str(auc(train_fpr, train_tpr)))
plt.plot(test_fpr, test_tpr, label="test AUC ="+str(auc(test_fpr, test_tpr)))
plt.legend()
plt.xlabel("FPR")
plt.ylabel("TPR")
plt.title("ROC for Train and Test data with best_fit")
plt.grid()
plt.show()
```



In[30]:

clf.coef_[0]

In[31]:

feature_weights=sorted(zip(clf.coef_[0],column_names),reverse = True)

In[32]:

feature_weights
checking various machine learning algorithms for best fit..!
In[33]:

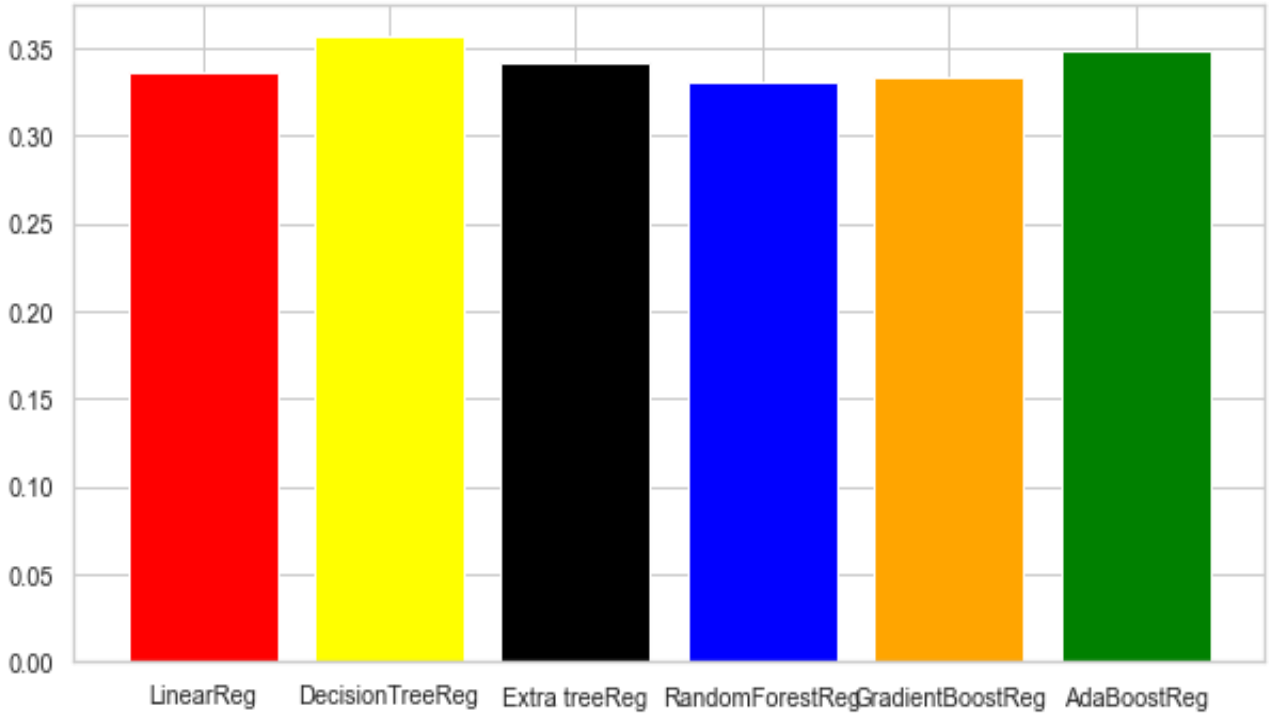
import matplotlib.pyplot as plt
from scipy.stats import itemfreq
import seaborn as sns
from sklearn import linear_model
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import Lasso
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression, Ridge, LassoCV, ElasticNetCV
from sklearn.metrics import mean_squared_error, make_scorer
#from sklearn.model_selection import train_test_split
get_ipython().run_line_magic('matplotlib', 'inline')
import datetime
from datetime import date, timedelta
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import r2_score

```
from sklearn.metrics import mean_squared_error
from sklearn.ensemble import ExtraTreesRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import AdaBoostRegressor
from sklearn.ensemble import AdaBoostClassifier


# ## (1)LinearRegression

# In[34]:

# Fit the linear model
model = linear_model.LinearRegression()
results = model.fit(X_train, y_train)
print(results)
y_pred = results.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
rmse1 = np.sqrt(mse)
print("The RMSE IS :",rmse1)
r2 = r2_score(y_test, y_pred)
print("\nSCORE",r2)

# In[35]:

import statsmodels.api as sm

model = sm.OLS(y_train, X_train)
results = model.fit()
# Statsmodels gives R-like statistical output
results.summary()

sing scikitlearn
reg=linear_model.LinearRegression()
cv_results=cross_val_score(reg,X_train,y_train,cv=5)
print(cv_results)
print(np.mean(cv_results))
print(np.std(cv_results))

#Using cross validation of score 5
# In[37]:
#regularization
ridge = Ridge(alpha=0.1, normalize = True)
ridge.fit(X_train,y_train)
ridge_pred=ridge.predict(X_test)
ridge.score(X_test,y_test)
# ## (2)DecisionTree Regressor
```

```
# In[38]:
dtr=DecisionTreeRegressor(max_depth=10,min_samples_leaf=5,max_leaf_nodes=5)
# In[39]:
dtr.fit(X_train,y_train)
y_pred=dtr.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
rmse2 = np.sqrt(mse)
print("RMSE = ",rmse2)
r2 = r2_score(y_test, y_pred)
print(r2)

# ## (3)ExtraTreesRegressor

# In[40]:

etr = ExtraTreesRegressor()

# Choose some parameter combinations to try

parameters = {'n_estimators': [10],
          'criterion': ['mse'],
          'max_depth': [5],
          'min_samples_split': [2],
          'min_samples_leaf': [1]
         }
#We have to use RandomForestRegressor's own scorer (which is R^2 score)

#Determines the cross-validation splitting strategy /to specify the number of folds in a
(Stratified)KFold

grid_obj = GridSearchCV(etr, parameters,
                cv=3,
                n_jobs=-1, #Number of jobs to run in parallel
                verbose=1)
grid_obj = grid_obj.fit(X_train, y_train)

# Set the clf to the best combination of parameters
etr = grid_obj.best_estimator_

# Fit the best algorithm to the data.
etr.fit(X_train, y_train)

# In[41]:

y_pred = etr.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
rmse3 = np.sqrt(mse)
print("RMSE = ",rmse3)
```

```
r2 = r2_score(y_test, y_pred)
print(r2)

# ## (4) RandomForestRegression

# In[42]:

from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

model = RandomForestRegressor()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
rmse4 = np.sqrt(mse)
print("RMSE = ",rmse4)
r2 = r2_score(y_test, y_pred)
print(r2)

# ## (5) Gradient Boosting Regressor

# In[43]:

modelgbr = GradientBoostingRegressor(n_estimators=100,max_depth=2, random_state=42)
modelgbr.fit(X_train,y_train)
y_pred = modelgbr.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
rmse5 = np.sqrt(mse)
print("RMSE = ",rmse5)
r2 = r2_score(y_test, y_pred)
print(r2)

# ## (6) Ada Boosting Regressor

# In[44]:

regr = AdaBoostRegressor(random_state=42, n_estimators=100)
regr.fit(X_train,y_train)
y_pred = regr.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
rmse6 = np.sqrt(mse)
print("RMSE = ",rmse6)
r2 = r2_score(y_test, y_pred)
print(r2)

# In[45]:
```

```
reg = ['LinearReg','DecisionTreeReg','Extra treeReg','RandomForestReg','GradientBoostReg','AdaBoostReg']
rmse = [rmse1,rmse2,rmse3,rmse4,rmse5,rmse6]
c = ['red', 'yellow', 'black', 'blue', 'orange','green']
plt.figure(figsize=(10, 5))
plt.bar(reg,rmse,color=c)
# we plotted using rmse value -->ie,lesser rmse value have more accuracy
```




```
# In[46]:

def will_rain_fall_for_this_conditions(xq):

    xq["WindGustDir"]=WindGustDir_encode.transform([xq["WindGustDir"]])
    xq["WindDir9am"]=WindDir9am_encode.transform([xq["WindDir9am"]])
    xq["WindDir3pm"]=WindDir3pm_encode.transform([xq["WindDir3pm"]])
    xq["RainToday"]=RainToday_encode.transform([xq["RainToday"]])
    xq=np.array(list((xq.values())))
    final_xq = scaler.transform(xq.reshape(1, -1))
    chance=clf.predict_proba(final_xq)[:,1]
    if chance>=0.5:
        print("Yes, there is a {} % chance of rain can fall on tommorow ".format(chance*100))
    else:
        print("No, there is only {}% chance of rainfall hence we cannot expect rain on tommorow ".format(chance*100))
    print("Because today's Humidity at 3pm ={}%,Atmosphereic Pressure at 9am={}millibars,and Wind Gust Speed ={}km/hr, which are very good sign for rainfall"
```

```
        .format(Humidity3pm,Pressure9am,WindGustSpeed))


# In[47]:

#Giving one query point here

MinTemp   = 26.2
MaxTemp   = 31.7
Rainfall   = 2.8
Evaporation   = 5.4
Sunshine   = 3.5
WindGustDir   = "NNW"
WindGustSpeed   = 57
WindDir9am   = "NNW"
WindDir3pm   = "NNW"
WindSpeed9am   = 20
WindSpeed3pm   = 13
Humidity9am   = 81
Humidity3pm   = 95
Pressure9am   = 1007.2
Pressure3pm   = 1006.1
Cloud9am   = 7
Cloud3pm   = 8
Temp9am   = 28.8
Temp3pm   = 25.4
RainToday   ="Yes"

# In[48]:

point = [MinTemp,MaxTemp,Rainfall,
        Evaporation,Sunshine,WindGustDir,
        WindGustSpeed,WindDir9am,WindDir3pm,
        WindSpeed9am,WindSpeed3pm,Humidity9am,
        Humidity3pm,Pressure9am,Pressure3pm,
        Cloud9am,Cloud3pm,Temp9am,Temp3pm,RainToday]

xq=dict()
for i,name in enumerate(column_names):
    xq[name]=point[i]

# In[49]:

will_rain_fall_for_this_conditions(xq)

# ### Yes, there is a [99.04083842] % chance of rain can fall on tommorow

# In[50]:
```

```
import pickle
#saving the model

# In[53]:
pickle.dump(modelgbr,open('rainfall.pkl','wb'))
pickle.dump(model,open('linreg.pkl','wb'))
pickle.dump(model,open('randFor.pkl','wb'))
```

8.Testing

8.1 User Acceptance Testing

User Acceptance Testing (UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done.

User acceptance testing (UAT), also called *application testing* or *end-user testing*, is a phase of software development in which the software is tested in the real world by its intended audience. UAT is often the last phase of the software testing process and is performed before the tested software is released to its intended market. The goal of UAT is to ensure software can handle real-world tasks and perform up to development specifications.

In UAT, users are given the opportunity to interact with the software before its official release to see if any features have been overlooked or if it contains any bugs. UAT can be done in-house with volunteers, by paid test subjects using the software or by making the test version available for download as a free trial. The results from the early testers are forwarded to the developers, who make final changes before releasing the software commercially.

UAT is effective for ensuring quality in terms of time and software cost, while also increasing transparency with users. UAT also enables developers to work with real cases and data, and if successful, the process can validate business requirements.

9.Results

9.1 Performance Metrics

Project metrics are used to track the progress and performance of a project.Monitoring parts of a project like productivity, scheduling, and scope make it easier for team leaders to see what's on track.As a project evolves, managers need access to changing deadlines or budgets to meet their client's expectations.

Performance testing is a non-functional software testing method used to check the speed, scalability, reliability, responsiveness, and performance of an app/website. Various performance testing methods include a spike, volume, endurance, stress, load, etc. These performance testing types help determine the app performance under fluctuating networks, varying user loads, varying bandwidths, etc. During performance testing, certain key performance indicators (KPIs), also known as performance testing metrics, are used to measure the effectiveness of this testing method. These KPIs define the effectiveness of the performance tests for businesses.

Performance testing metrics are the measures or parameters gathered during the performance and load testing processes. With the help of these metrics, performance test engineers or QA teams determine the success of the performance testing process and further identify the critical areas in the software that needs more attention/improvement.

10.ADVANTAGES AND DISADVANTAGES:

Advantages:

(i) Besides disease, rainfall can also determine how fast a crop will grow from seed,

(ii)when it will be ready for harvesting,it should be useful

(iii)A good balance of rain and proper irrigation can lead to faster-growing plants,

(iv)plants can cut down on germination time and length between seeding &harvest

Disadvantages:

(i)Requires more time and cost.

(ii)Difficult to understand the technical terms.

(iii)Inflexibility .

11 .CONCLUSIONS:

The efficient way of performing exploratory analysis of rainfall in indian agriculture using machine learning algorithms is implemented using the rainfall prediction website that is hosted on IBM Cloud platform.

To ensure the smooth functioning of the web site operation. I have hosted the website in IBM Db2 & Kubernates Cluster to make sure the operations are running successfully Cloud lambda function is used and to deploy the application IBM Db2 service is used.

12.Future Scope:

Upgrading the UI that is more user-friendly which will help many users to access the website and also ensures that many farmers and stakeholders can be added into the community.

Using elastic load balancer, it helps to handle multiple requests at the same time which will maintain the uptime of the website with negligible downtime .Adding more authetic features to the website and past history review.

Appendix:

Source code:

app.py

```
# Importing essential libraries
from flask import Flask, render_template, request, url_for, redirect, flash,session,abort
from flask import *
from google_auth_oauthlib.flow import Flow
#from authlib.integration.flask_client import OAuth
import pickle
import catboost as ctp
import numpy as np
import os
import pathlib


# Load the Random Forest CLassifier model
filename = './Model/rainfall.pkl'
clf = pickle.load(open(filename, 'rb'))

app = Flask(__name__)
app.secret_key = "super secret key"

os.environ['OAUTHLIB_INSECURE_TRANSPORT'] = '1'
GOOGLE_CLIENT_ID = "734412505534-sjfq66jpvdhnog36hjg3t6fi7mk8qsjr.apps.googleusercontent.com"

client_secrets_file = os.path.join(pathlib.Path(__file__).parent, "client_Auth.json")


flow = Flow.from_client_secrets_file(
    client_secrets_file=client_secrets_file,
    scopes=["https://www.googleapis.com/auth/contacts",
    "https://www.google.com/m8/feeds/",
```

```
    "https://www.googleapis.com/auth/contacts","openid"],
    redirect_uri="http://127.0.0.1:5000/callbackByGoogle"
    )                                          #"https://www.googleapis.com/auth/userinfo.profile",
"https://www/googleapis.com/auth/userinfo.email",

@app.route("/loginByGoogle")
def loginByGoogle():
   authorization_url, state = flow.authorization_url()
   session["state"] = state
   return redirect(authorization_url)

@app.route("/callbackByGoogle")
def callbackByGoogle():
    flow.fetch_token(authorization_response = request.url)
    if not session["state"] == request.args["state"]:
        abort(500)
    credentials = flow.credentials
    request_session = request.session()
    cached_session = cachecontrol.CacheControl(request_session)
    token_request = google.auth.transport.requests.Reauest(sessio = cached_session)

    id_info = id_token.verify_oauth2.token(
        id_token=credentials._id_token,
        request = token_request,
        audience = GOOGLE_CLIENT_ID
    )
    session['google_id']= id_info.get('sub')
    session['name']=id_info.det('name')
    return redirect('/Dashboard')


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

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

@app.route('/req', methods=['POST'])
def req():
    if request.method == 'POST':
        mail = request.form['mailid']
        username = request.form['username']
        password = request.form['password']
        confirm = request.form['conpassword']
    return render_template('login.html')

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

@app.route('/submit', methods=['POST'])
def submit():
    print(request)
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        if username != "VishnuAS@0073" or password != "Eizo@0073":
            flash("you are not allowed to logged in")
        else:
            flash("you are successfuly logged in")
            return redirect(url_for('dashboard'))
    return render_template('login.html')

@app.route('/dashboard')
def dashboard():
    return render_template('index.html')
@app.route('/predict', methods=['POST'])
def predict():
```

```
    temp_array = list()
    if request.method == 'POST':
        min_temp = float(request.form['MinTemp'])
        max_temp = float(request.form['MaxTemp'])
        rainfall = float(request.form['Rainfall'])
        evaporation = float(request.form['Evaporation'])
        sunshine = float(request.form['Sunshine'])
        wind_gust_speed = float(request.form['WindGustSpeed'])
        wind_speed_9am = float(request.form['WindSpeed9am'])
        wind_speed_3pm = float(request.form['WindSpeed3pm'])
        humidity9am = float(request.form['Humidity9am'])
        humidity3pm = float(request.form['Humidity3pm'])
        pressure9am = float(request.form['Pressure9am'])
        pressure3pm = float(request.form['Pressure3pm'])
        cloud9am = float(request.form['Cloud9am'])
        cloud3pm = float(request.form['Cloud3pm'])
        temp9am = float(request.form['Temp9am'])
        temp3pm = float(request.form['Temp3pm'])
        rainToday = request.form['RainToday']
        if rainToday == 'Yes':
            rainToday = 1
        else:
            rainToday = 0
        year = request.form['year']
        month = request.form['month']
        day = request.form['day']

        location = request.form['Location']
        if location == 'Albany':
            temp_array = temp_array + [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
        elif location == 'Albury':
            temp_array = temp_array + [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
        elif location == 'AliceSprings':
```

```
            temp_array = temp_array + [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
        elif location == 'BadgerysCreek':
            temp_array = temp_array + [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
        elif location == 'Ballarat':
            temp_array = temp_array + [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
        elif location == 'Bendigo':
            temp_array = temp_array + [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
        elif location == 'Brisbane':
            temp_array = temp_array + [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
        elif location == 'Cairns':
            temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
        elif location == 'Canberra':
            temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
        elif location == 'Cobar':
            temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
        elif location == 'CoffsHarbour     ':
            temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
        elif location == 'Dartmoor':
            temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
        elif location == 'Darwin':
            temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
        elif location == 'GoldCoast':
            temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

```
    elif location == 'Hobart':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'Katherine':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'Launceston':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'Melbourne':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'MelbourneAirport':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'Mildura':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'Moree':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'MountGambier':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'MountGinini':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'Newcastle':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'Nhil':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'NorahHead':
```

```
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'NorfolkIsland':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'Nuriootpa':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'PearceRAAF':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'Penrith':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'Perth':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'PerthAirport':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'Portland':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'Richmond':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'Sale':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'SalmonGums':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'Sydney':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

```
    elif location == 'SydneyAirport':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'Townsville':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'Tuggeranong':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'Uluru':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'WaggaWagga':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'Walpole':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'Watsonia':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'Williamtown':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'Witchcliffe':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'Wollongong':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    elif location == 'Woomera':
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1]
    else:
```

```
        temp_array = temp_array + [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    wind_gust_dir = request.form['WindGustDir']
    if wind_gust_dir == 'ENE':
        temp_array = temp_array + [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
    elif wind_gust_dir == 'ESE':
        temp_array = temp_array + [0,1,0,0,0,0,0,0,0,0,0,0,0,0,0]
    elif wind_gust_dir == 'N':
        temp_array = temp_array + [0,0,1,0,0,0,0,0,0,0,0,0,0,0,0]
    elif wind_gust_dir == 'NE':
        temp_array = temp_array + [0,0,0,1,0,0,0,0,0,0,0,0,0,0,0]
    elif wind_gust_dir == 'NNE':
        temp_array = temp_array + [0,0,0,0,1,0,0,0,0,0,0,0,0,0,0]
    elif wind_gust_dir == 'NNW':
        temp_array = temp_array + [0,0,0,0,0,1,0,0,0,0,0,0,0,0,0]
    elif wind_gust_dir == 'NW':
        temp_array = temp_array + [0,0,0,0,0,0,1,0,0,0,0,0,0,0,0]
    elif wind_gust_dir == 'S':
        temp_array = temp_array + [0,0,0,0,0,0,0,1,0,0,0,0,0,0,0]
    elif wind_gust_dir == 'SE':
        temp_array = temp_array + [0,0,0,0,0,0,0,0,1,0,0,0,0,0,0]
    elif wind_gust_dir == 'SSE':
        temp_array = temp_array + [0,0,0,0,0,0,0,0,0,1,0,0,0,0,0]
    elif wind_gust_dir == 'SSW':
        temp_array = temp_array + [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0]
    elif wind_gust_dir == 'SW':
        temp_array = temp_array + [0,0,0,0,0,0,0,0,0,0,0,1,0,0,0]
    elif wind_gust_dir == 'W':
        temp_array = temp_array + [0,0,0,0,0,0,0,0,0,0,0,0,1,0,0]
    elif wind_gust_dir == 'WNW':
        temp_array = temp_array + [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0]
    elif wind_gust_dir == 'WSW':
        temp_array = temp_array + [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1]
    else :
        temp_array = temp_array + [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
```

```
wind_dir_9am = request.form['WindDir9am']
if wind_dir_9am == 'ENE':
   temp_array = temp_array + [1,0,0,0,0,0,0,0,0,0,0,0,0,0]
elif wind_dir_9am == 'ESE':
   temp_array = temp_array + [0,1,0,0,0,0,0,0,0,0,0,0,0,0]
elif wind_dir_9am == 'N':
   temp_array = temp_array + [0,0,1,0,0,0,0,0,0,0,0,0,0,0]
elif wind_dir_9am == 'NE':
   temp_array = temp_array + [0,0,0,1,0,0,0,0,0,0,0,0,0,0]
elif wind_dir_9am == 'NNE':
   temp_array = temp_array + [0,0,0,0,1,0,0,0,0,0,0,0,0,0]
elif wind_dir_9am == 'NNW':
   temp_array = temp_array + [0,0,0,0,0,1,0,0,0,0,0,0,0,0]
elif wind_dir_9am == 'NW':
   temp_array = temp_array + [0,0,0,0,0,0,1,0,0,0,0,0,0,0]
elif wind_dir_9am == 'S':
   temp_array = temp_array + [0,0,0,0,0,0,0,1,0,0,0,0,0,0]
elif wind_dir_9am == 'SE':
   temp_array = temp_array + [0,0,0,0,0,0,0,0,1,0,0,0,0,0]
elif wind_dir_9am == 'SSE':
   temp_array = temp_array + [0,0,0,0,0,0,0,0,0,1,0,0,0,0]
elif wind_dir_9am == 'SSW':
   temp_array = temp_array + [0,0,0,0,0,0,0,0,0,0,1,0,0,0]
elif wind_dir_9am == 'SW':
   temp_array = temp_array + [0,0,0,0,0,0,0,0,0,0,0,1,0,0]
elif wind_dir_9am == 'W':
   temp_array = temp_array + [0,0,0,0,0,0,0,0,0,0,0,0,1,0]
elif wind_dir_9am == 'WNW':
   temp_array = temp_array + [0,0,0,0,0,0,0,0,0,0,0,0,0,1]
elif wind_dir_9am == 'WSW':
   temp_array = temp_array + [0,0,0,0,0,0,0,0,0,0,0,0,0,1]
else :
   temp_array = temp_array + [0,0,0,0,0,0,0,0,0,0,0,0,0,0]
```

```
wind_dir_3pm = request.form['WindDir3pm']
if wind_dir_3pm == 'ENE':
   temp_array = temp_array + [1,0,0,0,0,0,0,0,0,0,0,0,0,0]
elif wind_dir_3pm == 'ESE':
   temp_array = temp_array + [0,1,0,0,0,0,0,0,0,0,0,0,0,0]
elif wind_dir_3pm == 'N':
   temp_array = temp_array + [0,0,1,0,0,0,0,0,0,0,0,0,0,0]
elif wind_dir_3pm == 'NE':
   temp_array = temp_array + [0,0,0,1,0,0,0,0,0,0,0,0,0,0]
elif wind_dir_3pm == 'NNE':
   temp_array = temp_array + [0,0,0,0,1,0,0,0,0,0,0,0,0,0]
elif wind_dir_3pm == 'NNW':
   temp_array = temp_array + [0,0,0,0,0,1,0,0,0,0,0,0,0,0]
elif wind_dir_3pm == 'NW':
   temp_array = temp_array + [0,0,0,0,0,0,1,0,0,0,0,0,0,0]
elif wind_dir_3pm == 'S':
   temp_array = temp_array + [0,0,0,0,0,0,0,1,0,0,0,0,0,0]
elif wind_dir_3pm == 'SE':
   temp_array = temp_array + [0,0,0,0,0,0,0,0,1,0,0,0,0,0]
elif wind_dir_3pm == 'SSE':
   temp_array = temp_array + [0,0,0,0,0,0,0,0,0,1,0,0,0,0]
elif wind_dir_3pm == 'SSW':
   temp_array = temp_array + [0,0,0,0,0,0,0,0,0,0,1,0,0,0]
elif wind_dir_3pm == 'SW':
   temp_array = temp_array + [0,0,0,0,0,0,0,0,0,0,0,1,0,0]
elif wind_dir_3pm == 'W':
   temp_array = temp_array + [0,0,0,0,0,0,0,0,0,0,0,0,1,0]
elif wind_dir_3pm == 'WNW':
   temp_array = temp_array + [0,0,0,0,0,0,0,0,0,0,0,0,0,1]
elif wind_dir_3pm == 'WSW':
   temp_array = temp_array + [0,0,0,0,0,0,0,0,0,0,0,0,0,1]
else :
   temp_array = temp_array + [0,0,0,0,0,0,0,0,0,0,0,0,0,0]
```

```
    temp_array = [min_temp, max_temp, rainfall, evaporation, sunshine, wind_gust_speed, wind_speed_9am, wind_speed_3pm, humidity9am, humidity3pm, pressure9am, pressure3pm, cloud9am, cloud3pm, temp9am, temp3pm, rainToday, year, month, day] + temp_array
    data = np.array([temp_array])
    my_prediction = int(clf.predict(data)[0])

    return render_template('result.html', my_prediction=my_prediction)

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

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

signin.html

```
<!DOCTYPE html>
<html lang="en" >
<head>
 <meta charset="UTF-8">
 <title>Sign up</title>
 <link href="https://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet">
 <link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet" integrity="sha384-wvfXpqpZZVQGK6TAh5PVlGOfQNHSoD2xbE+QkPxCAFlNEevoEH3Sl0sibVcOQVnN" crossorigin="anonymous">
 <link rel="stylesheet" href="../static/log style.css">

</head>
<body>
<!-- partial:index.partial.html -->
<div class="box-form">
        <div class="left">
                <div class="overlay">
                <h1>Hello World.</h1>
```

```
                <p>Exploratory Analysis of RainFall Data in India for Agriculture</p>
                </div>
        </div>

                <div class="right">
                <h4>Sign up</h4>
                <p>Do have an account? <a href="{{ url_for('login')}}">go to login</a> it takes less
than a minute</p>
                <form class="inputs" form action="{{ url_for('req')}}" method="post">
        <h3>Username</h3>
                        <input type="text" placeholder="username" id="username"
name="username">
                        <h3>Mail</h3>
                        <input type="email" placeholder="mail" id="mailid" name="mailid">
        <h3>Password</h3>
                        <input type="password" placeholder="password" id="username"
name="password">
                        <h3>confirm Password</h3>
                        <input type="password" placeholder="conpassword" id="conpasword"
name="conpassword">
                        <br>
                        <br><br>
                        <input type="submit" value="log-in" name="login">
                </form>

        </div>

</div>
</body>
</html>
```

login.html

```
<!DOCTYPE html>
<html lang="en" >
<head>
```

```
 <meta charset="UTF-8">
 <title>Login</title>
 <link href="https://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet">
 <link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet" integrity="sha384-wvfXpqpZZVQGK6TAh5PVlGOfQNHSoD2xbE+QkPxCAFlNEevoEH3Sl0sibVcOQVnN" crossorigin="anonymous">
 <link rel="stylesheet" href="../static/log style.css">

</head>
<body>
<!-- partial:index.partial.html -->
<div class="box-form">
	<div class="left">
		<div class="overlay">
		<h1>Hello World.</h1>
		<p>Exploratory Analysis of RainFall Data in India for Agriculture</p>
		<span>
			<p>login with social media</p>
			<a href="{{url_for('loginByGoogle')}}"><i class="fa fa-google" aria-hidden="true"></i> Login with google</a>
		</span>
		</div>
	</div>

		<div class="right" >
		<h5>Login</h5>
		<p>Don't have an account? <a href="{{url_for('signup')}}">Creat Your Account</a> it takes less than a minute</p>
		<form class="inputs" action="{{url_for('submit')}}" method="post">
			<input type="text" placeholder="user name" id="username" name="username">
			<br>
			<input type="password" placeholder="password" id="username" name="password">
```

```
                    <br>
                    <br><br>
                    <input type="submit" value="log-in" name="login">
                </form>

        </div>

</div>
</body>
</html>
```

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.3.1/css/bootstrap.min.css">
        <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></script>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.3.1/js/bootstrap.min.js"></script>
        <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
        <title>Rainfall Prediction App</title>
</head>
<body>
        <div class= "home">
                <div class="row">
                        <div class="col-md-3 left-side">
                                <h3>Welcome</h3>
```

```
<p>Here, You can Check Rain Prediction here!</p>
<table>
  <tr>
    <th>
      Team ID:
    </th>
    <th>
      PNT2022TMID10910
    </th>
  </tr>
  <tr>
    <th>
      Team leader:
    </th>
    <th>
      Priya Vishnu
    </th>
  </tr>
  <tr>
    <th>
      Team member 1:
    </th>
    <th>
      mahesh kumaar
    </th>
  </tr>
  <tr>
    <th>
      Team member 2:
    </th>
    <th>
      Suryaprakash
    </th>
  </tr>
  <tr>
```

```
      <th>
        Team member 3:
      </th>
      <th>
        vijaykumar
      </th>
    </tr>
</table>
                </div>
                <div class="col-md-9 right-side">
                        <form action="{{ url_for('predict')}}" method="post" >
                                <h1 class="heading">Rain Prediction for Agriculture</h1>
                                <div class="row right-form">
      <div class="col-md-6 my-2">
        <div class="md-form">
          <label for="MinTemp"> Minimum Temperature</label>
          <input type="text" class="form-control" id="MinTemp" name="MinTemp">
        </div>
      </div>
      <div class="col-md-6 my-2">
        <div class="md-form">
          <label for="MaxTemp">Maximum Temperature</label>
          <input type="text" class="form-control" id="MaxTemp" name="MaxTemp">
        </div>
      </div>
      <div class="col-md-6 my-2">
        <div class="md-form">
          <label for="Rainfall">Rainfall</label>
          <input type="text" class="form-control" id="Rainfall" name="Rainfall">
        </div>
      </div>
      <div class="col-md-6 my-2">
        <div class="md-form">
          <label for="Evaporation">Evaporation</label>
          <input type="text" class="form-control" id="Evaporation" name="Evaporation">
```

```
            </div>
          </div>
          <div class="col-md-6 my-2">
            <div class="md-form">
              <label for="Sunshine">Sunshine</label>
              <input type="text" class="form-control" id="Sunshine" name="Sunshine">
            </div>
          </div>
          <div class="col-md-6 my-2">
            <div class="md-form">
              <label for="WindGustSpeed">WindGustSpeed</label>
              <input type="text" class="form-control" id="WindGustSpeed"
name="WindGustSpeed">
            </div>
          </div>
          <div class="col-md-6 my-2">
            <div class="md-form">
              <label for="WindSpeed9am">WindSpeed at 9am</label>
              <input type="text" class="form-control" id="WindSpeed9am"
name="WindSpeed9am">
            </div>
          </div>
          <div class="col-md-6 my-2">
            <div class="md-form">
              <label for="WindSpeed3pm">WindSpeed at 3pm</label>
              <input type="text" class="form-control" id="WindSpeed3pm"
name="WindSpeed3pm">
            </div>
          </div>
          <div class="col-md-6 my-2">
            <div class="md-form">
              <label for="Humidity9am">Humidity at 9am</label>
              <input type="text" class="form-control" id="Humidity9am"
name="Humidity9am">
            </div>
```

```
            </div>
            <div class="col-md-6 my-2">
              <div class="md-form">
                <label for="Humidity3pm">Humidity at 3pm</label>
                <input type="text" class="form-control" id="Humidity3pm"
name="Humidity3pm">
              </div>
            </div>
            <div class="col-md-6 my-2">
              <div class="md-form">
                <label for="Pressure9am">Pressure at 9am</label>
                <input type="text" class="form-control" id="Pressure9am"
name="Pressure9am">
              </div>
            </div>
            <div class="col-md-6 my-2">
              <div class="md-form">
                <label for="Pressure3pm">Pressure at 3pm</label>
                <input type="text" class="form-control" id="Pressure3pm"
name="Pressure3pm">
              </div>
            </div>
            <div class="col-md-6 my-2">
              <div class="md-form">
                <label for="Cloud9am">Cloud 9am</label>
                <input type="text" class="form-control" id="Cloud9am" name="Cloud9am">
              </div>
            </div>
            <div class="col-md-6 my-2">
              <div class="md-form">
                <label for="Cloud3pm">Cloud 3pm</label>
                <input type="text" class="form-control" id="Cloud3pm" name="Cloud3pm">
              </div>
            </div>
            <div class="col-md-6 my-2">
```

```
                <div class="md-form">
                  <label for="Temp9am">Temperature at 9am</label>
                  <input type="text" class="form-control" id="Temp9am" name="Temp9am">
                </div>
              </div>
              <div class="col-md-6 my-2">
                <div class="md-form">
                  <label for="Temp3pm">Temperature at 3pm</label>
                  <input type="text" class="form-control" id="Temp3pm" name="Temp3pm">
                </div>
              </div>
              <div class="col-md-6 my-2">
                <div class="md-form">
                  <label for="RainToday" name="RainToday">RainToday</label>
                  <select class="form-control" id="RainToday" name="RainToday" aria-
label="RainToday">
                    <option class="hidden" selected disabled>Select RainToday</option>
                    <option value="Yes">Yes</option>
                    <option value="No">No</option>
                  </select>
                </div>
              </div>
              <div class="col-md-6 my-2">
                <div class="md-form">
                  <label for="year">Year</label>
                  <input type="text" class="form-control" id="year" name="year">
                </div>
              </div>
              <div class="col-md-6 my-2">
                <div class="md-form">
                  <label for="month">Month</label>
                  <input type="text" class="form-control" id="month" name="month">
                </div>
              </div>
              <div class="col-md-6 my-2">
```

```
                <div class="md-form">
                  <label for="day">Day</label>
                  <input type="text" class="form-control" id="day" name="day">
                </div>
              </div>
              <div class="col-md-6 my-2">
                <div class="md-form">
                  <label for="Location" name="Location">Location</label>
                  <select class="form-control" id="Location" name="Location" aria-label="Location">
                    <option class="hidden" selected disabled>Select Location</option>
                    <option value="Adelaide">Adelaide</option>
                    <option value="Albany">Albany</option>
                    <option value="Albury">Albury</option>
                    <option value="AliceSprings">AliceSprings</option>
                    <option value="BadgerysCreek">BadgerysCreek</option>
                    <option value="Ballarat">Ballarat</option>
                    <option value="Bendigo">Bendigo</option>
                    <option value="Brisbane">Brisbane</option>
                    <option value="Cairns">Cairns</option>
                    <option value="Canberra">Canberra</option>
                    <option value="Cobar">Cobar</option>
                    <option value="CoffsHarbour">CoffsHarbour</option>
                    <option value="Dartmoor">Dartmoor</option>
                    <option value="Darwin">Darwin</option>
                    <option value="GoldCoast">GoldCoast</option>
                    <option value="Hobart">Hobart</option>
                    <option value="Katherine">Katherine</option>
                    <option value="Launceston">Launceston</option>
                    <option value="Melbourne">Melbourne</option>
                    <option value="MelbourneAirport">MelbourneAirport</option>
                    <option value="Mildura">Mildura</option>
                    <option value="Moree">Moree</option>
                    <option value="MountGambier">MountGambier</option>
                    <option value="MountGinini">MountGinini</option>
```

```
                    <option value="Newcastle">Newcastle</option>
                    <option value="Nhil">Nhil</option>
                    <option value="NorahHead">NorahHead</option>
                    <option value="NorfolkIsland">NorfolkIsland</option>
                    <option value="Nuriootpa">Nuriootpa</option>
                    <option value="PearceRAAF">PearceRAAF</option>
                    <option value="Penrith">Penrith</option>
                    <option value="Perth">Perth</option>
                    <option value="PerthAirport">PerthAirport</option>
                    <option value="Portland">Portland</option>
                    <option value="Richmond">Richmond</option>
                    <option value="Sale">Sale</option>
                    <option value="SalmonGums">SalmonGums</option>
                    <option value="Sydney">Sydney</option>
                    <option value="SydneyAirport">SydneyAirport</option>
                    <option value="Townsville">Townsville</option>
                    <option value="Tuggeranong">Tuggeranong</option>
                    <option value="Uluru">Uluru</option>
                    <option value="WaggaWagga">WaggaWagga</option>
                    <option value="Walpole">Walpole</option>
                    <option value="Watsonia">Watsonia</option>
                    <option value="Williamtown">Williamtown</option>
                    <option value="Witchcliffe">Witchcliffe</option>
                    <option value="Wollongong">Wollongong</option>
                    <option value="Woomera">Woomera</option>
                  </select>
                </div>
              </div>
              <div class="col-md-6 my-2">
                <div class="md-form">
                  <label for="WindGustDir" name = "WindGustDir">WindGustDir</label>
                  <select class="form-control" id="WindGustDir" name="WindGustDir" aria-
label="WindGustDir">
                    <option class="hidden" selected disabled>Select WindGustDir</option>
                    <option value="ENE">ENE</option>
```

```
                    <option value="ESE">ESE</option>
                    <option value="E">E</option>
                    <option value="N">N</option>
                    <option value="NE">NE</option>
                    <option value="NNE">NNE</option>
                    <option value="NNW">NNW</option>
                    <option value="NW">NW</option>
                    <option value="S">S</option>
                    <option value="SE">SE</option>
                    <option value="SSE">SSE</option>
                    <option value="SSW">SSW</option>
                    <option value="SW">SW</option>
                    <option value="W">W</option>
                    <option value="WNW">WNW</option>
                    <option value="WSW">WSW</option>
                  </select>
                </div>
              </div>
              <div class="col-md-6 my-2">
                <div class="md-form">
                  <label for="WindDir9am" name = "WindDir9am">Wind Direction at 9am</label>
                  <select class="form-control" id="WindDir9am" name = "WindDir9am" aria-
label="Wind Direction at 9am">
                    <option class="hidden" selected disabled>Select Wind Direction at
9am</option>
                    <option value="ENE">ENE</option>
                    <option value="ESE">ESE</option>
                    <option value="E">E</option>
                    <option value="N">N</option>
                    <option value="NE">NE</option>
                    <option value="NNE">NNE</option>
                    <option value="NNW">NNW</option>
                    <option value="NW">NW</option>
                    <option value="S">S</option>
                    <option value="SE">SE</option>
```

```
                    <option value="SSE">SSE</option>
                    <option value="SSW">SSW</option>
                    <option value="SW">SW</option>
                    <option value="W">W</option>
                    <option value="WNW">WNW</option>
                    <option value="WSW">WSW</option>
                  </select>
                </div>
              </div>
              <div class="col-md-6 my-2">
                <div class="md-form">
                  <label for="WindDir3pm" name = "WindDir3pm">Wind Direction at 3pm</label>
                  <select class="form-control" id="WindDir3pm" name = "WindDir3pm" aria-
label="Wind Direction at 3pm">
                    <option class="hidden" selected disabled>Select Wind Direction at
3pm</option>
                    <option value="ENE">ENE</option>
                    <option value="ESE">ESE</option>
                    <option value="E">E</option>
                    <option value="N">N</option>
                    <option value="NE">NE</option>
                    <option value="NNE">NNE</option>
                    <option value="NNW">NNW</option>
                    <option value="NW">NW</option>
                    <option value="S">S</option>
                    <option value="SE">SE</option>
                    <option value="SSE">SSE</option>
                    <option value="SSW">SSW</option>
                    <option value="SW">SW</option>
                    <option value="W">W</option>
                    <option value="WNW">WNW</option>
                    <option value="WSW">WSW</option>
                  </select>
                </div>
              </div>
```

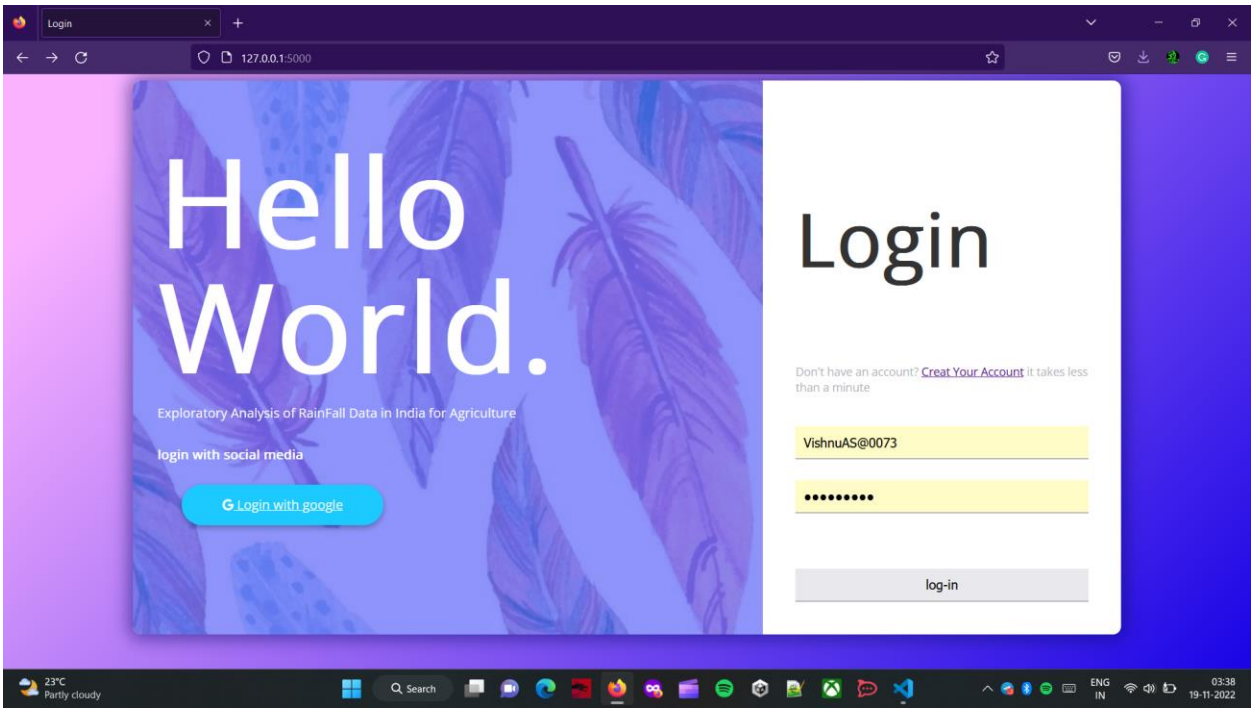
```
                <div class="col-md-6 my-2 d-flex align-items-end justify-content-around">
                  <button type="submit" class="btn btn-info button" style="margin-left: 109%; margin-top: 4%;">Predict</button>
                </div>
                                        </div>
                                    </form>
                            </div>
                    </div>

            </div>
</body>
</html>
```

result.html

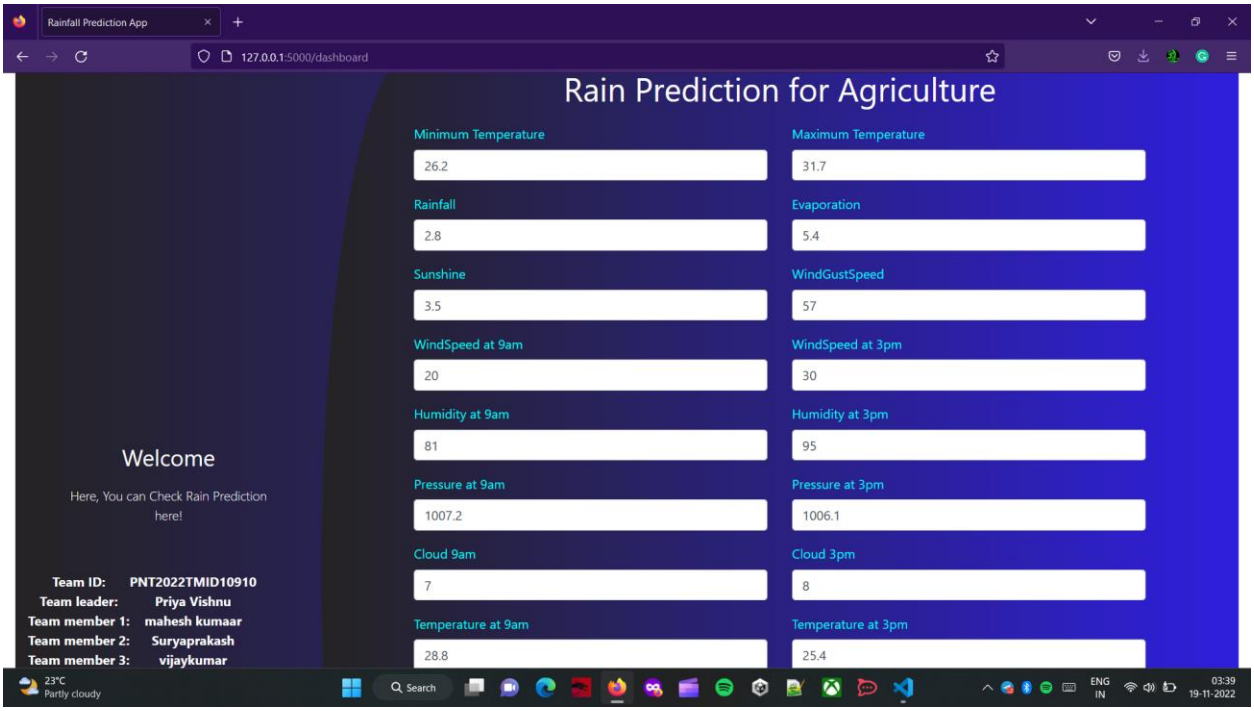
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.3.1/css/bootstrap.min.css">
        <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></script>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.3.1/js/bootstrap.min.js"></script>
        <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
  <title>Rainfall Prediction App</title>
</head>
<body>
  <!-- Website Title -->
```

```
            <div class="container home result-home" style="color: #fff;">
                <h2 class='container-heading'><span>Result - Rainfall Prediction</span></h2>
                    <div class='description'>
                    <p>A Machine Learning Web App, Built with Flask.</p>
                    </div>
                <!-- Result -->
                <div class="results">

        {% if my_prediction == 0 %}
          <div class="alert alert-warning" role="alert">
            <h2>Tomorrow will be no Rain fall</h2>
          </div>
        {% else %}
          <div class="alert alert-success" role="alert">
            <h2>Tomorrow will be Rain fall</h2>
          </div>
        {% endif %}
                </div>
      <h2><a href="{{ url_for('go_back')}}"> get back</a></h2>
            </div>


</body>
</html>
```

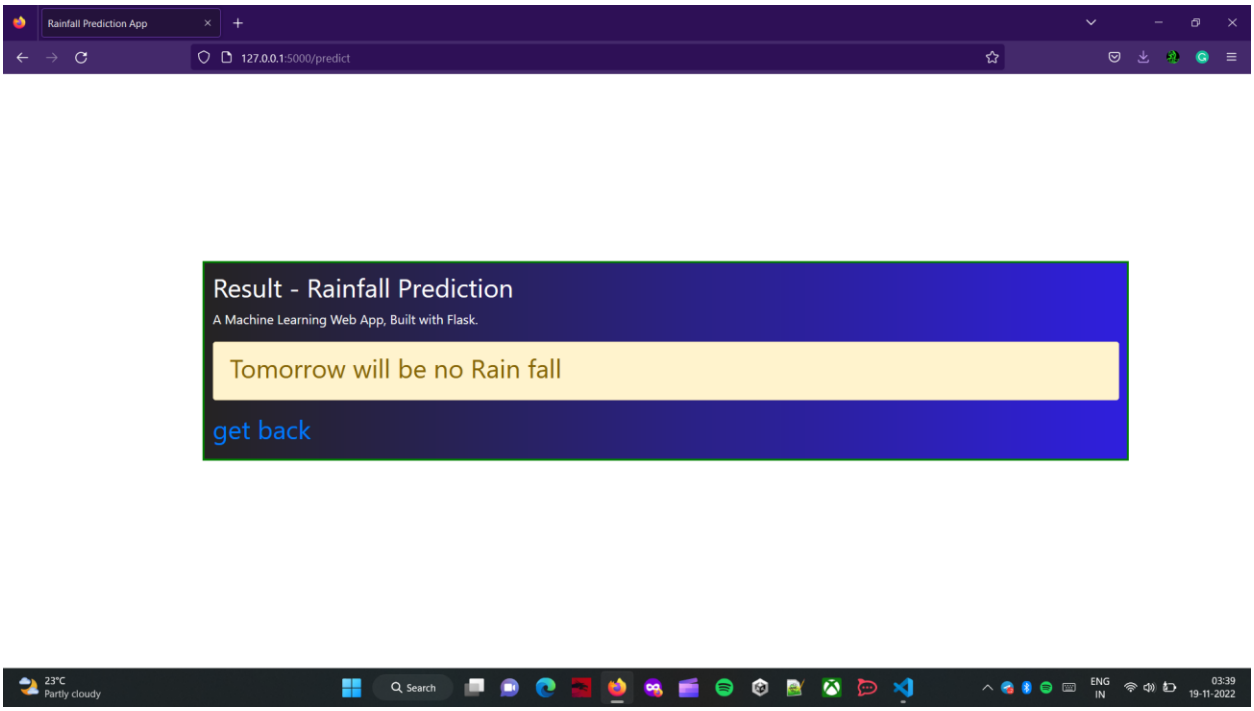
OUTPUT



Login page



main page



Result page

Github:

Repository link : https://github.com/IBM-EPBL/IBM-Project-8468-1658920292

Video:

Video link: https://youtu.be/8LzvSpvV2CM