Nanmudhalvan PROJECT REPORT ON

# RAINFALL PEDICTION

*In the partial fulfillment for the Award of Degree of*

**BACHELOR OF COMPUTER APPLICATIONS**

**Submitted By**


**NAME:SARANKUMAR K**

**Register No: 20122U09101**



**ARIGNAR ANNA GOVT COLLEGE**

**CHEYYAR**

**THIRUVANNAMALAI DISTRICT-604 407**

**DEPARTMENT OF COMPUTER APPLICATIONS**

## ABSTRACT

India is an agricultural country and its economy is largely based upon crop productivity and rainfall. For analyzing the crop productivity, rainfall prediction is require and necessary to all farmers. Rainfall Prediction is the application of science and technology to predict the state of the atmosphere. It is important to exactly determine the rainfall for effective use of water resources, crop productivity and pre planning of water structures. Using different data mining techniques it can predict rainfall. Data mining techniques are used to estimate the rainfall numerically. This paper focuses some of the popular data mining algorithms for rainfall prediction. Random Forest, K-Nearest Neighbor algorithm, Logistic regression, SVM, Decision Tree are some of the algorithms have been used. From that comparison, it can analyze which method gives better accuracy for rainfall prediction.

# INTRODUCTION

## 1.1 OBJECTIVE OF THE PROJECT:

The goal is to develop a machine learning model for Rainfall Prediction to potentially replace the updatable supervised machine learning classification models by predicting results in the form of best accuracy by comparing supervised algorithm.

### *1.1.1 Necessity:*

This prediction helps in predicting the rainfall and it helps in overcoming the crop productivity and to predict the state of atmosphere in agricultural countries. This Application is very easy to use. It can work accurately and very smoothly in a different scenario. It reduces the effort workload and increases efficiency in work. In aspects of time value, it is worthy.

### *1.1.2 Software development method:*

In many software applications program different methods and cases are followed such as, Waterfall model, Iterative model, Spiral model, V-model and Big Bang model. I used waterfall model in this application. I tried to use test case and case software approaches.

### *1.1.3 Layout of the document:*

This documentation starts with formal introduction. After introduction analysis and design of the project are described. In analysis and design of the project have many parts such as project proposal, mission, goal, target audience, environment. Use cases and test cases are in chapter 2 and chapter 3 respectively. Finally, this documentation finished with result and Conclusion part.

## 1.2 OVERVIEW OF THE DESIGNED PROJECT:

At first, we take the dataset from out resource then we have to perform data-preprocessing, visualization methods for cleaning and visualizing the dataset respectively and we applied the Machine Learning algorithms on the dataset then we generate the pickle file for best algorithm and flask is used as user interface for displaying the result.

# AIM AND SCOPE OF THE PRESENT INVESTIGATION

## 3.1 PROJECT PROPOSAL:

The project proposal is the term of documents. A project can describe the project proposal. It is the set of all plans of a project. Like, how the software works, what are the steps to complete the entire projects, and what are the software requirements and analysis for this project. In my project, I am doing all the steps and also risk and reward and other project dependencies in the project proposal.

### *3.1.1 Mission:*

An online Web based machine learning application is very popular and well known to everyone. Now a day's everybody wants to get it and work with it. Rainfall prediction is mostly useful for farmers in preplanning of water structure, crop productivity and it determines the rainfall for effective use of water. This simple method gives fast and accurate results in enhancing the rainfall.

### *3.1.2 Goal:*

The goal is to develop a machine learning model for predicting the rainfall.

## 3.2 SCOPE OF THE PROJECT:

The scope of this paper is to implement and investigate how different supervised binary classification methods impact default prediction. The model evaluation techniques used in this project are limited to precision, sensitivity, F1-score.

## 3.3 OVERVIEW OF THE PROJECT:

The overview of the project is to provide a web-based machine learning application to the user. Therefore, the user can directly know whether the rainfall is occur or not through our website over the internet.

## 3.4 EXISTING SYSTEM:

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

### *3.4.1 Disadvantages:*

The biggest disadvantage of this approach is that it fails when it comes for long term estimation.

## 3.5 PREPARING THE DATASET:

This dataset contains 145460 records of features extracted from kaggle, which were then classified into 2 classes:

- Rainy day
- Sunny day

## 3.6 PROPOSED SYSTEM:

### *3.6.1 Exploratory Data Analysis of Rainfall Prediction*

Multiple datasets from different sources would be combined to form a generalized dataset, and then different machine learning algorithms would be applied to extract patterns and to obtain results with maximum accuracy.

### *3.6.2 Data Wrangling*

In this section of the report will load in the data, check for cleanliness, and then trim and clean given dataset for analysis. Make sure that the

document steps carefully and justify for cleaning decisions.

### *3.6.3 Data collection*

The data set collected for predicting given data is split into Training set and Test set. Generally, we split the dataset into Training set and Test set. The Data Model which was created using machine learning algorithms are applied on the Training set and based on the test result accuracy, Test set prediction is done.

### *3.6.4 Building the classification model*

For predicting the rainfall, ML algorithm prediction model is effective because of the following reasons: It provides better results in classification problem.

- It is strong in preprocessing outliers, irrelevant variables, and a mix of continuous, categorical and discrete variables.
- It produces out of bag estimate error which has proven to be unbiased in many tests and it is relatively easy to tune with.

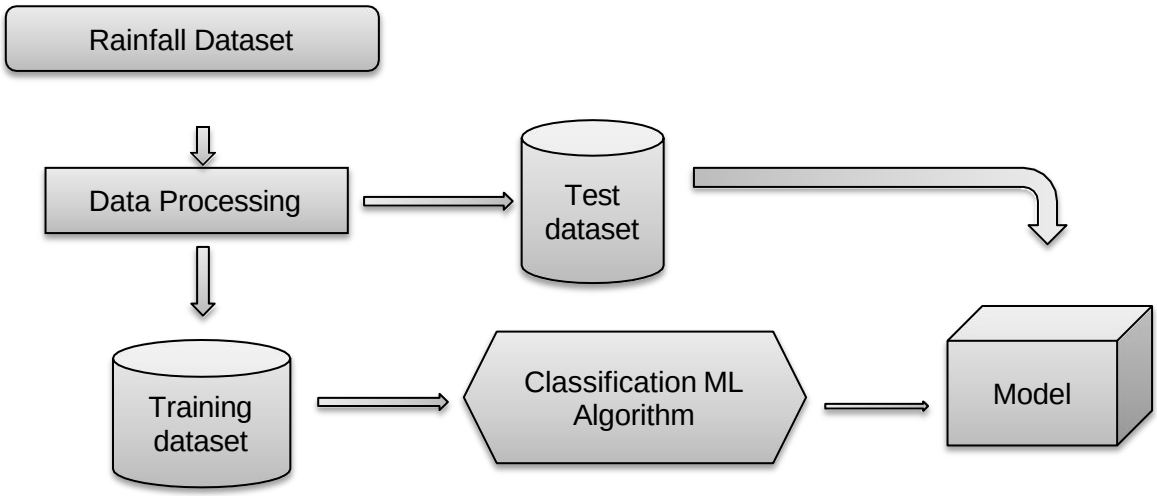



***Fig: 3.1: Architecture of Proposed model***

### *3.6.5 Advantages:*

- Performance and accuracy of the algorithms can be calculated and compared.
- Numerical Weather Prediction
- Statistical Weather Prediction
- Synoptic Weather Prediction

## 3.8 FLOW CHART:

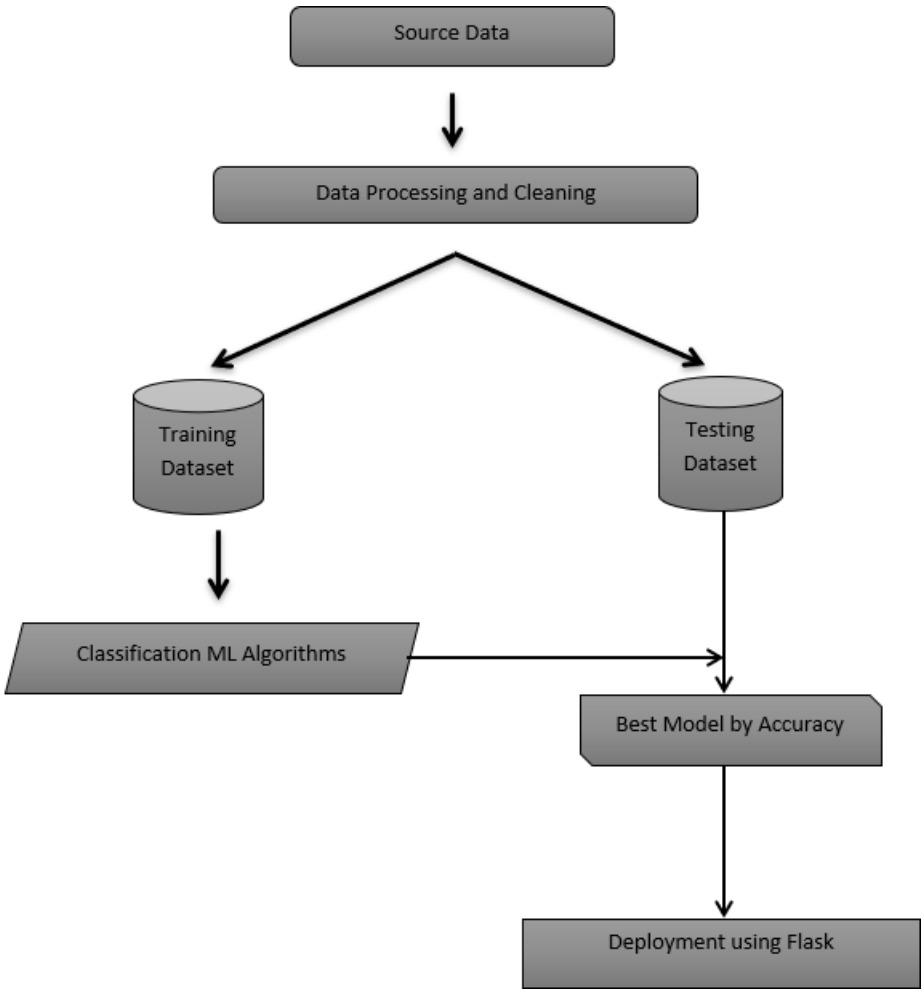



***Fig: 3.2: FLOW CHART***

# EXPERIMENTAL OR MATERIALS AND METHODS
# ALGORITHMS USED

## 4.1 SYSTEM STUDY:

To develop this model we use new modern technologies which are Machine Learning using Python for predicting and Flask is for user interface.

### *4.1.1 System requirement specifications:*

**a) Hardware requirements:**

- Processor : Intel
- RAM : 2GB
- Hard Disk : 80GB

**b) Software requirements:**

- OS : Windows
- Framework : Flask
- Technology : Machine Learning using Python
- Web Browser : Chrome, Microsoft Edge
- Code editor : Visual Studio Code, Google Colab, Anaconda or Jupyter notebook.

## 4.2 SYSTEM SPECIFICATIONS:

### *4.2.1 Machine Learning Overview:*

Machine learning is a field of study that looks at using computational algorithms to turn empirical data into usable models. The machine learning field grew out of traditional statistics and artificial intelligences communities. Through their business processes immense amounts of data have been and will be collected. This has provided an opportunity to re-invigorate the statistical and computational approaches to autogenerate useful models from data. Machine learning algorithms can be used to (a) gather understanding of the cyber phenomenon that produced the data under study, (b) abstract the understanding of

underlying phenomena in the form of a model, (c) predict future values of a phenomena using the above-generated model, and (d) detect anomalous behavior exhibited by a phenomenon under observation.

### *4.2.2 Flask Overview:*

Flask is an API of Python that allows us to build up web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple web-Application.

## 4.3 STEPS TO DOWNLOAD & INSTALL PYTHON:

Download the Latest version of the **Python** executable installer (https://www.python.org/downloads/). Watch the PIP list where pip is the package installer for python. Now upgrade the pip and setuptools using the command

| **Pip install --upgrade pip and Pip install --upgrade setuptools** |
|---|

### *4.3.1 IDE INSTALLATION FOR PYTHON*

IDE stands for Integrated Development Environment. It is a GUI (Graphical User Interface) where programmers write their code and produce the final products**.** Best IDE is Pycharm. So download the pycharm new version and install the software (https://www.jetbrains.com/pycharm/download/)

### *4.3.2 PYTHON FILE CREATION*

GO *To FILE MENU > CREATE > NEW > PYTHON FILE >(Name Your Python File as "HOUSE PRICE PPREDICTION" > SAVE*

## 4.4 PYTHON LIBRARIES NEEDED

There are many libraries in python. In those we only use few main libraries needed

### *4.4.1 NUMPY LIBRARY*

**NumPy** is an open-source numerical Python library. NumPy contains a multi- dimensional array and matrix data structures. It can be utilized to perform a number of mathematical operations on arrays such as trigonometric, statistical, and algebraic routines like mean, mode, standard deviation etc…,

Installation- **(https://numpy.org/install/)**

```
pip install NUMPY
```

Here we mainly use array, to find mean and standard deviation.

### *4.4.2 PANDAS LIBRARY*

**Pandas** is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the DataFrame. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables. There are several ways to create a DataFrame.

**Installation- (https://pandas.pydata.org/getting_started.html)**

```
pip install PANDAS
```

Here we use pandas for reading the csv files, for grouping the data, for cleaning the data using some operations.

### *4.4.3 MATPLOTLIB LIBRARY*

**Matplotlib** is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Use interactive figures that can zoom, pan, update, visualize etc..,

**Installation- (https://matplotlib.org/users/installing.html)**

```
pip install Matplotlib
```

Here we use pyplot mainly for plotting graphs.

**matplotlib**.**pyplot** is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

### *4.4.4 SEABRON LIBRARY*

**Seaborn** package was developed based on the Matplotlib library. It is used to create more attractive and informative statistical graphics. While seaborn is a different package, it can also be used to develop the attractiveness of matplotlib graphics.

**Installation-(https://seaborn.pydata.org/installing.html)**

```
pip install Seaborn
```

### *4.4.5 SCIKIT-LEARN LIBRARY*

**Scikit-learn** is a free machine learning library for the Python. It features various algorithms like support vector machine, random forests, regression and k-neighbors, and it also supports Python numerical and scientific libraries like NumPy and SciPy.

```
Pip install Scikit-Learn
```

**Installation-(https://scikit-learn.org/stable/install.html)**

Here use scikit-learn's regression methods for prediction purpose.

### *4.4.6 FLASK*

Flask is an API of Python that allows us to build up web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's framework and is also easier to learn because it

```
pip install flask
```

has less base code to implement a simple web-Application.

Here we use flask for the user-interface.

**4.5 MODULES:**

A modular design reduces complexity, facilities change (a critical aspect of software maintainability), and results in easier implementation by encouraging parallel development of different part of system. Software with effective modularity is easier to develop because function may be compartmentalized and interfaces are simplified. Software architecture embodies modularity that is software is divided into separately named and addressable components called modules that are integrated to satisfy problem requirements.

Modularity is the single attribute of software that allows a program to be intellectually manageable. The five important criteria that enable us to evaluate a design method with respect to its ability to define an effective modular design are: Modular decomposability, Modular Comps ability, Modular Understand ability, Modular continuity, Modular Protection.

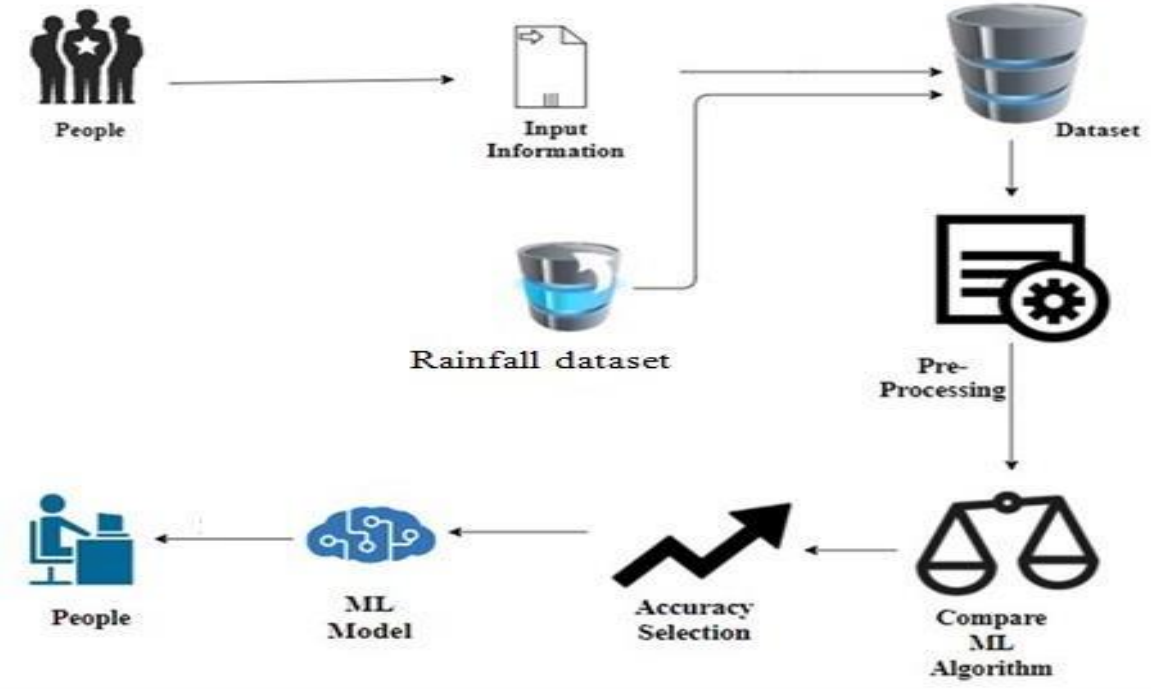



***Fig: 4.1: SYSTEM ARCHITECTURE***

### *4.7.5 ALGORITHM AND TECHNIQUES*

**Algorithm Explanation**

In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc. In Supervised Learning, algorithms learn from labeled data. After understanding the data, the algorithm determines which label should be given to new data based on pattern and associating the patterns to the unlabeled new data.

**Used Python Packages:**

**sklearn:**

- In python, sklearn is a machine learning package which include a lot of ML algorithms.

- Here, we are using some of its modules like train_test_split, DecisionTreeClassifier or Logistic Regression and accuracy_score.

**NumPy:**

- It is a numeric python module which provides fast maths functions for calculations.
- It is used to read data in numpy arrays and for manipulation purpose.

**Pandas:**

- Used to read and write different files.
- Data manipulation can be done easily with data frames.

**Matplotlib:**

- Data visualization is a useful way to help with identify the patterns from given dataset.
- Data manipulation can be done easily with data frames.

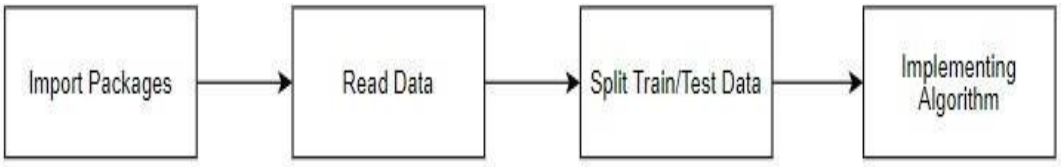
**Logistic Regression:**

It is a statistical method for analyzing a data set in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables. Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.).

In other words, the logistic regression model predicts P(Y=1) as a function of X.
Logistic regression Assumptions:

- Binary logistic regression requires the dependent variable to be binary.
- For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome.
- Only the meaningful variables should be included.
- The independent variables should be independent of each other. That is, the model should have little.
- The independent variables are linearly related to the log odds.
- Logistic regression requires quite large sample sizes.

MODULE DIAGRAM



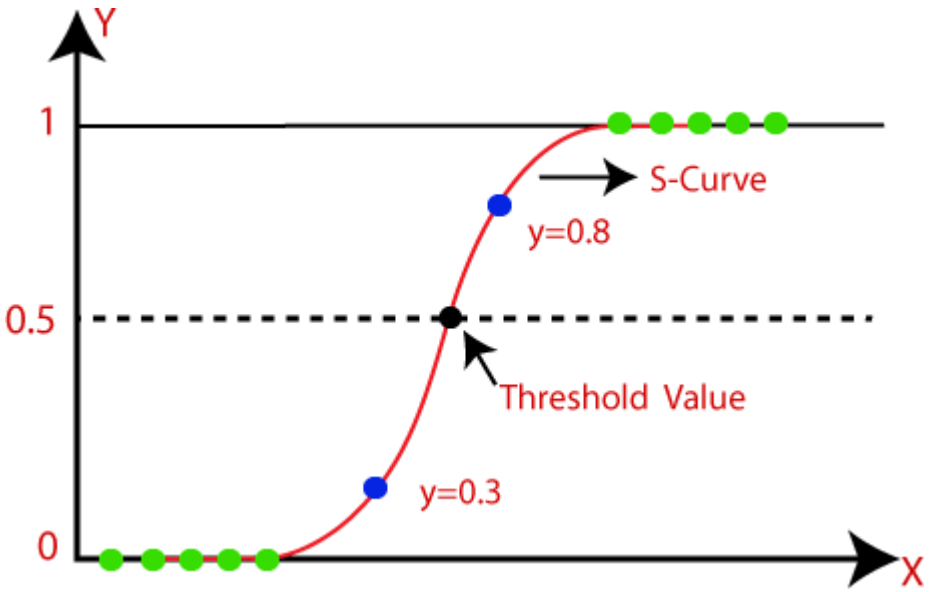




***Fig: 4.7: LOGISTIC REGRESSION***

GIVEN INPUT EXPECT OUTPUT

input: data

output: getting accuracy

**Random Forest Classifier:**

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of over fitting to their training set. Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision *trees,* resulting in a *forest of trees,* hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

The following are the basic steps involved in performing the random forest algorithm:

- Pick N random records from the dataset.
- Build a decision tree based on these N records.
- Choose the number of trees you want in your algorithm and repeat steps 1 and 2.

In case of a regression problem, for a new record, each tree in the forest predicts a value for Y (output). The final value can be calculated by taking the average of all the values predicted by all the trees in forest. Or, in case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

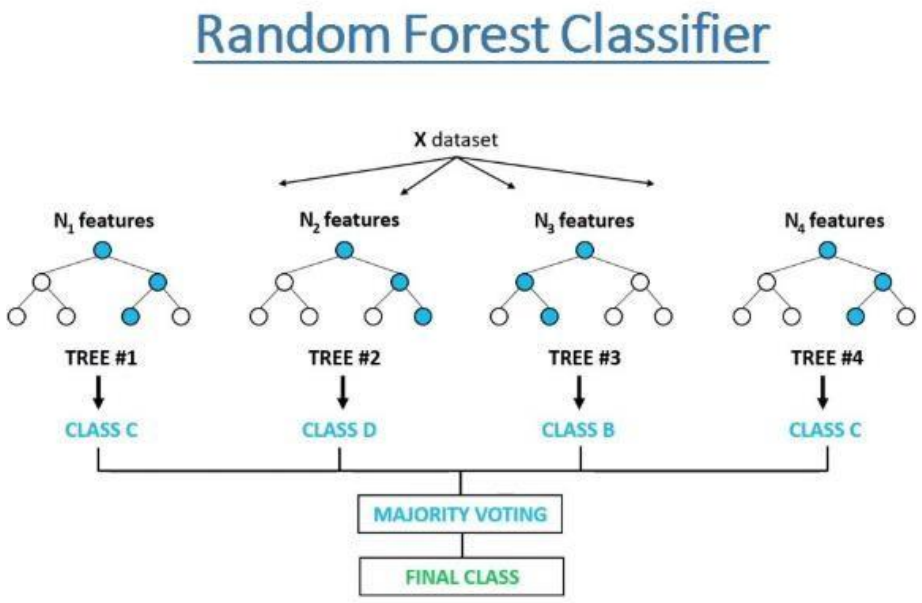



***Fig: 4.8: RANDOM FOREST CLASSIFIER***

GIVEN INPUT EXPECT OUTPUT

input: data

output: getting accuracy

**Decision Tree :**

It is one of the most powerful and popular algorithm. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables. Assumptions of Decision tree:

- At the beginning, we consider the whole training set as the root.
- Attributes are assumed to be categorical for information gain, attributes are assumed to be continuous.
- On the basis of attribute values records are distributed recursively.
- We use statistical methods for ordering attributes as root or internal node.

Decision tree builds classification or regression models in the form of a tree

structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. A decision node has two or more branches and a leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data. Decision tree builds classification or regression models in the form of a tree structure. It utilizes an if-then rule set which is mutually exclusive and exhaustive for classification. The rules are learned sequentially using the training data one at a time. Each time a rule is learned, the tuples covered by the rules are removed. This process is continued on the training set until meeting a termination condition. It is constructed in a top-down recursive divide-and-conquer manner. All the attributes should be categorical. Otherwise, they should be discretized in advance. Attributes in the top of the tree have more impact towards in the classification and they are identified using the information gain concept. A decision tree can be easily over-fitted generating too many branches and may reflect anomalies due to noise or outliers.

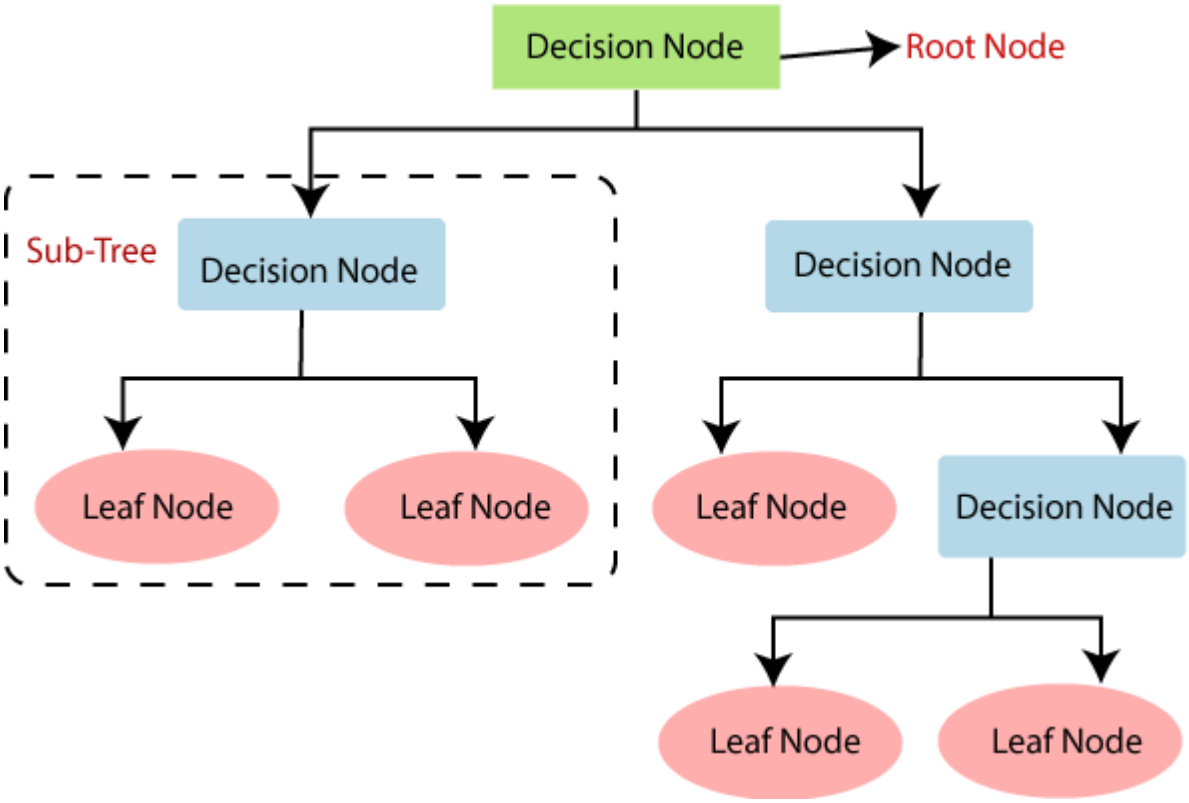



***Fig: 4.9: DECISION TREE CLASSIFIER***

**K-Nearest Neighbor**

K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique. It assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. It stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K-NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.

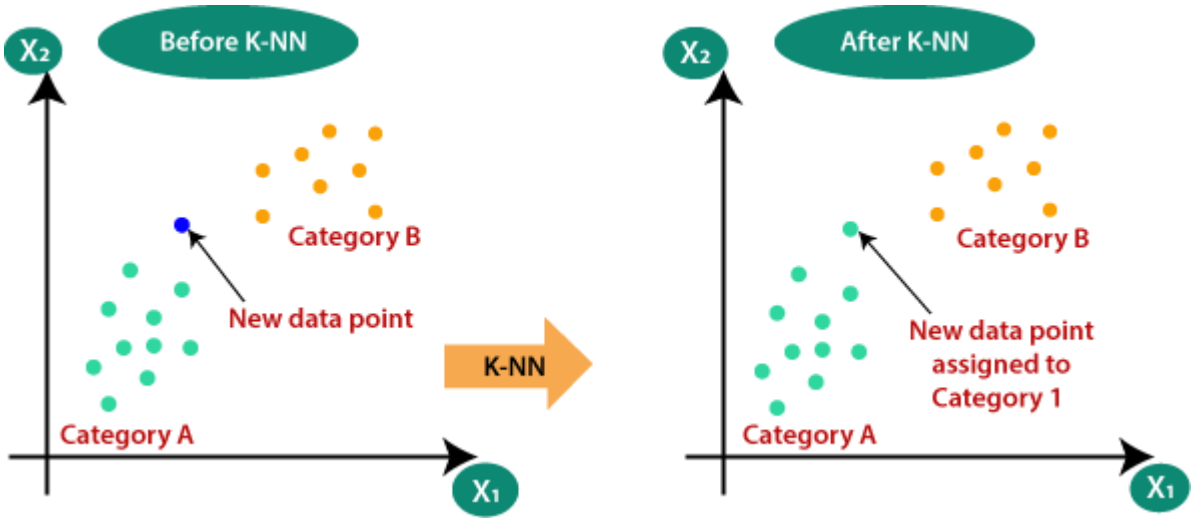



***Fig: 4.10: K-NEAREST NEIGHBOR***

**Support Vector Machine**

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. Primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes. So that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

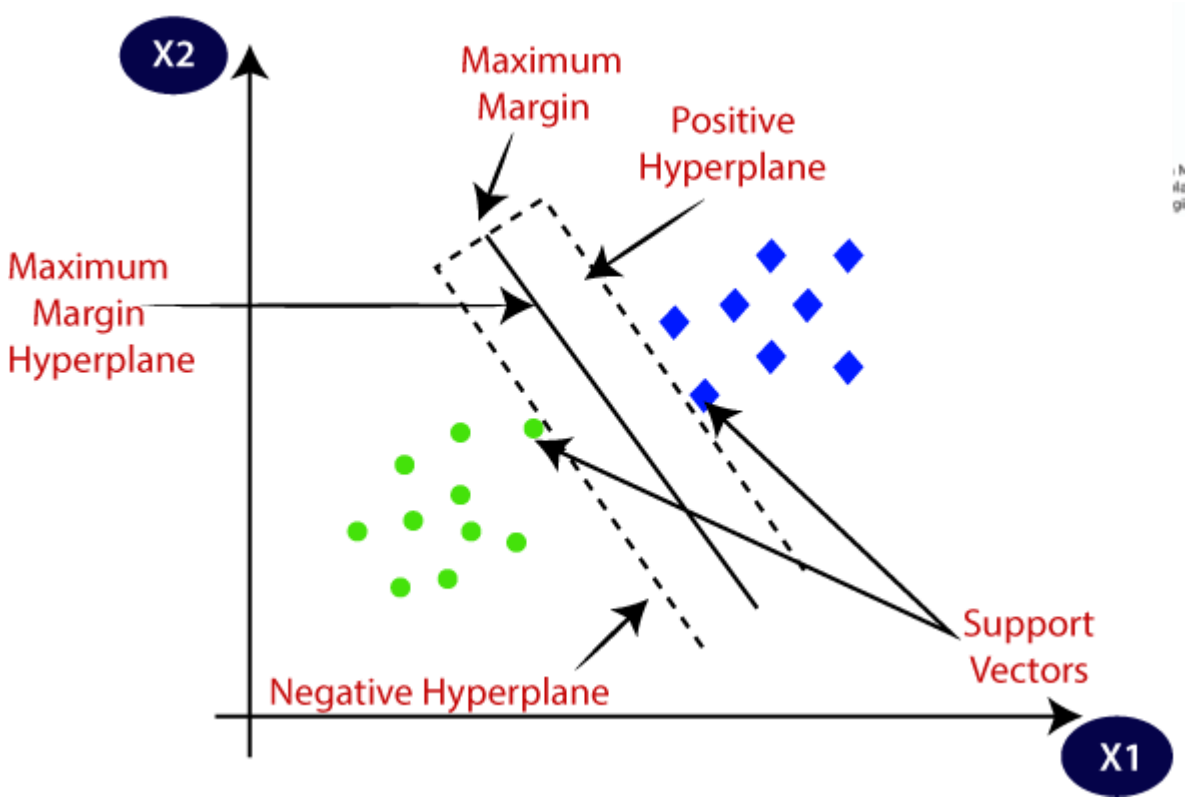



***Fig: 4.11: SUPPORT VECTOR MACHINE***

### *4.7.6 Deployment Using Flask (Web Framework):*

Flask is a micro web framework written in Python. It is classified as a micro-framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself.

Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

Flask was created by Armin Ronacher of Pocoo, an international group of Python enthusiasts formed in 2004. According to Ronacher, the idea was originally an April Fool's joke that was popular enough to make into a serious

application. The name is a play on the earlier Bottle framework.

When Ronacher and Georg Brand created a bulletin board system written in Python, the Pocoo projects Werkzeug and Jinja were developed.

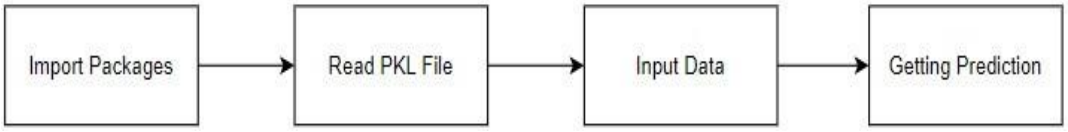
In April 2016, the Pocoo team was disbanded and development of Flask and related libraries passed to the newly formed Pallets project.

Flask has become popular among Python enthusiasts. As of October 2020, it has second most stars on GitHub among Python web-development frameworks, only slightly behind Django, and was voted the most popular web framework in the Python Developers Survey 2018.

The micro-framework Flask is part of the Pallets Projects, and based on several others of them.

**Flask is** based on Werkzeug, Jinja2 and inspired by Sinatra Ruby framework, available under BSD licence. It was developed at pocoo by Armin Ronacher. Although Flask is rather young compared to most Python frameworks, it holds a great promise and has already gained popularity among Python web developers. Let’s take a closer look into Flask, so-called “micro” framework for Python.

MODULE DIAGRAM




GIVEN INPUT EXPECTED OUTPUT

input : data values

output : predicting output

# RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

## 5.1 PERFORMANCE ANALYSIS:

Website performance optimization, the focal point of technologically superior website designs is the primary factor dictating the Rainfall occurred or not. After all, unimpressive website performance kills admission process when the torture of waiting for slow Web pages to load frustrates visitors into seeking alternatives – impatience is a digital virtue! And also the ml algorithms used in our project will give the best accurate result to the user for Rainfall prediction

We created the following six chapter in-depth speed optimization guide to show you how important it is to have a fast loading, snappy website! Countless research papers and benchmarks prove that optimizing your sites' speed is one of the most affordable and highest ROI providing investments!

Lightning-fast page load speed amplifies visitor engagement, retention, and boosts sales. Instantaneous website response leads to higher conversion rates, and every 1 second delay in page load decreases customer satisfaction by 16 percent, page views by 11 percent and conversion rates by 7 percent according to a recent Aberdeen Group research.

| Algorithm | Accuracy |
|---|---|
| **Logistic Regression** | **0.83** |
| **RandomForest** | **0.86** |
| **KNeighborsClassifier** | **0.84** |
| **DecisionTreeClassifier** | **0.78** |

**Table 5.1: Algorithms Accuracy**

## 5.2 DISCUSSION:

While discussions provide avenues for exploration and discovery, leading a discussion can be anxiety-producing: discussions are, by their nature, unpredictable, and require us as instructors to surrender a certain degree of control over the flow of information. Fortunately, careful planning can help us ensure that discussions are lively without being chaotic and exploratory without losing focus. When planning a discussion, it is helpful to consider not only cognitive, but also social/emotional, and physical factors that can either foster or inhibit the productive exchange of ideas.

# SUMMARY AND CONCLUSION

## 6.1 SUMMARY:

This project objective is to predict the Rainfall. So this online Rainfall prediction system will helps the farmers to analyzing the crop productivity, preplanning the water structure and estimate the rainy or not.

## 6.2 CONCLUSION:

The analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation. The best accuracy on public test set is higher accuracy score will be find out. This application can helps in predicting the Rainfall.

## 6.3 FUTURE WORK:

- Rainfall prediction to connect with cloud.
- To optimize the work to implement in Artificial Intelligence environment.

GITHUB LINK: https://github.com/Sarankumar123456/NANMUDHALVAN-PROJECT-SARANKUMAR.K.git

**REFERENCES:**

[1] singh, p., 2018. indian summer monsoon rainfall (ismr) forecasting using time series data: a fuzzy-entropy-neuro based expert system. geoscience frontiers, 9(4), pp.1243-1257.

[2] cramer, s., kampouridis, m., freitas, a. and alexandridis, a., 2017. an extensive evaluation of seven machine learning methods for rainfall prediction in weather derivatives. expert systems with applications, 85, pp.169-181.

[3] pour, s., shahid, s. and chung, e., 2016. a hybrid model for statistical downscaling of daily rainfall. procedia engineering, 154, pp.1424-1430.

[4] manjunath n, muralidhar b r, sachin kumar s, vamshi k and savitha p, 2021. rainfall prediction using machine learning and deep learning techniques. [online] irjet.net. available at: <https://www.irjet.net/archives/v8/i8/irjet-v8i850.pdf> [accessed 20 january 2022].

[5] tanvi patil and dr. kamal shah, 2021. weather forecasting analysis using linear and logistic regression algorithm. [online] irjet.net. available at: <https://www.irjet.net/archives/v8/i6/irjet-v8i6454.pdf> [accessed 20 january 2022].

[6] n. divya prabha and p. radha, 2019. prediction of weather and rainfall forecasting using classification techniques. [online] irjet.net. available at: <https://www.irjet.net/archives/v6/i2/irjet-v6i2154.pdf> [accessed 20 january 2022].

[7] waghmare, d., 2021. machine learning technique for rainfall prediction. international journal for research in applied science and engineering technology, 9(vi), pp.594-600.

[8] yashasathreya, vaishalibv, sagark and srinidhihr, 2021. flood prediction and rainfall analysis using machine learning. [online] irjet.net.available at: <https://www.irjet.net/archives/v8/i7/irjet-v8i7432.pdf> [accessed 20 january 2022].

## APPENDIX:

## A. SOURCE CODE:

### Jupiter Note Book

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
dataset=pd.read_csv("weatherAUS.csv")
pd.set_option('display.max_columns',70)
```

```
dataset.shape

dataset.isnull().sum()

#missing value percentage
round(dataset.isnull().sum()*100/len(dataset),2)
```

```
dataset.dtypes

for i in dataset.columns:
    print(i,"=" ,len(dataset[i].unique()))

dataset["RainToday"].value_counts()
```

```
for i in dataset.columns:
    if dataset[i].dtypes=="float64":
        sns.histplot(dataset[i])
        plt.show()

plt.figure(figsize=(14,9))
sns.heatmap(dataset.corr(),annot=True)

sns.barplot(x=dataset["year"],y=dataset["Rainfall"])
sns.barplot(x=dataset["month"],y=dataset["Rainfall"])

plt.figure(figsize=(12,8))
sns.countplot(x=dataset["month"],hue=dataset["RainToday"])
```

```
pd.crosstab(dataset["RainTomorrow"],dataset["RainToday"])

#replace Na in numerical columns with mean for columns with Na ratio higher
than 3%:
dataset['WindGustSpeed'].fillna(np.mean(dataset['WindGustSpeed'].dropna().valu
es), inplace = True)
dataset['Pressure9am'].fillna(np.mean(dataset['Pressure9am'].dropna().values),
inplace = True)
dataset['Pressure3pm'].fillna(np.mean(dataset['Pressure3pm'].dropna().values),
inplace = True)
dataset['WindGustDir']= dataset['WindGustDir'].fillna('Unknown')
dataset['WindDir9am']= dataset['WindDir9am'].fillna('Unknown')

from sklearn import preprocessing
a=preprocessing.normalize(dataset[names])
len(a)

d = preprocessing.normalize(scale_feat)
scaled_df = pd.DataFrame(d, columns=names)
scaled_df
```

```
from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier()

from sklearn.model_selection import train_test_split as tts
x_train,x_test,y_train,y_test=tts(x,y,test_size=0.2,random_state=100)

from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
from sklearn.metrics import confusion_matrix

def result(algo):
    var=algo
    var.fit(x_train,y_train)
    y_pred=var.predict(x_test)
    return accuracy_score(y_pred,y_test) , f1_score(y_pred,y_test)
,confusion_matrix(y_pred,y_test)

result(RandomForestClassifier())
result(DecisionTreeClassifier()
```

```
from flask import Flask,render_template,url_for,request,jsonify
from flask_cors import cross_origin
import pandas as pd
import numpy as np
import datetime
import pickle
import sklearn

app = Flask(__name__, template_folder="templates")
app.static_folder="static"

model = pickle.load(open("rain_model2.sav", 'rb'))
print("Model Loaded")

@app.route("/",methods=['GET'])
@cross_origin()
def home():
    return render_template("index.html")

@app.route("/predict",methods=['GET','POST'])
@cross_origin()
def predict():
    if request.method == "POST":

        minTemp = float(request.form['mintemp'])
        maxTemp = float(request.form['maxtemp'])
        windGustSpeed = float(request.form['windgustspeed'])

        input_cat = [rainToday,location , month ]
        input_num=[humidity3pm,pressure3pm,windGustSpeed,humidity9am,pressure9
am,temp3pm,minTemp
                   ,maxTemp,temp9am,windSpeed3pm,windSpeed9am,rainfall]
        norm_arr=sklearn.preprocessing.normalize([input_num])

        arr1=np.insert(norm_arr,2,rainToday)
        arr2=np.insert(arr1,9,location)
        arr3=np.insert(arr2,13,month)

        pred = model.predict([arr3])
        output = pred
        if output == 0:
            return render_template("sunny.html")
        else:
            return render_template("rainy.html")
    return render_template("predict.html")
```

**INDEX.HTML**

```
<!DOCTYPE html>
<html lang="es" dir="ltr">
<head>
    <title>Rainfall Prediction</title>
</head>
<body>
  <div class="photo"></div>
    <section class="text-gray-600 body-font">
  <div class="container mx-auto flex px-5 py-15 items-center justify-center
flex-col">
    <img class="lg:w-2/6 md:w-3/6 w-5/6 mb-10 object-cover object-center
rounded-xl" alt="img" src="{{ url_for('static', filename = 'umb.jpg') }}" >
    <div class="text-center lg:w-2/3 w-full">
      <h1 class="title-font sm:text-4xl text-3xl mb-4 font-medium text-gray-
900">Will it rain tomorrow?</h1>
      <p class="mb-8 leading-relaxed sm:text-2xl">Stay Ahead of time and find
out now with ML powered predictions </p>
      <div class="flex justify-center">
      <a href="/predict"> <button class="inline-flex text-white bg-indigo-500
border-0 py-2 px-6 focus:outline-none hover:bg-indigo-600 rounded text-
lg">Predict</button></a> </div> </div> </div> </section> </body> </html>
```

**Predication.HTML**

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href=
{{url_for('static',filename='predictor.css')}}>
    <title>Rain Prediction</title>
</head>
<body>
    <section id="prediction-form">
    </br>
    <div class="col-md-6 my-2 d-flex align-items-end justify-content-around">
       <a href="./"> <button type="submit" class="btn btn-info button"
style="margin-right: 100%;">Back</button></a>
    </div>
        <form class="form" action="/predict", method="POST">
            <h1 class="my-3 text-center">Predictor</h1>
            <div class="row">
                <div class="col-md-6 my-2">
                    <div class="md-form">
        <label for="mintemp" class="mintemp"> Minimum temprature (°C)</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="temp9am" class="temp9am">Temperature 9am
(°C)</label>
                        <input type="text" class="form-control" id="temp9am"
name="temp9am">
                    </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:
90%;">Predict</button>
        </div> </div> </form>    </section> <div>
 <h1><center> {{ prediction }} </center></h1> </div> </body> </html>
```

**Rainy DAY**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>rainy</title>
</head>
<body>
    <div class="bg-image"></div>
    <div class="bg-text">
      <div class="col-md-6 my-2 d-flex align-items-end justify-content-
around"> <a href="/predict"> <button type="submit" class="btn btn-info button"
style="margin-right: 100%;">Back</button></a>
      </div>
      <h1>TOMORROW WILL BE A RAINY DAY</h1>
      <p>MAKE SURE TO CARRY AN UMBRELLA WHEN YOU GO OUT </p>
    </div>    </body> </html>
```

**Sunny DAY**

```
<html lang="en">
<head> <title> Sunny Day </title> </head>
<body>
    <div class="bg-image"></div>
    <div class="bg-text">
      <div class="col-md-6 my-2 d-flex align-items-end justify-content-
around"> <a href="/predict"> <button type="submit" class="btn btn-info button"
style="margin-right: 100%;">Back</button></a>
      </div>
      <h1>TOMORROW WILL BE A Sunny DAY</h1>
      <p> GO OUT AND MAKE THE BEST OF IT </p>
    </div>    </body> </html>
```

## B. SCREENSHOTS

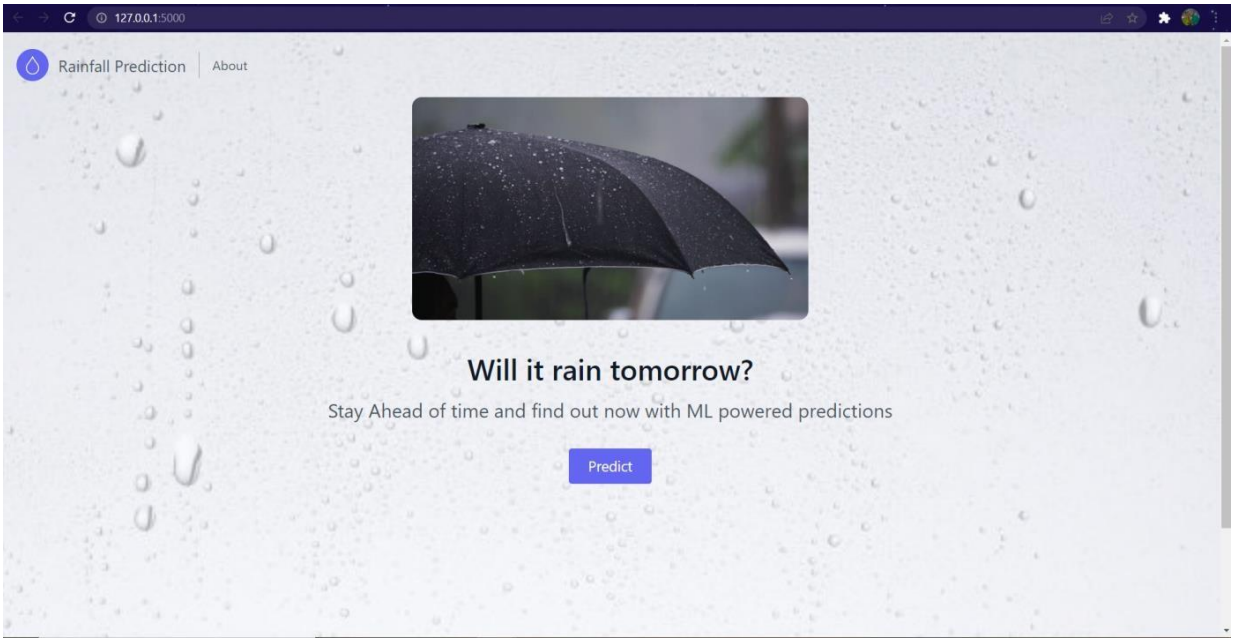



***Fig: 6.1: HOME PAGE***

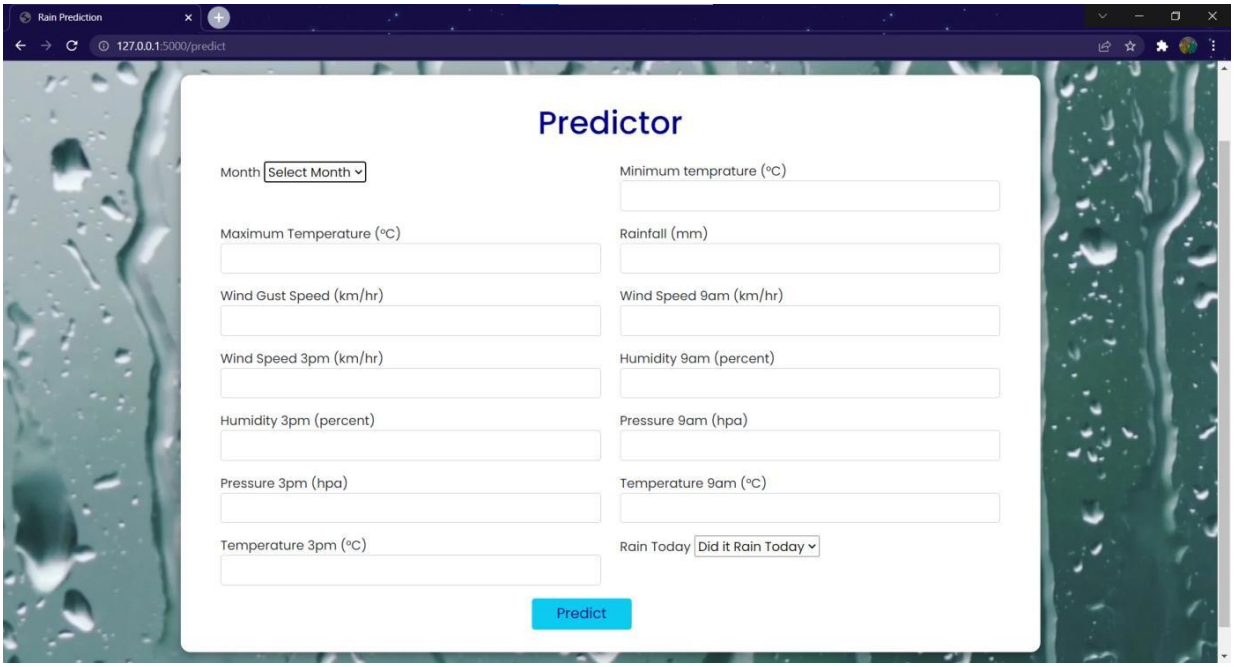



***Fig: 6.2: INPUTS PAGE***

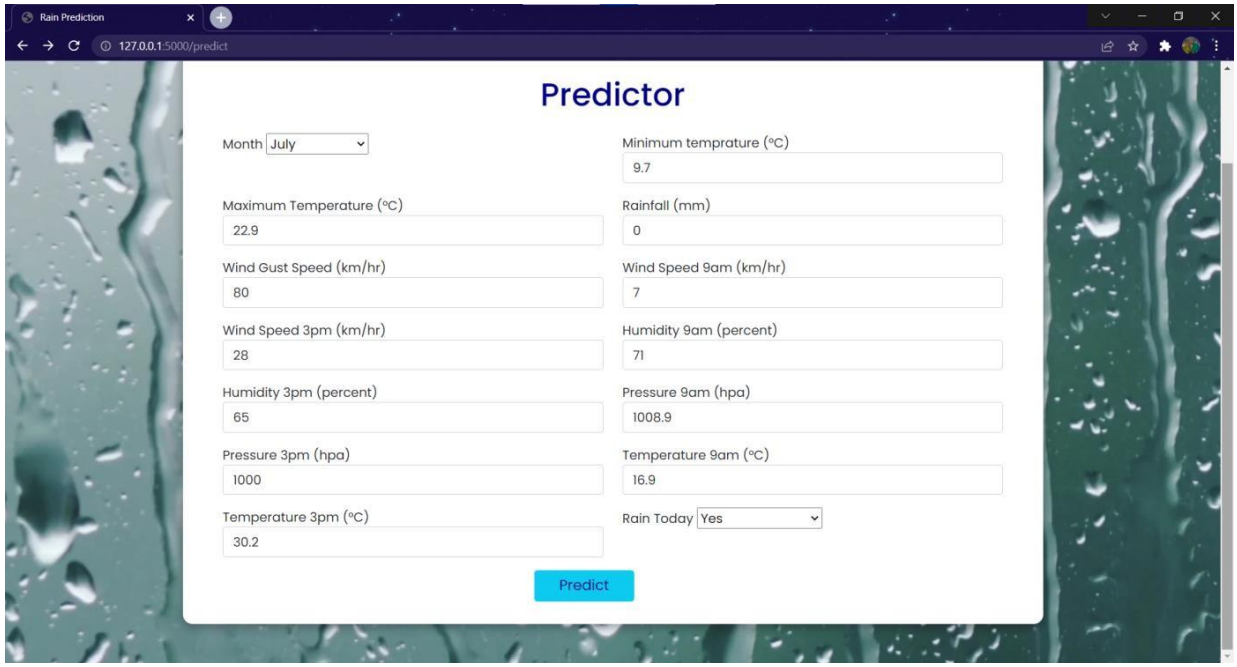



***Fig: 6.3: INPUTS GIVEN BY THE USER***

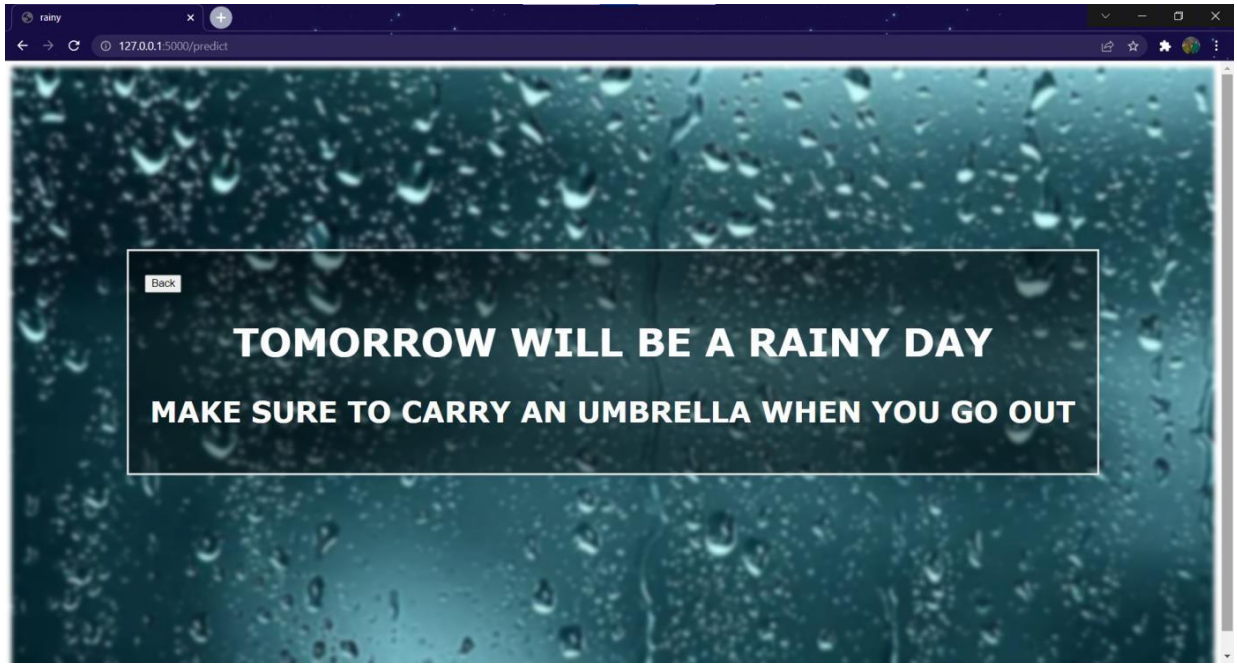



***Fig: 6.4: Rainy Day***

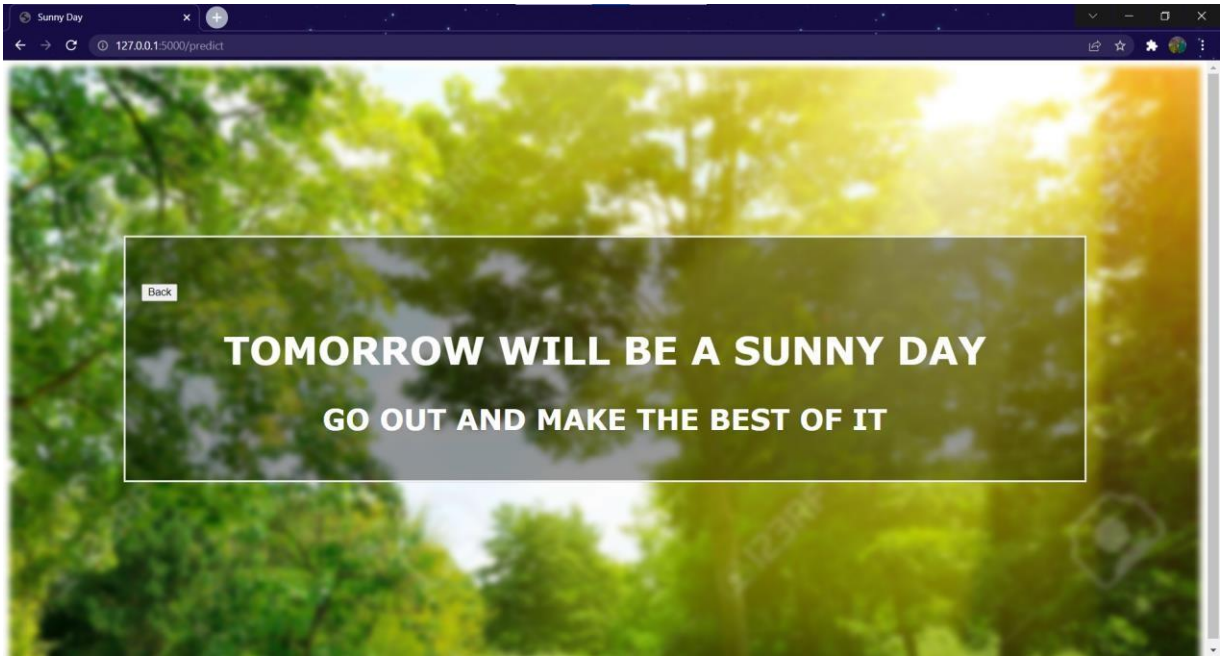



***Fig: 6.5: Sunny Day***