

**JAYAPRAKASH NARAYAN COLLEGE OF ENGINEERING**  
**DHARMAPUR, MAHABOONNAGAR – 509001**

AFFILIATED TO JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY,  
KUKATPALLY, HYDERABAD – 500072, Telangana, India.

**“RAINFALL PREDICTION”**



A Dissertation on Major Project submitted to the Jawaharlal Nehru Technological University, Hyderabad in partial fulfillment of the requirement for the award of degree of

**BACHELOR OF TECHNOLOGY**  
**IN**  
**COMPUTER SCIENCE AND ENGINEERING**

**Submitted By**

**M.SRIVIBHA**  
**( 16361A0597 )**

**Under the Guidance of**

**Mrs. A.SWATHI**  
**Asst. Professor**

**MAY 2020**

**Department of Computer Science and Engineering**



**JAYAPRAKASH NARAYAN COLLEGE OF ENGINEERING**  
**DHARMAPUR, MAHABOONNAGAR – 509001**

Web : [www.jpnce.ac.in](http://www.jpnce.ac.in), Phone : 8886680021

**2016-2020**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**JAYAPRAKASH NARAYAN COLLEGE OF ENGINEERING**

**DHARMAPUR, MAHABOONAGAR – 509001**

**(Affiliated to J.N.T.U.H., Approved by A.I.C.T.E)**



## **CERTIFICATE**

This is to Certify that the Major Project report on “**RAINFALL PREDICTION**” is a bonafide work done by M.SRIVIBHA ( 16361A0597 ), in partial fulfillment of the requirement of the award for the degree of Bachelor of Technology in “**Computer Science and Engineering**” J.N.T.U., Hyderabad during the year 2019-2020.

**Project Guide**

**H.O.D.**

**Mrs. A.SWATHI**

Asst. Professor,  
Dept of C.S.E.

**Dr. K.Guru Raghavendra Reddy**

Asst.Professor & Head,  
Dept of C.S.E.

**External Examiner :**

## ACKNOWLEDGEMENT

I owe a debt of gratitude to **Mrs A. SWATHI, Asst.Professor**, CSE, JPNCE for her admirable guidance and inspirational both theoretically and practically and most importantly for the drive to complete projects successfully. Working under such an eminent guide was my privilege.

I express my sincere thanks to **Dr.K.Guru Raghavendra Reddy, Asst.Professor** and HOD, CSE, JPNCE of all kinds of consideration, support and encouragement in carrying out this project successfully.

I would also like to thank **Dr.Sandeep.V.M. Principal**, JPNCE for his cooperation and encouragement.

I am grateful to the department of Computer Science and Engineering for providing us with excellent lab and library facilities.

I thank my parents for the love, care and moral support without which i would have not been able to complete this project. It has been a constant source of inspiration for all my academic endeavor.

**M. SRIVIBHA**  
**(16361A0597)**

## **ABSTRACT**

Heavy Rainfall Prediction is a major problem for meteorological department as it is closely associated with the economy and life of human. It is a cause for natural disasters like flood and drought which are encountered by people across the globe every year.

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

In this project we use Machine learning technique i.e, Linear Regression is a process of learning a specific task without any human intervention and improving the performance only by the continuous learning process, this tells us how many inches of rainfall can expect.

# **CONTENTS**

<b>S.NO</b>	<b>CHAPTER</b>	<b>Page No.</b>
	<b>ACKNOWLEDGEMENT</b>	<b>III</b>
	<b>ABSTRACT</b>	<b>IV</b>
<b>1.</b>	<b>INTRODUCTION</b>	<b>1</b>
<b>2.</b>	<b>LITERATURE SURVEY</b>	<b>3</b>
<b>3.</b>	<b>SYSTEM ANALYSIS</b>	<b>5</b>
<b>4.</b>	<b>SYSTEM DESIGN</b>	
	4.1 Modules	7
	4.2 UML Diagram	10
<b>5.</b>	<b>IMPLEMENTATION</b>	<b>14</b>
<b>6.</b>	<b>TESTING</b>	<b>18</b>
<b>7.</b>	<b>SCREENS</b>	<b>20</b>
<b>8.</b>	<b>CONCLUSION &amp; FUTURE ENHANCEMENT</b>	<b>23</b>
<b>9.</b>	<b>REFERENCES</b>	<b>24</b>

# 1.INTRODUCTION

**“Rainfall Prediction”** is used to know about rainfall so that farmer have an idea to manage too choose which crop has to cultivate. Moreover, it helps in managing resources of water. Information of rainfall in prior helps farmers to manage their crops better which result in growth of country’s economy. Fluctuation in rainfall timing and its quantity makes rainfall prediction a challenging task for meteorological scientists. In all the services provided by meteorological department, weather forecasting stands out on top for all the countries across the globe. The task is very complex as it requires numbers of specialized and also all calls are made without any certainty.

When it comes to weather forecasting, rainfall prediction is one of the most widely used research areas as numerous lives and property damages occur due to this. Intense rainfall has abundant impacts on society and on our daily life from cultivation to disaster measures. Previous rainfall prediction models that are widely used, makes use of many the complicated blend of mathematical instruments which was insufficient to get a higher classification rate. In this project, we propose predicting rainfall using linear regression analysis. Rainfall predictions are made by collecting quantitative data about the current state of the atmosphere. Accurate prediction of rainfall is a difficult task due to the dynamic nature of the atmosphere. To predict the future’s rainfall condition, the variation in the conditions in past years must be utilized. We have proposed the use of linear regression by making use of various parameters such as temperature, humidity, and wind. The proposed model tends to forecast rainfall based on the previous records of a particular geographic area, therefore, this prediction will prove to be much reliable. The performance of the model is more accurate when compared with traditional rainfall prediction systems.

The majority of agribusiness is dependent on precipitation as its standard wellspring of water, the time and measure of precipitation hold high importance and can impact the entire economy of the nation. Climate plays a role in our everyday life. From the earliest starting point of the human development, we are occupied with thinking about climatic changes. Weather forecasting is one of the most challenging issues seen by the world , in a most recent couple of century in the field of science and technology. Prediction is the phenomena of knowing what may happen to a system in the near future . Present weather observations are obtained by ground-based instruments and from the satellite through remote sensing. As India’s economy significantly depends on horticulure, precipitation plays an important part.

Weather condition is the state of atmosphere at a given time in terms of weather variables like rainfall, cloud conditions, temperature, etc., the existing models use data mining techniques to predict the rainfall. The main disadvantage of these systems is that it doesn't provide an estimate of the predicted rainfall. The system calculates average of values and understand the state of atmosphere, which doesn't yield estimate results. This paper represents a mathematical method called Linear Regression to predict the rainfall in various districts in southern states of India. The Linear Regression method is modified in order to obtain the most optimum error percentage by iterating and adding some percentage of error to the input values. This method provides an estimate of rainfall using different atmospheric parameters like average temperature and cloud cover to predict the rainfall. The linear regression is applied on the set of data and the coefficients are used to predict the rainfall based on the corresponding values of the parameters. The main advantage of this model is that this model estimates the rainfall based on the previous correlation between the different atmospheric parameters. Thus, an estimate value of what the rainfall could be at a given time period and place can be found easily.

## **2.LITERATURE SURVEY**

### **Prediction of all India summer monsoon rainfall using Error-Back propagation Neural Network**

In this paper, multilayered feedforward neural networks trained with the error-back-propagation (EBP) algorithm have been employed for predicting the seasonal monsoon rainfall over India. Three network models that use, respectively, 2, 3 and 10 input parameters which are known to significantly influence the Indian summer monsoon rainfall (ISMR) have been constructed and optimized. The results obtained thereby are rigorously compared with those from the statistical models. The predictions of network models indicate that they can serve as a potent tool for ISMR prediction.

### **All India summer monsoon rainfall prediction using an Artificial Neural Network**

The prediction of Indian summer monsoon rainfall (ISMR) on a seasonal time scales has been attempted by various research groups using different techniques including artificial neural networks. The prediction of ISMR on monthly and seasonal time scales is not only scientifically challenging but is also important for planning and devising agricultural strategies. This article describes the artificial neural network (ANN) technique with error- back-propagation algorithm to provide prediction of ISMR on monthly and seasonal time scales. The ANN technique is applied to the five time series of June, July, August, September monthly means and seasonal mean (June + July + August + September) rainfall from 1871 to 1994 based on Parthasarathy data set. The previous five years values from all the five time-series were used to train the ANN to predict for the next year. The details of the models used are discussed. Various statistics are calculated to examine the performance of the models and it is found that the models could be used as a forecasting tool on seasonal and monthly time scales.



## **Modeling and prediction of rainfall using Artificial Neural Network and ARIMA techniques**

Rainfall forecasting plays an important role in catchment management applications, the flood warning system being one of them. Rainfall forecasting is one of the most difficult tasks given the variability of space, time and other given conditions change rapidly. Over the years, with the evolution of the intelligent computing methods, many rainfall prediction methods have been proposed, Artificial Neural Network being one of the most prominent. Since the last decade, many researchers have proposed different artificial neural network models in order to create accurate rainfall prediction models. In this paper, different artificial neural networks have been created for the rainfall prediction of Pondicherry, a coastal region in India. These ANN models were created using three different training algorithms namely, feed-forward back propagation algorithm, layer recurrent algorithm and feed-forward distributed time delay algorithm. The number of neurons for all the models was kept at 20. The mean squared error was measured for each model and the best accuracy was obtained by feed-forward distributed time delay algorithm with MSE value as low as .0083.

### **3.SYSTEM ANALYSIS**

#### **Existing system**

If we observe the previous existing projects of rainfall predictions they had considered the data-sets which belongs to the average rainfall of months based on year-wise data. They had predicted the average rainfall of upcoming months and they faced few errors like RMSE , MSE and correlation coefficient.

#### **Proposed system**

In this project by considering the Austin weather dataset, predicted average rainfall based on the climatic parameters like Humidity, DewPoint, SeaLevelPressure, Visibility Miles and Precipitation Inches. Using Linear Regression can predict average Rainfall with more accuracy compared to other algorithms.

## Requirements

Software Requirements

**Operating System:** windows 7,8 & 10(32/64 Bit)

**Language Used:** Python

Python is a simple, general purpose, high level, and object-oriented programming language. It is an interpreted scripting language also. Guido Van Rossum is known as the founder of Python programming. It is a general purpose, dynamic, high-level, and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures. It is easy to learn yet powerful and versatile scripting language, which makes it attractive for Application Development. Python's syntax and dynamic typing with its interpreted nature make it an ideal language for scripting and rapid application development.

Python supports multiple programming pattern, including object-oriented, imperative, and functional or procedural programming styles. Python is not intended to work in a particular area, such as web programming. That is why it is known as multipurpose programming language because it can be used with web, enterprise, 3D CAD, etc. We don't need to use data types to declare variable because it is dynamically typed so we can write `a=10` to assign an integer value in an integer variable. Python makes the development and debugging fast because there is no compilation step included in Python development, and edit-test-debug cycle is very fast.

**Libraries:** Numpy, pandas, scikit-learn

Numpy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. This tutorial explains the basics of NumPy such as its architecture and environment. It also discusses the various array functions, types of indexing, etc. An introduction to Matplotlib is also provided. All this is explained with the help of examples for better understanding. NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Pandas is defined as an open-source library that provides high-performance data manipulation in Python. The name of Pandas is derived from the word Panel Data, which means an Econometrics from Multidimensional data. Data analysis requires lots of processing, such as restructuring, cleaning or merging, etc. There are different tools are available for fast data processing, such as Numpy, Scipy, Cython, and Panda. But we prefer Pandas because working with Pandas is fast, simple and more expressive than other tools. Pandas is built on top of the Numpy package, means Numpy is required for operating the Pandas. Before Pandas, Python was capable for data preparation, but it only provided limited support for data analysis. So, Pandas came into the picture and enhanced the capabilities of data analysis. It can perform five significant steps required for processing and analysis of data irrespective of the origin of the data, i.e., load, manipulate, prepare, model, and analyze.

Scikit-learn is an open source machine learning library that supports supervised and unsupervised learning. It also provides various tools for model fitting, data preprocessing, model selection and evaluation, and many other utilities. Scikit-learn provides dozens of built-in machine learning algorithms and models, called estimators. Each estimator can be fitted to some data using its fit method. In scikit-learn, pre-processors and transformers follow the same API as the estimator objects they actually all inherit from the same Base Estimator class. The transformer objects don't have a predict method but rather a transform method that outputs a newly transformed sample matrix. Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

#### **Software:** Anaconda

With over 20 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries. Search our cloud-based repository to find and install over 7,500 data science and machine learning packages. With the conda-install command, you can start using thousands of open-source Conda, R, Python and many other packages. Individual Edition is an open source, flexible solution that provides the utilities to build, distribute, install, update, and manage software in a cross-platform manner. Conda makes it easy to manage multiple data environments that can be maintained and run separately without interference from each other.

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS.

## 4.SYSTEM DESIGN

### 4.1 Modules

#### Data Collection

Data collection is the process of gathering and measuring information on variables of interest, in an established systematic fashion that enables one to answer stated research questions, test hypotheses, and evaluate outcomes.

The data set can be collected from various sources such as a file, database, sensor and many other such sources but the collected data cannot be used directly for performing the analysis process as there might be a lot of missing data, extremely large values, unorganized text data or noisy data. Therefore, to solve this problem Data Preparation is done.

can also use some free data sets which are present on the internet. Kaggle and UCI Machine learning Repository are the repositories that are used the most for making Machine learning models. Kaggle is one of the most visited websites that is used for practicing machine learning algorithms.

#### Data Preprocessing

Data Preprocessing is the act of manipulating raw data (which may come from disparate data sources) into a form that can readily and accurately be analysed. It includes many discrete tasks such as loading data or data ingestion, data fusion, data cleaning, data augmentation, and data delivery.

Data pre-processing is one of the most important steps in machine learning. It is the most important step that helps in building machine learning models more accurately. In machine learning, there is an 80/20 rule. Every data scientist should spend 80% time for data pre-processing and 20% time to actually perform the analysis. Some of the types are:

**1. Missing data:** Missing data can be found when it is not continuously created or due to technical issues in the application.

**2. Noisy data:** This type of data is also called outliers, this can occur due to human errors (human manually gathering the data) or some technical problem of the device at the time of collection of data.

**3. Inconsistent data:** This type of data might be collected due to human errors (mistakes with the name or values) or duplication of data.

## Choose a Model

**Linear Regression** to predict the amount of rainfall. It tells us how many inches of rainfall we can expect. While a Regression problem is when the target variable is continuous (i.e. the output is numeric).

In statistics, linear regression is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression.

Linear regression was the first type of regression analysis to be studied rigorously, and to be used extensively in practical applications. This is because models which depend linearly on their unknown parameters are easier to fit than models which are non-linearly related to their parameters and because the statistical properties of the resulting estimators are easier to determine.

## Training and Testing the model on data

The process of training an ML model involves providing an ML algorithm (that is, the learning algorithm) with training data to learn from. The training data must contain the correct answer, which is known as a target. The learning algorithm finds patterns in the training data that map the input data attributes to the target (the answer that you want to predict), and it outputs an ML model that captures these patterns.

Train the classifier using 'training data set', tune the parameters and then test the performance of your classifier on unseen 'test data set'. An important point to note is that during training the classifier only the training is available. The test data set must not be used during training the classifier. The test set will only be available during testing the classifier.

**Training set:** The training set is the material through which the computer learns how to process information. Machine learning uses algorithms to perform the training part. A set of data used for learning, that is to fit the parameters of the classifier.

**Test set:** A set of unseen data used only to assess the performance of a fully-specified classifier.

In a data set, a training set is implemented to build up a model, while a test (or validation) set is to validate the model built.

## **Evaluation**

Evaluation allows to test model against data that has never been used for training. This metric allows us to see how the model might perform against data that it has not yet seen. This is meant to be representative of how the model might perform in the real world.

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future.

## **Predictions**

Prediction refers to the output of an algorithm after it has been trained on a historical dataset and applied to new data



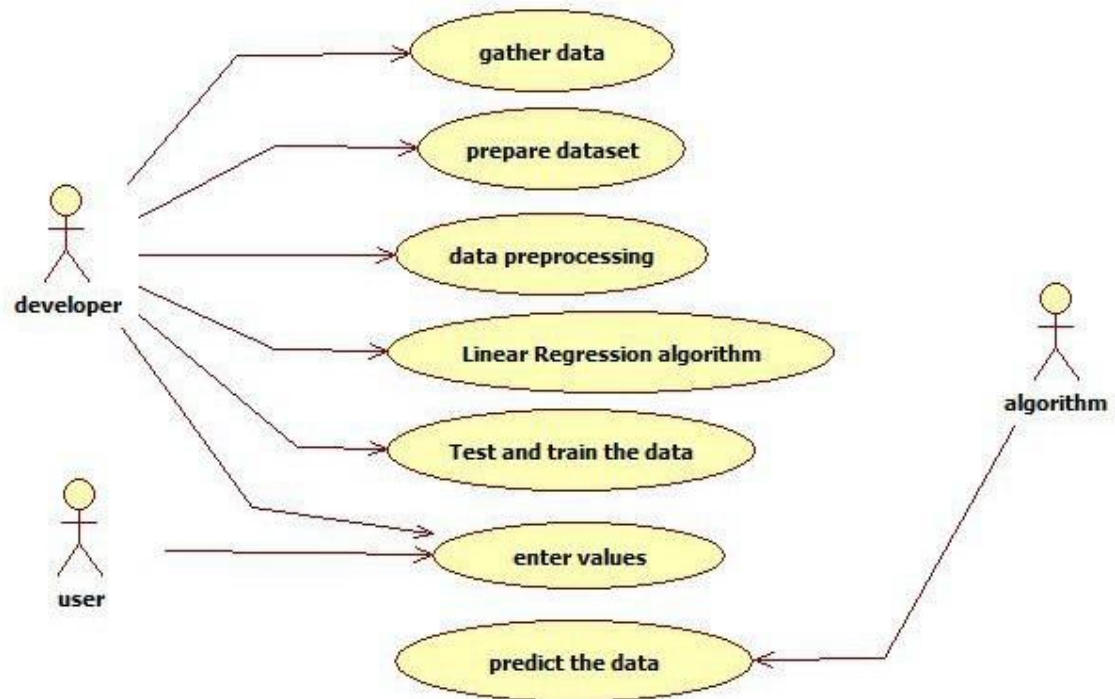
## 4.2 UML DIAGRAMS

### Use case Diagram

The use case diagram describes, what are the actions are to be performed by the user and it also used to analyze the system's high-level instructions. A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

Use case diagrams are considered for high level requirement analysis of a system. When the requirements of a system are analyzed, the functionalities are captured in use cases. We can say that use cases are nothing but the system functionalities written in an organized manner. The second thing which is relevant to use cases are the actors. Actors can be defined as something that interacts with the system. Actors can be a human user, some internal applications, or may be some external applications. When we are planning to draw a use case diagram, we should have the following items identified.

Developer collects and prepares the dataset, then dataset is preprocessed and given it to an algorithm, then the data is divided into test data and train data. Through the train data algorithm learns and predicts the output.



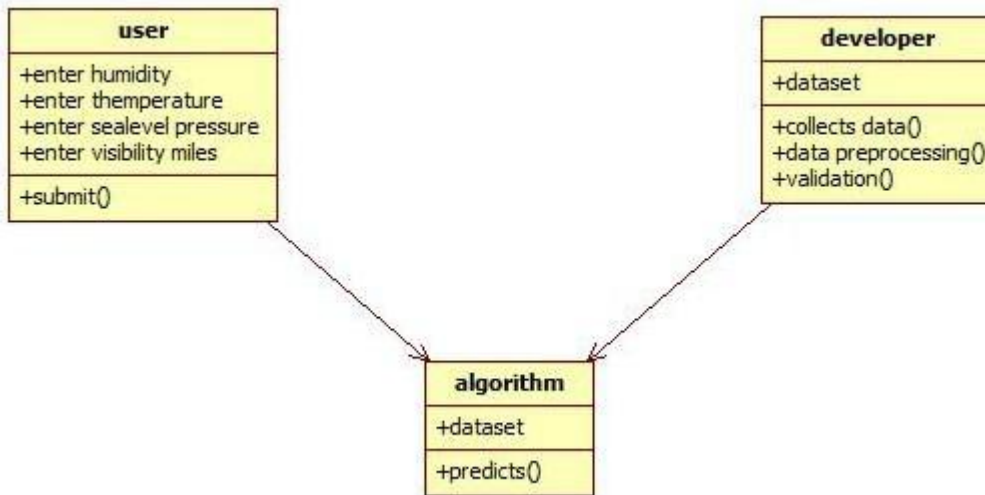
**Fig no: 4.2.1 Use case Diagram**

## **Class Diagram**

Class Diagram is used to show what are the logical entities involved in the project. Class diagrams contain classname, attributes (also referred to as data fields) and behaviors (also referred to as member functions).

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

Here the association exists between user, developer and algorithm. Here attributes are dataset and parameters and methods are like prediction and submit.



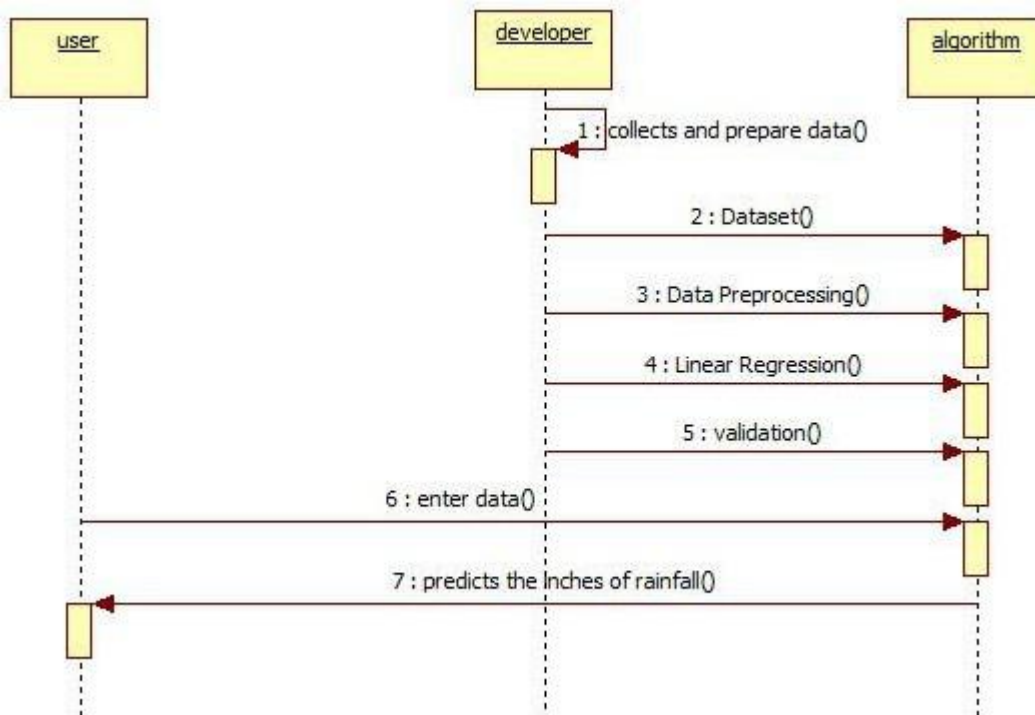
**Fig no: 4.2.2 Class Diagram**

## **Sequence Diagram**

This diagram is used to show the sequence of interactions between User, Developer and Algorithm. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

Lifelines exist between User, developer and algorithm. It explains how each operation is carried out.



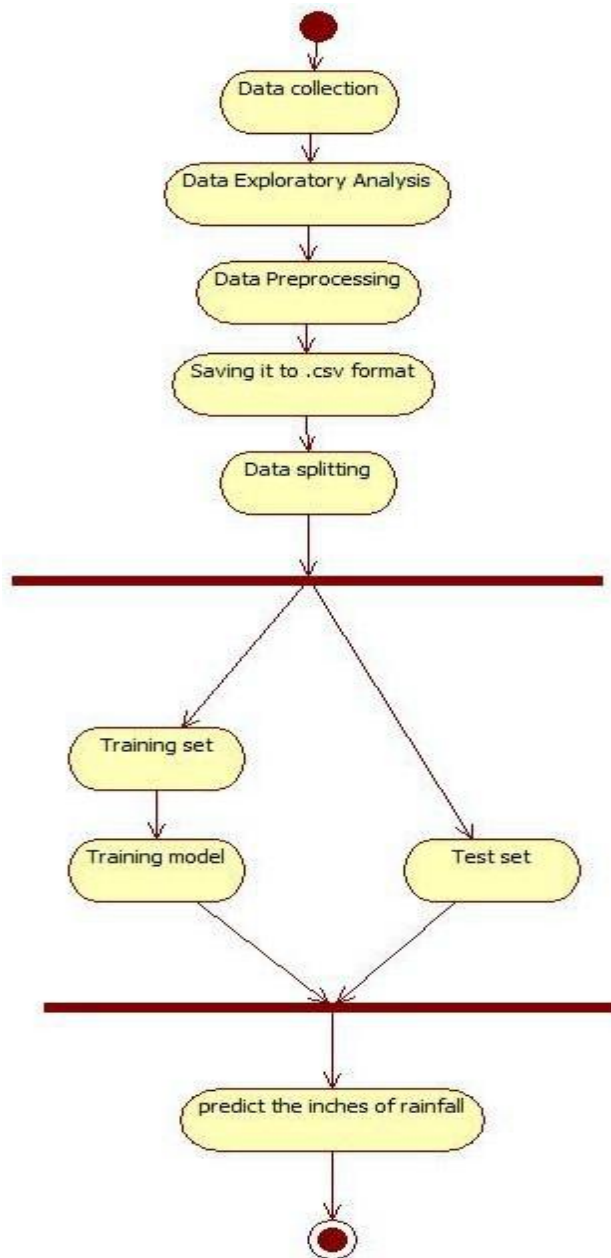
**Fig no: 4.2.3 Sequence diagram**

## **Activity Diagram**

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. It is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

Activity diagrams are mainly used as a flowchart that consists of activities performed by the system. Activity diagrams are not exactly flowcharts as they have some additional capabilities. Before drawing an activity diagram, we must have a clear understanding about the elements used in activity diagram. The main element of an activity diagram is the activity itself. An activity is a function performed by the system. After identifying the activities, we need to understand how they are associated with constraints and conditions.



**Fig no:4.2.4 Activity Diagram**



## 5.IMPLEMENTATION

Feature	Description
Date	The date of observation
TempHighF	The maximum temperature in fahrenheit
TempAvgF	The average temperature in fahrenheit
TempLowF	The minimum temperature in fahrenheit
DewPointHighF	The dew point is the temperature to which air must be cooled to become saturated with water vapor at high level in fahrenheit
DewPointAvgF	The dew point is the temperature to which air must be cooled to become saturated with water vapor at average level in fahrenheit
DewPointLowF	The dew point is the temperature to which air must be cooled to become saturated with water vapor at low level in fahrenheit
HumidityHighPercent	High Humidity is the concentration of water vapour present in the air.
HumidityAvgPercent	Average Humidity is the concentration of water vapour present in the air.
HumidityLowPercent	Low Humidity is the concentration of water vapour present in the air.
SeaLevelPressureHighInches	The high sea level pressure is the atmospheric pressure at sea level in inches
SeaLevelPressureLowInches	The low sea level pressure is the atmospheric pressure at sea level in inches
SeaLevelPressureAvgInches	The average sea level pressure is the atmospheric pressure at sea level in inches

VisibilityHighMiles	The visibility can be very high, such as being able to see through the atmosphere or at distances
VisibilityAvgMiles	The visibility can be average, such as being able to see through the atmosphere or at distances
VisibilityLowMiles	The visibility can be low, such as being able to see through the atmosphere or at distances
WindHighMPH	high wind speed in miles per hour
WindAvgMPH	average wind speed in miles per hour
WindGustMPH	wind gust is a brief increase in the speed of the wind in miles per hour
PrecipitationSumInches	The amount of rainfall recorded for the day in inches
Events	fog, thunderstorm,rain

After preprocessing Date,Events,SeaLevelPressureLowInches and HumidityHighPercent are not considered.

### **P-Value**

The p-value for each term tests the null hypothesis that the coefficient is equal to zero (no effect). A low p-value ( $< 0.05$ ) indicates that you can reject the null hypothesis. In other words, a predictor that has a low p-value is likely to be not a meaningful addition to your model because changes in the predictor's value are related to changes in the response variable.

Conversely, a larger (insignificant) p-value suggests that changes in the predictor are not associated with changes in the response.

```
import statsmodels.api as sm
model=sm.
OLS(y,x).fit()
model.summary()
```

Here OLS is Ordinary Least Square Error, when we run the above statements the following table will appear.

:

### OLS Regression Results

<b>Dep. Variable:</b>	y	<b>R-squared :</b>	0.350
<b>Model:</b>	OLS	<b>Adj. R-squared :</b>	0.341
<b>Method:</b>	Least Squares	<b>F-statistic:</b>	38.90
<b>Date:</b>	Tue, 19 May 2020	<b>Prob (F-statistic):</b>	4.20e-108
<b>Time:</b>	16:39:43	<b>Log-Likelihood:</b>	-513.76
<b>No. Observations:</b>	1319	<b>AIC:</b>	1064.
<b>Df Residuals:</b>	1301	<b>BIC:</b>	1157.
<b>Df Model:</b>	18		
<b>Covariance Type:</b>	nonrobust		

coef	std err	t	P> t	[0.025	0.975]
<b>TempHighF</b>	0.0161	0.013	1.238	0.216	-0.009 0.041
<b>TempAvgF</b>	-0.0288	0.025	-1.136	0.256	-0.079 0.021
<b>TempLowF</b>	0.0145	0.013	1.105	0.269	-0.011 0.040
<b>DewPointHighF</b>	0.0135	0.004	3.660	0.000	0.006 0.021
<b>DewPointAvgF</b>	-0.0259	0.006	-4.292	0.000	-0.038 -0.014
<b>DewPointLowF</b>	0.0134	0.003	3.864	0.000	0.007 0.020
<b>HumidityHighPercent</b>	0.0031	0.009	0.369	0.712	-0.014 0.020
<b>HumidityAvgPercent</b>	-0.0126	0.017	-0.751	0.453	-0.046 0.020
<b>HumidityLowPercent</b>	0.0114	0.008	1.353	0.176	-0.005 0.028
<b>SeaLevelPressureHighInches</b>	-0.1212	0.351	-0.346	0.730	-0.809 0.567
<b>SeaLevelPressureAvgInches</b>	-0.0072	0.618	-0.012	0.991	-1.220 1.206
<b>SeaLevelPressureLowInches</b>	0.1150	0.356	0.324	0.746	-0.582 0.813

<b>VisibilityHighMiles</b>	0.0956	0.016	5.881	0.000	0.064	0.128
<b>VisibilityAvgMiles</b>	-0.0942	0.012	-7.576	0.000	-0.119	-0.070
<b>VisibilityLowMiles</b>	-0.0072	0.005	-1.405	0.160	-0.017	0.003
<b>WindHighMPH</b>	0.0482	0.008	6.045	0.000	0.033	0.064
<b>WindAvgMPH</b>	-0.0417	0.007	-5.745	0.000	-0.056	-0.027
<b>WindGustMPH</b>	-0.0031	0.005	-0.656	0.512	-0.012	0.006

Generally we use OLS(Ordinary Least Square Error),MSE(Mean Squared Error) and RMSE(Root Mean Square Error) to find the error rates in the data model.

### Linear Regression

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used. Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output).

Python Scikit-Learn library for machine learning can be used to implement regression functions. Before fitting data into Linear Regression it must undergo data cleaning by using Numpy and pandas and then **Test and train** the data.

### Test and Train

split the data into training and test sets. By using Scikit-Learn's built-in `train_test_split()` method.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```

The above script splits 70% of the data to training set while 30% of the data to test set.

The test\_size variable is where we actually specify the proportion of test set.

## **Training the algorithm**

Scikit-Learn is extremely straight forward to implement linear regression models, it imports the LinearRegression class, instantiate it, and call the fit() method along with our training data. This is about as simple as it gets when using a machine learning library to train on your data.

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(x_train,y_train)
```

linear regression model basically finds the best value for the intercept and slope, which results in a line that best fits the data.

## **Mean Square Error**

The mean squared error tells you how close a regression line is to a set of points. It does this by taking the distances from the points to the regression line these distances are the “errors” and squaring them.,It's called the mean squared error.The smaller the means squared error, the closer you are to finding the line of best fit. Depending on your data, it may be impossible to get a very small value for the mean squared error.

```
from sklearn.metrics import mean_squared_error
y_predict=regressor.predict(x_test)
accuracy=mean_squared_error(y_test,y_predict)
print(accuracy)
```

### **Root Mean Square Error:**

RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. The square root of the Mean Square Error is Root Mean Square

```
from sklearn import metrics
y_pred = regressor.predict(x_test)
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_predict)))
```

### **Tkinter**

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

```
import Tkinter
top=Tkinter.Tk()
#code to add widgets will go here....
top.mainloop()
```

To create a main window, tkinter offers a method Tk(), The basic code used to create the main window of the application.

mainloop() is used when your application is ready to run. mainloop() is an infinite loop used to run the application, wait for an event to occur and process the event as long as the window is not closed.

### **Tkinter Widgets**

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

The Button widget is used to display buttons in your application.

The Entry widget is used to display a single-line text field for accepting values from a user.

The Label widget is used to provide a single-line caption for other widgets. It can also contain images.

The Text widget is used to display text in multiple lines

grid method()—geometry manager organizes widgets in a table-like structure in the parent widget.

## 6.TESTING

Generally testing can be done either manually or automatically. In RAINFALL PREDICTION Manual testing is done. During testing, validation of this application is done, and it is checked for any defects or errors. If the project contains any error in it, it generates wrong output. To avoid this, manual testing is done. In order to get the correct output, correct input must be given.

Machine Learning models would also need to be tested as conventional software development from the quality assurance perspective. Techniques such as black box and white box testing would, thus, apply to Machine Learning models as well for performing quality control checks on Machine Learning models. Machine Learning represents a class of software that learns from a given set of data and then makes predictions on the new data set based on its learning. In other words, the Machine Learning models are trained with an existing data set in order to make the prediction on a new data set.

### **Black Box Testing**

Blackbox testing is testing the functionality of an application without knowing the details of its implementation including internal program structure, data structures, etc. Test cases for blackbox testing are created based on the requirement specifications. Therefore, it is also called as specification-based testing. When applied to Machine Learning models, blackbox testing would mean testing Machine Learning models without knowing the internal details such as features of the Machine Learning model, the algorithm used to create the model etc. The challenge, however, is to identify the test oracle which could verify the test outcome against the expected values.

### **Blackbox Testing Techniques for Machine Learning Models**

The following represents some of the techniques which could be used to perform blackbox testing on Machine Learning models:

- Model performance
- Metamorphic testing
- Dual coding
- Coverage guided fuzzing



## **Model Performance**

Testing model performance is about testing the models with the test data/new data sets and comparing the model performance in terms of parameters such as accuracy/recall etc., to that of pre-determined accuracy with the model already built and moved into production. This is the most trivial of different techniques which could be used for blackbox testing.

## **Metamorphic Testing**

In metamorphic testing, one or more properties are identified that represent the metamorphic relationship between input-output pairs. In metamorphic testing, the test cases that result in success lead to another set of test cases which could be used for further testing of Machine Learning models.

## **Dual Coding**

With dual coding technique, the idea is to build different models based on different algorithms and comparing the prediction from each of these models given a particular input data set. Let's say, a classification model is built with different algorithms such as random forest, SVM, neural network. All of them demonstrate a comparative accuracy of 90% or so with random forest showing the accuracy of 94%. This results in the selection of random forest. However, during testing, the model for quality control checks, all of the above models are preserved and input is fed into all of the models. For inputs where the majority of remaining models other than random forest gives a prediction which does not match with that of the model built with random forest, a bug/defect could be raised in the defect tracking system. These bugs could later be prioritized and dealt with by data scientists.

## **Coverage Guided Fuzzing**

Coverage guided fuzzing is a technique where data to be fed into the Machine Learning models could be planned appropriately such that all of the features activations get tested. Take for an instance, the models built with neural networks, decision trees, random forest etc. Let's say the model is built using neural networks. The idea is to come up with data sets (test cases) which could result in the activation of each of the neurons present in the neural network. This technique sounds more like a white-box testing.

## 7.SCREENS

This is the screen displayed, after executing the code. Here the default value is 0 and 0.0 which appear before the user enter.

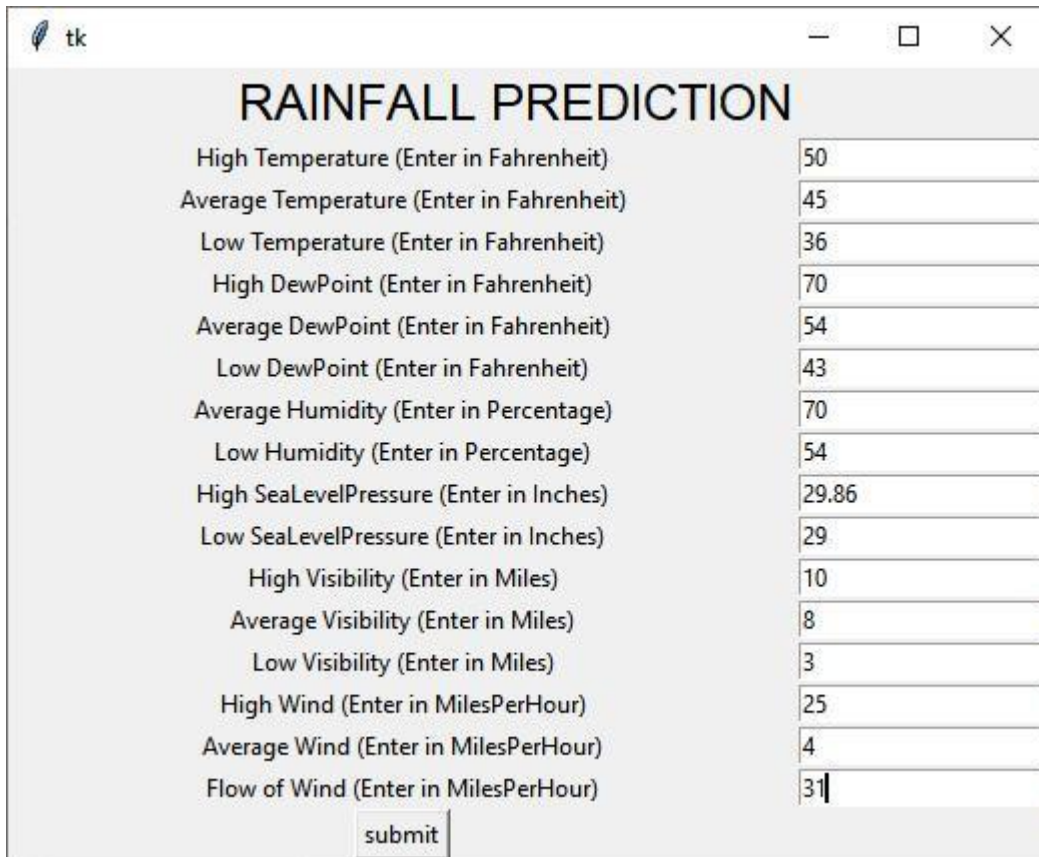
The screenshot shows a Tkinter window titled "RAINFALL PREDICTION". The window contains a list of weather parameters on the left and corresponding input fields on the right. The default values are 0 for most parameters and 0.0 for SeaLevelPressure. A "submit" button is located at the bottom left of the input area.

Parameter	Default Value
High Temperature (Enter in Fahrenheit)	0
Average Temperature (Enter in Fahrenheit)	0
Low Temperature (Enter in Fahrenheit)	0
High DewPoint (Enter in Fahrenheit)	0
Average DewPoint (Enter in Fahrenheit)	0
Low DewPoint (Enter in Fahrenheit)	0
Average Humidity (Enter in Percentage)	0
Low Humidity (Enter in Percentage)	0
High SeaLevelPressure (Enter in Inches)	0.0
Low SeaLevelPressure (Enter in Inches)	0.0
High Visibility (Enter in Miles)	0
Average Visibility (Enter in Miles)	0
Low Visibility (Enter in Miles)	0
High Wind (Enter in MilesPerHour)	0
Average Wind (Enter in MilesPerHour)	0
Low Wind (Enter in MilesPerHour)	0

submit

**Fig no: 7.1 First Screen**

This is the screen where user has entered the values.



The screenshot shows a Tkinter window titled "RAINFALL PREDICTION". The window contains a list of 15 meteorological parameters, each followed by an input field. The values entered in the fields are: 50, 45, 36, 70, 54, 43, 70, 54, 29.86, 29, 10, 8, 3, 25, 4, and 31. A "submit" button is located at the bottom of the form.

Parameter	Value
High Temperature (Enter in Fahrenheit)	50
Average Temperature (Enter in Fahrenheit)	45
Low Temperature (Enter in Fahrenheit)	36
High DewPoint (Enter in Fahrenheit)	70
Average DewPoint (Enter in Fahrenheit)	54
Low DewPoint (Enter in Fahrenheit)	43
Average Humidity (Enter in Percentage)	70
Low Humidity (Enter in Percentage)	54
High SeaLevelPressure (Enter in Inches)	29.86
Low SeaLevelPressure (Enter in Inches)	29
High Visibility (Enter in Miles)	10
Average Visibility (Enter in Miles)	8
Low Visibility (Enter in Miles)	3
High Wind (Enter in MilesPerHour)	25
Average Wind (Enter in MilesPerHour)	4
Flow of Wind (Enter in MilesPerHour)	31

submit

**Fig no: 7.2 Second Screen**

This is the screen appears after submitting the values and the output is displayed below the submit button i.e average rainfall predicted.

The screenshot shows a Tkinter window titled "RAINFALL PREDICTION". It contains a list of 15 input fields with corresponding labels. The labels are: High Temperature (Enter in Fahrenheit), Average Temperature (Enter in Fahrenheit), Low Temperature (Enter in Fahrenheit), High DewPoint (Enter in Fahrenheit), Average DewPoint (Enter in Fahrenheit), Low DewPoint (Enter in Fahrenheit), Average Humidity (Enter in Percentage), Low Humidity (Enter in Percentage), High SeaLevelPressure (Enter in Inches), Low SeaLevelPressure (Enter in Inches), High Visibility (Enter in Miles), Average Visibility (Enter in Miles), Low Visibility (Enter in Miles), High Wind (Enter in MilesPerHour), Average Wind (Enter in MilesPerHour), and Flow of Wind (Enter in MilesPerHour). The values entered in the fields are: 50, 45, 36, 70, 54, 43, 70, 54, 29.86, 29, 10, 8, 3, 25, 4, and 31. Below the input fields is a "submit" button. At the bottom of the window, it says "The precipitation in inches for the input is: [[0.74910446]]".

Parameter	Value
High Temperature (Enter in Fahrenheit)	50
Average Temperature (Enter in Fahrenheit)	45
Low Temperature (Enter in Fahrenheit)	36
High DewPoint (Enter in Fahrenheit)	70
Average DewPoint (Enter in Fahrenheit)	54
Low DewPoint (Enter in Fahrenheit)	43
Average Humidity (Enter in Percentage)	70
Low Humidity (Enter in Percentage)	54
High SeaLevelPressure (Enter in Inches)	29.86
Low SeaLevelPressure (Enter in Inches)	29
High Visibility (Enter in Miles)	10
Average Visibility (Enter in Miles)	8
Low Visibility (Enter in Miles)	3
High Wind (Enter in MilesPerHour)	25
Average Wind (Enter in MilesPerHour)	4
Flow of Wind (Enter in MilesPerHour)	31

submit

The precipitation in inches for the input is: [[0.74910446]]

**Fig no: 7.3 Third Screen**

## **8.CONCLUSION AND FUTURE ENHANCEMENT**

### **Conclusion**

The estimation of rainfall is of great importance in terms of water resources management, human life and their environment. It can be met with the incorrect or incomplete estimation problems because rainfall estimation is affected from the geographical and regional changes and properties.

### **Future Enhancement**

As we have a huge amount of data, we can apply Deep Learning models such as Multilayer Perceptron, Convolutional Neural Network, and others. It would be great to perform a comparative study between the Machine learning classifiers and Deep learning models.

## 9.REFERENCES

- <https://towardsdatascience.com/linear-regression-detailed-view-ea73175f6e86>
- <https://www.geeksforgeeks.org/ml-linear-regression/>
- [https://www.tutorialspoint.com/python/python\\_gui\\_programming.htm](https://www.tutorialspoint.com/python/python_gui_programming.htm)
- <https://scikit-learn.org/stable/>
- <https://stackoverflow.com/questions/27928275/find-p-value-significance-in-scikit-learn-linearregression>
- <https://towardsdatascience.com/the-complete-guide-to-linear-regression-in-python-3d3f8f06bf8>
- <https://stackabuse.com/linear-regression-in-python-with-scikit-learn/>