VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

RAINFALL PREDICTION

A Social and Information Networks (CSE3021) project report

Submitted to:

Dr. Punitha.K

Submitted by:

V Sai preetham (19BCE1434)

Mohit Kaushik (19BCE1522)

Rajeev Krishna(19BCE1360)

In partial fulfilment for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

June 2021

TABLE OF CONTENTS

Page

1.Abstract………………………………………………………………………… 4

2.Introduction………………………………………………………………… 5

3.Feasibility study………………………………………………………………… 6

4.Design and flow of models……………………………………………………. 7

4.1 Strategy……………………………………………………………………… 7

5. About the data set ……………………………………………………….... 6

6. Modules List……………………………………………………………………7.

6.1 Data Collection…………………………………………………………………7

6.2 R-studio Visualization………………………………………………………… 8

6.3 Prediction of Algorithms……………………………………………………… 8

6.4 Rainfall prediction………………………………………………………… 8

6.5 Details about Rainfall…………………………………………………………. 8

6.6 News and Updates and videos…………………………………………………….8

6.7 Help and Contact…………………………………………………………….8

7.Algorithms……………………………………………………………………………9

8.Risk Analysis…………………………………………………………………………9

9.Implementation…………………………………………………………………….10

9.1. Analysis…………………………………………………………………… 18

9.2 Visualization……………………………………………………………………21

9.3. Website……………………………………………………………………………26

10.Conclusion………………………………………………………………………. 32.

11.References………………………………………………………………………… 32

LIST OF FIGURES

Page

Fig 1.0. Telangana rainfall…… ………………………………………………….... 13

Fig 1.1. Command Window R-studio……… …………………………………… 13

Fig 1.2. Density Graph……………………………………………………………... 14

Fig 1.3. Line Graph… ……………………………………………………… 14

Fig 1.4. Pie Chart…………………………………………………………………… 15

Fig 1.5. Bar Graph …………......…………………………………………… 15

Fig 1.6. Histogram Graph……… ……………………………………………… 16

Fig 1.7. Visualization of total years……………………………………………… 21

Fig 1.8. Data group by……………………………………………………………… 22

Fig 1.9. Linear data for Entire months…………………………………………… 22

Fig 2.0. Plot Data…………………………………………………………………… 23

Fig 2.1. Subplot for different months…………………………...…………………… 23

Fig 2.2. Plot Figure…………………………………...……………………… 24

Fig 2.3. Plot figure for different months……………………………………………… 24

Fig 2.5. Comparison of ground and prediction in 2005…………………… 25

Fig 2.6. Comparison of ground and prediction in 2010 and 2015 25

Fig 2.7. sign-up……………………………………………………………………… 26

Fig 2.8. login...……………………………………………………………………… 26

Fig 2.9. Home………………………………………………………………………… 27

Fig 3.0. Rainfall prediction………………………………………………………… 28

Fig 3.1. Rainfall prediction in Hyderabad …………………………………........ ... 28

Fig 3.2. Rainfall prediction in Chennai……………………………………........ 29

Fig 3.3. Details of rainfall……… ……………………………………………… 29

Fig 3.4. Videos management ………… ……………………………………… 30

Fig 3.5. News and updates ………………………………………………………… 30

Fig 3.6. Contact us………………………………………………………………… 31

Fig 3.7. Help………………………………………………………………………… 31

RAINFALL PREDICTION

1.Abstract:

Rainfall is the most crucial process of nature. All the living being rely on water and rainfall is a process that is responsible for the continual process of water cycle. Many farmers will depend upon the rainfall for cultivating many essential crops for us, government also looking ahead of prediction of rainfall which will be useful for the future purposes, many human activities like agriculture are dependent on rainfall, especially in a country like India. Thus, it is very important and necessary to predict the rainfall patterns to estimate the flooding and drowning events. Application of algorithms is the best way to forecast rainfall. These algorithms predict rainfall numerically. There are two kinds of approaches for it. They are empirical and dynamic methods. Empirical approach consists of evaluating historical data, identifying the pattern or relationship between the given atmospheric variables that determine rainfall. It includes many kinds of clustering and classification techniques and also the implementation of the rainfall prediction is very crucial in these days for the farmers and government hence there are many methods and many algorithms which will be very useful for the predicting the rainfall in these days, Dynamic approach consists of dynamically changing training samples whose results can be applied to other large samples of data. In this rainfall prediction by using the machine learning algorithms how rainfall fall can be predicted by machine learning algorithms good accuracy and also using the machine learning algorithms it will be very good for the finding of the rainfall prediction of the machine learning algorithms of dynamic approach like linear regression algorithms, SVM machine learning algorithm ,ARIMA machine learning algorithms and also the classifiers such as the decision tree classifier and also the machine leaning algorithms hence it would be really a best way to include the training and testing of the machine learning algorithms which will be predicting the rainfall for the future days also hence it would be great to use the machine learning algorithms and classifiers for the high efficiency of the machine learning algorithms

2.Introduction:

Knowledge of rainfall patterns help in drought and flood management. Considering suitable attributes for prediction of rainfall is not an easy task. The intensity and frequency estimation is very useful for an agriculture based country like India, especially in the coastal regions in the states of Karnataka, Kerala, Andhra Pradesh, Orissa and West Bengal. So to correctly estimate the rainfall pattern why not use the power of computer science and data mining.

Linear Regression is a supervised machine learning algorithm where the predicted output is continuous and has a constant slope. It's used to predict values within a continuous range, (e.g. sales, price) rather than trying to classify them into categories.

Decision Tree is a process of obtaining data from a database with certain techniques and process. It consists of identifying the pattern in the given data, called training sample, or analyzing a set of already classified objects, whose results can be used to predict the results of other data with missing attributes, called test sample.

It aims at accurate analysis of data and generation of precise results. This analysis and prediction of data in machine learning can be done by many methods like classification, clustering, regression. Rainfall prediction can be done by algorithms like, SVM, Logistic regression, Linear regression, decision tree as they provide a methodology to classify and predict data better than the traditional statistical techniques. These algorithms observe the relationship between the attributes in the data set. In this paper values for atmospheric variables like pressure, temperature, wind speed, wind direction, rainfall percentage and humidity were considered for 10 years.

Rainfall prediction was done by a classification method of Decision Tree and Regression method of linear Regression, SVM and also the ARIMA algorithm for the prediction of the rainfall for the present and for the future days of the rainfall in the percentage format and had collected the API from the open weather website which yields to give the humidity and the windspeed and collects the present rainfall percentage in the particular place around the globe

3.Feasibility study:

- This explains about rainfall is more important for food production plan, water resources and all activities in nature. Due to heavy rain or dry period in critical stages crop growth may reduce crop yield. Rain fall prediction is more significant in agricultural countries like India monsoon rain fall is highly complex. Using this prediction method, we can forecast rainfall for our states also.
- This explained there are many data mining techniques like GRNN, MLP, NNARX, CART, etc.to implement rainfall prediction. They compare various algorithms for suitable rain fall prediction. These techniques can generate decision or prediction models based on data.
- This explains about accurate and exact estimated rainfall prediction and all techniques in data mining is not possible to estimate the rain fall prediction. data mining is much-needed opportunity to deliver information for stake holders and decision makers. rain fall estimation and prediction varies from techniques.

4.Design and Flow of modules:

Following are the modules in which we have divided are project:

- Making prediction model and forming the strategy for the prediction and setting the definite approach.
- Analysing the data set from the **Kaggle** website
- Visualizing the data set using R-studio and Jupyter notebook
- Creation of the train and test data set from the available data and making a feature extraction model.

- ➢ Implementing linear regression ML models and SVM machine learning model and doing a comparative analysis to find out where the prediction is best.

- ➢ Implementing the website for the Rainfall prediction

4.1 Strategy:

- ✓ Collecting the data set from the Kaggle website & Arranging Data in an Accessible manner.
- ✓ Collecting the data from the Kaggle website for the past 10 years and we will collect the data set which includes the variables like wind speed, humidity, rainfall, temperature, etc. which helps to visualize and predict the data
- ✓ Visualizing the data set from the collected data set.
- ✓ Visualizing the data in the form of histograms, bar graphs, pie charts, density, etc. using R-Tool and it will be useful for the prediction of the rainfall part.
- ✓ Prediction of the data set from the collected data set.
- ✓ Predict the data set by implementing the algorithms like linear regression. logistic regression, random forest, decision tree, entropy, etc. which helps in finding the rainfall percentage using python in Jupyter notebook
- ✓ Drawing conclusions from the Collected data set and Observations.
- ✓ Concluding from the predictions valuable Data points which can give us meaningful conclusions about the scenario.
- ✓ Uploading the data set to a Web portal for Easy Accessibility.
- ✓ The Valuable Data Gained from the given data set will be uploaded to an open ended web portal for easy accessibility to the users within the cost of free for the end users and also for the administrators

5.About the data set:

The data set is from the www.kaggle.com website from the past 20 years data set which contains the variables as wind speed, rainfall, humidity, temperature, wind direction etc which contains the data set of 4377 columns and 19 different rows

And also the data set from the www.dataworld.com to visualize the data set in the R-studio which helps in the part of the prediction and used in the part of implementing the website of rainfall prediction

6.Modules List:

There are many number of modules for the report according to the slot of work divides into only 7 modules

6.1. Data collection:

Rainfall Prediction data set collection is very important for the predicting the rainfall for the different purpose of the prediction, visualization

6.2.R-studio Visualization:

Rainfall Visualization is very important task is used to predict the rainfall the usage of the visualization will be done under the r-studio tool

6.3. Predicting of Algorithms:

Rainfall Prediction is a very important task in which machine learning algorithms are used to predict the rainfall prediction in various algorithms like linear regression ,SVM ,and Arima algorithms ,etc.

6.4. Rainfall prediction:

Rainfall Prediction is a navigation link where you can easily see the rainfall amount or weather of upcoming week of your region or of globally.

6.5. Details about the rainfall:

In this section, the user get to know the amount of rainfall in different region of India It will show which region got maximum webpage and which region got minimum amount of rainfall.

6.6. News and updates and videos

Here, the user can see all the news related to the weather in the world, the humidity level, the wind speed and many more information they got, In the Videos section, the user will be able to see all the videos related to weather forecasting and the live video feature is also added in this part

6.7. Contact and help

If the user is having any query, this on the contact page the users can communicate with our team.

7.Algorithms:

Linear Regression Model Representation

Linear regression is an attractive model because the representation is so simple.
The representation is a linear equation that combines a specific set of input values (x) the solution to which is the predicted output for that set of input values (y). As such, both the input values (x) and the output value are numeric.

For example, in a simple linear regression problem (a single x and a single y), the form of the model would be:

$y = A0 + A1*x$

In higher dimensions when we have more than one input (x), the line is called a plane or a hyper-plane. The representation therefore is the form of the equation and the specific values used for the coefficients (e.g., A0 and A1 in the above example).

SVM machine

SVM is an attractive model because the representation is so simple.
The representation is a linear equation that combines a specific set of input values (x) the solution to which is the predicted output for that set of input values (y). As such, both the input values (x) and the output value are numeric.

For example, in a simple SVM machine of w instance of the problem (a single x and a single y), the form of the model would be:

$f(x) = (w.x) + b \ n \ w \in R$

ARIMA

Automated regression integrated moving average is an attractive model because the representation is so simple.
The representation is a algorithm that combines a specific set of input values (x) the solution to which is the predicted output for that set of input values (y). As such, both the input values (x) and the output value are numeric.

For example, in a simple linear regression problem (a single x and a single y),and n is the average of the dataset the form of the model would be:

y = A0 + A1*x/n

8.Risk Analysis:

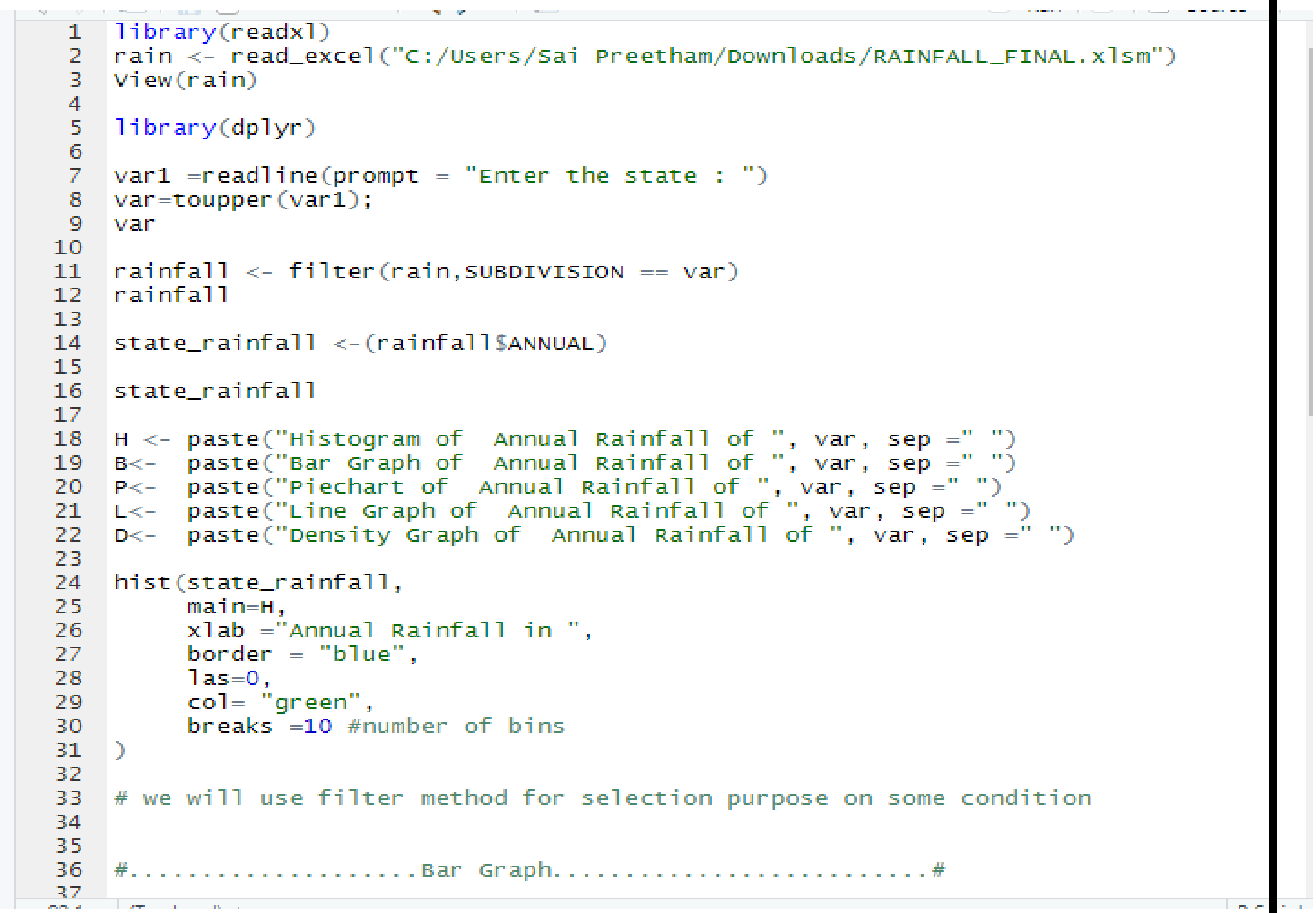
Due to the large amount of data in rainfall, achieving highly efficiency is somewhat difficult for the machine learning algorithms like linear regression model ,SVM model, Decision tree classifier , Self-Organising Map and ARIMA (Auto regressive integrated moving average) Algorithms there is high chance of risk in these types of machine learning algorithms and collection of the API form the website is quite difficult from the open weather website and also from the weather world website it will be very critical to find the API when there is a heavy rain at that particular place and at particular location it will be unable to fetch the details if the weather condition at that particular situation is not good at all for predicting of the website

9.Implementation:

First we implemented the visualization in the R-studio:

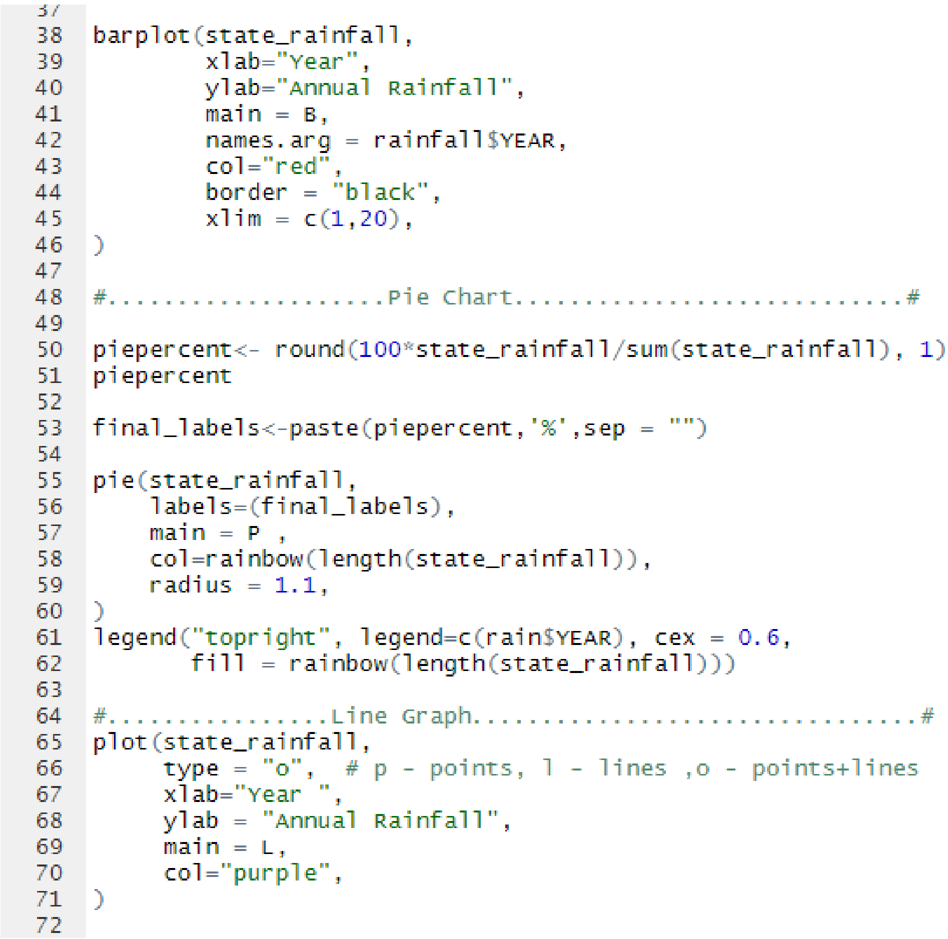
Code:

visualization of the dataset



```
library(readxl)
rain <- read_excel("C:/Users/Sai Preetham/Downloads/RAINFALL_FINAL.xlsm")
View(rain)

library(dplyr)

var1 =readline(prompt = "Enter the state : ")
var=toupper(var1);
var

rainfall <- filter(rain,SUBDIVISION == var)
rainfall

state_rainfall <-(rainfall$ANNUAL)

state_rainfall

H <- paste("Histogram of  Annual Rainfall of ", var, sep =" ")
B<-  paste("Bar Graph of  Annual Rainfall of ", var, sep =" ")
P<-  paste("Piechart of  Annual Rainfall of ", var, sep =" ")
L<-  paste("Line Graph of  Annual Rainfall of ", var, sep =" ")
D<-  paste("Density Graph of  Annual Rainfall of ", var, sep =" ")

hist(state_rainfall,
     main=H,
     xlab ="Annual Rainfall in ",
     border = "blue",
     las=0,
     col= "green",
     breaks =10 #number of bins
)

# we will use filter method for selection purpose on some condition


#....................Bar Graph..........................#
```

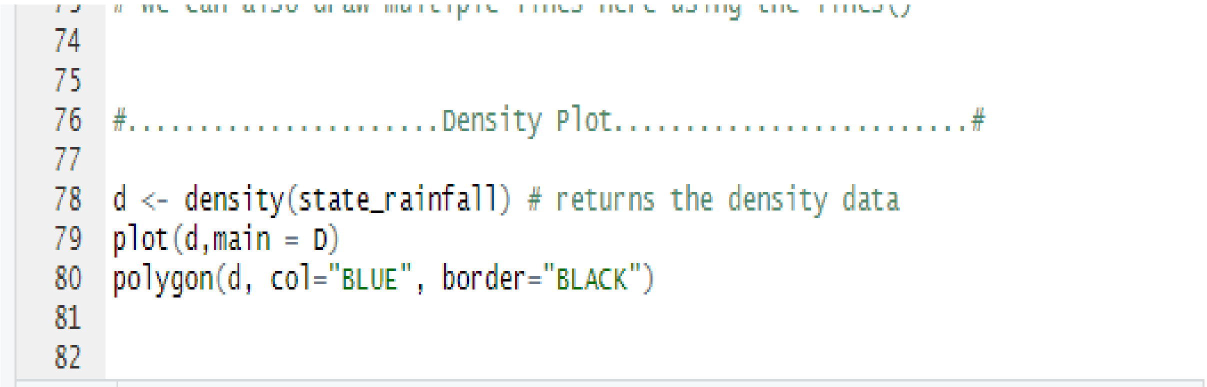
R-studio code for the visualization of the dataset



```
barplot(state_rainfall,
        xlab="Year",
        ylab="Annual Rainfall",
        main = B,
        names.arg = rainfall$YEAR,
        col="red",
        border = "black",
        xlim = c(1,20),
)

#...................Pie Chart...........................#

piepercent<- round(100*state_rainfall/sum(state_rainfall), 1)
piepercent

final_labels<-paste(piepercent,'%',sep = "")

pie(state_rainfall,
    labels=(final_labels),
    main = P ,
    col=rainbow(length(state_rainfall)),
    radius = 1.1,
)
legend("topright", legend=c(rain$YEAR), cex = 0.6,
       fill = rainbow(length(state_rainfall)))

#...............Line Graph..............................#
plot(state_rainfall,
     type = "o",  # p - points, l - lines ,o - points+lines
     xlab="Year ",
     ylab = "Annual Rainfall",
     main = L,
     col="purple",
)

```

R-studio code for the visualization of the dataset

visualization of the dataset



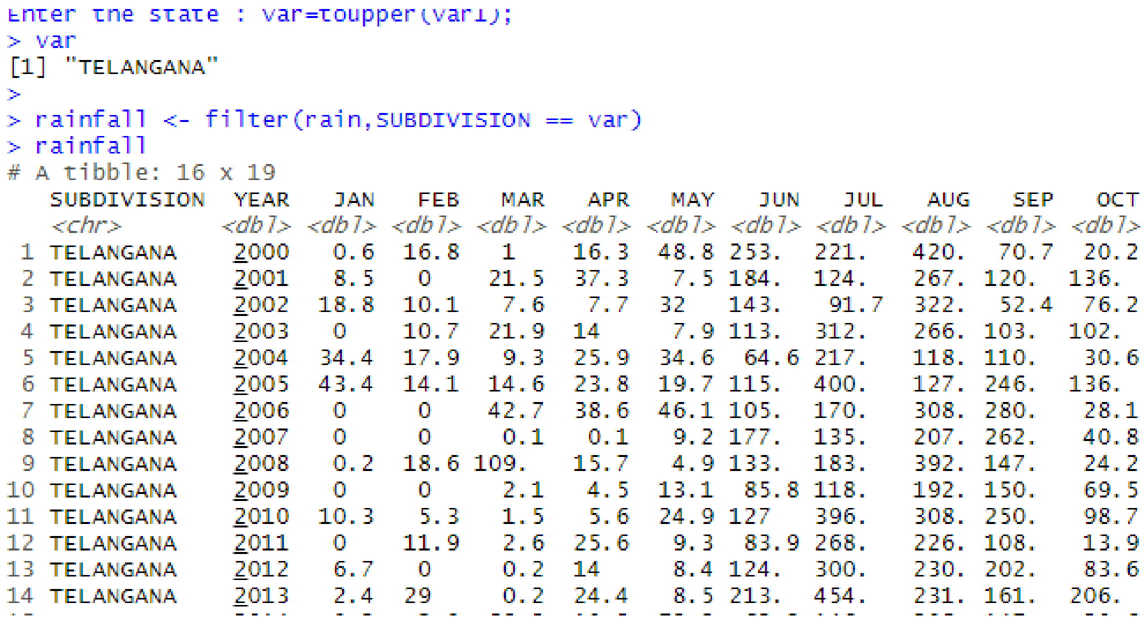
```
# We can also draw multiple lines here using the lines()


#....................Density Plot.......................#

d <- density(state_rainfall) # returns the density data
plot(d,main = D)
polygon(d, col="BLUE", border="BLACK")


```

R-studio code for the visualization of the dataset

Output:

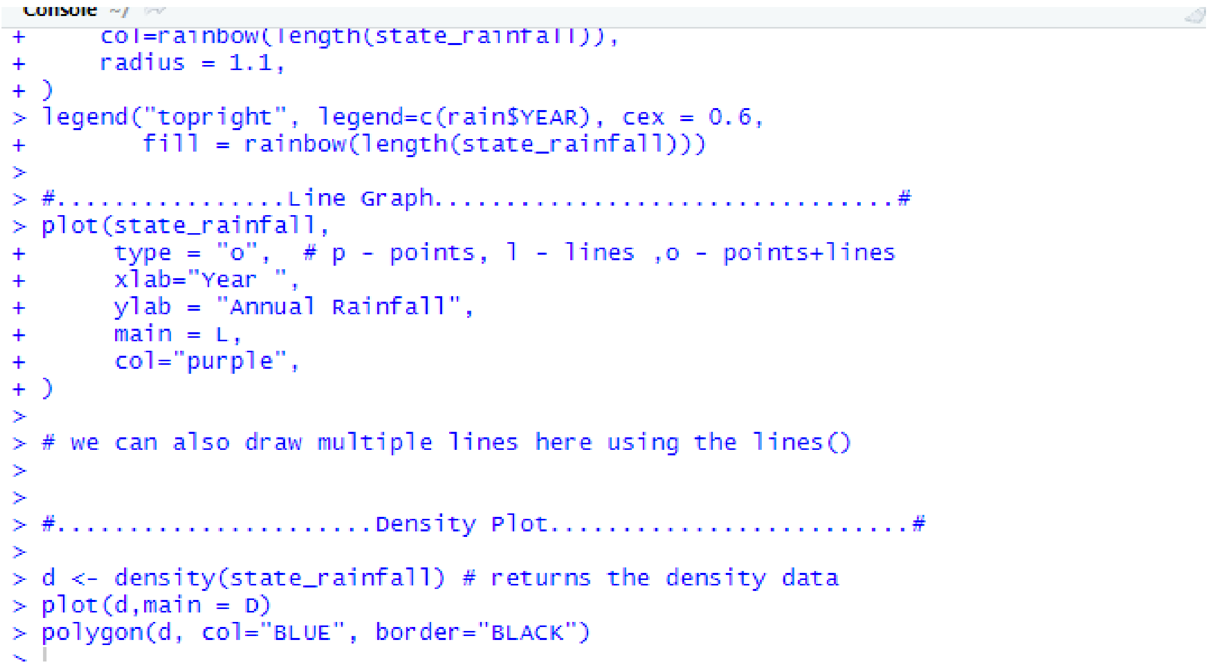
Input is given as Telangana

Telangana rainfall



```
Enter the state : var=toupper(var1);
> var
[1] "TELANGANA"
>
> rainfall <- filter(rain,SUBDIVISION == var)
> rainfall
# A tibble: 16 x 19
   SUBDIVISION  YEAR   JAN   FEB   MAR   APR   MAY   JUN   JUL   AUG   SEP   OCT
   <chr>       <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
 1 TELANGANA    2000   0.6  16.8   1    16.3  48.8 253.  221.   420.  70.7  20.2
 2 TELANGANA    2001   8.5   0    21.5  37.3   7.5 184.  124.   267. 120.  136. 
 3 TELANGANA    2002  18.8  10.1   7.6   7.7  32   143.   91.7  322.  52.4  76.2
 4 TELANGANA    2003   0    10.7  21.9  14     7.9 113.  312.   266. 103.  102. 
 5 TELANGANA    2004  34.4  17.9   9.3  25.9  34.6  64.6 217.   118. 110.   30.6
 6 TELANGANA    2005  43.4  14.1  14.6  23.8  19.7 115.  400.   127. 246.  136. 
 7 TELANGANA    2006   0     0    42.7  38.6  46.1 105.  170.   308. 280.   28.1
 8 TELANGANA    2007   0     0     0.1   0.1   9.2 177.  135.   207. 262.   40.8
 9 TELANGANA    2008   0.2  18.6 109.   15.7   4.9 133.  183.   392. 147.   24.2
10 TELANGANA    2009   0     0     2.1   4.5  13.1  85.8 118.   192. 150.   69.5
11 TELANGANA    2010  10.3   5.3   1.5   5.6  24.9 127   396.   308. 250.   98.7
12 TELANGANA    2011   0    11.9   2.6  25.6   9.3  83.9 268.   226. 108.   13.9
13 TELANGANA    2012   6.7   0     0.2  14     8.4 124.  300.   230. 202.   83.6
14 TELANGANA    2013   2.4  29     0.2  24.4   8.5 213.  454.   231. 161.  206. 
```

Fig-1.0



```
+     col=rainbow(length(state_rainfall)),
+     radius = 1.1,
+ )
> legend("topright", legend=c(rain$YEAR), cex = 0.6,
+        fill = rainbow(length(state_rainfall)))
>
> #...............Line Graph..............................#
> plot(state_rainfall,
+      type = "o",  # p - points, l - lines ,o - points+lines
+      xlab="Year ",
+      ylab = "Annual Rainfall",
+      main = L,
+      col="purple",
+ )
>
> # we can also draw multiple lines here using the lines()
>
>
> #.....................Density Plot.......................#
>
> d <- density(state_rainfall) # returns the density data
> plot(d,main = D)
> polygon(d, col="BLUE", border="BLACK")
```

Fig-1.1

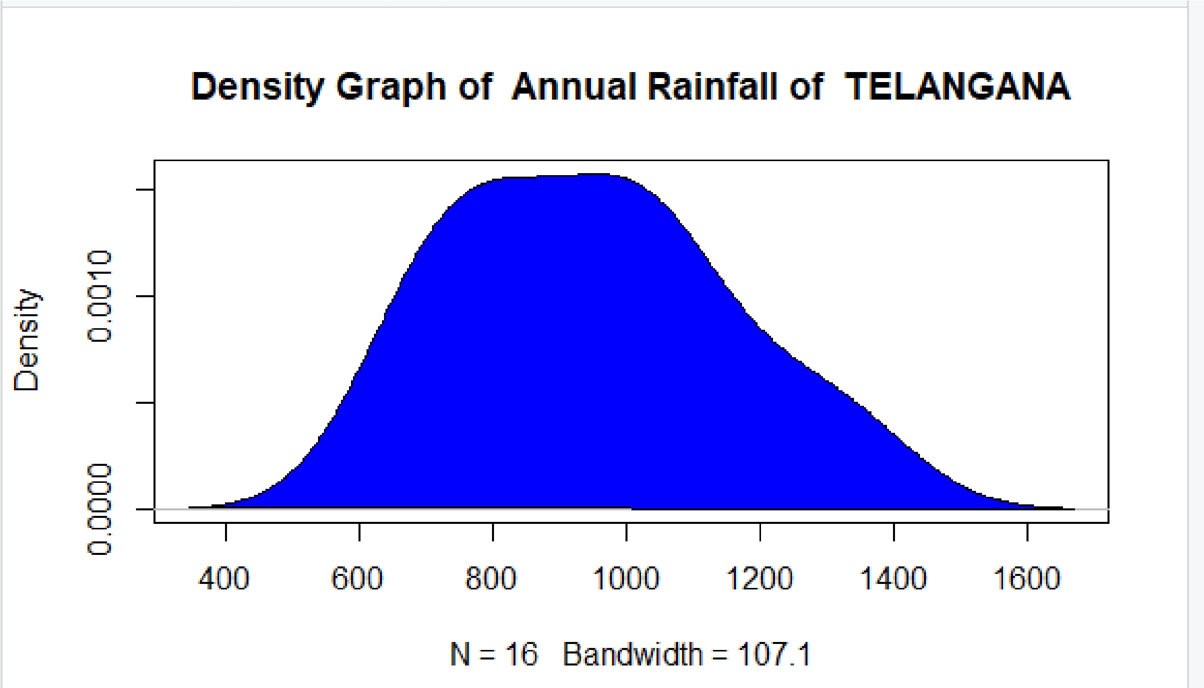


Fig-1.2

Line graph for the annual rainfall of Telangana

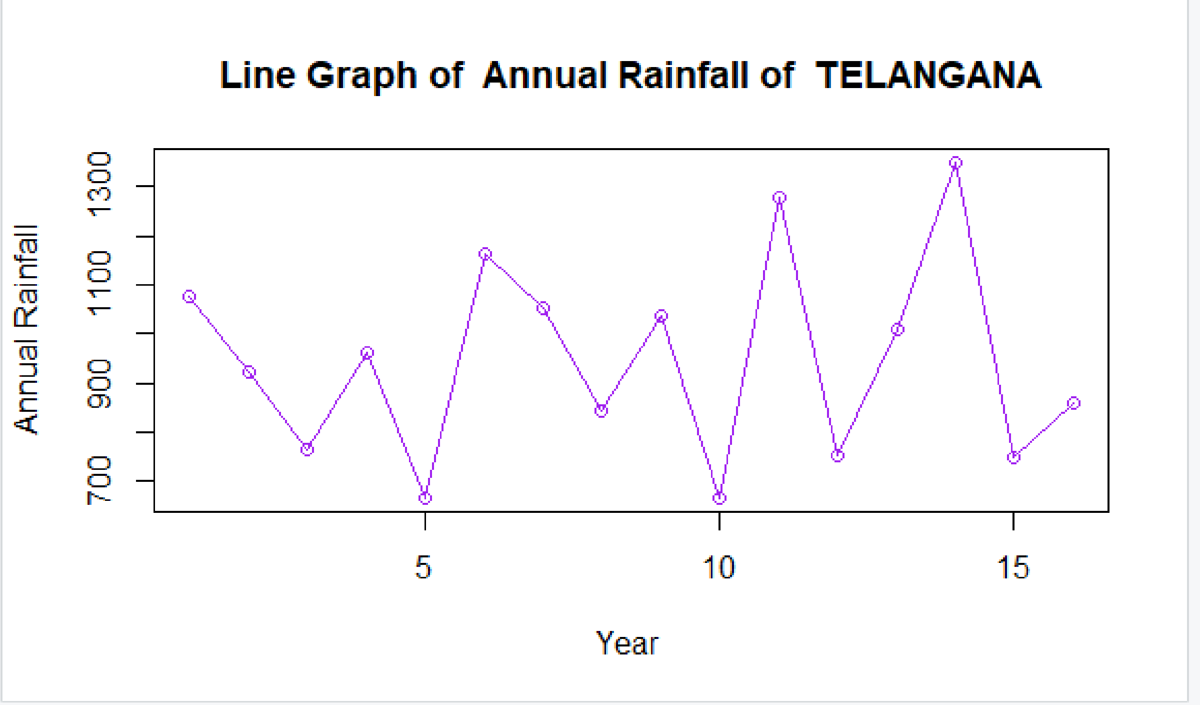


Fig-1.3

Pie chart for the annual rainfall of Telangana

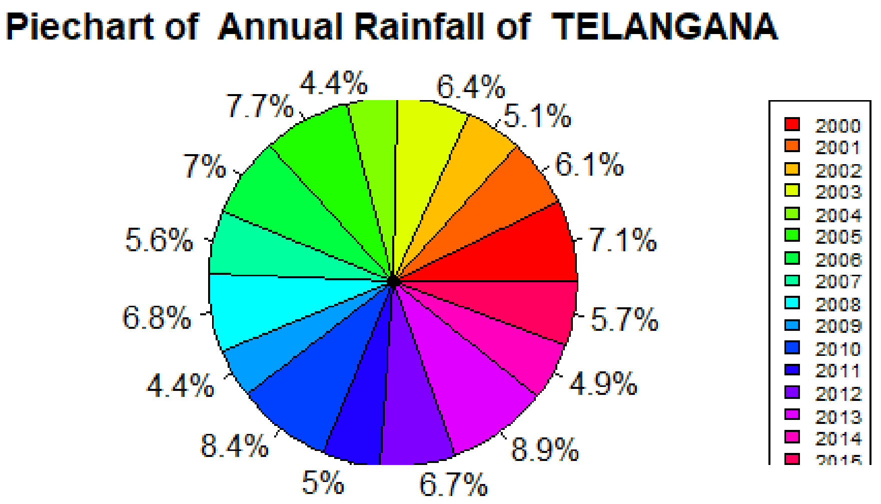


Fig-1.4

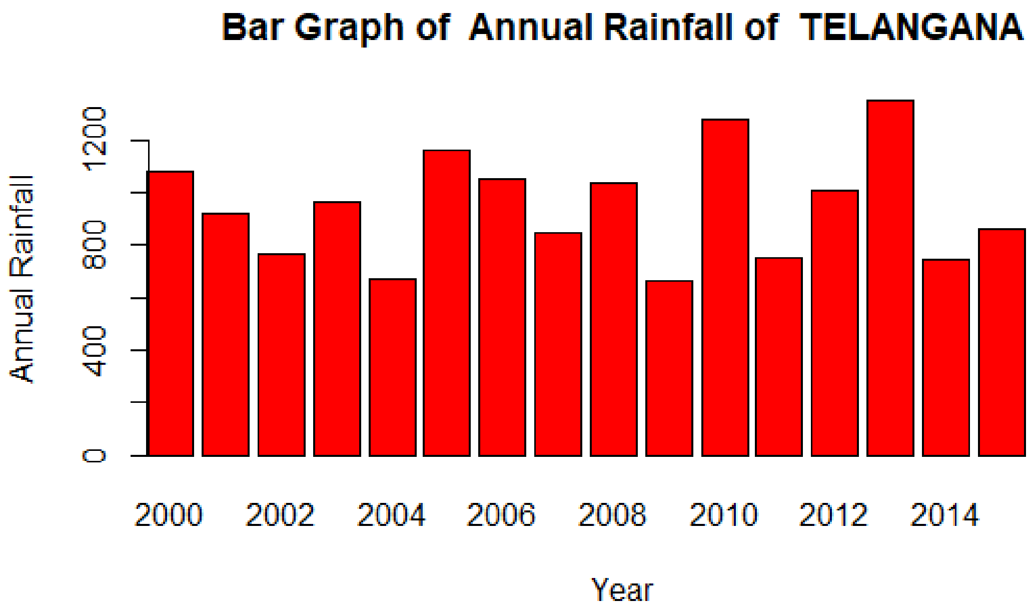


Fig-1.5

Histogram graph for the annual rainfall of Telangana

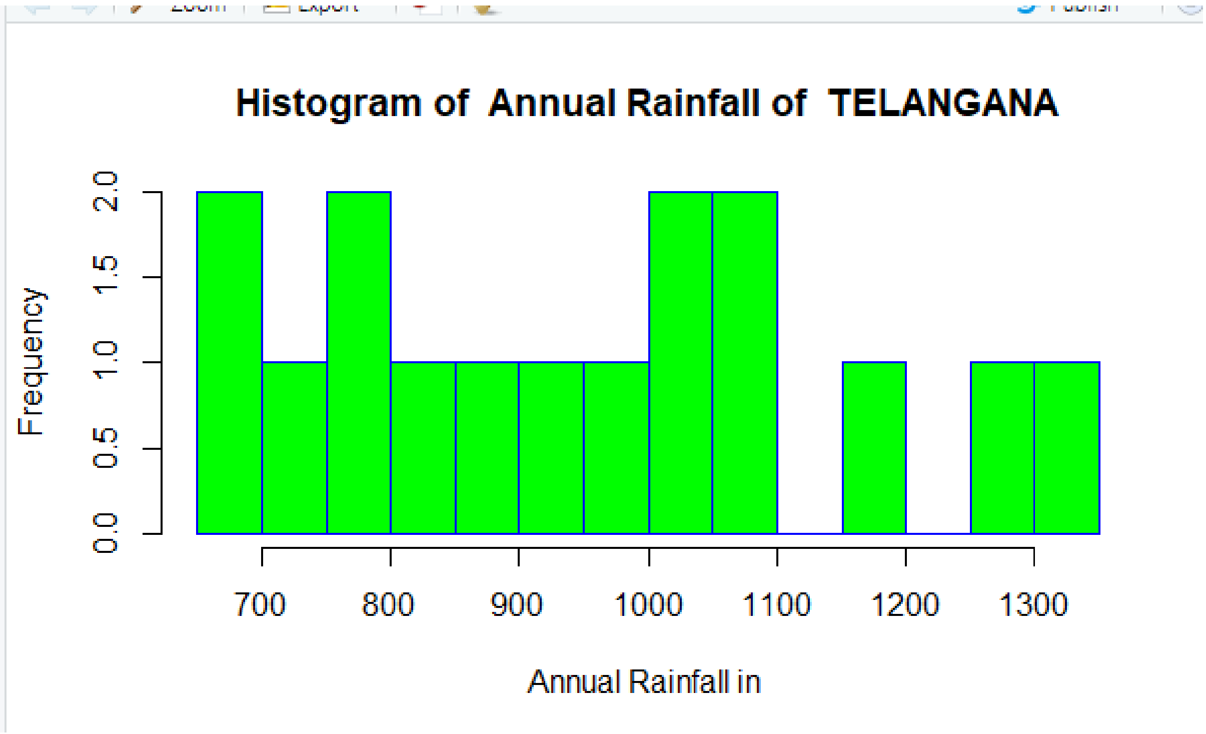
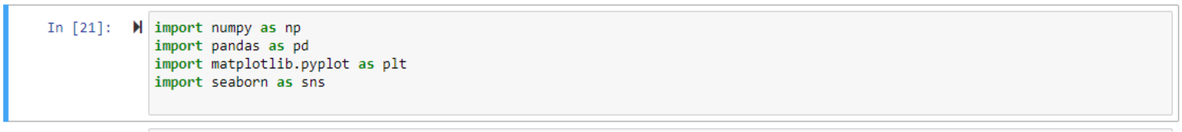


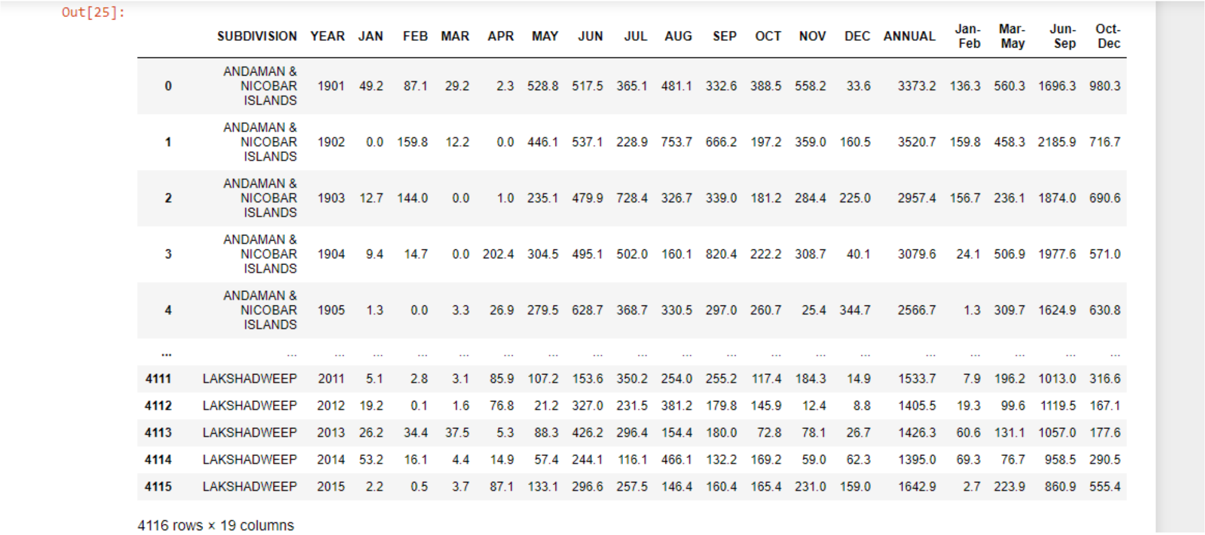
Fig-1.6

We will first import all the necessary libraries and modules:



```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Listing the data set and Machine learning algorithms in the Jupyter notebook in the python language

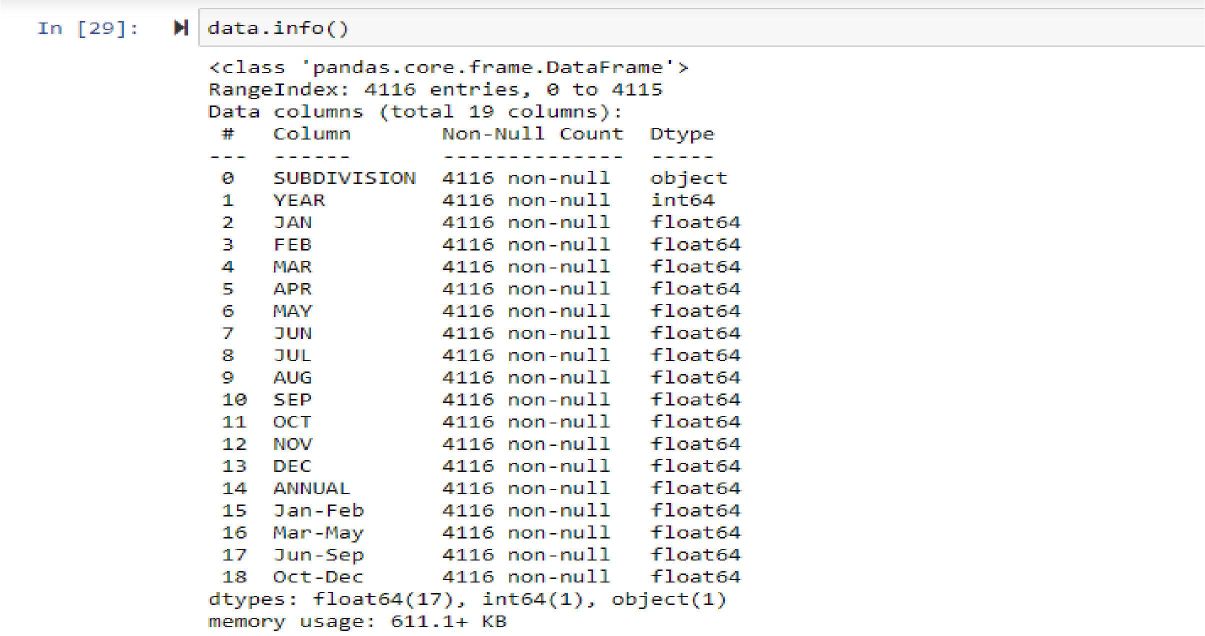


Out[25]:

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May	Jun-Sep	Oct-Dec
0	ANDAMAN & NICOBAR ISLANDS	1901	49.2	87.1	29.2	2.3	528.8	517.5	365.1	481.1	332.6	388.5	558.2	33.6	3373.2	136.3	560.3	1696.3	980.3
1	ANDAMAN & NICOBAR ISLANDS	1902	0.0	159.8	12.2	0.0	446.1	537.1	228.9	753.7	666.2	197.2	359.0	160.5	3520.7	159.8	458.3	2185.9	716.7
2	ANDAMAN & NICOBAR ISLANDS	1903	12.7	144.0	0.0	1.0	235.1	479.9	728.4	326.7	339.0	181.2	284.4	225.0	2957.4	156.7	236.1	1874.0	690.6
3	ANDAMAN & NICOBAR ISLANDS	1904	9.4	14.7	0.0	202.4	304.5	495.1	502.0	160.1	820.4	222.2	308.7	40.1	3079.6	24.1	506.9	1977.6	571.0
4	ANDAMAN & NICOBAR ISLANDS	1905	1.3	0.0	3.3	26.9	279.5	628.7	368.7	330.5	297.0	260.7	25.4	344.7	2566.7	1.3	309.7	1624.9	630.8
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4111	LAKSHADWEEP	2011	5.1	2.8	3.1	85.9	107.2	153.6	350.2	254.0	255.2	117.4	184.3	14.9	1533.7	7.9	196.2	1013.0	316.6
4112	LAKSHADWEEP	2012	19.2	0.1	1.6	76.8	21.2	327.0	231.5	381.2	179.8	145.9	12.4	8.8	1405.5	19.3	99.6	1119.5	167.1
4113	LAKSHADWEEP	2013	26.2	34.4	37.5	5.3	88.3	426.2	296.4	154.4	180.0	72.8	78.1	26.7	1426.3	60.6	131.1	1057.0	177.6
4114	LAKSHADWEEP	2014	53.2	16.1	4.4	14.9	57.4	244.1	116.1	466.1	132.2	169.2	59.0	62.3	1395.0	69.3	76.7	958.5	290.5
4115	LAKSHADWEEP	2015	2.2	0.5	3.7	87.1	133.1	296.6	257.5	146.4	160.4	165.4	231.0	159.0	1642.9	2.7	223.9	860.9	555.4

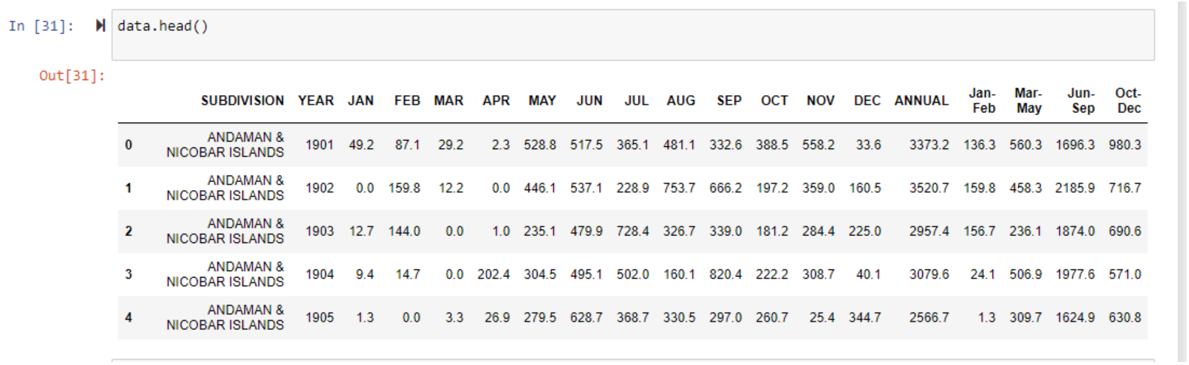
4116 rows × 19 columns

This is the info of data showed in the Jupyter notebook using python language



```
In [29]:    data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4116 entries, 0 to 4115
Data columns (total 19 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   SUBDIVISION  4116 non-null   object 
 1   YEAR         4116 non-null   int64  
 2   JAN          4116 non-null   float64
 3   FEB          4116 non-null   float64
 4   MAR          4116 non-null   float64
 5   APR          4116 non-null   float64
 6   MAY          4116 non-null   float64
 7   JUN          4116 non-null   float64
 8   JUL          4116 non-null   float64
 9   AUG          4116 non-null   float64
 10  SEP          4116 non-null   float64
 11  OCT          4116 non-null   float64
 12  NOV          4116 non-null   float64
 13  DEC          4116 non-null   float64
 14  ANNUAL       4116 non-null   float64
 15  Jan-Feb      4116 non-null   float64
 16  Mar-May      4116 non-null   float64
 17  Jun-Sep      4116 non-null   float64
 18  Oct-Dec      4116 non-null   float64
dtypes: float64(17), int64(1), object(1)
memory usage: 611.1+ KB
```

Info of the data is shown in the above figure



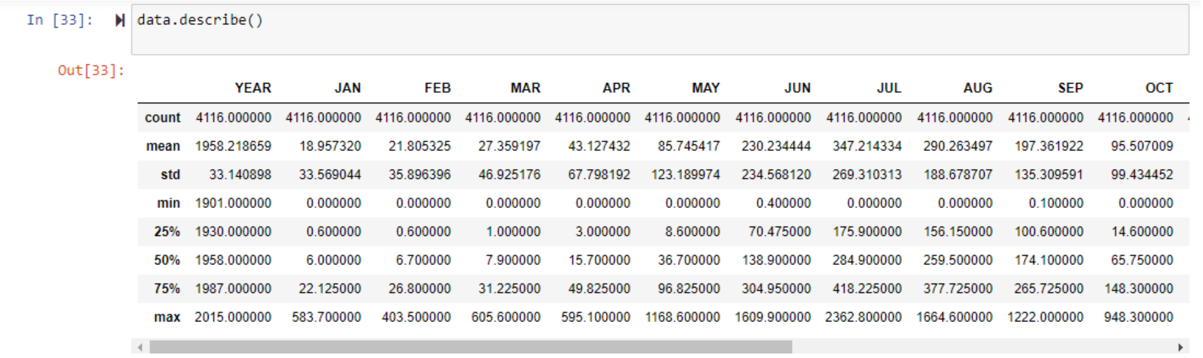
In [31]:

```
data.head()
```

Out[31]:

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May	Jun-Sep	Oct-Dec
0	ANDAMAN & NICOBAR ISLANDS	1901	49.2	87.1	29.2	2.3	528.8	517.5	365.1	481.1	332.6	388.5	558.2	33.6	3373.2	136.3	560.3	1696.3	980.3
1	ANDAMAN & NICOBAR ISLANDS	1902	0.0	159.8	12.2	0.0	446.1	537.1	228.9	753.7	666.2	197.2	359.0	160.5	3520.7	159.8	458.3	2185.9	716.7
2	ANDAMAN & NICOBAR ISLANDS	1903	12.7	144.0	0.0	1.0	235.1	479.9	728.4	326.7	339.0	181.2	284.4	225.0	2957.4	156.7	236.1	1874.0	690.6
3	ANDAMAN & NICOBAR ISLANDS	1904	9.4	14.7	0.0	202.4	304.5	495.1	502.0	160.1	820.4	222.2	308.7	40.1	3079.6	24.1	506.9	1977.6	571.0
4	ANDAMAN & NICOBAR ISLANDS	1905	1.3	0.0	3.3	26.9	279.5	628.7	368.7	330.5	297.0	260.7	25.4	344.7	2566.7	1.3	309.7	1624.9	630.8

Head of the data is shown in the above figure



In [33]:

```
data.describe()
```

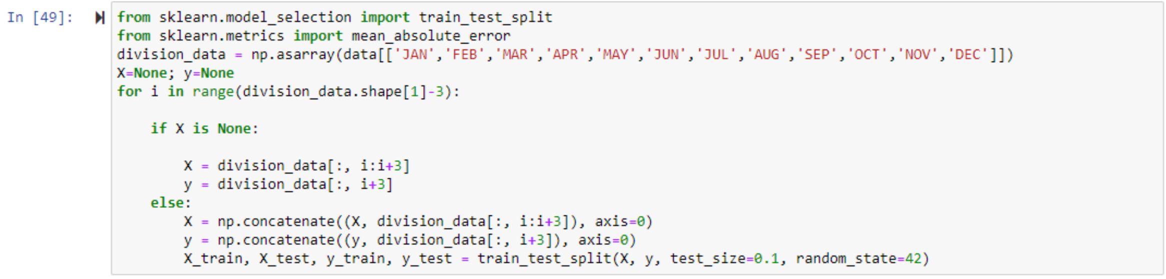
Out[33]:

	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
count	4116.000000	4116.000000	4116.000000	4116.000000	4116.000000	4116.000000	4116.000000	4116.000000	4116.000000	4116.000000	4116.000000
mean	1958.218659	18.957320	21.805325	27.359197	43.127432	85.745417	230.234444	347.214334	290.263497	197.361922	95.507009
std	33.140898	33.569044	35.896396	46.925176	67.798192	123.189974	234.568120	269.310313	188.678707	135.309591	99.434452
min	1901.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.400000	0.000000	0.000000	0.100000	0.000000
25%	1930.000000	0.600000	0.600000	1.000000	3.000000	8.600000	70.475000	175.900000	156.150000	100.600000	14.600000
50%	1958.000000	6.000000	6.700000	7.900000	15.700000	36.700000	138.900000	284.900000	259.500000	174.100000	65.750000
75%	1987.000000	22.125000	26.800000	31.225000	49.825000	96.825000	304.950000	418.225000	377.725000	265.725000	148.300000
max	2015.000000	583.700000	403.500000	605.600000	595.100000	1168.600000	1609.900000	2362.800000	1664.600000	1222.000000	948.300000

Describing the data shown in the above figure

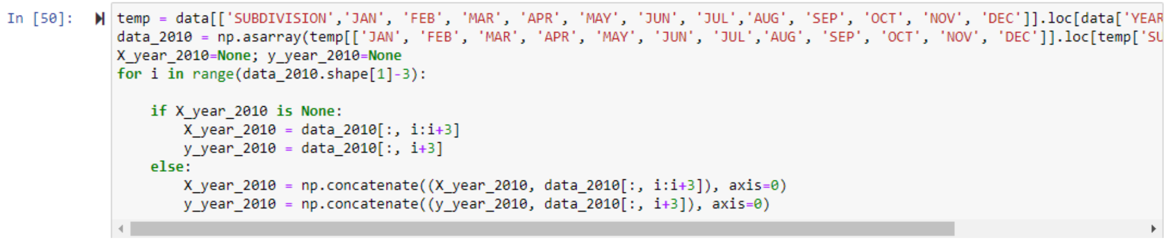
9.1 Analysis:

The Machine learning algorithms of linear regression and SVM algorithm



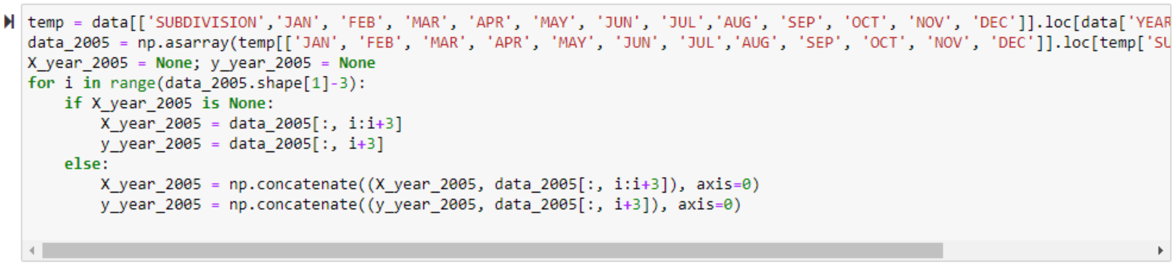
In [49]:

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error
division_data = np.asarray(data[['JAN','FEB','MAR','APR','MAY','JUN','JUL','AUG','SEP','OCT','NOV','DEC']])
X=None; y=None
for i in range(division_data.shape[1]-3):

    if X is None:

        X = division_data[:, i:i+3]
        y = division_data[:, i+3]
    else:
        X = np.concatenate((X, division_data[:, i:i+3]), axis=0)
        y = np.concatenate((y, division_data[:, i+3]), axis=0)
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)
```

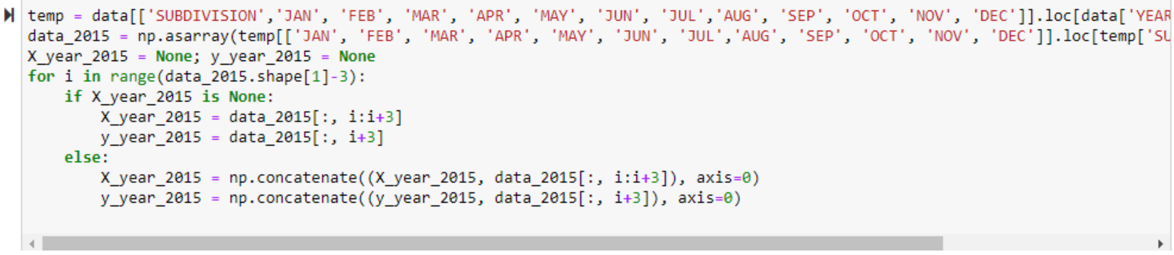


In [50]:

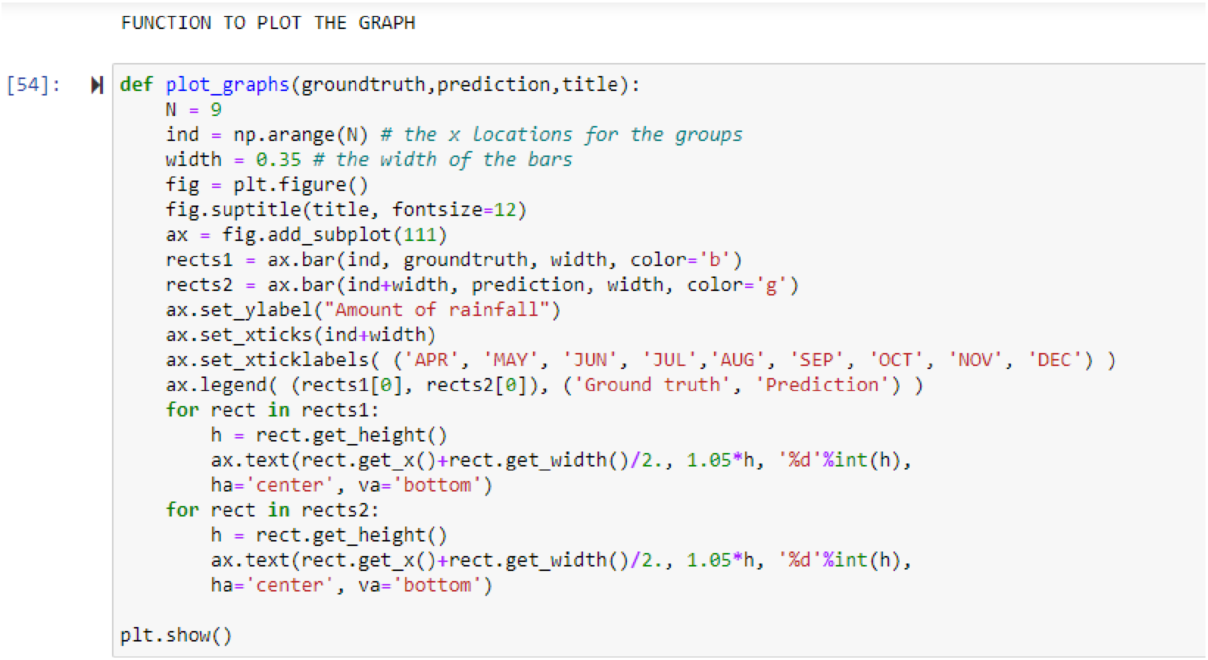
```
temp = data[['SUBDIVISION','JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL','AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[data['YEAR
data_2010 = np.asarray(temp[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL','AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[temp['SU
X_year_2010=None; y_year_2010=None
for i in range(data_2010.shape[1]-3):

    if X_year_2010 is None:
        X_year_2010 = data_2010[:, i:i+3]
        y_year_2010 = data_2010[:, i+3]
    else:
        X_year_2010 = np.concatenate((X_year_2010, data_2010[:, i:i+3]), axis=0)
        y_year_2010 = np.concatenate((y_year_2010, data_2010[:, i+3]), axis=0)
```



```
temp = data[['SUBDIVISION','JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL','AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[data['YEAR
data_2005 = np.asarray(temp[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL','AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[temp['SU
X_year_2005 = None; y_year_2005 = None
for i in range(data_2005.shape[1]-3):
    if X_year_2005 is None:
        X_year_2005 = data_2005[:, i:i+3]
        y_year_2005 = data_2005[:, i+3]
    else:
        X_year_2005 = np.concatenate((X_year_2005, data_2005[:, i:i+3]), axis=0)
        y_year_2005 = np.concatenate((y_year_2005, data_2005[:, i+3]), axis=0)
```

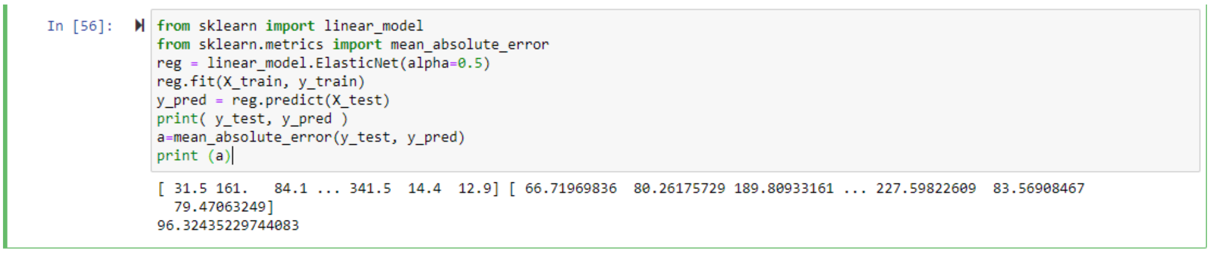


```
temp = data[['SUBDIVISION','JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL','AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[data['YEAR
data_2015 = np.asarray(temp[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL','AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[temp['SU
X_year_2015 = None; y_year_2015 = None
for i in range(data_2015.shape[1]-3):
    if X_year_2015 is None:
        X_year_2015 = data_2015[:, i:i+3]
        y_year_2015 = data_2015[:, i+3]
    else:
        X_year_2015 = np.concatenate((X_year_2015, data_2015[:, i:i+3]), axis=0)
        y_year_2015 = np.concatenate((y_year_2015, data_2015[:, i+3]), axis=0)
```



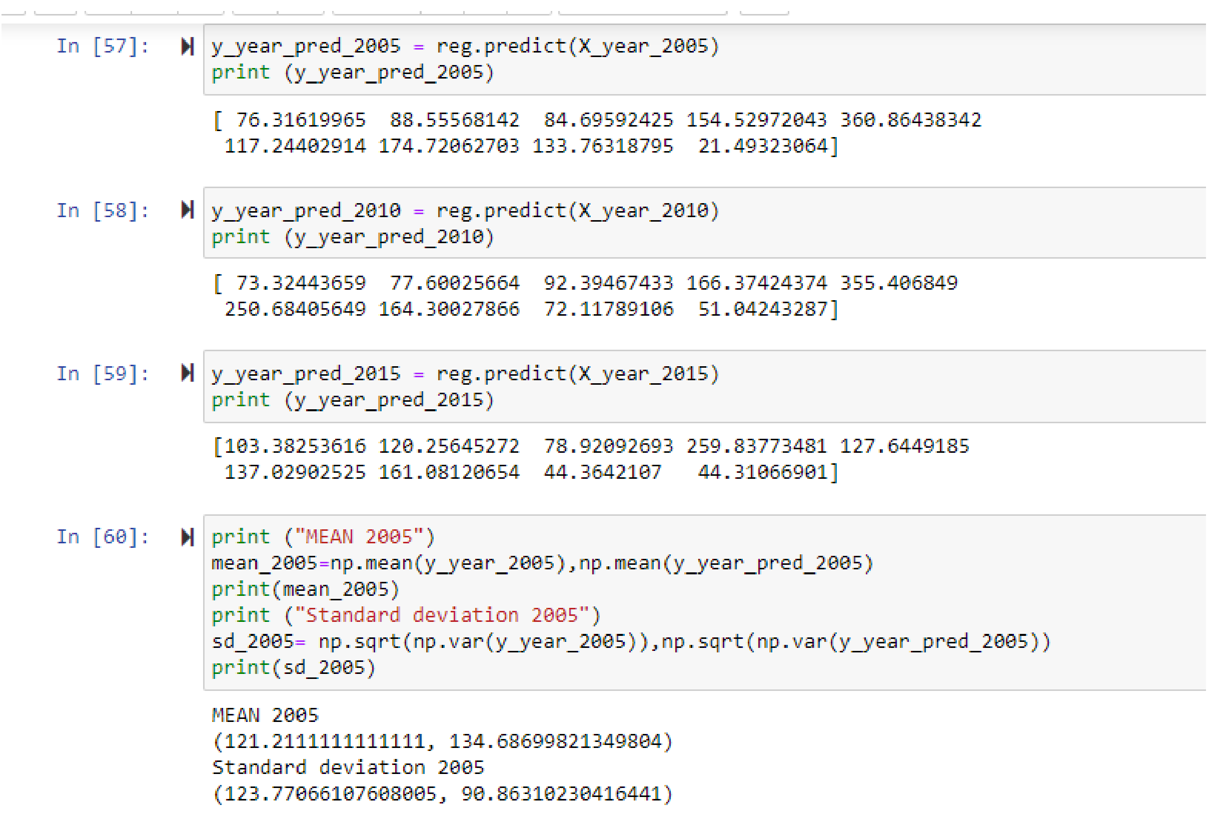
FUNCTION TO PLOT THE GRAPH

```
[54]:  def plot_graphs(groundtruth,prediction,title):
           N = 9
           ind = np.arange(N) # the x locations for the groups
           width = 0.35 # the width of the bars
           fig = plt.figure()
           fig.suptitle(title, fontsize=12)
           ax = fig.add_subplot(111)
           rects1 = ax.bar(ind, groundtruth, width, color='b')
           rects2 = ax.bar(ind+width, prediction, width, color='g')
           ax.set_ylabel("Amount of rainfall")
           ax.set_xticks(ind+width)
           ax.set_xticklabels( ('APR', 'MAY', 'JUN', 'JUL','AUG', 'SEP', 'OCT', 'NOV', 'DEC') )
           ax.legend( (rects1[0], rects2[0]), ('Ground truth', 'Prediction') )
           for rect in rects1:
               h = rect.get_height()
               ax.text(rect.get_x()+rect.get_width()/2., 1.05*h, '%d'%int(h),
               ha='center', va='bottom')
           for rect in rects2:
               h = rect.get_height()
               ax.text(rect.get_x()+rect.get_width()/2., 1.05*h, '%d'%int(h),
               ha='center', va='bottom')

       plt.show()
```

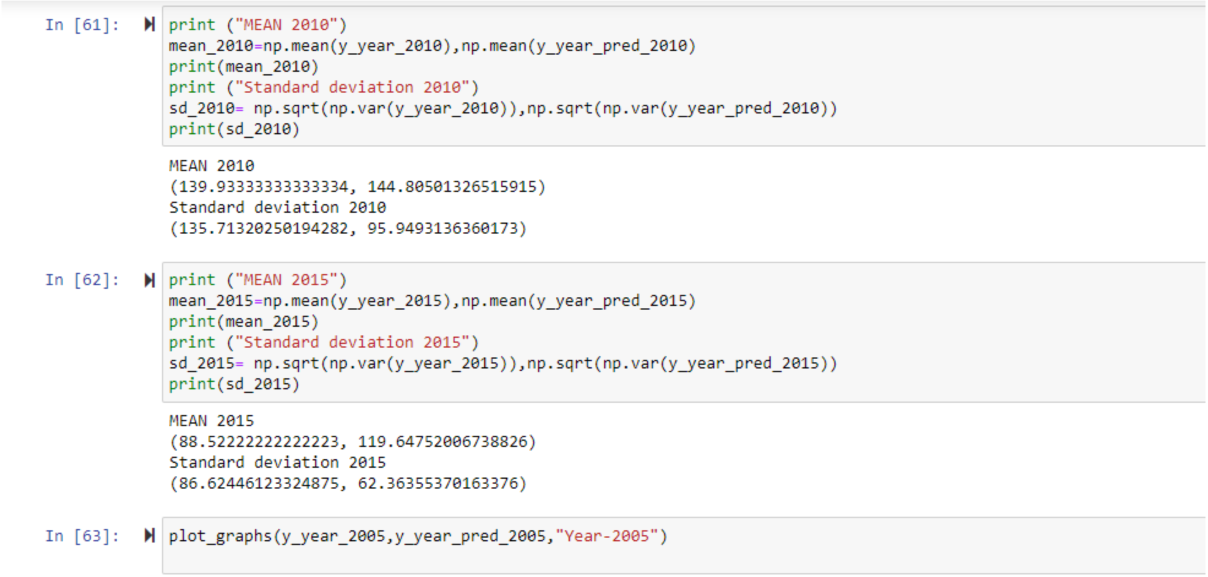


```
In [56]:  from sklearn import linear_model
          from sklearn.metrics import mean_absolute_error
          reg = linear_model.ElasticNet(alpha=0.5)
          reg.fit(X_train, y_train)
          y_pred = reg.predict(X_test)
          print( y_test, y_pred )
          a=mean_absolute_error(y_test, y_pred)
          print (a)

          [ 31.5 161.   84.1 ... 341.5  14.4  12.9] [ 66.71969836  80.26175729 189.80933161 ... 227.59822609  83.56908467
            79.47063249]
          96.32435229744083
```

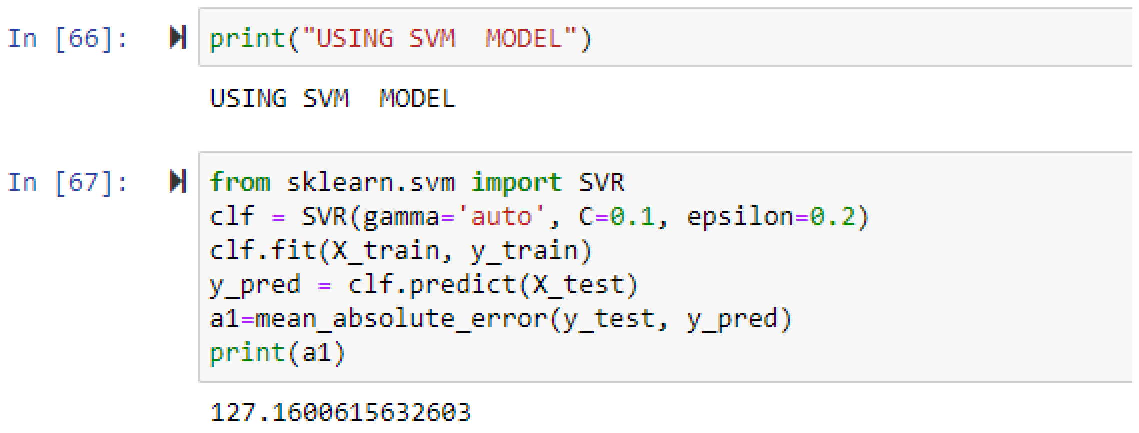
Training of the data set is done for the gap of the 5 years from the 2000-2015 from the data set and had predicted the accuracy of the algorithm in the testing set of the data from the data set so that we can predict for the future calculations and here we trained the data for 2005,2010 and 2015 using the Linear regression and SVM algorithms



```
In [57]:  y_year_pred_2005 = reg.predict(X_year_2005)
          print (y_year_pred_2005)

          [ 76.31619965  88.55568142  84.69592425 154.52972043 360.86438342
           117.24402914 174.72062703 133.76318795  21.49323064]

In [58]:  y_year_pred_2010 = reg.predict(X_year_2010)
          print (y_year_pred_2010)

          [ 73.32443659  77.60025664  92.39467433 166.37424374 355.406849
           250.68405649 164.30027866  72.11789106  51.04243287]

In [59]:  y_year_pred_2015 = reg.predict(X_year_2015)
          print (y_year_pred_2015)

          [103.38253616 120.25645272  78.92092693 259.83773481 127.6449185
           137.02902525 161.08120654  44.3642107   44.31066901]

In [60]:  print ("MEAN 2005")
          mean_2005=np.mean(y_year_2005),np.mean(y_year_pred_2005)
          print(mean_2005)
          print ("Standard deviation 2005")
          sd_2005= np.sqrt(np.var(y_year_2005)),np.sqrt(np.var(y_year_pred_2005))
          print(sd_2005)

          MEAN 2005
          (121.2111111111111, 134.68699821349804)
          Standard deviation 2005
          (123.77066107608005, 90.86310230416441)
```



```
In [61]:  print ("MEAN 2010")
          mean_2010=np.mean(y_year_2010),np.mean(y_year_pred_2010)
          print(mean_2010)
          print ("Standard deviation 2010")
          sd_2010= np.sqrt(np.var(y_year_2010)),np.sqrt(np.var(y_year_pred_2010))
          print(sd_2010)

          MEAN 2010
          (139.93333333333334, 144.80501326515915)
          Standard deviation 2010
          (135.71320250194282, 95.9493136360173)

In [62]:  print ("MEAN 2015")
          mean_2015=np.mean(y_year_2015),np.mean(y_year_pred_2015)
          print(mean_2015)
          print ("Standard deviation 2015")
          sd_2015= np.sqrt(np.var(y_year_2015)),np.sqrt(np.var(y_year_pred_2015))
          print(sd_2015)

          MEAN 2015
          (88.52222222222223, 119.64752006738826)
          Standard deviation 2015
          (86.62446123324875, 62.36355370163376)

In [63]:  plot_graphs(y_year_2005,y_year_pred_2005,"Year-2005")
```



```
In [66]:  print("USING SVM  MODEL")

          USING SVM  MODEL

In [67]:  from sklearn.svm import SVR
          clf = SVR(gamma='auto', C=0.1, epsilon=0.2)
          clf.fit(X_train, y_train)
          y_pred = clf.predict(X_test)
          a1=mean_absolute_error(y_test, y_pred)
          print(a1)

          127.1600615632603
```

9.2 Visualization:

Visualization of total years

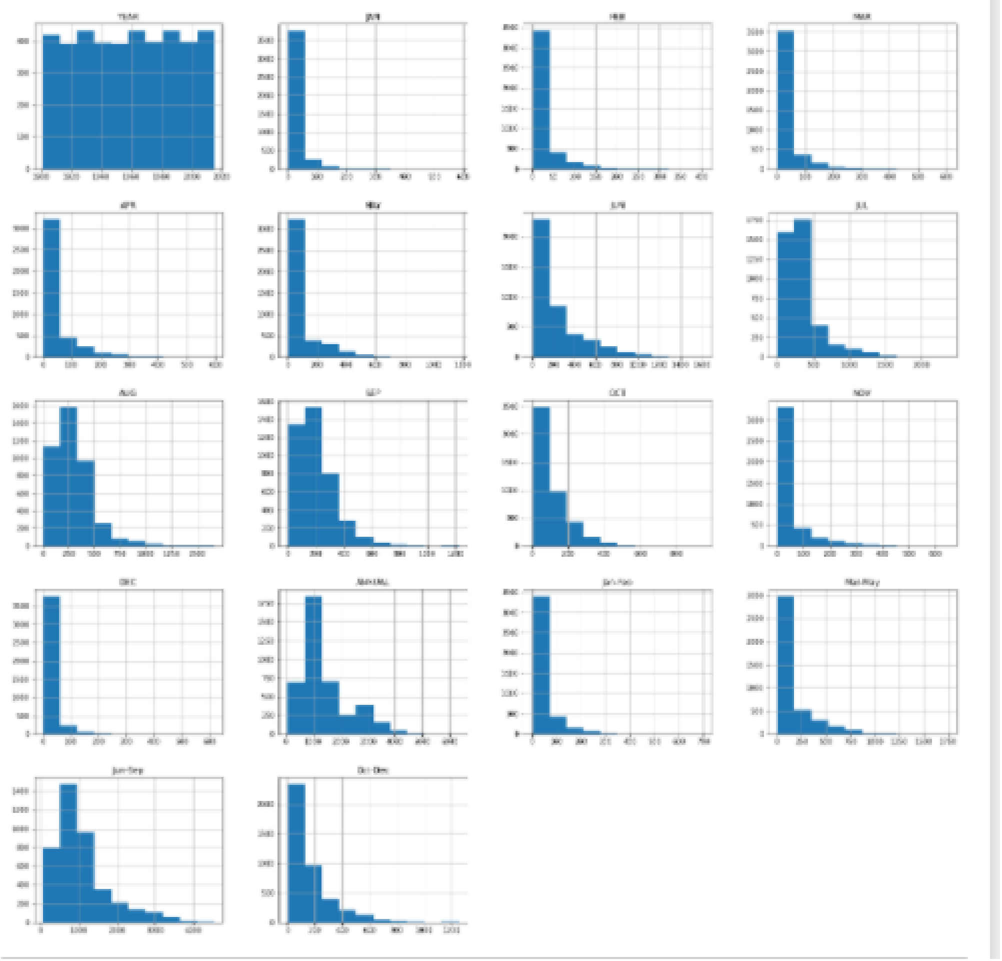
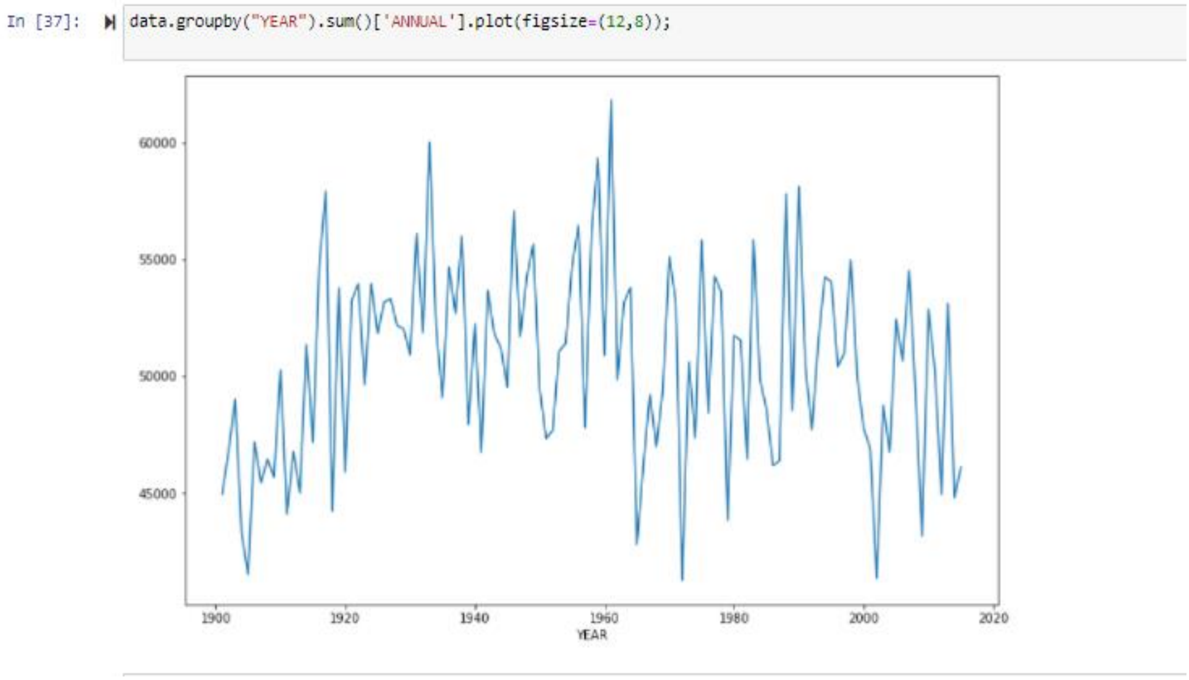


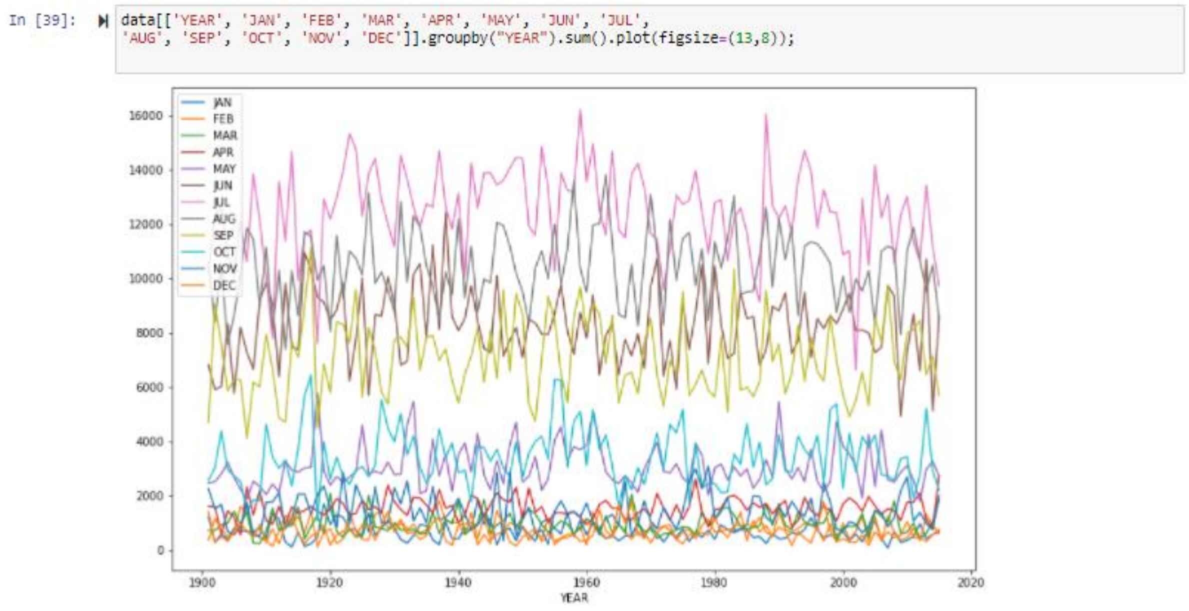
Fig-1.7



```
In [37]:  data.groupby("YEAR").sum()['ANNUAL'].plot(figsize=(12,8));
```

Fig-1.8

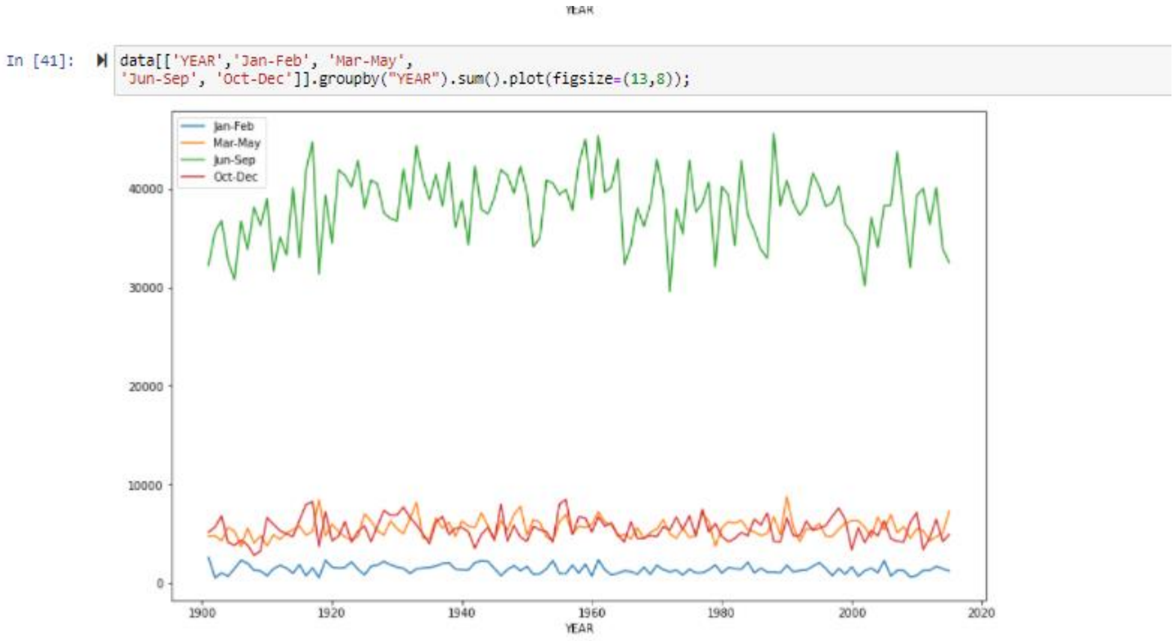
Linear graph for entire months



```
In [39]:  data[['YEAR', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
          'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].groupby("YEAR").sum().plot(figsize=(13,8));
```

Fig-1.9

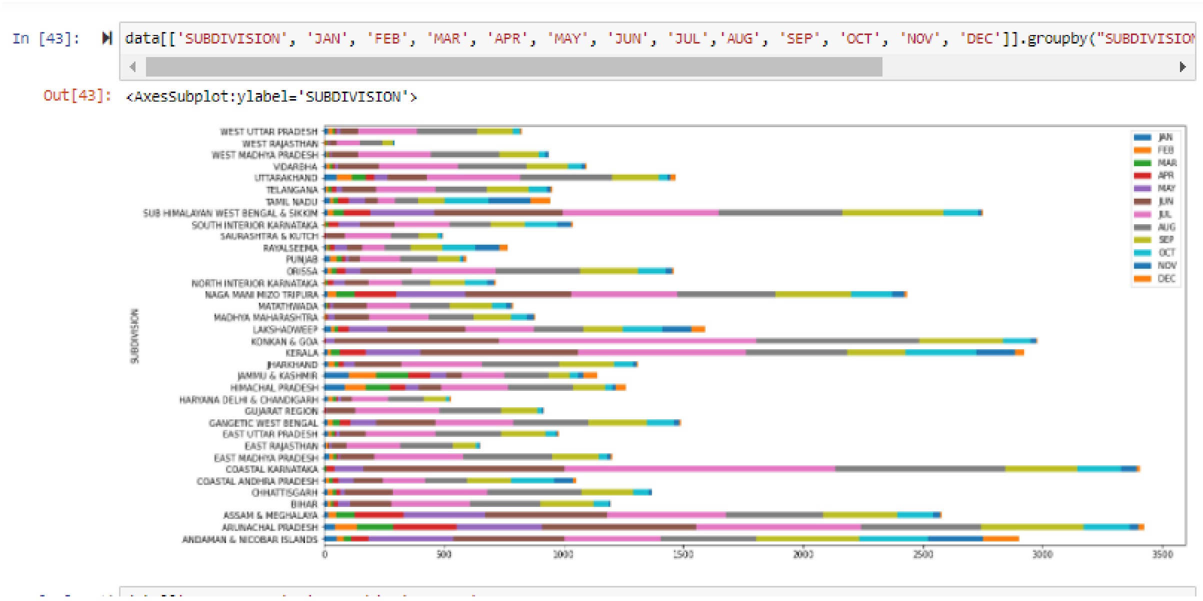
Plotting the data is shown for all the years and for all the months



```
In [41]: data[['YEAR','Jan-Feb', 'Mar-May',
         'Jun-Sep', 'Oct-Dec']].groupby("YEAR").sum().plot(figsize=(13,8));
```

Fig-2.0

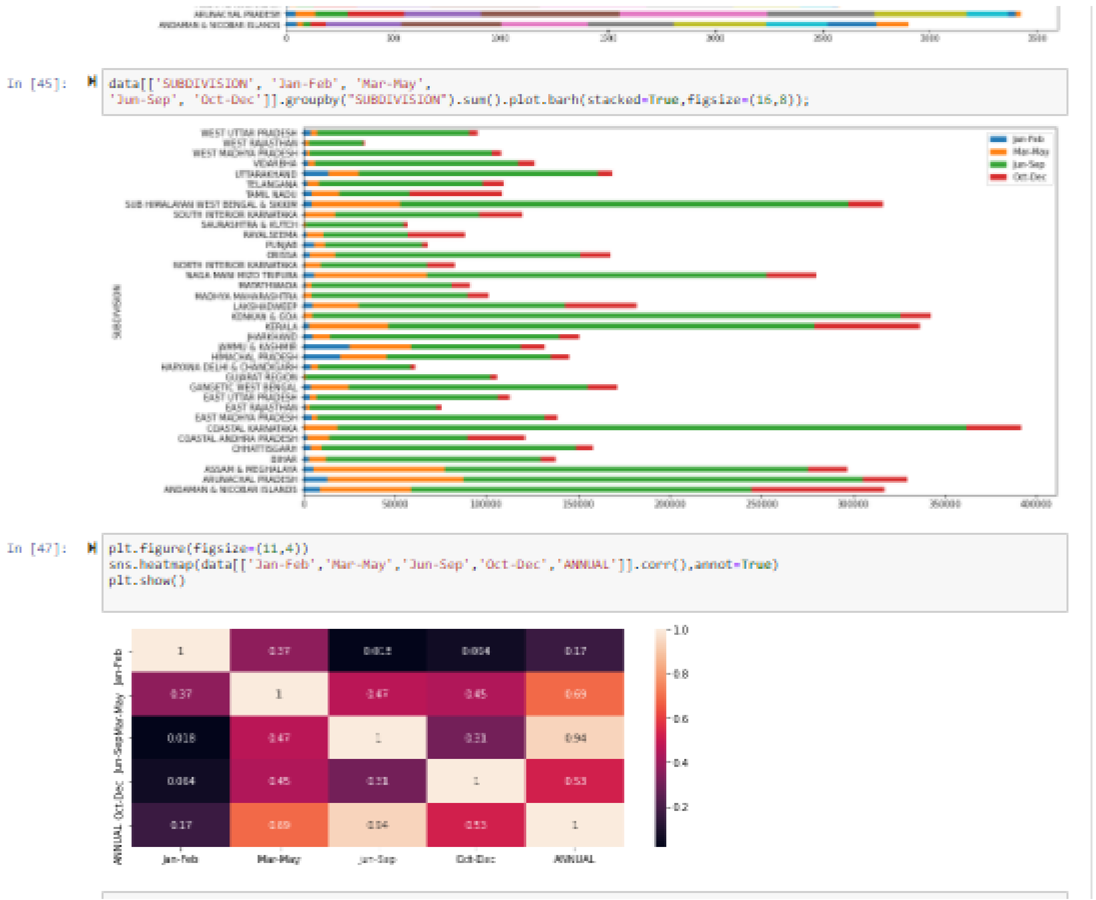
Subplot of all states in different years



```
In [43]: data[['SUBDIVISION', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL','AUG', 'SEP', 'OCT', 'NOV', 'DEC']].groupby("SUBDIVISION

Out[43]: <AxesSubplot:ylabel='SUBDIVISION'>
```

Fig-2.1

Plot figure



In [45]:

```
data[['SUBDIVISION', 'Jan-Feb', 'Mar-May',
'Jun-Sep', 'Oct-Dec']].groupby("SUBDIVISION").sum().plot.barh(stacked=True,figsize=(16,8));
```

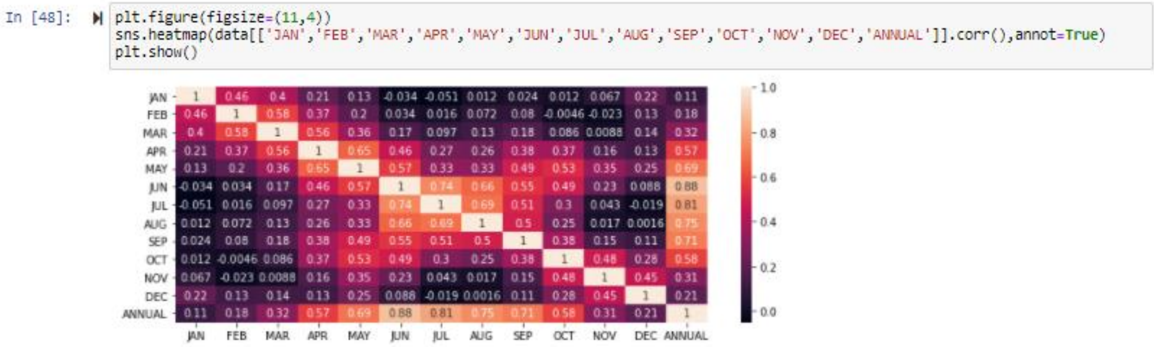
In [47]:

```
plt.figure(figsize=(11,4))
sns.heatmap(data[['Jan-Feb','Mar-May','Jun-Sep','Oct-Dec','ANNUAL']].corr(),annot=True)
plt.show()
```

Jan-Feb 1 0.37 0.018 0.064 0.17
Mar-May 0.37 1 0.47 0.45 0.69
Jun-Sep 0.018 0.47 1 0.31 0.94
Oct-Dec 0.064 0.45 0.31 1 0.53
ANNUAL 0.17 0.69 0.94 0.53 1
Jan-Feb
Mar-May
Jun-Sep
Oct-Dec
ANNUAL
1.0
0.8
0.6
0.4
0.2

Fig-2.2

Plot figure of all different months



In [48]:

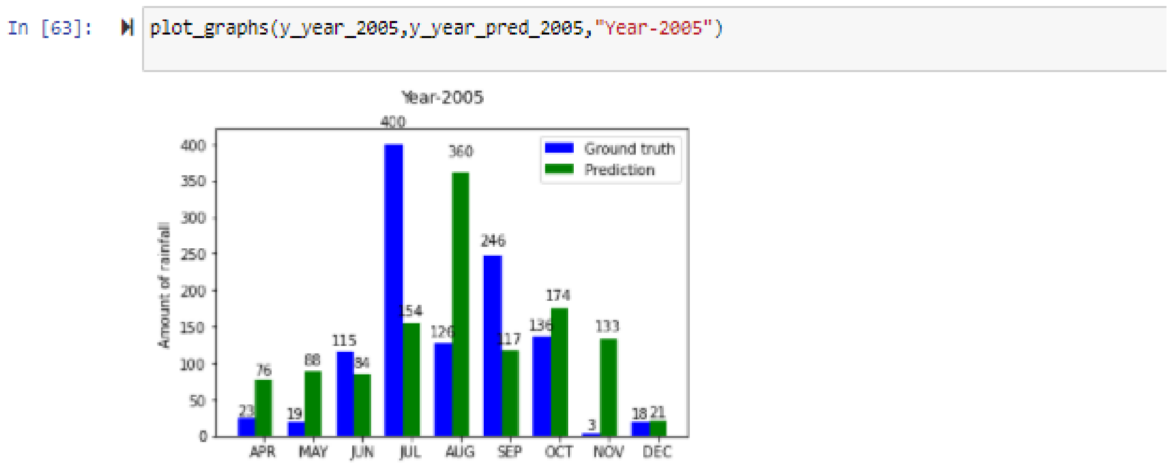
```
plt.figure(figsize=(11,4))
sns.heatmap(data[['JAN','FEB','MAR','APR','MAY','JUN','JUL','AUG','SEP','OCT','NOV','DEC','ANNUAL']].corr(),annot=True)
plt.show()
```

JAN 1 0.46 0.4 0.21 0.13 -0.034 -0.051 0.012 0.024 0.012 0.067 0.22 0.11
FEB 0.46 1 0.58 0.37 0.2 0.034 0.016 0.072 0.08 -0.0046 -0.023 0.13 0.18
MAR 0.4 0.58 1 0.56 0.36 0.17 0.097 0.13 0.18 0.086 0.0088 0.14 0.32
APR 0.21 0.37 0.56 1 0.65 0.46 0.27 0.26 0.38 0.37 0.16 0.13 0.57
MAY 0.13 0.2 0.36 0.65 1 0.57 0.33 0.33 0.49 0.53 0.35 0.25 0.69
JUN -0.034 0.034 0.17 0.46 0.57 1 0.74 0.66 0.55 0.49 0.23 0.088 0.88
JUL -0.051 0.016 0.097 0.27 0.33 0.74 1 0.69 0.51 0.3 0.043 -0.019 0.81
AUG 0.012 0.072 0.13 0.26 0.33 0.66 0.69 1 0.5 0.25 0.017 0.0016 0.75
SEP 0.024 0.08 0.18 0.38 0.49 0.55 0.51 0.5 1 0.38 0.15 0.11 0.71
OCT 0.012 -0.0046 0.086 0.37 0.53 0.49 0.3 0.25 0.38 1 0.48 0.28 0.58
NOV 0.067 -0.023 0.0088 0.16 0.35 0.23 0.043 0.017 0.15 0.48 1 0.45 0.31
DEC 0.22 0.13 0.14 0.13 0.25 0.088 -0.019 0.0016 0.11 0.28 0.45 1 0.21
ANNUAL 0.11 0.18 0.32 0.57 0.69 0.88 0.81 0.75 0.71 0.58 0.31 0.21 1
JAN
FEB
MAR
APR
MAY
JUN
JUL
AUG
SEP
OCT
NOV
DEC
ANNUAL
1.0
0.8
0.6
0.4
0.2
0.0

Fig-2.3

Plotting the graph using the linear regression model in the year of 2005 gives as: After the training and testing of data there is the comparison plot graph between the predicted value and the ground truth value here are the images of the prediction and ground truth value

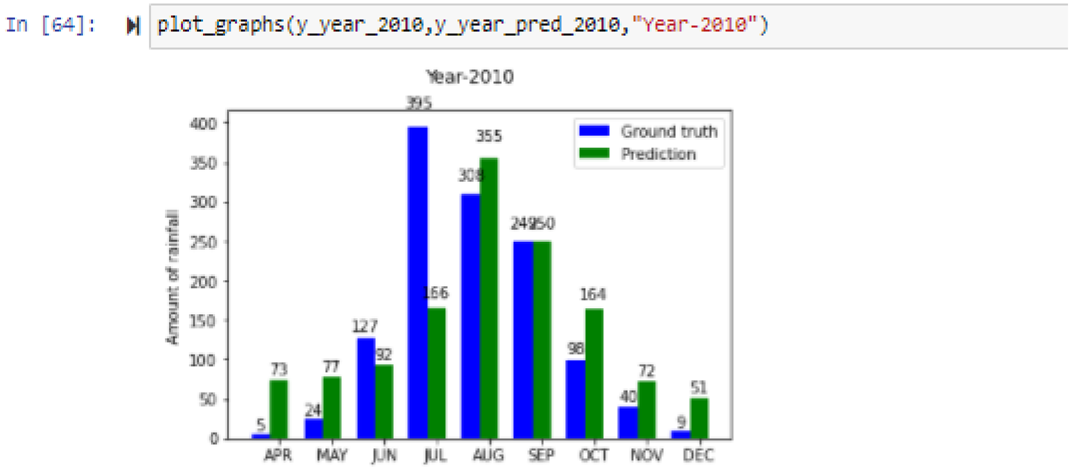
Comparison of ground and prediction in 2005



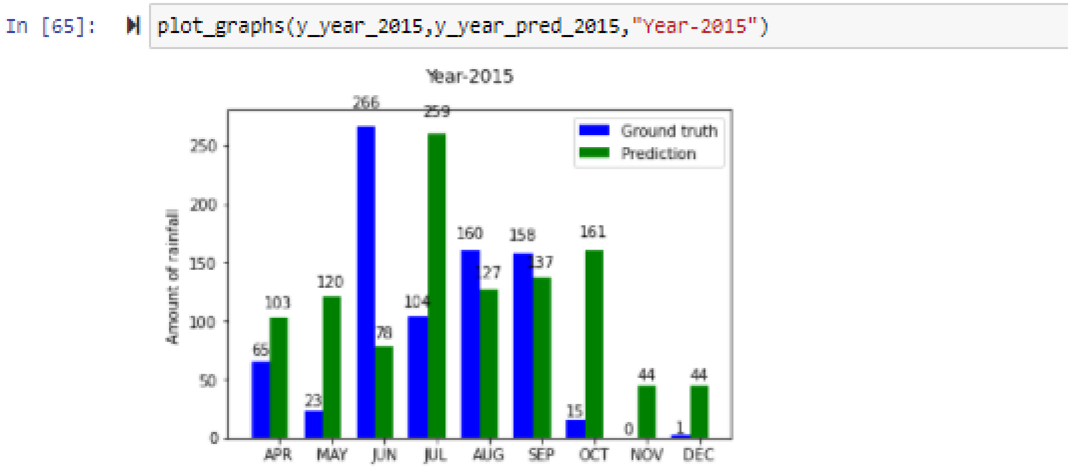
```
In [63]:  plot_graphs(y_year_2005,y_year_pred_2005,"Year-2005")
```

Fig-2.5

Plotting the graph using the linear regression model in the year of 2010 and 2015 gives as: **Comparison between ground and prediction in 2010 and 2015**



```
In [64]:  plot_graphs(y_year_2010,y_year_pred_2010,"Year-2010")
```



```
In [65]:  plot_graphs(y_year_2015,y_year_pred_2015,"Year-2015")
```

Fig-2.6

9.3 Website:

This is the sign-up page in which the new users should be fill the details and the details will be stored in the database **Sign-up**

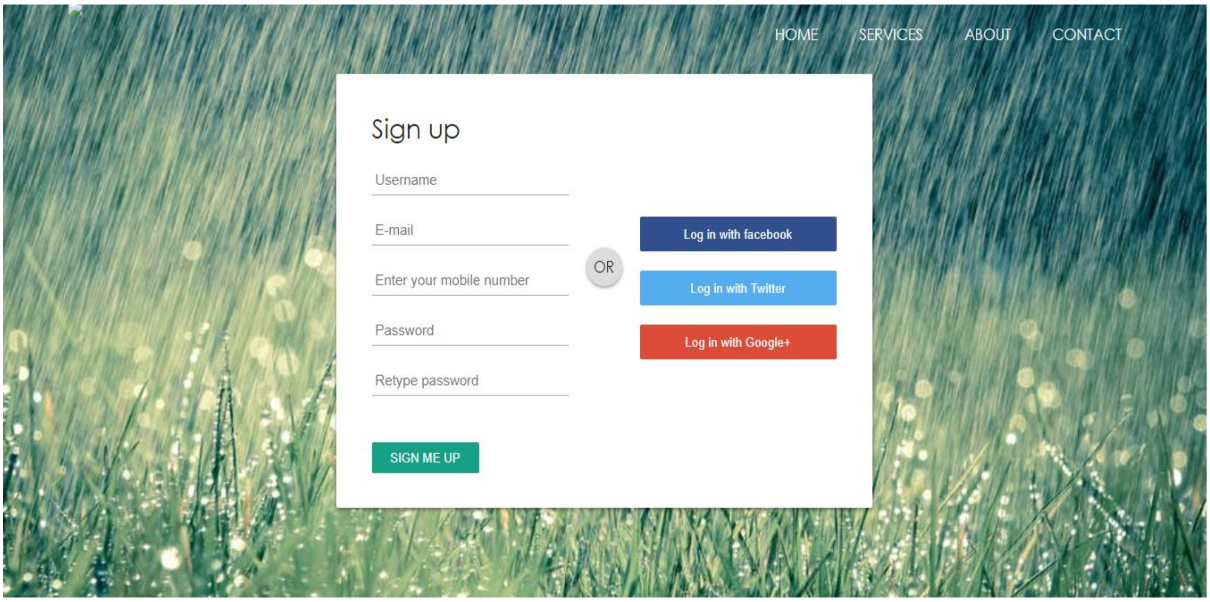


Fig-2.7

This is the login page in which the old users can directly enter the login credentials to view the home page **Login**

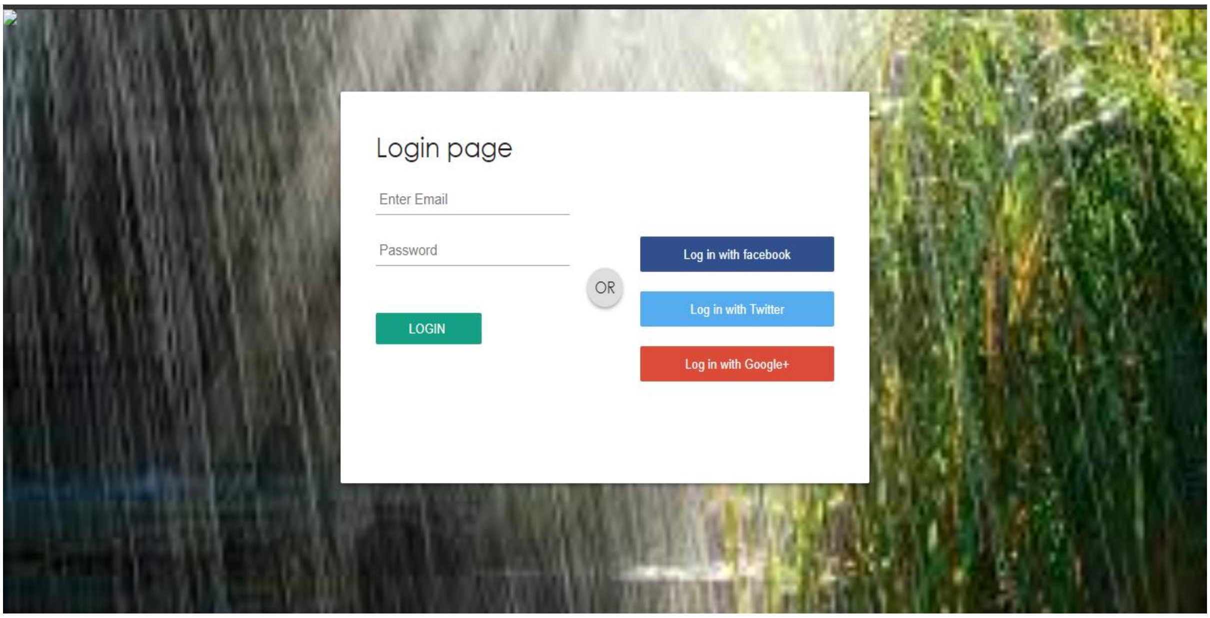
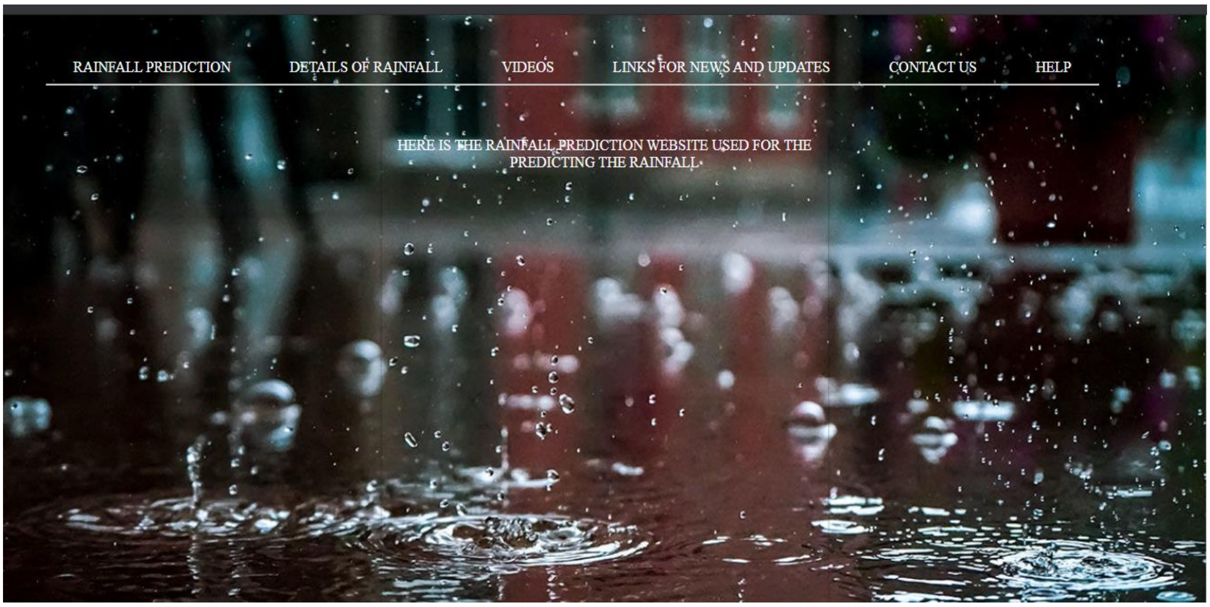


Fig-2.8

This is homepage in which we have 6 modules there is rainfall prediction modules which describes the percentage of rain by comparing the algorithms and in the details of rainfall we will be giving the details of rainfall across India and in the videos we will have the video management system and in the contact us and help us pages will be describing our details of project and members involved in that

Home

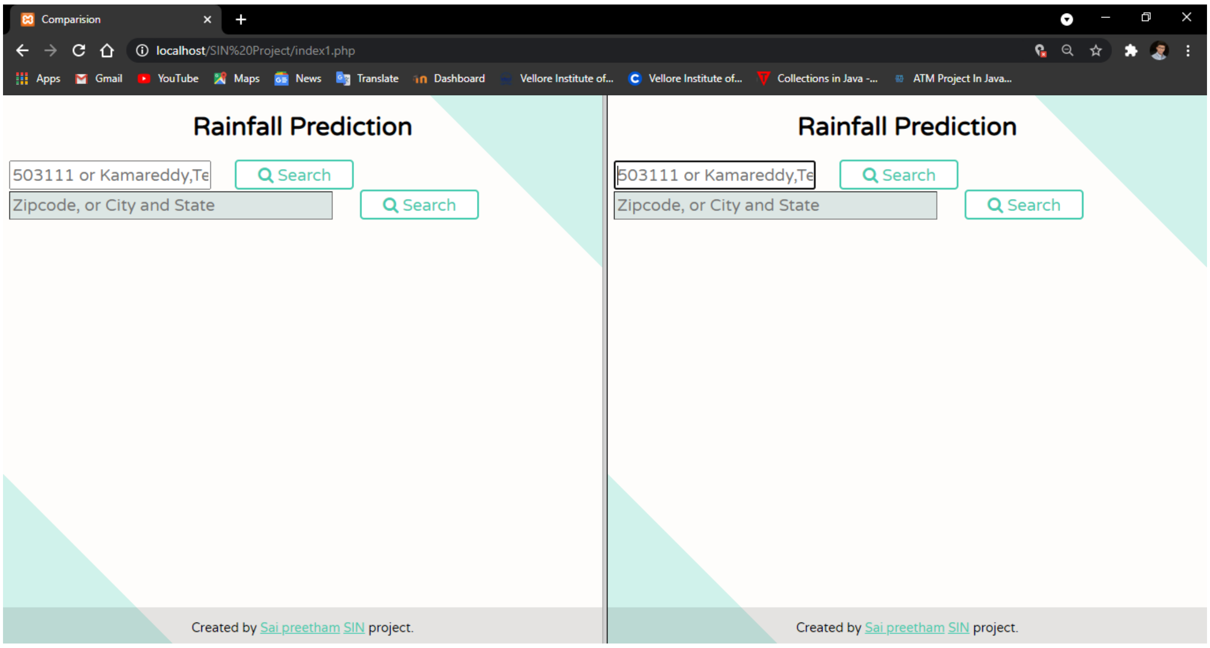


HERE IS THE RAINFALL PREDICTION WEBSITE USED FOR THE PREDICTING THE RAINFALL

Fig-2.9

Rainfall prediction page compares the linear regression and the ARIMA machine learning algorithm:

Rainfall prediction



Rainfall Prediction

503111 or Kamareddy,Te

Zipcode, or City and State

Created by Sai preetham SIN project.

Rainfall Prediction

503111 or Kamareddy,Te

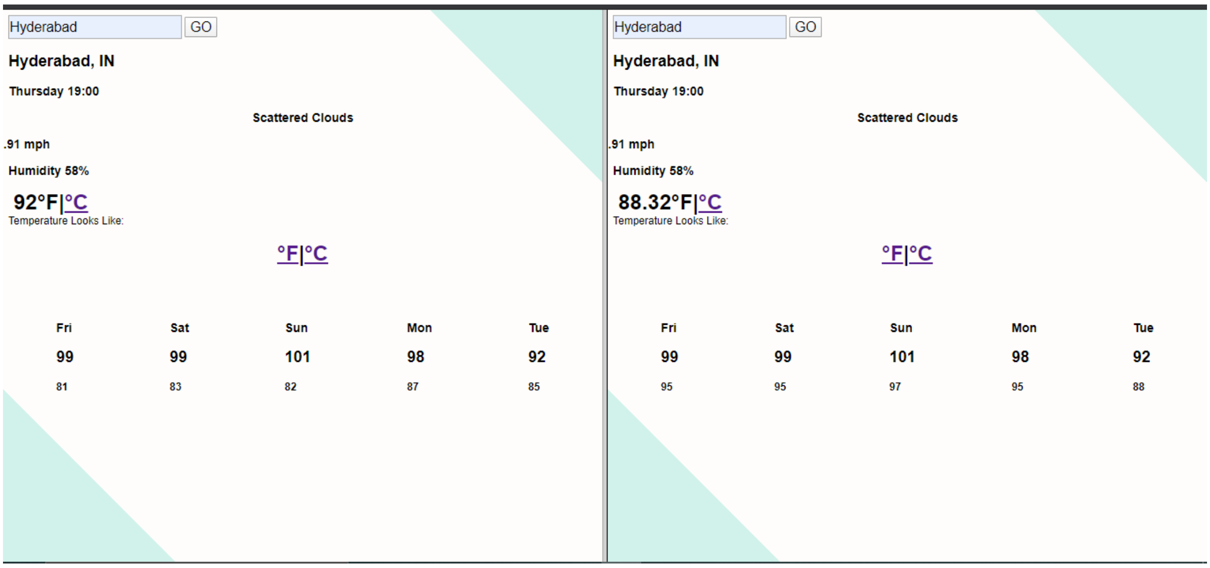
Zipcode, or City and State

Created by Sai preetham SIN project.

Fig-3.0

Comparison between the linear regression model and ARIMA model(Automated regression integrated moving average):

Rainfall prediction in Hyderabad



Hyderabad

Hyderabad, IN

Thursday 19:00

Scattered Clouds

.91 mph

Humidity 58%

92°F|°C

Temperature Looks Like:

°F|°C

Fri	Sat	Sun	Mon	Tue
99	99	101	98	92
81	83	82	87	85

Hyderabad

Hyderabad, IN

Thursday 19:00

Scattered Clouds

.91 mph

Humidity 58%

88.32°F|°C

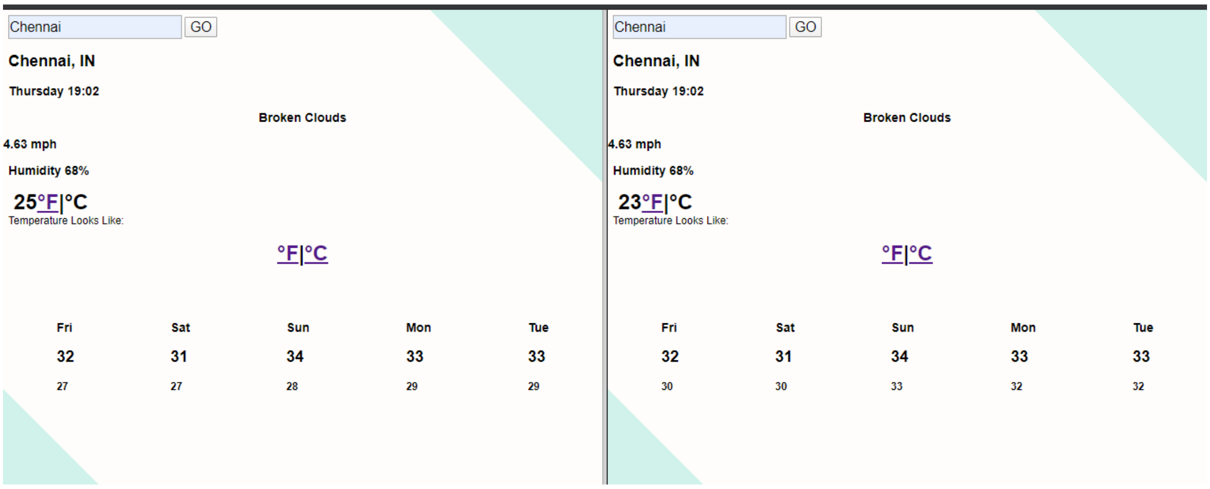
Temperature Looks Like:

°F|°C

Fri	Sat	Sun	Mon	Tue
99	99	101	98	92
95	95	97	95	88

Fig-3.1

Rainfall prediction in Chennai



Chennai GO

Chennai, IN

Thursday 19:02

Broken Clouds

4.63 mph

Humidity 68%

25°F|°C

Temperature Looks Like:

°F|°C

Fri	Sat	Sun	Mon	Tue
32	31	34	33	33
27	27	28	29	29

Chennai GO

Chennai, IN

Thursday 19:02

Broken Clouds

4.63 mph

Humidity 68%

23°F|°C

Temperature Looks Like:

°F|°C

Fri	Sat	Sun	Mon	Tue
32	31	34	33	33
30	30	33	32	32

Fig-3.2

Details of rainfall page will describes the details of the all districts and states in the India about the rainfall statistics: **Details of Rainfall**

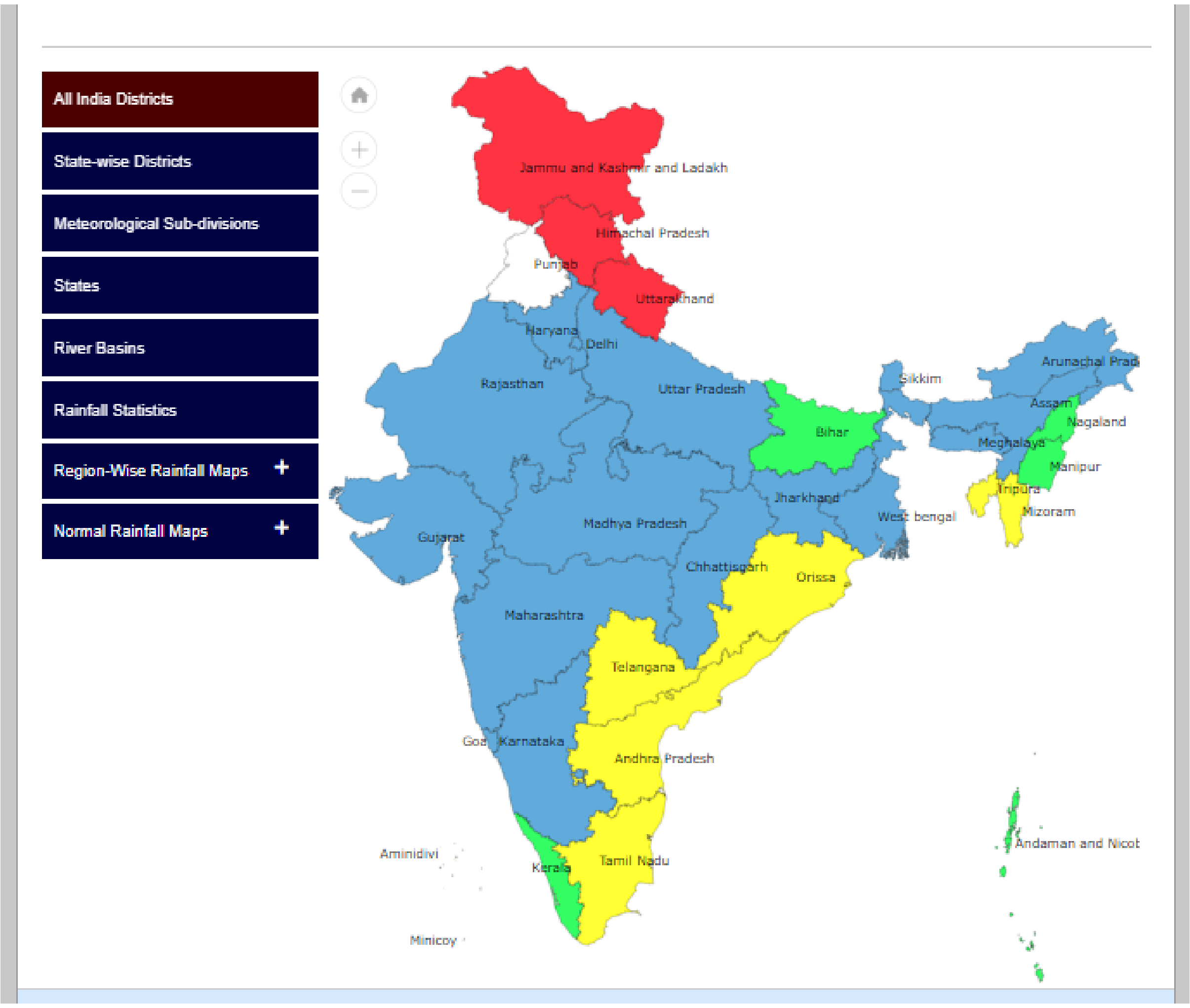


Fig-3.3

Video management system helps to view the live forecast and live streaming videos across the globe: **Videos-Management**

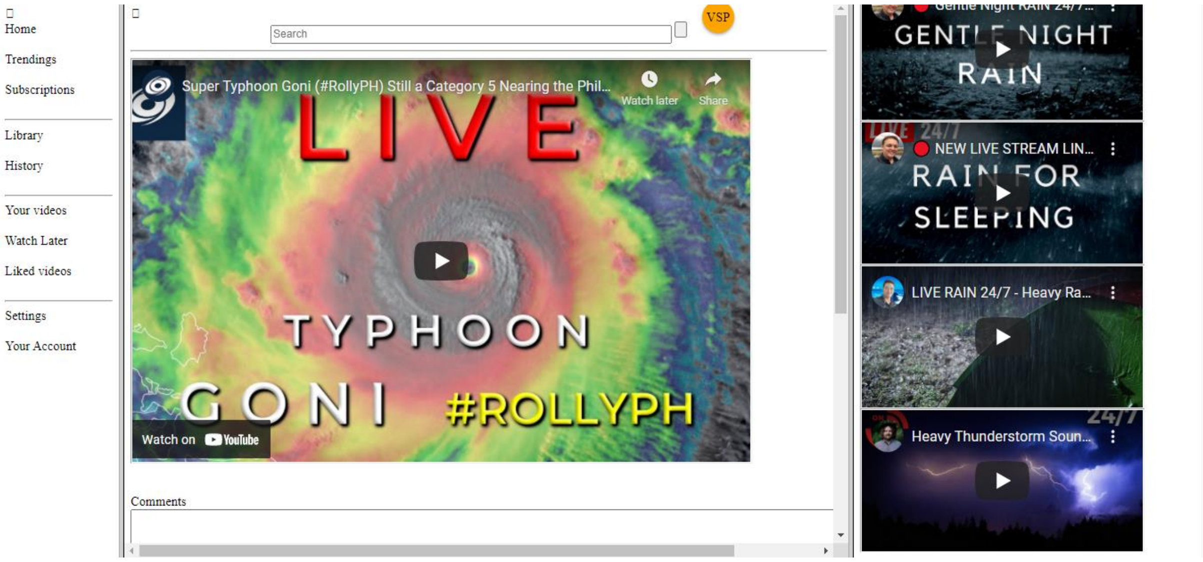
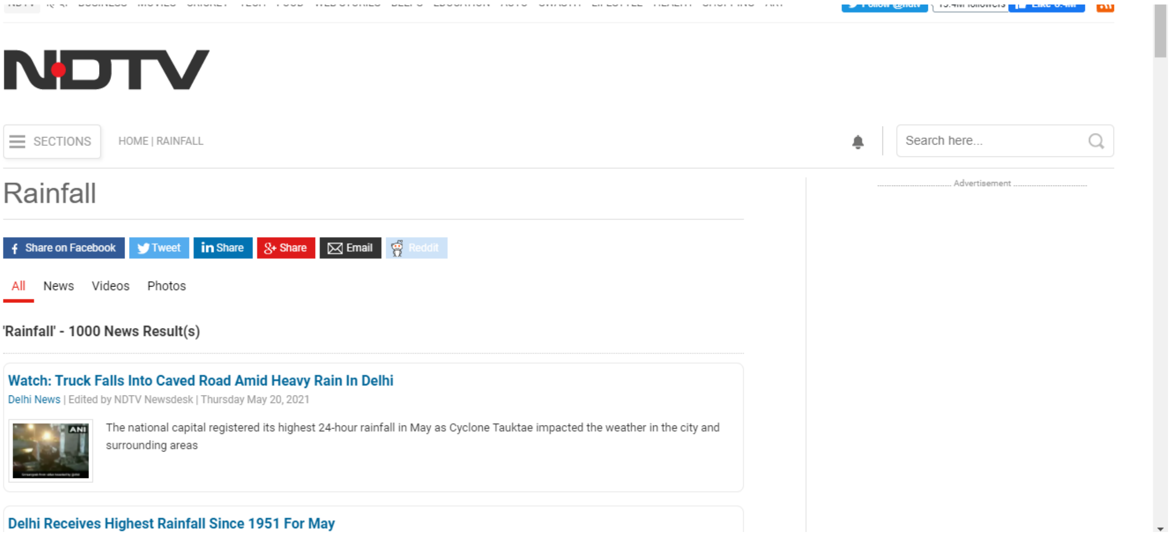


Fig-3.4

News and updates for the rainfall will helps to find the news and updates of the rainfall every day and every hour so that the users can get more knowledge about the rainfall in the globally passion: **News and updates**



Rainfall

'Rainfall' - 1000 News Result(s)

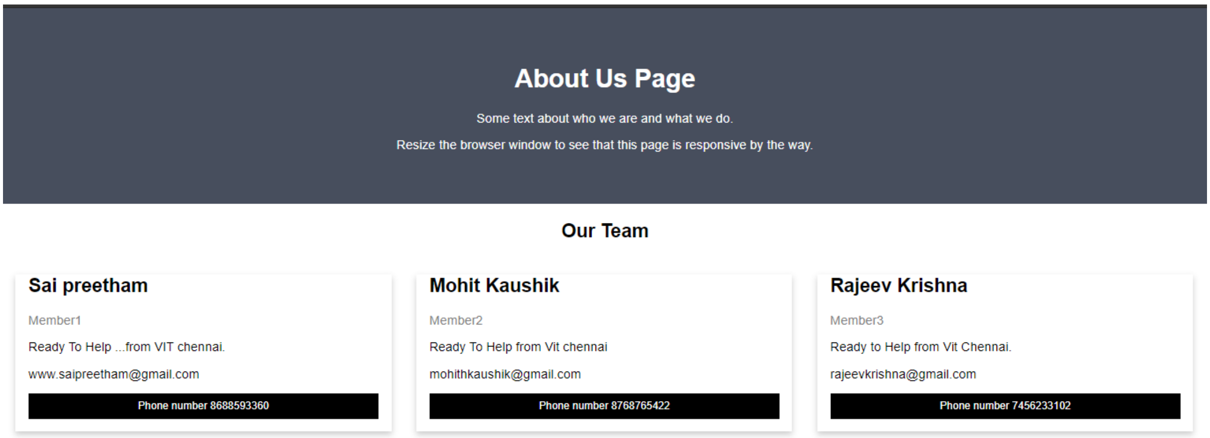
Watch: Truck Falls Into Caved Road Amid Heavy Rain In Delhi
Delhi News | Edited by NDTV Newsdesk | Thursday May 20, 2021

The national capital registered its highest 24-hour rainfall in May as Cyclone Tauktae impacted the weather in the city and surrounding areas

Delhi Receives Highest Rainfall Since 1951 For May

Fig-3.5

Contact us page describes the team members worked and also had the phone numbers to contact them: **Contact us**



About Us Page

Some text about who we are and what we do.

Resize the browser window to see that this page is responsive by the way.

Our Team

Sai preetham

Member1

Ready To Help ...from VIT chennai.

www.saipreetham@gmail.com

Phone number 8688593360

Mohit Kaushik

Member2

Ready To Help from Vit chennai

mohithkaushik@gmail.com

Phone number 8768765422

Rajeev Krishna

Member3

Ready to Help from Vit Chennai.

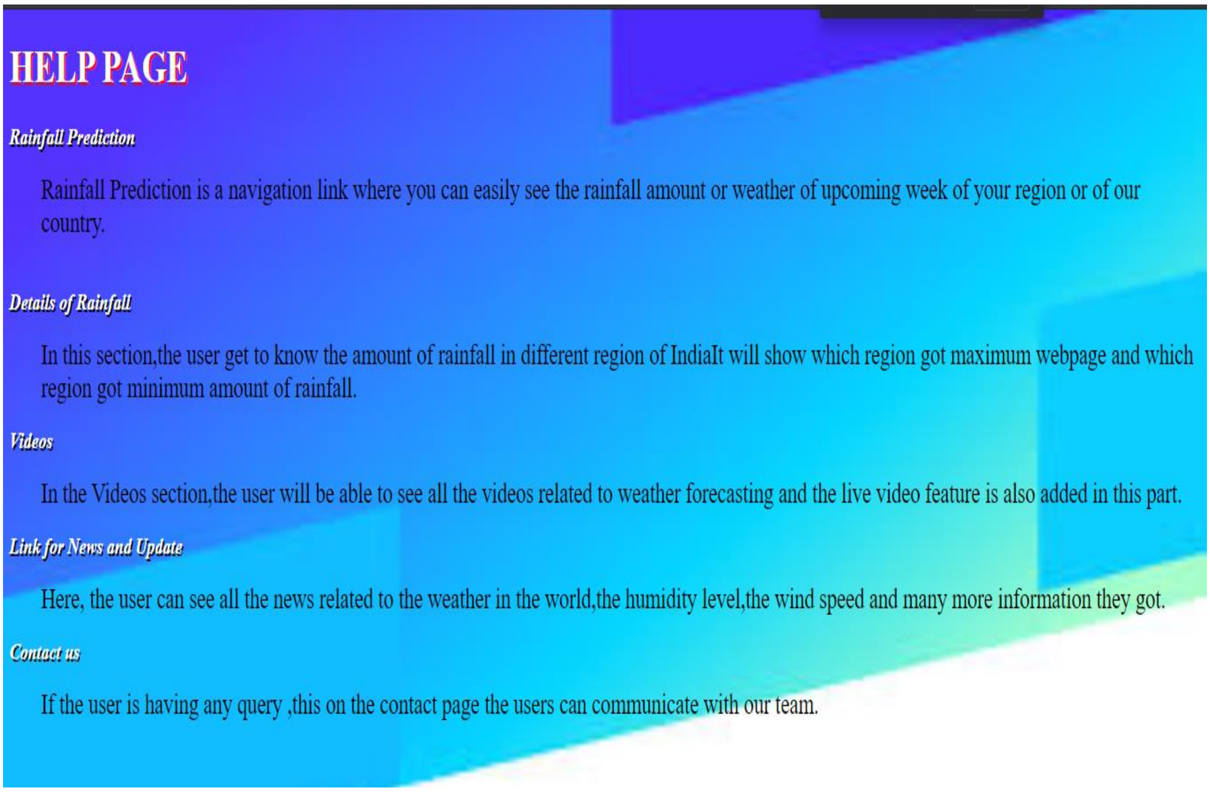
rajeevkrishna@gmail.com

Phone number 7456233102

Fig-3.6

Help page will describes the overview of the project

Help



HELP PAGE

Rainfall Prediction

Rainfall Prediction is a navigation link where you can easily see the rainfall amount or weather of upcoming week of your region or of our country.

Details of Rainfall

In this section,the user get to know the amount of rainfall in different region of IndiaIt will show which region got maximum webpage and which region got minimum amount of rainfall.

Videos

In the Videos section,the user will be able to see all the videos related to weather forecasting and the live video feature is also added in this part.

Link for News and Update

Here, the user can see all the news related to the weather in the world,the humidity level,the wind speed and many more information they got.

Contact us

If the user is having any query ,this on the contact page the users can communicate with our team.

Fig-3.7

10. Conclusion:

Machine learning algorithms like Linear regression and Arima algorithm will be helping for the Website for the rainfall prediction using the machine learning algorithms like linear regression, SVM, Self-organisation Map Machine learning algorithms hence it would be useful for the farmers and the government so that they can be predict the rainfall which are useful for the cultivation of crops and finding the floods in the coastal area ,hence the machine learning algorithms will be very useful website and important for the farmers and the government

11.References:

1.M.Kannan,S.Prabhakaran,P.Ramachandran,"Rainfall Forecasting using Machine Learning technique".

2.Shoba G,DR.Shobha G,"Rain fall prediction using Machine Learning techniques."

3.MR.M.S.Chaudhari,DR.N.K.Choudhari,"Study of Various rainfall estimation &prediction using Machine Learning techniques."

4.MR.DHAWAL HIRANI, DR.NITIN MISHRA,"A survey on rainfall prediction techniques."