# PROFESSIONAL TRAINING REPORT

**at**

**Sathyabama Institute of Science and Technology (Deemed to be University)**

Submitted in partial fulfillment of the requirements for the award of

Bachelor of Engineering Degree in Computer Science and Engineering

By

## V.L.V.PRAYUSH

**REG. NO. 39110800**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SCHOOL OF COMPUTING**

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,**

**CHENNAI – 600119, TAMILNADU**

**APRIL 2022**

|  |  |  |
| --- | --- | --- |
|  | **SATHYABAMA**  **INSTITUTE OF SCIENCE AND TECHNOLOGY** (DEEMED TO BE UNIVERSITY) **Accredited with Grade “A” by NAAC**  (Established under Section 3 of UGC Act, 1956)  JEPPIAAR NAGAR, RAJIV GANDHI SALAI  CHENNAI– 600119  [**www.sathyabama.ac.in**](http://www.sathyabama.ac.in) |  |

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of **V.L.V.PRAYUSH**(**Reg. No: 39110800)** who carried out the project entitled “**Rainfall prediction using machine learning**” under my supervision from January 2022 to April2022.

## Internal Guide

## Dr.S.Dhamodaran, M.E., Ph.D.,

**Head of the Department**

**Dr. L. Lakshmanan, M.E., Ph.D.,**



## Submitted for Viva voce Examination held on

**InternalExaminer ExternalExaminer**

**DECLARATION**

I,**V.L.V.PRAYUSH**hereby declare that the project report entitled “**Rainfall prediction using machine learning”**done by me under the guidance of **Dr.S.Dhamodaran** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

## DATE:

**PLACE: SIGNATURE OF THECANDIDATE**

**ACKNOWLEDGEMENT**

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D**, **Dean**, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of **Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.S.Dhamodaran, M.E., Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

**TRAINING CERTIFICATE**

**ABSTRACT**

This project is carried on the heuristic prediction of rainfall using machine learning techniques. As we all know agriculture was the leading issue of our country and economy. Even though a regular rain pattern is usually played important role for healthy agriculture but too much rainfall or too little rainfall can be injurious, even it led to destructive for crops. The decreasing trends in seasonal rainfall and post-monsoon rainfall and increasing occurrence of the deficit rainfall years indicates the probable intensification of water scarcity. This project calculates the rate of rainfall in previous years according to various crops season like rabbi, Kharif, zaid and predicts the rainfall in future seasons. We have selected a real data set which consists of past year’s rainfall rate accorded to various seasons. This website results in helping farmers to make a correct decision to harvest a particular crop accordingly to crops seasons.

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 approximate or incomplete estimation problems because rainfall estimation is affected from the geographical and regional changes and properties. This paper presents review of different methods used for rainfall prediction and problems one might encounter while applying different approaches for rainfall forecasting.

(i)

|  |  |  |
| --- | --- | --- |
|  | **TABLE OF CONTENTS** |  |
| **CHAPTER No.** | **TITLE** | **PAGE No** |
|  | ABSTRACT | i |
|  | LIST OF FIGURES | iv |
| **1** | **INTRODUCTION** | **1** |
|  | 1.1 RAIN  1.2 TYPES OF RAIN | 1 |
|  | 1.3. TYPES OF RAINFALL BASED ON INTENSITY | 2 |
| **2.** | **AIM AND SCOPE OF THE PRESENT INVESTIGATION** | **3** |
|  | 2.1 AIM | 3 |
|  | 2.2 SCOPE | 3 |
|  | 2.3 DATASET INFORMATION | 3 |

**3. EXPERIMENTAL OR MATERIALS AND METHODS, ALGORITHMSUSED 5**

|  |  |  |
| --- | --- | --- |
|  | 3.1 TYPES OF REGRESSION ALGORITHMS USED  3.1.1 LINEAR REGRESSION ALGORITHM  3.2 LINEAR REGRESSION LINE | 5  5    7 |
|  | 3.3 ALGORITHM | 10 |
|  |  |  |
|  |  |  |
| **4.** | **RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS** | **12** |
|  | 4.1 RESULT | 12 |
|  | 4.2 DISCUSSION  4.3 PERFORMANCE ANALYSIS | 12  13 |
| **5** | **SUMMARY AND CONCLUSIONS**    5.1 SUMMARY    5.2 CONCLUSION | **13**  13  14 |
|  | **REFERENCES** | **14** |
|  | **APPENDIX** |  |
|  | A.SCREENSHOTS AND OUTPUTS | 15 |
|  | B.PROGRAM/SCOURCE CODE | 32 |

# (iii)

# LIST OF FIGURES

## FIGURENO. FIGURENAME PAGENO.

1.1 RAIN 01

3.1 LINEAR REGRESSION 06

5.1 HEAT MAP 17

5.2 ANNUAL RAINFALL 18

5.3 OVERALL RAINFALL(1) 19

5.4 OVERALL RAINFALL(2) 20

5.5 OVERALL RAINFALL(3) 21

5.6 HISTOGRAM 22

5.7 OUTPUT GRAPH(1) 26

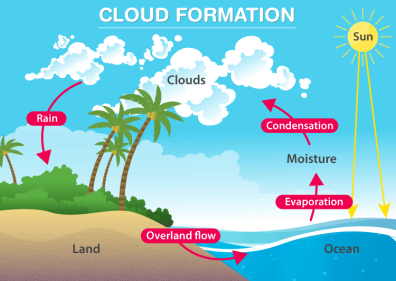
5.8 OUTPUT GRAPH(2) 27

5.9 OUTPUT GRAPH(3) 29

5.10 OUTPUT GRAPH(4) 30

(iv)

**CHAPTER-1**:INTRODUCTION

 (fig:1.1:RAIN)

1.1 RAIN:

Rain is liquid precipitation of water falling from the sky. Raindrops fall to Earth when clouds become saturated, or filled, with water droplets. Millions of water droplets bump into each other as they gather in a cloud. When a small water droplet bumps into a bigger one, it condenses, or combines, with the larger one. As this continues to happen, the droplet gets heavier and heavier. When the water droplet becomes too heavy to continue floating around in the cloud, it falls to the ground.

1.2 TYPES OF RAINFALL:

Rainfall has been classified into three main types based on the origin –

1. Convectional rainfall
2. Orographic or relief rainfall
3. Cyclonic or frontal rainfall

***1.2.1Convectional Rainfall – Major Characteristics***

* The air on getting heated becomes light and rises up in convection currents.
* As the air rises, it expands and drops the temperature and subsequently, condensation takes place and cumulus clouds are formed.
* Heavy rainfall with lightning and thunder takes place which does not last long.
* Such rain is usual in the summer or in the hotter part of the day.
* This type of rainfall generally takes place in the equatorial regions and internal parts of the continents, predominantly in the northern hemisphere.
* This rainfall is usually associated with hail and graupel.

1

***1.2.2 Orographic Rainfall – Major Characteristics***

* When the saturated air mass comes across a mountain, it is forced to rise.

• The rising air expands, eventually, the temperature falls, and the moisture gets condensed.

• The principal characteristic of this type of rain is that the windward slopes get more rainfall.

• After giving rain on the windward side, when these winds reach the other slope, they drop away, and their temperature increases. Then their ability to take in moisture increases and hence, these leeward slopes remain dry and rainless.

* The region situated on the leeward side is known as the rain-shadow area.

***1.2.3 Cyclonic Rainfall – Major Characteristics***

* Cyclonic activity causes cyclonic rain and it occurs along the fronts of the cyclone.
* When two masses of air of unlike density, temperature, and humidity meet then it is formed.
* The layer that separates them is known as front.
* Warm front and the cold front are the two parts of the front.
* At the warm front, the warm lighter wind increases slightly over the heavier cold air.
* As the warm air rises, it cools, and the moisture present in it condenses to form clouds
* This rain falls gradually for a few hours to a few days.

1.3TYPES OF RAINFALL BASED ON INTENSITY

The types of rainfall based on intensity can be classified as:

1. Light rain – Rate of rain varies between 0 to 2.5 millimetres

2. Moderate rain – Rate of rain varies between 2.6 millimetres to 7.6 millimetres

3. Heavy rain – Rate of rain is beyond 7.6 millimetre

2

**CHAPTER-2**: AIM AND SCOPE OF PRESENT INVESTIGATION

2.1AIM:

To predict the Rainfall in the given Rainfall Data set from the past years using machine learning algorithm.

2.2 SCOPE:

The application of science and technology that predicts the state of the atmosphere at any given particular time period is known as Weather forecasting. There are many different methods to a weather forecast. Weather forecast notices are important because they can be used to prevent the destruction of life and environment. So basically this project helps farmers to decide the proper crops for their fields. For that problem, we use a machine learning technique to first determine the rainfall and after that farmers can get the guidance for their crops.

2.3 DATASET INFORMATION:

The dataset used in this system contains the rainfall of several regions in and across the country. It contains rainfall from 1901 – 2015 for the same. Along with that annual rainfall is also been used and the rainfall between the transition of two months. There are in total 4116 rows present in the dataset. The dataset is been collected from data.gov.in.

Category – Rainfall in India

Released under – NDSAP

Contributor – Ministry of Earth Sciences, IMD

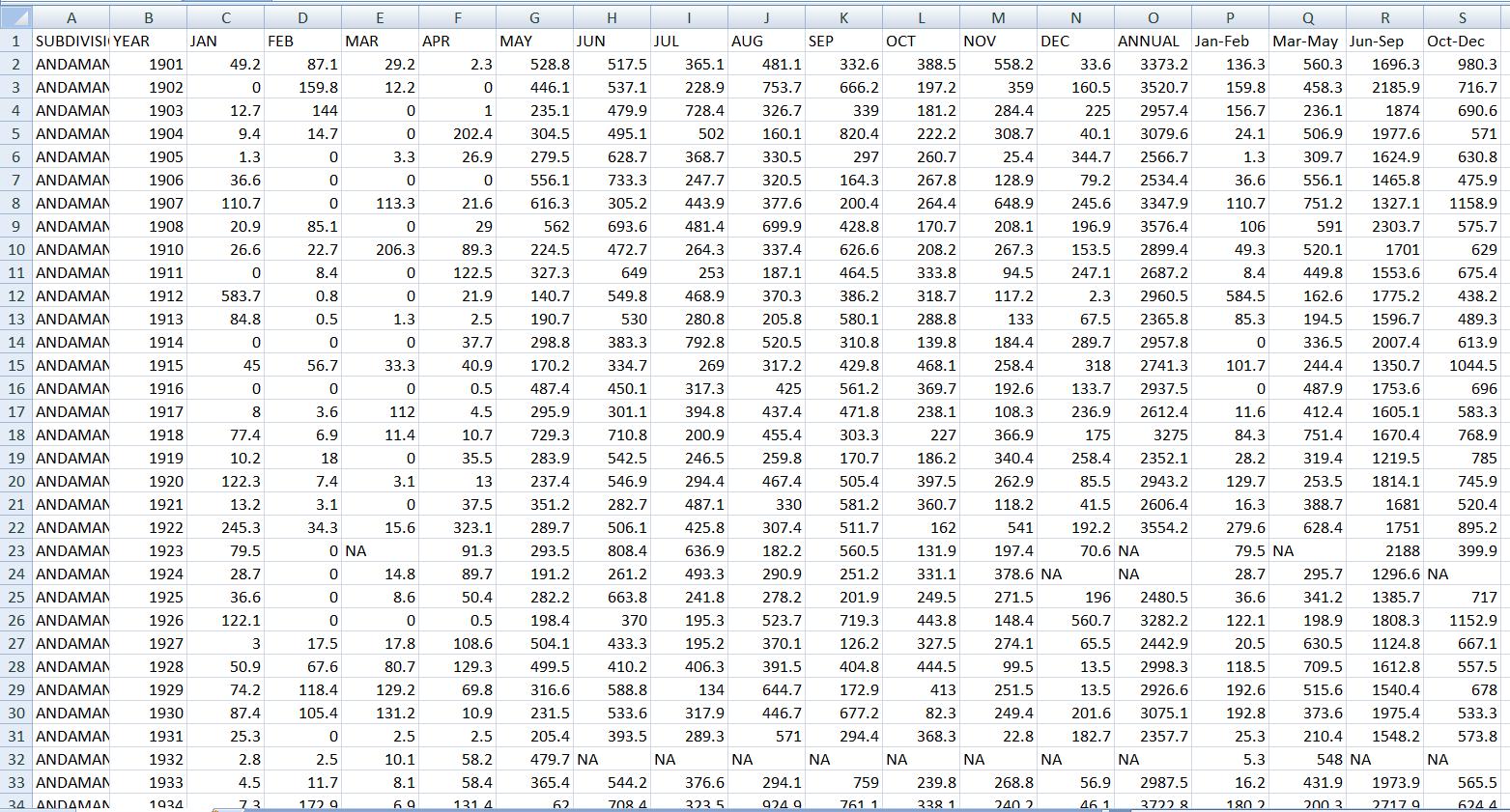
Group – Rainfall

3

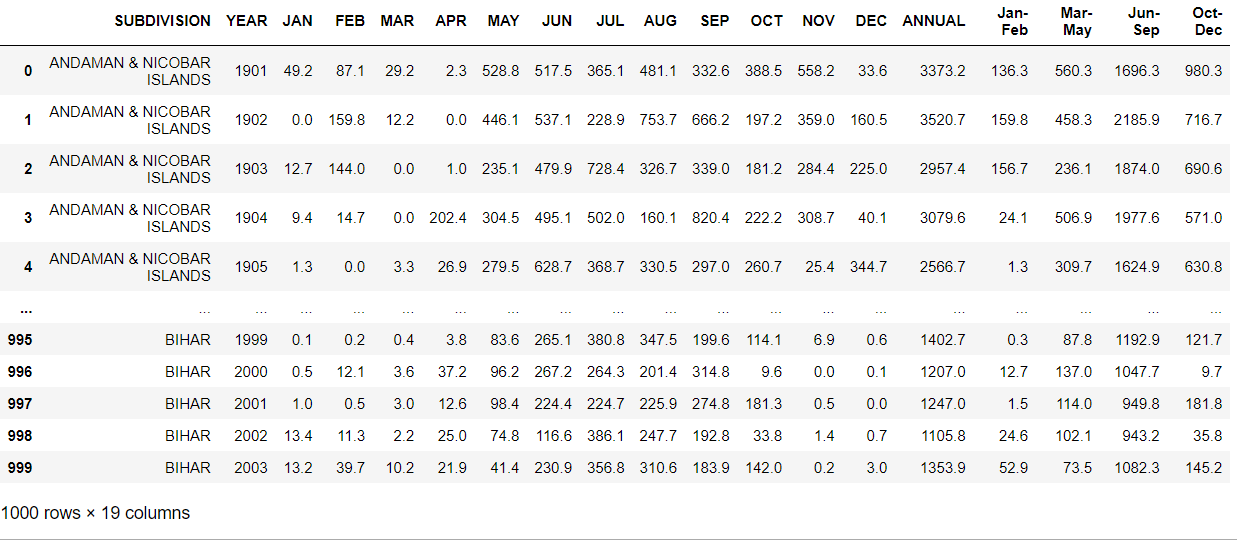
Sectors – Atmosphere science, earth sciences, science & technology

Source: OGD

Below is the snapshot for the same.



(Fig:2.1-Dataset of Rainfall in India in csv)



(Fig:2.1-Loaded Dataset of rainfall in jupiter)

4

**CHAPTER-3**:EXPERIMENTAL OR MATERIAL AND METHODS;ALGORITHMS USED

The given problem is based on the regression problem . so, we are using the regression algorithms to find the objective of our project.

3.1 Types of regression algorithms in Machine Learning:

1. Linear Regression.
2. Logistic Regression.
3. Polynomial Regression.
4. Support Vector Regression.
5. Decision Tree Regression.
6. Random Forest Regression.
7. Ridge Regression.
8. Lasso Regression.

* We used mainly Linear regression algorithm

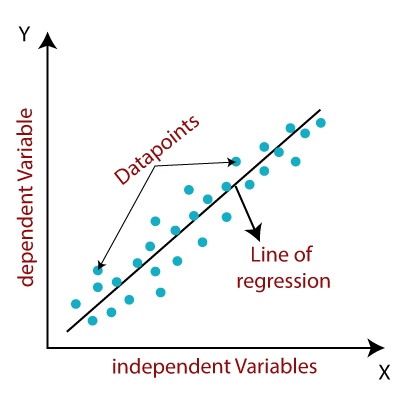
*3.1.1* ***Linear Regression Algorithm***:

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as **sales, salary, age, product price,** etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

The linear regression model provides a sloped straight line representing the relationship between the variables. Consider the below image:

5



(fig:3.1-Linear Regression)

Mathematically, we can represent a linear regression as:

y= a0+a1x+ ε

Here**,**

Y= Dependent Variable (Target Variable)  
X= Independent Variable (predictor Variable)  
a0= intercept of the line (Gives an additional degree of freedom)  
a1 = Linear regression coefficient (scale factor to each input value).  
ε = random error

The values for x and y variables are training datasets for Linear Regression model representation.

6

Types of Linear Regression

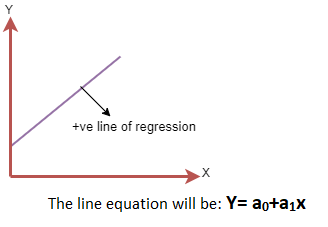
Linear regression can be further divided into two types of the algorithm:

* **Simple Linear Regression:**  
  If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.
* **Multiple Linear regression:**  
  If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

## 3.2 LINEAR REGRESSION LINE:

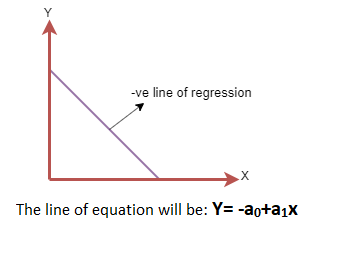
A linear line showing the relationship between the dependent and independent variables is called a **regression line**. A regression line can show two types of relationship:

* **PositiveLinearRelationship:**  
  If the dependent variable increases on the Y-axis and independent variable increases on X-axis, then such a relationship is termed as a Positive linear relationship.



7

* **NegativeLinearRelationship:**  
  If the dependent variable decreases on the Y-axis and independent variable increases on the X-axis, then such a relationship is called a negative linear relationship.



For Linear Regression, we use the **Mean Squared Error (MSE)** cost function, which is the average of squared error occurred between the predicted values and actual values. It can be written as: For Linear Regression, we use the **Mean Squared Error (MSE)** cost function, which is the average of squared error occurred between the predicted values and actual values. It can be written as:

For the above linear equation, MSE can be calculated as:

Linear Regression in Machine Learning

**Where,**

N=Total number of observation  
Yi = Actual value  
(a1xi+a0)= Predicted value.

**Residuals:** The distance between the actual value and predicted values is called residual. If the observed points are far from the regression line, then the residual will

8

be high, and so cost function will high. If the scatter points are close to the regression line, then the residual will be small and hence the cost function.

Gradient Descent:

* Gradient descent is used to minimize the MSE by calculating the gradient of the cost function.
* A regression model uses gradient descent to update the coefficients of the line by reducing the cost function.
* It is done by a random selection of values of coefficient and then iteratively update the values to reach the minimum cost function.

Model Performance:

The Goodness of fit determines how the line of regression fits the set of observations. The process of finding the best model out of various models is called **optimization**. It can be achieved by below method:

**1. R-squared method:**

* R-squared is a statistical method that determines the goodness of fit.
* It measures the strength of the relationship between the dependent and independent variables on a scale of 0-100%.
* The high value of R-square determines the less difference between the predicted values and actual values and hence represents a good model.
* It is also called a **coefficient of determination,** or **coefficient of multiple determination** for multiple regression.
* It can be calculated from the below formula:

Linear Regression in Machine Learning

9

3.3 ALGORITHM:

***3.3.1 STEPS****:*

1. Start
2. Import required libraries like numpy,pandas,matplotlib.pyplot,seaborn as np,pd,plt,sns
3. Read the given dataset and print the data
4. Drop unnecessary columns using drop function.
5. Checking null values in each column.
6. Drop null values using drop functionNow
7. Now print the array form of subdivisions
8. Now describe the data to understand more clearly.
9. Draw the heat map for the given data
10. Calculate the subdivision average annual rainfall and plot the data.
11. Now calculate and plot the overall rainfall in each year in the given data set
12. Calculate the Monthly Rainfalls Plot (Subdivision wise)
13. Calculate theMonthly Rainfalls Plot (Yearwise)
14. Drawing the histographic graphs for the given dataset
15. from sklearn.model\_selection import train\_test\_split,starting the split method
16. Training/Testing is split in the ratio 80:20 randomly.
17. Print the X,y values
18. Print the X\_train,X\_test values.
19. Print the y\_train,y\_test values.
20. Now,Prediction using Linear Regression
21. Single Linear Model on Whole Data Ignoring Subdivision.
22. To begin with, we shall predict the rainfall for the current month with predictor variables as the rainfall in previous three months.
23. Data is arranged into 36810 rows and 4 columns with first three columns as the predictor variables and the last column is dependent variable.
24. For each month from April to December, four columns are appended at the bottom of the data to create a data of 36810 rows and 4 columns.
25. Print the mean absolute difference of testing and traininig data and plot the graphs.
26. Similarly ,now single linear model on whole data including subdivisions.
27. In this section, we shall predict the rainfall for the current month with predictor variables as the rainfall in previous three months and the subdivision.
28. As subdivision is a categorical variable consisting of 36 values, 36 dummy binary variables are added inplace of categorical variable to use in the regression.
29. Print the mean absolute difference of testing and traininig data and plot the graphs.
30. Now,Linear Model Fitted to Each Subdivision Category Independently.
31. Similarly ,same as the above steps calculate the mean absolute difference score for the training and testing data and print the graph.
32. For example we want any year accuracy of predicted values.
33. Finding the linear model and fitting the model to both year values.
34. Now predicting the values of that year.
35. Calculating the mean absolute error for the test and predicted value and print the score.
36. STOP.

11

**CHAPTER-4:** RESULTS AND DISCUSSION,PERFORMANCE ANALYSIS

4.1 RESULTS:

Machine learning models achieved significantly higher prediction accuracy

over common clinical risk scores , By using the linear regression model we achieved mean absolute difference accuracy>0.85 in all the types we used in the problem. But we got mean absolute error 95% by using the data.

The code of our projectexecuted successfully!

4.2 DISCUSSION:

* In the given project Rainfall data set we have to predict Rainfall in the Given places by using Regression method.
* In Regression method we used Linear regression algorithm to predict the values and accuracy,and shown output in the form of graph.
* We have removed all unnecessary data from the data set to give more efficient accuracy scores and filled the null value columns with their mode values.
* We also created a testing model(basically a testing model is trained with dummy data/values) and compared with the original model to give better results.
* Finally we shown the result in the form of various graphs.

12

4.3 PERFORMANCE ANALYSIS:

Analysing the result of our project,

* Our project is a regression based problem so,we used regression algorithm.
* Mainly we plotted the starting heat map of all subdivisions.
* When we compare the result of our algorithm to the real world it will become more difficult to show the real output.
* Now a days rainfall is inappropriate due to the more global warming, unnecessary activites, pollution due to that we cannot predict the rain fall correctly.
* But we can predict the cyclonic rainfall due to changes in wheather.

**CHAPTER-5:**SUMMARY AND CONCLUSION

5.1 SUMMARY:

In today’s situation, rainfall is considered to be one of the sole responsible factors for most of the significant things across the world. In India, agriculture is considered to be one of the important factors for deciding the economy of the country and agriculture is solely dependent on rainfall. Apart From that in the coastal areas across the world, getting to know the amount of rainfall is very much necessary. In some of the areas which have water scarcity, to establish rain water harvester, prior prediction of the rainfall should be done. This project deals with the prediction of rainfall using machine learning . The project performs the comparative study of machine learning approaches then accordingly portrays the efficient approach for rainfall prediction. First of all, preprocess is performed. Preprocess is the process of representing the dataset in the form of several graphs such as bar graph, histogram etc.

13

5.2 CONCLUSION:

* Rainfall being one of the sole responsibilities for maximum economy of India, it should be considered the primary concern for most of us.
* The current approach for rainfall prediction fails in most of the complex cases.
* By using linear regression algorithm we got approximately 87% mean absolute difference
* Simillarly if we taken any particular subdivision in the data and calculated the model accuracy we will get difference in accuracy from place to place.
* The accuracy of the algorithm can be additionally tested on increase in the complexity.
* Henceforth, algorithm for testing daily basis dataset instead of accumulated dataset could be of paramount Importance for further research.
* More the accuracy of the system used for rainfall prediction, smarter will be the agriculture. Along with that, this will be an efficient tool for people in coastal areas of the country thereby making them well aware of the situation in advance.

REFERENCE:

*A flood prediction system developed using various machine ...* (n.d.). Retrieved April 9, 2022, from <https://www.researchgate.net/publication/352448189_A_Flood_Prediction_System_Developed_Using_Various_Machine_Learning_Algorithms>

Pinar Ozturk. (n.d.). Flood Prediction Using Machine Learning Models: Literature Review. Retrieved from <https://www.mdpi.com/2073-4441/10/11/1536> .

Python tutorial. (n.d.). Retrieved April 9, 2022, from <https://www.tutorialspoint.com/python/index.htm>

14

Liyew, C. M., & Melese, H. A. (2021, December 7). *Machine learning techniques to predict daily rainfall amount - Journal of Big Data*. SpringerOpen. Retrieved April 9, 2022, from <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00545-4>

Zeyi Chao, Fangling Pu, Yuke YinLing, B. and X. (2018). Research on real-time local rainfall prediction based on MEMS sensors. Journal of Sensors, 2018. <https://doi.org/10.1155/2018/6184713>

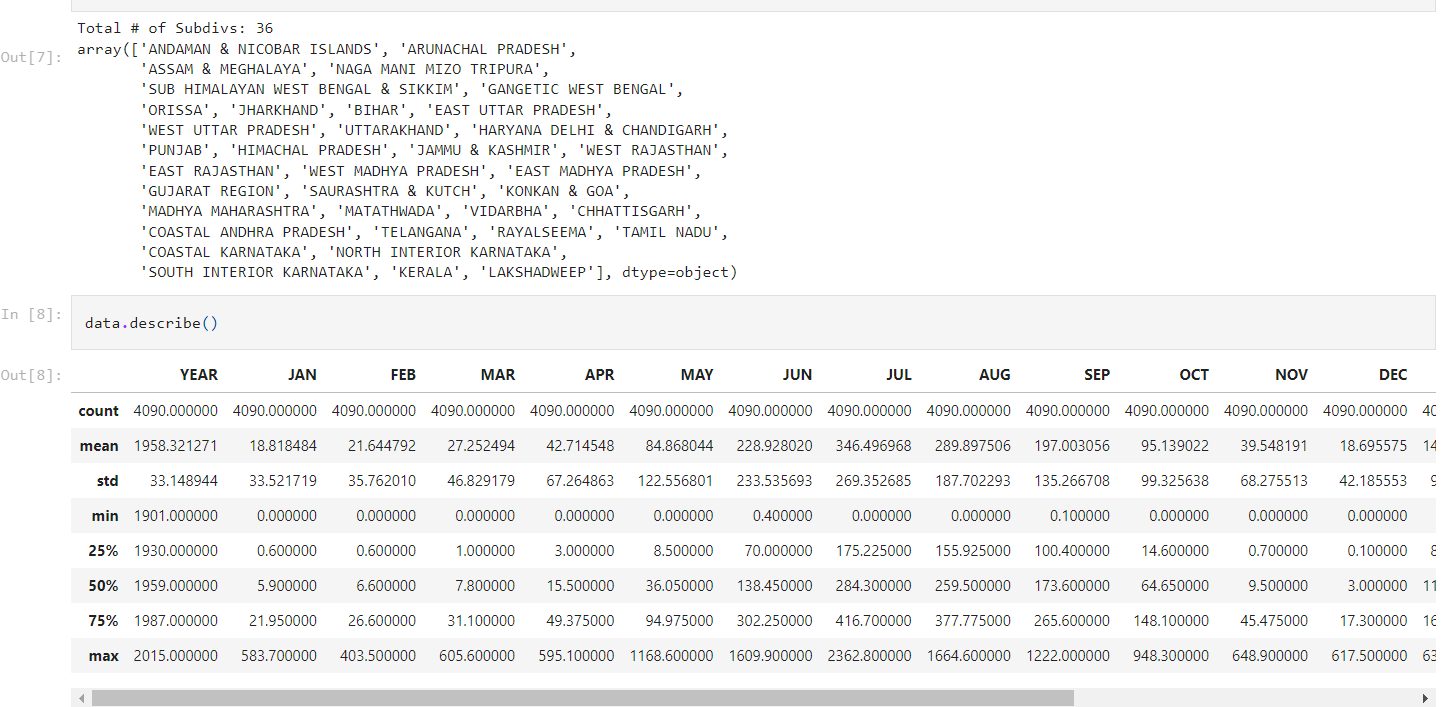
APPENDIX:

A.Screenshots and outputs:



15





16



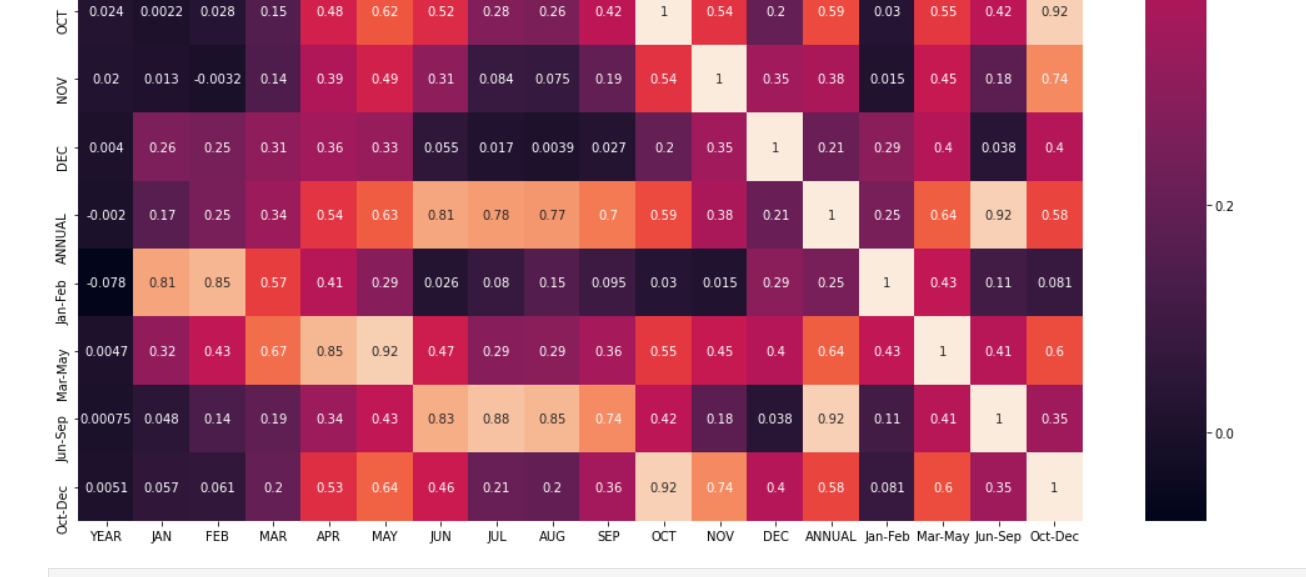
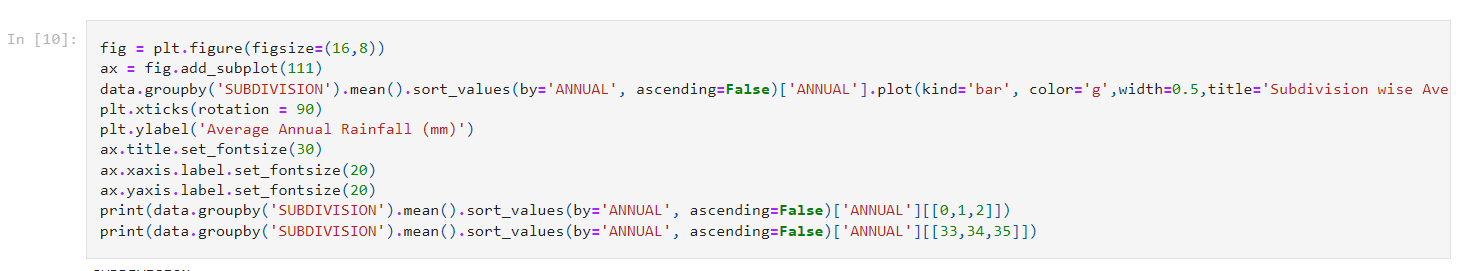


Fig:5.1

17



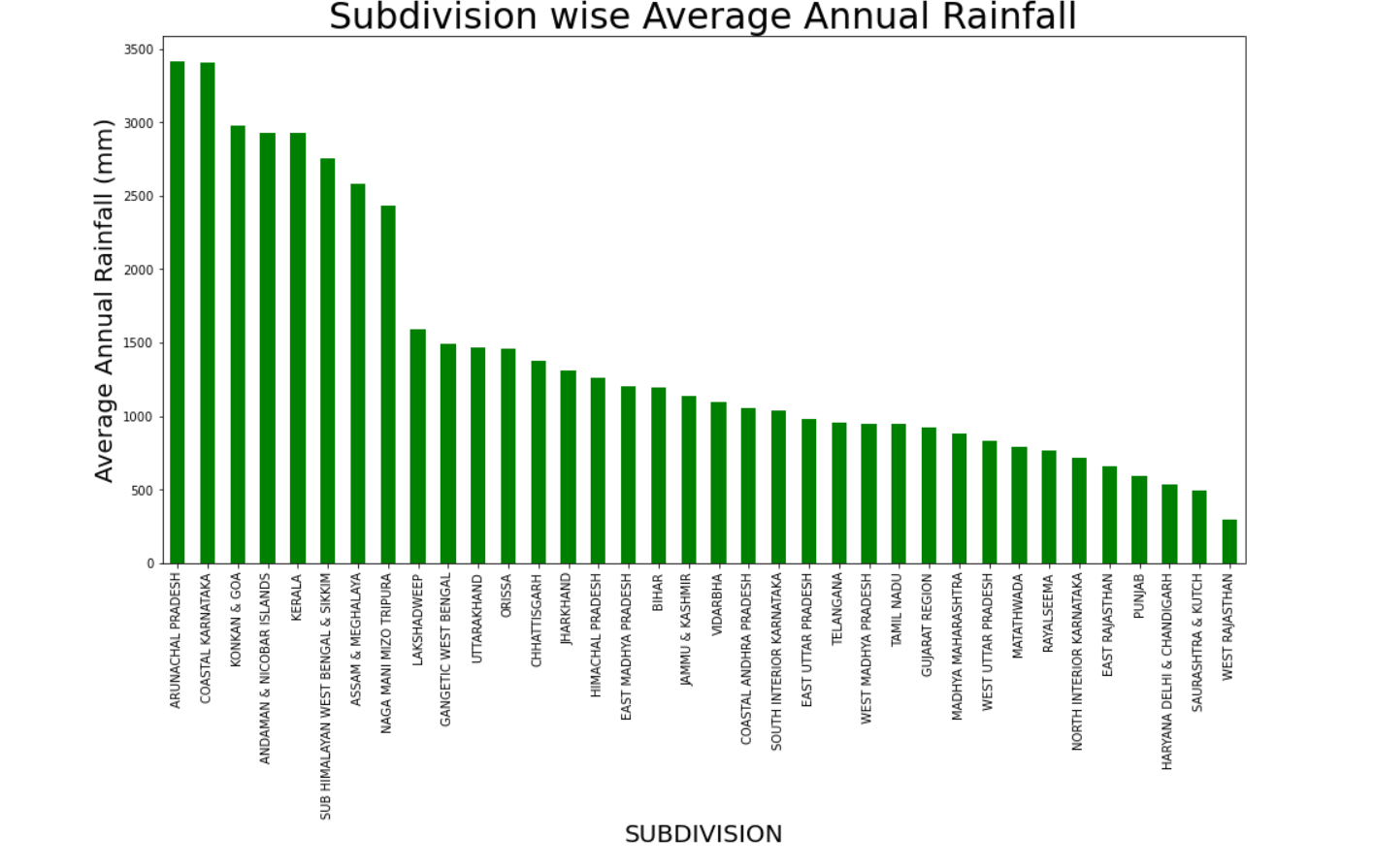


Fig:5.2



18

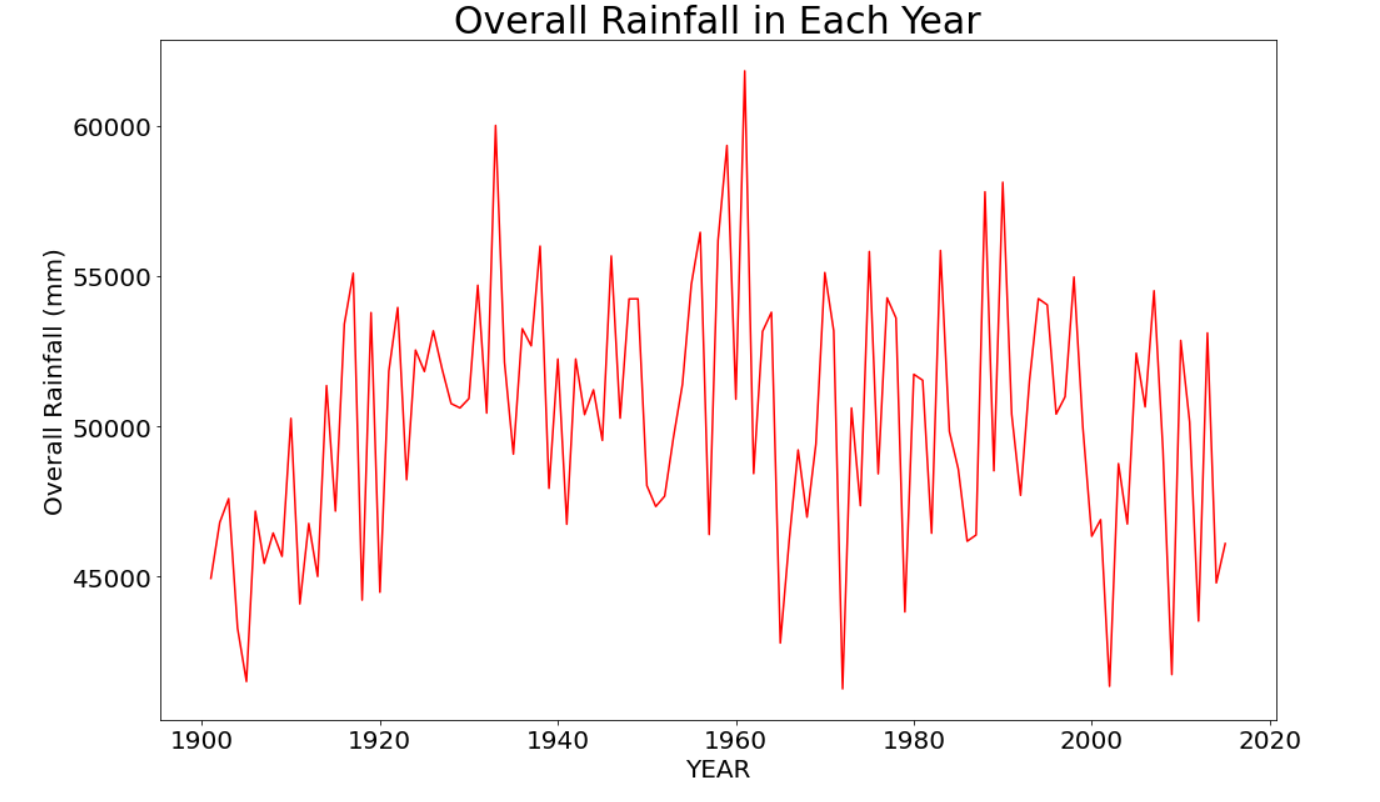
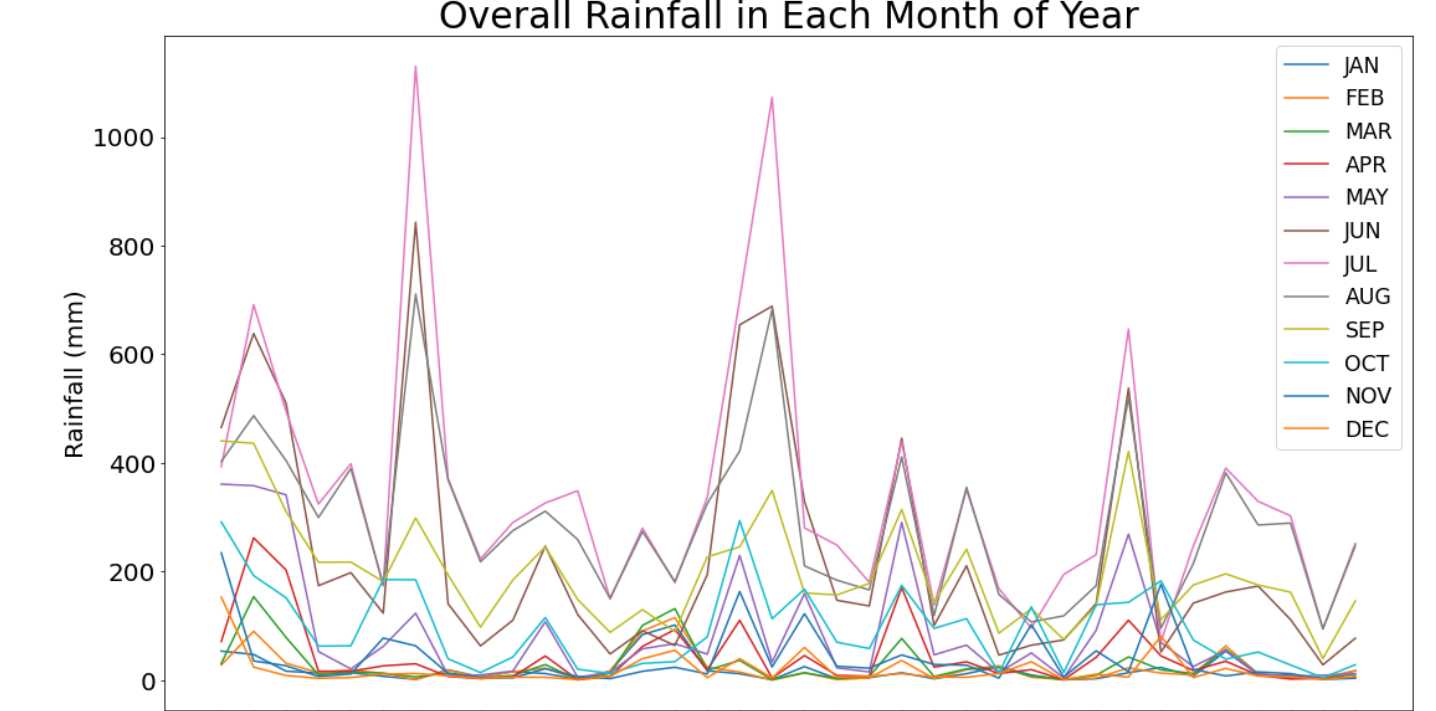


Fig:5.3



19



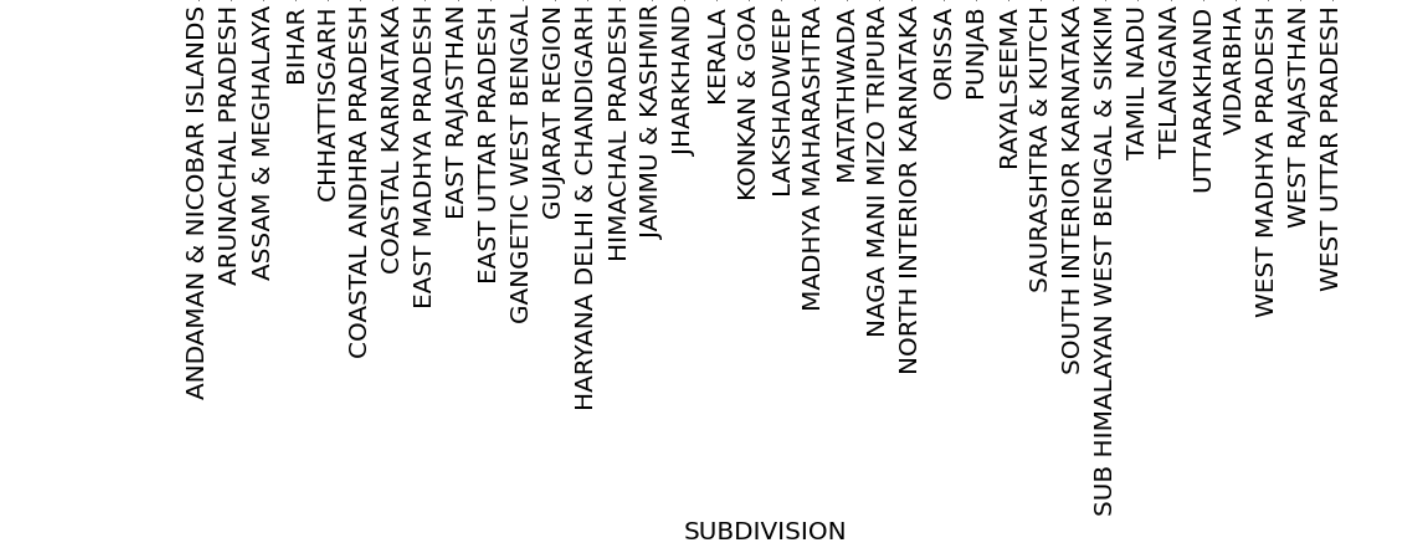
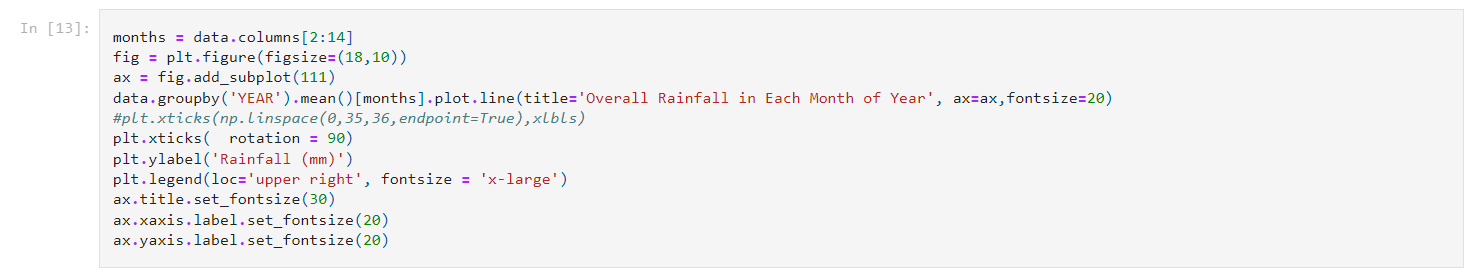


Fig:5.4



20

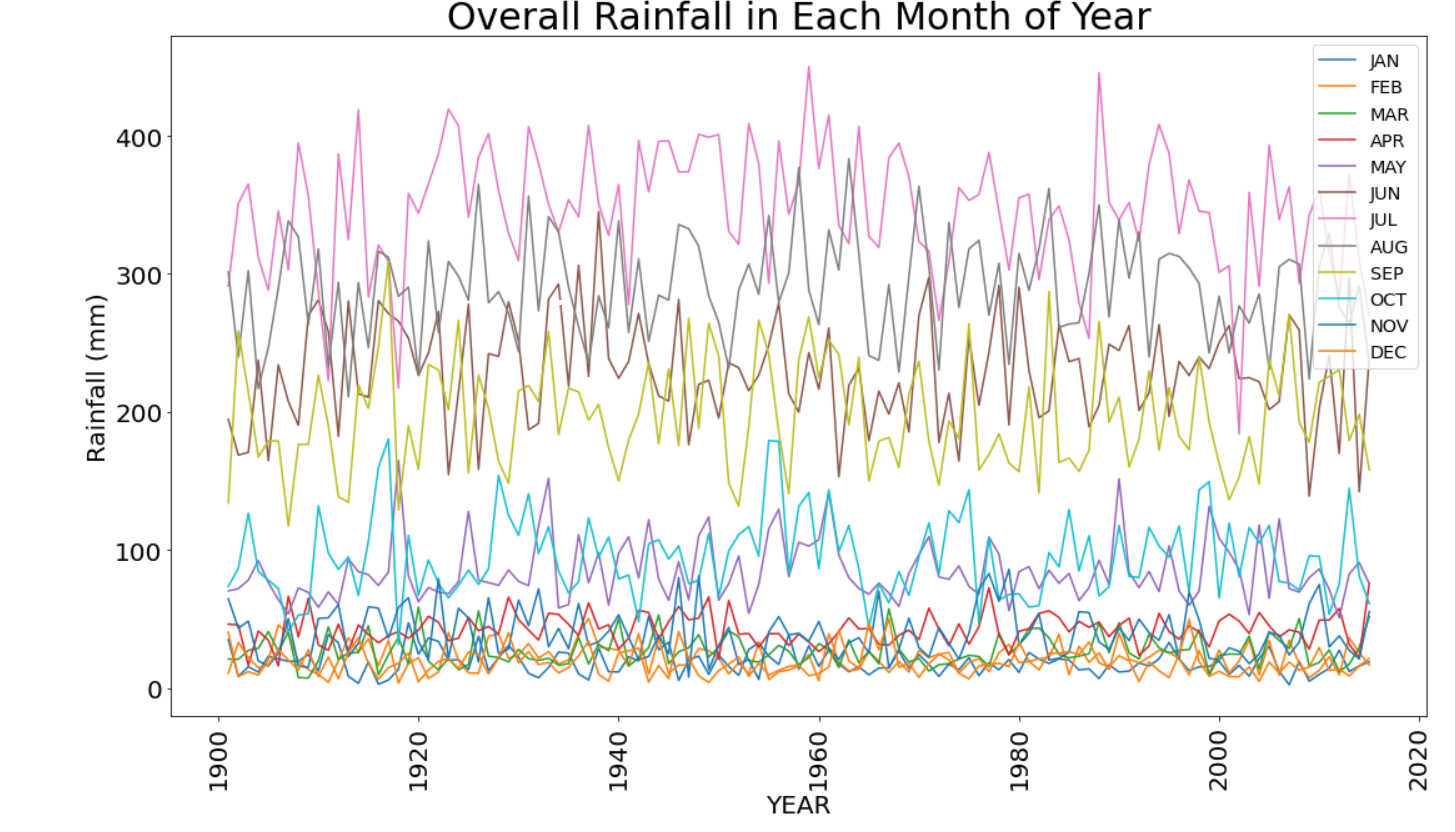
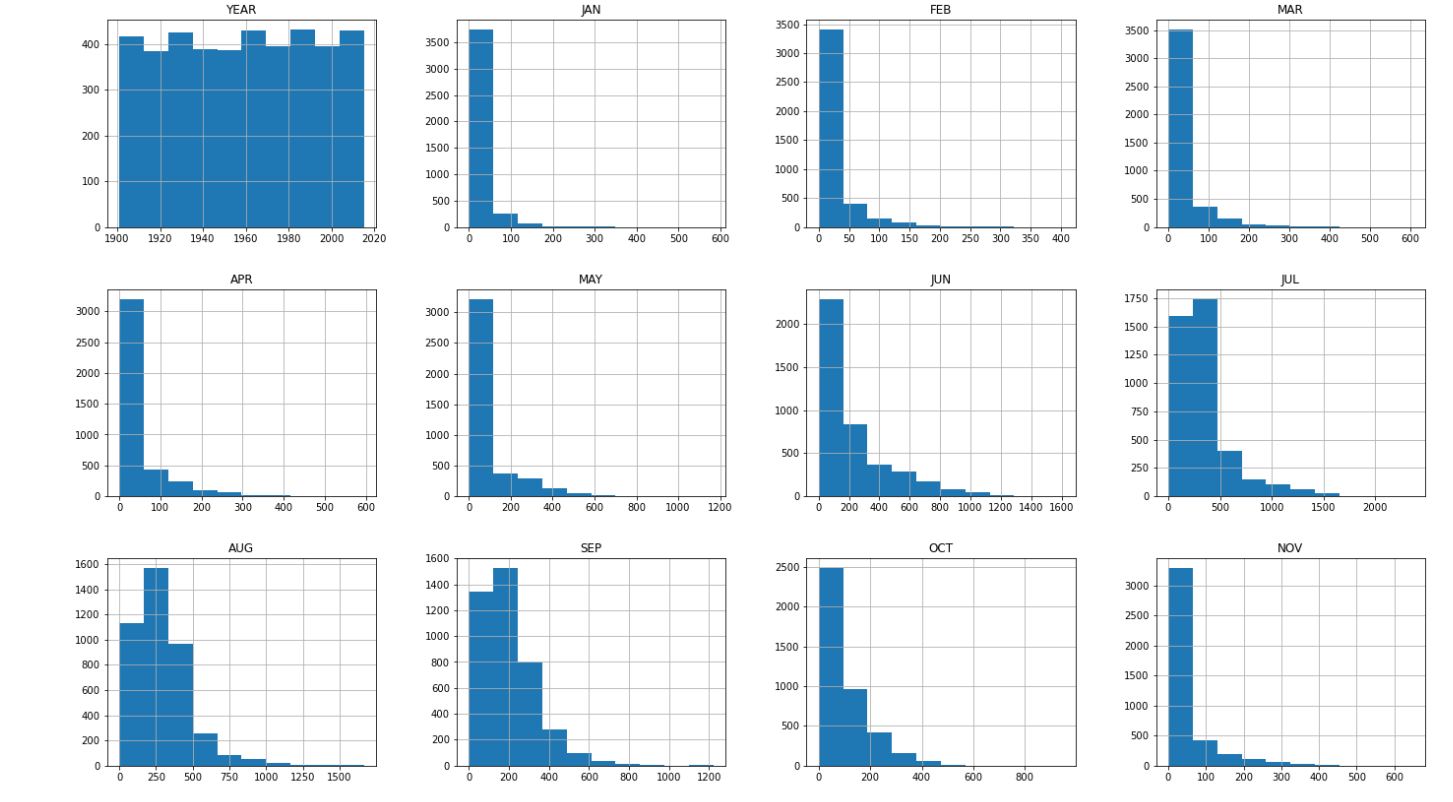


Fig:5.6



21



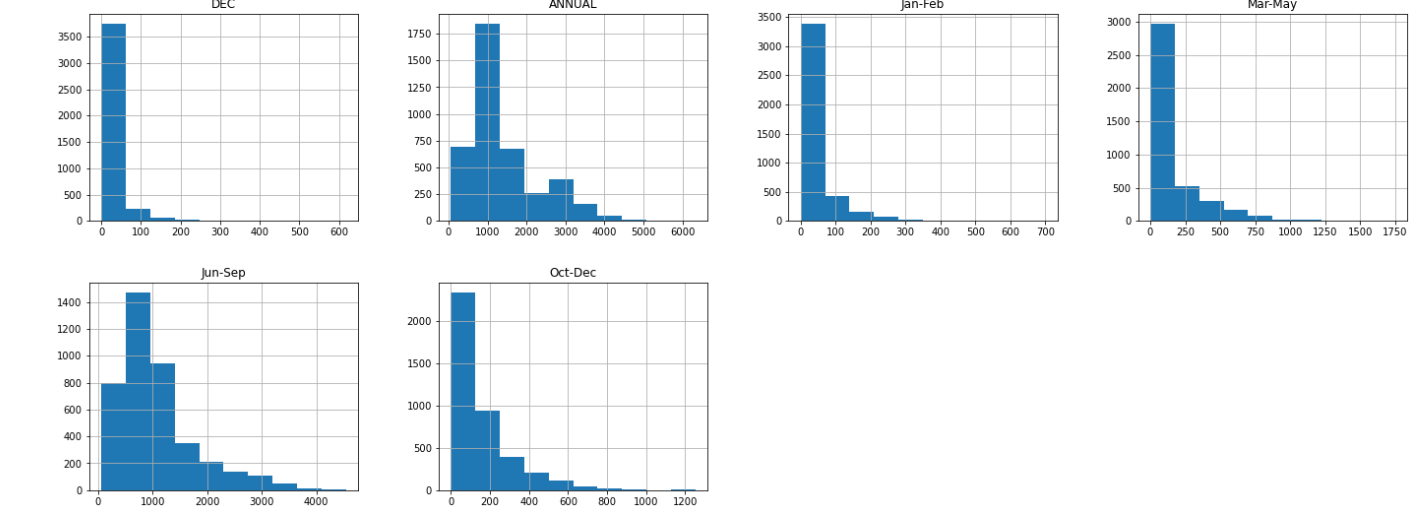


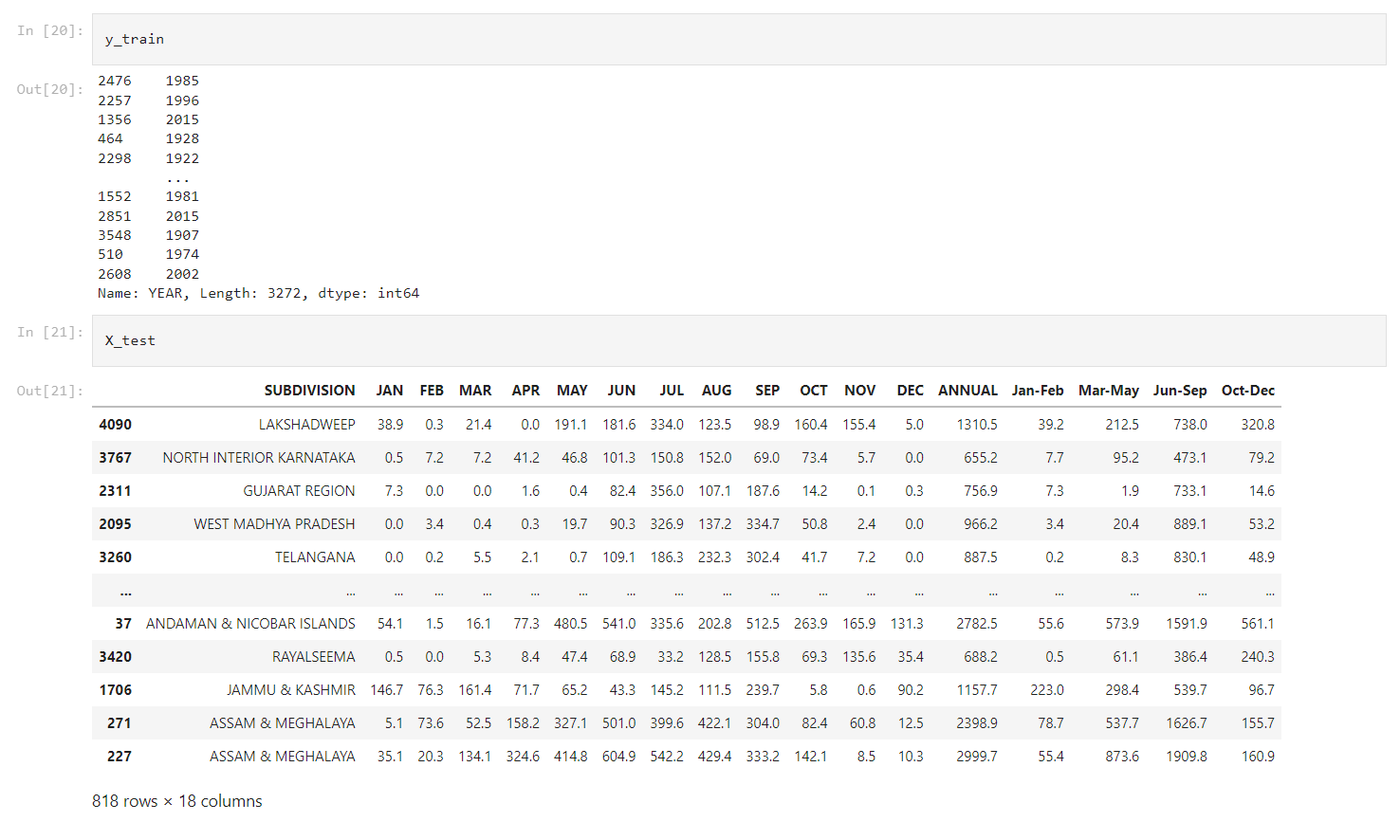
Fig:5.7

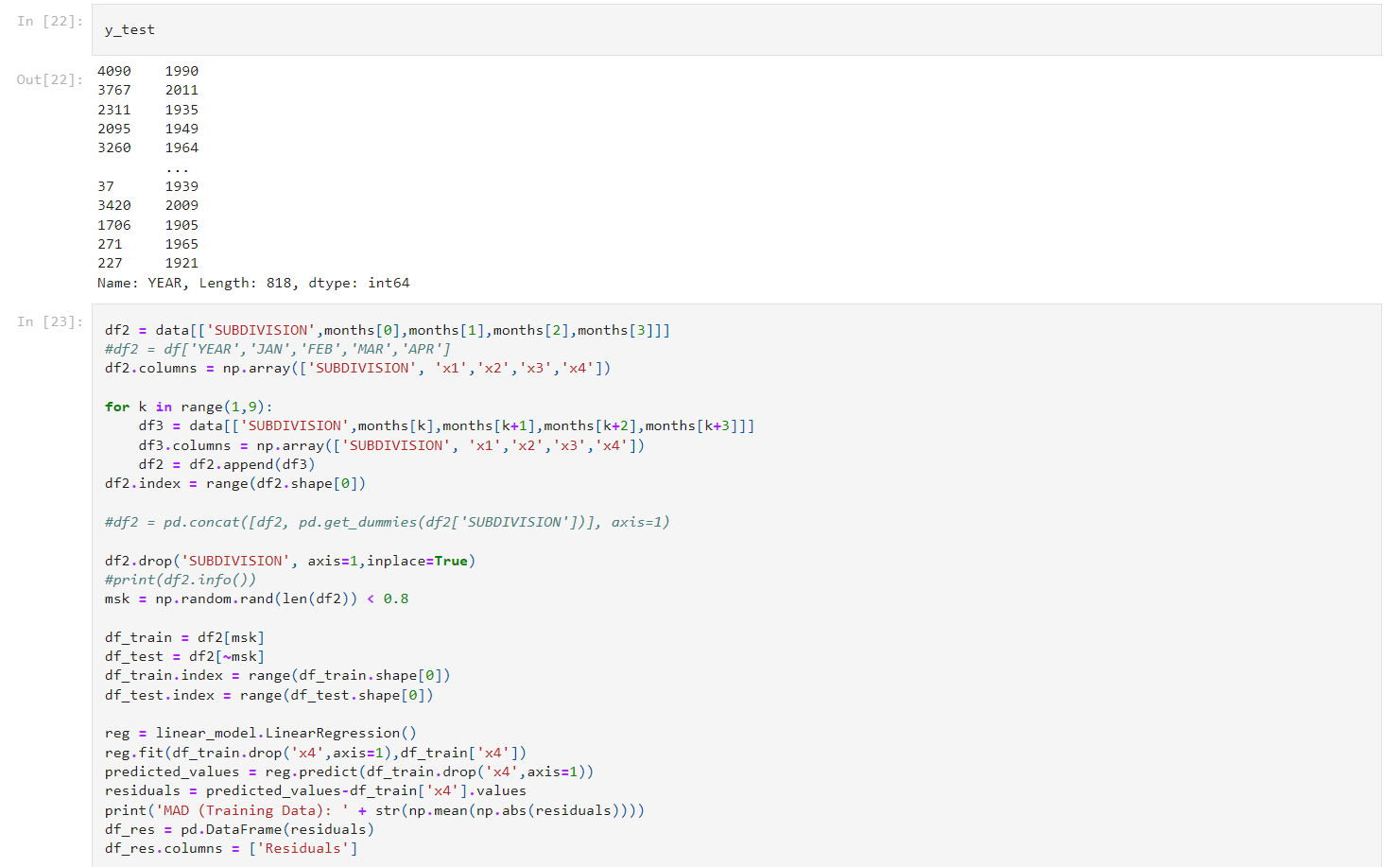
22



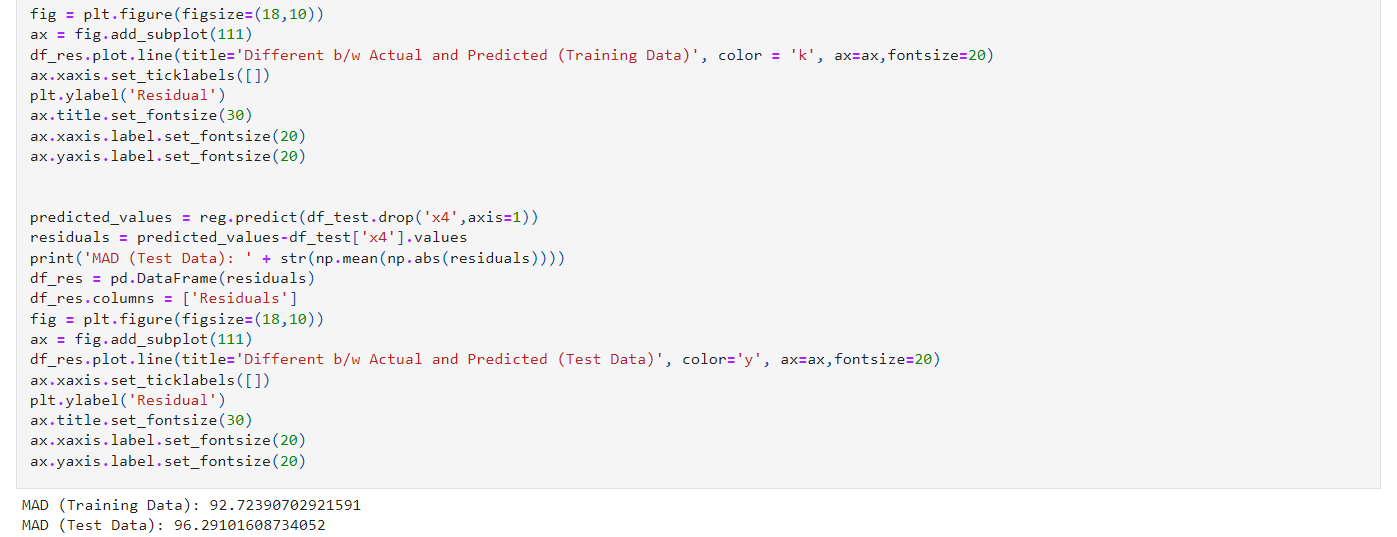


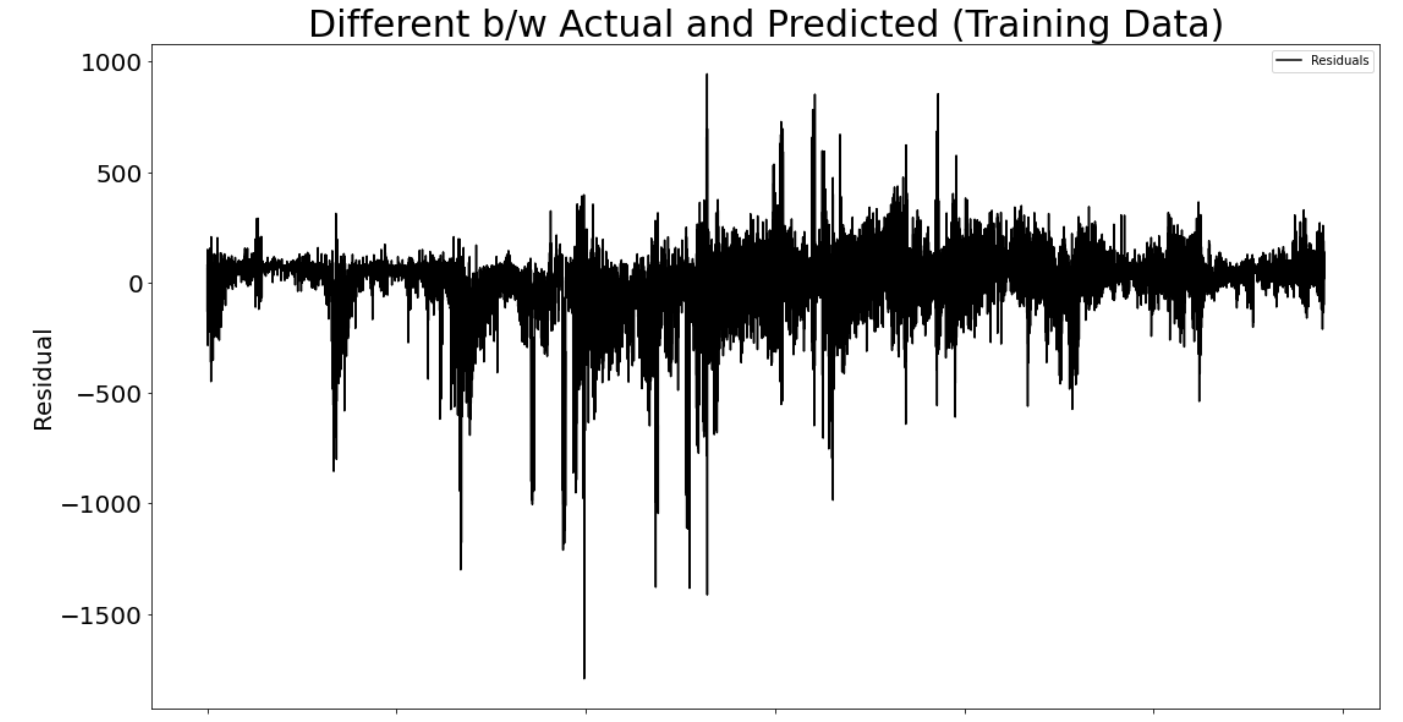
23





24





25

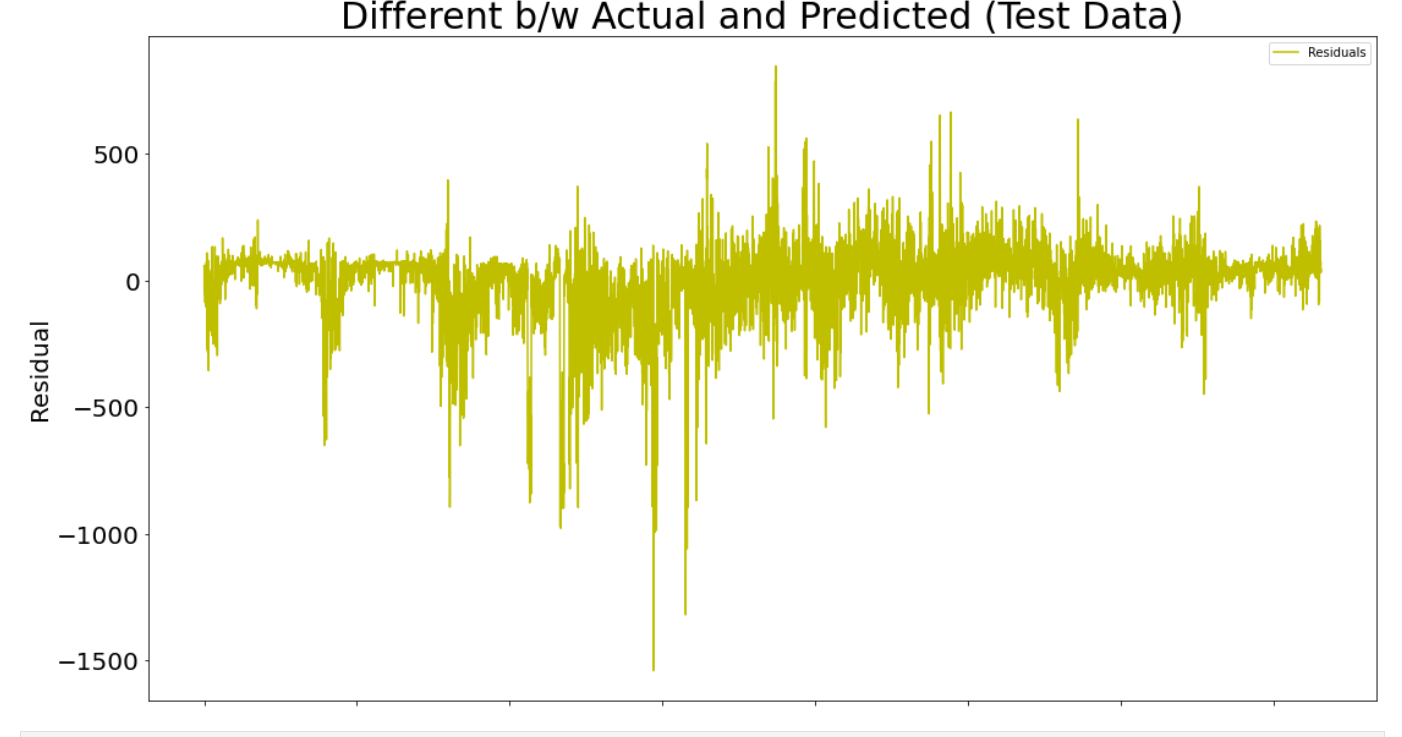
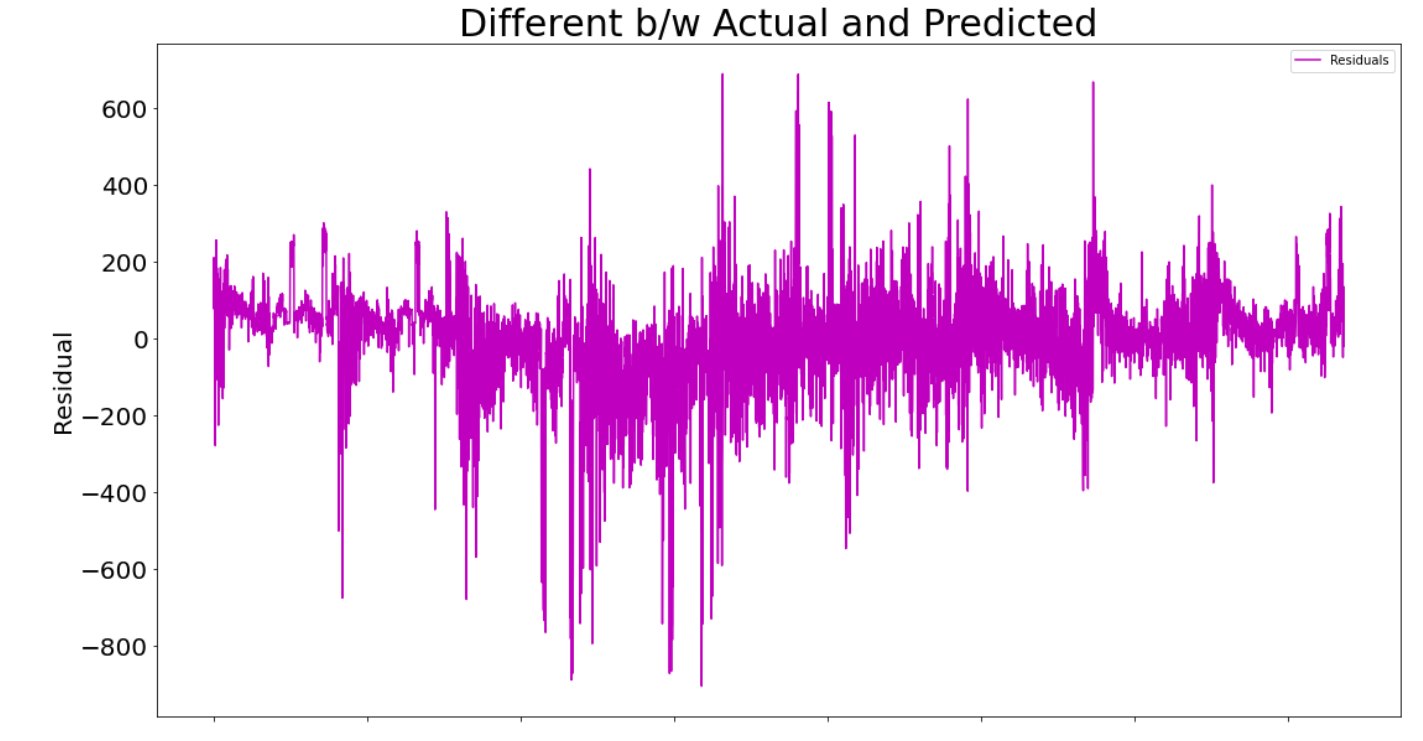
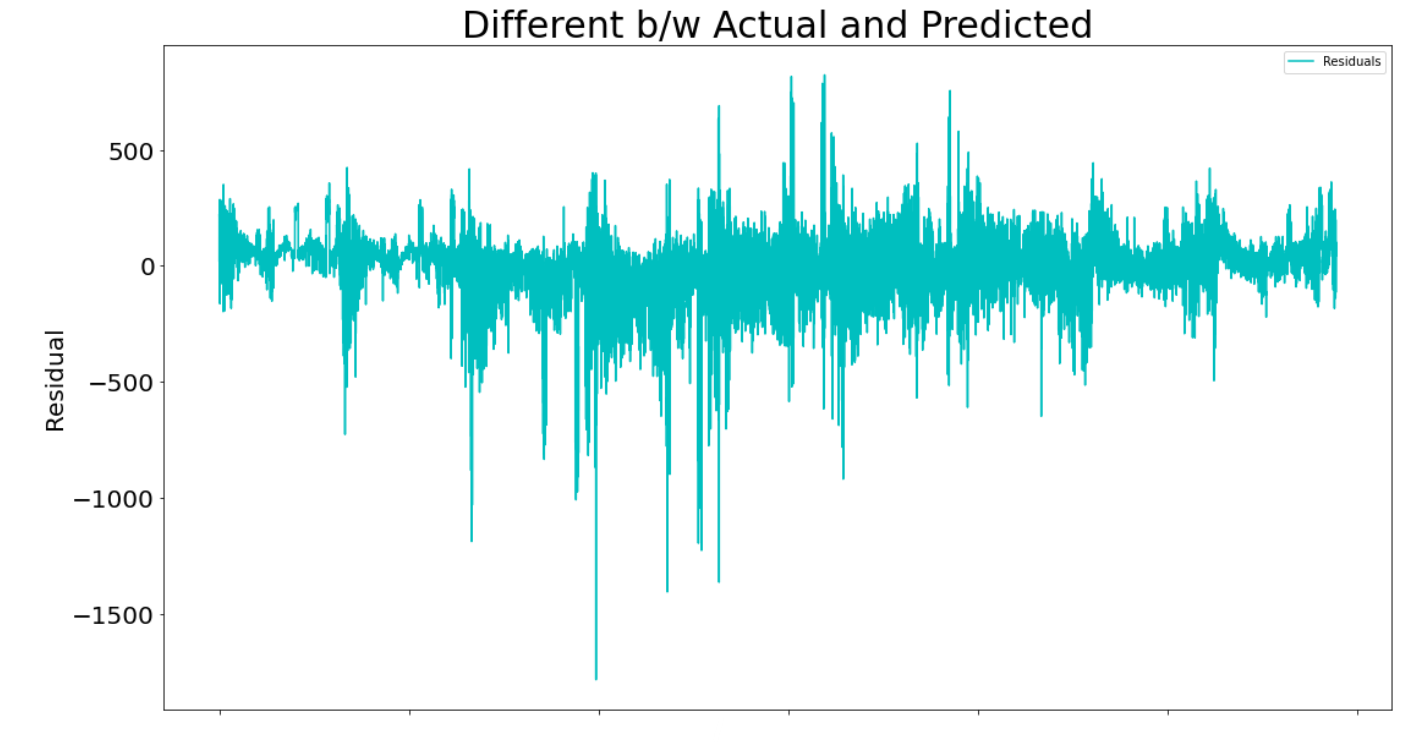


Fig:5.8



26

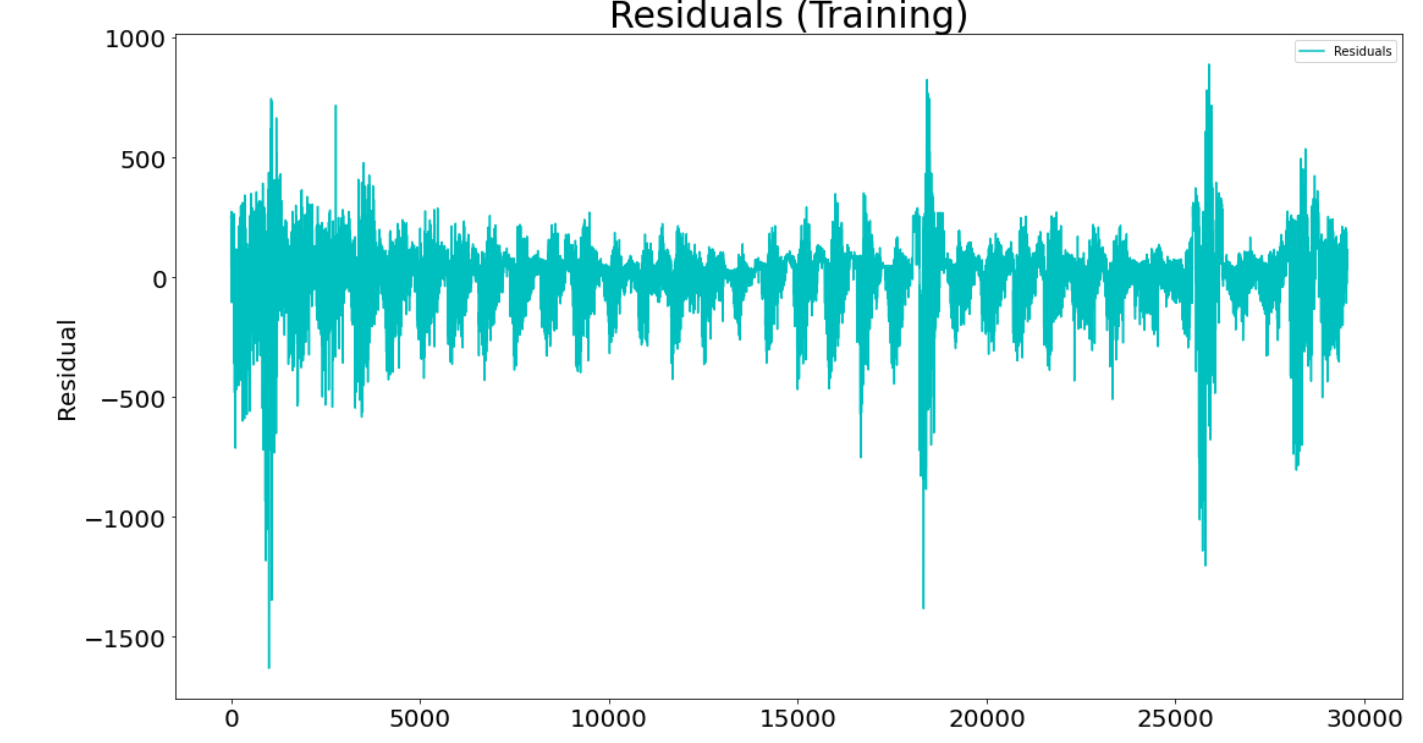
 fig:5.9

27





28



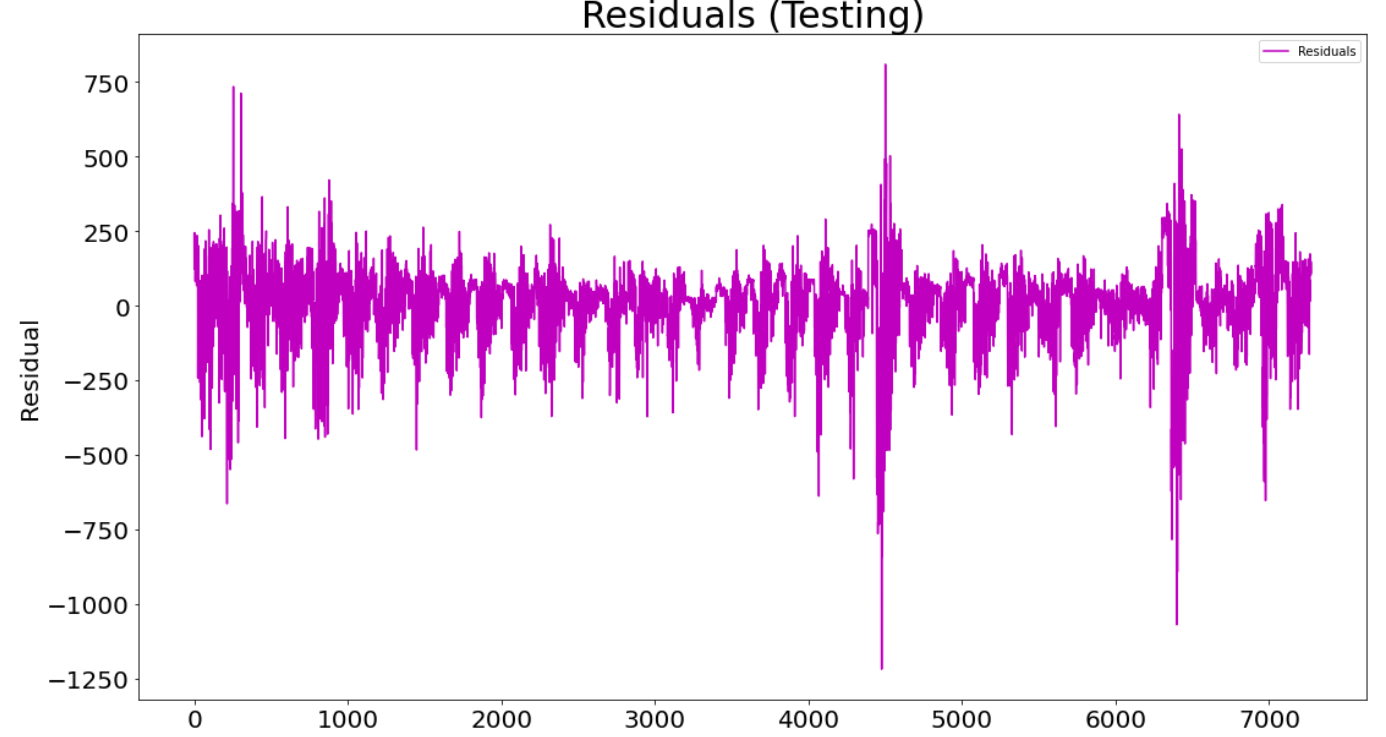


Fig :5.10

29

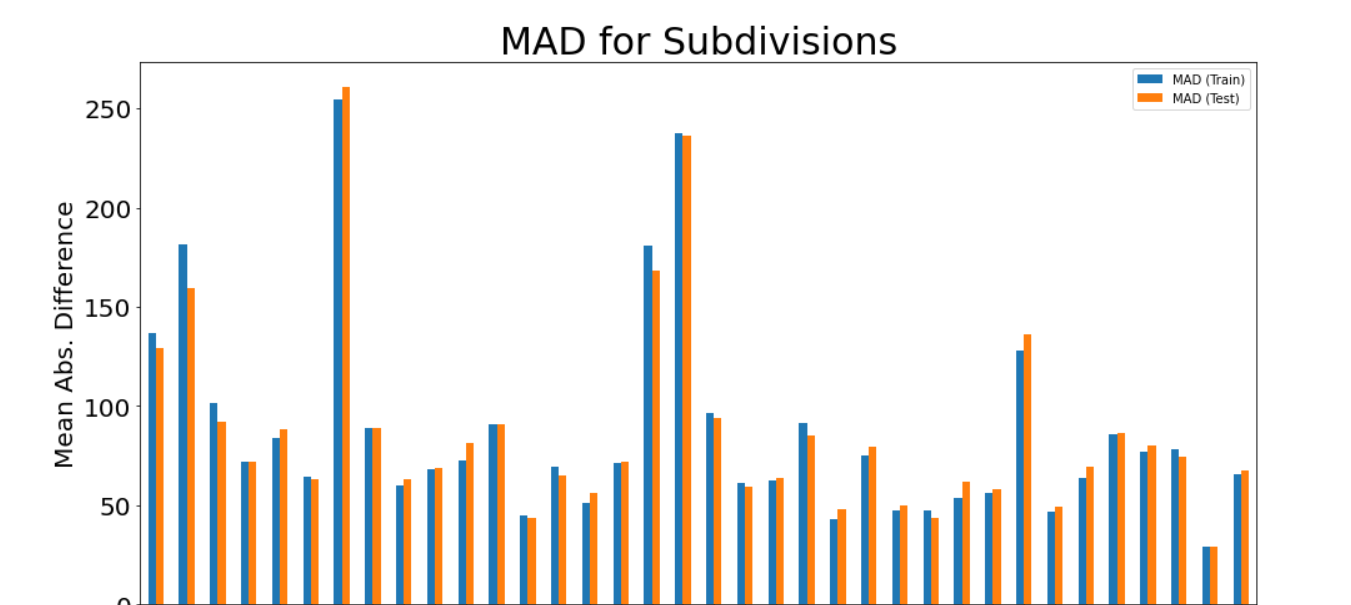
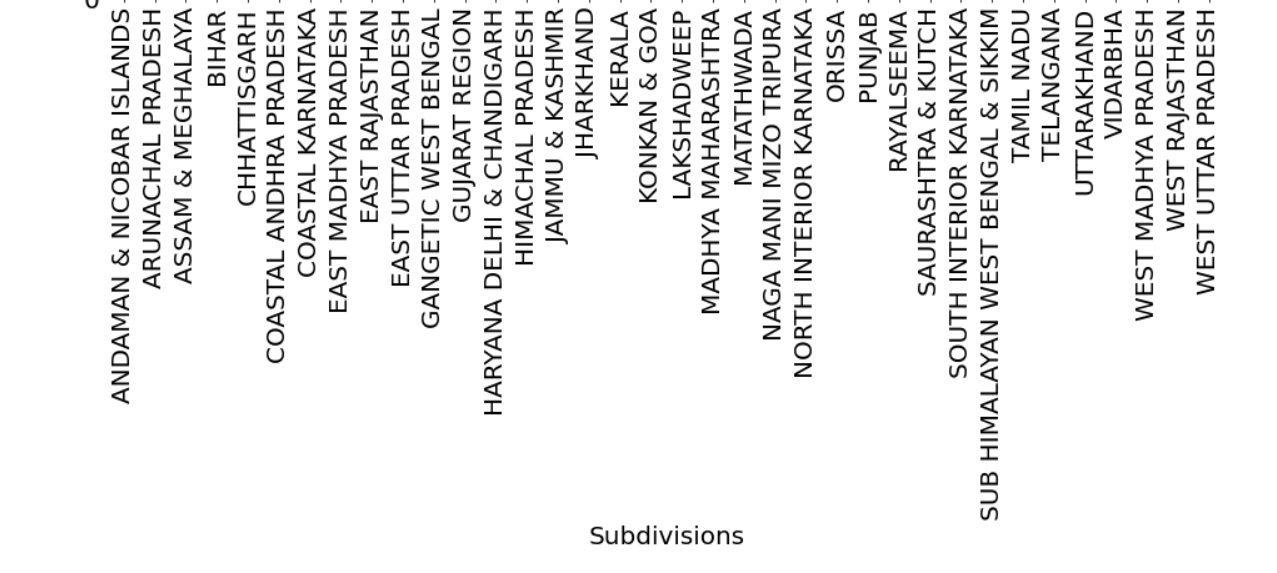
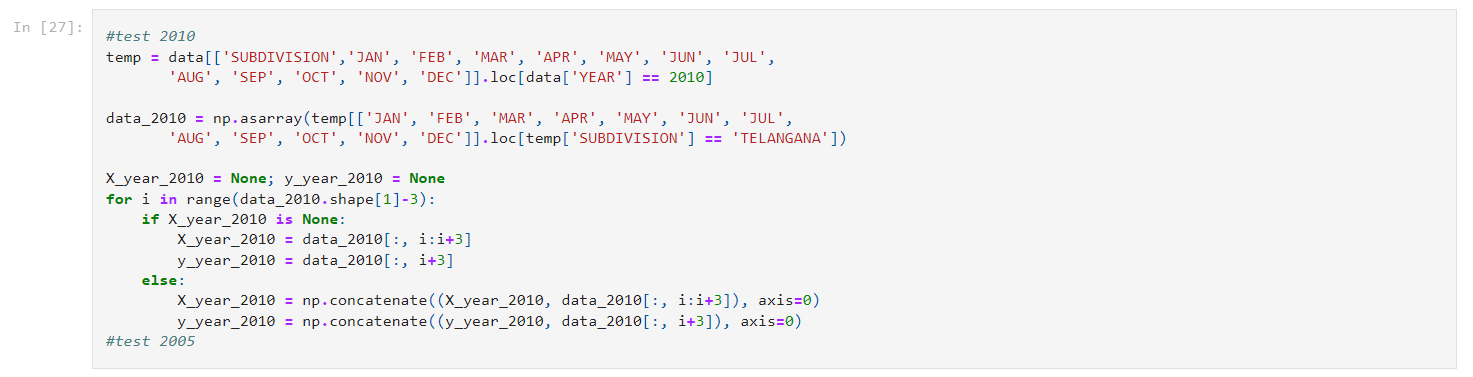
 

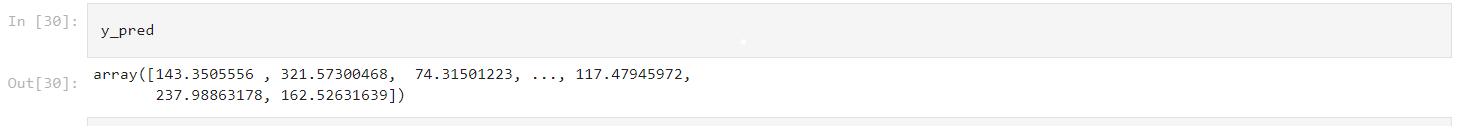
Fig :5.11



30







31

B.SOURCE CODE

In[1]: **import** numpy **as** np

**import** pandas **as** pd *# data processing, CSV file I/O (e.g. pd.read\_csv)*

**import** matplotlib.pyplot **as** plt

**from** sklearn **import** linear\_model

In [2]:**from** subprocess **import** check\_output

In [3]:data **=** pd**.**read\_csv(r'C:\Users\asus\Desktop/rainfall in india.csv')

In [4]:data**.**head(1000)

In [5]:data**.**dropna(how**=**'any', inplace**=True**)

In [6]:data**.**info()

In [7]:subdivs **=** data['SUBDIVISION']**.**unique()

num\_of\_subdivs **=** subdivs**.**size

print('Total # of Subdivs: ' **+** str(num\_of\_subdivs))

subdivs

In[8]: data**.**describe()

In[9]: **import** seaborn **as** sns

corrmat **=** data**.**corr(method **=** "spearman")

plt**.**figure(figsize**=**(18,18))

g**=**sns**.**heatmap(corrmat,annot**=True**)

In[10]: fig **=** plt**.**figure(figsize**=**(16,8))

ax **=** fig**.**add\_subplot(111)

data**.**groupby('SUBDIVISION')**.**mean()**.**sort\_values(by**=**'ANNUAL', ascending**=False**)['ANNUAL']**.**plot(kind**=**'bar', color**=**'g',width**=**0.5,title**=**'Subdivision wise Average Annual Rainfall', fontsize**=**10)

plt**.**xticks(rotation **=** 90)

plt**.**ylabel('Average Annual Rainfall (mm)')

ax**.**title**.**set\_fontsize(30)

32

ax**.**xaxis**.**label**.**set\_fontsize(20)

ax**.**yaxis**.**label**.**set\_fontsize(20)

print(data**.**groupby('SUBDIVISION')**.**mean()**.**sort\_values(by**=**'ANNUAL', ascending**=False**)['ANNUAL'][[0,1,2]])

print(data**.**groupby('SUBDIVISION')**.**mean()**.**sort\_values(by**=**'ANNUAL', ascending**=False**)['ANNUAL'][[33,34,35]])

In[11]: fig **=** plt**.**figure(figsize**=**(16,10))

ax **=** fig**.**add\_subplot(111)

dfg **=** data**.**groupby('YEAR')**.**sum()['ANNUAL']

plt**.**ylabel('Overall Rainfall (mm)')

ax**.**title**.**set\_fontsize(30)

ax**.**xaxis**.**label**.**set\_fontsize(20)

ax**.**yaxis**.**label**.**set\_fontsize(20)

print('Max: ' **+** str(dfg**.**max()) **+** ' ocurred in ' **+** str(dfg**.**loc[dfg **==** dfg**.**max()]**.**index**.**values[0:]))

print('Max: ' **+** str(dfg**.**min()) **+** ' ocurred in ' **+** str(dfg**.**loc[dfg **==** dfg**.**min()]**.**index**.**values[0:]))

print('Mean: ' **+** str(dfg**.**mean()))

In[12]: months **=** data**.**columns[2:14]

fig **=** plt**.**figure(figsize**=**(18,10))

ax **=** fig**.**add\_subplot(111)

data**.**groupby('YEAR')**.**mean()[months]**.**plot**.**line(title**=**'Overall Rainfall in Each Month of Year', ax**=**ax,fontsize**=**20)

*#plt.xticks(np.linspace(0,35,36,endpoint=True),xlbls)*

plt**.**xticks( rotation **=** 90)

plt**.**ylabel('Rainfall (mm)')

plt**.**legend(loc**=**'upper right', fontsize **=** 'x-large')

ax**.**title**.**set\_fontsize(30)

33

ax**.**xaxis**.**label**.**set\_fontsize(20)

ax**.**yaxis**.**label**.**set\_fontsize(20)

In[13]: data**.**hist(figsize**=**(24,24))

In[14]: **from** sklearn.model\_selection **import** train\_test\_split

y **=** data['YEAR']

X **=** data**.**drop('YEAR', axis**=**1)

In [15]:X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, train\_size**=**0.8)

In [16]:X

In[17]:y

X\_train

y\_train

X\_test

y\_test

In[18]: df2 **=** data[['SUBDIVISION',months[0],months[1],months[2],months[3]]]

*#df2 = df['YEAR','JAN','FEB','MAR','APR']*

df2**.**columns **=** np**.**array(['SUBDIVISION', 'x1','x2','x3','x4'])

**for** k **in** range(1,9):

df3 **=** data[['SUBDIVISION',months[k],months[k**+**1],months[k**+**2],months[k**+**3]]]

df3**.**columns **=** np**.**array(['SUBDIVISION', 'x1','x2','x3','x4'])

df2 **=** df2**.**append(df3)

df2**.**index **=** range(df2**.**shape[0])

*#df2 = pd.concat([df2, pd.get\_dummies(df2['SUBDIVISION'])], axis=1)*

df2**.**drop('SUBDIVISION', axis**=**1,inplace**=True**)

*#print(df2.info())*

34

msk **=** np**.**random**.**rand(len(df2)) **<** 0.8

df\_train **=** df2[msk]

df\_test **=** df2[**~**msk]

df\_train**.**index **=** range(df\_train**.**shape[0])

df\_test**.**index **=** range(df\_test**.**shape[0])

reg **=** linear\_model**.**LinearRegression()

reg**.**fit(df\_train**.**drop('x4',axis**=**1),df\_train['x4'])

predicted\_values **=** reg**.**predict(df\_train**.**drop('x4',axis**=**1))

residuals **=** predicted\_values**-**df\_train['x4']**.**values

print('MAD (Training Data): ' **+** str(np**.**mean(np**.**abs(residuals))))

df\_res **=** pd**.**DataFrame(residuals)

df\_res**.**columns **=** ['Residuals']

fig **=** plt**.**figure(figsize**=**(18,10))

ax **=** fig**.**add\_subplot(111)

df\_res**.**plot**.**line(title**=**'Different b/w Actual and Predicted (Training Data)', color **=** 'k', ax**=**ax,fontsize**=**20)

ax**.**xaxis**.**set\_ticklabels([])

plt**.**ylabel('Residual')

ax**.**title**.**set\_fontsize(30)

ax**.**xaxis**.**label**.**set\_fontsize(20)

ax**.**yaxis**.**label**.**set\_fontsize(20)

predicted\_values **=** reg**.**predict(df\_test**.**drop('x4',axis**=**1))

residuals **=** predicted\_values**-**df\_test['x4']**.**values

print('MAD (Test Data): ' **+** str(np**.**mean(np**.**abs(residuals))))

df\_res **=** pd**.**DataFrame(residuals)

35

df\_res**.**columns **=** ['Residuals']

fig **=** plt**.**figure(figsize**=**(18,10))

ax **=** fig**.**add\_subplot(111)

df\_res**.**plot**.**line(title**=**'Different b/w Actual and Predicted (Test Data)', color**=**'y', ax**=**ax,fontsize**=**20)

ax**.**xaxis**.**set\_ticklabels([])

plt**.**ylabel('Residual')

ax**.**title**.**set\_fontsize(30)

ax**.**xaxis**.**label**.**set\_fontsize(20)

ax**.**yaxis**.**label**.**set\_fontsize(20)

In[19]: df2 **=** data[['SUBDIVISION',months[0],months[1],months[2],months[3]]]

df2**.**columns **=** np**.**array(['SUBDIVISION', 'x1','x2','x3','x4'])

**for** k **in** range(1,9):

df3 **=** data[['SUBDIVISION',months[k],months[k**+**1],months[k**+**2],months[k**+**3]]]

df3**.**columns **=** np**.**array(['SUBDIVISION', 'x1','x2','x3','x4'])

df2 **=** df2**.**append(df3)

df2**.**index **=** range(df2**.**shape[0])

df2 **=** pd**.**concat([df2, pd**.**get\_dummies(df2['SUBDIVISION'])], axis**=**1)

df2**.**drop('SUBDIVISION', axis**=**1,inplace**=True**)

msk **=** np**.**random**.**rand(len(df2)) **<** 0.8

36

df\_train **=** df2[msk]

df\_test **=** df2[**~**msk]

df\_train**.**index **=** range(df\_train**.**shape[0])

df\_test**.**index **=** range(df\_test**.**shape[0])

reg **=** linear\_model**.**LinearRegression()

reg**.**fit(df\_train**.**drop('x4',axis**=**1),df\_train['x4'])

*#print(reg.coef\_)*

predicted\_values **=** reg**.**predict(df\_train**.**drop('x4',axis**=**1))

residuals **=** predicted\_values**-**df\_train['x4']**.**values

df\_res **=** pd**.**DataFrame(residuals)

df\_res**.**columns **=** ['Residuals']

print('MAD (Training Data): ' **+** str(np**.**mean(np**.**abs(residuals))))

fig **=** plt**.**figure(figsize**=**(18,10))

ax **=** fig**.**add\_subplot(111)

df\_res**.**plot**.**line(title**=**'Different b/w Actual and Predicted', color **=** 'c', ax**=**ax,fontsize**=**20)

ax**.**xaxis**.**set\_ticklabels([])

plt**.**ylabel('Residual')

ax**.**title**.**set\_fontsize(30)

ax**.**xaxis**.**label**.**set\_fontsize(20)

ax**.**yaxis**.**label**.**set\_fontsize(20)

predicted\_values **=** reg**.**predict(df\_test**.**drop('x4',axis**=**1))

residuals **=** predicted\_values**-**df\_test['x4']**.**values

37

df\_res **=** pd**.**DataFrame(residuals)

df\_res**.**columns **=** ['Residuals']

print('MAD (Test Data): ' **+** str(np**.**mean(np**.**abs(residuals))))

fig **=** plt**.**figure(figsize**=**(18,10))

ax **=** fig**.**add\_subplot(111)

df\_res**.**plot**.**line(title**=**'Different b/w Actual and Predicted', color**=**'m', ax**=**ax,fontsize**=**20)

ax**.**xaxis**.**set\_ticklabels([])

plt**.**ylabel('Residual')

ax**.**title**.**set\_fontsize(30)

ax**.**xaxis**.**label**.**set\_fontsize(20)

ax**.**yaxis**.**label**.**set\_fontsize(20)

In[20]: df\_res\_training **=** pd**.**DataFrame(columns**=**np**.**array(['Residuals']))

df\_res\_testing **=** pd**.**DataFrame(columns**=**np**.**array(['Residuals']))

list\_mad\_training **=** []

mean\_abs\_diff\_training **=** 0

list\_mad\_testing **=** []

mean\_abs\_diff\_testing **=** 0

**for** subd **in** subdivs:

df1 **=** data[data['SUBDIVISION']**==**subd]

df2 **=** df1[[months[0],months[1],months[2],months[3]]]

df2**.**columns **=** np**.**array(['x1','x2','x3','x4'])

**for** k **in** range(1,9):

df3 **=** df1[[months[k],months[k**+**1],months[k**+**2],months[k**+**3]]]

df3**.**columns **=** np**.**array(['x1','x2','x3','x4'])

df2 **=** df2**.**append(df3)

df2**.**index **=** range(df2**.**shape[0])

38

msk **=** np**.**random**.**rand(len(df2)) **<** 0.8

df\_train **=** df2[msk]

df\_test **=** df2[**~**msk]

df\_train**.**index **=** range(df\_train**.**shape[0])

df\_test**.**index **=** range(df\_test**.**shape[0])

reg **=** linear\_model**.**LinearRegression()

reg**.**fit(df\_train**.**drop('x4',axis**=**1),df\_train['x4'])

predicted\_values **=** reg**.**predict(df\_train**.**drop('x4',axis**=**1))

residuals **=** predicted\_values**-**df\_train['x4']**.**values

df\_res\_training **=** df\_res\_training**.**append(pd**.**DataFrame(residuals,columns**=**np**.**array(['Residuals'])))

mean\_abs\_diff\_training **=** mean\_abs\_diff\_training **+** np**.**sum(np**.**abs(residuals))

list\_mad\_training**.**append(np**.**mean(np**.**abs(residuals)))

predicted\_values **=** reg**.**predict(df\_test**.**drop('x4',axis**=**1))

residuals **=** predicted\_values**-**df\_test['x4']**.**values

df\_res\_testing **=** df\_res\_testing**.**append(pd**.**DataFrame(residuals,columns**=**np**.**array(['Residuals'])))

mean\_abs\_diff\_testing **=** mean\_abs\_diff\_testing **+** np**.**sum(np**.**abs(residuals))

list\_mad\_testing**.**append(np**.**mean(np**.**abs(residuals)))

df\_res\_training**.**index **=** range(df\_res\_training**.**shape[0])

mean\_abs\_diff\_training **=** mean\_abs\_diff\_training**/**df\_res\_training**.**shape[0]

print('Overall MAD (Training): ' **+** str(mean\_abs\_diff\_training))

fig **=** plt**.**figure(figsize**=**(18,10))

39

ax **=** fig**.**add\_subplot(111)

df\_res\_training**.**plot**.**line(title**=**'Residuals (Training)', color**=**'c',ax**=**ax,fontsize**=**20)

*#ax.xaxis.set\_ticklabels([])*

plt**.**ylabel('Residual')

ax**.**title**.**set\_fontsize(30)

ax**.**xaxis**.**label**.**set\_fontsize(20)

ax**.**yaxis**.**label**.**set\_fontsize(20)

df\_res\_testing**.**index **=** range(df\_res\_testing**.**shape[0])

mean\_abs\_diff\_testing **=** mean\_abs\_diff\_testing**/**df\_res\_testing**.**shape[0]

print('Overall MAD (Testing): ' **+** str(mean\_abs\_diff\_testing))

fig **=** plt**.**figure(figsize**=**(18,10))

ax **=** fig**.**add\_subplot(111)

df\_res\_testing**.**plot**.**line(title**=**'Residuals (Testing)', color**=**'m',ax**=**ax,fontsize**=**20)

*#ax.xaxis.set\_ticklabels([])*

plt**.**ylabel('Residual')

ax**.**title**.**set\_fontsize(30)

ax**.**xaxis**.**label**.**set\_fontsize(20)

ax**.**yaxis**.**label**.**set\_fontsize(20)

pd\_mad **=** pd**.**DataFrame(data**=**list\_mad\_training,columns**=**["MAD (Train)"])

pd\_mad["MAD (Test)"] **=** list\_mad\_testing;

pd\_mad['Subdivisions'] **=** subdivs;

fig **=** plt**.**figure(figsize**=**(16,8))

ax **=** fig**.**add\_subplot(111)

*#pd\_mad.groupby('Subdivisions').mean().plot(title='Overall Rainfall in Each Month of Year', ax=ax,fontsize=20)*

40

pd\_mad**.**groupby('Subdivisions')**.**mean()**.**plot**.**bar( width**=**0.5,title**=**'MAD for Subdivisions', ax**=** ax, fontsize**=**20)

plt**.**xticks(rotation **=** 90)

plt**.**ylabel('Mean Abs. Difference')

ax**.**title**.**set\_fontsize(30)

ax**.**xaxis**.**label**.**set\_fontsize(20)

ax**.**yaxis**.**label**.**set\_fontsize(20)

In[21]: **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[22]: *#test 2010*

temp **=** data[['SUBDIVISION','JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',

41

'AUG', 'SEP', 'OCT', 'NOV', 'DEC']]**.**loc[data['YEAR'] **==** 2010]

data\_2010 **=** np**.**asarray(temp[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',

'AUG', 'SEP', 'OCT', 'NOV', 'DEC']]**.**loc[temp['SUBDIVISION'] **==** 'TELANGANA'])

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)

*#test 2005*

In[22]: *#test 2015*

temp **=** data[['SUBDIVISION','JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',

'AUG', 'SEP', 'OCT', 'NOV', 'DEC']]**.**loc[data['YEAR'] **==** 2015]

data\_2015 **=** np**.**asarray(temp[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',

'AUG', 'SEP', 'OCT', 'NOV', 'DEC']]**.**loc[temp['SUBDIVISION'] **==** 'TELANGANA'])

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]

42

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)

In[23]: **from** sklearn **import** linear\_model

*# linear model*

reg **=** linear\_model**.**ElasticNet(alpha**=**0.5)

reg**.**fit(X\_train, y\_train)

y\_pred **=** reg**.**predict(X\_test)

print (mean\_absolute\_error(y\_test, y\_pred))

In[24]: y\_pred

GIT HUB LINK: <https://github.com/prayushveera/Rainfall-prediction-.git>

43