An Interdisciplinary Project Report

on

RAINFALL PREDICTION USING MACHINE LEARNING

Submitted by

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY
ANANTAPUR,ANANTHAPURAMU

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING (DATA SCIENCE)

Submitted By

M .Nithin Reddy-(21691A32165)
S.Dinesh kumar-(21691A3216)
E.Divya Siri Maneela-(21691A3217)

Under the guidance of
Mrs.AnuradhaPrudhvi
Assistant Professor
Department of Computer Science & Engineering

MADANAPALLE INSTITUTE OF TECHNOLGY & SCIENCE
(UGC – AUTONOMOUS)

(Affiliated to JNTUA, Ananthapuramu)
Accredited by NBA, Approved by AICTE, New Delhi)
AN ISO 9001:2008 Certified Institution
P. B. No: 14, Angallu, Madanapalle – 517325

2021-2025

✆ : 08571 - 280255, 280590

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE

AN ENGINEERING COLLEGE SPONSORED BY :
RATAKONDA RANGA REDDY EDUCATIONAL ACADEMY
Approved by AICTE, New Delhi and Affiliated to JNTUA, Anantapur
P.B. No. 14, Angallu, MADANAPALLE - 517 325. Chittoor Dist. A.P.
www.mits.edu

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE

BONAFIDE CERTIFICATE

This is to certify that the internship work entitled **“Rainfall prediction”** is a bonafide work carried out by M .Nithin Reddy-(21691A32165), S.Dinesh kumar-(21691A3216), E.Divya Siri Maneela-(21691A3217) Submitted in partial fulfillment of the requirements for the award of degree **Bachelor of Technology** in the Department of **Data Science**,**Madanapalle Institute of Technology and Science,Madanapalle,** affiliated to **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** during the academic year 2022-2023

Guide	**Head of the Department**
Mrs.AnuradhaPrudhvi	Dr. S. Kusuma,
Assistant Professor	Head &Asssociate Professor
Department of CSE-(DS)	Department of CSE-(DS)

ACKNOWLEDGEMENT

We sincerely thank the **Management** of **Madanapalle Institute of Technology and Science** for providing excellent infrastructure and lab facilities that helped me in completion of my internship.

We sincerely thank **Dr. C. Yuvaraj, M.E., Ph.D**., **Principal** for guiding and providing facilities for the successful completion of my internship at **Madanapalle Institute of Technology and Science,** Madanapalle.

We express our deep sense of gratitude to **Dr.S.Kusuma, Ph.D.,**Associate Professor& Head, Department of Data Science for her continuous support in making necessary arrangements for the successful completion of the internship.

We express our sincere thanks to the **Project Coordinator**, **Mr.G.Rajkumar,** for his tremendous support for the successful completion of the project.

.

WeI express our deep gratitude to our guide Mrs.Anuradha Prudhvi, Assistant Professor od dept of CSE-Data Science for her guidance and encouragement that helped us to complete this project.

We also wish to place on record our gratefulness to other **Faculty members of Department of DS** and also to our friends and our parents for their help and cooperation during our project work.

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE

(UGC-AUTONOMOUS INSTITUTION)

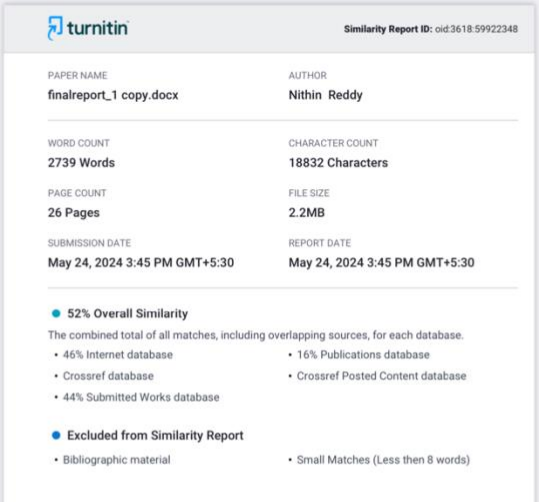
Affiliated to JNTUA, Ananthapuramu& Approved by AICTE, New Delhi
NAAC Accredited with A+ Grade, NIRF India Rankings 2022 - Band: 251-300 (Engg)
NBA Accredited - B.Tech. (CIVIL, CSE, ECE, EEE, MECH), MBA & MCA

RECOGNISED RESEARCH CENTER

Plagiarism Verification Certificate

This is to certify that the B.Tech Project report titled, "**Rainfall Prediction using machine learning"**submitted by **M.Nithin Reddy–21691A3265** has been evaluated using **Anti-Plagiarism Software, Turnitin** and based on the analysis report generated by the softwarethe report's similarity index is found to be 20%.

The following is the Turnitin report for the project report consisting of____ pages.



turnitin

Similarity Report ID: oid:3618:59922348

PAPER NAME finalreport_1 copy.docx	AUTHOR Nithin Reddy
WORD COUNT 2739 Words	CHARACTER COUNT 18832 Characters
PAGE COUNT 26 Pages	FILE SIZE 2.2MB
SUBMISSION DATE May 24, 2024 3:45 PM GMT+5:30	REPORT DATE May 24, 2024 3:45 PM GMT+5:30

● 52% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 46% Internet database
- Crossref database
- 44% Submitted Works database
- 16% Publications database
- Crossref Posted Content database

● Excluded from Similarity Report

- Bibliographic material
- Small Matches (Less then 8 words)

LPlagiarism report

GUIDE
Mrs.AnuradhaPrudhvi
Assistant Professor,
Department of CSE – (DS)

DECLARATION

I, the undersigned hereby declare that the results embodied in this Internship **“Rainfall prediction using machine learning”** is a bonafide record of the work done by me in partial fulfillment of the award of **Bachelor of Technology** in **Data Science** from **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu.** The content of this report i**s** not submitted to any other University/Institute forward of any other degree.

Place:
Date:

PROJECT ASSOCIATES
M .Nithin Reddy
S.Dinesh kumar
E.Divya Siri Maneela

I certify that the above statement made by the students is correct to the best of my knowledge.

Date: **Guide:**

ABSTRACT

Machine Learning is one of the emerging fields of Artificial Intelligence and it has many applications. It is a tool that uses data and Artificial Intelligence in its areas of application. The main idea behind the development of machine learning algorithms is to create a model that understands and analyzes the given data and helps in prediction. Machine learning methods can be applied to various domains.

With the increase in growth of technology, stress and anxiety in an individual are also increasing. Stress is invisible and it's like a slow poison. Stress, tension, and anxiety are the features that could compromise the psychological wellness of individuals. These factors could lead an individual to take their own lives. Therefore, it is important for any person to manage stress, to live a healthy and a balanced life. Many organizations including our government are trying to find the people under stress and get them treated. This can be achieved if there is a machine that can detect stress by understanding texts from posts. The main objective of this project is to create a stress detection machine. The rise of social media is changing people’s life. Now-a-days, with the growing popularity of social media, people are sharing their activities, moods and interactions through the social media posts. If we know whether the person is under stress or not, then we can treat them accordingly. Because, there might be some words that can trigger some sort of anxiety in the people under stress. The main objective of this project is to identify such people through their posts or blogs. The machine reads the posts and detects whether the person is under stress or not.

In the internship I have done on Machine Learning, I did a project on Stress detection. The goal of the project was to create a model to input text from the user and return whether the person is under stress or not. In order to analyze the human language, recognition is not enough so we use natural language processing. For example, if we write “I’m happy” in the input box, since the state you are expressing indicates that you are not under stress then the model should return ‘No stress’. And if the input given is “I need some help”, then the output should be ‘Stress’.

Keywords: Artificial intelligence, Natural Language Processing

CONTENTS

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

1.1 ABOUT MACHINE LEARNING

Machine learning is a sub-domain of computer science. It uses data and artificial intelligence in its area of applications. It is considered as the top-notch pass to the most interesthuging and growing careers in the current world. It is used to make predictions and gain insights. This can be achieved by providing the data to train the model. The learning can be mainly classified into two types, they are

I) Supervised Learning

II) Unsupervised Learning

Supervised Learning: The supervised learning can be achieved by a model if the data provided is labelled.

Classification and regression problems can be solved by using supervised learning

Classification: In this, we need to categorize the given set of data into classes.

Examples: Logistic Regression, Classification trees, Support Vector Machines, Random Forests, Artificial Neural Networks etc.

Regression: In this, we analyse the effect of the independent variable on dependent variables.

Examples: Linear Regression, Decision Trees, Bayesian Networks, Fuzzy Classification etc

Unsupervised Learning: The unsupervised learning can be achieved by a model if the data provided is unlabelled.

Clustering and Dimension reduction problems can be solved by Machine learning.1 using unsupervised learning.

Clustering: In this, we take the unlabelled data and understand it's features and group them accordingly.

Examples: K-means Clustering, Hierarchal Clustering, Gaussian Mixture models, Genetic Algorithms etc.

Dimension Reduction: This is the process of reducing the number of random variables under consideration to make the classification simple.

Examples: Principal Component Analysis, Tensor Decomposition, Multi-dimension Statistics, Random Projections etc.

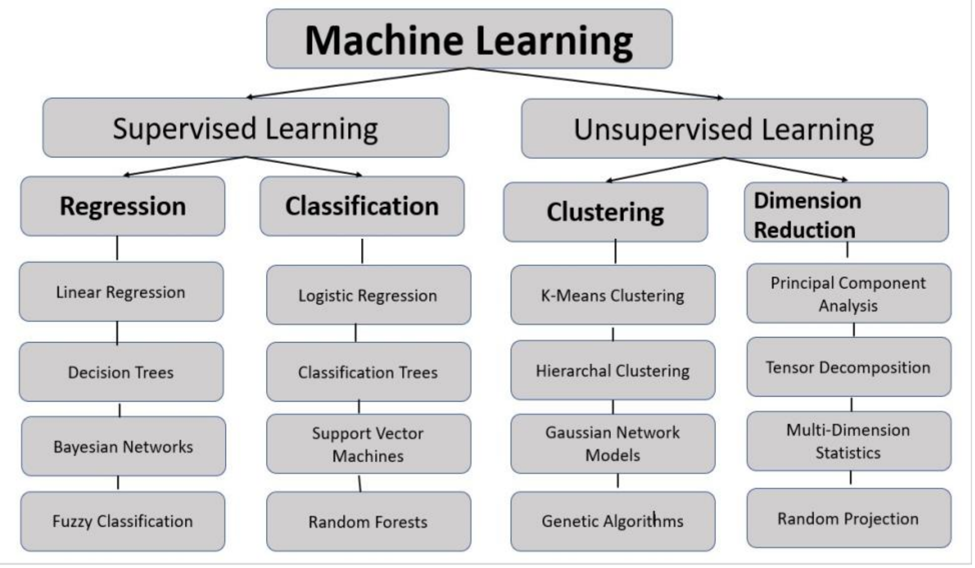


Fig. 1.1 Machine Learning

1.2 IMPORTANCE AND APPLICATIONS OF MACHINE LEARNING

With the growing economy, the world is changing and the internet has become the data generation machine. Machine learning helps the data analysts to organize and handle this data. It helps in analysing the data and provides valuable insights. Machine learning allows the software's to become more accurate.

It is known to everyone that large companies are describing the Machine Learning as "The future". It has many applications in various domains. Few of them are listed below.

1) Image Recognition
2) Automatic language Translation
3) Medical Diagnosis
4) Stock market Trading
5) Online Fraud Detection
6) Virtual Personal Assistant
7) Email Spam and Malware Filtering
8) Self-driving cars
9) Recommendation Systems (Movie recommendation, Music Recommendation) 10) Image recognition

1.3 LANGUAGE USED

For determining the price of diamond, I preferred Python programming language. It was created by
Guido van Rossum, and released in 1991. It is used for web development (server-side), software Development, mathematics, system scripting.

Python programming language (latest Python 3) is being used in web development, Machine Learning applications, along with all cutting edge technology in Software Industry.

Below are some facts about Python Programming Language:

- Python is currently the most widely used multi-purpose, high-level programming language.
- Python allows programming in Object-Oriented and Procedural paradigms.
- Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.
- Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.
- The biggest strength of Python is huge collection of standard library which can be used for The following:
- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks

1.4 NEED FOR THE MODEL

Machine learning is a subfield of artificial intelligence, which is broadly defined as capability of a machine to imitate intelligent human behavior. Is a branch of Artificial Intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn,gradually improving its accuracy. Machine learning classifiers fall into three primary categories. They are supervised Machine Learning, Unsupervised Machine Learning and Semi Supervised Machine Learning. The dataset which I imported was a labelled one so I implemented Sem model using Supervised Machine Learning. Machine learning has seen use cases ranging from predicting customer behavior to forming the operating system for selfdriving cars.Several machine learning algorithms were used to help in prediction of rainfall, among them Linear regression, Random forest regression, Decision tree and K- Neighbours.

CHAPTER – 2

TOOLS AND TECHNIQUES

2.1 PLATFORM USED

Anaconda -IDE(Juypter)

2.2 HARDWARE USED

Device name : Laptop/Desktop

Processor : intel core i5

RAM : 8.00 GB

Hard Disk : 500 GB

System type : 64-bit operating system, x-64 -based processor

2.3 SOFTWARE USED

Platform used for this is Juypter with programming language python.

The domain for this is Artificial Intelligence or Machine learning.

CHAPTER – 3

PROJECT WORK

3.1 PROJECT OVERVIEW

ABSTRACT

Machine learning is used a lot in our daily life now-a-days. So the basic application of the machine learning is finding a solution for problem based on the dataset. It find the solution by prediction based on the dataset. In India rainfall places a major role in the agriculture as it is a main source of survival. These days rainfall prediction become a major problem . By predicting the rainfall it gives an awareness to people and know in advance about rainfall to take certain precautions to protect the crop from the rainfall . Machine learning algorithms is mostly useful in the prediction of the rainfall . Here in this project we use different types of algorithms to the predict the rainfall.

Introduction

The main focus of the study is to forecast Rainfall which is very important because heavy and irregular rainfall can have many impacts like destruction of crops and farms, damage of property so a better forecasting model is essential for an early warning that can minimize risks to life and property and also managing the agricultural farms in better way. This prediction mainly helps farmers and also water resources can be utilized efficiently. Rainfall prediction is a challenging task and the results should be accurate. These traditional methods cannot work in an efficient way so by using machine learning techniques we can produce accurate results. We can just do it by having the historical data analysis of rainfall and can predict the rainfall for future seasons

Form the last few decade scientist and engineers are successfully production several models for making the accurate prediction in several field. Machine learning is also a field which is widely used for the prediction purposes or classifying the things. There are number of methods, listing from KNN, more complex method such as SVM and ANN (Artificial Neural Network). For metrology predictions ANNs pictured as alternative method which opposed to traditional method, are based on self-adaptive mechanisms that learn from examples and capture functional relationships between data, even if the relationships between the data is unknown or difficult to describe

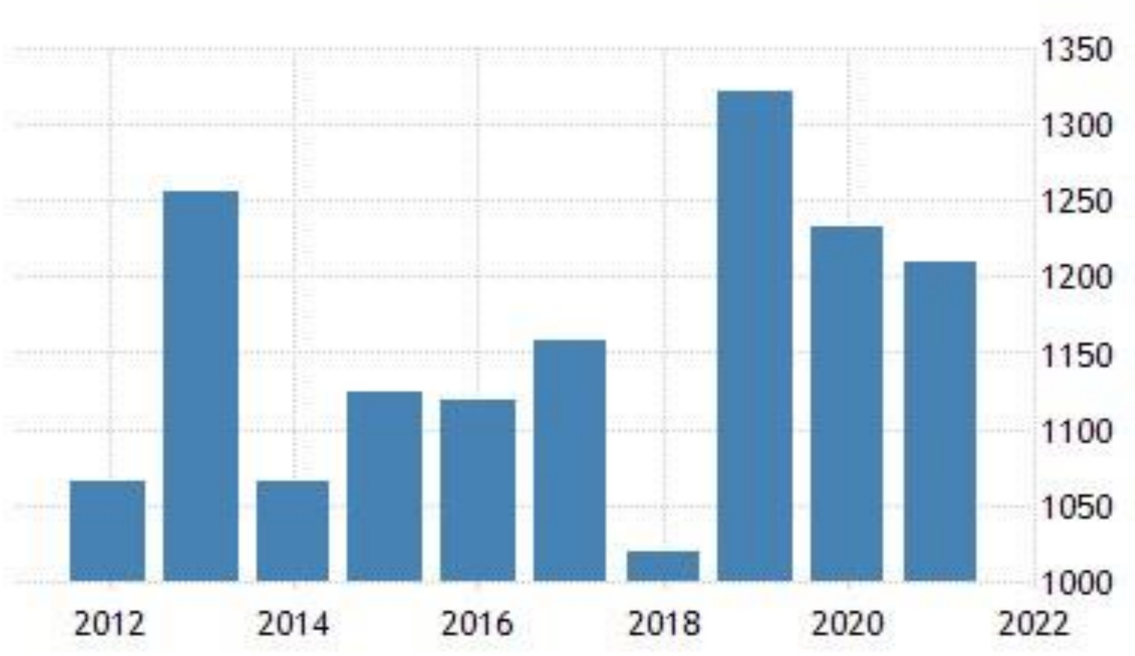
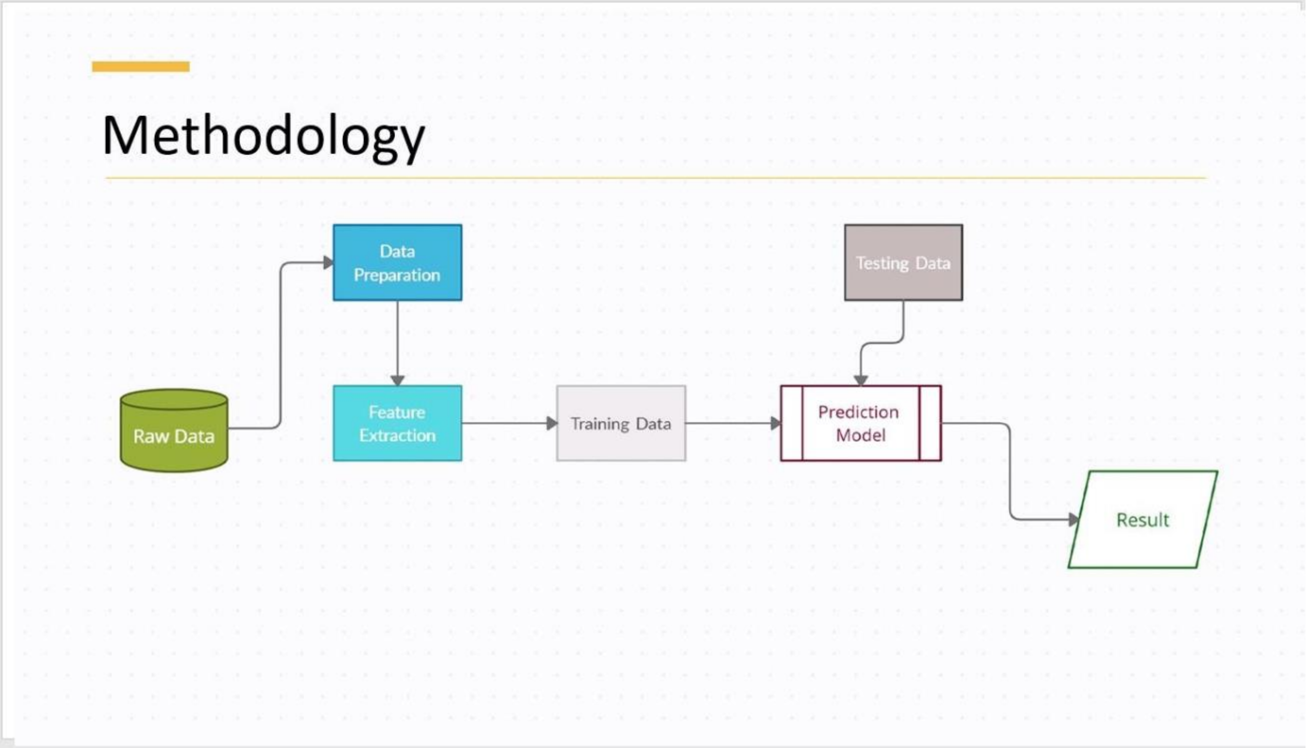


Fig 3.1: Rainfall in India

To solve this uncertainty, we used various machine learning techniques andmodels to make accurate and timely predictions. These paper aims to provide endto end machine learning life cycle right from Data preprocessing to implementing models to evaluating them. Data Preprocessing steps include imputing missingvalues, feature transformation, encoding categorical features, feature scaling andfeature selection. We implemented models such as Logistic Regression, DecisionTree, K Nearest Neighbour, Rule-based and Ensembles

3.2 Methodology:



Methodology

Fig 3.2: Methodology of the project

Raw Data

- We have selected Madanapalle region as the study area (31 mandals)
- Data collected from Sub Collector Office, Madanapalle
- The dataset consists of monthly Rainfall from the year 1988 – 2022
- Meteorological variables are collected from India Meteorological Department.

Data Preparation

- Transforming the raw data by formatting **Feature Extraction**
- process of transforming raw data into numerical features
- Variables considered: Windspeed, Cloud cover, Humidity, Min Temperature, Max

Temperature, Mean Temperature **Raw**

Data

- We have selected Madanapalle region as the study area (31 mandals)
- Data collected from Sub Collector Office, Madanapalle
- The dataset consists of monthly Rainfall from the year 1988 – 2022
- Meteorological variables are collected from India Meteorological Department.

Data Preparation

- Transforming the raw data by formatting **Feature Extraction**
- process of transforming raw data into numerical features
- Variables considered: Windspeed, Cloud cover, Humidity, Min Temperature, Max Temperature, Mean Temperature

3.3Algorithm Linear regression

Linear regression is a supervised learning machine learning algorithm. It carries out a regression task. Based on independent variables, regression models a goal prediction value.

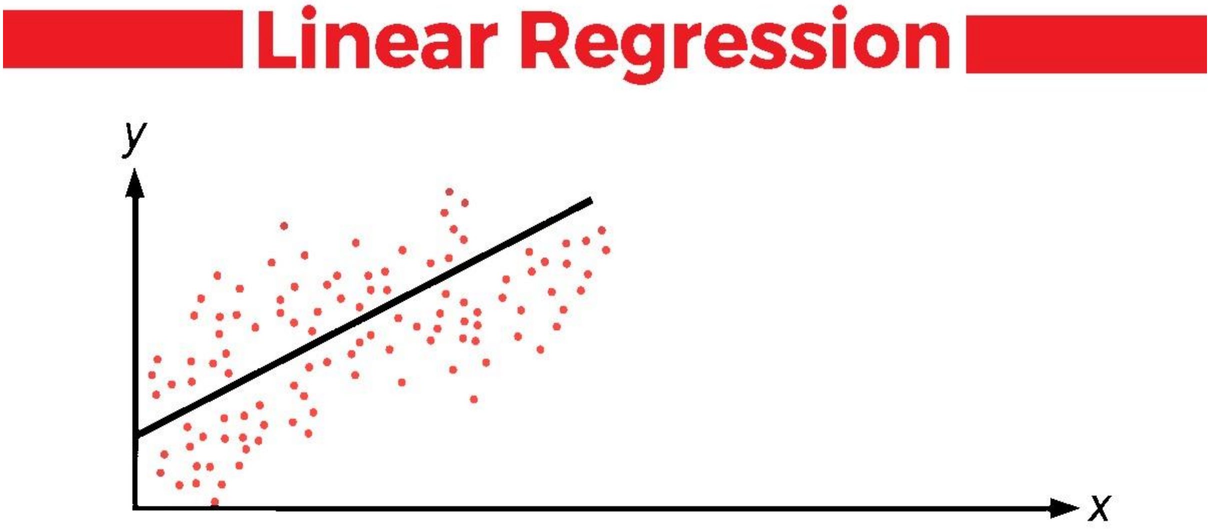


Fig.3.3: Linear regression

Naïve Bayes

The Bayes' Theorem is used to create a collection of classification algorithms known as Naive Bayes classifiers. It is a family of algorithms that share a similar idea, namely that each pair of features being classified is independent of the others.

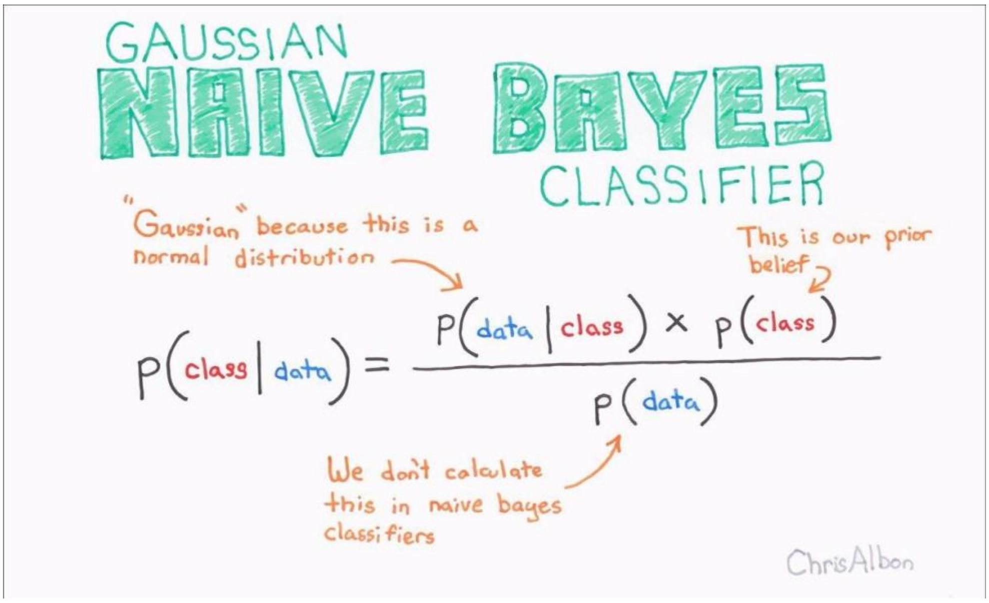

GAUSSIAN
NAIVE BAYES
CLASSIFIER
"Gaussian" because this is a normal distribution
This is our prior belief
P(class | data) = P(data | class) × p(class) / P(data)
We don't calculate this in naive bayes classifiers
ChrisAlbon

Fig.3.4:Naive Bayes

Random Forest

Random forest is a machine learning technique that use a collection of decision trees to produce output with greater flexibility, accuracy, and accessibility. This technique outperforms decision trees because decision trees have worse accuracy than the randam forest algorithm.

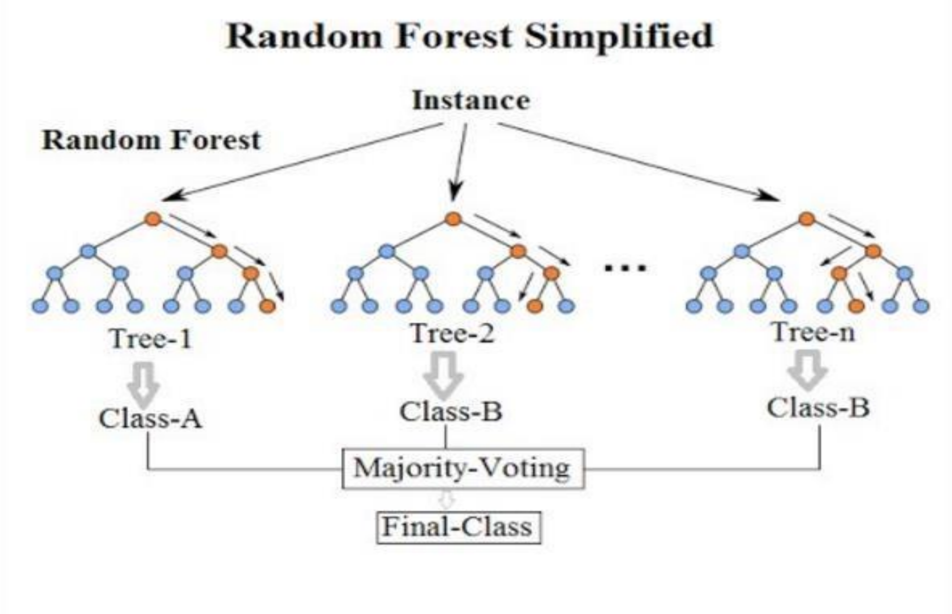


Fig.3.5:Random Forest

Knearest Neighbors

K-Nearest Neighbours is one of Machine Learning's most basic but crucial categorization algorithms. Pattern recognition, data mining, and intrusion detection are just a few of the applications it finds in the supervised learning domain.

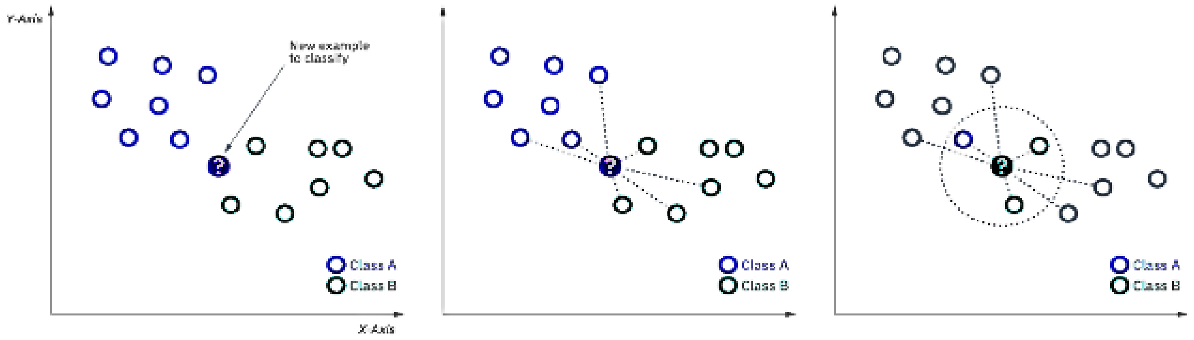


Fig.3.6:KNearestNeighbors

Decision Tree

The most powerful and widely used tool for categorization and prediction is the decision tree. A decision tree is a flowchart-like tree structure in which each internal node represents an attribute test, each branch reflects the test's outcome, and each leaf node (terminal node) stores a class label.

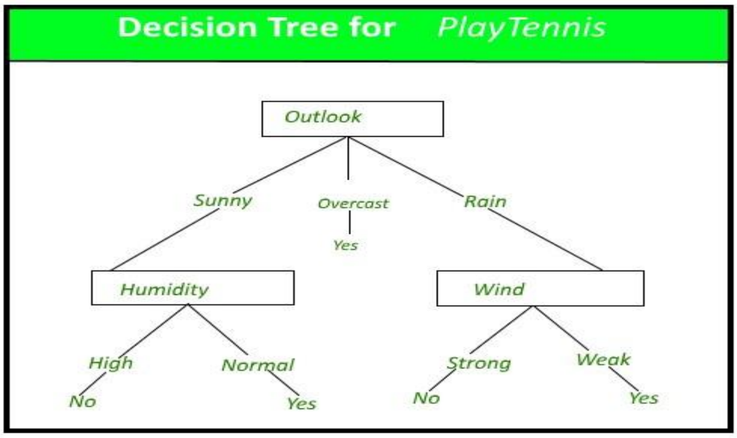


Fig.3.7:Decision Tree

Support Vector Machine

A supervised learning system called a support vector machine is used to solve classification and regression problems. Many people prefer the support vector machine because it produces significant correctness while using less computing power. It's primarily used to solve categorization challenges.

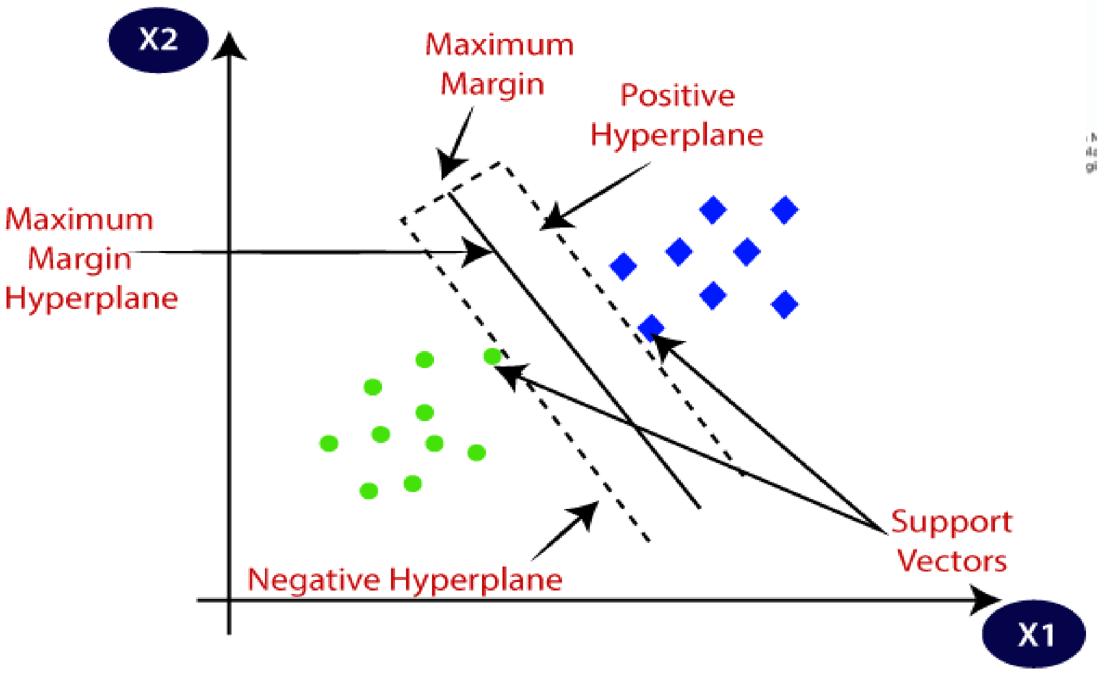


Fig.3.8: Support Vector Machine

Heatmap

A heatmap is a visual representation of data in a color-coded matrix. Color intensity fluctuates depending on the value of the property in the visualisation. A heatmap is a data visualisation that uses a color-coded matrix to display information. Color intensity fluctuates depending on the value of the property in the visualisation

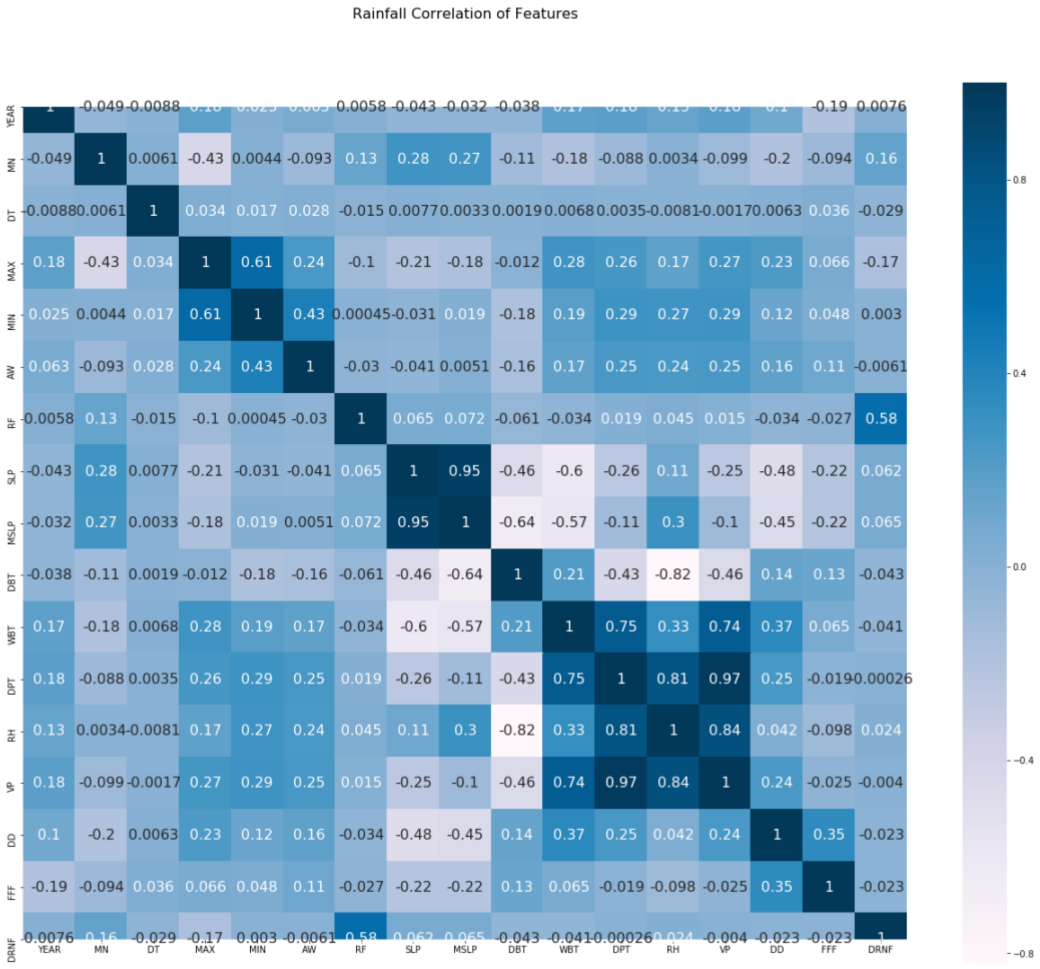


Fig.3.9:Heatmap

Accuracy of Machine learning algorithms

According to the graph below, Random Forest is the most accurate machine learning method when all scores are compared

Linear Regression	: 0.3733002424817602,
KNearest Neighbors	: 0.7788018433179723,
Decision Tree	: 0.7753456221198156,
Support vector Machine	: 0.7723502304147466
Naïve Bayes	: 0.7511520737327189,
Random Forest	: 0.903808694101612

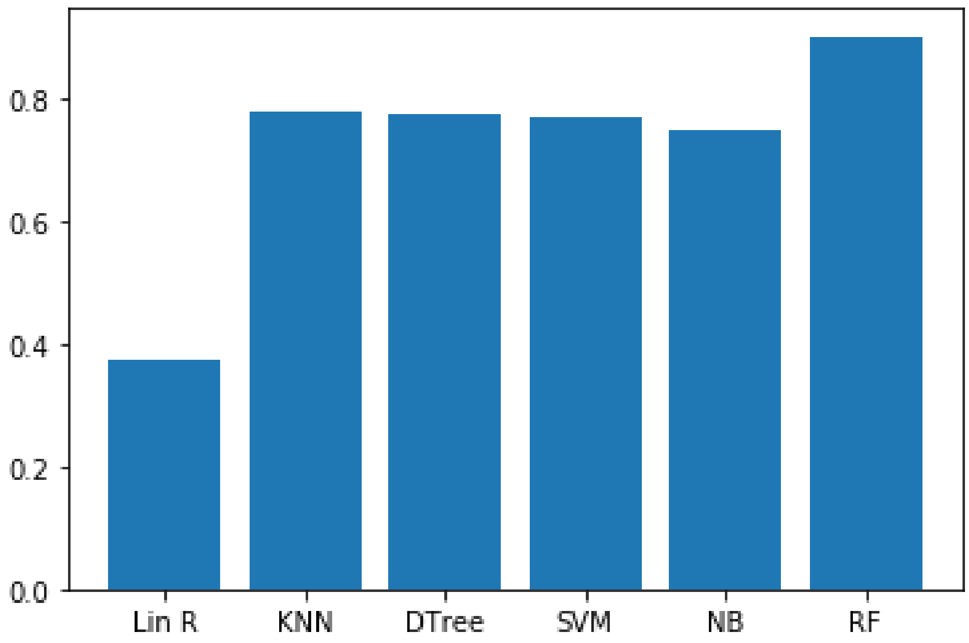


Fig.3.10: Accuracy of Machine Learning Algorithms

CONCLUSION

Finally, based on its accuracy, Random Forest method was chosen as the prediction model.

Implementation of Random Forest

```
def random_forest(l):      train_features,
test_features, train_labels, test_labels = train_test_split(features, Y)
train_features.iloc[-1]=l      model=RandomForestRegressor()
model.fit(train_features, train_labels)
predicted_value=model.predict(train_features)      return predicted_value[-1]
```

BIBLOGRAPHY

[1] Liyew, C.M., Melese H.A. Machine learning techniques to predict daily rainfall amount. J Big Data 8, 153 (2021) 40537-021-00545-4

[2] Hiyam A bobaker, Yousif Ahmed, Sondos W. A. Mohamed "Rainfall Prediction using Multiple Linear Regressions Model" 2020 International Conference on Computer, Control, Electrical, and Electronics Engineering 978-1-7281-9111 2020 IEEE

[3] R. Kingsy Grace, B. Suganya "Machine Learning based Rainfall Prediction International conference on advanced computing and communication systems (ICACCS) ISBN: 978-17281-5197-7 2020 IEEE XPLORE

[4] Nikhil Tiwari, Anmol Singh "A Novel Study of Rainfall in the Indian States and Predictive Analysis using Machine Learning Algorithms" 2020 International Conference on Computational Performance Evaluation 978-1-7281-6644 2020 IEEE

[5] Mylapalle Yeshwanth, Palla Ratna Sai Kumar, Dr. G. Mathivanan "Comparative Study of Machine Learning Algorithms for Rainfall Prediction" International Journal of Trend in Scientific Research and Development ISSN: 2456 – 6470 Volume: 3

[6] N. K. A. Appiah-Badu et al "Rainfall Prediction Using Machine Learning Algorithms for the Various Ecological Zones of Ghana" IEEE Volume 10

[7] Scherko Murad, Yusra Mohammed "Comparable investigation for rainfall forecasting using different data mining approaches in Sulaymaniyah city in Iraq" International journal of environmental science and technology DOI: 10.18488/journal.72.2020.41.11.18

[8] Gurpreet Singh, Deepak kumar "Hybrid prediction models for rainfall forecasting"

Appendix

Source code

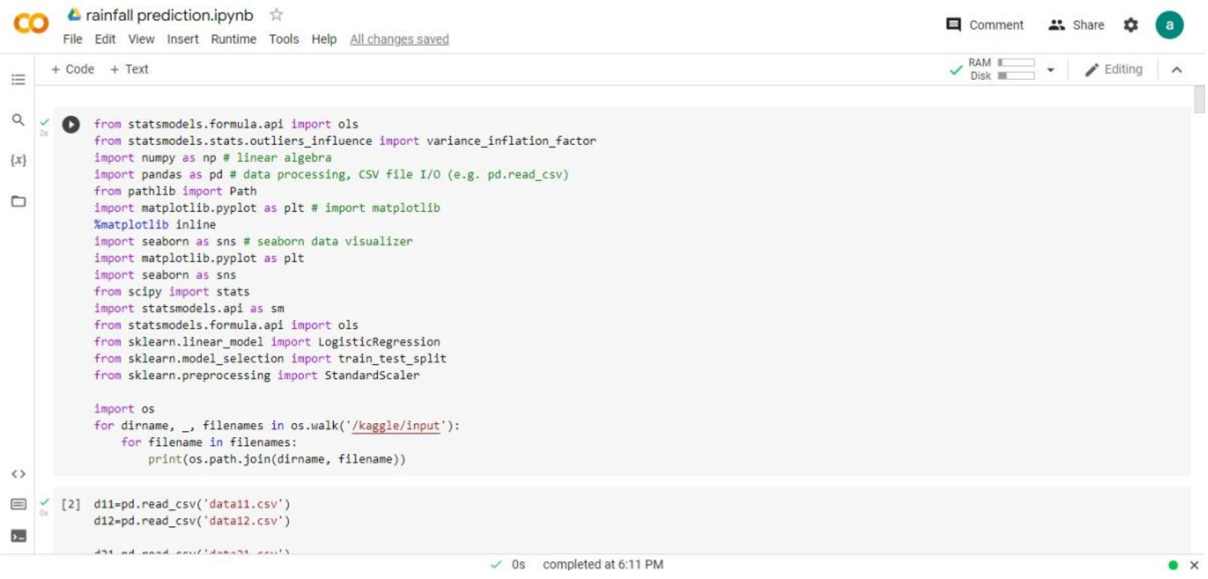
```
from statsmodels.formula.api import ols from
statsmodels.stats.outliers_influence import variance_inflation_fac tor
import numpy as np # linear algebra import pandas as pd # data
processing, CSV file I/O (e.g. pd.read_csv) from pathlib import Path
import matplotlib.pyplot as plt # import matplotlib
%matplotlib inline import seaborn as sns # seaborn
data visualizer import matplotlib.pyplot as plt
import seaborn as sns from scipy import stats import
statsmodels.api as sm from statsmodels.formula.api
```

```
import ols from sklearn.linear_model import
LogisticRegression from sklearn.model_selection
import train_test_split from sklearn.preprocessing
import StandardScaler
 import os for dirname, _, filenames in
os.walk('/kaggle/input'):     for filename in
filenames:         print(os.path.join(dirname,
filename)) d11=pd.read_csv('data11.csv')
d12=pd.read_csv('data12.csv')
 d21=pd.read_csv('data21.csv')
d22=pd.read_csv('data22.csv')
d12.drop(d12[d12['HR']<47].index, inplace = True)
d12.reset_index(inplace = True) d12
#d12.drop(['YEAR'], axis = 1) d12.drop(d12.columns[[0,1, 2,
3,4,14]], axis = 1, inplace = True) d12
df = pd.concat([d11,d12],axis=1) df
#removing the null rows in data base df=df.dropna()
features_list = list(df.drop(columns='RF').columns)
columns = list(df.columns) print(features_list)
m=[] l=[] for i in df['DRNRF(mnts)']:
    if(str(i).replace(" ",'')!=''):
        m+=[float(i)]
    else:         m+=[0] for i in
df['DRNRF(hrs)']:
if(str(i).replace(" ",'')!=''):
        l+=[float(i)*60]
else:         l+=[0] for
i in range(len(m)):
    m[i]+=l[i]
 df['DRNF']=m df
df=df.drop(columns=['DRNRF(hrs)', 'DRNRF(mnts)'])
df.info() FFF=[]
AW=[]
RH=[]
VP=[] DD=[] for i in df['FFF']:
if(str(i).replace(" ",'')!=''):
FFF+=[float(i)]     else:         FFF+=[0] for i in
df['AW']:     if(str(i).replace(" ",'')!='' and
str(i)!='AW'):         AW+=[float(i)]     else:
AW+=[0] for i in df['RH']:     if(str(i).replace("
",'')!=''):         RH+=[float(i)]     else:
RH+=[0] for i in df['VP']:     if(str(i).replace("
",'')!=''):         VP+=[float(i)]     else:
VP+=[0] for i in df['DD']:     if(str(i).replace("
",'')!=''):         DD+=[float(i)]     else:
DD+=[0] df['FFF']=FFF df['AW']=AW df['RH']=RH
df['VP']=VP
df['DD']=DD df.info() columns = list(df.columns)
print(columns) isnull = df.isnull().sum() isnull def
```

```
standardize_var(x):     mean = np.mean(x)     std =
np.sqrt(np.sum(np.square(x-mean))/(len(x)-1))
return  ((x-mean)/std)/np.sqrt(len(x)-1)
  sdf =
df.apply(standardize_var)
sdf_X = sdf[['YEAR', 'MN', 'DT', 'MAX', 'MIN', 'AW', 'SLP', 'MSLP', 'DB
T', 'WBT', 'DPT', 'RH', 'VP', 'DD', 'FFF', 'DRNF']] corr =
np.array(sdf_X.corr())  corr_inv = np.linalg.inv(corr)

fit = ols('RF~YEAR+MN+DT+MAX+MIN+AW+SLP+MSLP+DBT+WBT+DPT+RH+VP+DD+FFF+D
RNF',data=sdf).fit()
  variables = []  reg_coef = []  vif = []
for i in range(len(sdf_X.columns)):
col_name = sdf_X.columns[i]
variables.append(col_name)
reg_coef.append(fit.params[col_name])
vif.append(corr_inv[i][i])
     df_res = pd.DataFrame()
df_res['Variable'] = variables
df_res['Estimate'] = reg_coef
df_res['VIF'] = vif
 df_res
colormap = plt.cm.PuBu  plt.figure(figsize=(22,18))
plt.title("Rainfall Correlation of Features", y = 1.1, size = 16)
sns.heatmap(df.astype(int).corr(), linewidths = 0.0, vmax = 1.0,
square = True, cmap = colormap, linecolor = "white", annot
= True, annot_kws = {"size" : 16})
features = df[['YEAR', 'MN', 'DT', 'MAX', 'MIN', 'AW', 'SLP', 'MSLP', '
DBT', 'WBT', 'DPT', 'RH', 'VP', 'DD', 'FFF', 'DRNF']]
Y = df['RF'] print(features)
train_features, test_features, train_labels, test_labels = train_test_s
plit(features, Y) scaler = StandardScaler()
train_features = scaler.fit_transform(train_features) test_features
= scaler.transform(test_features) from sklearn import preprocessing
lab = preprocessing.LabelEncoder() train_labels =
lab.fit_transform(train_labels) accuracy={} from
sklearn.linear_model import LinearRegression model =
LinearRegression() model.fit(train_features, train_labels)
accuracy["Lin R"]=model.score(train_features, train_labels)
print("LinearRegression:",model.score(train_features, train_labels))
 from sklearn.linear_model import LogisticRegression model =
LogisticRegression() model.fit(train_features, train_labels)
accuracy["Log R"]=model.score(train_features, train_labels)
print("LogisticRegression:",model.score(train_features, train_labels))
from sklearn.linear_model import ARDRegression model = ARDRegression()
model.fit(train_features, train_labels)
print(model.score(train_features, train_labels)) from sklearn.neighbors
import KNeighborsClassifier model = KNeighborsClassifier()
model.fit(train_features, train_labels)
```

```
accuracy["KNN"]=model.score(train_features, train_labels) print("K
Nearest Neighbors:",model.score(train_features, train_labels)) from
sklearn.tree import DecisionTreeClassifier model =
DecisionTreeClassifier(criterion="entropy", max_depth=5)
model.fit(train_features, train_labels)
DecisionTreeClassifier(criterion='entropy', max_depth=5)
accuracy["DTree"]=model.score(train_features, train_labels)
print("Decision Tree:",model.score(train_features, train_labels))
from sklearn import svm model = svm.SVC()
model.fit(train_features, train_labels)
accuracy["SVM"]=model.score(train_features, train_labels)
print("Support Vector Machine:",model.score(train_features, train_label
s))
from sklearn.naive_bayes import GaussianNB model
= GaussianNB()
model.fit(train_features, train_labels) GaussianNB()
accuracy["NB"]=model.score(train_features, train_labels)
print("Naïve Bayes:",model.score(train_features, train_labels))
from sklearn.ensemble import RandomForestRegressor
model=RandomForestRegressor() model.fit(train_features,
train_labels)
RandomForestRegressor()
accuracy["RF"]=model.score(train_features, train_labels) print("Random
Forest:",model.score(train_features, train_labels)) accuracy import
matplotlib.pyplot as plt algo=list(accuracy.keys())
accu=list(accuracy.values())
plt.bar(range(len(accuracy)),accu,tick_label=algo) m=0 algorithm='' for
i in accuracy:     if(accuracy[i]>m):         m=accuracy[i]
algorithm=i print("The algorithm which provides highest algorithm
is",algorithm,"wi th accuracy ",m)  def random_forest(l):
train_features, test_features, train_labels, test_labels = train_te
st_split(features, Y)     train_features.iloc[-1]=l
model=RandomForestRegressor()     model.fit(train_features,
train_labels)     predicted_value=model.predict(train_features)
return predicted_value[-1] user_input=[] for i in features:
    print("Enter the value of ",i,":")
user_input+=[float(input())] phone_number=''
while(len(phone_number)!=10):
    phone_number=input('Enter the whatsapp number:')
try:
        n=int(phone_number)
except:
        print("Enter phone number is not valid.")
if (len(phone_number)!=10):
        print("Enter phone number is not valid.")
result =random_forest(user_input) result
```

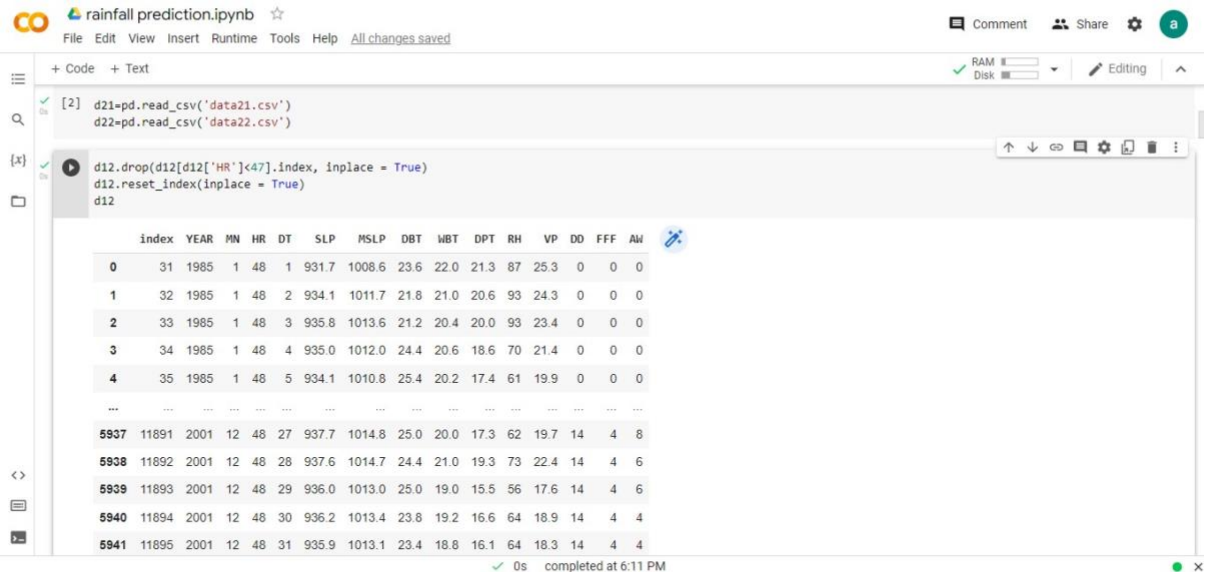
screenshots


rainfall prediction.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing

```
from statsmodels.formula.api import ols
from statsmodels.stats.outliers_influence import variance_inflation_factor
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
from pathlib import Path
import matplotlib.pyplot as plt # import matplotlib
%matplotlib inline
import seaborn as sns # seaborn data visualizer
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
import statsmodels.api as sm
from statsmodels.formula.api import ols
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
[2] d11=pd.read_csv('data11.csv')
    d12=pd.read_csv('data12.csv')
```

Fig A.1- Importing packages

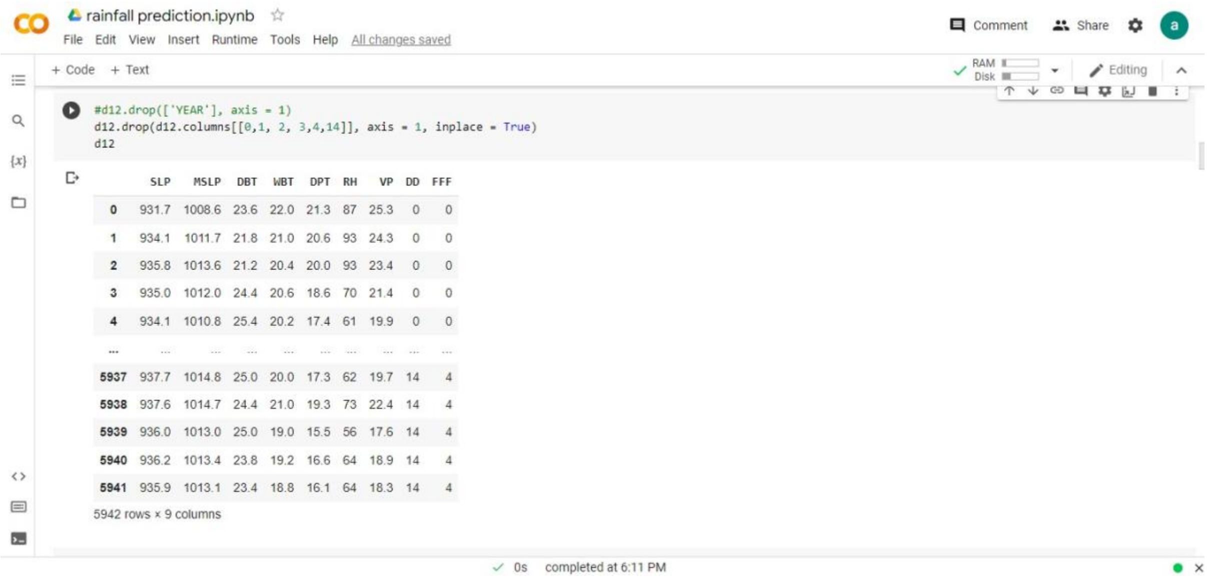

rainfall prediction.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing

```
[2] d21=pd.read_csv('data21.csv')
    d22=pd.read_csv('data22.csv')
```

```
d12.drop(d12[d12['HR']<47].index, inplace = True)
d12.reset_index(inplace = True)
d12
```

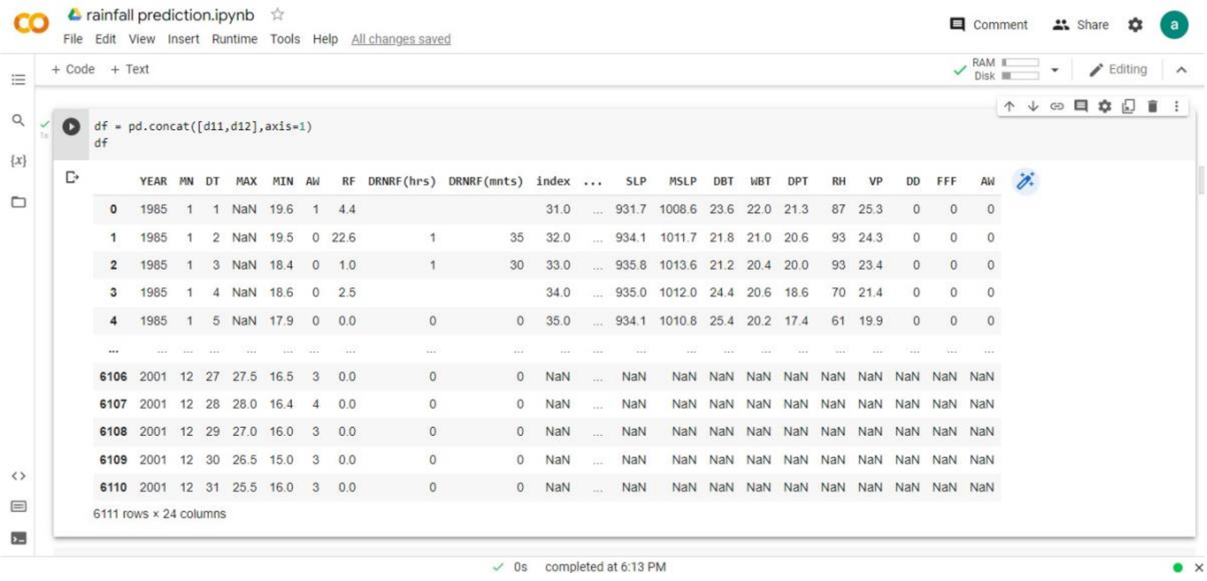
```
      index  YEAR  MN  HR  DT    SLP    MSLP   DBT   WBT   DPT  RH    VP  DD  FFF  AW
   0     31  1985   1  48   1  931.7  1008.6  23.6  22.0  21.3  87  25.3   0    0   0
   1     32  1985   1  48   2  934.1  1011.7  21.8  21.0  20.6  93  24.3   0    0   0
   2     33  1985   1  48   3  935.8  1013.6  21.2  20.4  20.0  93  23.4   0    0   0
   3     34  1985   1  48   4  935.0  1012.0  24.4  20.6  18.6  70  21.4   0    0   0
   4     35  1985   1  48   5  934.1  1010.8  25.4  20.2  17.4  61  19.9   0    0   0
 ...    ...   ...  ..  ..  ..    ...     ...   ...   ...   ...  ..   ...  ..  ...  ..
5937  11891  2001  12  48  27  937.7  1014.8  25.0  20.0  17.3  62  19.7  14    4   8
5938  11892  2001  12  48  28  937.6  1014.7  24.4  21.0  19.3  73  22.4  14    4   6
5939  11893  2001  12  48  29  936.0  1013.0  25.0  19.0  15.5  56  17.6  14    4   6
5940  11894  2001  12  48  30  936.2  1013.4  23.8  19.2  16.6  64  18.9  14    4   4
5941  11895  2001  12  48  31  935.9  1013.1  23.4  18.8  16.1  64  18.3  14    4   4
```

Fig A.2- Importing dataset


rainfall prediction.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
completed at 6:11 PM

```
#d12.drop(['YEAR'], axis = 1)
d12.drop(d12.columns[[0,1, 2, 3,4,14]], axis = 1, inplace = True)
d12
```

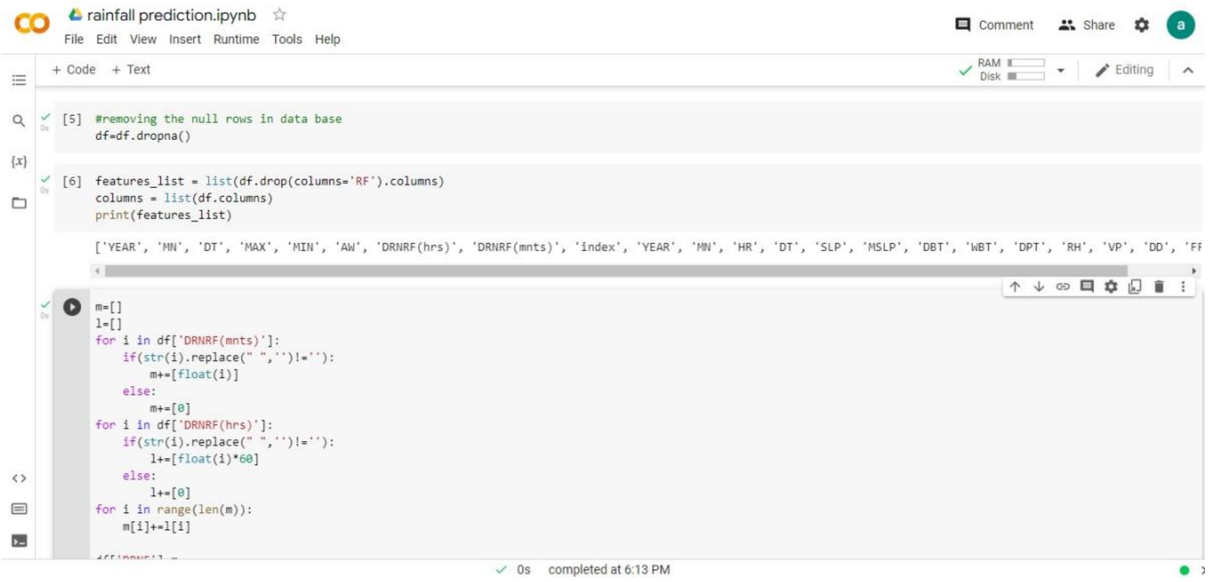
```
        SLP    MSLP   DBT   WBT   DPT  RH    VP  DD  FFF
   0  931.7  1008.6  23.6  22.0  21.3  87  25.3   0    0
   1  934.1  1011.7  21.8  21.0  20.6  93  24.3   0    0
   2  935.8  1013.6  21.2  20.4  20.0  93  23.4   0    0
   3  935.0  1012.0  24.4  20.6  18.6  70  21.4   0    0
   4  934.1  1010.8  25.4  20.2  17.4  61  19.9   0    0
 ...    ...     ...   ...   ...   ...  ...  ...  ...  ...
5937  937.7  1014.8  25.0  20.0  17.3  62  19.7  14    4
5938  937.6  1014.7  24.4  21.0  19.3  73  22.4  14    4
5939  936.0  1013.0  25.0  19.0  15.5  56  17.6  14    4
5940  936.2  1013.4  23.8  19.2  16.6  64  18.9  14    4
5941  935.9  1013.1  23.4  18.8  16.1  64  18.3  14    4
5942 rows × 9 columns
```


rainfall prediction.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
completed at 6:13 PM

```
df = pd.concat([d11,d12],axis=1)
df
```

```
      YEAR  MN  DT   MAX   MIN  AW    RF  DRNRF(hrs)  DRNRF(mnts)  index  ...    SLP    MSLP   DBT   WBT   DPT   RH    VP   DD  FFF   AW
   0  1985   1   1   NaN  19.6   1   4.4                            31.0  ...  931.7  1008.6  23.6  22.0  21.3   87  25.3    0    0    0
   1  1985   1   2   NaN  19.5   0  22.6           1           35   32.0  ...  934.1  1011.7  21.8  21.0  20.6   93  24.3    0    0    0
   2  1985   1   3   NaN  18.4   0   1.0           1           30   33.0  ...  935.8  1013.6  21.2  20.4  20.0   93  23.4    0    0    0
   3  1985   1   4   NaN  18.6   0   2.5                            34.0  ...  935.0  1012.0  24.4  20.6  18.6   70  21.4    0    0    0
   4  1985   1   5   NaN  17.9   0   0.0           0            0   35.0  ...  934.1  1010.8  25.4  20.2  17.4   61  19.9    0    0    0
 ...   ...  ..  ..   ...   ...  ..   ...         ...          ...    ...  ...    ...     ...   ...   ...   ...  ...   ...  ...  ...  ...
6106  2001  12  27  27.5  16.5   3   0.0           0            0    NaN  ...    NaN     NaN   NaN   NaN   NaN  NaN   NaN  NaN  NaN  NaN
6107  2001  12  28  28.0  16.4   4   0.0           0            0    NaN  ...    NaN     NaN   NaN   NaN   NaN  NaN   NaN  NaN  NaN  NaN
6108  2001  12  29  27.0  16.0   3   0.0           0            0    NaN  ...    NaN     NaN   NaN   NaN   NaN  NaN   NaN  NaN  NaN  NaN
6109  2001  12  30  26.5  15.0   3   0.0           0            0    NaN  ...    NaN     NaN   NaN   NaN   NaN  NaN   NaN  NaN  NaN  NaN
6110  2001  12  31  25.5  16.0   3   0.0           0            0    NaN  ...    NaN     NaN   NaN   NaN   NaN  NaN   NaN  NaN  NaN  NaN
6111 rows × 24 columns
```

Fig A.3- Combining dataset


rainfall prediction.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
Editing

```
[5] #removing the null rows in data base
    df=df.dropna()
```

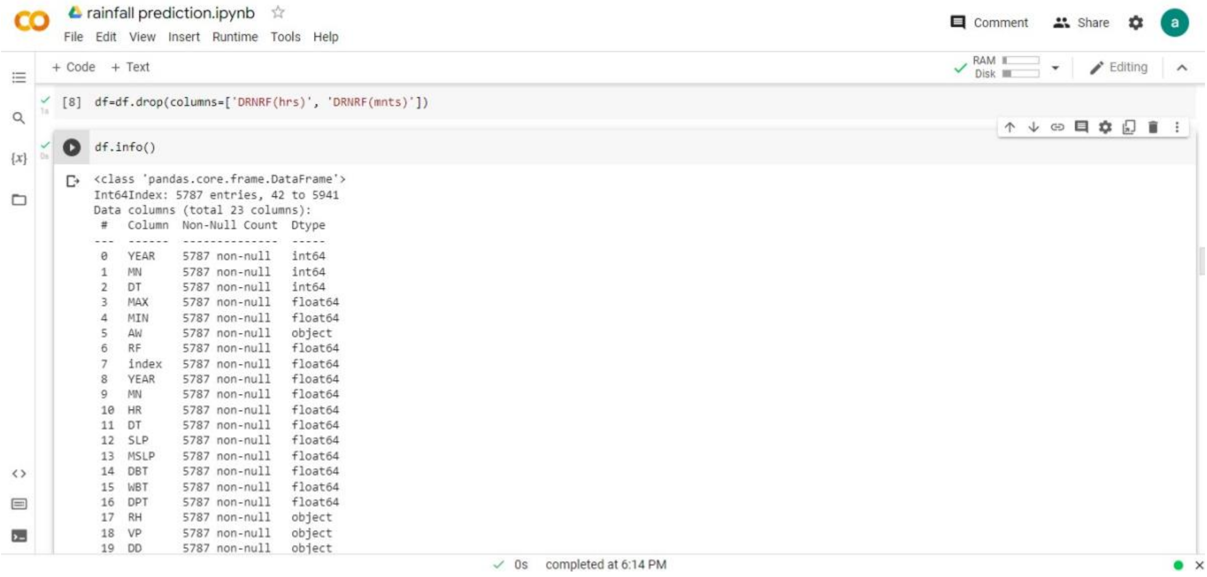
```
[6] features_list = list(df.drop(columns='RF').columns)
    columns = list(df.columns)
    print(features_list)
```

```
['YEAR', 'MN', 'DT', 'MAX', 'MIN', 'AW', 'DRNRF(hrs)', 'DRNRF(mnts)', 'index', 'YEAR', 'MN', 'HR', 'DT', 'SLP', 'MSLP', 'DBT', 'WBT', 'DPT', 'RH', 'VP', 'DD', 'FF
```

```
m=[]
l=[]
for i in df['DRNRF(mnts)']:
    if(str(i).replace(" ",'')!=''):
        m+=[float(i)]
    else:
        m+=[0]
for i in df['DRNRF(hrs)']:
    if(str(i).replace(" ",'')!=''):
        l+=[float(i)*60]
    else:
        l+=[0]
for i in range(len(m)):
    m[i]+=l[i]
```

✓ 0s completed at 6:13 PM

Fig A.4 - Removing the empty values

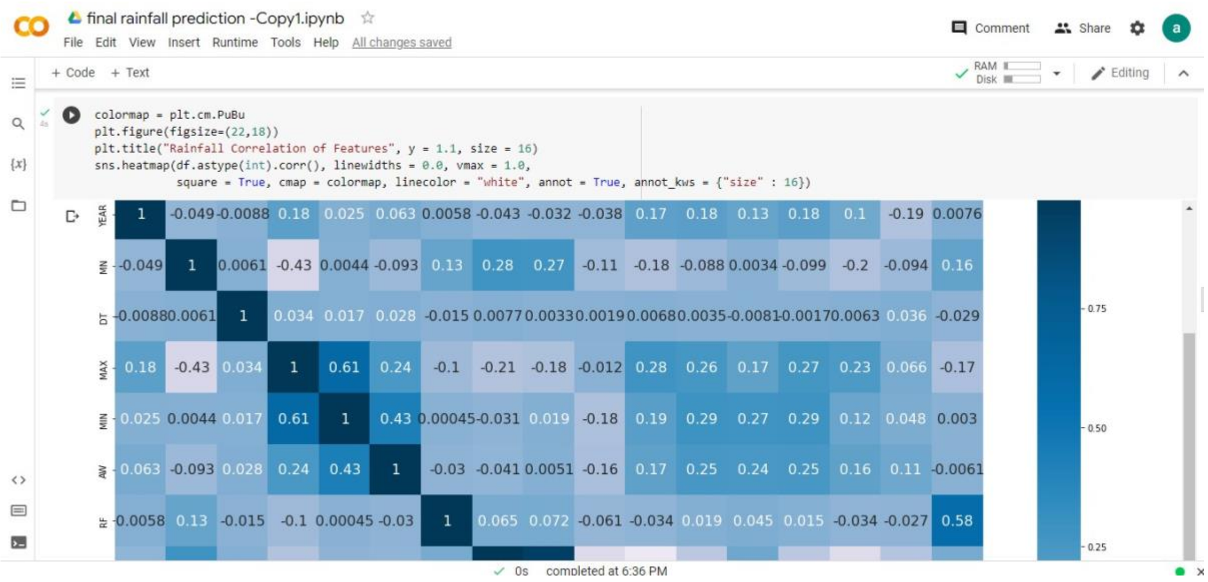

rainfall prediction.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
Editing

```
[8] df=df.drop(columns=['DRNRF(hrs)', 'DRNRF(mnts)'])
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5787 entries, 42 to 5941
Data columns (total 23 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   YEAR    5787 non-null   int64
 1   MN      5787 non-null   int64
 2   DT      5787 non-null   int64
 3   MAX     5787 non-null   float64
 4   MIN     5787 non-null   float64
 5   AW      5787 non-null   object
 6   RF      5787 non-null   float64
 7   index   5787 non-null   float64
 8   YEAR    5787 non-null   float64
 9   MN      5787 non-null   float64
 10  HR      5787 non-null   float64
 11  DT      5787 non-null   float64
 12  SLP     5787 non-null   float64
 13  MSLP    5787 non-null   float64
 14  DBT     5787 non-null   float64
 15  WBT     5787 non-null   float64
 16  DPT     5787 non-null   float64
 17  RH      5787 non-null   object
 18  VP      5787 non-null   object
 19  DD      5787 non-null   object
```

Fig A.5 – Information about the columns in the dataset

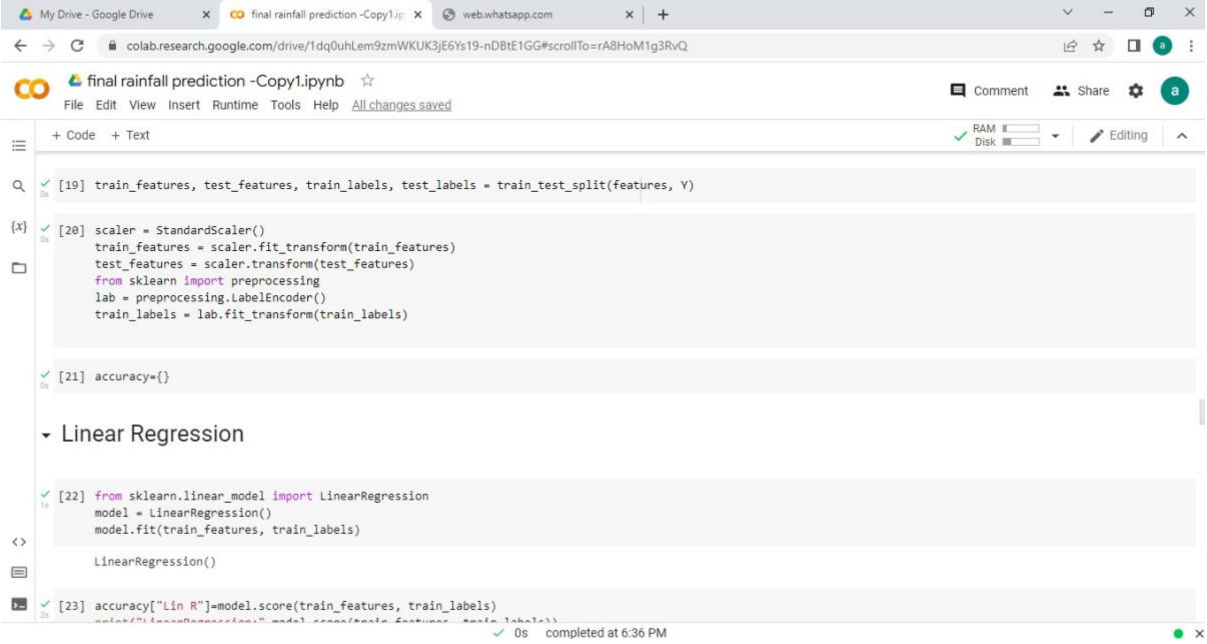

final rainfall prediction -Copy1.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
0.75
0.50
0.25

```
colormap = plt.cm.PuBu
plt.figure(figsize=(22,18))
plt.title("Rainfall Correlation of Features", y = 1.1, size = 16)
sns.heatmap(df.astype(int).corr(), linewidths = 0.0, vmax = 1.0,
            square = True, cmap = colormap, linecolor = "white", annot = True, annot_kws = {"size" : 16})
```

YEAR	1	-0.049	-0.0088	0.18	0.025	0.063	0.0058	-0.043	-0.032	-0.038	0.17	0.18	0.13	0.18	0.1	-0.19	0.0076
MN	-0.049	1	0.0061	-0.43	0.0044	-0.093	0.13	0.28	0.27	-0.11	-0.18	-0.088	0.0034	-0.099	-0.2	-0.094	0.16
DT	-0.0088	0.0061	1	0.034	0.017	0.028	-0.015	0.0077	0.0033	0.0019	0.0068	0.0035	-0.0081	-0.0017	0.0063	0.036	-0.029
MAX	0.18	-0.43	0.034	1	0.61	0.24	-0.1	-0.21	-0.18	-0.012	0.28	0.26	0.17	0.27	0.23	0.066	-0.17
MIN	0.025	0.0044	0.017	0.61	1	0.43	0.00045	-0.031	0.019	-0.18	0.19	0.29	0.27	0.29	0.12	0.048	0.003
AW	0.063	-0.093	0.028	0.24	0.43	1	-0.03	-0.041	0.0051	-0.16	0.17	0.25	0.24	0.25	0.16	0.11	-0.0061
RF	-0.0058	0.13	-0.015	-0.1	0.00045	-0.03	1	0.065	0.072	-0.061	-0.034	0.019	0.045	0.015	-0.034	-0.027	0.58

✓ 0s completed at 6:36 PM

Fig A.6- Heatmap


My Drive - Google Drive
final rainfall prediction -Copy1.ip
web.whatsapp.com
colab.research.google.com/drive/1dq0uhLem9zmWKUK3jE6Ys19-nDBtE1GG#scrollTo=rA8HoM1g3RvQ
final rainfall prediction -Copy1.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing

```
[19] train_features, test_features, train_labels, test_labels = train_test_split(features, Y)
```

```
[20] scaler = StandardScaler()
     train_features = scaler.fit_transform(train_features)
     test_features = scaler.transform(test_features)
     from sklearn import preprocessing
     lab = preprocessing.LabelEncoder()
     train_labels = lab.fit_transform(train_labels)
```

```
[21] accuracy={}
```

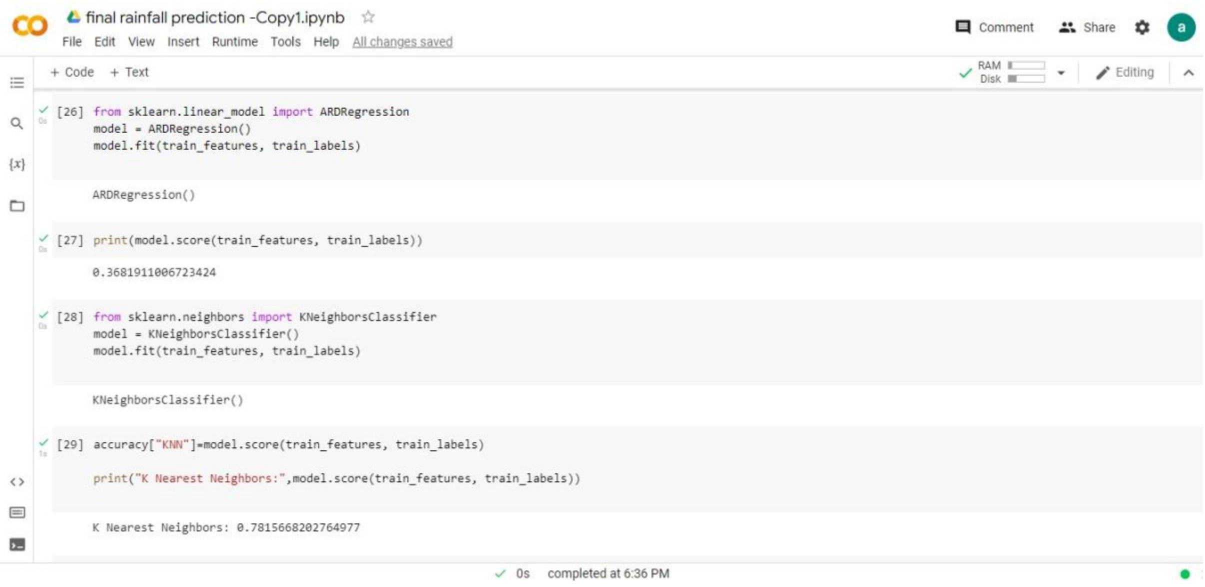
Linear Regression

```
[22] from sklearn.linear_model import LinearRegression
     model = LinearRegression()
     model.fit(train_features, train_labels)

     LinearRegression()
```

```
[23] accuracy["Lin R"]=model.score(train_features, train_labels)
     print("LinearRegression:",model.score(train_features, train_labels))
```

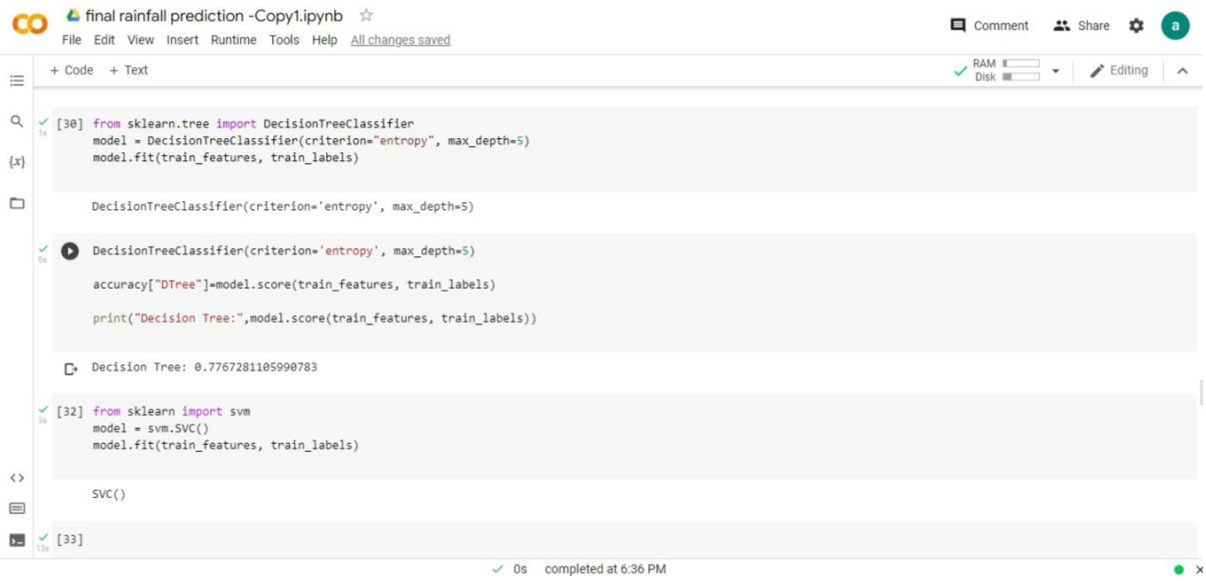
✓ 0s completed at 6:36 PM

Fig A.7- Dividing the data, training of the dataset


final rainfall prediction -Copy1.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
0s completed at 6:36 PM

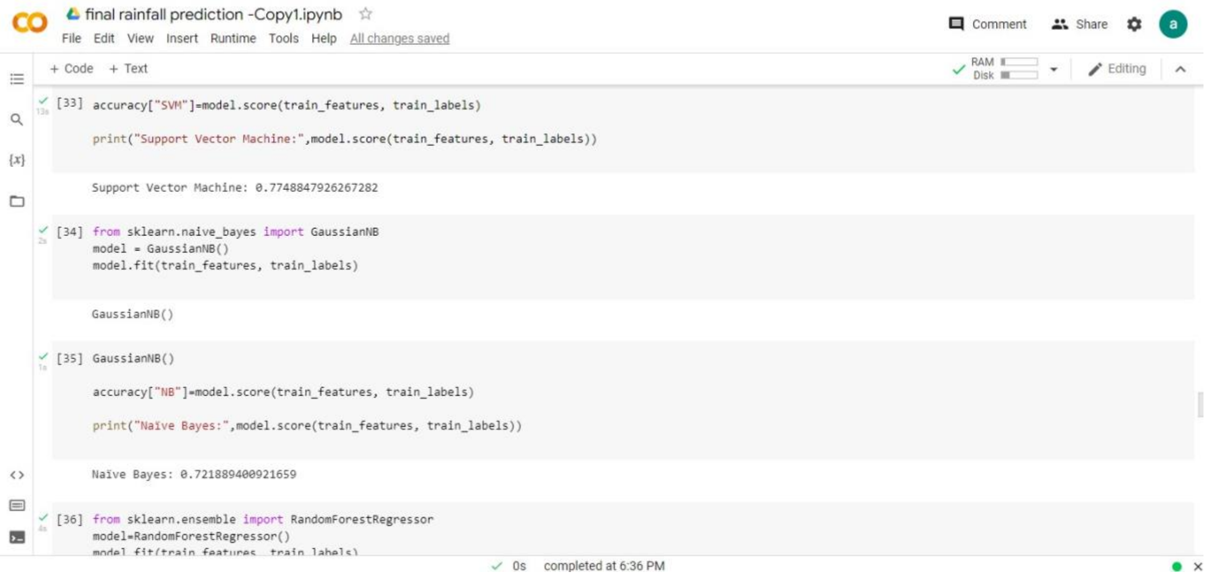
```
[26] from sklearn.linear_model import ARDRegression
     model = ARDRegression()
     model.fit(train_features, train_labels)

     ARDRegression()

[27] print(model.score(train_features, train_labels))

     0.3681911006723424

[28] from sklearn.neighbors import KNeighborsClassifier
     model = KNeighborsClassifier()
     model.fit(train_features, train_labels)

     KNeighborsClassifier()

[29] accuracy["KNN"]=model.score(train_features, train_labels)

     print("K Nearest Neighbors:",model.score(train_features, train_labels))

     K Nearest Neighbors: 0.7815668202764977
```

Fig A.8- Knearest neighbor


final rainfall prediction -Copy1.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
0s completed at 6:36 PM

```
[30] from sklearn.tree import DecisionTreeClassifier
     model = DecisionTreeClassifier(criterion="entropy", max_depth=5)
     model.fit(train_features, train_labels)

     DecisionTreeClassifier(criterion='entropy', max_depth=5)

     DecisionTreeClassifier(criterion='entropy', max_depth=5)

     accuracy["DTree"]=model.score(train_features, train_labels)

     print("Decision Tree:",model.score(train_features, train_labels))

     Decision Tree: 0.7767281105990783

[32] from sklearn import svm
     model = svm.SVC()
     model.fit(train_features, train_labels)

     SVC()

[33]
```

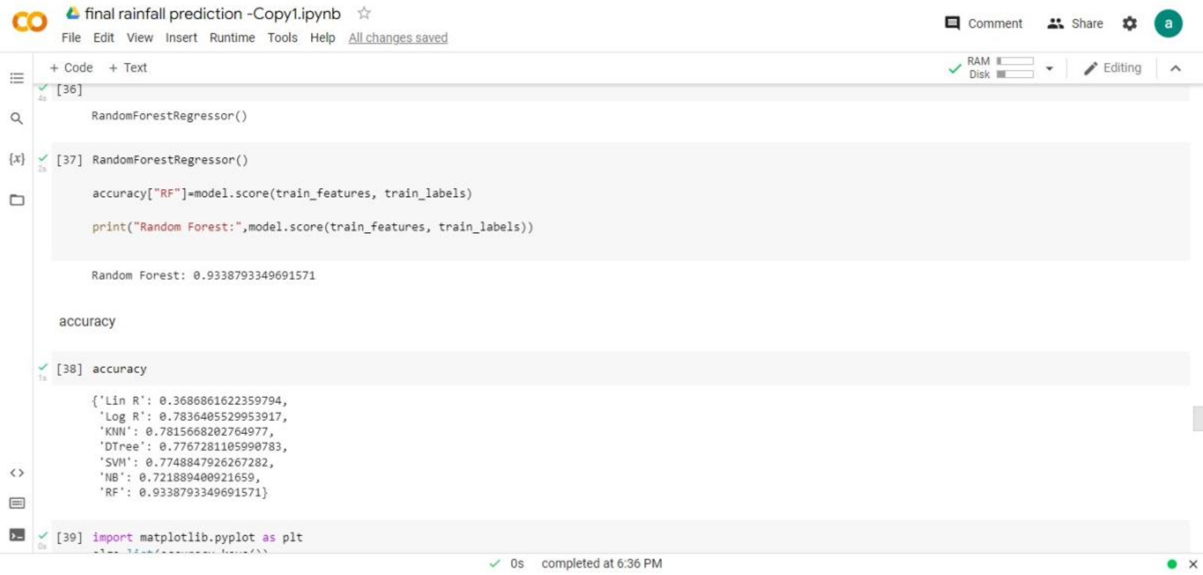
Fig A.9- Decision tree, support vector tree



final rainfall prediction -Copy1.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
0s completed at 6:36 PM

```
[33] accuracy["SVM"]=model.score(train_features, train_labels)

     print("Support Vector Machine:",model.score(train_features, train_labels))

     Support Vector Machine: 0.7748847926267282

[34] from sklearn.naive_bayes import GaussianNB
     model = GaussianNB()
     model.fit(train_features, train_labels)

     GaussianNB()

[35] GaussianNB()

     accuracy["NB"]=model.score(train_features, train_labels)

     print("Naive Bayes:",model.score(train_features, train_labels))

     Naive Bayes: 0.721889400921659

[36] from sklearn.ensemble import RandomForestRegressor
     model=RandomForestRegressor()
     model.fit(train_features, train_labels)
```

Fig A.10- Guassian naïve bayes



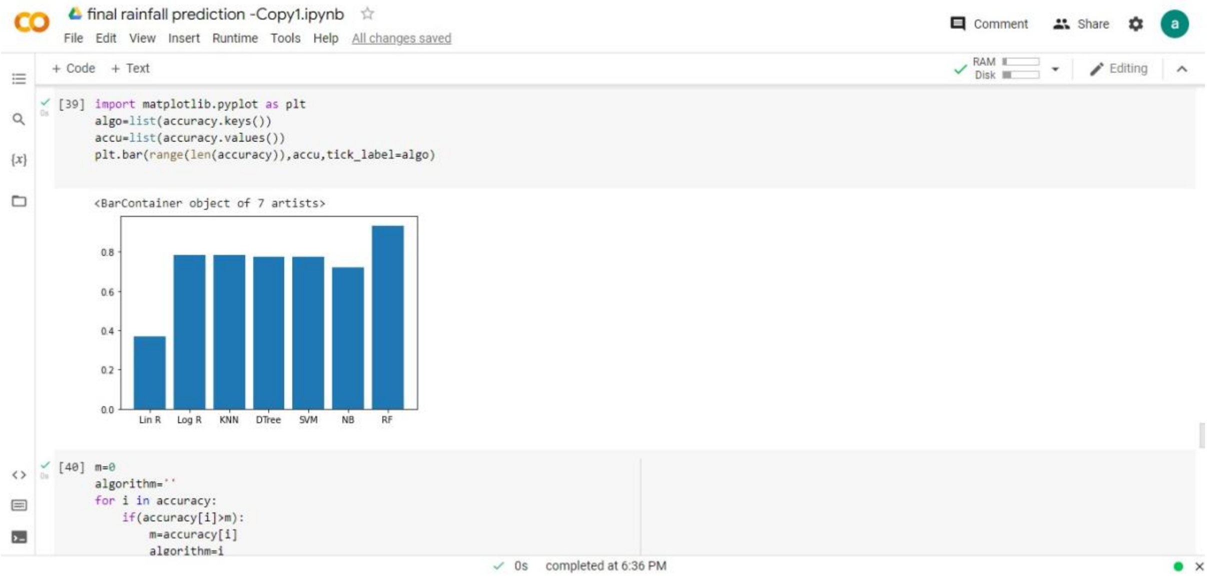
final rainfall prediction -Copy1.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
0s completed at 6:36 PM

```
[36]

     RandomForestRegressor()

[37] RandomForestRegressor()

     accuracy["RF"]=model.score(train_features, train_labels)

     print("Random Forest:",model.score(train_features, train_labels))

     Random Forest: 0.9338793349691571
```

accuracy

```
[38] accuracy

     {'Lin R': 0.3686861622359794,
      'Log R': 0.7836405529953917,
      'KNN': 0.7815668202764977,
      'DTree': 0.7767281105990783,
      'SVM': 0.7748847926267282,
      'NB': 0.721889400921659,
      'RF': 0.9338793349691571}

[39] import matplotlib.pyplot as plt
```

Fig A.11- Random forest



final rainfall prediction -Copy1.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
RAM Disk
Editing
+ Code + Text

```
[39] import matplotlib.pyplot as plt
     algo=list(accuracy.keys())
     accu=list(accuracy.values())
     plt.bar(range(len(accuracy)),accu,tick_label=algo)
```

```
<BarContainer object of 7 artists>
```

0.8
0.6
0.4
0.2
0.0
Lin R
Log R
KNN
DTree
SVM
NB
RF

```
[40] m=0
     algorithm=''
     for i in accuracy:
         if(accuracy[i]>m):
             m=accuracy[i]
             algorithm=i
```

✓ 0s completed at 6:36 PM

Fig A.12- Accuracy