RAIN FALL PREDICTION USING LINEAR REGRESSION

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

By

K. AMRUTHA NIVAS	**2203A51304**
T. NIKITHA	**2203A51325**
S. MANISHA	**2203A51218**
P. HIMA BINDU	**2203A51432**

Under the guidance of

Mr. N.Venkatesh

Associate Professor, CS & AI.

SR UNIVERSITY

Ananthasagar, Warangal.

CERTIFICATE

This is to certify that this project entitled " **RAINFALL PREDICTION USING LINEAR REGRESSION**" is the project work carried out by **K.AMRUTHA NIVAS, T.NIKITHA, S.MANISHA, P.HIMA BINDU** as a project work for the course **Artificial intelligence and Machine learning** to award the degree **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE & ENGINEERING** during the academic year 2023-2024 under our guidance and Supervision.

Mr.N.Venkatesh
Assoc. Prof. CS & AI
SR University ,
Ananthasagar, Warangal.

Dr.M.Sheshikala
Assoc. Prof. & HOD(CSE),
SR University,
Ananthasagar, Warangal.

External Examiner

ACKNOWLEDGEMENT

We owe an enormous debt of gratitude to our project guide Mr. Dr. N.Venkatesh, Assoc. Prof. CS and AI as well as Head of the CSE Department Dr.M.Sheshikala, Associate Professor for guiding us from the beginning through the end of the Capstone Phase-II project with their intellectual advices and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction. Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

ABSTRACT

Rainfall prediction plays a crucial role in various domains, including agriculture, water resource management, and disaster preparedness. In recent years, researchers have explored machine learning models to enhance the accuracy of rainfall predictions. This study focuses on leveraging linear regression, along with other popular models, to improve rainfall forecasting.

The models used like **Linear Regression**: Linear regression is a fundamental statistical technique that establishes a linear relationship between input features and the target variable.

K-Nearest Neighbour (KNN): KNN is a non-parametric algorithm that classifies data points based on their proximity to neighboring points. In rainfall prediction, KNN considers the rainfall patterns of nearby locations to estimate the target value.

Support Vector Machine (SVM): SVM aims to find the hyperplane that best separates different rainfall levels, allowing accurate predictions.

Decision Tree: It split data based on feature values, creating a tree-like structure. By analyzing historical rainfall data, decision trees can predict future precipitation based on specific conditions.

The success of these models heavily relies on the quality and size of the dataset. For this study, we utilize historical weather data collected from automatic weather stations. The dataset includes variables such as temperature, humidity, wind speed, and cloud cover. Spanning various geographical locations and several years, this rich dataset serves as an essential resource for training and evaluating the models. Our experiments demonstrate that combining linear regression with other models significantly improves rainfall prediction accuracy.

TABLE OF CONTENTS

1. INTRODUCTION

Over time, advancements in intelligent computing have led to the development of various techniques for predicting rainfall, with Artificial Neural Networks (ANNs) emerging as a popular choice. ANNs play a vital role in rainfall forecasting, which is crucial for countries like India heavily reliant on agriculture, as it impacts crop yields and water resource management. Additionally, accurate rainfall forecasts are essential for catchment management and flood warning systems. However, predicting rainfall accurately remains challenging due the inherent randomness of weather, spatial and temporal variability, and the dynamic nature of climate phenomena.

Currently, precipitation data is primarily collected through three methods: rain gauges, satellite-derived rainfall data, and radar rainfall estimation. While rain gauge data is accurate, it is limited to localized conditions and lacks spatial representativeness. Satellite and radar data provide broader coverage but are prone to accuracy limitations. Automatic weather stations offer reliable data, but their uneven distribution poses challenges. Although rain gauge data is accurate, its lack of continuity hinders capturing regional climate trends. Existing ground weather stations do not meet today's accuracy demands, necessitating research breakthroughs to enhance forecast precision, especially with the advent of big data.

Our objective is to develop a robust weather forecasting model that utilizes extensive weather data to uncover hidden associations and improve forecast accuracy. This involves not only collecting data on climate, geography, and the environment but also leveraging advanced computational techniques to make precise predictions based on this data—an ongoing challenge in meteorology.

2. LITERATURE REVIEW

In previous research papers, we have observed that different machine learning algorithms have been used. Few papers are based on deep learning also. The field of Artificial Intelligence has been the suitable area to carry out all types of predictions on the dataset by extracting and data preprocessing. Logistic Regression, Support Vector Machine, Naïve Bayes Classification, Linear regression and ridge regression etc. are the various machine learning algorithms the have been used. We have observed that the algorithms work together by generating the pattern among the available dataset and proceeding with prediction. Mid Infrared Spectroscopy combined with few machine learning algorithms. Deep learning is something that works by generating biases and weights in the layers, rule based takes the bulk values and signifies a rule in it. SVM are used with algorithms especially which follows a close correlation among the variables taken into consideration.

Artificial Neural Network inspired by the structure and function of the human brain. PLS regression stands for Partial Square regression, which is a statistical technique used for modelling the relationship between the two sets of variables. In PLS regression, both the predictor variables and the response variables are transformed into new sets of variables called latent variables, which are linear combination of the original variables. PLS regression is useful for predicting a response variable from a large number of predictor variables, even when these variables are highly correlated. It is commonly used in fields such as chemistry, biology, and engineering, where there are many variables to consider in modelling complex systems. It is also used in data analysis and machine learning to identify important variables and reduce dimensionality of the data

Reference	**Model used**	**Accuracy**	**Gaps identified**
M.Kannan et al.	*Global*	*Regression*	*Rainfall,humidity*
S. Chattopadhyay	*Global*	*ANN*	*Rainfall*
P. Dutta, H. Tahbilder	*Global*	*Regression*	*Rainfall*
P. Goswami, Srividya	*Global*	*ANN*	*Mean rainfall*
S. Kannan, S. Ghosh	*Local (river)*	*Decision tree, CART, Kmean*	*Rainfall,humidity*
A. Naik	*Global*	*Monthly*	*Wind,speed,temperature,humidity*
S.nanda	*Global*	*Yearly*	*Min_max temperature*
R.Deshpande	*Local*	*Monthly*	*Rainfall*
G.shrivastava	*Local*	*Yearly*	*Humidity,dew point,pressure*

P.dutta,H.Tahbild er	*Global*	*Monthly*	*Min-Max,temperature,wind direction,humidity,rainfall*

3. PROPOSEDMETHODOLOGY

A.DATA SET

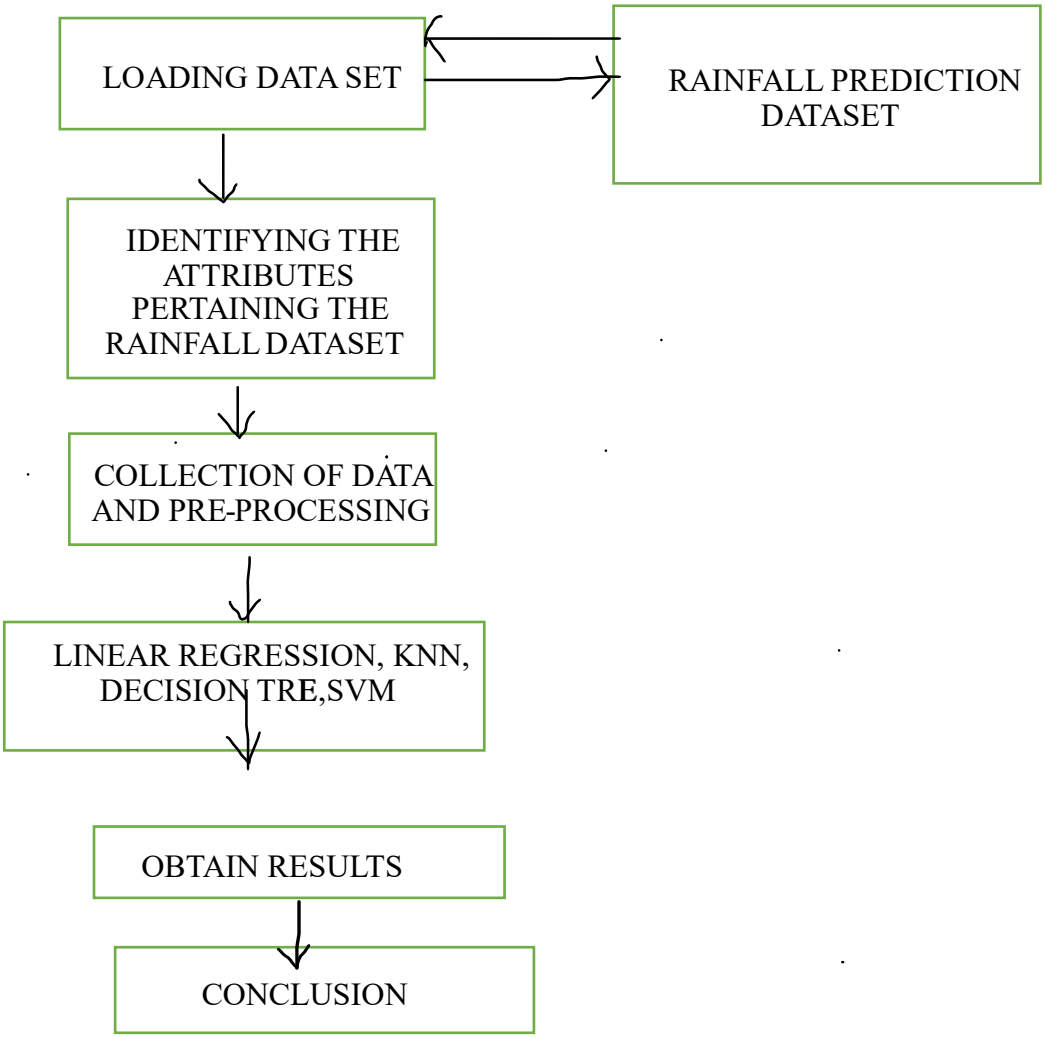
The rainfall prediction dataset is a comprehensive collection of historical weather data from various geographical regions. It encompasses information on average rainfall, wind speed, humidity, temperature, and climate conditions it captures monthly and yearly variations in rainfall patterns across different places.

For each location, the dataset records the highest and lowest average rainfall, providing insights into extreme weather events. Notably, certain regions experience heavy rainfall during specific months, while others remain relatively dry. These variations are critical for understanding local climate dynamics.

Moreover, the dataset includes wind speed data, which correlates with rainfall patterns. High wind speeds often accompany heavy rainfall, while calm winds may indicate drier conditions. Humidity levels also play a significant role, affecting precipitation rates and overall climate comfort.

Temperature data further enriches the dataset, revealing seasonal trends. For instance, tropical regions exhibit consistent warmth, leading to higher average rainfall. In contrast, temperate climates experience distinct seasons, impacting precipitation distribution.

MODEL ARCHITECTURE



Algorithms Implemented:

In this project Dogecoin price prediction and prediction, we use three approaches:

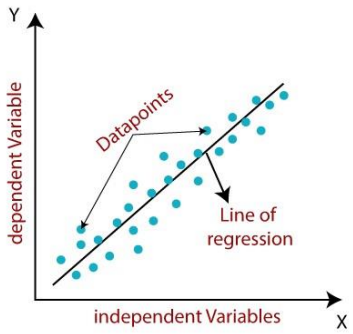
- Linear regression
- K-Nearest Neighbour
- Support Vector Machine
- Decision Tree
- Random Forest

Linear regression:

Linear regression is a supervised machine learning method that is used by the Train Using AutoML tool and finds a linear equation that best describes the correlation of the explanatory variables with the dependent variable. This is achieved by fitting a line to the data using least squares. The line tries to minimiz the sum of the squares of the residuals. The residual is the distance between the line and the actual value of the explanatory variable. Finding the line of best fit is an iterative process.

Advantages of linear regression algorithm:

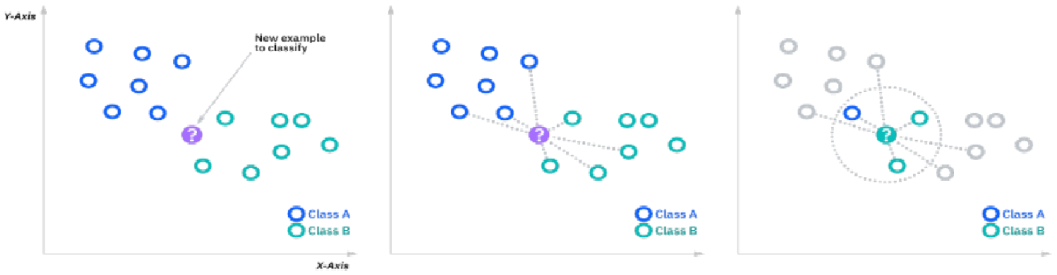
- Linear regression performs exceptionally well for linearly separable data
- Easier to implement, interpret and efficient to train
- It handles overfitting pretty well using dimensionally reduction techniques, regularization, and cross-validation
- One more advantage is the extrapolation beyond a specific data set



K-Nearest Neighbour:

The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.

While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.



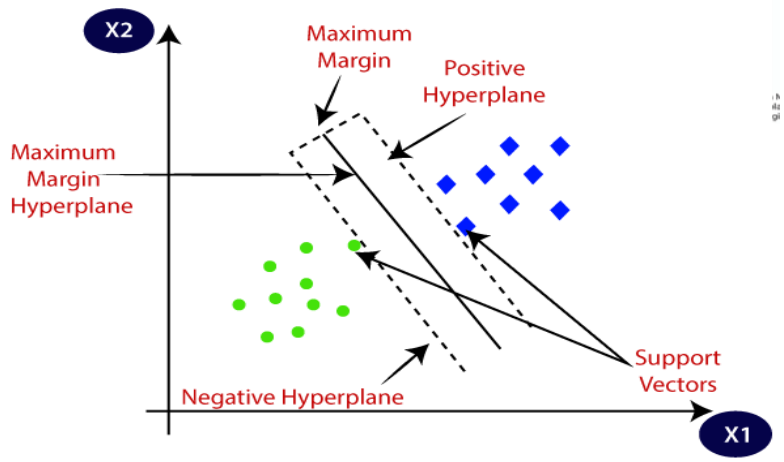
KNN Formula:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$$

Support Vector Machine

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

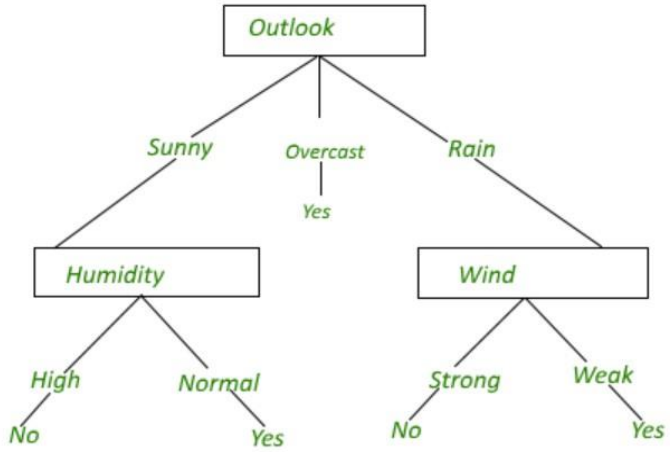


Decision tree

Decision trees are a nonparametric supervised learning method used for classification and regression. The deeper the tree, the more complex the decision rules and the fitter the model. Decision tree uses the tree representation to solve the problem. In which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree. The primary challenge in the decision tree implementation is to identify the attributes. There are two popular attribute selection measures they are Entropy and Gini index. Entropy is the measure of uncertainty of a random variable, it characterizes the impurity of an arbitrary collection of examples. The higher the entropy more the information content

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

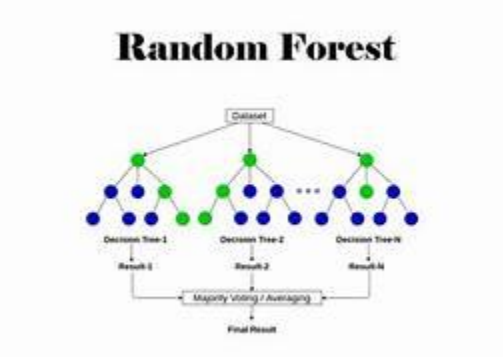
$$IG(T, A) = Entropy(T) - \sum_{v \epsilon A} \frac{|T_v|}{T} \cdot Entropy(T_v)$$

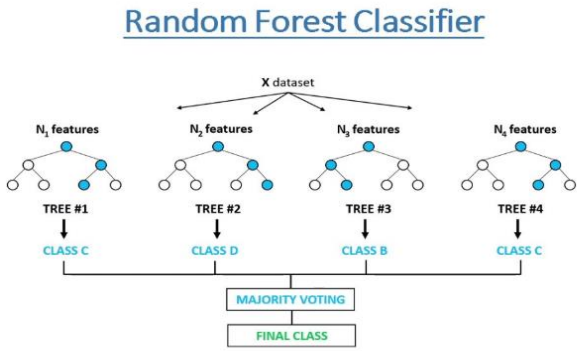


Random Forest

Random Forest is an ensemble learning method that combines multiple decision trees.

It reduces overfitting by averaging the predictions of individual trees. Each tree is trained on a random subset of the data and features. Hyperparameters to tune include the number of trees (estimators) and the maximum depth of each tree.





4. RESULTS

DATASET:

```
import pandas as pd
d=pd.read_csv('/content/rainfall
prediction.csv') print(d)
```

output:

```
          STATE_UT_NAME      DISTRICT   JAN   FEB   MAR   APR  \
0   ANDAMAN And NICOBAR ISLANDS       NICOBAR  107.3  57.9  65.2  117.0
1   ANDAMAN And NICOBAR ISLANDS   SOUTH ANDAMAN   43.7  26.0  18.6  90.5
2   ANDAMAN And NICOBAR ISLANDS   N & M ANDAMAN   32.7  15.9   8.6  53.4
3   ARUNACHAL PRADESH        LOHIT   42.2  80.8  176.4  358.5   4         ARUNACHAL PRADESH     EAST SIANG
    33.3  79.5  105.9  216.5
..                ...         ...   ...   ...   ...   ...   ...
636             KERALA       IDUKKI   13.4  22.1   43.6  150.4
637             KERALA      KASARGOD    2.3   1.0    8.4   46.9
638             KERALA  PATHANAMTHITTA   19.8  45.2   73.9  184.9
639             KERALA       WAYANAD    4.8   8.3   17.5   83.3
640             LAKSHADWEEP    LAKSHADWEEP   20.8  14.7   11.8   48.9


    MAY   JUN   JUL   AUG   SEP   OCT   NOV   DEC  ANNUAL  Jan-Feb  \
```

0 358.5 295.5 285.0 271.9 354.8 326.0 315.2 250.9 2805.2 165.2
1 374.4 457.2 421.3 423.1 455.6 301.2 275.8 128.3 3015.7 69.7
2 343.6 503.3 465.4 460.9 454.8 276.1 198.6 100.0 2913.3 48.6
3 306.4 447.0 660.1 427.8 313.6 167.1 34.1 29.8 3043.8 123.0
4 323.0 738.3 990.9 711.2 568.0 206.9 29.5 31.7 4034.7 112.8
.. ..
636 232.6 651.6 788.9 527.3 308.4 343.2 172.9 48.1 3302.5 35.5
637 217.6 999.6 1108.5 636.3 263.1 234.9 84.6 18.4 3621.6 3.3
638 294.7 556.9 539.9 352.7 266.2 359.4 213.5 51.3 2958.4 65.0
639 174.6 698.1 1110.4 592.9 230.7 213.1 93.6 25.8 3253.1 13.1 640 171.7 330.2 287.7 217.5 163.1
157.1 117.7 58.8 1600.0 35.5

Mar-May Jun-Sep Oct-Dec
0 540.7 1207.2 892.1
1 483.5 1757.2 705.3
2 405.6 1884.4 574.7
3 841.3 1848.5 231.0
4 645.4 3008.4 268.1
.. ..
636 426.6 2276.2
564.2 637 272.9
3007.5 337.9
638 553.5 1715.7 624.2
639 275.4 2632.1 332.5
640 232.4 998.5 333.6

[641 rows x 19
columns]

Linear regression:

```
from sklearn.linear_model import
LinearRegression lr=LinearRegression()
mm=lr.fit(x_train,y_train)
yp=mm.predict(x_test) print(yp)
```

output:

```
[1233.9 1223.4 1327.9 1057.6 2641.8  646.5  961.1 1070.6  485.7 1122.9
1029.6 3470.6 1209.3  308.1 2958.4  498.  2814.4 1796.5 1068.5  646.1
2440.7 1973.9 1081.4 2859.3 1293.1 3468.3  898.2  992.9 1235.7 1535.5
 3094.5  966.7  793.4  449.2  747.1  544.  1803.2  818.   508.1 3218.7
  746.9 2480.6  839.2 1336.5  460.6 1533.5 6379.9 1003.3  837.  1087.7
 2127.5  622.8 1123.6  685.6 1366.2 1680.7 1481.6  788.4  777.  2512.6
  992.2  747.1 1336.5  388.8  863.6 2805.2 1416.2  708.4 1293.3  902.6
  974.9  747.1 1474.3  613.9  449.4  700.4 2731.1 1921.1  807.8 2123.9
 1528.2  655.  1091.6 1618.3 3302.5  572.  1146.8 1385.5 1148.6 1109.9
 2374.1  886.1 2116.9  818.7  897.4 2098.  1005.6  419.5  714.4 1363.3
 1448.3  936.2 1155.4 1062.7  871.5  720.  1008.4  455.6 1192.2 1191.5
```

```
 2814.4  986.3  963.9  252.9  850.1 1229.  1104.7  301.6 1474.1 3399.4
1010.8 1504.4 1530.9 1392.7 1584.9 1462.1  692.7 2556.6 1206.7]
```

```
From       sklearn.metrics       import
mean_squared_error
print(mean_squared_error(yp,y_test))
```

output:

```
2.862281872213113e-25
```

```
from       sklearn.metrics       import
mean_absolute_error
print(mean_absolute_error(yp,y_test))
```

output:

```
3.615505827542091e-13  mse =
mean_squared_error(y_test,
yp)    print("Mean    Squared
Error:", mse)
```

output:

```
Mean Squared Error: 2.862281872213113e-25
 mae                          =
mean_absolute_error(y_test,
yp)    print("Mean    Absolute
Error:", mae)
```

Output:

Mean Absolute Error: 3.615505827542091e-13

K-Nearest Neighbour:

```
from       sklearn.neighbors       import
KneighborsRegressor         knn          =
KNeighborsRegressor(n_neighbors=3)
knn.fit(x_train,    y_train)    y_pred    =
knn.predict(x_test)
 mae                             =
mean_absolute_error(y_test,
y_pred)    print("Mean    Absolute
Error:", mae)
```

output:

Mean Absolute Error: 50.94470284237724

```
mse = mean_squared_error(y_test,
y_pred)    print("Mean    Squared
Error:", mse)
```

output:

```
Mean Squared Error: 7804.861584840647
```

```
 from      sklearn.metrics      import
mean_squared_error
print(mean_squared_error(yp,y_test))
```

output:

```
70357.10162790696
```

```
 from      sklearn.metrics      import
mean_absolute_error
print(mean_absolute_error(yp,y_test))
```

output: 101.26434108527131

Support Vector Machine:

```
from  sklearn.svm  import
SVR        model        =
SVR(kernel='linear')
model.fit(x_train,y_trai
n)
y_pred=model.predict(x_t
est)
 mae                              =
mean_absolute_error(y_test,
y_pred)    print("Mean    Absolute
Error:", mae)
```

output:

Mean Absolute Error: 0.04753588620007877

```
mse = mean_squared_error(y_test,
y_pred)    print("Mean    Squared
Error:", mse)
```

output:

Mean Squared Error: 0.003450724484773055

```
from      sklearn.metrics      import
mean_squared_error
print(mean_squared_error(yp,y_test))
```

output:

```
70357.10162790696
```

```
from sklearn.metrics import mean_absolute_error
print(mean_absolute_error(yp,y_test))
```

output:

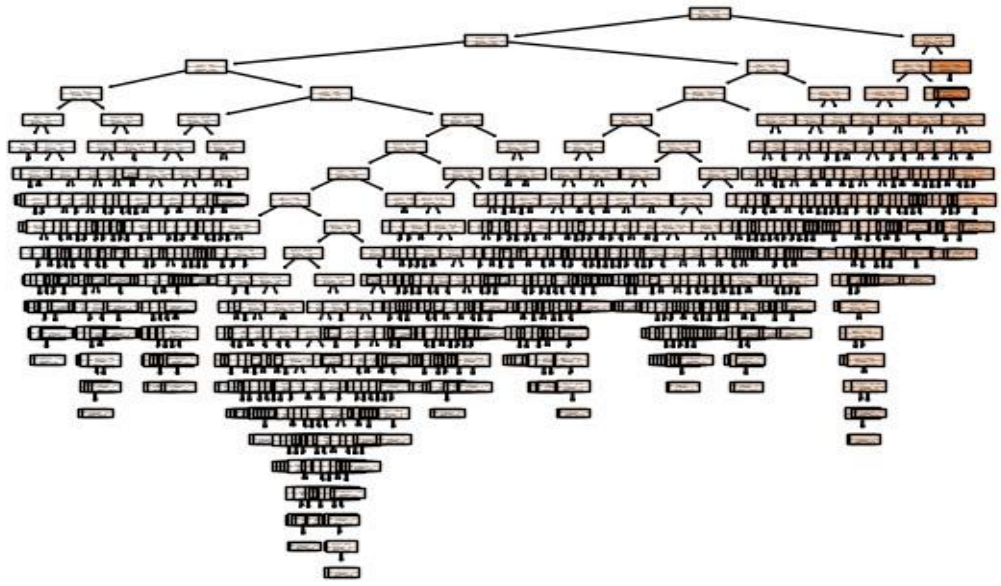
```
101.26434108527131
```

Decision Tree:

```
from sklearn.tree import DecisionTreeRegressor
model=DecisionTreeRegressor()
model.fit(x_train,y_train)
yp=model.predict(x_test)      print(yp)
from sklearn.metrics import mean_squared_error
print(mean_squared_error(y_test,yp))
from sklearn import tree
tree.plot_tree(model,filled=True)
```

output:

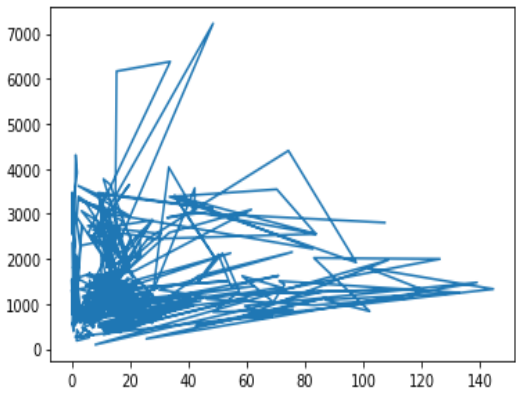
R



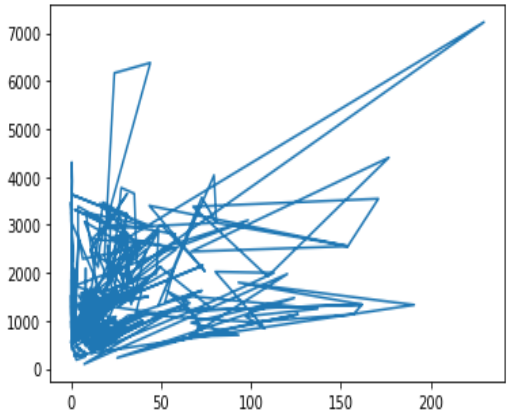
RESULTS:

sno	Machine Learning Model	Mean square error
1	Linear regression	2.862281872213113e-25
2	k-nearest neighbour	7804.861584840647
3	Decision tree	24676.522558139535
4	Support vector machine	0.003450724484773055

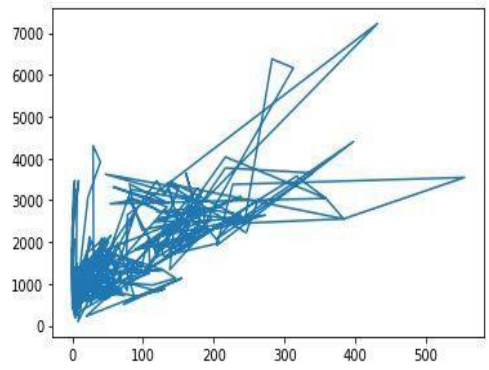
The following are plotting of each feature against the target.



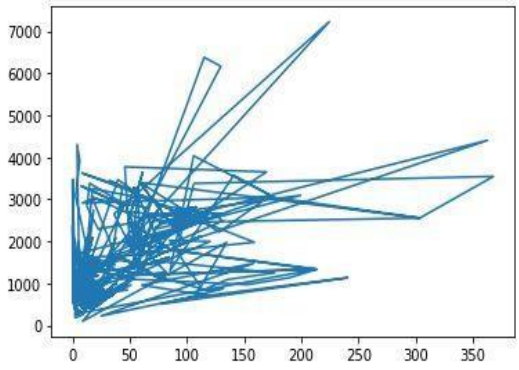
Jan vs annual



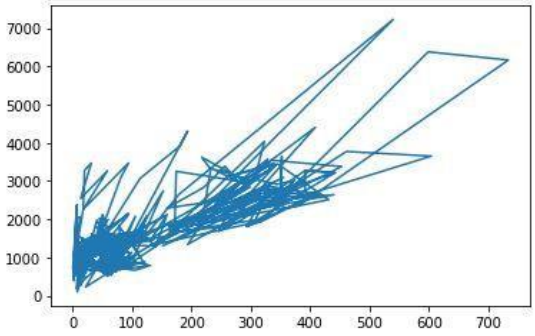
feb



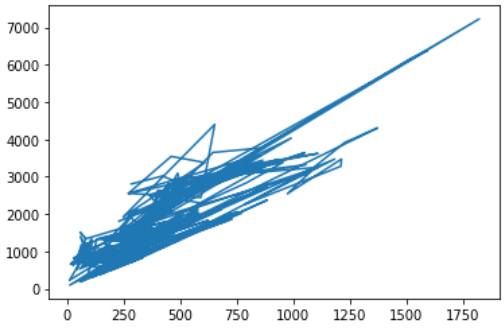
mar



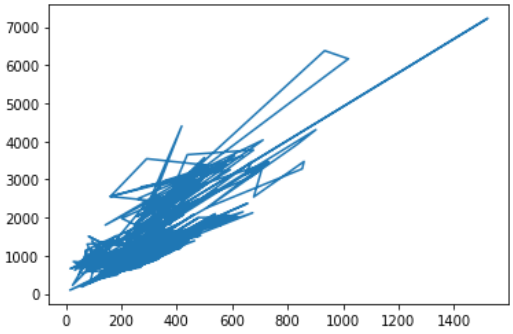
apr



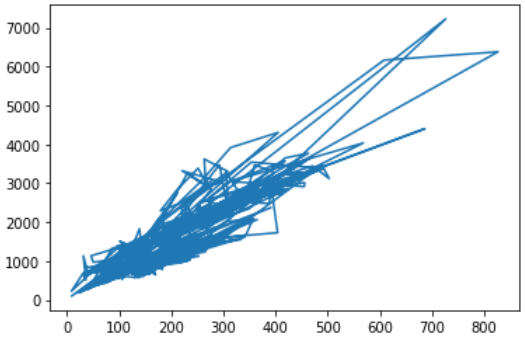
may



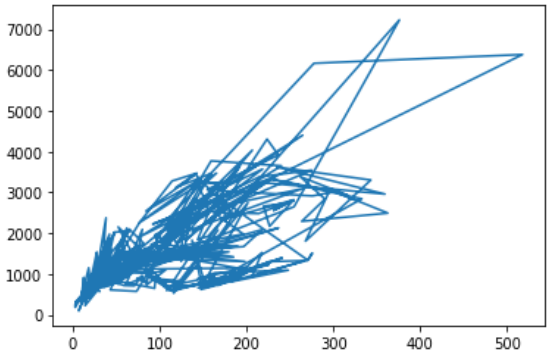
june



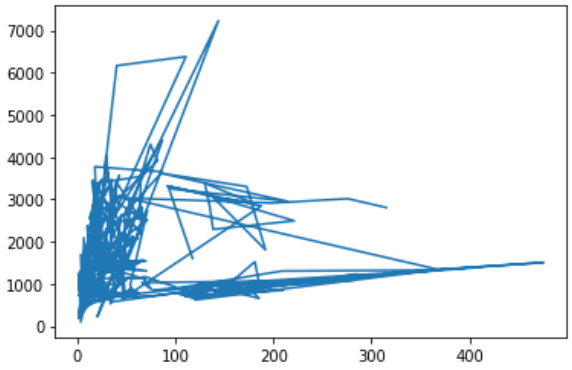
july



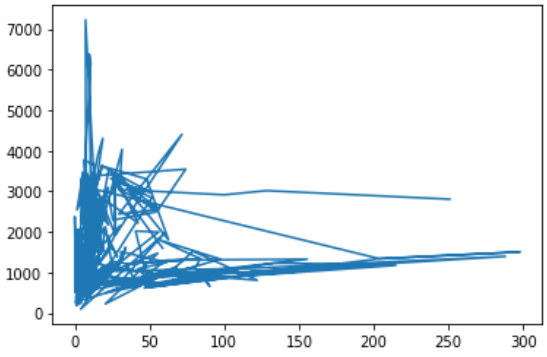
august



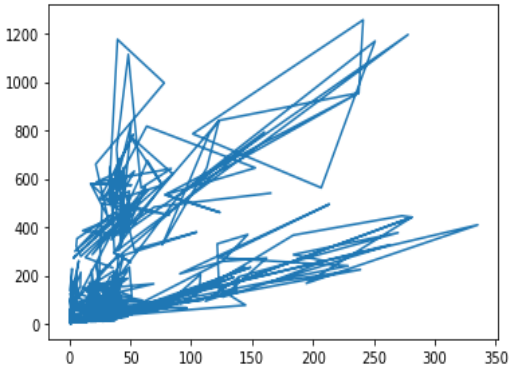
September



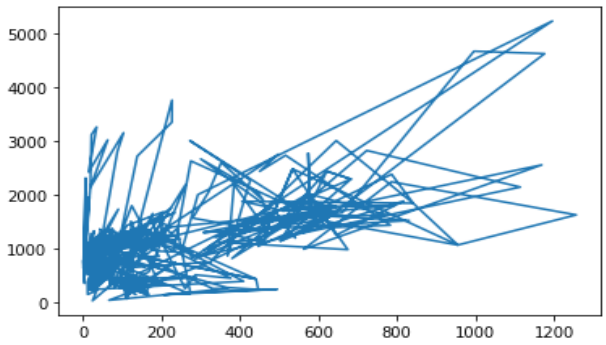
october

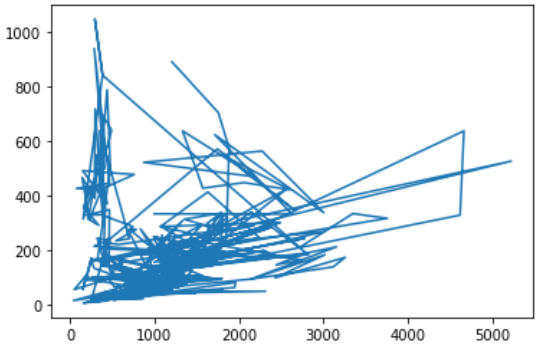


November

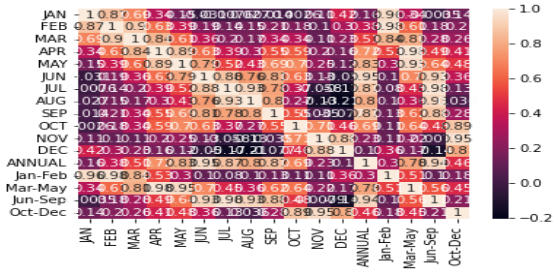


December

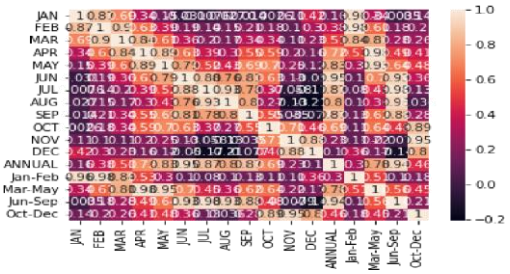




Jan-feb vs mar-may



CO-VARIANCE



CO-RELATION

5. CONCLUSION

• There are some specific problems in the world that pushes the capability of data science and the technology available in this field to their edge among them one is rainfall predicition

• We can easily conclude that for rainfall prediction this is the best way to use it by forming a range of highest and lowest predicted values by adding bias in the model

•Rainfall prediction main objective is prediction of amount of rain in a specific well or division by using various techinques and finding out which one is best

•Future scope of rainfall prediction

The future scope of rainfall prediction is very promising, with advancements in technology and data analysis techniques. Some of the potential developments in this field include:

•Improvements in Data Collection
•Integration of Big Data
•Advances in Cloud Computing
•Development of Early Warning Systems

1. •In summary, the future of rainfall prediction looks bright, and with continued research and innovation, we can expect more accurate and reliable predictions that can help people and communities prepare for extreme weather events.

6. REFERENCES:

http://repository.wit.ie/3326/1/InfomationScience_postprint.pdf

https://www.sciencedirect.com/science/article/pii/S0022030215004932

https://www.kaggle.com/code/darsh79/starter-rainfall-in-india-99bfc809-4

https://www.tandfonline.com/doi/abs/10.4081/ijas.2009.s2.399

https://orbi.uliege.be/handle/2268/224000

https://www.sciencedirect.com/science/article/pii/S0022030221005099

7. EXECUTION:

Githublinks:

K. AMRUTHA NIVAS:

https://github.com/Amruthavarshini75/AIML-BATCH-03/blob/main/aiml_project.ipynb

T. NIKITHA

http://github.com/NikithaThota16/AIML-BATCH-03/blob/main/Aiml_project.ipynb

S.MANISHA

https://github.com/SangaManisha/AIML-B3/blob/main/Aiml_project.ipynb

P.HIMA BINDHU

https://github.com/PadidalaHimabindu/AIML_BATCH-3/blob/main/AIML_Project.ipynb