

Rainfall Prediction using Linear approach & Neural Networks and Crop Recommendation based on Decision Tree

Deepali Patil

Shree L.R. Tiwari College of Engineering
Asst.Professor/Head of Department

Shakib Badarpura

Shree L.R. Tiwari College of Engineering

Abhishek Jain

Shree L.R. Tiwari College of Engineering
U. G Student

Aniket Gupta

Shree L.R. Tiwari College of Engineering
U. G Student

Abstract— Rainfall is one of the most vital components of agriculture and also predicting it is the most challenging task. In general, weather and rainfall are highly non-linear and complex phenomena, which require progressive computer modeling and simulation for their precise prediction. Numerous and diverse machine learning models are used to predict the rainfall which are Multiple Linear Regression, Neural networks, K-means, Naïve Bayes and more. These systems implement one of these applications by extracting, training and testing data sets and finding and predicting the rainfall. This study shows the using of Multiple Linear Regression and Neural networks to predict rainfall and Decision Trees algorithm to recommend crops. Thus, we inferred that we can predict the rainfall and recommend crops with reasonable accuracy.

Keywords— *Neural networks, Linear Regression, Decision Tree, Rainfall, Crop Recommendation, Machine Learning.*

I. INTRODUCTION

The Rainfall Prediction model is implemented by using two Algorithms which are Multiple Linear Regression and Neural Networks. Rainfall Prediction using Linear Regression and Neural Networks is to find the correlation between diverse features in dataset which contributes to Rainfall and to find correct weights and Biases which leads to accurate Prediction of Rainfall respectively. Initially, the dataset with multiple features is cleaned and pre-processed to make it suitable for use and feed it into machine learning algorithm. A correlation matrix is created which shows the correlation between different independent variables and the dependent (Predictor) variable. Once the data is cleaned and processed, we can build our Multiple Linear Regression and Neural Network Model and fit it on our data. The main objective of our system is to predict the Rainfall based on different features like humidity, temperature, pressure etc. with reasonable accuracy. Agriculture and Rainfall are highly correlated with each other. As the technology evolved, developments were made in different sectors including agriculture. Many IT Giants started to provide information related to weather such as temperature, rainfall, humidity, etc. which can be used by farmers in agriculture. different sectors including agriculture. Therefore, we also have developed a Crop Recommendation System which will Recommend Crops to the user based on different inputs provided. This in turn will help oneself

(farmer) to have a better idea about the irrigation and types of crops to be grown.

II. RELATED WORK

[2] A. Kala, Dr. S. Ganesh Vaidyanathan presents an algorithm based on Artificial Neural Network (ANN) such as Feed Forward Neural Network (FFNN) model is built for predicting the rainfall. Artificial neural networks (ANN) are the valuable and alluring soft computing method for prediction. ANN is based on self-adaptive mechanism in which the model learns from historical data capture functional relationships between data and make predictions on current data. The accurate prediction of rainfall is a major criterion for managing the water resources. The prediction accuracy is measured using confusion matrix and RMSE. The results show that the prediction model based on ANN indicates acceptable accuracy.

[7] Swain, S., Patel, P., & Nandi, S. presents a paper which demonstrates, a multiple linear regression model has been developed to reckon annual precipitation over Cuttack district, Odisha, India. The model forecasts precipitation for a year considering annual precipitation data of its three preceding years. The model testing was performed over a century-long dataset of annual precipitation i.e. for 1904-2002. Assuming the intercept or constant of the multiple linear regression model as zero, the equation developed thereby displayed a superb result. The model predictions showed an excellent association with the observed data i.e. the coefficient of determination (R^2) and adjusted R^2 value was obtained to be 0.974 and 0.963 respectively. This reconciliation justifies the application of the developed model over the study area to forecast rainfall, thereby aiding in proper planning and management

[4] Thirumalai, C., Harsha, K. S., Deepak, M. L., & Krishna, K. C. carried on the heuristic prediction of rainfall using machine learning techniques. Agriculture is the major part of our country and economy. A steady rain pattern generally plays an essential role for healthy agriculture but too much rainfall or too little rainfall can be harmful, even it led to devastating of crops. The paper even includes the rate of

rainfall in past years according to various crops seasons and predicts the rainfall for further seasons. The paper also measures the various categories of data through linear regression technique in metrics for efficient understanding of agriculture in India. It consists of a real dataset of past year's rainfall rate based on various seasons. Results of their application help farmers to make a correct decision to harvest a particular crop accordingly to crops seasons.

III. METHODOLOGY

1.) Rainfall Prediction

A. Dataset Used.

The dataset for Rainfall prediction knows as "Austin weather Dataset" was collected from Kaggle. The dataset contains many features which includes temperature, humidity, pressure, dew point, visibility etc. The data was having irregularities and hence were removed in the data preprocessing step. The dataset for crop recommendation was taken from open source GitHub repository. This dataset contains different features like temperature, humidity, rainfall, pH value and the crops that grown under particular values of these features.

A. Data Cleaning and Pre-Processing

Data cleansing is the process of detecting and correcting inaccurate or outlier records from a dataset and then replacing, modifying, or deleting the wrong data which can affect accuracy of our model. In our case, the data has few days where the required factors weren't recorded and the rainfall in centimeters was marked as 'T' if there was trace precipitation.

74	72	70	67	93	71	49	29.97	29.89	29.81	10	10	10	10	14	7	26	0		
76	72	70	64	87	67	46	29.99	29.88	29.76	10	10	5	16	7	27	T	0		
68	75	71	66	91	78	65	29.92	29.81	29.75	10	8	1	14	4	22	0.55	R		
64	70	61	56	97	64	30	29.9	29.85	29.81	10	10	9	13	3	20	0			
66	79	66	62	87	64	41	29.89	29.83	29.74	10	10	10	14	5	21	0			
68	74	72	67	67	79	68	56	29.98	29.82	29.71	10	9	1	24	8	47	1.49	R	
68	-	-	-	-	95	75	55	30.01	29.95	29.87	-	-	-	15	4	22	0		
73	-	-	-	-	95	77	59	29.99	29.93	29.84	-	-	-	15	6	22	0		
74	-	-	-	-	-	-	-	-	-	-	-	-	-	14	9	23	0		
75	-	-	-	-	94	72	49	30.05	29.99	29.93	-	-	-	12	8	24	T		
75	-	-	-	-	69	69	49	30.04	29.99	29.92	10	10	10	12	9	22	0		
77	-	-	-	-	72	67	91	68	44	30.03	29.96	29.88	10	9	4	16	9	0.02	R
76	-	-	-	-	71	66	88	68	47	30.05	30	29.94	10	10	13	6	22	0.01	R
75	-	-	-	-	71	67	88	69	50	30.11	30.04	29.96	10	10	9	13	5	19	0
75	-	-	-	-	72	68	94	72	49	30.07	29.99	29.87	10	10	8	15	6	21	T
76	-	-	-	-	72	68	85	70	55	29.96	29.9	29.82	10	9	8	16	8	22	T
68	75	72	66	88	75	61	30.04	29.99	29.92	10	8	1	16	4	16	0.27	R		
74	74	72	69	94	72	49	30.05	30	29.94	10	9	0	22	4	15	0.24	R		
71	76	72	68	94	77	59	30.07	30.01	29.92	10	8	0	16	5	14	0.49	R		
71	75	71	69	94	76	57	30.02	29.95	29.85	10	10	6	15	7	20	0.01	R		
75	74	72	70	88	69	49	29.93	29.87	29.79	10	10	8	17	8	21	T			
78	74	73	71	85	69	52	29.97	29.88	29.83	10	10	8	16	9	21	T			
77	74	73	70	88	69	49	30.04	29.97	29.91	10	9	7	17	9	27	0			
77	74	71	67	88	64	39	30.03	29.94	29.86	10	9	7	17	8	27	0			

Figure 01. Missing and Wrong Data

Since the algorithm requires numbers, we can't work with alphabets popping up in the data. Hence, we need to clean the data before applying it on our model. This data is not suitable for our model and hence we converted them into such values which can be used in our model and also transformation of which doesn't affect the output.

B. Finding Co-relation and Co-relation coefficient

Correlation can be defined as statistical measure that shows the extent to which two or more variables are dependent on each other. A positive correlation shows that if one variable increases the other also increases while a negative correlation

indicates if one variable increases the other decreases. A correlation coefficient is a statistical measure which can be used to know which features in dataset are more related or dependent on output variable and thus helps in feature selection.

$$r_{xy} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \quad (1)$$

A correlation matrix aids us to identify the features or independent variables which are highly correlated and neglect those which are not correlated thus helping us to decrease the complexity of our model.

Source	X	TempHighF	TempAvgF	TempLowF	DewPointHighF
X	1.00000000	0.183865543	0.191918562	0.190204675	0.194715133
TempHighF	0.183865543	1.000000000	0.970654991	0.881976916	0.74461205
TempAvgF	0.191918562	0.970654991	1.000000000	0.968572677	0.80801434
TempLowF	0.190204675	0.881976916	0.968572677	1.000000000	0.83238423
DewPointHighF	0.194715133	0.744612053	0.80801434	0.832384230	1.00000000
DewPointAvgF	0.204817882	0.755616268	0.83722225	0.877661917	0.96775438
DewPointLowF	0.208228191	0.725670918	0.82042106	0.874380214	0.89802581
HumidityHighPercent	0.078140835	0.123647368	0.19245590	0.252131169	0.51072160
HumidityAvgPercent	0.052896484	-0.116140811	0.02276285	0.165001730	0.40971732
HumidityLowPercent	0.026172299	-0.25255832	-0.09518684	0.074977127	0.26381820
SeaLevelPressureAvgInches	-0.006422022	-0.148704449	-0.13852374	-0.119066340	0.12835087
VisibilityHighMiles	0.039850732	-0.028280491	-0.04312809	-0.042962155	0.25075107
VisibilityAvgMiles	0.084237404	0.226654670	0.14846326	0.064995828	0.07743065
VisibilityLowMiles	0.054801945	0.190854755	0.10072182	0.002420312	-0.13029546
WindHighMPH	-0.059664704	0.012029261	0.01411884	0.015881866	0.11715104
WindAvgMPH	-0.053670793	-0.009296504	0.03426740	0.076016302	0.07498255
WindGustMPH	-0.039783420	0.027675811	0.02666524	0.024318903	0.12576792
PrecipitationSumInches	-0.014928164	-0.069868744	-0.02044246	0.034314885	0.13615923

Figure 02. Correlation Coefficients of Data

C. Normalization (Scaling of Data)

Scaling or Normalization is a method used to normalize the range of independent variables or features of data. Data Normalization is usually performed during the data preprocessing step. Normalizing the data helps the model to be less complex as all the values are converted between a particular range of values. In our case we normalized the data in a range of -1 to 1. The following formula is used for normalization.

$$x = \frac{x - \mu}{\sigma} \quad (2)$$

We Normalized or scaled the data using the formula mention above. We used the R inbuilt function "Scale" to normalize our train data. As of normalizing the test data we cannot normalize it using the scale function as it would consider the Mean and Standard deviation of whole data to scale it. As we cannot give our model any prior knowledge of any kind of data whether it be Mean or S.D, we normalized the test data by separately calculating its mean and S.D and then using the above formula to find its equivalent normalized value.

Note: We normalized the data only for Neural Networks

TempHighF	TempAvgF	TempLowF	DewPointHighF	DewPointAvgF	DewPointLowF	HumidityHighPercent	HumidityAvgPe
0.95723085	1.00889413	1.04080802	0.886144941	0.937065536	1.01799255	0.53030927	0.16882519
1.02331058	1.00889413	1.04080802	0.953794758	0.999917876	1.13456771	0.18486717	0.01250557
1.15547002	1.14693807	1.04080802	0.953794758	1.062770216	1.19285529	0.53030927	0.24698500
1.22154974	1.21596004	1.10892818	0.953794758	1.062770216	1.19285529	0.53030927	0.09066538
1.22154974	1.14693807	1.04080802	0.953794758	0.937065536	0.84312982	0.53030927	-0.06565424
1.02331058	1.07791610	1.04080802	0.953794758	0.811360856	0.60997951	0.27124270	-0.37829348
0.89115113	0.80182822	0.70020719	0.886144941	0.811360856	0.84312982	-0.33324596	-0.22197386
0.82507141	1.00889413	1.10892818	1.089094390	1.125622555	1.25114287	0.53030927	0.48146443
1.08939030	1.14693807	1.17704835	1.021444574	1.125622555	1.25114287	0.53030927	0.40330462
0.49467280	0.59476231	0.70020719	1.021444574	1.062770216	1.07628013	1.04844241	1.26306253
-0.49652303	-0.02643541	0.49584669	0.750845309	0.811360856	0.66826708	1.04844241	1.73202139
-0.03396498	0.24952246	0.49584669	0.209646780	0.371394477	0.60997951	-0.07417939	0.32514481
0.58075252	0.45671837	0.35960636	0.277296596	0.245689797	0.08539130	0.443995374	-0.45645329
0.36251336	0.31867443	0.22336602	0.074347148	-0.068571903	-0.03118385	-0.07417939	-0.45645329
0.16427419	0.11160853	0.01900533	-0.399201565	-0.257128923	-0.03118385	-1.28315672	-1.00357196
-0.03396498	-0.02643541	-0.04911464	0.006697332	-0.005719563	0.14367888	-0.07417939	-0.14381405

Figure 03. Normalized or Scaled Data

D. Building a Model (Multiple Linear Regression and Neural Networks)

i) Multiple Linear Regression

In statistics, linear regression can be defined as linear approach to demonstrate the correlation between a dependent variable and one or more independent variables. In the case if there is only one independent variable, it is called as simple linear regression. If there are more than one independent variable, the process is called multiple linear regression. This term has different meaning from multivariate linear regression, where multiple correlated dependent variables are predicted, rather than a single variable. Linear regression has many applications like prediction, forecasting, or error reduction. A predictive model can be fitted using Linear regression with the collected dataset or observed values. Once the model is fitted, it can be used to make predictions. The multiple linear regression equation is as follows:

Once the model is fitted, it can be used to make predictions. The multiple linear regression equation is as follows:

$$\hat{y} = b_0 + b_1X_1 + b_2X_2 + \dots + b_pX_p \tag{3}$$

where \hat{y} is the predicted or expected value of the dependent variable which is rainfall in our case, $X_1, X_2...X_p$ are different independent or predictor variables like temperature, humidity etc. When all of the independent variables like temperature, humidity... X_p are equal to zero we get b_0 which is value of Y . The estimated regression coefficients are represented as $b_1, b_1...b_p$. The regression coefficient denotes the change in rainfall (\hat{y}) relative to a one-unit change in the independent variables. In our case that is in the multiple regression situation, b_1 can be defined as the change that happens when there is a unit change in X_1 , keeping all other independent variables constant.

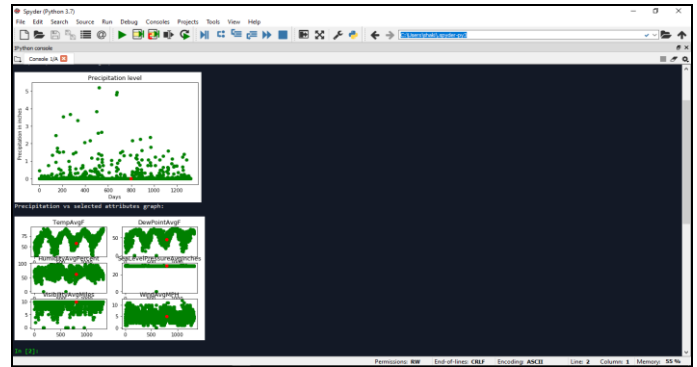


Figure 04. Multiple Linear Regression

In our case the dependent variable is Rainfall (Precipitation) and the independent variables are humidity, temperature, pressure and others. The days are represented on X-axis and scale of features or independent variables is denoted on Y-axis. In the above graph we can observe that the rainfall can be high when the temperature is high.

ii) Neural Networks

Artificial Neural Networks is one of the most popular machine learning and deep learning algorithms. They are inspired by human neurons which are capable of making human like decisions with help of computations. For example, in our case we trained the Neural Networks with different features like humidity, temperature, pressure etc. and they learn to identify and analyze the rainfall based on these features using the results of training dataset. The very simple neural network might contain only one input neuron, one hidden neuron, and one output neuron. It takes several dependent variables = input parameters, multiplies them by their coefficients = weights, and runs them through a ReLU activation function and a unit step function.

$$o_j = f(\sum w_i, j_{ai} + b_i) \tag{4}$$

In our case the input layer will contain the number of neurons equal to the input features.

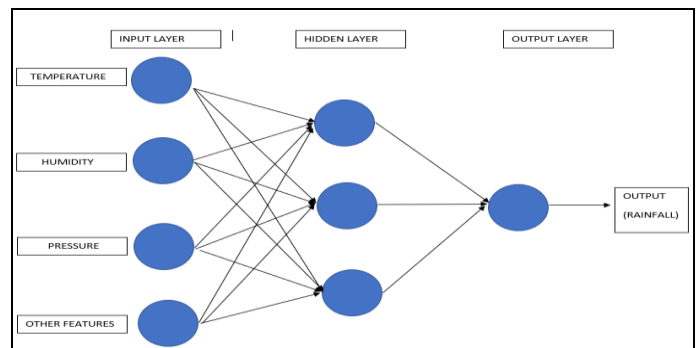


Figure 05. Neural Networks

The inputs will be multiplied with weights and then forwarded to the hidden layer for further computation. An activation function will be used which is discussed further. We have only one output layer as we have to predict only one variable which is Rainfall.

Activation Function (Rectified Linear Unit ReLu)

The ReLU stands for Rectified linear Unit. it is widely used in deep learning. As you can see, the ReLU activation function is half rectified. $f(z)$ is 0 when z is less than zero. It converts all negative values to 0 and $f(z)$ is equal to z when z is positive or 0.

The activation function can be denoted as:

$$R(z) = \text{Max}(0 \text{ to infinity}) \quad (5)$$

Range: [0 to infinity)

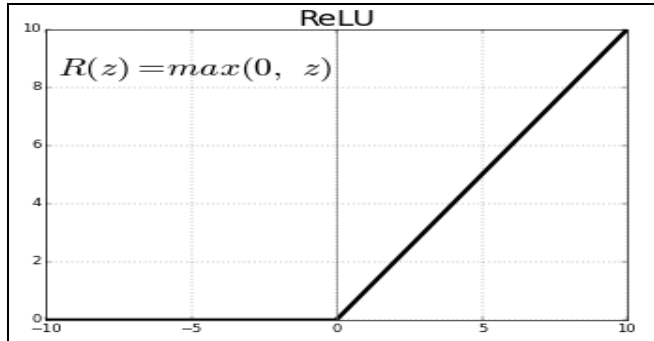


Figure 06. Graph of Activation Function (ReLU)

When a neural network is trained on the training set, it is initialized with a set of weights. Optimization of weights is done during the training period and ideal weights are produced. A weighted sum of inputs is produced by the neuron as mentioned below.

$$Y = \sum (\text{weight} * \text{input}) + \text{bias} \quad (6)$$

let us consider, if the inputs are: x_1, x_2, \dots, x_n

And the weights are represented as: w_1, w_2, \dots, w_n

At the end, the computed value is given to the activation function which is ReLU in our case which prepares the output.

$$\text{Activation function} (x_1w_1 + x_2w_2 + \dots + x_nw_n + \text{bias}) \quad (7)$$

2. Crop Recommendation

A. Decision Tree Regression

Decision Tree is a machine learning algorithm that uses a flowchart-like tree structure or can be like a model consisting of decisions and all of their possible results, including outputs, input costs and utility. It is one type of the supervised learning algorithms. Decision trees can be used for both types of output variable like categorical or continuous.

Decision tree regression algorithm spots features of an object. Decision tree regressor trains a model in a tree like formation and predicts the data for future to have meaningful continuous output. The meaning of continuous output means that it is not denoted by known set of values or numbers.

Discrete output example: A decision tree regressor model which predicts whether there will be rain tomorrow or not.

Continuous output example: A decision tree regressor model which can predict the profit of a company from the sales of a particular kind of product.

The decision trees use the core algorithm named as ID3 which uses a top- down approach. It uses greedy search through the branches of the decision tree with no backtracking available. With the help of information gain (I.G) and standard deviation (S.D), we can use ID3 algorithm to implement Decision tree regressor.

Information Gain

The secondary name for Information gain is Kullback-Leibler divergence which is represented by IG (S, A) for a set S. Information Gain can be easily defined as the effective variation in entropy after deciding on a particular attribute A. With respect to independent variables I.G calculates or measures the relative change in entropy.

$$IG(S, A) = H(S) - H(S, A) \quad (8)$$

Standard Deviation

In a decision tree algorithm, the data is partitioned into smaller subsets which contains instances with similar/homogenous values. It starts building from the root node and then data partition happens. To calculate the homogeneity of the sample we use a measure known as standard deviation. A complete homogenous numerical sample has a S.D = 0.

$$S(T, X) = \sum_{c \in X} P(c) S(c) \quad (9)$$

Standard Deviation Reduction

The decrease in standard deviation (S.D) after a data is split on an attribute is called as standard deviation reduction. The main aim of developing a decision tree model is to find such an attribute which returns the highest reduction in standard deviation (S.D).

Step 1: The S.D of the target is calculated.

Step 2: The dataset is then divided on basis of the different attributes. The S.D for each branch is calculated. Finally, we subtract the resulting S.D from the S.D before the split. The result is the S.D reduction.

$$SDR(T, X) = S(T) - S(T, X) \quad (10)$$

Step 3: The attribute having the largest standard deviation (S.D) reduction is chosen for the decision node.

Step 4: The dataset is partitioned based on the values of the selected attribute. Until all data is processed this process is run recursively on the non-leaf branches.

IV. SOFTWARE DESIGN

A. Python

We used python as our programming language to implement machine learning algorithms. The main reasons for choosing python are mentioned below:

- Numerous Machine Learning Libraries
- Data Visualization libraries and functions
- Software Development
- easy to use for Mathematics and Data Analytics

Features

- Python can work on different platforms like Windows, Mac, Linux, Raspberry Pi, etc.
- It has simple syntax.
- With python and its libraries, we can implement program in fewer lines as compared to other languages.
- Python is an interpreted language.

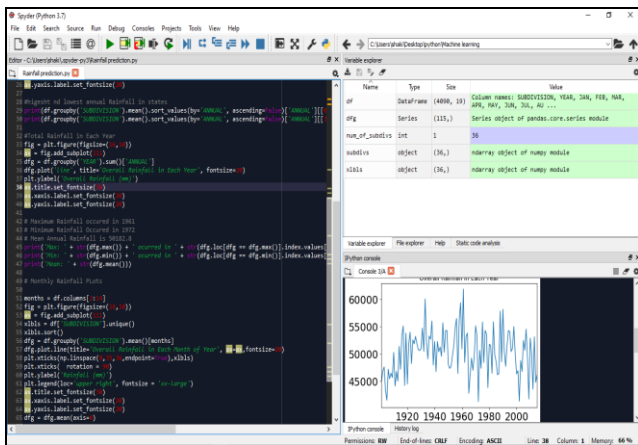


Figure 07. Python IDE (Spyder).

B. TensorFlow

We took help from one of the most used and popular machine learning libraries which is TensorFlow.

TensorFlow architecture works in three parts:

- Preprocessing the data
- Build the model
- Train and estimate the model

TensorFlow library has numerous machine learning and neural machine computation functions which can aid to make complex task easy. Many reasons to use TensorFlow are as follows:

- In neural machine translations, it can help to reduce errors up to 50%.
- It provides flexibility and multi-level abstraction.
- It helps to train the model faster and also aids to run more experiments.

TensorFlow library provides numerous different API to build deep learning architecture like CNN or RNN.

C. Keras

We used Keras for building our neural networks model. It is a high-level neural network API which can be used with other libraries such as TensorFlow or Theano. The main advantages of using Keras for building neural networks are as follows:

- We can use pre-trained models such as VGG16, ResNet, inception networks etc.
- It reduces cognitive Load.
- It has simple and consistent API's.
- It reduces the number of Human Action required for common computations. etc.

D. NumPy and Pandas

For scientific computations and to work with high-performance arrays and matrices we have used NumPy. It is very useful for processing multidimensional arrays and is an open source library. Additionally, we have used Pandas package for data manipulation.

E. Sklearn

Scikit-learn or sklearn is an open source and free machine learning library available in python. With the help of sklearn library we can incorporate diverse classification, regression and clustering algorithms. In our case we tend to use one of its many algorithms named as Decision Tree Regressor.

F. R and RStudio

R is another popular programming language in field of machine learning. We can implement variety of linear/non-linear operations, classification etc. and also neural networks using its inbuilt libraries and packages. We have used R programming to develop a Neural network model to predict the Rainfall. Libraries like Keras and TensorFlow are available in R and hence were used according

We used RStudio which is an IDE for R programming language. One can use RStudio in two formats a desktop application or via a web browser. We have used the desktop version of R studio to perform and build our model.

V. RESULTS

We fed multiple inputs such as temperature, humidity, wind speed into our two Machine learning models viz Multiple Linear Regression Neural Networks and computed the output as shown below

We fed our manual input to the model in the form of an array to get the output from the model

The output for Prediction of Rainfall using Multiple Linear Regression is as below.

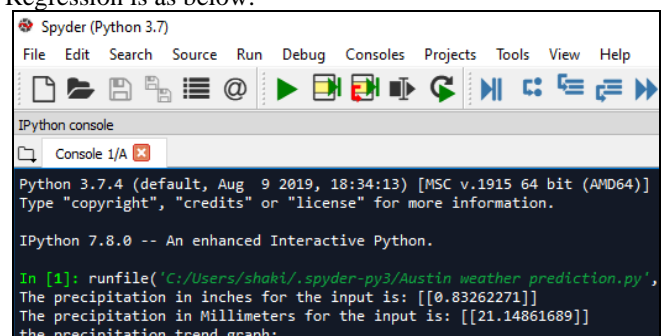


Figure 08. Output of MLR

The output from our Neural Network model shows the mean absolute error which was 0.1459216. Here's a snapshot of it.

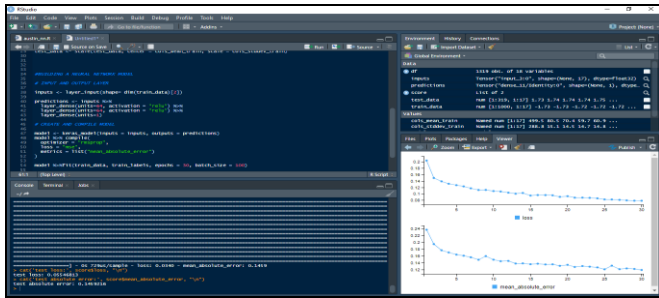


Figure 09. Output of Neural Network

The Accuracy of Decision Tree came out to be 90-92%. The output for Crop Recommendation using Decision Tree Regressor is as shown below:

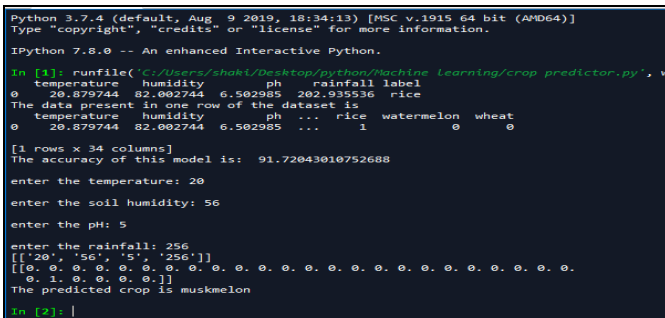


Figure 10. Output of Decision tree

VI. CONCLUSION

Hence, using machine learning techniques like Multiple Linear Regression and Neural Networks we can predict the rainfall with considerable accuracy. However, the accuracy of the model depends on the type of data that has been fed. Modifications needs to be done when data of different structure and type is used to predict the rainfall. Additionally, with the help of multiple input features like Rainfall, Humidity, temperature we also recommended crops that can be grown using another popular machine learning technique known as Decision Tree. The accuracy of the decision tree was quite satisfying and could aid many farmers to make better decisions.

REFERENCES

- [1] Deepti Gupta, Udayan Ghose, A Comparative Study of Classification Algorithms for Forecasting Rainfall, 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions).
- [2] A. Kala, Dr. S. Ganesh Vaidyanathan, Prediction of Rainfall using Artificial Neural Network, Proceedings of the International Conference on Inventive Research in Computing Applications (ICIRCA 2018).
- [3] Inyaem, U. (2018). Construction Model Using Machine Learning Techniques for the Prediction of Rice Produce for Farmers. 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC). doi:10.1109/icivc.2018.8492883
- [3] Thirumalai, C., Harsha, K. S., Deepak, M. L., & Krishna, K. C. (2017). Heuristic prediction of rainfall using machine learning techniques. 2017 International Conference on Trends in Electronics and Informatics (ICEI).
- [4] Abishek, B., Priyatharshini, R., Eswar, M. A., & Deepika, P. (2017). Prediction of effective rainfall and crop water needs using data mining techniques. 2017 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR).
- [5] Suvidha Jambekar ; Shikha Nema ; Zia Saquib 2018. Prediction of Crop Production in India Using Data Mining Techniques. Fourth International Conference on Computing Communication Control and Automation (ICCCBEA)
- [7] Swain, S., Patel, P., & Nandi, S. (2017). A multiple linear regression model for precipitation forecasting over Cuttack district, Odisha, India. 2017 2nd International Conference for Convergence in Technology (I2CT). doi:10.1109/i2ct.2017.8226150
- [8] Kulkarni, N. H., Srinivasan, G. N., Sagar, B. M., & Cauvery, N. K. (2018). Improving Crop Productivity Through A Crop Recommendation System Using Ensembling Technique. 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS).
- [9] B.Vasanthal, R.Tamilkodi2, L.Venkateswara kiran3 rainfall pattern prediction using real time global climate parameters through machine learning 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)
- [10] Nigam, A., Garg, S., Agrawal, A., & Agrawal, P. (2019). Crop Yield Prediction Using Machine Learning Algorithms. 2019 Fifth International Conference on Image Information Processing (ICIIP). doi:10.1109/icqip47207.2019.8985951
- [11] Ahuna, M.N., Afullo, T.J. & Alonge, A.A. 2019. Rain attenuation prediction using artificial neural network for dynamic rain fade mitigation.
- [12] Kishor, R. V., Shatrughan, K. P., Balasaheb, M. K., Sadashiv, M. B., Sachin, V., Gaike, V. V., & Seetamraju, M. (2018). Agromet Expert System for Cotton and Soyabean Crops in Regional Area. 2018 International Conference On Advances in Communication and Computing Technology (ICACCT).
- [13] <https://www.kaggle.com/grubenn/austin-weather> (Dataset Download)