Predicting Rainfall for agriculture in India using Regression

Supervisor : Dr. Rashmi Gupta

Submitted by : Sandeep Dhyani

Table of contents

]

DECLARATION

Plagiarism activity is a serious punishable offense that usually occurs when the resources,work and idea is copied from someone else document. I state That following study is entirely carried out by me with my thoughts,ideas and self work.Researches and resources for this study are cited in Reference section.I have gone through and understood the rules and regulations for the plagiarism.While doing the study i kept this thing into consideration that if any kind of paraphrasing or direct translation occurred then it is also a plagiarism. Hence, i hereby declare that all of the work done in this study is my own work and not copied by someone else work or thesis.

Acknowledgements

Keeping the agriculture sector in mind, i am thank full to the farmers as food is the most important necessity of the human being or any living being in the planet earth. So i would like to express all my sincere thanks to Dr.Rashmi Gupta for her guidance without whom it would not be possible to complete my study.I am also thankful to the the government portals as well for the data set which the have provided in public and are referred where ever needed.

List of appendices

Abstract

In recent years the machine learning has been proven as a powerful tool for predicting the rainfall that could be useful in many sectors. This study will focus on predicting rainfall for the agriculture sector. Indian climatic conditions vary in terms of rainfall, which can be divided and observed on the basis of states. In addition, if rainfall is categorized state-wise; the constant & highest trend can be observed in the state called Meghalaya. While the lowest rainfall can be observed in both of the following states - Leh and Rajasthan. Agriculture is the crucial player in the economy of India, and it is highly dependent on agriculture and forestry, which are affected by rainfall [Krishna Kumar et al., 2004].

Disaster due to heavy rainfall like floods leads to the destruction of crops which affects the farming sectors. If the prediction for rainfall is made by taking monthly and seasonal data of the crop into consideration; then it would be beneficial for the agriculture sector. This study will be applying the regression algorithms by different models, which can help in predicting the rainfall. To achieve such results, this study will be using five various regression models and select the best one among - Multiple linear regression, KNN regression, SVM(Support Vector Machine) regression, DTR(Decision tree regression), RFE(Random forest regression. The aim is to develop a model that can predict the rainfall that will help the agriculture sector, so that rainfall doesn't become a barrier for the agricultural production.

CHAPTER 1

INTRODUCTION

Agriculture is one of the biggest and important sectors in India. India has approximately 142 million hectors of land for agriculture. Out of the total land, about 65 percent of the land is below the level of rain-fed farming[Sahai et al., 2000]. Agriculture sectors have a vital role in the Indian economy as agriculture products are also exported out of India. Although India has a vast area of land for farming and agricultural produce is at its optimum level in fewer states but few factors shed negative effects on farming, especially on the farmers due to many factors, among them rainfall is one of the factors which lead to landslide and flood. Due to this the public and private sectors are unable to generate the profit which they could have made.

Keeping the agriculture sector into the consideration as food sector is one of the most necessities of the human being or any living being on the planet earth, which gives the motivation to study in a particular area. Many farmers are turning their lands into barrens, and some studies show that farmers are leaving agricultural practices for better livelihood, which is indicated towards the losses in agricultural farming due to unpredictable weather conditions. A study by [Chuang, 2019] analyzed that there is Loss in the production of the crops due to rainfall has an impact on the farmer's income too, which is discouraging for many people who are indulged in farming
Furthermore, research by [Murali and Afifi, 2014] provided that the variability in rainfall has an impact on agriculture, food security, and human migration in one of the districts of Janjigir-Champa, located in Chhattisgarh, resulting in out-migration to other areas to find alternative living conditions. Many states in India are facing problems in farming due to erratic rainfall like Bihar in terms of agriculture contributes around approximately 19 percent to gross domestic product and about 70 percent of labor work in rural areas.The majority of the district in Bihar has faced a decline in rainfall. Maharashtra is one of the prominent agricultural states in India. Many crops are planted in Maharashtra like jowar(Sorghum), bajra(Pearl millet), rice, wheat, urad(Black gram), and many other pulses. As being one of the major crops growing state it has been observed that drought is the chiefly responsible for the suicides due to low income and farmer in debt, in addition to this a study by [Udmale et al., 2014] based on data survey by 223 farm households found that suicidal

cases as a result of low income and also indebtedness occur in Maharashtra due to the impact of drought.It was noticed that due to drought there was a downward trend in horticultural crops, cereal yields, and job losses associated with low farmer income. Where in the northern part of the Karnataka due to incessant rain there was the damage of the crop. The district called Dharwad had excess rainfall than usual from 33MM to 77MM in the month of August. Almost around 12,857ha of the area was affected due to the erratic rainfall. Moreover, it has been observed that in some northern parts of India farmers have lost one-third of the crops due to the non-seasonal rainfall and hailstorm and may face more risk due to that.

Predicting rainfall is a daunting task as the data from the rainfall is a time series and it varies from time to time and the rainfall is inconsistent and not reliable in nature[Singh et al., 2014], as the climate season changes[Hasan et al., 2015]. Rainfall prediction has attracted not only to private sectors but also to a government organization.The reliable prediction can help the Agriculture sectors in decision making and avoiding the loss as they make a decision on what kind of farming can be done according to the rainfall that may lead to profit in agriculture business as farming is majorly dependant on the rainfall. In India, there are three major seasons for cropping that are Karif, Rabi, and Zaid. The Kharif season starts from July till October, and then after there is Rabi season that starts from October until March and the last one is Zaid from March to June. Much new research shows that in India due to Rainfall there is a high impact on the forest and also Climate has an effect on tourism. There are many cases of natural disasters due to heavy rainfall that has affected farming lands related to rainfall like a landslide, flood, and soil erosion. So natural disasters, like flood and landslides, can not be stopped but if we predict the rainfall then we can take some measures that can be help full for this natural phenomena[Parmar et al., 2017].Hence having the prior awareness and knowledge of rainfall will help Indian farmers and policymakers to reduce harm to the crops and human hardships during adverse rainfall in support of this a study by [Sahai et al., 2000] used the ANN model with an error-back-propagation algorithm to forecast ISMR on seasonal and monthly time scales. So having adequate earlier knowledge about rainfall behavior will help Indian farmers minimize crop harm. To tackle this kind of natural disaster we can predict the rainfall by using the appropriate model that can do a prediction for this to achieve we need to apply a machine learning method. In recent years, there are numerous models that have been implied by the researchers so that the accuracy of the rainfall prediction is increased.

There are various research that has been done and various numerical weather forecasts have been introduced for predicting the weather but they have some limitations[Ramana et al., 2013].

Focus of this research is to use the machine learning regression models that can predict rainfall in the most effective way on the given data. Regression is the statistical empirical technique that is vastly used in many areas like in business, behavioral sciences, and climatic prediction[Kannan et al., 2010]. Regression is supervised learning, it is a statistical approach that finds the relation between the variables and is used for predicting the outcome based on the relationship between the variable that is acquired by the data. There are five machine learning regression models that will be used named Decision tree regression, KNN regression, Multiple linear regression,RFE(Random forest regression), and Support Vector Machine regression and compare which models perform well. Hence after applying the models, we can see the measurement of the rainfall in MM(millimeter), and according to the need for crops, the prediction can be used. In the current dataset, we do not have features for the season of the crop that are Kharif, Rabi, and Zaib, so more features will also be added according to the season of the crops which will be more help-full for the agriculture sectors to understand more about rainfall.

1.1. SIGNIFICANCE OF THE STUDY

Agriculture is the core of India and is highly dependent on the rainfall [Krishna Kumar et al., 2004]to make an adequate profit, the farmers need to have sufficient knowledge about factors which are impacting the production of the crops, among them rainfall is the key factor. To create such progressive scenarios we need to implement conventional or machine learning models that are important to achieve the results. Research by[Rosenzweig and Udry, 2014] analyzed the effects of the prediction and rainfall on planting- and harvest-stage agricultural wage rates. that Indian Meteorological Department prediction help farmers to gain higher profit in long-range Kharif-season. The data for six villages from the period 2005 to 2011 was used. So the prediction enhanced the average profit done by the farmers and can reduce the migration of the labor from the village, due to the losses in agriculture.

If someone has agriculture business then it will be dependant on the number of production and the amount of profit it will make. The farmers produce the crops in the three major seasons that is Karif, Rabi, and Zaid, Many farmers in India believe in doing the traditional farming that is taking the raw idea of the climate which is riskier and which results in a loss in the production of the crop. Many of the farmers migrate from one village to another because of loss in the farming production due to the appropriate farming methods and raw knowledge of the rainfall condition.The current study aims to analyze and validate that model of machine learning among the suggested algorithms that can do the prediction of rainfall for the subject area. So if they will be having the proper knowledge about the rainfall about how much measure rainfall there will be and when there will be rainfall, so it could be living a saver for the farmers.

Therefore with the help of the rainfall prediction they can plant the crops like for example crops like peas, lentils and fava beans don't need more water to grow and the crops like maize or sugarcane and banana need much water, hence if the farmer will have the knowledge of the rainfall water, then they can plant the crops according to that. so the current study on predicting the rainfall by using machine learning and using the best model. This will be helping the farmers in planting the traditional crop according to the rainfall prediction.

1.2. AIM

This study aims to develop and train the machine learning model and then verify the outcomes of it, which can predict the rainfall keeping agriculture sector in mind that it can help the farmer and according to the prediction they can take the future measures to avoid loss in the production in the agriculture sector and also natural disasters. The study will also compare the various regression models and the values of the R-squared score and RMSE error percentage. This prediction can also be helpful to many sectors like forest plantation and tourism. Hence, the purpose is to identify the best-suited regression model among all five, that can effectively predict the rainfall through previous data training.

1.3. OBJECTIVES

Below are main objectives to achieve our defined aim.

- Determine the goal of the project.
- Analysing the measurement of the rainfall monthly,annually,seasonally and according the the season of the crops from year 1901 to 2015.
- Preparation of a data-set from the extracted data.
- Train,test and then analyse the results of applied models under consideration(machine learning)
- Creating a well integrated system which can predict the rainfall.

1.4. HYPOTHESIS

When applied with the best machine learning algorithm(regression models), rainfall prediction can help the agriculture sector to make policy according to that and also plantation of the crops. The prediction can also help the farmer to take necessary measures for the future aspect. Thus in the case study, we will be applying the following regression models below to achieve the results and then comparing the outcomes, and selecting the best model among below.

- Multiple linear regression
- DTR(Decision tree regression)
- KNN regression
- SVM(Suport Vector Machine) regression
- RFE(Random forest regression)

After applying the models in the data, after that comparison will be done and afterward, it is to testify that; selected model can do prediction of rainfall with or above 70 percent of correctness that is R-squared score and RMSE(Root Mean Square Error) less than 20 percent of compared to the target variable's mean value.

Hence , its a one tailed test:

Null Hypothesis : RMSE is less than 20% and R2 score is greater than 70%
Alternative hypothesis : both, or one of the condition(s) in null hypothesis do not satisfied and conventional model perform(s) better.

1.5. RESEARCH QUESTIONS

Which (regression) machine-learning model among the five models under consideration, can be applied that can predict the rainfall ?

In this study, five models are used for the prediction of rainfall. The focus of this study is to use such a model while considering the traditional model in mind and predict the rainfall considering the

agriculture sector that can be helpful for taking the future measures to avoid loss in the agriculture sector and the natural disasters that affect the farming.

The research questions are further divided as:

- Which among the input variables or features influence the target variable product?
- What are the season of the cropping in India?
- What are the findings of previous machine learning studies?
- What parameters are required to form a data set for solving the described problem?
- What machine learning model to be used for prediction maximum precision?
- What results are observed after training and testing the dataset, and which machine learning model performs at its best?

CHAPTER 2

LITERATURE REVIEW AND BACKGROUND

In India agriculture workforce has the majority of the people with around 58 percent [Singh, 1999] and it contributes around 17 percent in Indian GDP[Deshpande, 2017].In the past, there are many types of research conducted to analyze rainfall and their effects on agriculture by various research groups. There has been much research that has been done for predicting the rainfall. While using well-trained machine learning algorithm, and increasing the efficiency to predict rainfall has shown an upward trend. Having prior information about the rainfall can be useful. A study by [Sahai et al., 2000], which used the ANN model with an error-back-propagation algorithm for prediction of ISMR on seasonal and monthly time scales shows that having sufficient prior information about the rainfall behavior can help the Indian farmer to reduce the loss in crop production. Five years of data were taken for predicting the rainfall for the coming year. The study depicted that monthly rainfall can be predicted during the monsoon season, which shows the impact of single-season on annual rainfall(monthly).In support, a study by [Kumar and Parikh, 2001] used the cross selection as evidence to evaluate the link between the farm level or agricultural gross-revenue and climate variables and estimated the functional relationship between them by using linear, quadratic, and interaction variables to understand the sensitivity of the climate. The study shows how climate changes affect farming production in India.

While climate is an important factor, another study by[Krishna Kumar et al., 2004] analyze the relationship of the crop–climate for India by using the historical statistics of the major crops that include (wheat, groundnut, sugarcane, and rice) and the aggregate production of the food grains, cereals, pulses, and seed oil. The outcome depicted that response to moon-soon rainfall is predictable even before the start of the growing season.

A study by [Zaw and Naing, 2008] implemented a model for prediction of rainfall using the empirical statistical methodology, Multi variables polynomial regression (MPR). Global climate data for 37 years from 1970 to 2006 was used. The study depicted that the estimated amount of rainfall was similar to the actual value, as a result of many studies.

Another research was done for prediction of monsoon rainfall by [Awan and Maqbool, 2010] using the Artificial Neural Networks for predicting the monsoon rainfall which included Back-propagation(BP) and Learning Vector Quantization (LVQ), for this study data of monsoon rainfall from 1960 to the 2004 year was used to train the neural network model and analyze the efficiency of the models over next five years test period from 2005 to 2009. The result depicted that Neural Network techniques were over the existing Statistical Down-scaling technique was good for accuracy. It was also found that by using the historical data on monsoon rainfall the proposed technique also overcome the dependency on the numbers of parameters.

In many studies, the widely used empirical approach is Regression and Artificial Neural Network(ANN) and has been a popular algorithm among the researchers. It has shown pretty good results for predicting the rainfall. In addition to it, there is research by[Gupta et al., 2013]depicted that the prediction of Indian rainfall through the study of the neural network model to determine ANN applicability by using the back-propagation algorithm and supervised learning. That shows that the Designed back-propagation neural network is good for predicting the rainfall due to prediction values are closer to actual values.

One more study by [Chifurira and Chikobvu, 2014] aimed to construct the predictive model of rainfall using the climatic factors like SOI(Southern Oscillation Index) and Darwin SLP(Sea level Pressure) for Zimbabwe, minimum a year in advance. In the study, it was found that combining regression with time series analysis provides a power full method for prediction of rainfall annually with the help of values Darwin SLP and SOI.

A Study by [Udmale et al., 2014] based on a survey from data through 223 farm households it was observed that in Maharashtra due to the impact of the drought there are suicidal cases due to low income and also the indebtedness. The result depicted that due to drought there was down Trend in the horticultural crops, yields of cereals and job losses that were linked to low farmer income.

In addition, the research by [Murali and Afifi, 2014] provided the rainfall variability has an impact on the agriculture, food security and migration of human in one of the districts that is Janjigir-Champa which is located in Chhattisgarh which is resulting in migrating of people to another area as alternative live-hood.
In addition, a study by [Singh et al., 2014] did study on bihar agriculture and the impact of rainfall in the farming as bihar contribute 19 percent to the state domestic product and approximately 77 % districts experienced downward trend in rainfall by 5 to 25 percent, it was noticed that the effect of the rainfall in the production of the rice also Varying rainfall from the time June to September adversely affects the state rice production.

Moreover, research by[Rosenzweig and Udry, 2014] shown that Indian Meteorological Department prediction helps farmers to gain higher profit in long-range Kharif-season. So the prediction enhanced the average profit done by the farmers and can reduce the migration of the labor due to the loss from the village.

Using regression there was research by [Sethi and Garg, 2014] employed the MLP(Multiple Linear Regression) for early prediction of the rainfall. Data of 30 years from 1973 to 2002 was used, the data-set comprises of weather data like precipitation vapor pressure, the mean temperature and the cloud cover of the city UDAIPUR that is situated in Rajasthan northwestern side of India. The result depicted the predicted values were close to the actual values.

Research by [Hasan et al., 2015] used Support Vector Regression for prediction of the rainfall for Bangladesh. The data that was gathered was from 2008 to 2014 of Chittagong situated in Bangladesh via the Meteorological department. The data was raw and it was prepossessed and refined manually to fit into the algorithm. The study depicted that the proposed model predicted better than any regular technique used.

There was research done by [Swain et al., 2017] which developed the multiple linear regression model to calculate the rainfall annually in Cuttack district that is situated Odisha, India. Previous three-year annual rainfall values were used and the result depicted that the model was able to

generate a quite good result and delivered matching data with the actual one.

Also, research was done by [Prabakaran et al., 2017] to predict rainfall using the modified linear regression in several states in India. The method depicted that the model for predicting the rainfall that had a lower percent of error compared to the other data mining methods like clustering, backpropagation that provides the generalized values instead of estimates..

Supporting study by [Mohapatra et al., 2017] investigates the data mining technique by using the regression model for data of wet days frequency, precipitation, and rainfall of Bangalore city situated in India of 100 years from 1902 to 2002. The regression model was made to train and validate by considering the actual rainfall data of that particular area, which was used to predict the rainfall for future years. The model depicted sufficient accuracy for the rainfall for the three seasons.

Also, research by [Ahmed et al., 2013] predicted the rainfall and also its relationship with several atmospheric variables by using the traditional approach. The multiple linear regression was applied on six years of data of Coonor, from Nilgris district in Tamil Nadu(India). The study revealed that the model performs well with an accurate result.

Another research by [Cramer et al., 2017]applied and compared predictive performance of Seven models that were named as M5 Rules, support Vector Regression, Genetic Programming, Radial Basis Neural Networks, KNN(k-Nearest Neighbours), state-of-the-art (Markov chain extended and M5 Model trees. with rainfall prediction).The detailed analysis shows that machine learning surpasses current state-of-the-art. The research gives ample evidence that cumulative quantities of rainfall ability to predict are more than daily amount.

In one study by [Geetha and Nasira, 2014] used data mining techniques and predicted weather event such as fog, rainfall, cyclones, and thunderstorms by employing the decision tree model. Only relevant attributes of two-year data of 2013 and 2014 were used for the model, the result of the study depicted that the decision trees was an effective technique for predicting the rainfall.

Also, a study by [Parmar et al., 2017]predicted rainfall and aimed to give non-expert easy access about the prediction techniques and the approaches that are used. In research, several models were used for rainfall prediction and the findings depicted that Artificial Neural Network makes its most favored approach due to nonlinear relationships in rainfall data and its ability to grasp from past.

By studying the previous research done by various scholar above many techniques and ideas can be grasped which can help to learn more about solving the issues which are intended to achieve. Hence by using the machine learning algorithms the prediction can be more efficient for achieving the goal and there are ways to predict the rainfall. Taking a step forward I am aiming to use the regression technique on the data-test above numerical values. As the values in the data-set are numerical so it is suited for the regression.

Studies and their findings				
Author	Sample	Title	Source	Findings
Deshpande, 2017	National analysis of economic and agricultural data	Role of agriculture in Indian economy	PRS Legislative Research	In India agriculture employs around 58 percent workforce, and it contributes approx 17 percent in Indian GDP.
Sahai et al., 2000	Seasonal and monthly time scales on rainfall data	All India summer monsoon rainfall prediction using an artificial neural network	Climate dynamics Journal	Efficient rainfall predictions can help farmers reduce their losses.
Kumar and Parikh, 2001	Agricultural and climatic data	Indian agriculture and climate sensitivity	IEEE	How the climate changes effects the farming production in India
Krishna Kumar et al., 2004	Climate and crops data for food grains like wheat and rice	Climate impacts on Indian agriculture	International Journal of Climatology: A Journal of the Royal Meteorological Society	In extension to the study by Sahai et al., 2000, this study shows that the rainfall for monsoon season can predicted even before the start of previous harvesting season.
Zaw and Naing, 2008	37 years of rainfall data	Empirical statistical modeling of rainfall prediction over Myanmar	World Academy of Science, Engineering and Technology	Close prediction of the rainfall as per to test values

Studies and their findings				
Author	Sample	Title	Source	Findings
Awan and Maqbool, 2010	Rainfall data from year 1960 to 2004	Application of artificial neural networks for monsoon rainfall prediction	The Institute of Electrical and Electronics Engineers (IEEE)	Neural networks performed better than conventional methods and predicted rainfall with much accuracy
Gupta et al., 2013	Analysis data of rainfall data	Time series analysis of forecasting Indian rainfall	International Journal of Inventive Engineering and Sciences (IJIES)	Artificial neural networks performed efficiently to predict rainfall over 90% of accuracy
Chifurira and Chikobvu, 2014	Analysis of Rainfall and climatic variables	A weighted multiple regression model to predict rainfall patterns: Principal component analysis approach	Mediterranean Journal of Social Sciences	With the help of Darwin SLP and SOI, regression model can predict the rainfall very efficiently.
Udmale et al., 2014	Survey data of 233 household	Farmers[U+05F3] perception of drought impacts, local adaptation and administrative mitigation measures in Maharashtra State, India	International Journal of Disaster Risk Reduction	Study depicted the down trend in production of horticulture crops.
Murali and Afifi, 2014	Analysis of rainfall and crop production data	Rainfall variability, food security and human mobility in the Janjgir-Champa district of Chhattisgarh state, India	Climate and Development	Direct impacts of the rainfall variability on the agriculture
Rosenzweig and Udry, 2014	Analysis of rainfall data	Rainfall forecasts, weather, and wages over the agricultural production cycle	American Economic Review	Prediction of rainfall helped farmers to gain profits in crop production in kharif season

Studies and their findings				
Author	Sample	Title	Source	Findings
Sethi and Garg, 2014	Analysis of 30 years rainfall data from year 1973 to 2000	Exploiting data mining technique for rainfall prediction	International Journal of Computer Science and Information Technologies	Close prediction to the actual values of rainfall.
Swain et al., 2017	Analysis of rainfall data in south India	A multiple linear regression model for precipitation forecasting over Cuttack district, Odisha, India	IEEE	Close prediction to the actual values of rainfall.
Mohapatra et al., 2017	Rainfall data of 100 years from year 1902 to 2002	Rainfall prediction based on 100 years of meteorological data	IEEE	Efficient seasonal predictions for rainfall.
Ahmed et al., 2013	Analysis of rainfall and atmospheric variables	Rainfall Prediction Using Multiple Regression Technique	International Journal of Applied Engineering Research	Close prediction to the actual values of rainfall.
Cramer et al., 2017	Rainfall and climatic variables	An extensive evaluation of seven machine learning methods for rainfall prediction in weather derivatives	Expert Systems with Applications	Findings shown that machine learning models outperform the traditional methods for prediction of rainfall.

Studies and their findings				
Author	Sample	Title	Source	Findings
Geetha and Nasira, 2014	Two years of rainfall data	Data mining for meteorological applications: Decision trees for modeling rainfall prediction	2014 IEEE International Conference on Computational Intelligence and Computing Research	Findings shown that decision trees performed best with the prediction of rainfall based on small dataset
Parmar et al., 2017	Analysis of rainfall data	Machine learning techniques for rainfall prediction: A review	International Conference on Innovations in information Embedded and Communication Systems	Finding shown that artificial neural networks are best favored approach for predicting rainfall.

Several studies has been done in terms of predicting the rainfall, as given in above literature review. Although, it has been proved in most of the researches that machine learning models are better than conventional and traditional models, in terms of predicting rainfall.This leads to motivate us to drill down more into the studies done previously and produce some concrete result out of these clustered and resourceful studies. We have seen that study by [Sahai et al., 2000] depicted that annual rainfall can be depicted on the basis of monsoon season only, which is supported in contrast by [Krishna Kumar et al., 2004] with study showing that monsoon season can be predicted even before the cropping season.

In addition to above studies, [Zaw and Naing, 2008] has illustrated that the machine learning model perform with satisfying efficiency on the basis of 37 years of data, this leads us to investigate for more amount of data to be able to predict more accurately.Moreover, a study by [Awan and Maqbool, 2010] shows that artificial neural networks perform best on predicting rainfall.Overall, our aim is to predict the rainfall based on 115 years of data, which is addition to the

study done by [Sahai et al., 2000], which can thus reduce losses of farmers in India.

2.1. PROBLEM STATEMENT

The key problem in predicting the rainfall is that the time series data keeps changing due to changes in climate, and geography of the country [Hasan et al., 2015]. The focus of this study is to apply the best regression models to predict the rainfall and then compare all of the models under consideration. The models under study are Decision tree regression, KNN regression, Multiple linear regression, RFE(Random forest regression), and Support Vector Machine regression.

CHAPTER 3

RESEARCH METHODOLOGY

3.1. RESEARCH DESIGN

3.1.1. Work Flow

The work flow is designed for the completion of research are:

- Determine the research project's aim.
- Study of literature and analysis of current model data.
- Extraction of data from data source and identifying and analyse the most impacting features and target variable.
- Preparation of data and further cleaning of the data from extracted data for training models for machine learning purposes.
- Implementation of the most successful current model.
- Creating, optimizing and cross validating the algorithms of the new model.
- Studying the current model's set backs and future implementations.

The following steps needs to be followed for the rainfall prediction for this study:

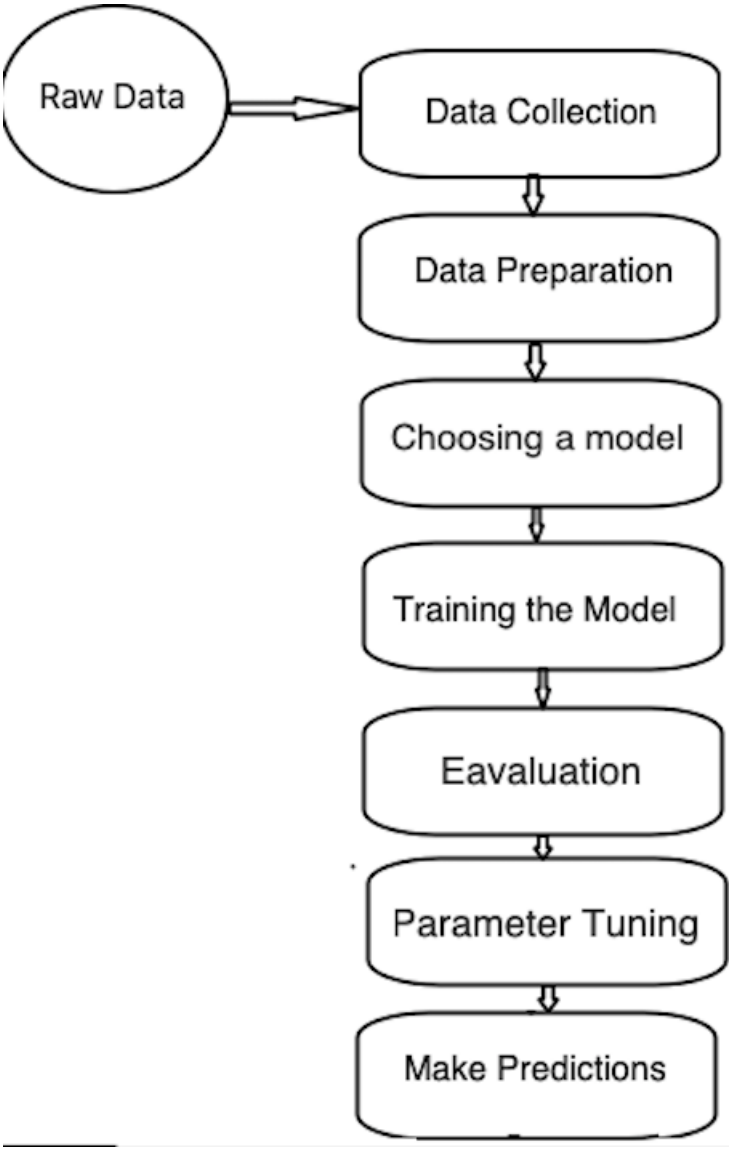

Figure 3.1: phases for study

Source of image : paintx

First comes data collection as it is the most important step as the model needs some dataset. The data is gathered from the source, data can be in any format in xl format, Jason format, or in the CSV format. Therefore Getting the correct data that can be used for the task that has to be achieved has to be done. After the data is downloaded then it needs to be checked whether it needs to be clean or not then according to need then it is processed for further step.

The next step is the preparation of the data, in this phase the aim is to identify and reduce any potential biases in our data. It is the next challenging task as the data which is gathered can be in any format and can be noisy or may carry missing values as the noisy data may reduce the efficiency of the model, therefore data needs to be clean and then filling up of the missing values and then unnecessary data need to be removed. After that, the best data after all this process is to be selected and stored which is needed for further analyses

Then comes the third phase in which the models are selected after the data is cleaned and finalize, then the best data-set is selected and Further choosing the machine learning algorithm which has to be applied according to the need. The algorithms can be selected according to the basis of data-set as data can be numerical or it can be categorical. There are various algorithms for different tasks it may be a classification or the regression, so the model needs to be selected according to the need.

Further, the training part is a crucial part of the machine learning process. In this part learning in "bulk" is done and machines learn from the given data. In this phase, the data is split-ted into train and test by 30 percent and 30 percent and 70 percent respectively, it requires patience and experimentation are done in this phase as machine need to learn from data before the prediction is done. This phase can be rewarding if the model succeeds in the task for which it was built for.

After the model is trained from the given data then the efficiency of the model needs to be checked. In evaluation step, the performance of the applied models need to compare and the result of the various algorithm is analyzed whether the models can achieve the task or not, further RMSE and the R score is compared from the previous one.

Once the evaluation has been done, then it needs to be checked if any further improvement is needed to be done, like if during the evaluation phase the model did not obtain a proper result or if there is over-fitting or under-fitting problems, so we must return to the training step. So parameter tuning is the next step right after the Evaluation this step is known as hyper-meter tuning this

is one of the necessary steps.The original set parameters have to be tested whether the result can be better and improve the model if there are any short-comes. This phase is also known as an experimental process.

Once all the steps have been done then we get an answer to the question i.e to do the task for which the model was built do the prediction. It is the last and is one of the important steps where the model is ready for the practical application. The model that has been trained from the data is now ready to make a conclusion. A good well-executed model and efficient model can improve the decision-making process for the user.

3.2. METHODOLOGY

The study comprises the following three phases. Firstly to define the aim and after than studying the earlier work and to extract data from verified sources and then in the next step, we go to the second phase that is preparing the data and annotation to train the machine learning models yet to be done. The next of the two phases are made to identify the most efficient algorithm/model for performance prediction of the target variable and validating the output of the model and therefore tuning the further precision.

- **Data extraction and preparation**

  Extracting the data-set from the sources and then further processing it for the analysis. The data which is downloaded need to analyze as the data have null values and kind of unnecessary values that are not needed, so cleaning of the data and then re-sample it before performing the machine learning.

- **Analysing features and preparation of seasonal rainfall data**

  Analyzing the features of the data according to the crops need. The data includes the rainfall monthly, seasonally, and annually.

- **Applying and cross-validation of machine learning model**

  The data-set has most numerical variables including the target variable and only one variable that is categorical, so we will be implementing regression for the prediction as the data comprises most quantitative variables. Various regression models will be applied and then

the accuracy will be compared among all the five models and then the most efficient model will be selected for the prediction.

3.2.1. Models under consideration

- **Multiple linear regression** It is also known as multiple regression, it is a common form of linear regression. It is a statistical technique which uses a various independent variable for predicting the result of the response variable. It is an extended version of linear (ordinary least squares) regression which consists of a single independent variable for modeling.
  the formula for Multiple linear regression.

$$Y = mx1 + mx2 + mx3 + b$$

- **DTR(Decision Tree Tegression)**

  Decision tree regression is supervised learning, which constructs regression or classification models as a tree-type structure. Decision tree regression breaks the data set in a smaller subset while in parallel incrementally creating a related decision tree that ends with the node's decision and nodes for leaves. It has nodes and leaves: The nodes describing the condition.Depending on the condition that is either yes(True) or no(false) child of the node.
  The outcome of the algorithm is represented by end nodes or leaves.

- **KNN Regression**
  KNN K-Nearest Neighbors that is used in machine learning and is a simple algorithm yet very powerful in doing a certain task, its a nonparametric supervised learning algorithm. In the KNN(K-Nearest Neighbors) it uses data and then classify according to that new data point similarity measures.KNN regression can be used for the classification as well as in regression problems. It is very simple to implement and it doesn't require training before making the prediction, so any new data can be added that will not have an impact on the accuracy of the model.

- **SVM(Suport Vector Machine)**

  A Support Vector Machine is a supervised learning algorithm. It can perform classification, regression, and also the outlier detection. It identifies a hyper-plane(Hyperplanes are decision boundaries) which then categorizes data points. The vectors(cases) which describe hyperplane are called the support vectors. The support vectors are the data points that are closer to the hyperplane and hence affect the hyperplane's position and its orientation.

- **RFR(Random forest Regression)**

  Random forest regression is one of the important machine learning algorithms.It is an ensemble learning method used for classification, regression. Additional randomness is added in the Random forest for the model when the trees are building.

CHAPTER 4

IMPLEMENTATIONS AND RESULTS

4.1. RESULT AND EVALUATION

Many models are used for the rainfall prediction by many researchers. Mostly used models for rainfall prediction are regression, Artificial Neural Network, Decision Tree algorithm. For this research, I will be focusing on the case study based on data collection. It will include active participation from start to end phase of model training to the evaluation of the accurate prediction of rainfall.

Our main focus is on implementing Regression techniques to learn from data as the data is more numerical so applying the various regression models. Regression is supervised learning, it is a statistical approach that finds the relation between the variables and is used for predicting the outcome based on the relationship between the variable that is acquired by the data. I will be applying several regression models like decision tree regression, KNN(k-nearest neighbors algorithm) regression, Multiple linear regression, RFE(Randomforest regression) and SVM(Support Vector Machine regression).

4.1.1. DATA PREPARATION

4.1.2. Data Extraction and Resources used:

The study wouldn't be completed without using resources to extract in CSV format and after that transforming the data into a meaningful and single data-set.

Data extraction and understanding: The aim was to apply and compare the five regression model efficiency on the data. The subject area was selected on the bases of problems that were faced by agriculture sectors due to effect of rainfall in the agriculture business and an attempt to provide the solution through machine learning algorithms which could help the farmers to do better with the help of the current study and increase the production of the crops.
The data which is processed is in the form of CSV file from 1901 to 2015, the features that are included in this data-set are subdivision, month, annual rainfall and season rainfall of all Indian

states, so understanding the data features and analyze it by considering agriculture aspect. The data for rainfall is as shown in the below screen captured figure.

```
new_one.head(10)
```

	subdivision	year	jan	feb	mar	apr	may	jun	jul	aug	sep	oct	nov	dec	annual	jan_feb	mar_may	jun_sep	oct_dec
0	ANDAMAN & NICOBAR ISLANDS	1901	49.2	87.1	29.2	2.3	528.8	517.5	365.1	481.1	332.6	388.5	558.2	33.6	3373.2	136.3	560.3	1696.3	980.3
1	ANDAMAN & NICOBAR ISLANDS	1902	0.0	159.8	12.2	0.0	446.1	537.1	228.9	753.7	666.2	197.2	359.0	160.5	3520.7	159.8	458.3	2185.9	716.7
2	ANDAMAN & NICOBAR ISLANDS	1903	12.7	144.0	0.0	1.0	235.1	479.9	728.4	326.7	339.0	181.2	284.4	225.0	2957.4	156.7	236.1	1874.0	690.6
3	ANDAMAN & NICOBAR ISLANDS	1904	9.4	14.7	0.0	202.4	304.5	495.1	502.0	160.1	820.4	222.2	308.7	40.1	3079.6	24.1	506.9	1977.6	571.0
4	ANDAMAN & NICOBAR ISLANDS	1905	1.3	0.0	3.3	26.9	279.5	628.7	368.7	330.5	297.0	260.7	25.4	344.7	2566.7	1.3	309.7	1624.9	630.8
5	ANDAMAN & NICOBAR ISLANDS	1906	36.6	0.0	0.0	0.0	556.1	733.3	247.7	320.5	164.3	267.8	128.9	79.2	2534.4	36.6	556.1	1465.8	475.9
6	ANDAMAN & NICOBAR ISLANDS	1907	110.7	0.0	113.3	21.6	616.3	305.2	443.9	377.6	200.4	264.4	648.9	245.6	3347.9	110.7	751.2	1327.1	1158.9
7	ANDAMAN & NICOBAR ISLANDS	1908	20.9	85.1	0.0	29.0	562.0	693.6	481.4	699.9	428.8	170.7	208.1	196.9	3576.4	106.0	591.0	2303.7	575.7
8	ANDAMAN & NICOBAR ISLANDS	1910	26.6	22.7	206.3	89.3	224.5	472.7	264.3	337.4	626.6	208.2	267.3	153.5	2899.4	49.3	520.1	1701.0	629.0
9	ANDAMAN & NICOBAR ISLANDS	1911	0.0	8.4	0.0	122.5	327.3	649.0	253.0	187.1	464.5	333.8	94.5	247.1	2687.2	8.4	449.8	1553.6	675.4

Figure 4.1: Data

Source of image : jupyter notebook

Rainfall data of 115 years according to sub-division from 1901-2015,[data.world 2017]: The portal provides the data by the government of rainfall in India from the year 1901 to 2015 in comma-separated values file(CSV), which have variables like -Rainfall according to the states in India on the monthly basis, Rainfall season-wise and annually, the measurement of the rainfall is in MM(Millimeter).

Rainfall data for State-wise

The Rainfall data is available in CSV format for required years, it provides the data by the government of rainfall from 1901 to 2015 there are every state that are shown in the data set, and their rainfall measurement in MM for the input variables according to the month and season.

Loading and checking data for null values

Jupiter notebook is used as the platform to process, all the necessary libraries have been imported

that will be used, further considering the regression model as it will be applied in the current study and to complete the coding section for the preparation of the data. Importing of the libraries to load and analyze the data, feature selection, and regression analysis. The screenshot below depicts the importing of all the packages and libraries needed for the study.

```
In [14]: #Import packages
         import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
         %matplotlib inline
         from matplotlib.pylab import rcParams
         import seaborn as sns
         rcParams['figure.figsize']=10,8
         #importing normalisation package
         from sklearn.preprocessing import StandardScaler
         #importing train test split
         from sklearn.model_selection import train_test_split
         #importing mean squared error and mean absolute error
         from sklearn.metrics import mean_squared_error,mean_absolute_error,r2_score
         from scipy import stats
         #importing feature selection libraries
         from sklearn.feature_selection import SelectKBest
         from sklearn.feature_selection import  RFE
         from sklearn import model_selection
         #importing label encoder to encode text data into their numerical factors
         from sklearn.preprocessing import LabelEncoder, OneHotEncoder
         #importing regression packages
         from sklearn.linear_model import LinearRegression
         from sklearn.ensemble import RandomForestRegressor
         from sklearn.datasets import make_regression
         from sklearn.neighbors import KNeighborsRegressor
         from sklearn.svm import SVR
         from sklearn.tree import DecisionTreeRegressor
```

Figure 4.2: Importing packages

Source of image : jupyter notebook

After that next step is to load the data-set as the models need to be trained and check if there are any null values in the data and removing the null values is one of the very important tasks while training the models because if a data-set carries a null value, it can adversely impact the accuracy and the performance of the machine learning, so we have to check and remove all the null values before applying any machine learning algorithm and then print first five rows to have initial insight on the data-set.

```
#Importing data
data= pd.read_excel('data_dissertation.xlsx')
#checking nulls
print("Status of nulls in data - ",data.isnull().values.any())
data=data.dropna()
print("Status of nulls in data - ",data.isnull().values.any())
new_data=pd.DataFrame(data)
new_data.head(5)
```

```
Status of nulls in data -  True
Status of nulls in data -  False
```

Figure 4.3: Loading data and checking null

Source of image : jupyter notebook

After data was loaded then null values were checked and few null values were found in the data, which has to be removed as it can affect the accuracy of the model, so null values were dropped and then check again whether there are any null values, then data is to be split-ted into the test.

4.1.3. Features and target variable

The target variable is annual rainfall, which is to be predicted. The data is then split-ted into the features and the target variable with test and test percent by using the train-test-split of the sklearn library in python. There are some categorical values in the data hence we have to transform the categorical data features into numerical factor or one Label Encoder, we will use Label Encoder library from sklearn. It is used to transform all the values that are categorical to numerical factors and further x is transformed into an array with the help of function call as fit-transform, which is split-ted in train and test data. Therefore the input and target variables are split-ted into train and test data, the test size is 30 and the train size is 70.

```
X=new_data.drop('annual',1)
Y=new_data['annual']

#Encoding the categorical variables:
labelencoder_X = LabelEncoder()

X['subdivision']= labelencoder_X.fit_transform(X['subdivision'])

#Splitting the dataset into the Training and Test dataset
X_train, X_test, Y_train, Y_test = train_test_split( X, Y, test_size=0.3, random_state=33)
```

Figure 4.4: Label Encoder

Source of image : jupyter notebook

4.1.4. Creating new features

The current study is for predicting the rainfall keeping agriculture sector into consideration as the data has rainfall according to monthly, season and the annual rainfall, there are few more factor which has to be added which can be more help full from the agriculture aspect. In the data, there are features like a month, season, and annual rainfall of rainfall but we do not have any features for the season of the crops as the prediction of the rainfall is done majorly for the agriculture purpose. The features such as seasonal-wise rainfall data have been replaced with the seasonal data as per cropping season.so we need to add three more features that are according to the season of the crops. The features will be Kharif,Rabi, and Zaid. kharif season starts from July till October, and then after there is Rabi season that starts from October until March and the last one is Zaid from March to June. So the month's rainfall measurement will be merged according to that like Kharif season starts from July - October, Rabi season that starts from October to March and Zaid from March - June therefore the monthly data will be merged according to the season of the crops. So these features are added as shown below in the figure.

```
new_data=pd.DataFrame(data)
#Creatig the columns according to the season of the crops.
new_data["Kharif"]=new_data["jul"]+ new_data["aug"]+new_data["sep"]+new_data["oct"]
new_data["Rabi"]=new_data["oct"]+ new_data["nov"]+new_data["dec"]+new_data["jan"]+new_data["feb"]+new_data["mar"]
new_data["Zaib"]=new_data["mar"]+ new_data["apr"]+new_data["may"]+new_data["jun"]+new_data["jul"]
warnings.filterwarnings('ignore')

new_data.head(5)
```

	subdivision	year	jan	feb	mar	apr	may	jun	jul	aug	sep	oct	nov	dec	annual	Kharif	Rabi	Zaib
0	ANDAMAN & NICOBAR ISLANDS	1901	49.2	87.1	29.2	2.3	528.8	517.5	365.1	481.1	332.6	388.5	558.2	33.6	3373.2	1567.3	1145.8	1442.9
1	ANDAMAN & NICOBAR ISLANDS	1902	0.0	159.8	12.2	0.0	446.1	537.1	228.9	753.7	666.2	197.2	359.0	160.5	3520.7	1846.0	888.7	1224.3
2	ANDAMAN & NICOBAR ISLANDS	1903	12.7	144.0	0.0	1.0	235.1	479.9	728.4	326.7	339.0	181.2	284.4	225.0	2957.4	1575.3	847.3	1444.4
3	ANDAMAN & NICOBAR ISLANDS	1904	9.4	14.7	0.0	202.4	304.5	495.1	502.0	160.1	820.4	222.2	308.7	40.1	3079.6	1704.7	595.1	1504.0
4	ANDAMAN & NICOBAR ISLANDS	1905	1.3	0.0	3.3	26.9	279.5	628.7	368.7	330.5	297.0	260.7	25.4	344.7	2566.7	1256.9	635.4	1307.1

Figure 4.5: Creating new features

Source of image : jupyter notebook

as show in the screenshot the features have been added according to the season of the crops.

4.1.5. Implementation of regression models

The Five of the regression algorithms are stored in an array which is named as "models" for testing the null hypothesis. The models which were saved were Multiple linear regression, KNN regression, SVM(Suport Vector Machine) regression, DTR(Decision tree regression), and RFE(Random forest regression) and also the Root Mean Square Error (RMSE), R-squared scores, name of the algorithms, predicted and actual values.

```
# preparing models
models = []
models.append(('Multiple Linear Regression', LinearRegression()))
models.append(('Random Forest Regression', RandomForestRegressor()))
models.append(('K Neighbors Regression', KNeighborsRegressor(n_neighbors=3)))
models.append(('Support Vector Regression', SVR(C=1.0, epsilon=0.2)))
models.append(('Decision Tree Regression', DecisionTreeRegressor(random_state = 0)))

# evaluate each model in turn
rmse_scores = []
r_scores = []
names = []
predicted = []
actual = []

for name, model in models:

  model.fit(X_train,Y_train)

  #Predicting the Test set results:
  y_pred = np.round(model.predict(X_test),decimals=2)
  rmse = np.sqrt(mean_squared_error(Y_test,y_pred))
  r_score=r2_score(Y_test, y_pred)

  #storing value in array to be used in data frame
  predicted=y_pred
  actual=Y_test
  rmse_scores.append(np.round(rmse,2))
  r_scores.append(np.round(r_score*100,2))
  names.append(name)


df=pd.DataFrame({'Algorithm': names,'RMSE': rmse_scores, 'R2_Score': r_scores})


#df.boxplot(by="name", column="RMSE")
plotdata = pd.DataFrame({"RMSE":rmse_scores},index=['MLR','RF','KNR','SVR','DTR'])
plotdata.plot.bar(figsize=(8,5),title="Algorithm RMSE comparision")
df
```

Figure 4.6: Preparing models

Source of image : jupyter notebook

For every algorithm, the for loop is iterated and Root Mean Square Error (RMSE) and R-squared scores are then stored in a data frame.

4.1.6. Outputs of the regression

Below are shown all five algorithms that were used for the study and there RMSE and R-squared accordingly. when compared the score of all RMSE and R-squared it can be seen that the multiple linear regression is the most efficient one with having the RMSE score of 0.10 and R-squared score of 100 percent,It could also be noted that the Random Forest Regression also performed reasonably well as being similar to MLR with RMSE of 87.12 percent and R-squared of 99.11 percent. The decision tree regression also performed promisingly well with RMSE score of 137.34 and R-squared score of 97.79 percent. whereas KNN regression amounted the RMSE score of 83.85 and R-squared scores of 99.18. The worst performing among all is the SVM(support vector machine regression) having RMSE score of 974.25 and R-squared score of -11.01, Thus makes it rejected according to the null hypothesis.

	Algorithm	RMSE	R2_Score
0	Multiple Linear Regression	0.10	100.00
1	Random Forest Regression	87.12	99.11
2	K Neighbors Regression	83.85	99.18
3	Support Vector Regression	974.25	-11.01
4	Decision Tree Regression	137.34	97.79

Figure 4.7: Outputs

Source of image : jupyter notebook

Below is the screenshot of the bar graph that shows the RMSE of all five Regression used in the study.

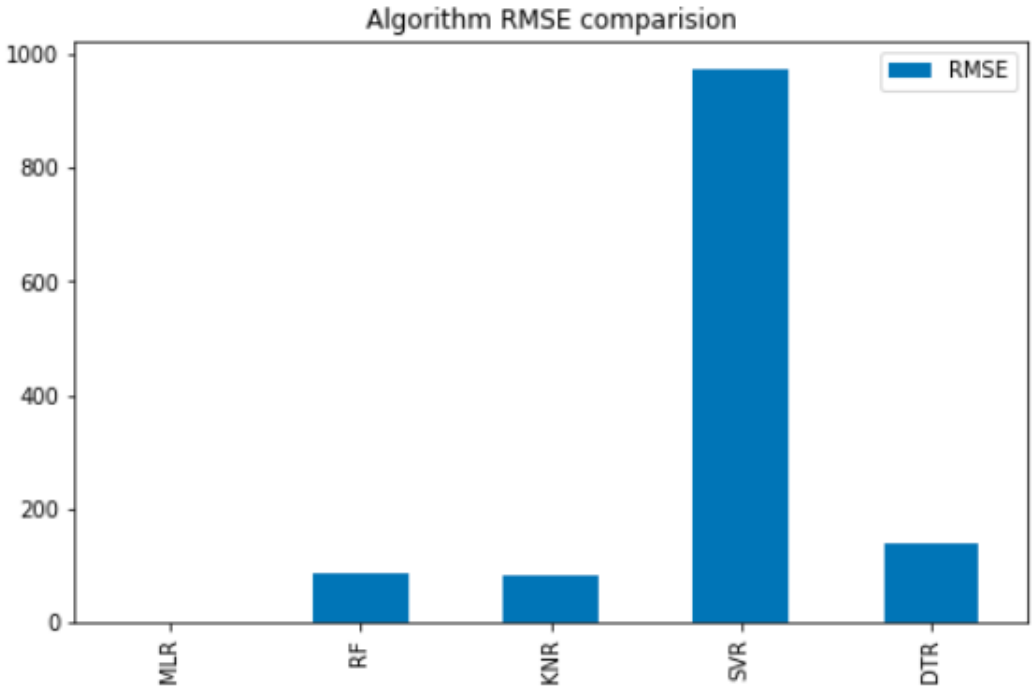

Figure 4.8: Graph for RMSE

Source of image : jupyter notebook

4.1.7. Checking ordinary least square

Ordinary least squares regression is the statistical analytical technique which shows the relationship in between the one or more explanatory variable and a target variable.

To show ordinary least square(OLS) summary for Multiple LR are below.

```
: ##########################################################################
  ##################to show OLS summarry for Multiple LR####################
  ##########################################################################
  import statsmodels.api as sm

  #Encoding the categorical variables:
  labelencoder_X = LabelEncoder()

  new_data['subdivision']= labelencoder_X.fit_transform(new_data['subdivision'])

  #y-target features
  Y=new_data.iloc[:,new_data.columns == 'annual'].values
  #X-input features
  X=new_data.iloc[:,new_data.columns != 'annual'].values



  X_opt= X[:, [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16]]
  #Optimal X contains the highly impacted independent variables

  regressor_OLS=sm.OLS(endog = Y, exog = X_opt ).fit()
  regressor_OLS.summary()
```

Figure 4.9: Checking ordinary least square

Source of image : jupyter notebook

It can be seen that the R-squared score for least square method is 100 percent.

OLS Regression Results

Dep. Variable:	y	**R-squared (uncentered):**	1.000
Model:	OLS	**Adj. R-squared (uncentered):**	1.000
Method:	Least Squares	**F-statistic:**	8.397e+10
Date:	Thu, 20 Aug 2020	**Prob (F-statistic):**	0.00
Time:	18:56:20	**Log-Likelihood:**	3669.1
No. Observations:	4090	**AIC:**	-7310.
Df Residuals:	4076	**BIC:**	-7222.
Df Model:	14		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	0.0001	0.000	0.853	0.394	-0.000	0.000
x2	2.286e-06	2.66e-06	0.858	0.391	-2.94e-06	7.51e-06
x3	0.3093	5.13e-05	6029.978	0.000	0.309	0.309
x4	0.3094	5.32e-05	5811.688	0.000	0.309	0.310
x5	-0.3197	4.03e-05	-7937.284	0.000	-0.320	-0.320
x6	0.3711	3.39e-05	1.1e+04	0.000	0.371	0.371
x7	0.3712	2.09e-05	1.77e+04	0.000	0.371	0.371
x8	0.3712	1.49e-05	2.5e+04	0.000	0.371	0.371
x9	-0.1649	1.11e-05	-1.48e+04	0.000	-0.165	-0.165
x10	0.4639	1.19e-05	3.9e+04	0.000	0.464	0.464
x11	0.4639	1.39e-05	3.35e+04	0.000	0.464	0.464
x12	-0.2268	1.76e-05	-1.29e+04	0.000	-0.227	-0.227
x13	0.3092	2.88e-05	1.08e+04	0.000	0.309	0.309
x14	0.3093	4.11e-05	7529.591	0.000	0.309	0.309
x15	0.5361	5.85e-06	9.16e+04	0.000	0.536	0.536
x16	0.6907	1.03e-05	6.67e+04	0.000	0.691	0.691
x17	0.6288	8.11e-06	7.76e+04	0.000	0.629	0.629

Omnibus:	1.212	**Durbin-Watson:**	1.998
Prob(Omnibus):	0.546	**Jarque-Bera (JB):**	1.164
Skew:	-0.016	**Prob(JB):**	0.559
Kurtosis:	3.077	**Cond. No.**	6.52e+16

Figure 4.10: Ordinary least square Result

Source of image : jupyter notebook

4.1.8. Implying feature selection by RFE method

The feature selection is the process in which the best feature is selected that is most related to the predictive variable, it can be manually or automatically. Using the Feature selection can reduce the over-fitting as there will be features that are more relevant and thus it can improve the accuracy because of less miss-leading data.

Recursive Feature Elimination(RFE) is used in the current study as it fits the current model and is very effective while suggesting the less amount of features, the features with less weight-age is re-

moved. Recursive Feature Elimination(RFE) is shown below in the screenshot. Optimum numbers of features that are suggested are 12 by RFE, Score with 12 features is 1.000000. The extracted names of the features are shown below. The label encoder library is used from the sklearn to convert the categorical variable into numerical. There is a feature that is "subdivision" that is categorical that represent the states in India, so it was converted to numerical with the help of label encoder.

The optimum number of features and the score is shown below in the figure.

```
#checking total recommended features by RFE - reverse feature elimination method
import warnings
from sklearn.exceptions import DataConversionWarning
warnings.filterwarnings(action='ignore', category=DataConversionWarning)


X=new_data.drop('annual',1)
Y=new_data['annual']

#Encoding the categorical variables:
labelencoder_X = LabelEncoder()

X['subdivision']= labelencoder_X.fit_transform(X['subdivision'])

#no of features
nof_list=np.arange(1,17)
high_score=0
#Variable to store the optimum features
nof=0
score_list =[]
for n in range(len(nof_list)):
    X_train, X_test, y_train, y_test = train_test_split(X,Y, test_size = 0.3, random_state = 0)
    model = LinearRegression()
    rfe = RFE(model,nof_list[n])
    X_train_rfe = rfe.fit_transform(X_train,y_train)
    X_test_rfe = rfe.transform(X_test)
    model.fit(X_train_rfe,y_train)
    score = model.score(X_test_rfe,y_test)
    score_list.append(score)
    if(score>high_score):
        high_score = score
        nof = nof_list[n]
print("Optimum number of features: %d" %nof)
print("Score with %d features: %f" % (nof, high_score))
```

```
Optimum number of features: 12
Score with 12 features: 1.000000
```

Figure 4.11: Feature checking

Source of image : jupyter notebook

4.1.9. Indexing of top 12 variables

As suggested in previous column, we will try to get the names of the top 12 variables from the Recursive RFE(Feature Elimination).RFE model was Initialized and then data Transformed using RFE(Recursive Feature Elimination) and then the data was Fitted to model.It can be seen that the 12 optimum features are 'jan','feb', 'apr', 'may', 'jun', 'aug', 'sep','nov', 'dec', 'Kharif','Rabi', 'Zaib'.these are are features that contribute most to the target variable as shown below.

```
###### As suggested in previous column, we will try to get the names of the top 12 variables

cols = list(new_data.drop('annual',1).columns)
y=new_data['annual']

model = LinearRegression()
#Initializing RFE model
rfe = RFE(model, 12)
#Transforming data using RFE
X_rfe = rfe.fit_transform(X,y)
#Fitting the data to model
model.fit(X_rfe,y)
temp = pd.Series(rfe.support_,index = cols)
selected_features_rfe = temp[temp==True].index
print(selected_features_rfe)
```

```
Index(['jan', 'feb', 'apr', 'may', 'jun', 'aug', 'sep', 'nov', 'dec', 'Kharif',
       'Rabi', 'Zaib'],
      dtype='object')
```

Figure 4.12: Getting feature names

Source of image : jupyter notebook

4.1.10. Predicting results with recommended features(using feature elimination technique)

After applying feature selection by recursive feature elimination(RFE) method top 12 recommended features are taken under consideration for regression analysis to verify previous results. It is also expected to achieve more accuracy. Recursive feature elimination(RFE) has been used for eliminating unnecessary features from the data-set, which suggested 12 features. The result after applying feature selection by recursive feature elimination(RFE) method are shown below:

	Algorithm	RMSE	R2_Score
0	Multiple Linear Regression	0.10	100.00
1	Random Forest Regression	92.30	99.00
2	K Neighbors Regression	78.02	99.29
3	Support Vector Regression	974.25	-11.01
4	Decision Tree Regression	128.20	98.08

Figure 4.13: Predicting results with recommended features

Source of image : jupyter notebook

Regression analysis after applying feature selection shows that there is slight or no effects on r-square scores for all five algorithms. Multiple linear regression still performs at the top with a 100 percent r-square score and 0.10 RMSE score, which is unchanged with respect to regression analysis without feature selection. In addition, random forest regression is also performing close to the previous r-square score with 99.00 percent and RMSE from 87.12 to 92.30.In case of KNN and DTR there is also slight changes as compared to previous result with RMSE of 78.02 and 128.20 and R score of 99.29 and 98.08 respectively. Where as support vector machine remains unchanged. We can summarise the analysis after feature selection as a beneficial step towards improving the efficiencies of the models under consideration, where multiple linear regression tops the chart.

4.1.11. features importance

Significant features or feature importance is evaluated by a decrease in the impurity of the nodes weighted by the ease of that node. The below figure shows the feature importance in the descending order for all features, where it can be seen that "Zaib" is in the top of the chart with the biggest margin that is 0.88 percent, further followed by "Kharif" and "rabi" with 0.07 percent and 0.018 respectively.All other features like "sep","aug","nov","jun","dec","may","Jan","apr" except of the feature "Feb" with the least one with 0.001 percent among all the features.

	Importances
Zaib	0.886343
Kharif	0.076904
Rabi	0.018035
aug	0.002837
nov	0.002632
jun	0.002326
may	0.002246
dec	0.002193
jan	0.002034
sep	0.001803
apr	0.001591
feb	0.001055

Figure 4.14: Important features

Source of image : jupyter notebook

Below is the graph that depict the feature importance using the forest regression.

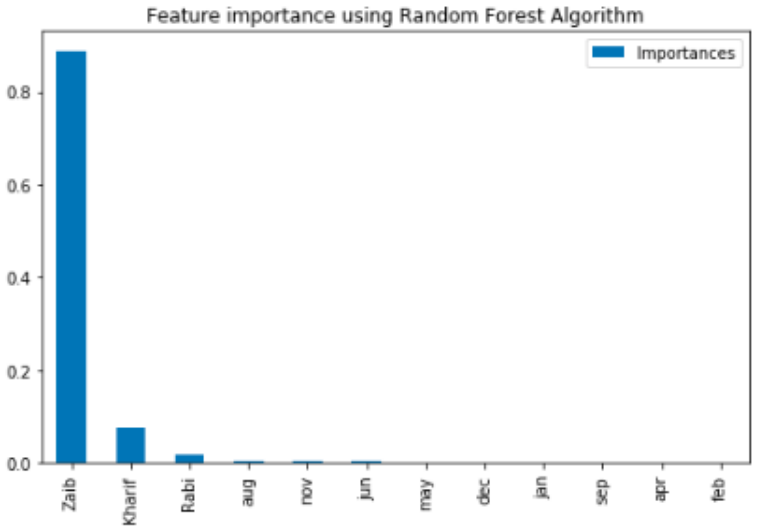

Figure 4.15: Feature importance using the forest regression

Source of image : jupyter notebook

4.1.12. Re-predicting results with top 12 features

After evaluating the feature importance, then the prediction is to be done again to analyze whether there is any difference in the prediction or not.
below is the screen capture depicting the result of the models.

	Algorithm	RMSE	R2_Score
0	Multiple Linear Regression	0.10	100.00
1	Random Forest Regression	92.30	99.00
2	K Neighbors Regression	78.02	99.29
3	Support Vector Regression	974.25	-11.01
4	Decision Tree Regression	128.20	98.08

Figure 4.16: Re-predicting results with top 12 features

Source of image : jupyter notebook

4.1.13. Evaluating the means absolute error

Means absolute error(MAE) is to evaluate the absolute mean distance between the actual data and the predicted data. It can be said that if absolute difference means has a negative sign in the result, then it is omitted. Mean absolute error of Linear and random forest regression, to find the difference between both the models is evaluated. In other words, it can be said that checking the overfilling, the noise (i.e. random fluctuations) in the training set is learned as rules/patterns by the model. However, these noisy learned representations do not apply to new unseen data and thus, the model's performance (i.e. accuracy, MSE, MAE-Means absolute error) is negatively impacted.

	Algorithm	RMSE	R score	Mean abs error	Mean abs error %
0	Multiple Linear Regression	0.1	100.0	[0.07448247758761191]	0.01
1	Random Forest Regression	92.3	99.0	[60.26407497962509]	4.61

The result depicted that the Multiple Linear Regression RMSE score is 0.10 with an R score 100

Figure 4.17: Means absolute error

Source of image : jupyter notebook

percent and the Mean abs error is 0.01 percent. Whereas Random Forest Regression RMSE score is 92.3 and R score 99.0 percent and Mean abs error is 4.61 percent. Hence it can be seen that Multiple Linear Regression is the most efficient model in the current study with a better result.

4.1.14. Statistical test for significant difference between the two models(linear regression and random forest regressor)

In order to conduct a statistical test to calculate the importance of the difference and the efficiency between two the models evaluated by MLR(multiple linear regression) and RF(random forest regression) the features such as the actual values, predicted values and the relative error values are shown in the table below. The comparison of both MLR and RF actual value and predicted values and the error are shown below.

	Actual Value	Predicted Value LR	Predicted Value RF	LR error	RF error
0	2423.0	2422.90	2582.09	-0.10	159.09
1	383.9	383.90	391.32	0.00	7.42
2	1332.7	1332.70	1299.88	0.00	-32.82
3	808.8	808.71	834.30	-0.09	25.50
4	697.8	697.90	691.98	0.10	-5.82
5	1444.4	1444.41	1425.13	0.01	-19.27
6	2755.4	2755.41	2700.33	0.01	-55.07
7	746.7	746.71	724.46	0.01	-22.24
8	962.4	962.50	1013.76	0.10	51.36
9	1514.5	1514.40	1525.72	-0.10	11.22
10	1436.7	1436.71	1401.47	0.01	-35.23
11	1072.0	1072.01	1074.47	0.01	2.47
12	547.8	547.71	563.73	-0.09	15.93
13	984.7	984.60	992.16	-0.10	7.46
14	1047.1	1047.08	920.26	-0.02	-126.84
15	793.5	793.71	798.13	0.21	4.63
16	636.1	636.20	765.15	0.10	129.05
17	2487.4	2487.39	2404.40	-0.01	-83.00
18	3814.5	3814.50	3177.86	0.00	-636.64

Figure 4.18: Statistical test

Source of image : jupyter notebook

Performing t-test

T-test is carried out on the table below by using the stats package of scipy library to compare the error and the p-values of MLR(multiple linear regression) and RF(random forest regression), that can assist in determining if there is any significant statistical difference between both of the models or not. The t-test is an inferential statistic that is used to evaluate whether there is any substantial difference between the two models or not. The below screenshot shows the mean error of both MLR(multiple linear regression) and RF(random forest regression) and the P-value.

```
:  #performing t-test on final two algorithms

   from scipy import stats


   print("Mean error of LR and RF respectively",np.mean(np.absolute(difference['LR error'])),
         np.mean(np.absolute(difference['RF error'])),"\n")
   #print(np.mean(difference['LR error']),np.mean(difference['RF error']))

   print(stats.ttest_rel(difference['LR error'],difference['RF error']))

   Mean error of LR and RF respectively 0.07474327628361668 57.681018744906346

   Ttest_relResult(statistic=2.740330694004615, pvalue=0.006226841672247562)
```

Figure 4.19: Performing t-test

Source of image : jupyter notebook

If the p-value less than 0.05 then it can be said that it is statistically significant and if A p-value higher than 0.05 then it is not statistically significant and indicates strong evidence for the null hypothesis. Below it can be seen that the Mean error of MLR(multiple linear regression) are respectively 0.074 and 57.54.The pvalue is 0.006 as the p-value is less than 0.5, so the p-value is significant to consider in any confidence interval and the mean difference of MLR is lesser than RF model, Hence we can conclude that the two models are significantly different.

CHAPTER 5

DISCUSSIONS AND CONCLUSION

Although many techniques are used to predict the rainfall, by doing research in the models that are currently used in the prediction of the rainfall by going through in the shot-comes and the possibility to change them accordingly to make the prediction more precise.
In this study, the aim is to identify which machine learning(regression) above all five that are Decision tree regression, KNN regression, Multiple linear regression, RFE(Random forest regression) and Support Vector Machine regression can predict the rainfall with most efficiency keeping the agriculture sector into consideration.

5.0.1. Analysing the influence of input variables or features on target variable i.e annual rainfall?

In the current studies data of the rainfall is used and applied to check whether there are any effects of features that are monthly rainfall and season of the crop on the annual rainfall. It was found that the Zaib, Kharif, and Rabi features were most important features with 88, 0.07, and 0.01 percent respectively. The findings explain that Zaib is a very important factor for annual rainfall.

5.0.2. Which of the regression model is best for the prediction with the efficient outcome?

In the current study after comparing all the five regression models Decision tree regression, KNN regression, Multiple linear regression, RFE(Random forest regression), and Support Vector Machine regression. The result showed that the RMSE score for multiple linear regression is 0.10 with 100 percent R score and 0.01 percent for the mean abs error. Whereas, Random Forest Regression RMSE score is 92.30 and R score 99.29 percent and Mean abs error is 4.01 percent. To sum up it can be said that there is a difference among mean error of random forest regression and multiple linear regression, by 60.26 & 0.07 respectively. In support the p-value supports the difference as significant by 0.006, hence the statistical test proves the result as significantly different. Multiple linear regression has been proven as the most efficient model among all five for the prediction of rainfall in the current study.

5.0.3. Future work

The outcome of the model used in the study interestingly outperformed specifically MLR(Multiple linear regression) but in future additional features exploration has to be done and thus it may increase the data size and new features may carry categorical values, which can reduce the percent of error, so for that in future employment of other models like ANN(Artificial Neural Network) may be used.

REFERENCES

[Ahmed et al., 2013] Ahmed, I., Menon, S., and KB, N. (2013). Rainfall prediction using multiple regression technique. *International Journal of Applied Engineering Research*, 8(19):2013.

[Awan and Maqbool, 2010] Awan, J. A. and Maqbool, O. (2010). Application of artificial neural networks for monsoon rainfall prediction. In *2010 6th International Conference on Emerging Technologies (ICET)*, pages 27–32. IEEE.

[Chifurira and Chikobvu, 2014] Chifurira, R. and Chikobvu, D. (2014). A weighted multiple regression model to predict rainfall patterns: Principal component analysis approach. *Mediterranean Journal of Social Sciences*, 5(7):34.

[Chuang, 2019] Chuang, Y. (2019). Climate variability, rainfall shocks, and farmers' income diversification in india. *Economics Letters*, 174:55–61.

[Cramer et al., 2017] Cramer, S., Kampouridis, M., Freitas, A. A., and Alexandridis, A. K. (2017). An extensive evaluation of seven machine learning methods for rainfall prediction in weather derivatives. *Expert Systems with Applications*, 85:169–181.

[Deshpande, 2017] Deshpande, T. (2017). State of agriculture in india. *PRS Legislative Research*, pages 6–7.

[Geetha and Nasira, 2014] Geetha, A. and Nasira, G. (2014). Data mining for meteorological applications: Decision trees for modeling rainfall prediction. In *2014 IEEE International Conference on Computational Intelligence and Computing Research*, pages 1–4. IEEE.

[Gupta et al., 2013] Gupta, A., Gautam, A., Jain, C., Prasad, H., and Verma, N. (2013). Time series analysis of forecasting indian rainfall. *International Journal of Inventive Engineering and Sciences (IJIES)*, 1(6):42–45.

[Hasan et al., 2015] Hasan, N., Nath, N. C., and Rasel, R. I. (2015). A support vector regression model for forecasting rainfall. In *2015 2nd International Conference on Electrical Information and Communication Technologies (EICT)*, pages 554–559. IEEE.

[Kannan et al., 2010] Kannan, M., Prabhakaran, S., and Ramachandran, P. (2010). Rainfall forecasting using data mining technique.

[Krishna Kumar et al., 2004] Krishna Kumar, K., Rupa Kumar, K., Ashrit, R., Deshpande, N., and Hansen, J. W. (2004). Climate impacts on indian agriculture. *International Journal of Climatology: A Journal of the Royal Meteorological Society*, 24(11):1375–1393.

[Kumar and Parikh, 2001] Kumar, K. K. and Parikh, J. (2001). Indian agriculture and climate sensitivity. *Global environmental change*, 11(2):147–154.

[Mohapatra et al., 2017] Mohapatra, S. K., Upadhyay, A., and Gola, C. (2017). Rainfall prediction based on 100 years of meteorological data. In *2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*, pages 162–166. IEEE.

[Murali and Afifi, 2014] Murali, J. and Afifi, T. (2014). Rainfall variability, food security and human mobility in the janjgir-champa district of chhattisgarh state, india. *Climate and Development*, 6(1):28–37.

[Parmar et al., 2017] Parmar, A., Mistree, K., and Sompura, M. (2017). Machine learning techniques for rainfall prediction: A review. In *International Conference on Innovations in information Embedded and Communication Systems*.

[Prabakaran et al., 2017] Prabakaran, S., Kumar, P. N., and Tarun, P. S. M. (2017). Rainfall prediction using modified linear regression. *ARPN Journal of Engineering and Applied Sciences*, 12:3715–3718.

[Ramana et al., 2013] Ramana, R. V., Krishna, B., Kumar, S., and Pandey, N. (2013). Monthly rainfall prediction using wavelet neural network analysis. *Water resources management*, 27(10):3697–3711.

[Rosenzweig and Udry, 2014] Rosenzweig, M. R. and Udry, C. (2014). Rainfall forecasts, weather, and wages over the agricultural production cycle. *American Economic Review*, 104(5):278–83.

[Sahai et al., 2000] Sahai, A., Soman, M., and Satyan, V. (2000). All india summer monsoon rainfall prediction using an artificial neural network. *Climate dynamics*, 16(4):291–302.

[Sethi and Garg, 2014] Sethi, N. and Garg, K. (2014). Exploiting data mining technique for rainfall prediction. *International Journal of Computer Science and Information Technologies*, 5(3):3982–3984.

[Singh et al., 2014] Singh, S., Singh, K. M., Singh, R., Kumar, A., and Kumar, U. (2014). Impact of rainfall on agricultural production in bihar: A zone-wise analysis. *Environment & Ecology*, 32(4A):1571–1576.

[Singh, 1999] Singh, V. (1999). Indian agriculture. *Indian Econ. Data Res. Centre, New Delhi*, pages 397–478.

[Swain et al., 2017] Swain, S., Patel, P., and Nandi, S. (2017). A multiple linear regression model for precipitation forecasting over cuttack district, odisha, india. In *2017 2nd International Conference for Convergence in Technology (I2CT)*, pages 355–357. IEEE.

[Udmale et al., 2014] Udmale, P., Ichikawa, Y., Manandhar, S., Ishidaira, H., and Kiem, A. S. (2014). Farmers[U+05F3] perception of drought impacts, local adaptation and administrative mitigation measures in maharashtra state, india. *International Journal of Disaster Risk Reduction*, 10:250–269.

[Zaw and Naing, 2008] Zaw, W. T. and Naing, T. T. (2008). Empirical statistical modeling of rainfall prediction over myanmar. *World Academy of Science, Engineering and Technology*, 2(10):500–504.

APPENDIX A

SCREENS AND DESCRIPTION OF PYTHON CODES

Importing and preparing the data needed

The necessary packages of pandas are used for importing and handling data as a data frame and for executing all mathematical and array related operations numpy package is used, for performing regression, splitting data into train and test set and for the feature selection Sklearn library from Scikit-learn is used for importing the packeges.Matplot libraries for creating the plots is also used.

```
#Import packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from matplotlib.pylab import rcParams
import seaborn as sns
rcParams['figure.figsize']=10,8
#importing normalisation package
from sklearn.preprocessing import StandardScaler
#importing train test split
from sklearn.model_selection import train_test_split
#importing mean squared error and mean absolute error
from sklearn.metrics import mean_squared_error,mean_absolute_error,r2_score
from scipy import stats
#importing feature selection libraries
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import  RFE
from sklearn import model_selection
#importing label encoder to encode text data into their numerical factors
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
#importing regression packages
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.datasets import make_regression
from sklearn.neighbors import KNeighborsRegressor
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
import warnings
warnings.filterwarnings('ignore')
```

Importing data and the checking for null values

Importing the data and then further checking any null values and then dropping the null values.

```
#Importing data
data= pd.read_excel('data_dissertation.xlsx')
#checking nulls
print("Status of nulls in data - ",data.isnull().values.any())
data=data.dropna()
print("Status of nulls in data - ",data.isnull().values.any())
new_data=pd.DataFrame(data)
new_data.head(5)
```

```
Status of nulls in data -  True
Status of nulls in data -  False
```

	subdivision	year	jan	feb	mar	apr	may	jun	jul	aug	sep	oct	nov	dec	annual	jan_feb	mar_may	jun_sep	oct_dec
0	ANDAMAN & NICOBAR ISLANDS	1901	49.2	87.1	29.2	2.3	528.8	517.5	365.1	481.1	332.6	388.5	558.2	33.6	3373.2	136.3	560.3	1696.3	980.3
1	ANDAMAN & NICOBAR ISLANDS	1902	0.0	159.8	12.2	0.0	446.1	537.1	228.9	753.7	666.2	197.2	359.0	160.5	3520.7	159.8	458.3	2185.9	716.7
2	ANDAMAN & NICOBAR ISLANDS	1903	12.7	144.0	0.0	1.0	235.1	479.9	728.4	326.7	339.0	181.2	284.4	225.0	2957.4	156.7	236.1	1874.0	690.6
3	ANDAMAN & NICOBAR ISLANDS	1904	9.4	14.7	0.0	202.4	304.5	495.1	502.0	160.1	820.4	222.2	308.7	40.1	3079.6	24.1	506.9	1977.6	571.0
4	ANDAMAN & NICOBAR ISLANDS	1905	1.3	0.0	3.3	26.9	279.5	628.7	368.7	330.5	297.0	260.7	25.4	344.7	2566.7	1.3	309.7	1624.9	630.8

Creating and deleting features to the dataset

There were some features that were added named as "Zaib","Kharif" and "Rabi".

Some of the seasonal features were deleted from csv level that were "jan-feb","mar-may","jun-sep" and "oct-dec". New features were added to the dataset while using the python code.

The data set after adding the features is shown below.

```
new_data=pd.DataFrame(data)
#Creatig the columns according to the season of the crops.
new_data["Kharif"]=new_data["jul"]+ new_data["aug"]+new_data["sep"]+new_data["oct"]
new_data["Rabi"]=new_data["oct"]+ new_data["nov"]+new_data["dec"]+new_data["jan"]+new_data["feb"]+new_data["mar"]
new_data["Zaib"]=new_data["mar"]+ new_data["apr"]+new_data["may"]+new_data["jun"]+new_data["jul"]
warnings.filterwarnings('ignore')

new_data.head(5)
```

	subdivision	year	jan	feb	mar	apr	may	jun	jul	aug	sep	oct	nov	dec	annual	Kharif	Rabi	Zaib
0	ANDAMAN & NICOBAR ISLANDS	1901	49.2	87.1	29.2	2.3	528.8	517.5	365.1	481.1	332.6	388.5	558.2	33.6	3373.2	1567.3	1145.8	1442.9
1	ANDAMAN & NICOBAR ISLANDS	1902	0.0	159.8	12.2	0.0	446.1	537.1	228.9	753.7	666.2	197.2	359.0	160.5	3520.7	1846.0	888.7	1224.3
2	ANDAMAN & NICOBAR ISLANDS	1903	12.7	144.0	0.0	1.0	235.1	479.9	728.4	326.7	339.0	181.2	284.4	225.0	2957.4	1575.3	847.3	1444.4
3	ANDAMAN & NICOBAR ISLANDS	1904	9.4	14.7	0.0	202.4	304.5	495.1	502.0	160.1	820.4	222.2	308.7	40.1	3079.6	1704.7	595.1	1504.0
4	ANDAMAN & NICOBAR ISLANDS	1905	1.3	0.0	3.3	26.9	279.5	628.7	368.7	330.5	297.0	260.7	25.4	344.7	2566.7	1256.9	635.4	1307.1

Splitting the data in train and test and converting the categorical values into numerical

There are some catagorical values in the data, So Converting categorical values into numerical using LabelEncoder.

```
X=new_data.drop('annual',1)
Y=new_data['annual']

#Encoding the categorical variables:
labelencoder_X = LabelEncoder()

X['subdivision']= labelencoder_X.fit_transform(X['subdivision'])

#Splitting the dataset into the Training and Test dataset
X_train, X_test, Y_train, Y_test = train_test_split( X, Y, test_size=0.3, random_state=33)
```

Train and test sets are created using the $train_test_split library, with 30\% and 70\% data respectively, which is follow$

Preparing models for train and test fit and applying rgression

applying regression as array and saving data output in as data-frame

```
# preparing models
models = []
models.append(('Multiple Linear Regression', LinearRegression()))
models.append(('Random Forest Regression', RandomForestRegressor()))
models.append(('K Neighbors Regression', KNeighborsRegressor(n_neighbors=3)))
models.append(('Support Vector Regression', SVR(C=1.0, epsilon=0.2)))
models.append(('Decision Tree Regression', DecisionTreeRegressor(random_state = 0)))

# evaluate each model in turn
rmse_scores = []
r_scores = []
names = []
predicted = []
actual = []

for name, model in models:

  model.fit(X_train,Y_train)

  #Predicting the Test set results:
  y_pred = np.round(model.predict(X_test),decimals=2)
  rmse = np.sqrt(mean_squared_error(Y_test,y_pred))
  r_score=r2_score(Y_test, y_pred)

  #storing value in array to be used in data frame
  predicted=y_pred
  actual=Y_test
  rmse_scores.append(np.round(rmse,2))
  r_scores.append(np.round(r_score*100,2))
  names.append(name)


df=pd.DataFrame({'Algorithm': names,'RMSE': rmse_scores, 'R2_Score': r_scores})


#df.boxplot(by="name", column="RMSE")
plotdata = pd.DataFrame({"RMSE":rmse_scores},index=['MLR','RF','KNR','SVR','DTR'])
plotdata.plot.bar(figsize=(8,5),title="Algorithm RMSE comparision")
df
```

Depicting OLS summarry for Multiple LR

```
: ##########################################################################
  ##################to show OLS summarry for Multiple LR####################
  ##########################################################################
  import statsmodels.api as sm

  #Encoding the categorical variables:
  labelencoder_X = LabelEncoder()

  new_data['subdivision']= labelencoder_X.fit_transform(new_data['subdivision'])

  #y-target features
  Y=new_data.iloc[:,new_data.columns == 'annual'].values
  #X-input features
  X=new_data.iloc[:,new_data.columns != 'annual'].values



  X_opt= X[:, [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16]]
  #Optimal X contains the highly impacted independent variables

  regressor_OLS=sm.OLS(endog = Y, exog = X_opt ).fit()
  regressor_OLS.summary()
```

Applying Feature Selectcion method

checking total recommended features by RFE - reverse feature elimination method.

```
#checking total recommended features by RFE - reverse feature elimination method
import warnings
from sklearn.exceptions import DataConversionWarning
warnings.filterwarnings(action='ignore', category=DataConversionWarning)


X=new_data.drop('annual',1)
Y=new_data['annual']

#Encoding the categorical variables:
labelencoder_X = LabelEncoder()

X['subdivision']= labelencoder_X.fit_transform(X['subdivision'])

#no of features
nof_list=np.arange(1,17)
high_score=0
#Variable to store the optimum features
nof=0
score_list =[]
for n in range(len(nof_list)):
    X_train, X_test, y_train, y_test = train_test_split(X,Y, test_size = 0.3, random_state = 0)
    model = LinearRegression()
    rfe = RFE(model,nof_list[n])
    X_train_rfe = rfe.fit_transform(X_train,y_train)
    X_test_rfe = rfe.transform(X_test)
    model.fit(X_train_rfe,y_train)
    score = model.score(X_test_rfe,y_test)
    score_list.append(score)
    if(score>high_score):
        high_score = score
        nof = nof_list[n]
print("Optimum number of features: %d" %nof)
print("Score with %d features: %f" % (nof, high_score))
```

```
Optimum number of features: 12
Score with 12 features: 1.000000
```

Getting names of top 12 variables

As suggested in previous column, we will try to get the names of the top 12 variables.

```
cols = list(new_data.drop('annual',1).columns)
y=new_data['annual']

model = LinearRegression()
#Initializing RFE model
rfe = RFE(model, 12)
#Transforming data using RFE
X_rfe = rfe.fit_transform(X,y)
#Fitting the data to model
model.fit(X_rfe,y)
temp = pd.Series(rfe.support_,index = cols)
selected_features_rfe = temp[temp==True].index
print(selected_features_rfe)
```

Re-predicting results with top 12 features

```
X=new_data.drop('annual',1)
X=X.drop('mar',1)
X=X.drop('jul',1)
X=X.drop('oct',1)
X=X.drop('subdivision',1)
X=X.drop('year',1)

Y=new_data['annual']


#Splitting the dataset into the Training and Test dataset
X_train, X_test, Y_train, Y_test = train_test_split( X, Y, test_size=0.3, random_state=33)

# prepare models
models = []
models.append(('Multiple Linear Regression', LinearRegression()))
models.append(('Random Forest Regression', RandomForestRegressor(max_depth=10, random_state=21)))
models.append(('K Neighbors Regression', KNeighborsRegressor(n_neighbors=8)))
models.append(('Support Vector Regression', SVR(C=1.0, epsilon=0.2)))
models.append(('Decision Tree Regression', DecisionTreeRegressor(random_state = 0)))

# evaluate each model in turn
rmse_scores = []
r_scores = []
names = []
predicted = []
actual = []

for name, model in models:

  model.fit(X_train,Y_train)

  #7 Predicting the Test set results:
  y_pred = np.round(model.predict(X_test),decimals=2)
  rmse = np.sqrt(mean_squared_error(Y_test,y_pred))
  r_score=r2_score(Y_test, y_pred)

  #storing value in array to be used in data frame
  predicted=y_pred
  actual=Y_test
  rmse_scores.append(np.round(rmse,2))
  r_scores.append(np.round(r_score*100,2))
  names.append(name)


df=pd.DataFrame({'Algorithm': names,'RMSE': rmse_scores, 'R2_Score': r_scores})



#df.boxplot(by="name", column="RMSE")
plotdata = pd.DataFrame({"RMSE":rmse_scores},index=['MLR','RF','KNR','SVR','DTR'])
plotdata.plot.bar(figsize=(8,5),title="Algorithm RMSE comparision")

df
```

Analysing feature importance with top 12 features, using random forest trees(feature importance(s) library)

```
X=new_data.drop('annual',1)
X=X.drop('mar',1)
X=X.drop('jul',1)
X=X.drop('oct',1)
X=X.drop('subdivision',1)
X=X.drop('year',1)


Y=new_data['annual']



#Splitting the dataset into the Training and Test dataset
X_train, X_test, Y_train, Y_test = train_test_split( X, Y, test_size=0.3, random_state=33)

regressor=RandomForestRegressor()
regressor.fit(X_train,Y_train)

#Creating dataframe of feature importances
feature_importances = pd.DataFrame(regressor.feature_importances_,
index = X_train.columns,columns=['Importances']).sort_values('Importances',ascending=False)


# Plot the impurity-based feature importances of the forest
plotdata = pd.DataFrame(feature_importances,index=feature_importances.index)
plotdata.plot.bar(figsize=(8,5),title="Feature importance using Random Forest Algorithm")



feature_importances
```

Means absolute error of Linear and random forest regression, to find the difference between both the models

```
models = []
models.append(('Multiple Linear Regression', LinearRegression()))
models.append(('Random Forest Regression', RandomForestRegressor(max_depth=10, random_state=21)))

# evaluate each model in turn
rmse_scores = []
names = []
m_ab_ers = []
m_ab_er_pers = []
r_scores= []

for name, model in models:

  model.fit(X_train,Y_train)

  #7 Predicting the Test set results:
  y_pred = np.round(model.predict(X_test),decimals=2)
  rmse = np.sqrt(mean_squared_error(Y_test,y_pred))
  m_ab_er=mean_absolute_error(Y_test, y_pred, multioutput='raw_values')
  m_ab_er_per=np.round(np.mean(np.abs((Y_test - y_pred) / Y_test)) * 100,2)

  r_score=r2_score(Y_test, y_pred)

  #storing value in array to be used in data frame
  r_scores.append(np.round(r_score*100,2))
  rmse_scores.append(np.round(rmse,2))
  names.append(name)
  m_ab_ers.append(m_ab_er)
  m_ab_er_pers.append(m_ab_er_per)

df=pd.DataFrame({'Algorithm': names,'RMSE': rmse_scores,'R score':r_scores, 'Mean abs error': m_ab_ers,
                 'Mean abs error %':m_ab_er_pers})

df
```

Statistical significance (comparison of MLR and RF for respective error in final predictions)

```
from scipy import stats

print("Mean error of LR and RF respectively",np.mean(np.absolute(difference['LR error'])),
      np.mean(np.absolute(difference['RF error'])),"\n")
#print(np.mean(difference['LR error']),np.mean(difference['RF error']))

print(stats.ttest_rel(difference['LR error'],difference['RF error']))
```

APPENDIX B

ARTEFACTS AND THEIR STRUCTURES IN DIRECTORIES

The powerpoint(.ppt) presentation, and artefacts for data & python programming codes are submitted with following directory structures.

Directories & Files structure -

```
/
└── artefacts
    ├── data extract screens - Screen shots for data sources
    ├── final data - final dataset prepared from original data sources
    ├── original resources - original data sources as extracted and downloaded
    └── Python files - .py and .ipynb files for dissertation python codes, and dataset
```

Structure for presentation -

```
/
└── Presentation
    ├── Presentation - a power point file with presentation work
    └── Slide show - A direct slide show version of ppt file
```