International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 5 | ISSN : 2456-3307

Weather Forecasting using R

Pritam Sah, Prof. Jayant Adhikari, Prof. Rajesh Babu
Tulsiramji Gaikwad Patil College of Engineering and Technology, Wardha Road, Nagpur, Maharashtra, India

ABSTRACT

In this project, we are using the public data and mining the useful pattern so as to get the appropriate results at the end. R programming is a language for statistical computing and can be used to graphically display the output. Using R programming following statistical works can be done such as linear and nonlinear modelling, clustering, graphical representation of data techniques, classification.. One of the advantage of R's is the way in which well graphically designed graphs on-quality plots can be produced, mathematical symbols and formulae. In this project, we are using 3 algorithms i.e Logistic Regression, Decision Tree, Random Forest to forecast whether Rainfall may occur or not. By using 3 algorithms we are trying to increase the accuracy of weather forecasting.

Keywords : R programming, Logistic Regression, Decision Tree, Random Forest, weather forecasting.

I. INTRODUCTION

Data analysis is the first process which is to be done. Data analysis means cleaning the data, transforming it into the useful data, modeling and extracting the useful data from the dataset.

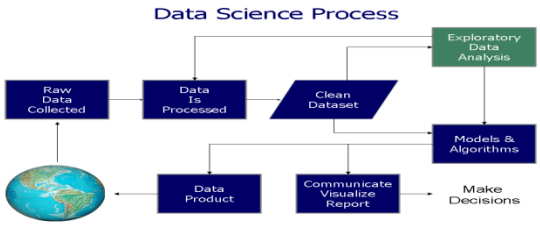

Fig 1. Phases of Data Science Process

In the first phase of data science process raw data is collected that can be public data which can be used to collect relevant data that has to be analyzed. In second phase i.e. Data Processing and Cleaning, the blank values present in the dataset or missing values is to be removed and data is further cleaned. In next stage i.e. Exploratory data analysis, the refined code i.e. data analysis scripts is written to analyze and obtain useful information in the data. In models and algorithm phase, the different algorithm which has to be applied on public data to get the output is applied. Algorithm is defined and applied on the data by giving cleaned public data as input to algorithm. In this project we are using Logistic Regression, Decision Tree and Random Forest. In next phases the output is represented in the graphical format and the written reports are created at the end. Weather forecasting is done to save lives, decreasing property damages and limiting the crop damage. Forecasts are often utilized to make many important decisions on a daily basis. Recently (June 2017), Maharashtra farmer complains against IMD (India Meteorological Department) for wrong rain forecast. Annoyed with inaccurate weather predictions, a farmer from Beed district has filed a complaint against Colaba and Pune weather stations of the India Meteorological Department (IMD) for their 'misleading' rain forecast. This is the first time such a complaint has been filed against IMD. Complaint was filed in the

Dindrud police, Thavre in which it was written that the farmers which owns an farm land at Anandgaon, Beed district has completed sowing the seeds on farm land in June by believing on IMD forecast . The IMD is the weather forecasting agency that had predicted heavy rainfall will occur during June and July, but unfortunately no heavy rainfall occured in the month of June and July. This false prediction result in the loss of crop, money etc.. A Packages which we are using in our project are : knitr, caret, gmodels, lattice, ggplot2, ROCR, corplot. Knitr package provides a general purpose tool for dynamic report generation in R using Literate Programming technique. Caret (Classification and regression training) package contains functions that streamlines the process for creating models used for prediction. It contains the features for splitting the dataset, processing the dataset, pattern matching feature, re-sampling, variable importance detection. Gmodels package contains various R programming tools for model fitting. Lattice package is a powerful and elegant high level data visualization system. Ggplot2 package is a system for declaratively creating graphics. ROCR contains graphs, sensitivity, specificity curves, lift charts, and precision/recall plots. Corrplot package contains graphical display of a correlation matrix, confidence interval and some algorithm to do matrix reordering. In this project we will use the public data to analyze weather forecasting. So, basically we are going to take public dataset of 1 year having fields like date, location, min temp, max temp, rainfall, evaporation, sunshine, wind gust air, wind gust speed, etc. and by using different functions and by analyzing the data we will predict the weather for next coming days.

Logistic Regression

Logistic regression is a type of regression model in which response variable i.e. dependent variable has a categorical values such as True/False or 0/1. Logistic Regression calculates the probability of binary responses which can be used as response variable based on the mathematical equation relating with the predictor variables. Mathematical equation for logistic regression is "y=1/(1+e^-(a+b1*1+ b2 *2+b3*3+b4*

4+...)", where, y is response variable, x is the predictor variable, a and b are the coefficient which are numeric constant. The function which is used to create the regression model is glm() function. Syntax for glm() is glm(formula,data,family) as shown in fig. 2 where **formula** is the symbol presenting the relationship between the variables, **data** is the set giving the values of the variables and family is the R object to give the details of the model and its value is binomial for logistic regression.

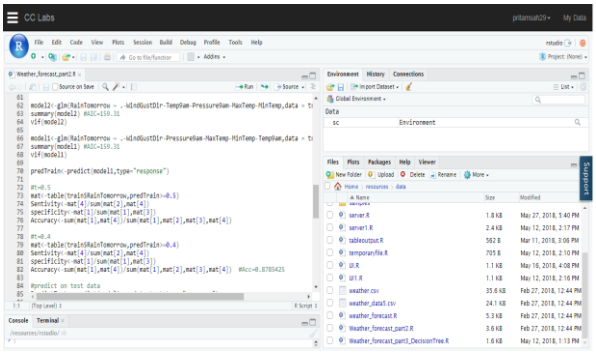

Fig. 2 Logistic Regression Implementation

Decision Tree

The decision tree is a graph to depict choices and their respective result which is in the form of tree. Nodes in the graph depict an choice or event and edges in tree depict the conditions or the decision rules. By mapping the observations about the data, the target value is predicted. Classification as well as regression can be done with the decision tree. Some of the application of decision tree is Direct Marketing, Retention of customer, Detection of fraud activities, medical problem diagnosis. If in a tree every node has a two child then that tree is called as binary tree. Rpart() is used to implement the decision tree in which one model is created by giving the cleaned public dataset as a input as shown in fig 3.

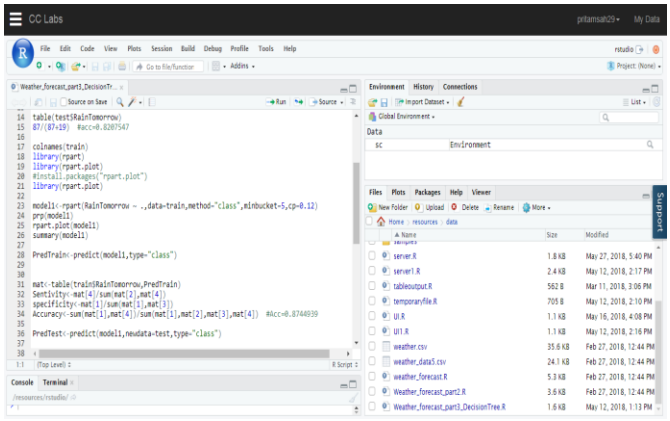

Fig 3. Implementation of Decision Tree

Random Forest

In random forest, large number of decision trees are generated and every observation is fed into each decision tree. The common outcome of each observation will be used as the final output. In each iteration, a new observation is fed into all the trees and then taking a majority vote for every classification model. Error estimate is created for every case which were not used while building the tree. This is known as **OOB (Out-of-bag)** error estimate which is illustrated in the form of percentage. The function used to implement Random Forest is randomForest(). Syntax for randomForest() is : randomForest(formula,data) as shown in fig 4, where **formula** is a formula describing the predictor and responsive varibales and **data** is the name of the data set which is being used.

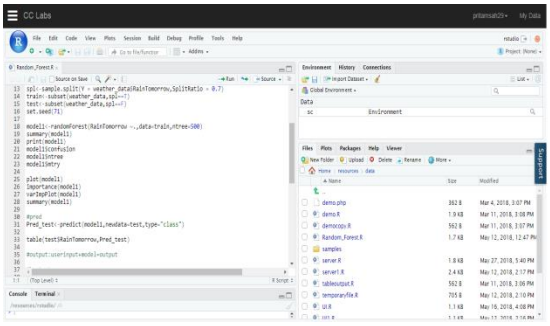

Fig 4. Implementation of Random Forest

II. Methodology

In public data our dependent variable is "RainTomorrow" which tells whether rain will come in coming day or not. Firstly, we will install all the packages required to implement code like knitr, caret, gmodels, etc. To use the installed library we need to write "library(library name)". Then we will take our public data which is in csv format into one variable. Syntax for reading csv file is "weather_data <- read.csv("weather.csv", header = TRUE, sep = ",", stringsAsFactors = TRUE)". We can get the summary of the dataset as shown in Fig 5 below:

```
> weather_data<-read.csv("weather_data5.csv")
> summary(weather_data)
    MinTemp          MaxTemp        Evaporation       Sunshine       WindGustDir  WindGustSpeed
 Min.   :-5.300   Min.   : 7.60   Min.   : 0.200   Min.   : 0.000   NW     : 72   Min.   :13.00
 1st Qu.: 2.400   1st Qu.:15.10   1st Qu.: 2.400   1st Qu.: 5.900   NNW    : 42   1st Qu.:31.00
 Median : 7.500   Median :19.80   Median : 4.200   Median : 8.700   E      : 36   Median :39.00
 Mean   : 7.357   Mean   :20.61   Mean   : 4.565   Mean   : 7.926   WNW    : 35   Mean   :40.06
 3rd Qu.:12.500   3rd Qu.:25.50   3rd Qu.: 6.400   3rd Qu.:10.600   ENE    : 30   3rd Qu.:46.00
 Max.   :20.900   Max.   :35.80   Max.   :13.800   Max.   :13.600   ESE    : 23   Max.   :98.00
                                                                    (Other):115
 WindSpeed9am     WindSpeed3pm     Humidity9am      Humidity3pm      Pressure9am      Pressure3pm
 Min.   : 0.000   Min.   : 0.00   Min.   :36.00   Min.   :13.00   Min.   : 996.5   Min.   : 996.8
 1st Qu.: 6.000   1st Qu.:11.00   1st Qu.:64.00   1st Qu.:32.00   1st Qu.:1015.2   1st Qu.:1012.7
 Median : 7.000   Median :17.00   Median :72.00   Median :43.00   Median :1020.0   Median :1017.1
 Mean   : 9.683   Mean   :18.02   Mean   :71.87   Mean   :44.45   Mean   :1019.5   Mean   :1016.7
 3rd Qu.:13.000   3rd Qu.:24.00   3rd Qu.:80.00   3rd Qu.:55.00   3rd Qu.:1024.4   3rd Qu.:1021.4
 Max.   :41.000   Max.   :52.00   Max.   :99.00   Max.   :96.00   Max.   :1035.7   Max.   :1033.2

    Cloud9am        Cloud3pm         Temp9am          Temp3pm       RainTomorrow
 Min.   :0.000   Min.   :0.000   Min.   : 0.10   Min.   : 5.10   No :289
 1st Qu.:1.000   1st Qu.:1.000   1st Qu.: 7.70   1st Qu.:14.30   Yes: 64
 Median :4.000   Median :4.000   Median :12.60   Median :18.60
 Mean   :3.912   Mean   :4.028   Mean   :12.44   Mean   :19.27
 3rd Qu.:7.000   3rd Qu.:7.000   3rd Qu.:17.00   3rd Qu.:24.00
 Max.   :8.000   Max.   :8.000   Max.   :24.70   Max.   :34.50
```

Fig 5. Summary of the dataset

We need to clean our dataset i.e. removing NULL values, removing unwanted field. To remove unwanted fields we can use "-c" for eg. "weather_data2<-subset(weather_data,select = -c(Date, Location, RISK_MM, Rainfall, RainToday))". To check the NULL values there is a function as "is.na()" which will return the NULL values in every field present in the dataset as shown in Fig 6.

```
> cols_Na<-apply(weather_data2,2,function(x){ sum(is.na(x))})
> cols_Na
      MinTemp       MaxTemp   Evaporation      Sunshine   WindGustDir WindGustSpeed
            0             0             0             3             3             2
   WindDir9am    WindDir3pm  WindSpeed9am  WindSpeed3pm  Humidity9am  Humidity3pm
           31             1             7             0             0             0
  Pressure9am   Pressure3pm      Cloud9am      Cloud3pm       Temp9am       Temp3pm
            0             0             0             0             0             0
 RainTomorrow
            0
```

Fig 6. Finding NA's values

Then we apply chi-sqaure test. Chi-square is used to find out the statistical importance between the differences of sub-nodes and parent node. Chi-square works on categorical variable (e.g. "Yes" or "No"), it also performs splits. If higher is the value of Chi-Sqaure then higher will be the significance of

differences ofn parent node and sub nodes. Formula for chi-sqaure is (Actual-Expected)^2/Expected)^1/2. Chi-sqaure develops CHAID (Chi-Sqaure Automatic Interaction Detector).

After applying Chi-Square test, we will split the data into 70% train data and 30% test data. Now we will create a Logistic Regression model using glm() function as "model1<-glm(RainTomorrow ~ .-WindGustDir-Pressure9am-MaxTemp-MinTemp-Temp9am,data = train,family = "binomial")" which will give AIC i.e. **Akaike** information criterion value which should be more so as to have more accuracy. To find out the accuracy i.e. True to True value and False to False value we create a matrix as "mat<-table(train$RainTomorrow,predTrain>=0.5)" from which we can get a accuracy value as "Accuracy<-sum(mat[1],mat[4])/sum(mat[1],mat[2],mat[3],mat[4])" as shown in figure 7 below:

```
> predTrain<-predict(model1,type="response")
> #t=0.5
> mat<-table(train$RainTomorrow,predTrain>=0.5)
> Sentivity<-mat[4]/sum(mat[2],mat[4])
> specificity<-mat[1]/sum(mat[1],mat[3])
> Accuracy<-sum(mat[1],mat[4])/sum(mat[1],mat[2],mat[3],mat[4])
> Accuracy
[1] 0.8825911
```

Fig 7. Output of Accuracy variable after applying table()

We will find for the highest accuracy as possible. Now we will apply Decision Tree algorithm to our train data. In this we will also split the data into 70% train data and 30% test data using split() as shown in Fig 8.

```
library(caTools)
spl<-sample.split(Y = weather_data$RainTomorrow,SplitRatio = 0.7)
train<-subset(weather_data,spl==T)
test<-subset(weather_data,spl==F)
```

Fig 8. Splitting of data in train and test data

For implementation of Decision Tree algorithm we can use rpart() as "model1<-rpart(RainTomorrow ~ .,data=train,method="class",minbucket=5,cp=0.12)".

As done in Logistic Regression we can also get a accuracy of the model as shown in Fig. 7. The last algorithm which we are going to implement is Random Forest which is more efficient algorithm than Logistic Regression and Decision Tree. For implementing Random Forest algorithm we need to use "randomForest()" as "model1<-randomForest(RainTomorrow~.,data=train,ntree=500)" and the same procedure will be followed to check the accuracy of the algorithm as shown in Fig 7.

In final implementation as shown in Fig 9 and Fig 10, we need to upload the csv file and then that file is used as input to model in the next screen and we need to select the dependent and independent variable to get the accuracy and the matrix value. In matrix value we get the False to False value and True to True value to check how accurate True to True or False to False is given by the model.

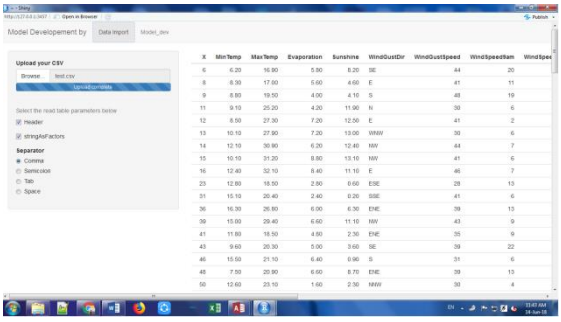

Fig 9. Final Implementation

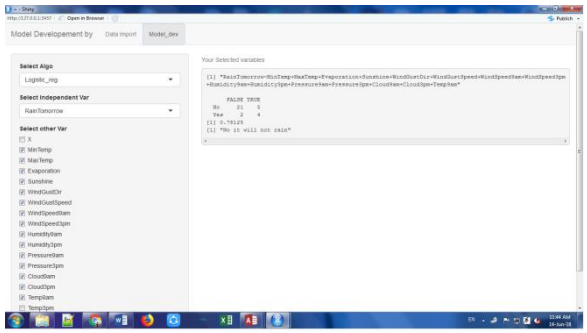

Fig 10. Final Implementation with Accuracy value

III. CONCLUSION

Thus, we have cleaned, find useful pattern and data fields from the public dataset. 3 algorithms have been applied on the public dataset and also the accuracy is being increased by using more efficient algorithm such as Random Forest. Accuracy of each and every algorithm is calculated. The output is efficient as we have applied 3 algorithms and compared there results. In future, more efficient algorithm can be used to predict the accurate weather forecasting. This project can be useful for the farmer so that they should not face any type of losses due to incorrect weather prediction in future.

IV. REFERENCES

[1]. Sanjay Chakraborty, N.K.Nagwani, Lopamudra Dey "Weather Forecasting using Incremental K-Means clustering", in CiiT International Journal of Data Mining & Knowledge Engineering, May 2012

[2]. S.Chakraborty and N.K.Nagwani ,"Performance evaluation of incremental K-means clustering algorithm ", in IFRSA International Journal of Data Warehousing & Mining (IIJDWM), vol.1, 2011,pp-54-59

[3]. Susmita Datta and Somnath Datta, "Comparisons and validation of statistical clustering techniques for microarray gene expression data" Bioinformatics, vol. 19, pp.459–466, 2003.