# FOREST FIRE PREDICTION WITH MACHINE LEARNING

**K.G. Shangavi [1], K.K. Sindu[2], S. Gomathi[3]**

[1] I MSc Data Analytics, Department of Computer Science (PG)

PSGR Krishnammal College for Women, Coimbatore, India

[2] I MSc Data Analytics, Department of Computer Science (PG)

PSGR Krishnammal College for Women, Coimbatore, India

[3] Assistant Professor, Department of Computer Science (PG)

PSGR Krishnammal College for Women, Coimbatore, India.

## ABSTRACT

Data mining brings understandings, outlines, and descriptive and predictive representations from the large amounts of data available today in many organizations. We have many tools for data mining. One of those tools is rattle which runs on R programming language. In this paper, we used rattle to analyze forest fires which occurred in different regions during different periods of time. We can study whether rattle is an efficient tool by analyzing its time taken for delivering the result, error rate and much more.

Keywords: Data mining, machine learning, rattle, decision tree, error matrix

## INTRODUCTION

Data mining is turning the data into information. Every industry collects data and applies it for feeding new knowledge. Data mining is very essential nowadays because of the volume of data available, commonly in the gigabytes and terabytes and fast approaching the petabytes. It is also characterized by the complexity of that data, both in terms of awaiting discovery in the data and the data types available today, including text, image, audio, and video. The business environments are also rapidly changing, and analyses need to be performed regularly and models updated regularly to keep up with today's dynamic world. There are many tools emerge for data mining. One of those is R. R is the very powerful tool in data mining. But R doesn't operate on graphical user interface. To overcome this, rattle package provides GUI specifically for data mining using R. It also delivers rapid result and a stepping stone toward considering R as a programming language for data analysis [1][2].

## WORKING WITH RATTLE

The Rattle interface is designed as a simple interface for data mining. The standard process is

to click the each tab for doing the corresponding actions. For any tab, we choose from the options and then click the Execute button to perform the appropriate tasks. We use decision tree and forest options for analyzing the dataset [3].

**Table 1. Data description**

| ATTRIBUTES | DATA TYPE | DESCRIPTION |
|---|---|---|
| MONTH | DATE | Month in which forest fire occurred |
| DATE | DATE | Date in which forest fire occurred. |
| FINE FUEL MOISTURE CODE(FFMC) | FLOAT | Numeric rating of the moisture content of litter and other cured fine fuels |
| DUFF MOISTURE CODE(DMC) | FLOAT | Numeric rating of the average moisture content of loosely compacted organic layers of moderate depth |
| DROUGHT CODE(DC) | FLOAT | Numeric rating of the average moisture content of deep, compact organic layers |
| INITIAL SPREAD INDEX(ISI) | FLOAT | Numeric rating of the expected rate of fire spread |
| TEMPERATURE | FLOAT | Temperature in the affected area |
| RELATIVE HUMIDITY(RH) | FLOAT | Amount of water vapour present in air |
| WIND | FLOAT | Movement of air in the affected area |
| RAIN | FLOAT | Percent of rain in the affected area |

**DECISION TREE**

A tool which uses a tree-like graph or branching method to formulize the values of data is called decision tree. It is one of the

analytical displaying approaches used in statistics, data mining and machine learning.

Table 2. Decision tree rules

Classification trees are the tree models where the target variable can take a finite set of values. In tree structures, class labels are represented by leaves and conjunctions of features that lead to those class labels are represented by branches[3][5].

*rpart(formula = X ~ ., data = crs$dataset[crs$train, c(crs$input,*

*crs$target)], method = "class", model = TRUE, parms = list(split = "information"), control = rpart.control(usesurrogate = 0, maxsurrogate = 0))*

**TREE AS RULES:**

**Error Matrix**

Error matrix reports whether the predicted value is true or false[3].

| | |
|---|---|
| Rule number: 11<br>[X=4 cover=21<br>(38%) prob=6.00]<br>rpart< 7.5<br>rpart>=3.5<br>rpart< 6.5 | Rule number: 10<br>[X=1 cover=14<br>(25%) prob=3.00]<br>rpart< 7.5<br>rpart>=3.5<br>rpart>=6.5 |
| Rule number: 3<br>[X=4 cover=8<br>(15%) prob=2.00]<br>rpart>=7.5 | Rule number: 4<br>[X=7 cover=12<br>(22%) prob=0.00]<br>rpart< 7.5<br>rpart< 3.5 |

Error matrix for the Decision Tree model on forestfires_test_score_idents.csv [validate] (counts)

*Predicted 1 2 3 4 5 6 7 8 9 Error*

*Actual 1 0 0 0 0 0 0 0 0 0 NaN*

*2 1 0 0 0 0 0 2 0 0 100*

*3 0 0 0 0 0 0 1 0 0 100*

*4 0 0 0 2 0 0 0 0 0 0*

*5 0 0 0 2 0 0 1 0 0 100*

*6 0 0 0 1 0 0 0 0 0 100*

*7 0 0 0 1 0 0 0 0 0 100*

*8 0 0 0 0 0 0 0 0 0 NaN*

*9 0 0 0 0 0 0 0 0 0 NaN*

**FOREST**

Forests, known as random decision forests, can be used to build analytical models for both classification and deterioration problems. Collective methods use multiple learning models to gain better predictive results. In the case of a random forest, error can be estimated which cannot be estimated in decision tree [5][6].

*randomForest(formula = as.factor(X) ~ .,*

*data = crs$dataset[crs$train, c(crs$input, crs$target)],*

*ntree = 500, mtry = 1, importance = TRUE, replace = FALSE, na.action =*

*randomForest::na.roughfix)*

**FOREST MODEL RULES**

Table 3: Forest model rules

| | TREE | FOREST |
|---|---|---|
| *TIME TAKEN* | 0.04sec | 0.18sec |
| ERROR RATE | 80% | 90.91% |
| Tree 1 Rule 1 Node 4 Decision 5<br>1: rpart <= 5<br>2: rpart <= 2.5 | | Tree 1 Rule 2 Node 8 Decision 2<br>1: rpart <= 5<br>2: rpart > 2.5<br>3: rpart <= 3.5 |
| Tree 1 Rule 3 Node 9 Decision 2<br>1: rpart <= 5<br>2: rpart > 2.5<br>3: rpart > 3.5 | | Tree 1 Rule 4 Node 6 Decision 4<br>1: rpart > 5<br>2: rpart <= 6.5 |
| Tree 1 Rule 5 Node 10 Decision 1<br>1: rpart > 5<br>2: rpart > 6.5<br>3: rpart <= 7.5 | | Tree 1 Rule 6 Node 11 Decision 4<br>1: rpart > 5<br>2: rpart > 6.5<br>3: rpart > 7.5 |

**Error Matrix**

Error matrix reports whether the predicted value is true or false [3].

Error matrix for the Random Forest model on forestfires_test_score_idents.csv [test] (counts):

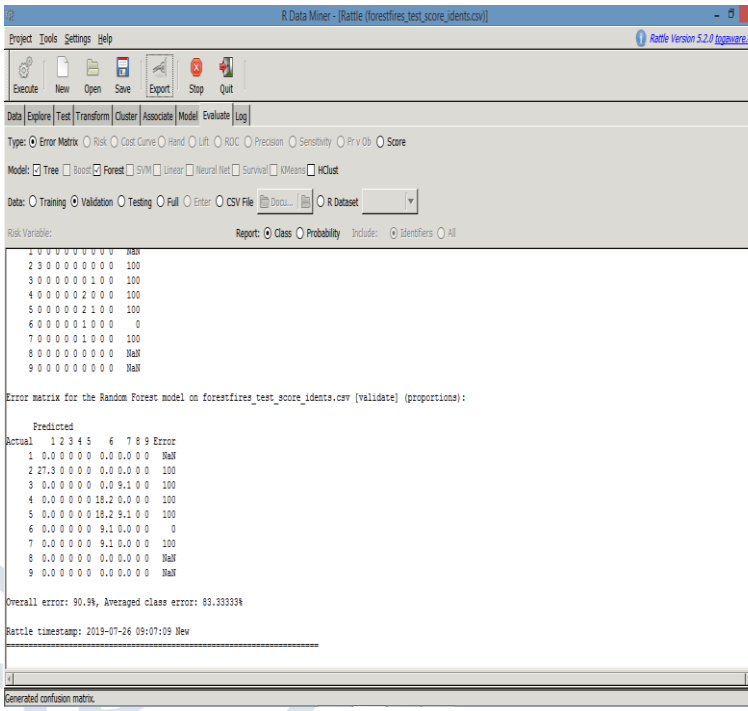
| *Predicted* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *Error* |
|---|---|---|---|---|---|---|---|---|---|---|
| *Actual* *1* | *0* | *1* | *0* | *0* | *0* | *0* | *0* | *0* | *0* | *100* |
| *2* | *1* | *0* | *0* | *0* | *0* | *0* | *0* | *0* | *0* | *100* |
| *3* | *2* | *0* | *0* | *0* | *0* | *0* | *0* | *0* | *0* | *100* |
| *4* | *0* | *0* | *0* | *0* | *0* | *2* | *1* | *0* | *0* | *100* |
| *5* | *1* | *0* | *0* | *0* | *0* | *0* | *0* | *0* | *0* | *100* |
| *6* | *0* | *0* | *0* | *0* | *0* | *0* | *1* | *0* | *0* | *0* |
| *7* | *2* | *0* | *0* | *0* | *0* | *0* | *1* | *0* | *0* | *100* |
| *8* | *0* | *0* | *0* | *0* | *1* | *0* | *0* | *0* | *0* | *100* |
| *9* | *0* | *0* | *0* | *0* | *0* | *0* | *0* | *0* | *0* | *NaN* |

Tabl4. Time taken matrix

*Table 5. Overall result*

| | Overall Error | Average |
|---|---|---|
| Decision tree model | 81.8% | 83.33% |
| Random Forest model | 92.3% | 87.5% |

ERROR MATRIX




## TEST RESULTS

PARAMETER:

- Hypothesized Ratio: 1
- Numerator df: 8
- Denumerator df: 8

SAMPLE ESTIMATES:

- Ratio of Variances: 0.6894

STATISTIC:

- F: 0.6894

P VALUE:

- Alternative Two-Sided: 0.6111
- Alternative Less: 0.3055
- Alternative Greater: 0.6945

CONFIDENCE INTERVAL:

- Two-Sided: 0.1555, 3.0561
- Less: 0, 2.3701
- Greater: 0.2005, Inf

## RESULTS AND DISCUSSION

In day to life the generation of data for every second is tremendous.the necessary of machine learning tools also increases.In this paper we analysed RATTLE in R programming with forest fire data. On evaluating the error matrix we found decision tree model is more efficient than the random forest model.

## CONCLUSION

By the results, we can conclude that decision tree is more efficient than random forest by means of both time taken and error rate. Rattle is a free software, we can update it without cost and it is easy to use even for non-programmers. So it continues to undergo development, extending its arms in data mining with friendly gesture.

## REFERENCES

[1]. Williams, Graham J. "Rattle: a data mining GUI for R." *The R Journal* 1.2 (2009): 45-55.

[2]. Williams, G. J. (2009). Rattle: a data mining GUI for R. *The R Journal*, *1*(2), 45-55.

[3]. uazzelli, A., Zeller, M., Lin, W.C. and Williams, G., 2009. PMML: An open standard for sharing models. *The R Journal*, *1*(1), pp.60-65.

[4]. Aggarwal, Sandeep. "Data Mining Tools: A Comparative and Analytical Study."

[5]. Jovic, A., Brkic, K., & Bogunovic, N. (2014, May). An overview of free software tools for general data mining. In *2014 37th*

*International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 1112-1117). IEEE.

[6]. Cortez P. Data mining with neural networks and support vector machines using the R/rminer tool. InIndustrial Conference on Data Mining 2010 Jul 12 (pp. 572-583). Springer, Berlin, Heidelberg.

[7]. Thornton, Heidi R., Jace A. Delaney, Grant M. Duthie, Brendan R. Scott, William J. Chivers, Colin E. Sanctuary, and Ben J. Dascombe. "Predicting self-reported illness for professional team-sport athletes." *International journal of sports physiology and performance* 11, no. 4 (2016): 543-550.

[8]. Sherafatian, Masih, and Fateme Arjmand. "Decision tree-based classifiers for lung cancer diagnosis and subtyping using TCGA miRNA expression data." *Oncology Letters* 18, no. 2 (2019): 2125-213