Mining Weather Data Using Rattle

[1] Venkata Abhishek Bavirisetty, [2] Avinash Ankireddy, [3] Prasad Seemakurthi, [4] K.V.N. Rajesh

[1, 2, 3, 4] Department of Information Technology, Vignan's Institute Of Information Technology, Duvvada, Visakhapatnam-530049, A.P., India

Abstract

Data mining is the efficient discovery of necessary data from the group of different heterogeneous databases. But there is a presence of inconsistencies in that discovered data. Due to this many organizations are facing problems and fail to obtain desired results. In order to reduce these problems we have used the inconsistency removal, redundancy methods and finally capture the graphs. All these processes are done with computationally efficient and scalable R code, which mainly deals with the data-mining tasks, such as frequent pattern discovery and classification and so on. In this paper we present some of our research on developing code for representing patterns and building predictive models of graphs for the WEATHER dataset in R console applet. Finally this paper will end up with performing three efficient tasks, which are mandatory for any data set, which are named as pre-processing, principal component analysis and classification and prediction. All the above tasks are generated through R language with the help of R console applet.

Keywords: *Preprocessing, Principal Component Analysis (PCA), Classification and Prediction(C&P).*

1. Introduction

In the software organizations almost fifty percent of the projects are getting failed. This may be due to many reasons. So we have taken one of them (i.e) due to providing of the inconsistent data as input. So in order to overcome this particular problem we have used a technology called Rattle which does all its work in R language by the help of R console applet and further proceed by conducting three stages namely Data-preprocessing,

Principal Component Analysis and finally with Classification and Prediction. RATTLE: Rattle (the R Analytical Tool To Learn Easily) provides a simple and logical interface for data mining tasks [1]. R language, was developed at Bell Laboratories by Rick Becker, John Chambers and Allan Wilks [2].

The main goal is to provide an interface that takes you through the basics of data mining, as well as illustrating the R code that is used to achieve this. R provides a wide variety of statistical and graphical techniques, including linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, and others.

Preprocessing Stage:

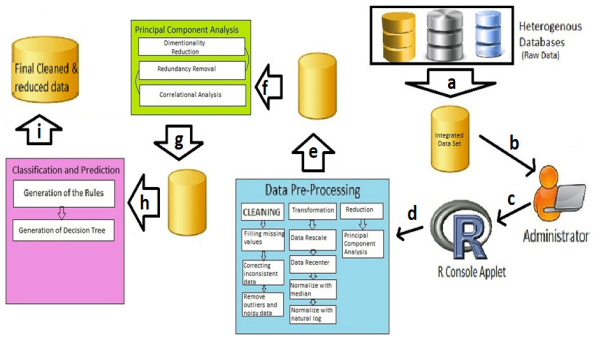
Data Cleaning: It is mainly applied to remove noise and correct inconsistencies in the data[3]. This stage will be responsible for conduction the following:

- Removal of the inconsistent values.
- Replacing of the NA values.

Data Transformation: In this, the data is transformed or consolidated into forms appropriate for mining[3]. Data Transformation can be involve the following:

- Normalization
- Data re-center
- Smoothing
- Data Scaling

Data Reduction: In this, we will reduce the dataset dimensions or divide the datasets in to subsets number according to the user requirement [3]. After this we will remove the redundancies and conduct the correlation analysis before & after the removal. This is observed in graphs.



a)integrating b)accessing integrated data c)sending same data to R d)sending data for cleaning e)update database with preprocessing result f)sending cleaned data to PCA g) update database with PCA result h) sending PCA result to C&P i) sending final result to database

Figure 1: System Architecture

Figure 1 describes about all the operations conducted. First, we will integrate the dataset in to one from the heterogeneous datasets. Then user will use this integrated data, which consists inconsistent data.

So first we provide this data to the data preprocessing stage and clean the inconsistent data. Then the cleaned data is updated in to the data set.

Then the updated data set is given for Principal Component Analysis which reduces the data. Finally this reduced and cleaned data is used for prediction.

Design:

The class diagram is a static diagram. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application [4].

Classes, which represent entities with common characteristics or features [4]. These features include attributes, operations and associations.

Associations, which represent relationships that relate two or more other classes where the relationships have common characteristics or features [4].

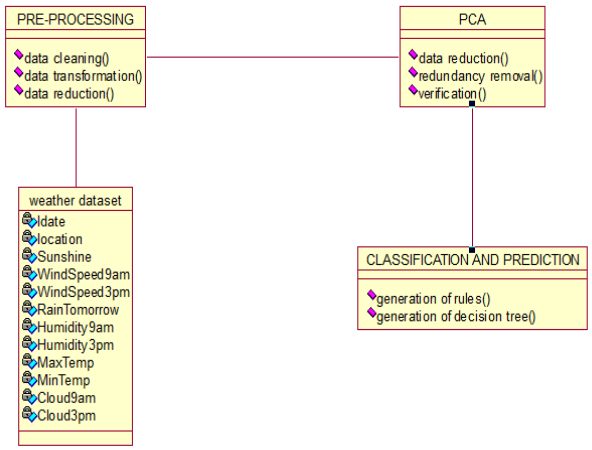


Figure 2: Class Diagram of the entire operation.

Each class will consist of three parts class name, attributes, and operations. In this we will be considering four classes and named them as weather dataset, preprocessing, PCA, classification & prediction. The attributes are mentioned in their corresponding classes if there is any absence of attributes then they are left empty. For the operations part also follows the same rule. For example when we consider the weather dataset class it consists only the attributes but there is no single operation, as the dataset won't perform any kind of operations so it is kept as empty.

No class can access the dataset directly because every raw dataset will be having inconsistencies and NA values etc. so in order to avoid we should first perform the data preprocessing. In the same way we cannot construct the tree without removing of the redundancies and other kind of unnecessary data so we should perform first preprocessing then next go with PCA then finally classification and prediction.

Sequence Diagram describes about the interactions between the classes. These interactions are modeled as exchange of messages. These diagrams focus on classes and the messages they exchange to accomplish some desired behavior. Sequence diagrams are a type of interaction diagrams. Sequence diagrams contain the following elements:

Class roles, which represent roles that objects, may play within the interaction [5].

Lifelines, which represent the existence of an object over a period of time [5].

Activations, which represent the time during which an object is performing an operation [5].

Messages, shows the communication between objects [5].

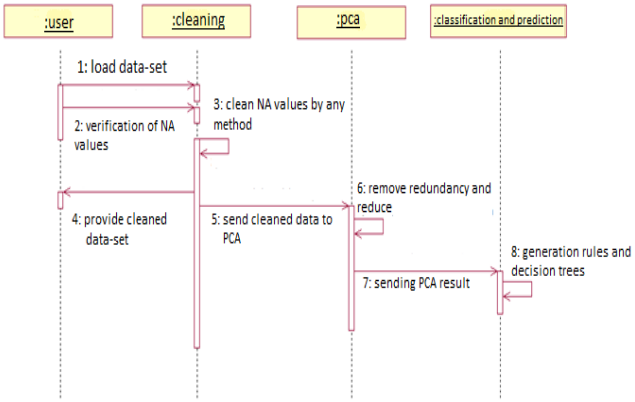


Figure 3: Sequence Diagram of operations conducted in all the three tasks.

The above sequence diagram describes about the three stages, which are conducted in this. In this first the user loads the dataset and verifies whether it is cleaned or not. If not it is cleaned by the techniques that will be mentioned and then process this data to the next stage (pca). Then in the pca stage it is examined whether the data consists of any redundancies and also examine whether there is any possibility for reducing the data. Then finally process the result to the next stage for construction of the rules and decision trees.

2. Preprocessing

Data mining delivers insights, patterns, and descriptive and predictive models from the large amounts of data available today in many organizations [1]. The Rattle (R Analytical Tool to Learn Easily) package provides a graphical user interface specifically for data mining using R. It has been developed specifically to ease the transition from basic data mining, as necessarily offered by GUIs, to

sophisticated data analyses using a powerful statistical language.

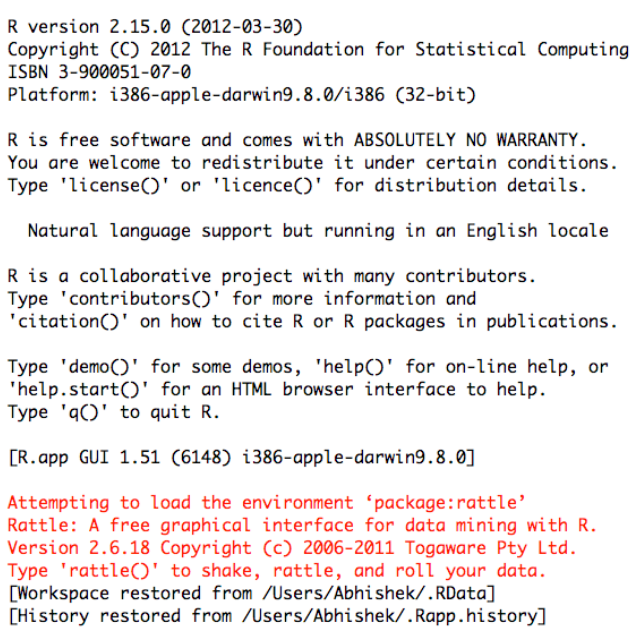


Figure 4: Sample view of the R console applet.

2.1 Data Cleaning

Before performing any kind of operations on the data set first we must load the dataset and store in any of the location by giving some name and is mandatory to have the dataset in ".csv" or ".dat". As our data set consists of rows and columns which resembles the table format we use ".csv" format which we can generate with the Microsoft excel. The syntax for loading or reading the dataset is as follows:

sri07<-read.csv("/Users/Abhishek/Downloads/weather.csv")

Here the weather.csv is the name to the dataset, /Users/Abhishek/Downloads is the path of the weather data-set and finally the read.csv is the actual command used for reading the dataset (weather.csv) and this is stored in a variable named sri07. While proceeding through first stage we should consider the basic details of the dataset like number of columns, rows and names of the columns [6]. This can be achieved by the following syntax:

dim(sri07)
Output: [1] 366 24
names(sri07)
Output:

[1] "Date" "Location" "MinTemp"
[4] "MaxTemp" "Rainfall" "Evaporation"
[7] "Sunshine" "WindGustDir" "WindGustSpeed"
[10] "WindDir9am" "WindDir3pm" "WindSpeed9am"
[13] "WindSpeed3pm" "Humidity9am" "Humidity3pm"
[16] "Pressure9am" "Pressure3pm" "Cloud9am"
[19] "Cloud3pm" "Temp9am" "Temp3pm"
[22] "RainToday" "RISK_MM" "RainTomorrow"

Removing the tuples that contains the NA values:

The removal of NA values is done by the below syntax which gives the resultant dataset after removing all the tuples which consists of NA values from the source dataset [7]. Here the "sri07" is the source dataset and the "sri07.new" is the resultant.

sri07.new<-na.omit(sri07)

The following command will gives the values of minimum, 1st quartile, mean, median, 3rd quartile and the maximum values with the presence of NA values(for more clarification it can be applied before and after omitting the NA values). This may help to find out whether there are any outliers.

summary(sri07)

2.2 Replacement of NA with mean values

When we give the input for replacement of the NA value with the mean values it is recommended to give as a matrix it is mainly helpful while visualizing the result. The matrix syntax will convert the dataset or a part of dataset in to the matrix form. Here we are going to convert the variable to matrix form called "WindSpeed9am"(variable of the source dataset called sri07).

a<-as.matrix(sri07$WindSpeed9am)

- "a" is the sample name or location where the matrix form of the variable will be stored
- as.matrix is the actual command, which will convert the dataset to matrix form.
- To select a particular column of the dataset we will be using "$" symbol. Here we are considering the variable called WindSpeed9am from the dataset called sri07.

The mean syntax will replace all the NA values with the mean value (excluding the mean values) [8].

a[which(is.na(a)==TRUE)] = mean(a,na.rm = T)

2.3 Replacement of NA values with median values

For replacement with the median [8] value also should be given as a matrix which is mentioned as above

b<-as.matrix(sri07$Windspeed9am)

b[which(is.na(b)==TRUE)] = median(b,na.rm = T)

2.4 Replacement of the NA values with the global constant

Explains about the replacing of the NA value with the global constant, which is one of the methods in replacement techniques in "Data Cleaning" process.
The following syntax will be used to convert data set in to matrix form as said above:

a<-as.matrix(sri07$Windspeed9am)

The following syntax will be explaining about the conversion of all the NA value to global constant. Here in this process we have considered the global constant as "0" [9].

a[is.na(a)]<-0

2.5 Removal of outliers in the given dataset

Removal of the outliers [10] is one of the tasks in the data cleaning which in turn is one of the stages of the "Data Preprocessing". The purpose of this method is when there are any values, which are out range then that is considered to be an outlier. So due to this there may inconsistency in mean, median, max values so due to this reason we apply the method removal of outliers. It mainly considers about the values of mean and median values, it is recommended that both the values must be almost (or) absolutely equal.

Before going with the syntax's of the outliers we should first install the "outliers" package and load the package before performing the removal with the command[10]:

library(outliers)

The following syntax will give an array with all values False, except for the outlier (as defined in the package) documentation ("Finds value with largest difference between it and sample mean, which can be an outlier"). That value is returned as True.

outlier_tf= outlier(sri07$Humidity9am,logical=TRUE)

The following syntax will finds the location of the outlier by finding that "True" value within the "outlier_tf" array.

find_outlier= which(outlier_tf==TRUE,arr.ind=TRUE)

This creates a new dataset based on the old data, removing the one row that contains the outlier:

data_new = sri07[-find_outlier]

For clear means consider a sample data-set which consists of attribute values out of range

sampleweather<read.csv("/Users/Abhishek/Desktop/sampleweather.csv")

In the Sample weather data-set we have a values for WindGustSpeed and Humidity9am as 600 and 500 respectively which are out of range so these are consider to be outliers. These outliers are removed by applying the above R code.

2.6 Data Transformation

All the above-mentioned tasks are performed in R Data Miner, which is a part of R console applet which is available by giving Rattle() in the workspace. In R Data Miner selecting and generating the results implement all the operations[1]. But on the other side we are going to generate the code according to our desired result.

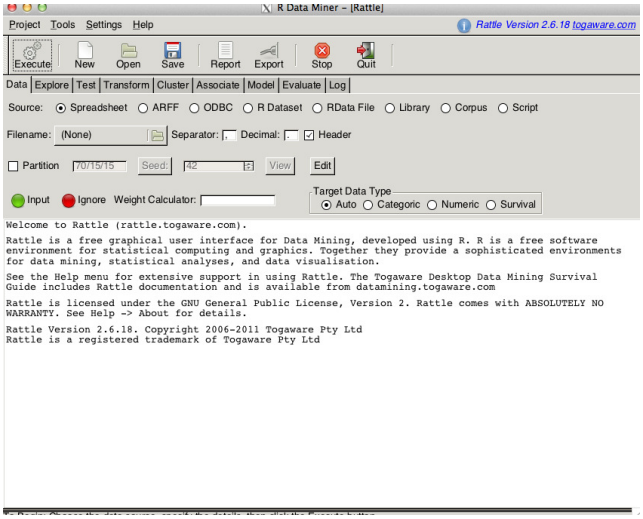


Figure 5: Sample View of the R Data Miner

We have two primary transformation techniques namely normalization and recoding.

Normalization involves

- Data re-center
- Data scaling [0-1]
- Normalizing with Median/MAD
- Normalizing with Natural log

Data re-center

This approach subtracts the mean value of the variable from each observation's value of variables (to re-center the variable) and then divide the values by their standard

deviation, which rescales the value back to a range within a few integer values around zero[11].

```
library(rattle)
sri07$RRC_Temp3pm <- scale(sri07$Temp3pm)

        sri07$RRC_Temp3pm
```

Data scaling [0-1]

Rescaling so that our data has a mean around zero might not be so intuitive for variables that are never negative [11]. Most numeric variables from the weather dataset naturally only take on positive values, including Rainfall and WindSpeed3pm. This approach simply recodes the data so that all values are between 0 and 1. Subtracting minimum value from the variable's value for each observation and then dividing by the difference between the minimum and maximum values do this.

```
library(reshape)
sri07$temp3_range<-rescaler(sri07$Temp3pm,"range")
```

2.7 Normalizing with Median/MAD

This option for recentring and rescaling our data is regarded as a robust (to outliers) version of the standard recenter option [11]. In this approach instead of using mean and standard deviation, we subtract the median and divide by median absolute deviation (MAD)

```
library(reshape)
data2_robust<-rescaler(data2,"robust")
```

2.8 Normalizing with Natural log

Logarithm transforms map a very broad range of (positive) numeric values into a narrower range of (positive) numeric values. The natural log function effectively reduces the spread of the values of the variable [11]. This is particularly useful when we have outliers with extremely large values compared with the rest of the population. Logarithms can use a so-called base with respect to which they do the transformation. We can use a base 10 transform to explain what the transform does.

The following R code is used to perform the transformation. We also recode any resulting "infinite" values (e.g., log(0)) to be treated as missing values.

```
sri07$RLG_Temp3pm <- log(sri07$Temp3pm)
sri07$RLG_Temp3pm[sri07$RLG_Temp3pm == -Inf] <-
NA
```

This is one of the stages of the "data preprocessing" which will reduce the dataset dimensions or decompose to number of subsets by the method of fragmentation and this stage consists of many techniques for the purpose of reducing the number of rows and columns.

Reduction by sub-setting the data:

```
newdata<-
subset(sri07,WindGustDir=="SE"&MinTemp>5,select=W
indSpeed3pm
```

3. Principal Component Analysis

There are several functions that calculate principal component statistics in R. Two of these are "prcomp()" and "princomp()"[12]. The PCA is independent of the procedure we follow but the final goal is to reduce the dimensions of the dataset which we has sent as the output. The object returned from the call to "prcomp()" has the following members:

- sdev
- rotation
- center
- scale

while the object returned from the call to "princomp()" has the following members:

- sdev
- loadings
- center
- scale
- n.obs
- scores

After taking the subset in to consideration it is mandatory to find out that if there are any redundancy values if present we must remove them. Before removing the redundancy values first we must check it they are really present in our subset. This can be known by the syntax called "duplicated".

```
duplicated(data2)
data2<-data2[!duplicated(data2),]
data2
```

4. Classification and Prediction

Working our way through the textual summary of the decision tree, we start with a report of the number of observations that were used to build the tree (i.e., 256):
Summary of the Decision Tree model for Classification [13].

```
n= 256
```

4.1 Tree Structure

A legend is provided to assist in reading the tree structure: node), split, n, loss, yval, (yprob)[13]

The legend indicates that a node number will be provided, followed by a split (which will usually be in the form of a variable operation value), the number of entities n at that node, the number of entities that are incorrectly classified (the loss), the default classification for the node (the yval), and then the distribution of classes in that node (the yprobs)[13]. The distribution is ordered by class and the order is the same for all nodes. The next line indicates that a "*" denotes a terminal node of the tree (i.e., a leaf node—the tree is not split any further at that node).

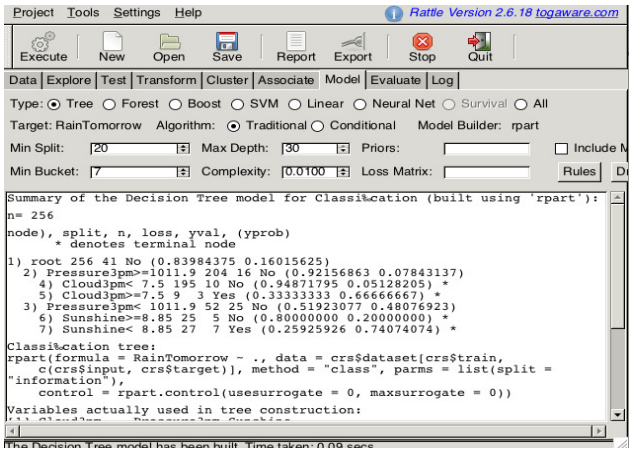


Figure 6: Generation of the Rules for Prediction

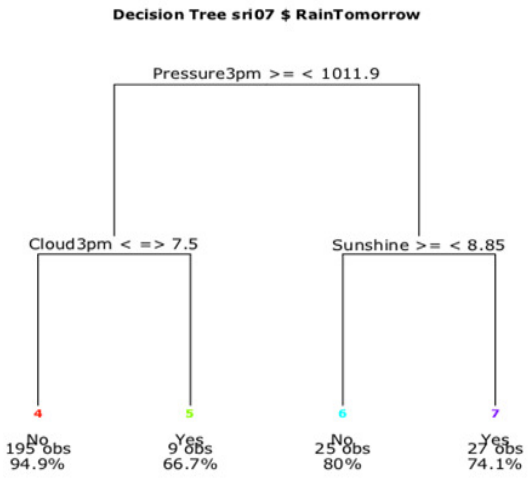


Figure 7: Final view of the decision tree after prediction

The above tree was produced from the above rules generated which were shown in the figure 6. According to those rules we will be predicting whether the rain will come or not.

5. Conclusion

Mining Weather Data using Rattle, will allows us to remove all the inconsistent data that are unnecessary. So that there will be accuracy and also able to obtain the desired results. This prediction may be helpful for the weather forecasting centers as it predicts weather conditions of a place. In our case we will be predicting whether the rain will appear tomorrow or not.

References

[1] Rattle: A Data Mining GUI for R by Graham J William http://journal.r-project.org/archive/2009-2/RJournal_2009-2_Williams.pdf
[2] http://stat.ethz.ch/R-manual/R- devel/doc/ html/about.html
[3] Data Mining Concepts and Techniques by Micheline Kamber and Jiawei Han
[4] www.tutorialspoint.com/uml/uml_class_diagram.html
[5] www.visual-paradigm.com/VPGallery/ diagrams/Sequence.html
[6] An Introduction to R by W. N. Venables, D. M.Smith and the R Development Core Team
[7] Sundar Dorai-Raj Mon Jun 13 19:19:27 CEST 2005.
[8] https://stat.ethz.ch/pipermail/r-help/2001-May/012722.html by Mark Myatt onWed May 9 12:20:15 CEST 2001.
[9] http://stackoverflow.com/questions/8161836/how-do-i-replace-na-values-with-zeros-in-r.
[10] http://r.789695.n4.nabble.com/How-to-remove-multiple-outliers-td3921689.html by "aajit" on Oct 20, 2011; 5:41pm.
[11] Data Mining with Rattle and R-The Art of Excavating Data for Knowledge Discovery by "GrahamWilliams".
[12] http://jeetworks.org/node/74 Custom Principal Component Analysis (PCA) Plots in R Submitted by Jeet Sukumaran on Mon, 07/26/2010 - 20:04.
[13] Extending the Linear Model with R- Mixes Effects, Generalized Linear and Nonparametric Regression Models by "Juliun J. Faraway".