**Using R in Machine Learning - the absolute basics for dummies**

Simple and easy way to implement a Machine learning model is using R and its predefined functions. Implementing a model in R doesn’t require a high-level understanding of machine learning concepts such as gradient descent, regression, fitting and so on. A basic idea about simple linear functions, probability, matrices would suffice for initial stages.

R is a free environment which compiles and runs on Windows, UNIX, and MacOS. Further details about software installation, background, and the documentation is available at <https://www.r-project.org/>

**A quick glance at R data types:**

We use variables to store the data in any programming language. Based on the data type definition, OS allocates the memory. Like Python in R, the data types are not declared for a variable. The frequently used R object types are Arrays, Lists, Vectors, Matrices, Factors, Data Frames.

*Factor*: To encode a vector as a factor

*# Vector or a column of a data frame*

*# Can be a data like Name of person and City*

*city <- c('San Jose', 'Chicago', 'New York', 'New York' , 'San Jose', 'Chicago','Aurora')*

*cities <- factor(city)*

*print(cities)*

*[1] San Jose Chicago  New York New York San Jose Chicago  Aurora*

*Levels: Aurora Chicago New York San Jose*

Here levels represent the unique values of cities.

*Data Frame****:*** A tabular object which can contain different data types. It is a list of vectors of equal length.

*# Data frame*

*products <- data.frame(*

*ASIN = c('B014FCC4NO', 'B00RE20CVO' ,'B00HEZV6AC'),*

*Category = c('HeadPhones', 'Cables', 'Security & Surveillance'),*

*Price = c(25.2, 55, 100.86))*

*print(products)*

*ASIN                 Category  Price*

*1 B014FCC4NO      HeadPhones  25.20*

*2 B00RE20CVO       Cables  55.00*

*3 B00HEZV6AC Security & Surveillance 100.86*

**R - Linear Regression**

Below sample will provide a basic idea on how to read and process the data, train the ML model, get predicted value from the model. Data processing in R will be easier if you have a prior knowledge of pandas data frames in python. Initial two steps are written with respect to the unprocessed dataset If the cleansed data is readily available you can start with step 3.

1. Data

Get the data from a valid source. Data can be extracted either from multiple databases, Data warehouse, World Wide Web or Other information repositories.

2. Data Cleansing and Integration

If the data is available at different sources and in different formats, the important step to start with is data preprocessing. Remove noise and inconsistent data and combine the data from multiple sources.

Load the required datasets into a data frame. R can read from rds, xlsx or CSV file. CSV, xlsx files maintain the row 0 as headers which can be read by choosing header value as TRUE.

#Read Data from multiple files to Data frame

*df1 <- read.csv("File1.csv", header=TRUE)*

*df2 <- read.csv("File2.csv", header=TRUE)*

#To change the column names in a data frame

*colnames(df1)[2] <- "NEW\_NAME"*

#Combine the data frames

*finaldata = rbind(finaldata, df1)*

*finaldata = rbind(finaldata, df2)*

*saveRDS(finaldata,"transformed.rds")*

*write.csv(file="transformed.csv", x=finaldata)*

Make sure that the column names, data types are same in all the year files before merging. Only the relevant data to the analysis task are retrieved from the databases.

3. Data Selection & Transformation

Read the data from a rds file and Subset the data set with required columns and included filters if required.

Select = c(column names) will allow to subset only the specified column from the parent dataset and columns name == “Specific value” will subset the rows containing the specified values in the stated columns.

*mydata <- readRDS("transformed.rds")*

*newdata.frame <- subset(mydata, STATUS == "CERTIFIED" & EMPLOYER\_NAME == "ABC" & JOB\_TITLE == "ENGINEER", select = c(YEAR,WAGE, STATE))*

*newdata.frame$WAGE <- as.numeric(sub(",","", newdata.frame$WAGE))*

The column values which should be treated as number can be converted to numeric type using as.numeric() function. For predicting the wage for future years, A mean wage value of the previous years can be set as an input to the model. Mean of column values can be calculated as below.

*nmatrix<-ddply(subset(newdata.frame,STATE == "CA"),~YEAR,summarize, mean=mean(WAGE))*

Plyr is the R package which has the function ddply to split data frame, apply functions and return a data frame. The summarize function is applied to mean wages and the result are returned in a data frame.

4. Plot Data:

A simple graph featuring Year on X axis and mean wage on Y axis can be plotted using ggplot2. These plots help to visualize the data patterns.

*ggplot(data=nmatrix,aes(YEAR,mean))+geom\_smooth(method="lm")+ggtitle("Mean Salary\n")*

5. Train (Mining & pattern evaluation)

Linear regression can be implemented on a dataset with one or multiple input variables to predict a real-valued output based on the input. If the model input variable is x, Linear Regression tries to fit the output variable as a function of x (f(x)). lm() function will create a simple regression model in R.

Here Mean wage is a function of year. Mean = f(YEAR).

# Apply linear regression to the training data set

*lm<-lm(mean~YEAR,data=nmatrix)*

*summary(lm)*

summary() function will provide the details about model residuals, coefficients, errors...

# Verify the fitted value and residuals for the train data linear model

*nmatrixfitting <- data.frame(nmatrix , fitted.value= fitted (lm), residual= resid (lm))*

# Verify the model fit - lwr and upr values

*predict(lm,interval="confidence")*

Detailed lm() usage -> <https://www.rdocumentation.org/packages/stats/versions/3.4.3/topics/lm>

6. Predict

Create a data frame with the year for which the wage is to be predicted. Predict the wage by applying the lm and update in the data frame.

*newyear <- data.frame(YEAR = 2018)*

*newyear$mean <- predict(lm,newyear,type = "response")*

The above step will create a column with the column name as ‘mean’ and stores the model response (wage prediction) for the new year.