****

**Week 6 : Final Project Report**

Shivani Grover

Neeti Panjwani

Sarvesh Prabhu

Chidvilas Tammireddy

Palak Kansara

Course: ALY 6040: Data Mining applications

CRN: 81062

Professor: Justin Grosz

Date: May 16, 2019

**INTRODUCTION** :

In order to understand the customers’ behavior, we need Data Mining. Data Mining is the process that involves collecting, filtering and analyzing data. This helps in covering advanced tools and technologies. We have chosen the Gapminder dataset. First major step in Data Mining includes Data cleaning, which is very efficient in every business nowadays. There is inaccuracy in every dataset, therefore cleaning these records from a dataset is called data cleaning. We will be covering a few data cleaning methods in our dataset to make sure our data is clean and can be further analyzed. To understand the variables in the Gapminder dataset and find the relationship between them, we perform Regression analysis. It is one of the most important statistical techniques for business applications and helps in determining the relationship between two or more variables. There are many techniques that can be used to make predictions and decide the best set of variables that can be used to build predictive models. We will concentrate on Linear, Ridge and Lasso Regression techniques and compare the results.

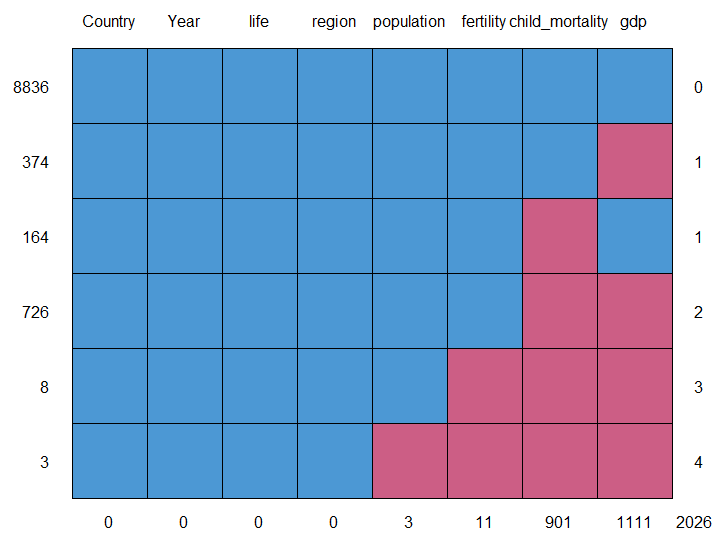
**PROBLEM STATEMENT:**

**To determine the expected GDP of various countries located in different regions using the information about other social factors such as child mortality rates, population, life expectancy, and fertility rates of those countries.**

**DATA CLEANING:**

The source of our data is the gapminder.org website. Gapminder provides with many datasets with data ranging from wide varieties of topics. The dataset has 10111 observations with 8 variables namely, country, region, year, fertility rate, population, child mortality rates, life expectancy, and gdp values corresponding each country for the year under consideration. The dataset provides the information from 1964 to 2013 for each country. There are a few techniques that we have used to remove inaccurate records from the dataset. For data cleaning, the platform used is RStudio and Tableau. The dataset available is not ready for analysis. It has many missing values.

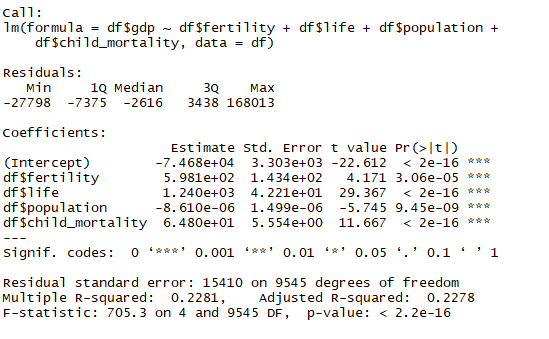
Using summary() function, it can be noted that variable ‘fertility’ has 11 ‘population’ has 3, child mortality has 901, and ‘gdp’ has 1111 N/A values. There are 2026 NA values constituting of 20% of the entire data.



1. There are common countries in which both gdp and child mortality data is missing. Thus, in total, there are 24 countries out of 204 which are omitted. This accounts for 11.7% of the total countries. Therefore, we can conveniently omit this dataset without adversely affecting the results. Also, we do not have complete data about these countries, considering them in the analysis may lead to some unwanted unforeseen errors when analysis will be conducted.
2. For only one metric completely missing values, we replaced NAs in the average gdp ad average child mortality by countries columns with 0. This is done to retain and consider the given other metrics values for that country instead of deleting the record.
3. For the countries which have few NA values, that is, less than 50, we replaced those missing values with the average value of the metric of that country or 0 if the average is null. This is done to reduce the data omitted and introduce some meaningful value in the dataset.

**ANALYSIS – REGRESSION TECHNIQUES:**

Performing Regression analysis on the cleaned dataset to determine the significant variables. We are determined to perform regression analysis to determine the dependence of GDP of those countries on other factors for which the data is available. Though, logically thinking it is not very convincing to perceive that GDP of a country depends upon just the four factors mentioned. To determine the proof of its level of dependency on other factors, we carry out regression analysis. We will check the presence of seasonality and trends in the dataset to facilitate prediction for the future. Here, logistic regression cannot be applied as the dependent variable is not binary. Thus, linear regression is applied with dependent variable being GDP and the independent variables being child mortality, population, fertility, and life expectancy. The first technique used is : **Linear Regression.** The summary after carrying out regression is shown in the figure below :



It is can be observed that all the independent variables contribute to the level of GDP of that country. This can be accepted from the p-values being less than 0.05 for 95% confidence intervals and large t-values of all the independent variables, proving that these do affect the dependent variable’s levels. The R2 value is 0.2278, which is not very large, indicating the fact that there might be some more factors contributing to the GDP level of the countries. The model formed is:

GDP = (-7.47\*104) + 5.98\*102)\*fertility + (1.24\*103)\*life + (-8.61\*10-6)\*population + (64)\*child\_mortality

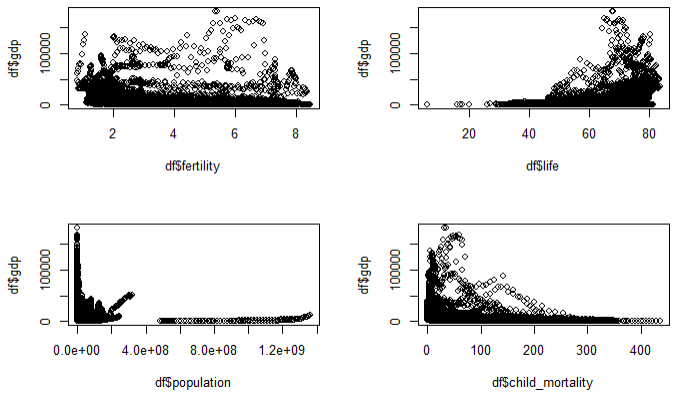


Fig 2: scatter plot of GDP V/s other parameter

From fig 2. It is clear that life expectancy and child mortality are stronger contributing factors to GDP than population and fertility, as they show some erratic behavior, which makes them weak variables affecting GDP. Now using the predict() function to determine the predicted GDP values using the model created and adding them as another variable in the dataset under consideration.

Error calculation is an important factor to determine the accuracy of the linear regression model generated. Error is the difference in between the observed values and the expected GDP value for each country. The error variable and its squared values are calculated and added to the dataset. Mean square error is the mean value of the square of the error calculated. It is a factor which helps to determine the applicability of the model on the given dataset.

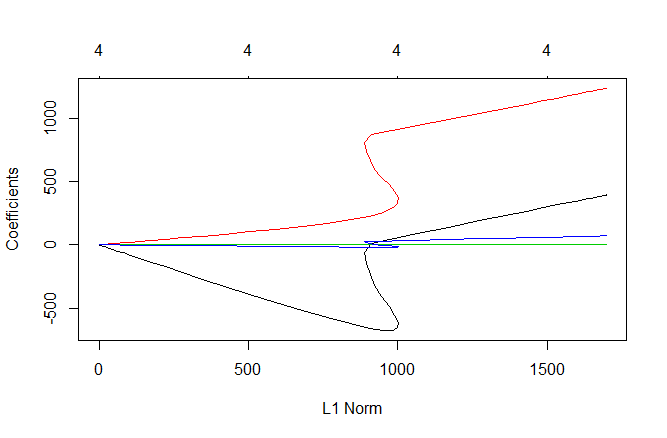


Fig 4: Mean Square Error

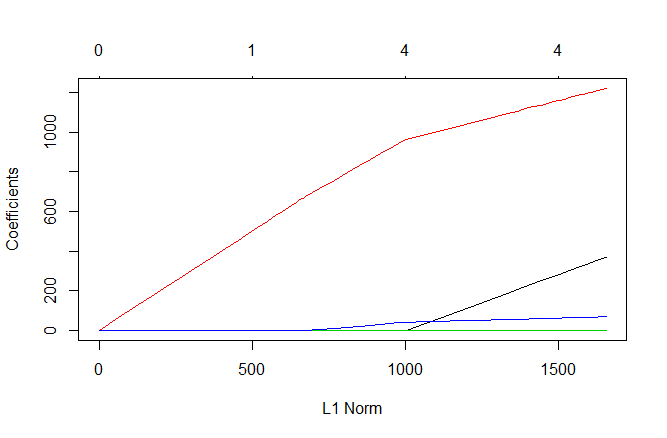
The linear regression method is very useful in determining the best fit model with the given variables getting the least possible error. But, a lot of times, the error value tends to become too large to accept the model. The obtained MSE for the given dataset is very large, proving that there may be more factors which needs to be considered to determine GDP more accurately or that other models and methods may provide much accurate GDP values with the existing data variables. Though with application of linear regression model, and the current variables, the error value is obtained is optimum.

The next steps are expected to find a model that can reduce the error obtained in this case. To check for optimization, we perform **Ridge** and **Lasso** **Regression**.

The Ridge regression model is fit if the alpha value is 0 while the Lasso model is fit if the alpha value is 1. In order to conclude the testing error of both the models, we split the samples into a Training and a Testing set. Once the range for the lambda values is set, we perform the regression models separately by installing the ‘glmnet’ package prior to the analysis. The Ridge regression graph after plotting looks like this:

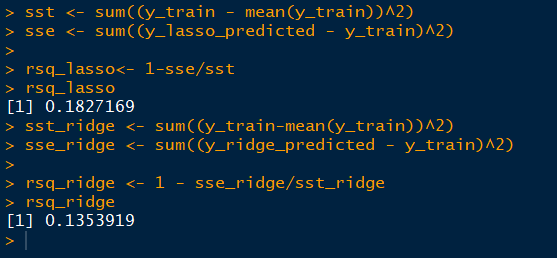


While the graph for Lasso Regression looks like this: This is found after finding the summary.



Even though we are controlling our model from fitting to the training datasets, we need to identify the optimal value of lambda in order to understand the tuning parameter. To do this, we perform cross-validation by using the cv.glmnet() built in function. This function by default performs 10-folds cross-validations. To focus on the parts of the ‘fit’ that matter, we produce the ridge and the lasso fit objects to find the deviance percentage. To find the predicted values that are close to the observed values, we have fit the data to the regression models.

With the preferred model’s lambda value, we are using the predicting the same dataset with different regression models. This prediction is done to find the optimum values from the rest obtained data. At the end, in order to find the expected value of the squared error loss. Here we are comparing the sum of the squares of the total and the total of the errors in order to differentiate the standard error value. Comparing the data folds in both the regression models, we get the R2 values as :



After computing the results, we observe the following :

A screenshot of a cell phone

Description automatically generated

**CONCLUSION** :

Analyzing the three models that we have used, a few factors that we considered to select the most effective regression model are understanding the data and identifying the relationship between all variables, statistical significance of parameters, R-squared values and the Mean-Squared Error values. Comparing these values in all the three models we see that linear model has a high end R2 value of **0.22** and the least error values bandwidth. We conclude to choose the Linear regression model as the best optimized method, as the results show that the other methods’ parameter estimates had larger variances, hence, making the model unreliable. Linear Regression decreases the size of the coefficients to the maximum with an R2 value of 0.2278, while Ridge and Lasso having 0.135 and 0.182 respectively. By exactly applying the fit model to the training data set, it can be said that fine tuning is still possible and Linear model, with the help of the cross-validation tests reduces the model complexity. The Linear model and the Ridge Regression models had similar error statistics, therefore, we are preferring the one that is simple and easy to understand. i.e. we are following the K.I.S.S principle.

**REFERENCES** :

Dataset retrieval from <https://www.gapminder.org/> website.

Elite Data Science (N.D) “Data Cleaning” Retrieved from <https://elitedatascience.com/data-cleaning>

Selva (n.d), ‘Linear Regression’. Retrieved from <http://r-statistics.co/Linear-Regression.html>

Ralph (April 23, 2010), ‘Simple Linear Regression’. Retrieved from <https://www.r-bloggers.com/simple-linear-regression-2/>

R Bloggers (May 23, 2017) ‘Ridge Regression and the Lasso’ retrieved from <https://www.r-bloggers.com/ridge-regression-and-the-lasso/>

**APPENDIX:**

file1<- read.csv('C:/Users/dellpc/Downloads/gapminder.csv', header=TRUE)

summary(file1)

#install.packages('mice')

library('mice')

md.pattern(file1)

#Removing the not required countries

#install.packages('dplyr')

#install.packages('stringr')

library(dplyr)

library(stringr)

file2 <- file1 %>%

filter(!(str\_detect(Country,"Ã.land")))

file3 <- file2%>%

filter(!(Country %in% c("Central African Rep.","Czech Rep.","Dominican Rep.","Korea, Dem. Rep.","Korea, Rep.","Kyrgyzstan","Laos","Netherlands Antilles","Saint Lucia", "Saint Vincent and the Grenadines","Tokelau","Yemen, Rep.")))

md.pattern(file3)

#finding the average values for each country and adding those columns in file5

#install.packages('tidyverse')

library(tidyverse)

file4 <- file3 %>%

group\_by(Country) %>%

summarise(Average\_mortality = mean(child\_mortality,na.rm = TRUE), Average\_gdp = mean(gdp, na.rm = TRUE))

file5 <- merge(file3,file4, by = "Country")

#Putting zero value for countries in which values are missing completely

file5$child\_mortality[is.na(file5$Average\_mortality)] <- 0

file5$Average\_gdp[is.na(file5$Average\_gdp)] <- 0

file5$child\_mortality[is.nan(file5$Average\_mortality)] <- 0

file5$Average\_gdp[is.nan(file5$Average\_gdp)] <- 0

View(file5)

file5$child\_mortality <- ifelse(is.na(file5$Average\_mortality),file5$Average\_mortality, file5$child\_mortality)

file5$gdp <- ifelse(is.na(file5$gdp),file5$Average\_gdp, file5$gdp)

df <- file5

df<-read.csv("G:/Data mining/week4/cd.csv")

df<-df[,1:9]

#Performing Linear Regression

gdp\_reg<-lm(df$gdp~df$fertility+df$life+df$population+df$child\_mortality,data=df)

summary(gdp\_reg)

#Getting a scatter plot of all independent variables with dependent variable

par(mfrow=c(2,2))

plot(df$fertility, df$gdp)

plot(df$life, df$gdp)

plot(df$population, df$gdp)

plot(df$child\_mortality, df$gdp)

# Predicting the GDP values with the model

prediction<-as.data.frame(predict(gdp\_reg,new\_data=df))

df1<-df

df1[, 10]<-prediction

#Error Calculation

df1[, 11] <- df1[, 10] - df1[, 8]

df1[, 12] <- df1[, 11]\*\*2

#View(df1)

MSE <- mean(df1[, 12])

#Naming the column in the dataset

names(df1)<-c("ID", "Country" , "Year" , "Fertility\_Rates" , "Life\_Expectancy" , "Population" , "Child\_Mortality\_Rates" , "Observed\_GDP" , "Region" , "Predicted\_GDP" , "Error" , "Error^2")

# Dividing in test and train dataset

library(rsample) # data splitting

library(glmnet) # implementing regularized regression approaches

library(dplyr) # basic data manipulation procedures

library(ggplot2)

df<-read.csv("G:/Data mining/week4/cd.csv")

data <- df[c('fertility','life','population','child\_mortality','gdp')]

d\_div<-sample(2,nrow(data),replace=T,prob = c(0.6,0.4))

d\_test<-data[d\_div==1,]

d\_train<-data[d\_div==2,]

x\_test <- model.matrix(gdp~.,d\_test)[,-1]

y\_test <- d\_test$gdp

x\_train<- model.matrix(gdp~.,d\_train)[,-1]

y\_train<-d\_train$gdp

#Giving a range to the lambda value

lambda <- 10^seq(10, -2, length = 100)

#Ridge Regression

ridge.mod <- glmnet(x\_train,y\_train, alpha = 0, lambda = lambda)

predict(ridge.mod, s = 0, type = 'coefficients')

summary(ridge.mod)

plot(ridge.mod)

#Cross-validating the Ridge model

model\_ridge <- cv.glmnet(x\_train,y\_train,alpha = 0)

summary(model\_ridge)

plot(model\_ridge)

#Ridge fit

ridge\_fit = glmnet(x\_train,y\_train,alpha = 0, lambda = model\_ridge$lambda.1se)

ridge\_fit

names(ridge\_fit)

ridge\_fit$beta[,1]

summary(ridge\_fit)

#Ridge Prediction

opt\_ridge\_lambda <- ridge\_fit$lambda

y\_ridge\_predicted <- predict(ridge\_fit, s = opt\_ridge\_lambda, newx = x\_train)

# Sum of squares Total & Error - Ridge

sst\_ridge <- sum((y\_train-mean(y\_train))^2)

sse\_ridge <- sum((y\_ridge\_predicted - y\_train)^2)

rsq\_ridge <- 1 - sse\_ridge/sst\_ridge

rsq\_ridge