LDLcalc: An R package for calculating and predicting low density lipoprotein values.

Petros Paplomatas

9/10/2021

This Package allows for a wide variety of ways to calculate (through equations) or predict (utilizing 9 Machine Learning methods, as well as a stack algorithm combining them) Low Density Lipoprotein values.

How to install the LDLcacl Package.

```
install.packages('LDLcalc',repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/ppapl/Documents/R/win-library/4.1'
## (as 'lib' is unspecified)
```

```
## package 'LDLcalc' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\ppapl\AppData\Local\Temp\RtmpiApsOA\downloaded_packages
```

Package Functions.

```
library(LDLcalc)
ls("package:LDLcalc")
```

```
## [1] "Corr_Plot"                     "LDL_eq"
## [3] "LDL_ML_Main"                   "LDL_ML_Main_All_Models"
## [5] "LDL_ML_Main_StackingAlgorithm" "LDL_ML_predict"
## [7] "LDLallEq"                      "model"
## [9] "SampleData"
```

Example on how to calculate LDL using some or all of the equations.

```
library(LDLcalc)

FriedewaldLDL<-LDL_eq(170.5,35.12,175,"Friedewald")
Martin360LDL<-LDL_eq(170.5,35.12,175,"Martin360")
print(paste("FriedewaldEquation", ":", FriedewaldLDL,"Martin360Equation", ":"
,Martin360LDL))
```

```
## [1] "FriedewaldEquation : 100.38 Martin360Equation : 106"
```

```
allequations<-LDLallEq(170.5,35.12,175)
```

```
## [1] "Friedewald : 100.38"
## [1] "Ahmadi : 165.455301354992"
## [1] "Chen : 104.342"
## [1] "Anandaraja : 93.95"
## [1] "NewFormula : 99.4734"
## [1] "deCordova : 101.751608"
## [1] "Vujovic : 109.832554744526"
## [1] "Hattori : 94.0072"
## [1] "Puavillai : 106.213333333333"
## [1] "Hatta : 91.63"
## [1] "Martin180 : 106"
## [1] "Martin360 : 106"
## [1] "Martin2000 : 107"
## [1] "DeLong : 107.38"
## [1] "Rao : 103.22375"
```

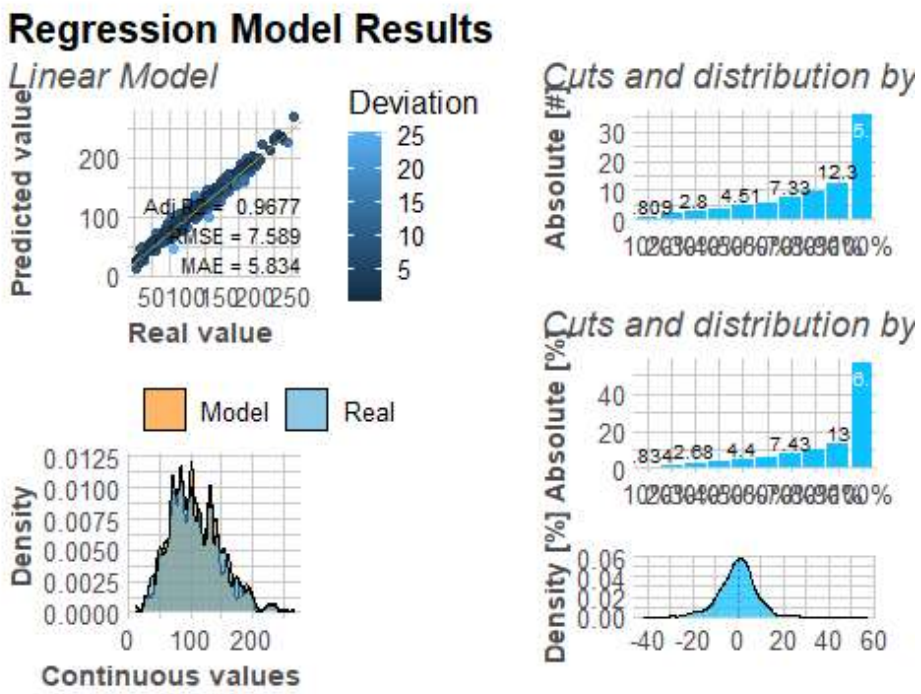
Time to train our model. We will create a model using “lm” method (there are 8 more methods available, which are available in the package document), a stacking model as well as a model which calculates all 9 models at the same time. In addition, we will create some plots, which will give us important information about the relation and the accuracy of the models.

```
# Test single ML models
model = LDL_ML_Main(SampleData,0.7,"lm",ReportMultiPlot=TRUE)

## Loading required package: ggplot2

## Loading required package: lattice

## Warning in .font_global(font, quiet = FALSE): Font 'Arial Narrow' is not
## installed, has other name, or can't be found
```



```
# Test stack algorithm model
stackModel = LDL_ML_Main_StackingAlgorithm(SampleData,0.7,ReportMultiPlot = T
RUE,ComparisonPlot = FALSE)
```

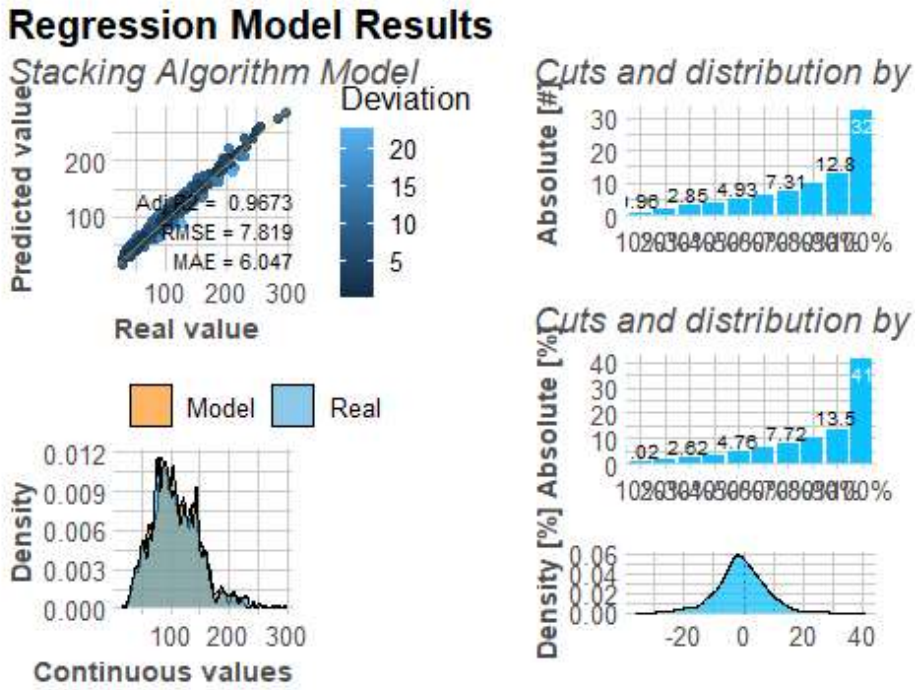
```
## Loading required package: earth
```

```
## Loading required package: Formula
```

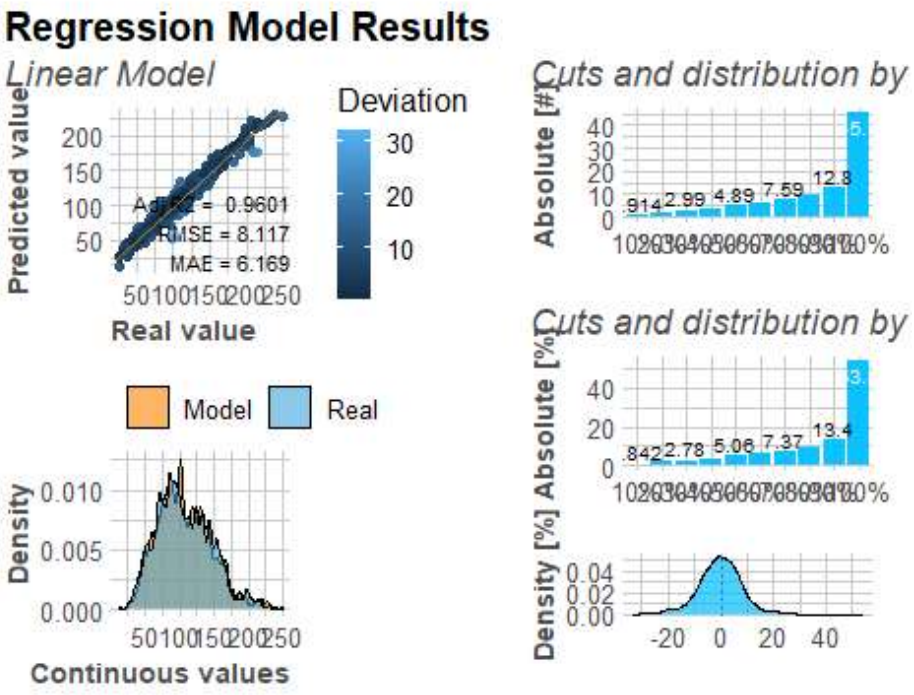
```
## Loading required package: plotmo
```

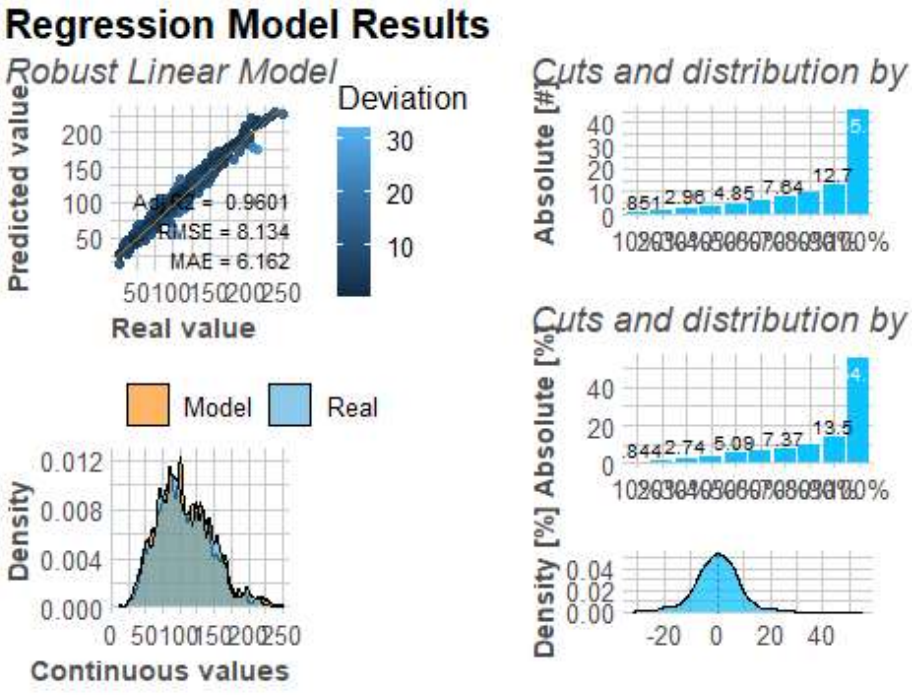
```
## Loading required package: plotrix
```

```
## Loading required package: TeachingDemos
```



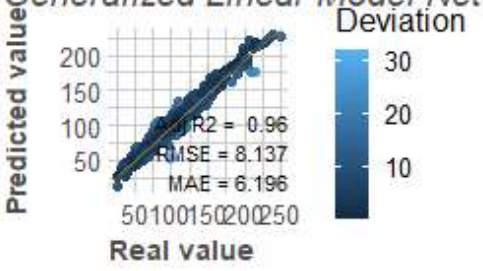
```
# Test all models at once
allModels = LDL_ML_Main_All_Models(SampleData,0.7,ReportMultiPlot = TRUE,Comparisonplot=TRUE)
```

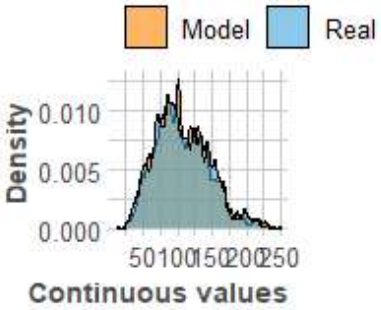




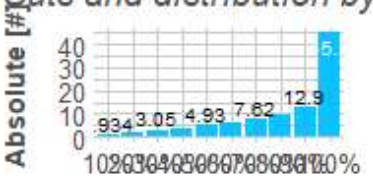
Regression Model Results

Generalized Linear Model Net

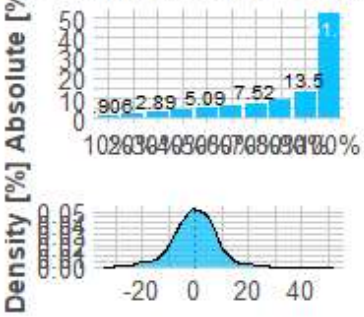




Cuts and distribution by

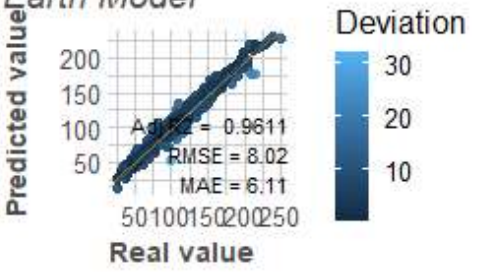


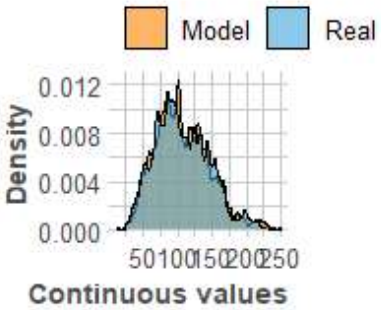
Cuts and distribution by



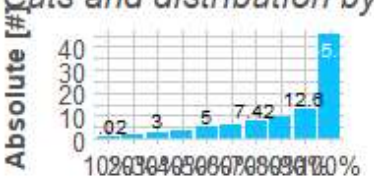
Regression Model Results

Earth Model

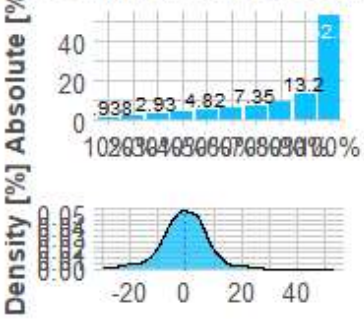




Cuts and distribution by

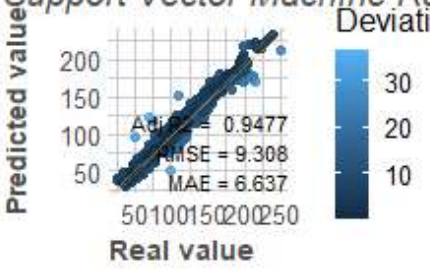


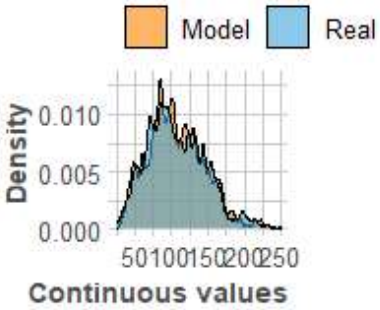
Cuts and distribution by



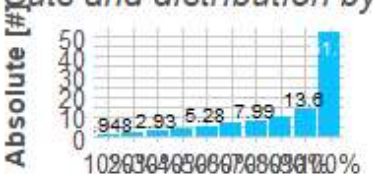
Regression Model Results

Support Vector Machine Radial

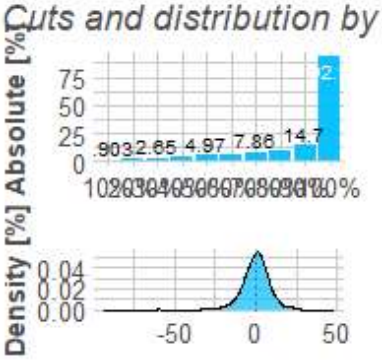




Cuts and distribution by

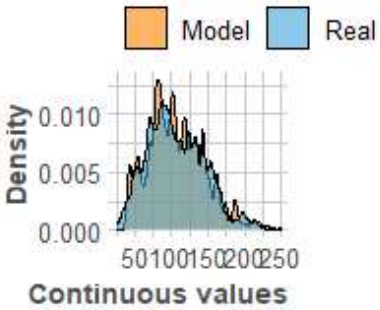


Cuts and distribution by

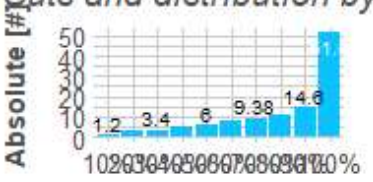


Regression Model Results

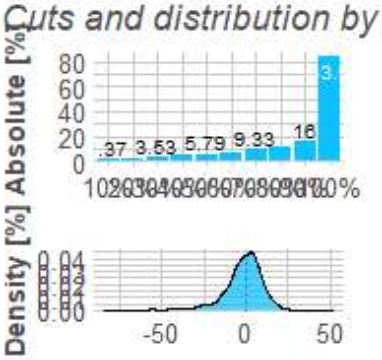
k-Nearest Neighbors Algorithm



Cuts and distribution by

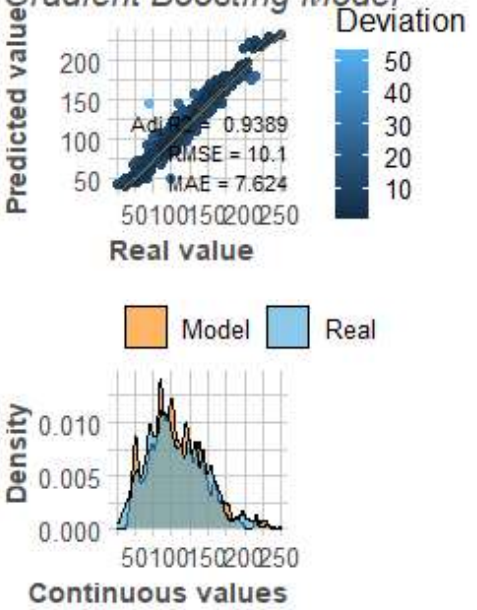


Cuts and distribution by

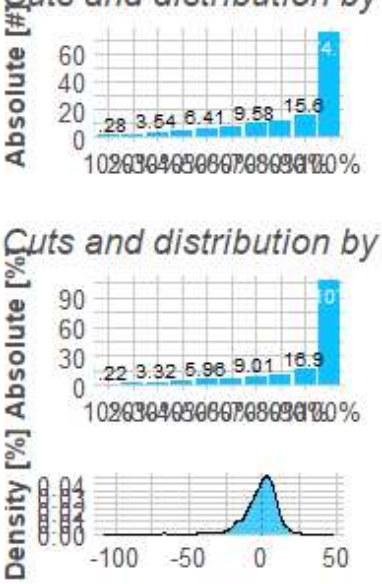


Regression Model Results

Gradient Boosting Model

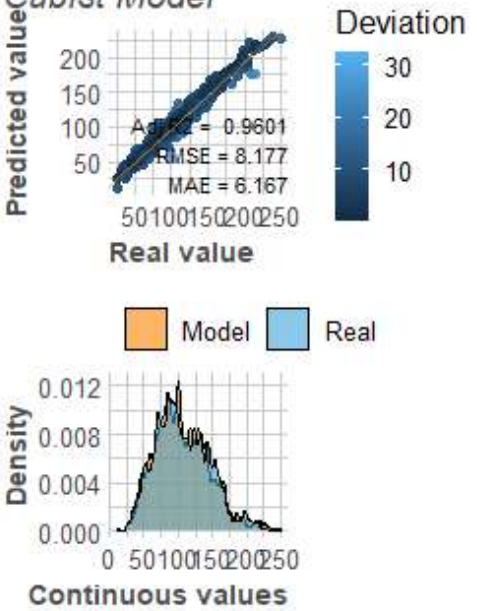


Cuts and distribution by

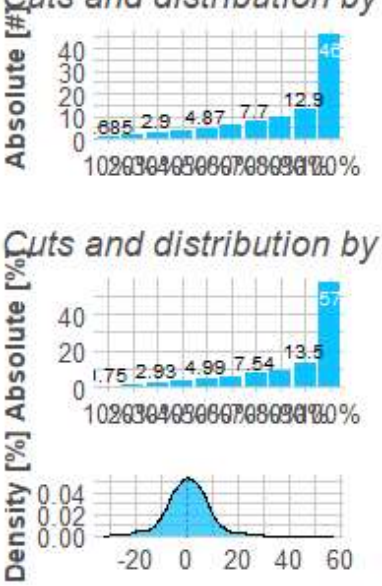


Regression Model Results

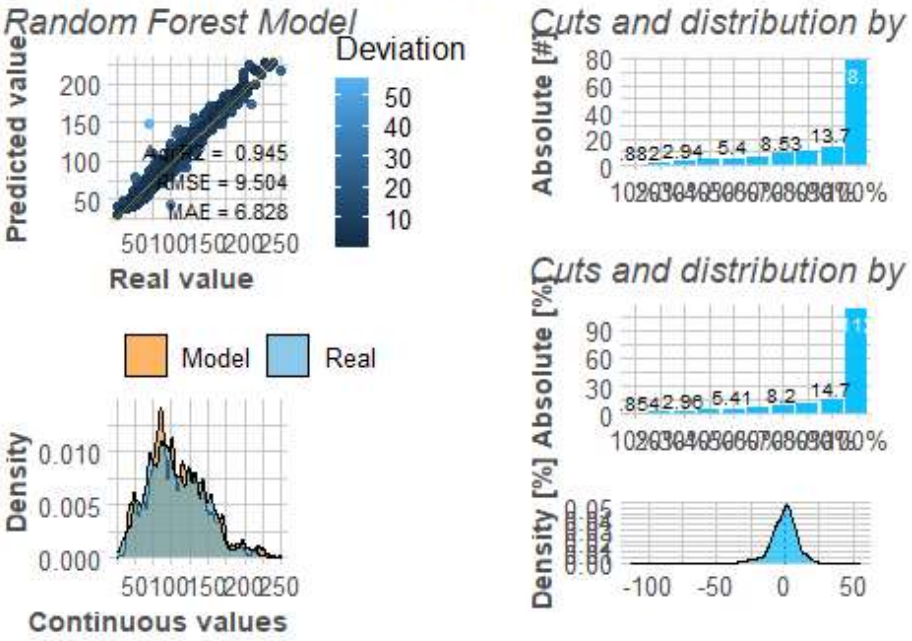
Cubist Model



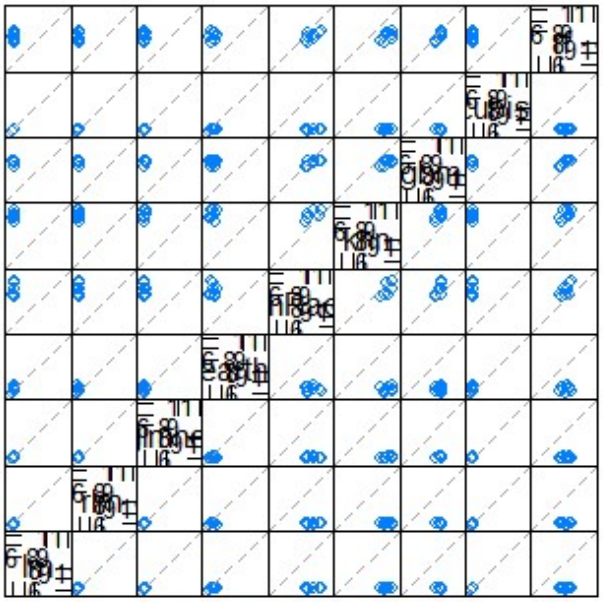
Cuts and distribution by



Regression Model Results



MAE



Scatter Plot Matrix

Now it is the time to make some predictions of the above models.

```
library(data.table)
modelPrediction = LDL_ML_predict(model,data.table(CHOL=170.5,HDL=35.12,TG=175
))

stackPredictions = LDL_ML_predict(stackModel,data.table(CHOL=170.5,HDL=35.12,
TG=175))

allModelsPredictions = LDL_ML_predict(allModels,data.table(CHOL=170.5,HDL=35.
12,TG=175))

print(paste("The prediction of lm model is :", modelPrediction ,"and the pred
iction of stackingalgorithm is : ",stackPredictions ))
```

```
## [1] "The prediction of lm model is : 106.965509826665 and the prediction o
f stackingalgorithm is :  107.557813609639"
```

```
print("The predictions of all models are : ")
```

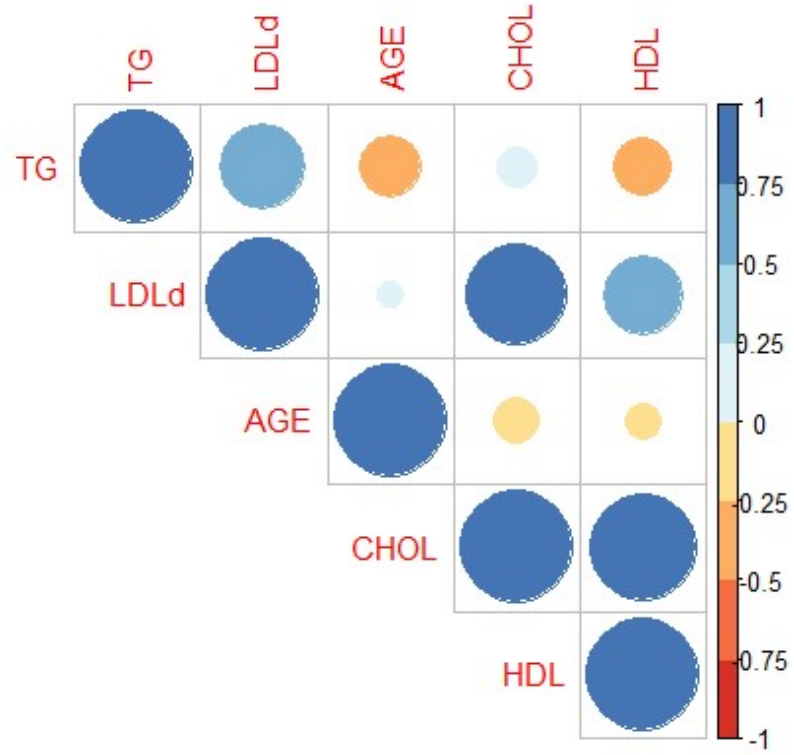
```
## [1] "The predictions of all models are : "
```

```
allModelsPredictions
```

```
##            lm      rlm   glmnet   earth svmRadial   knn      gbm   cubist
## [1,] 107.1505 106.9062 107.1277 107.137  109.7752 109.6 108.7245 106.5737
##            rf
## [1,] 105.8668
```

Notable plot showing correlation between the imported values.

```
CorrelationPlot<-Corr_Plot(SampleData[2:6])
```



Feel free to contact me at p.paplomatas@hotmail.com .