Predicting the Life Expectancy in a Country

Leonardo Ribeiro

13/10/2021

Life Expectancy in a Country

The purpose of this text is to discuss the creation a Machine Learning (ML) model that is able to predict, with a reasonable precision, the life expectancy in a country given the following variables:

Field	Description
Country	Country
Year	Year
Status	Developed or Developing status
Life expectancy	Life Expectancy in age
Adult Mortality	Adult Mortality Rates of both sexes (probability of dying between 15 and 60 years per 1000 population)
infant deaths	Number of Infant Deaths per 1000 population
Alcohol	Alcohol, recorded per capita (15+) consumption (in liters of pure alcohol)
percentage expenditure	Expenditure on health as a percent of Gross Domestic Product per capita(%)
Hepatitis B	Hepatitis B (HepB) immunization coverage among 1-year-olds (%)
Measles	Measles - number of reported cases per 1000 population
BMI	Average Body Mass Index of entire population
under-five deaths	Number of under-five deaths per 1000 population
Polio	Polio (Pol3) immunization coverage among 1-year-olds (%)
Total expenditure	General government expenditure on health as a percent of total government expenditure (%)
Diphtheria	Diphtheria tetanus toxoid and pertussis (DTP3) immunization coverage among 1-year-olds (%)
HIV/AIDS	Deaths per 1 000 live births HIV/AIDS (0-4 years)
GDP	Gross Domestic Product per capita (in USD)
Population	Population of the country
thinness 1-19 years	Prevalence of thinness among children and adolescents for Age 10 to 19 (%)
thinness 5-9 years	Prevalence of thinness among children for Age 5 to 9(%)
Income composition of resources	Income composition of resources
Schooling	Number of years of Schooling(years)

The data set used as basis for this study was found on Kaggle. For more details about each variable and its sources, visit **this link**.

The Goal

Given a set of specific data about a country, a Machine Learning model that returns the average life expectancy in it must be built.

Loading the Data

The first step is to load the data set of interest:

```
file <- list.files(pattern = ".csv")
raw_data <- read.csv(file)
dim(raw_data)
```

```
## [1] 2938   22
```

```
colnames(raw_data)
```

```
##  [1] "Country"                        "Year"
##  [3] "Status"                         "Life.expectancy"
##  [5] "Adult.Mortality"                "infant.deaths"
##  [7] "Alcohol"                        "percentage.expenditure"
##  [9] "Hepatitis.B"                    "Measles"
## [11] "BMI"                            "under.five.deaths"
## [13] "Polio"                          "Total.expenditure"
## [15] "Diphtheria"                     "HIV.AIDS"
## [17] "GDP"                            "Population"
## [19] "thinness..1.19.years"           "thinness.5.9.years"
## [21] "Income.composition.of.resources" "Schooling"
```

The data ensemble has 2938 rows and 12 columns (that represent characteristics of the countries). The names of those columns will be changed to better describe the data contained in it:

```
library(dplyr)
raw_data <- raw_data %>%
  rename( Thin.5.to.9.yo = thinness.5.9.years,
          Thin.10.to.19.yo = thinness..1.19.years,
          Measles.cases = Measles,
          Alcohol.consume = Alcohol,
          GDP.on.health = percentage.expenditure,
          HepB.immune = Hepatitis.B,
          Polio.immune = Polio,
          DPT.vacc = Diphtheria,
          Inc.comp.resource = Income.composition.of.resources,
          Years.in.School = Schooling,
          Deaths.under.5.yo = under.five.deaths,
          Infant.deaths = infant.deaths,
          Development.Status = Status)
```

The life expectancy in a country (which is the target variable on this study) is present on the column 'Life.expectancy'. Thus, the ML model will have to predict the value on this column.

Exploratory Data Analysis

This analysis' first step is to deal with 'NA' values in the data set:

```
de.find_values(raw_data, value = NA)
```

```
## [1] "The column Life.expectancy has 10 values equal to NA"
```

```
## [1] "The column Adult.Mortality has 10 values equal to NA"
## [1] "The column Alcohol.consume has 194 values equal to NA"
## [1] "The column HepB.immune has 553 values equal to NA"
## [1] "The column BMI has 34 values equal to NA"
## [1] "The column Polio.immune has 19 values equal to NA"
## [1] "The column Total.expenditure has 226 values equal to NA"
## [1] "The column DPT.vacc has 19 values equal to NA"
## [1] "The column GDP has 448 values equal to NA"
## [1] "The column Population has 652 values equal to NA"
## [1] "The column Thin.10.to.19.yo has 34 values equal to NA"
## [1] "The column Thin.5.to.9.yo has 34 values equal to NA"
## [1] "The column Inc.comp.resource has 167 values equal to NA"
## [1] "The column Years.in.School has 163 values equal to NA"
```

The columns 'HepB.immune' and 'Population' have a considerably large number of 'NA's present in it. For that reason, both will be removed from the data set and, only after that, the remaining rows containing 'NA' will be discarded.

```
raw_data <- dm.drop_cols(raw_data, c('HepB.immune', 'Population'))
raw_data <- na.omit(raw_data)
dim(raw_data)
```

```
## [1] 2301   20
```

Besides NA values, some misplaced zeroes can also cause trouble during this analysis, so they will be removed as well:

```
de.find_values(raw_data, value = 0)
```

```
## [1] "The column Infant.deaths has 668 values equal to 0"
## [1] "The column GDP.on.health has 5 values equal to 0"
## [1] "The column Measles.cases has 763 values equal to 0"
## [1] "The column Deaths.under.5.yo has 612 values equal to 0"
## [1] "The column Inc.comp.resource has 104 values equal to 0"
## [1] "The column Years.in.School has 9 values equal to 0"
```

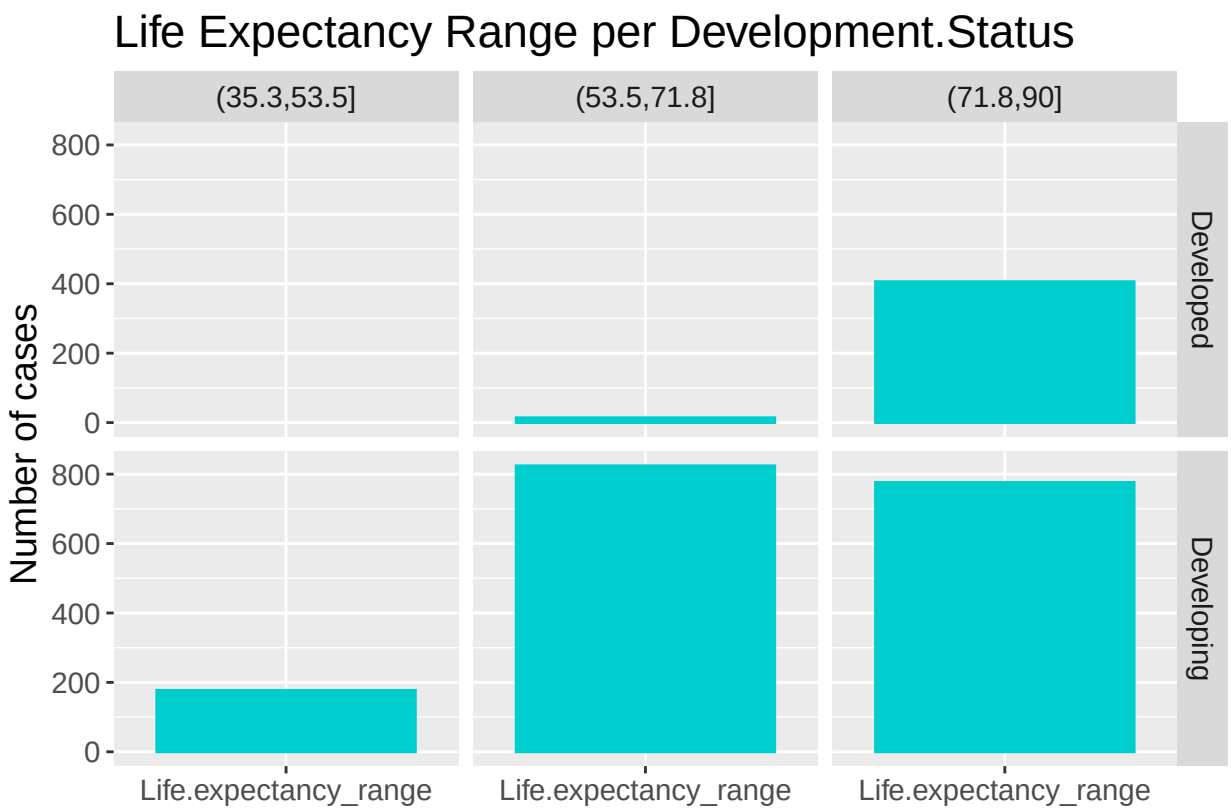
```
raw_data <- raw_data[raw_data$Inc.comp.resource != 0,]
raw_data <- raw_data[raw_data$Years.in.School != 0,]
dim(raw_data)
```

```
## [1] 2197   20
```

It is important to highlight that this decision to remove the zeroes was made after the evaluation of the correlation between the target variable 'Life.expectancy' and the predictor variables 'Inc.comp.resource' and 'Years.in.School'. For both predictors, the zeroes represented unwanted outliers.

Still on the subject of correlation, the relation between the categorical variable 'Development.Status' and life expectancy is depicted by the following graphic:

```
library(ggplot2)
raw_data <- dm.factorize_cols(raw_data, 'Development.Status')
raw_data <- dm.range_divide(raw_data, 'Life.expectancy', 3, 1)
dv.plot_multiple_bars_II(raw_data, c('Development.Status'),
                         'Life.expectancy_range',
                         title = 'Life Expectancy Range per',
                         xlabel = '',
                         ylabel = 'Number of cases')
```



The figure above shows that the life expectancy in developed countries is, for the major part, high (above 72 years). On the other hand, for developing countries, life expectancy is mainly divided between the ranges 53 to 71 and 72 to 90, although a considerable number of members of this group are also in the lower range. Based on that, one can conclude that there is a great inequality among developing countries and that developed countries are more homogeneous with respect to life expectancy.

Aside from the categorical variable mentioned, the data set also contains a series of numeric variables that impact the column Life.expectancy. The ones with the largest correlation coefficient are listed below:

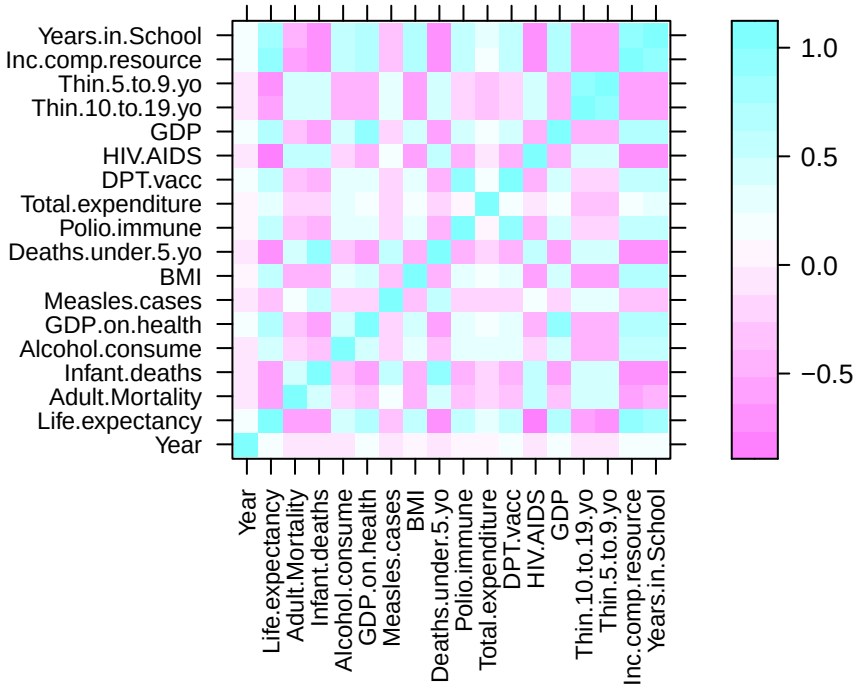
```
cors <- de.build_corr_coef_list(raw_data[,-c(1, 3, 21)], 'spearman')
de.get_corr_coef_w_target_var(data.frame(cors), 'Life.expectancy', 0.7)
```

```
## [1] "Corr coef between Life.expectancy and HIV.AIDS: -0.768563389743548"
## [1] "Corr coef between Life.expectancy and Inc.comp.resource: 0.905651209029379"
## [1] "Corr coef between Life.expectancy and Years.in.School: 0.822057706405847"
```

The result above shows that the correlation regarding Life.expectancy is stronger with the columns 'HIV.AIDS', 'Inc.comp.resource' and 'Years.in.School'. Additionally, the following graphic illustrates the strength of the correlation between those and the remaining variables in the data set:

```
require(lattice)
Map(dv.plot_corr_coeffs, cors,
    'spearman', 'Correlation between variables using the method')
```

Correlation between variables using the method spearman

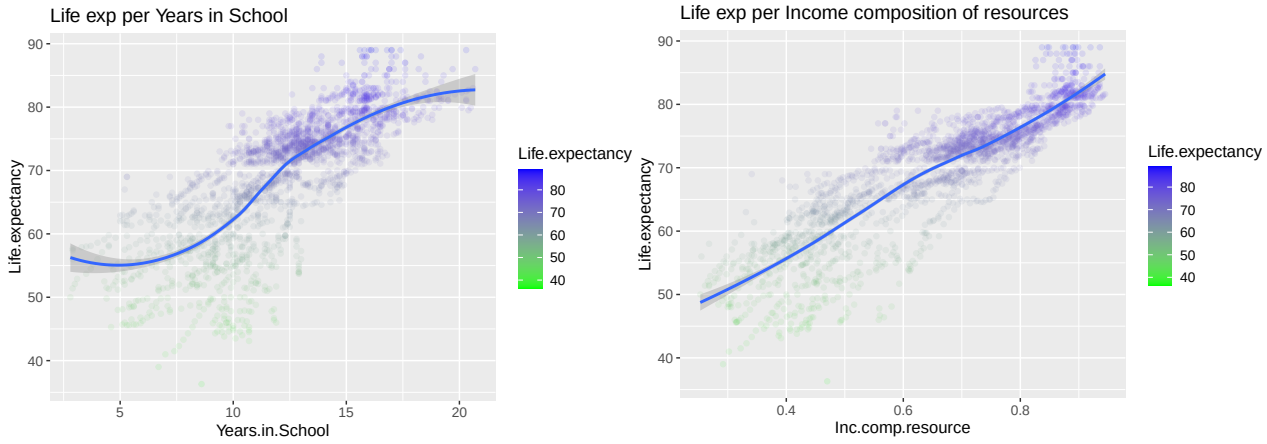


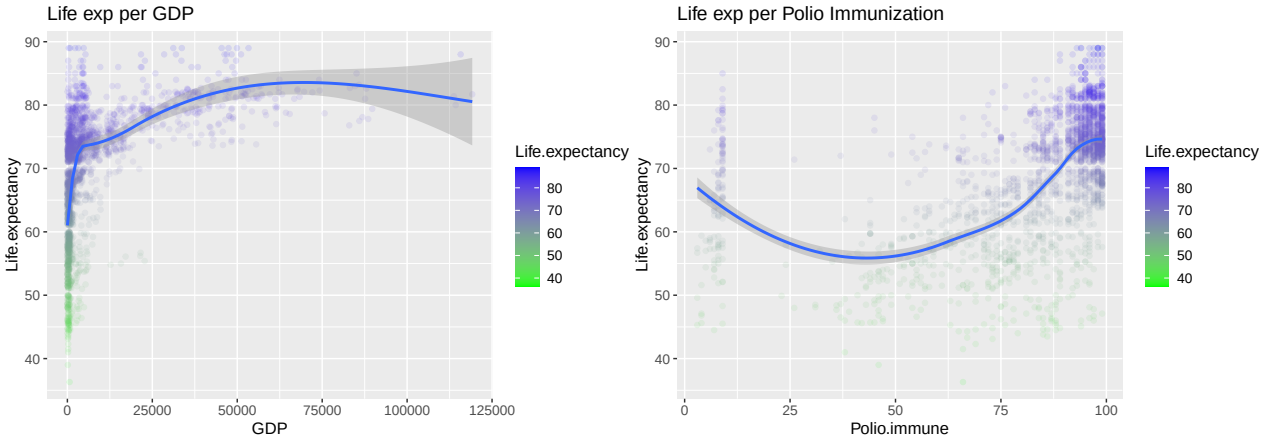
Thus, based on the last figure, one can notice that the columns "Years.in.School", "Inc.comp.resource", "GDP" and "Polio.immune" have a positive correlation with life expectancy. That is depicted by graphics below:

```
labels <- c("Life exp per Years in School",
            "Life exp per Income composition of resources",
            "Life exp per GDP",
            "Life exp per Polio Immunization")

xAxis <- c("Years.in.School", "Inc.comp.resource", "GDP",
           "Polio.immune")

Map(dv.plot_scatter, xAxis, labels)
```





Using the above graphics as basis, one can affirm that, in countries where people spend more time in school, the life expectancy is significantly larger. The same can be said about countries with a better income distribution - places where the index 'Income composition of resources' is larger also have citizens with a longer life span.

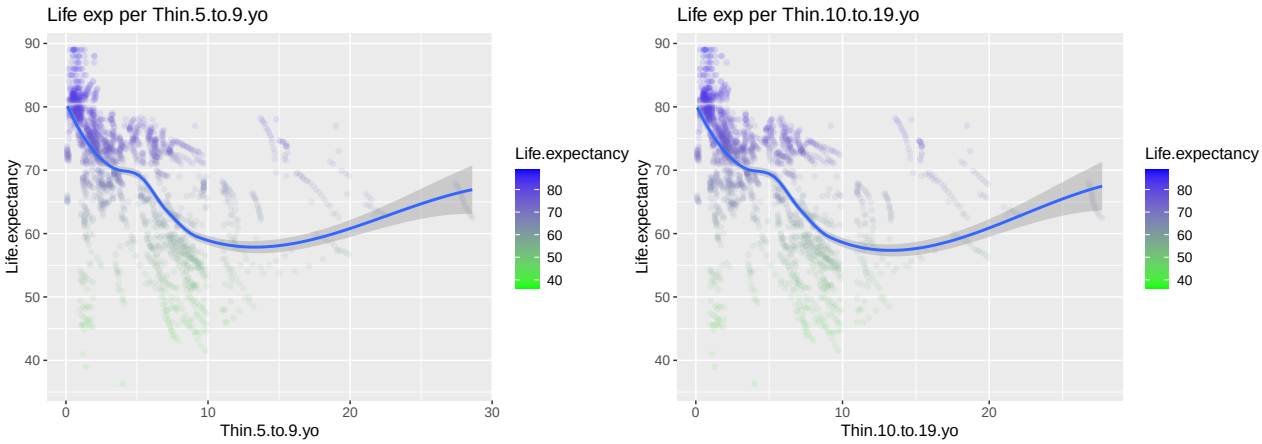
Moreover, by focusing on the statistically significant region in the graphics, it becomes clear that an increase in the GDP index is also associated with a rise on life expectancy. Also, regarding the poliomyelitis immunization, something curious happens: a considerable number of countries with a large life expectancy have low levels of immunization. That may be related to a carelessness due to the eradication of polio in those countries or even to the anti-vax movement. Disregarding this group, an increase in polio immunization is followed by an increase in life expectation.

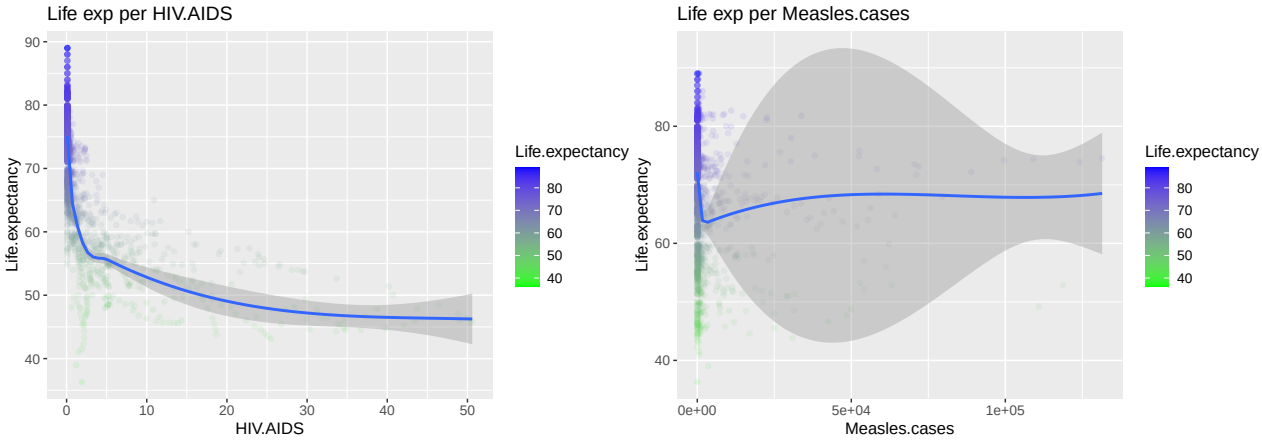
Next, variables with a negative correlation regarding life expectation will be analysed. More specifically, the columns "Thin.5.to.9.yo", "Thin.10.to.19.yo", "HIV.AIDS" and "Measles.cases".

```
labels_n <- c("Life exp per Thin.5.to.9.yo",
              "Life exp per Thin.10.to.19.yo",
              "Life exp per HIV.AIDS",
              "Life exp per Measles.cases")

xAxis_n <- c("Thin.5.to.9.yo",
             "Thin.10.to.19.yo",
             "HIV.AIDS",
             "Measles.cases")

Map(dv.plot_scatter, xAxis_n, labels_n)
```





On this second set of graphics, the first two are very similar - both point how the prevalence of thinness in different ages affect life expectancy. As those indicators decrease, life expectancy rises.

Furthermore, focusing again on the statistically meaningful regions, an increase on the values in the columns 'HIV.AIDS' and 'Measles.cases' seem to be linked to an abrupt decrease in life expectancy.

Having finished the correlation analysis, it is important to highlight the following: correlation does not imply causality. Even if correlation can be a clue, a deeper study would have to be performed in order to establish a cause and effect relation between two variables.

Data Manipulation

The data set used as basis for this work has a single categorical variable: Development.Status. In order to deal with it, the dummy coding technique will be used - assigning 1 to developed countries and 0 to the developing ones.

```
data <- raw_data
data$Development.Status = sapply(raw_data$Development.Status, function(x) {
  ifelse(x == 'Developed', 1, 0)
})
data <- dm.drop_cols(data, 'Life.expectancy_range')
```

Next, the remaining numeric variables will be normalized - presently, each column has a different scale, allowing its values to have distinct amplitudes. Normalization will homogenize the data and improve the ML model performance:

```
cols_to_normalize = c(colnames(data[,c(-1, -4)]))
data <- dm.normalize_cols(data, cols_to_normalize)
summary(data$GDP)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000000 0.003881 0.015409 0.065931 0.053286 1.000000
```

It is also important to see how the predictor variables are correlated among themselves:

```
de.get_corr_coef_predictor_vars(data.frame(cors), 'Life.expectancy', 0.8)
```

```
## [1] "Corr coef between Infant.deaths and Deaths.under.5.yo: 0.991995565541903"
## [1] "Corr coef between GDP.on.health and GDP: 0.93981102875336"
## [1] "Corr coef between Polio.immune and DPT.vacc: 0.932060215242153"
## [1] "Corr coef between Thin.10.to.19.yo and Thin.5.to.9.yo: 0.940492404310851"
## [1] "Corr coef between Inc.comp.resource and Years.in.School: 0.933452273915711"
```

The results above show that the columns 'Infant.deaths' and 'Deaths.under.5.yo' are strongly correlated. Therefore, in order to avoid over fitting, the first will be removed.

```
data <- dm.drop_cols(data, 'Infant.deaths')
```

Other columns with a significantly strong correlation are GDP and GDP.on.health, Polio.immune and DPT.vacc, thinness.5.9.years and thinness.10.19.years and Inc.comp.resource and Years.in.School. One could argue that those are essentially different characteristics and should all be considered, or that their strong correlation could create problems for the ML model. Nonetheless, the decision to remove or not those columns will only be made on the following section.

Now that the variables are normalized, have the correct type and the undesired columns were removed, the next step is the creation a ML model.

The Machine Learning Model

On this section, two ML models will be created - one using the function lm() and another using randomForest(). In both cases, the first step will be the same, i.e., to divide the data set into a set for training (70%) and a set for testing (30%):

```
random_indexes <- de.get_random_row_indexes(data, 70)
trainSet <- data[random_indexes, ]
testSet <- data[-random_indexes, ]
```

Model 1

For the first model, the relevance of the variables in the data set is depicted below:

```
relevance <- lm(Life.expectancy ~. - Country - Year - GDP - Polio.immune
                  - Thin.10.to.19.yo,
                data = trainSet)
summary(relevance)
```

```
##
## Call:
## lm(formula = Life.expectancy ~ . - Country - Year - GDP - Polio.immune -
##     Thin.10.to.19.yo, data = trainSet)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.5875  -1.9274  -0.0313   1.9181  12.0623
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         54.0122     0.5271 102.472  < 2e-16 ***
## Development.Status   0.9720     0.3009   3.230  0.00126 **
## Adult.Mortality     -8.1517     0.6438 -12.662  < 2e-16 ***
## Alcohol.consume     -4.2938     0.5229  -8.211 4.61e-16 ***
## GDP.on.health        1.0295     0.8671   1.187  0.23526
## Measles.cases       -0.2689     1.3592  -0.198  0.84318
## BMI                  0.1748     0.4425   0.395  0.69296
## Deaths.under.5.yo    0.1916     1.6490   0.116  0.90753
## Total.expenditure    3.0693     0.5427   5.656 1.85e-08 ***
## DPT.vacc             1.9654     0.4027   4.881 1.17e-06 ***
## HIV.AIDS           -18.6138     0.8681 -21.443  < 2e-16 ***
## Thin.5.to.9.yo      -0.9241     0.7622  -1.213  0.22551
## Inc.comp.resource   30.4499     1.1041  27.579  < 2e-16 ***
```

```
## Years.in.School     -3.3378     1.2944  -2.579  0.01001 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.246 on 1523 degrees of freedom
## Multiple R-squared:  0.8896, Adjusted R-squared:  0.8887
## F-statistic: 944.3 on 13 and 1523 DF,  p-value: < 2.2e-16
```

Thus, the 'R^2' factor is around 0.89, indicating that the model about to be built will have a very good performance. It is also clear that the most significant columns are Development.Status, Adult.Mortality, Alcohol.consume, Total.expenditure, DPT.vacc, HIV.AIDS, Inc.comp.resource and Years.in.School. Therefore, the ML model will be created by using only those variables:

```
model_1 <- lm(Life.expectancy ~ Development.Status + Adult.Mortality
                  + Alcohol.consume + Total.expenditure + DPT.vacc
                  + HIV.AIDS + Inc.comp.resource + Years.in.School,
                data = trainSet)
summary(model_1)
```

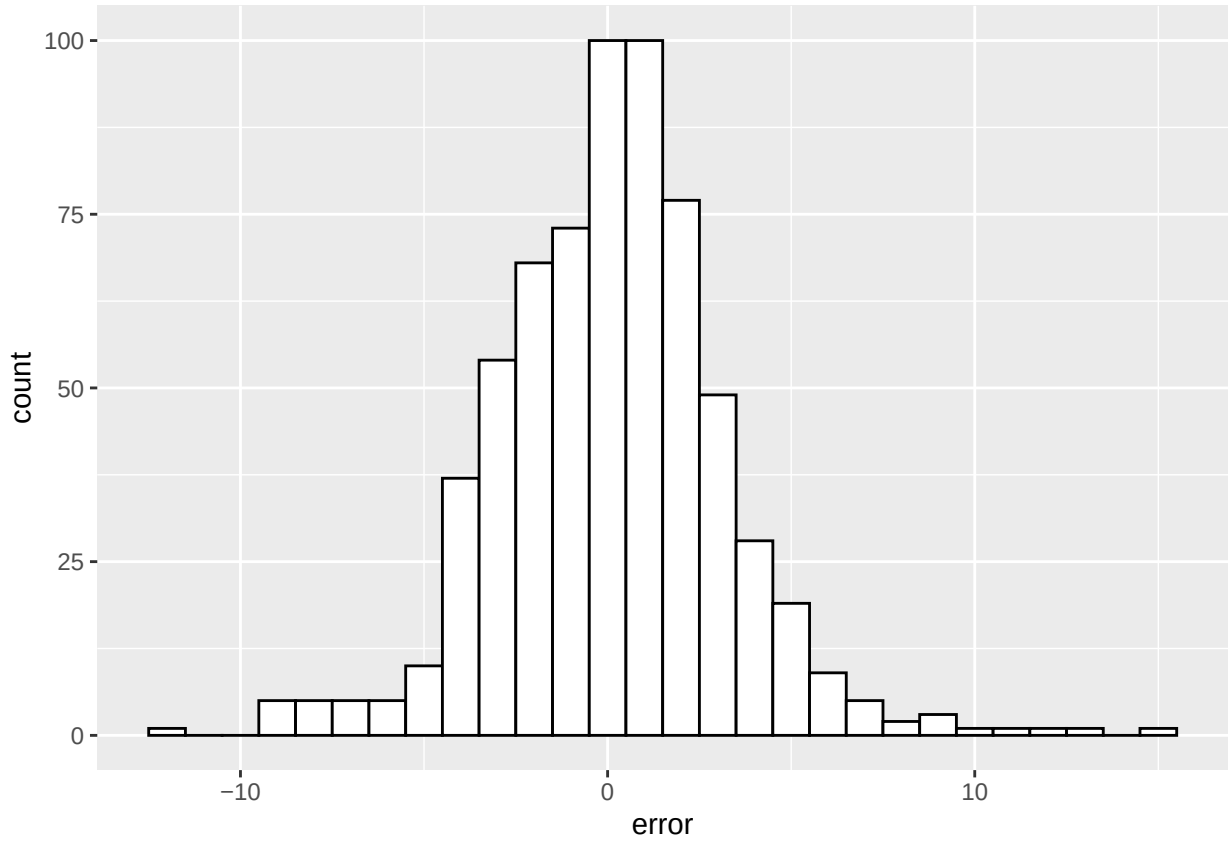
```
##
## Call:
## lm(formula = Life.expectancy ~ Development.Status + Adult.Mortality +
##     Alcohol.consume + Total.expenditure + DPT.vacc + HIV.AIDS +
##     Inc.comp.resource + Years.in.School, data = trainSet)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.4413  -1.8982  -0.0004   1.9398  12.0832
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         53.6267     0.4515 118.768  < 2e-16 ***
## Development.Status   1.0591     0.2879   3.678 0.000243 ***
## Adult.Mortality     -8.2268     0.6406 -12.842  < 2e-16 ***
## Alcohol.consume     -4.1790     0.5151  -8.113 1.00e-15 ***
## Total.expenditure    3.2790     0.5318   6.166 8.93e-10 ***
## DPT.vacc             1.9238     0.4005   4.803 1.72e-06 ***
## HIV.AIDS           -18.6239     0.8626 -21.591  < 2e-16 ***
## Inc.comp.resource   31.0131     1.0450  29.678  < 2e-16 ***
## Years.in.School     -3.4134     1.2865  -2.653 0.008056 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.245 on 1528 degrees of freedom
## Multiple R-squared:  0.8893, Adjusted R-squared:  0.8888
## F-statistic:  1535 on 8 and 1528 DF,  p-value: < 2.2e-16
```

The results above indicate that, even though the ML model takes into consideration only part of the variables, the factor `R^2` remains almost unaltered.

The next step is to apply this model on the testing set and, then, to analyze both its precision and its residues:

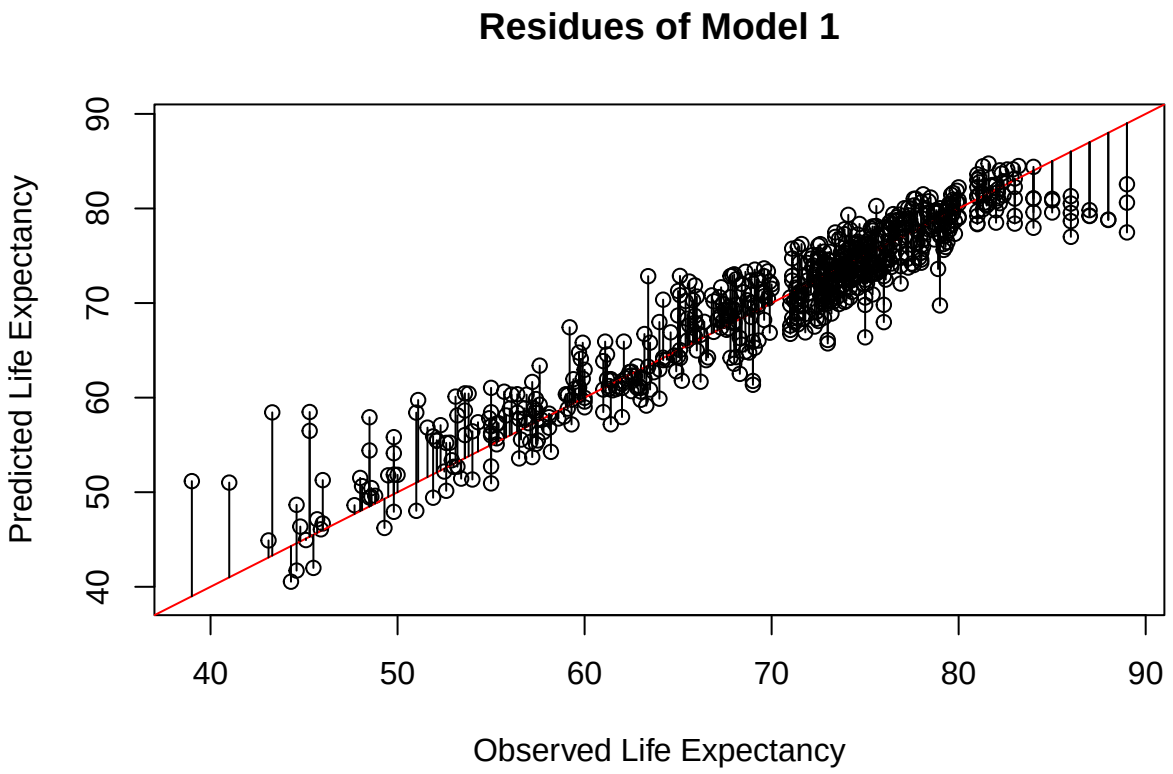
```
prediction = data.frame(predict(model_1, testSet, interval = 'confidence'))
score_1 <- data.frame(actual = testSet$Life.expectancy,
                      prediction = prediction$fit)
score_1 <- mutate(score_1, error = prediction - actual)
ggplot(score_1, aes(x = error)) +
```

```
  geom_histogram(binwidth = 1, fill = 'white', color = 'black')
```



Above, one can visualize how the residues are distributed. The difference between the predicted and observed values for life expectancy is concentrated around zero, even though there are some discrepancies. The residues can also be visualized on the figure below:

```
res <- -score_1$error
pred <- score_1$prediction
obs <- score_1$actual
var_range <- range(pred, obs)
plot(obs, pred,
    xlim = var_range, ylim = var_range,
    xlab = "Observed Life Expectancy",
    ylab = "Predicted Life Expectancy",
    main = "Residues of Model 1")
abline(0,1, col = "red")
segments(obs, pred, obs, pred + res)
```



Finally, in order for this model to compared with the next, the root mean squared error (RMSE) will be calculated:

```
RMSE <- sqrt(sum(score_1$error^2)/nrow(score_1))
RMSE
```

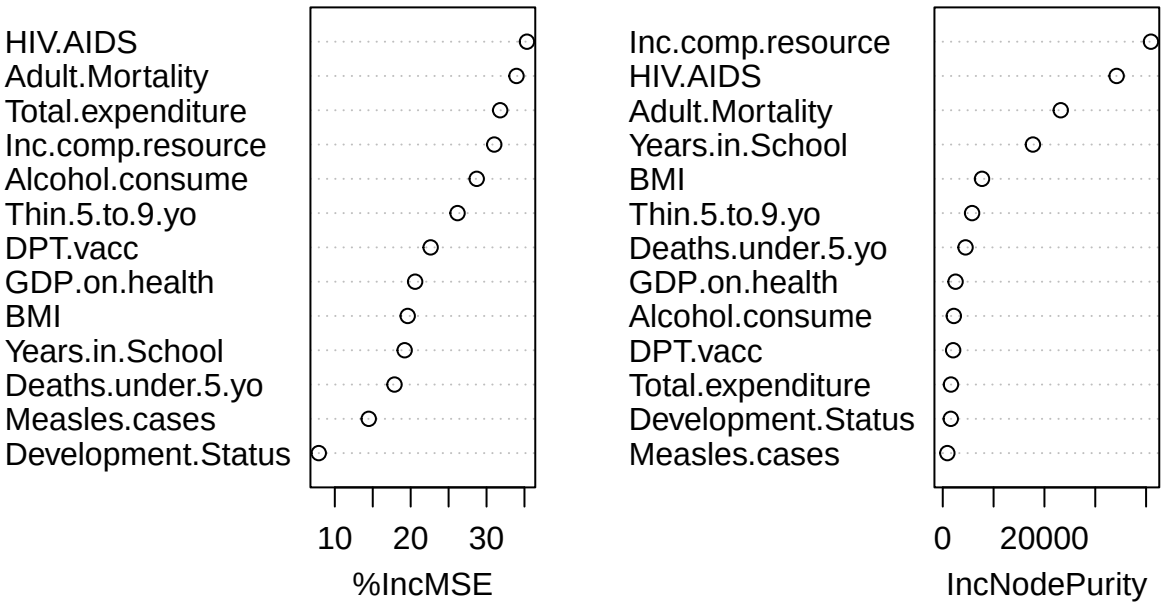
```
## [1] 3.120812
```

That concludes the discussion about the first ML model. Next, a new model will be created and, after that, both models will be compared to each other.

Model 2

As previously discussed, the variables GDP, Thin.10.to.19.yo and Polio.immune have a very high correlation coefficient (next to 1) regarding the columns GDP.on.health, Thin.5.to.9.yo and DPT.vacc, respectively. Removing those variables from the analysis, the relevance of the remaining columns is depicted on the following figure:

```
library(randomForest)
relevance <- randomForest(Life.expectancy ~. - Country - Year - GDP
                          - Thin.10.to.19.yo - Polio.immune,
                          data = trainSet,
                          ntree = 500,
                          nodesize = 4,
                          importance = T)
varImpPlot(relevance)
```



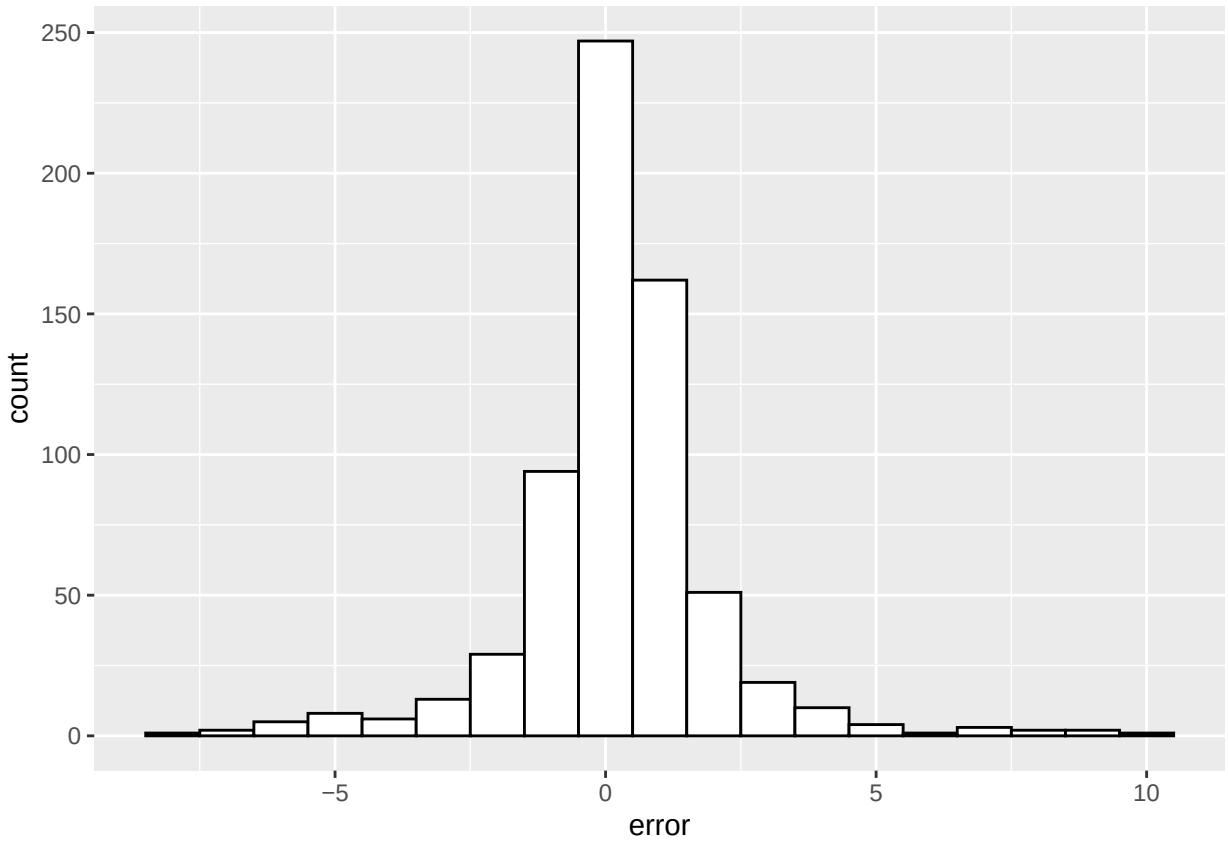
The column Development.Status is among the least important according to both criteria. Therefore, it will not be used for the construction of the ML model:

```
model_2 <- randomForest(Life.expectancy ~. - Country - Year
                        - GDP - Thin.10.to.19.yo - Polio.immune
                        - Development.Status,
                        data = trainSet,
                        ntree = 500,
                        nodesize = 4)
model_2
```

```
##
## Call:
##  randomForest(formula = Life.expectancy ~ . - Country - Year -      GDP - Thin.10.to.19.yo - Polio.i
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##           Mean of squared residuals: 3.937954
##                     % Var explained: 95.84
```

The results above show that the model created has an `R^2` value around 0.96, a very high value. Next, this model will be applied to the test data:

```
prediction = predict(model_2, testSet)
score_2 <- data.frame(actual = testSet$Life.expectancy,
                      predicted = prediction)
score_2 <- mutate(score_2, error = predicted - actual)
ggplot(score_2, aes(x = error)) +
  geom_histogram(binwidth = 1, fill = 'white', color = 'black')
```



Therefore, most of the residues values are next to zero - way more than for the first model. Another perspective regarding the residues is depicted next:

```
res <- (-1)*score_2$error
pred <- score_2$predicted
obs <- score_2$actual
var_range <- range(pred, obs)
plot(obs, pred,
    xlim = var_range, ylim = var_range,
    xlab = "Observed Life Expectancy",
    ylab = "Predicted Life Expectancy",
    main = "Residuals of Model 2")
abline(0,1, col = "red")
segments(obs, pred, obs, pred + res)
```

Finally, the last step is to calculate the RMSE for the second model:

```
RMSE <- sqrt(sum(score_2$error^2)/nrow(score_2))
RMSE
```

```
## [1] 1.841463
```

Therefore, the RMSE is considerably smaller for this model when compared with the previous one.

Conclusion

Among the variables in the data set, the column HIV.AIDS is the one with the strongest negative correlation with life expectancy, while the variables Years.in.School and Inc.comp.resource have the most significant positive correlation (again, that does not indicates causality). Still, other variables also have considerable influence on life expectancy and were instrumental in building both ML models.

Another important point is that the two Machine Learning models created had very good results: the first had a `R^2` value around 0.89 and a RMSE between 3.0 and 3.2, while for the second one those values were about 0.96 and between 1.9 and 2.0. Thus, one can say that the second model is superior to the first.

It is very interesting to notice how a study as simple as this one could be used to establish guidelines regarding public policies. In fact, it is indisputable how a through data analysis was able to provide a clearer picture of the problem discussed here, from the relation between the data until the understanding about the effect that each variable has on the target variable. This shows how Data Analysis and Machine Learning models are essential tools for companies to adopt during their decision making processes.