Prevendo a Expectativa de Vida em um País

Leonardo Ribeiro

13/10/2021

Expectativa de Vida em um País

O objetivo deste texto é discutir a criação de um modelo de Machine Learning (ML) que seja capaz de determinar a expectativa de vida em um país, dadas as seguintes variáveis:

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 litres of pure alcohol)
percentage expenditure	Expenditure on health as a percene 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 percene 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)

Para isso, será usado um conjunto de dados retirado do Kaggle, um site que provê dados sobre diferentes tópicos e que frequentemente patrocina competições voltadas à Ciência de Dados. O dataset original pode

ser encontrado **neste link**.

O Problema

Criar um modelo de ML que, com base em uma série de dados de um país, indique a expectativa de vida nele.

Carregando os Dados

O primeiro passo será carregar o arquivo com os dados de interesse:

```
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"
```

O conjunto de dados possui 2938 linhas e 22 colunas (estas representam dados dos países). Os nomes das colunas serão alterados para facilitar a interpretação do dataset:

```
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)
```

A expectativa de vida (atributo alvo deste estudo) é indicada pelos valores presentes coluna “Life.expectancy”. Portanto, o papel do modelo de ML será prever os valores desta coluna.

Análise Exploratória

O primeiro passo da análise será lidar com os valores “NA” do dataset em questão.

```
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"
```

As colunas 'HepB.immune' e 'Population' tem um número muito alto de NA's. Por esse motivo, elas serão removidas do dataset e, apenas após isso, as linhas restantes com valores 'NA' serão descartadas:

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

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

Além de valores 'NA', alguns valores '0' também podem causar problemas para o modelo de Machine Learning.

```
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"
```

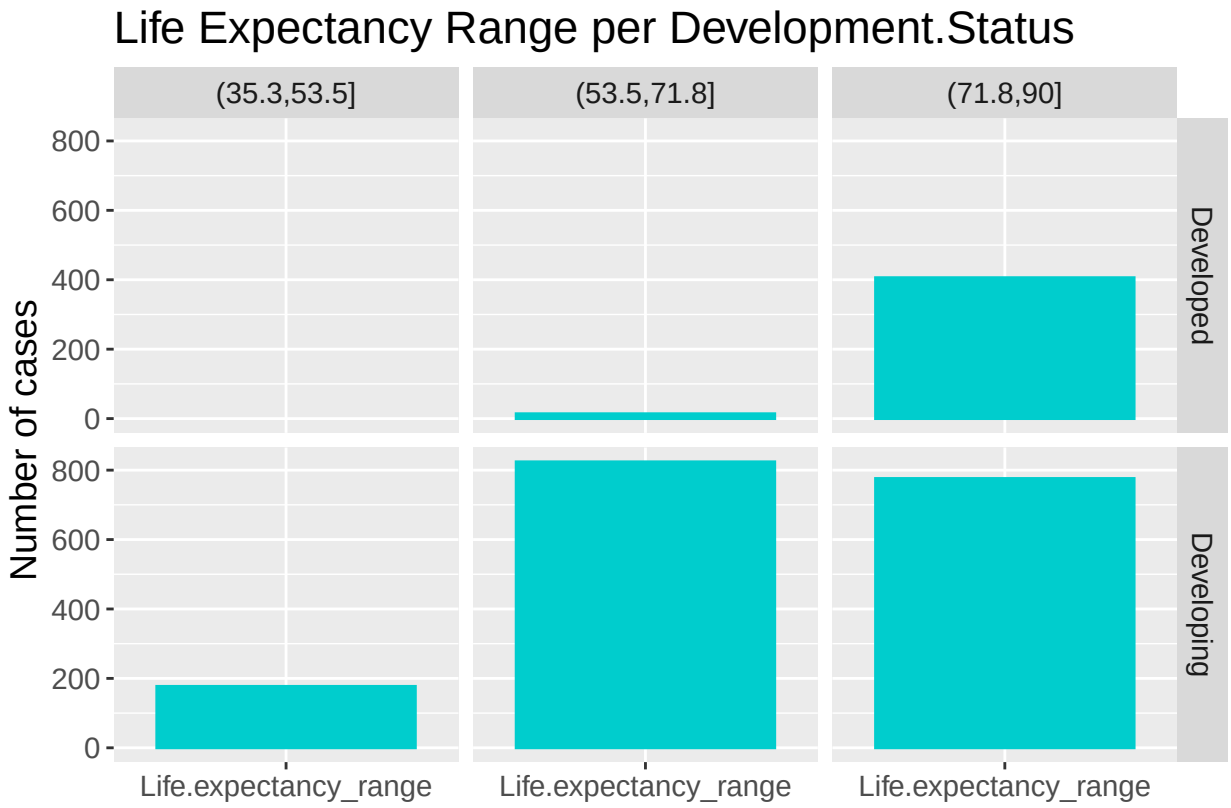
```
raw_data <- raw_data[raw_data$Inc.comp.resource != 0,]
raw_data <- raw_data[raw_data$Years.in.School != 0,]
dim(raw_data)
```

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

É importante ressaltar que a decisão de remover valores '0' mencionada acima foi tomada após a avaliação da correlação entre as variáveis preditoras 'Inc.comp.resource' e 'Years.in.School' com a coluna alvo 'Life.expectancy'. Para ambas as variáveis, os zeros eram outliers indesejados.

Ainda falando sobre correlação, a relação entre a variável categórica 'Development.Status' e a expectativa de vida é representada na seguinte figura:

```
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')
```



O gráfico acima mostra que a expectativa de vida em países desenvolvidos é, em sua maioria, alta (maior que 72 anos). Para países em desenvolvimento, ela se divide principalmente entre as faixas de 53 a 71 e 72 a 90, embora haja um número considerável deste grupo na faixa mais baixa. Isso permite a conclusão de que há uma grande desigualdade entre os países em desenvolvimento e que os países desenvolvidos tendem a ser mais homogêneos no que diz respeito à expectativa de vida.

Além da variável categórica mencionada, o dataset usado também contém uma série de variáveis numéricas que influenciam os valores encontrados na coluna Life.expectancy. As que possuem coeficiente de correlação com maior módulo são apontadas a seguir:

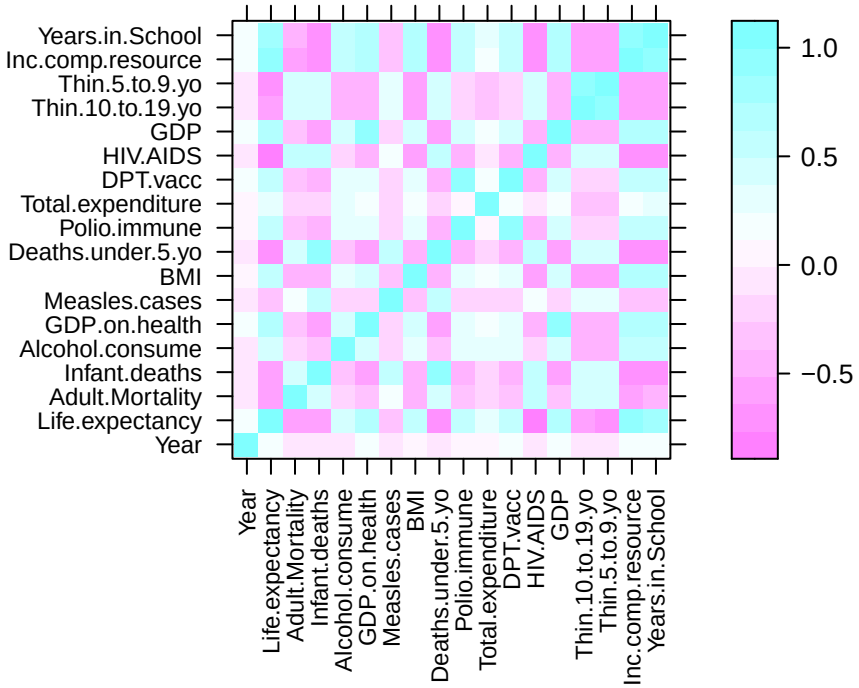
```
cors <- de.build_corr_coef_list(raw_data[,-c(1, 3, 21)], 'spearman')
de.get_corr_coef_w_target_var(data.frame(cors), 'Life.expectancy', 0.7)
```

```
## [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"
```

Logo, o coeficiente de correlação com a variável alvo deste estudo é maior para as colunas 'HIV.AIDS', 'Inc.comp.resource' e 'Years.in.School'. Ademais, o gráfico a seguir permite a representação visual destes e dos demais coeficientes de correlação:

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

Correlation between variables using the method spearman

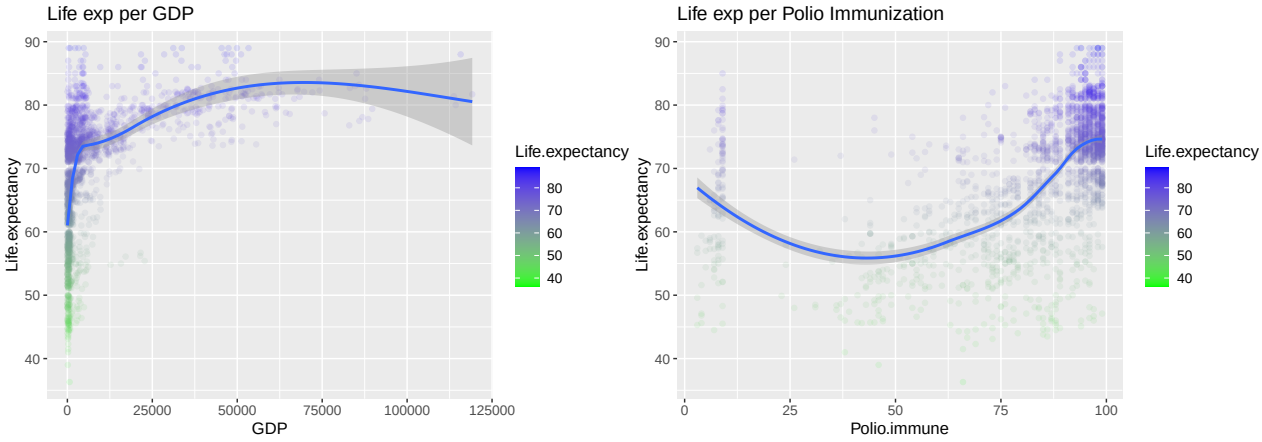


Com base na rede de correlação, é possível notar que as colunas "Years.in.School", "Inc.comp.resource", "GDP" e "Polio.immune" estão entre as que tem uma correlação positiva com a expectativa de vida. Os seguintes gráficos retratam essa correlação:

```
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)
```



A primeira conclusão que pode ser tirada é que, em países onde as pessoas passam mais tempo na escola, a expectativa de vida é sensivelmente maior. O mesmo pode ser dito para países com boa distribuição de renda: locais em que o índice 'Income composition of resources' é grande também têm uma expectativa de vida maior (com uma correlação praticamente linear).

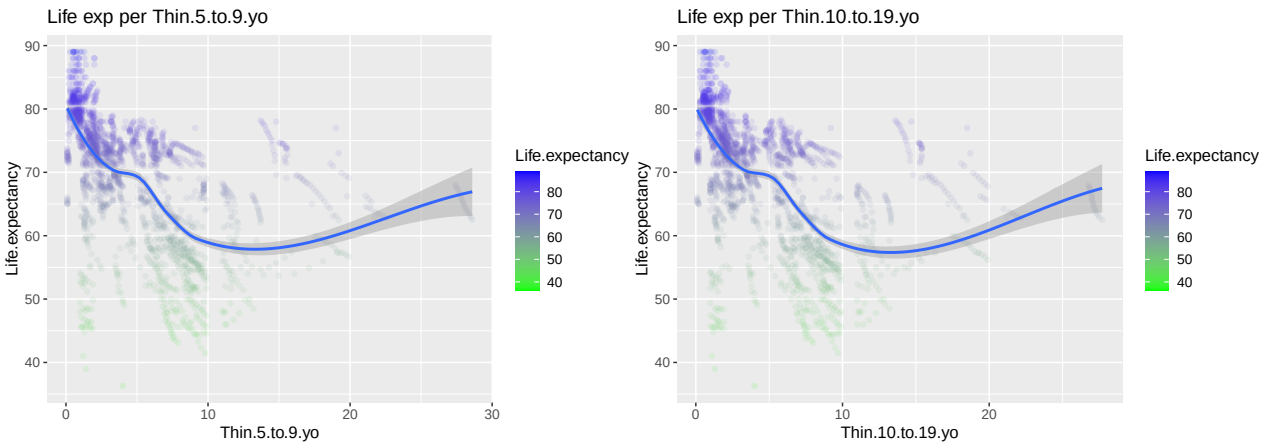
Ainda, em sua região mais estatisticamente significante, a subida no índice GDP também parece associada a um aumento na expectativa de vida. Com relação ao índice de imunização de Poliomelite, algo curioso acontece: um número considerável de países com alta expectativa de vida têm baixos índices de vacinação. Isso pode estar associado a um descuido relacionado a erradicação da doença neles. Excluindo essa região do gráfico, o aumento da imunização da Pólio também acompanha um aumento de expectativa de vida.

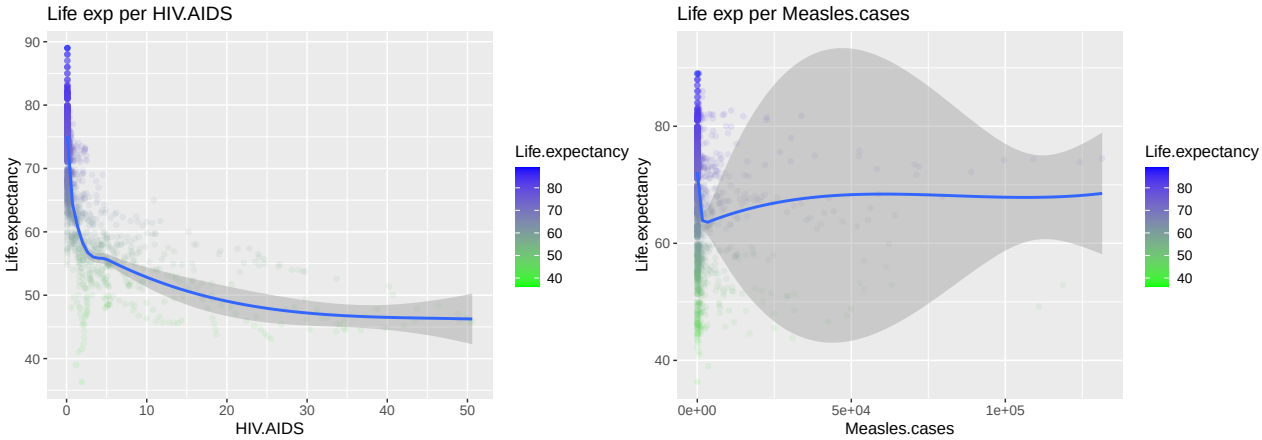
Em seguida, vamos analisar as seguintes colunas (que tem uma correlação negativa com a expectativa de vida): "Thin.5.to.9.yo", "Thin.10.to.19.yo", "HIV.AIDS" e "Measles.cases".

```
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)
```





Neste segundo conjunto de gráficos, os dois primeiros são bem semelhantes: tratam como a desnutrição em diferentes faixas etárias influencia na expectativa de vida. Quando as taxas de desnutrição aumentam, a expectativa de vida decresce.

Ainda, focando mais uma vez na região estatisticamente significante, um aumento relacionado aos valores das colunas 'HIV.AIDS' e 'Measles.cases' parecenm relacionados a uma brusca queda na expectativa de vida.

Com essas análises feitas, é sempre importante ressaltar o seguinte ponto: correlação não indica causalidade. Embora a correlação possa ser uma pista, um estudo mais aprofundado precisaria ser feito para que fosse possível apontar a existência de uma relação de causa e efeito entre duas variáveis.

Manipulação de Dados

O conjunto de dados usado neste trabalho possui uma variável categórica, o Development.Status. Para lidar com esse tipo de variável em um modelo de regressão, será usada a técnica de dummy coding.

```
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')
```

Agora, a coluna Development.Status também tem caráter numérico, com o valor '1' simbolizando países desenvolvidos e '0' indicando países em desenvolvimento.

Outra ação que deve ser tomada é a normalização das variáveis: cada coluna segue uma escala diferente, o que permite que seus valores tenham amplitudes diversas. Para normalizar a distribuição desses dados, será usada a técnica de normalização:

```
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
```

O último passo dessa análise exploratória será estudar a relação entre as variáveis preditoras:

```
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"
```

Observando esses resultados, fica evidente que as colunas 'Infant.deaths' e 'Deaths.under.5.yo' tratam, essencialmente, do mesmo dado. Para evitar overfitting, a primeira será removida.

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

Um argumento semelhante poderia ser usado para descartar outras colunas: por exemplo, fica claro que países com maior GDP gastam um percentual maior desse GDP em saúde, porém são medidas diferentes; países em que a campanha de vacinação para poliomelite é mais ampla também aplicam mais a vacina tríplice; quando a desnutrição atinge a faixa etária de 5 aos 9, ela costuma se propagar para a faixa de 10 aos 19; por fim, cidadãos de países com melhor distribuição de renda costumam passar mais anos na escola. A decisão de remover essas colunas será tomada mais a frente.

Agora que as variáveis estão normalizadas, com o tipo correto e as devidas colunas foram removidas, a próxima etapa é a construção de um modelo de Machine Learning.

Construção de modelo de ML

Neste trabalho, serão criados dois modelos de ML: um usando a função lm() e outro usando a função randomForest(). Em ambos os casos, o primeiro passo é o mesmo - a divisão do dataset em dados de teste e dados de treino:

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

Ao dataset 'trainSet', foram atribuídos 70% dos dados selecionados até este ponto. Os 30% restantes estão no dataset 'testSet'.

Modelo 1 - lm()

Para o modelo linear, a relevância das variáveis presentes no dataset (excluindo as que tem uma forte correlação entre si) é dada por:

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

```
##
## Call:
## lm(formula = Life.expectancy ~ . - Country - Year - GDP - Polio.immune -
##     Thin.10.to.19.yo, data = trainSet)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.5567  -1.8834  -0.0365   1.9401  12.1072
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         53.2077     0.5355  99.352  < 2e-16 ***
## Development.Status   0.8673     0.2969   2.921  0.00354 **
## Adult.Mortality     -8.1798     0.6506 -12.573  < 2e-16 ***
## Alcohol.consume     -3.9063     0.5197  -7.516 9.57e-14 ***
## GDP.on.health        1.1817     0.9017   1.310  0.19024
## Measles.cases       -0.3518     1.5141  -0.232  0.81631
## BMI                  0.4147     0.4366   0.950  0.34240
```

```
## Deaths.under.5.yo   -1.9748     1.4822  -1.332  0.18295
## Total.expenditure    3.4734     0.5400   6.432 1.68e-10 ***
## DPT.vacc             2.1614     0.4023   5.372 8.98e-08 ***
## HIV.AIDS           -19.5182     0.8990 -21.711  < 2e-16 ***
## Thin.5.to.9.yo       0.5116     0.7357   0.695  0.48690
## Inc.comp.resource   31.1175     1.1182  27.828  < 2e-16 ***
## Years.in.School     -3.8282     1.3311  -2.876  0.00408 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.252 on 1523 degrees of freedom
## Multiple R-squared:  0.8895, Adjusted R-squared:  0.8886
## F-statistic: 943.2 on 13 and 1523 DF,  p-value: < 2.2e-16
```

Nota-se que o fator `R^2` é de cerca de 0.89, o que indica uma performance muito boa. Além disso, as variáveis mais significantes são: Development.Status, Adult.Mortality, Alcohol.consume, Total.expenditure, DPT.vacc, HIV.AIDS, Inc.comp.resource e Years.in.School. O modelo de ML será criado usando apenas este conjunto:

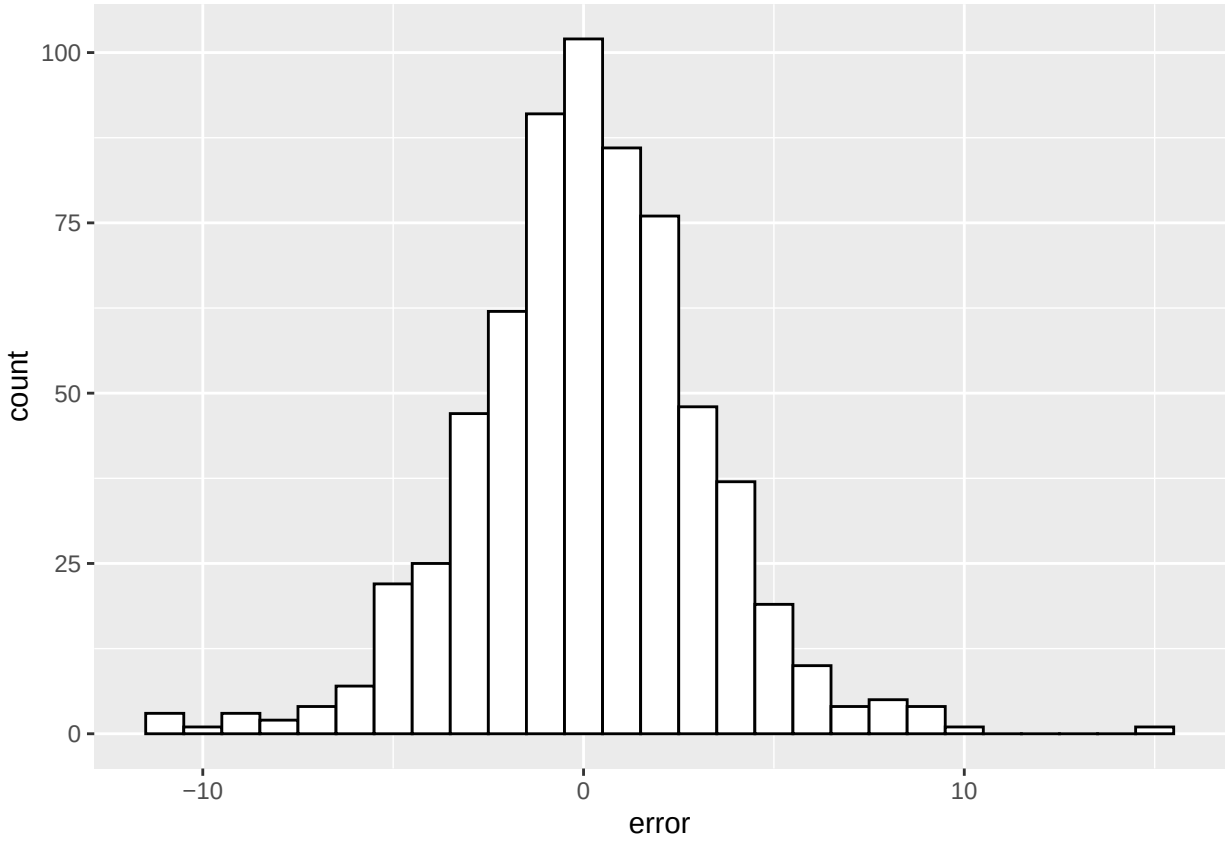
```
modelo_lm <- lm(Life.expectancy ~ Development.Status + Adult.Mortality
                  + Alcohol.consume + Total.expenditure + DPT.vacc
                  + HIV.AIDS + Inc.comp.resource + Years.in.School,
                data = trainSet)
summary(modelo_lm)
```

```
##
## 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
## -19.4755  -1.8686  -0.0679   1.9438  12.2355
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         53.1324     0.4643 114.443  < 2e-16 ***
## Development.Status   0.9376     0.2841   3.300  0.00099 ***
## Adult.Mortality     -8.1696     0.6492 -12.585  < 2e-16 ***
## Alcohol.consume     -3.9965     0.5106  -7.827 9.29e-15 ***
## Total.expenditure    3.6275     0.5288   6.860 9.95e-12 ***
## DPT.vacc             2.1827     0.3998   5.460 5.55e-08 ***
## HIV.AIDS           -19.3471     0.8922 -21.686  < 2e-16 ***
## Inc.comp.resource   31.6377     1.0579  29.905  < 2e-16 ***
## Years.in.School     -3.8370     1.3234  -2.899  0.00379 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.253 on 1528 degrees of freedom
## Multiple R-squared:  0.8891, Adjusted R-squared:  0.8885
## F-statistic:  1532 on 8 and 1528 DF,  p-value: < 2.2e-16
```

Os dados acima indicam que modelo de ML, apesar de levar em conta apenas as variáveis mais relevantes, obteve um fator `R^2` bem próximo ao do modelo de avaliação, que usava o dataset completo.

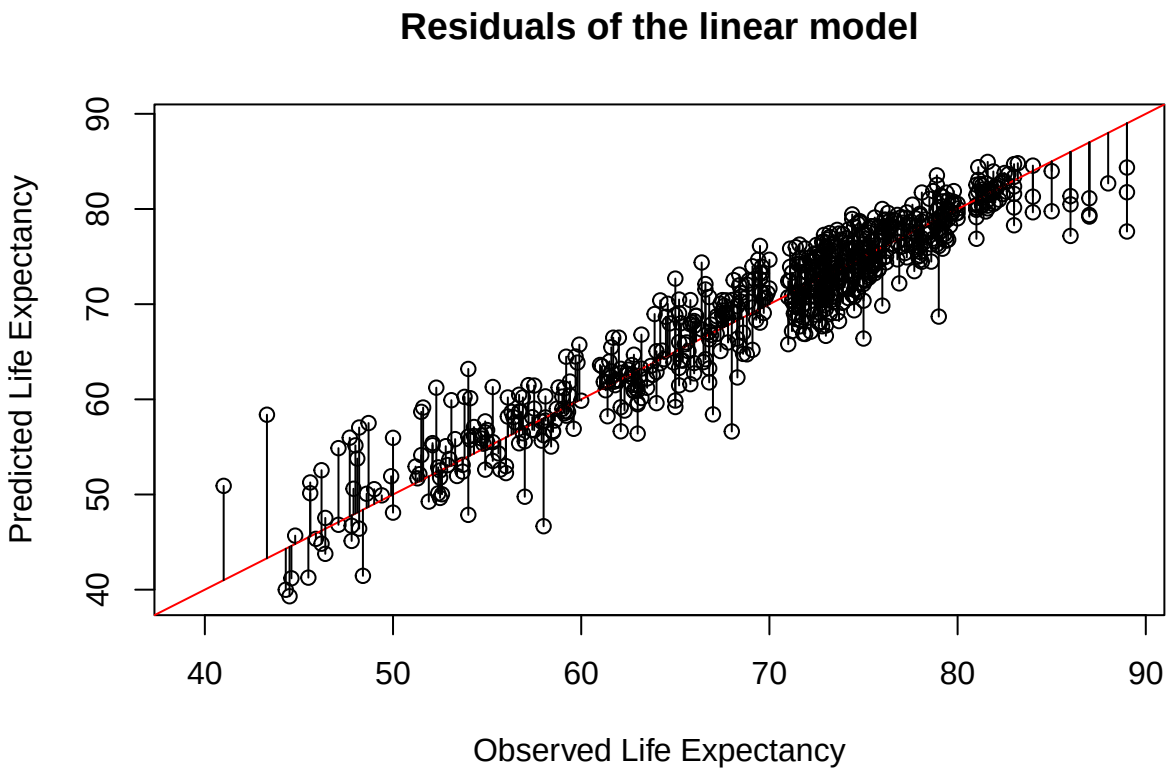
O passo seguinte é analisar os resíduos do modelo criado:

```
previsao = data.frame(predict(modelo_lm, testSet, interval = 'confidence'))
score_lm <- data.frame(actual = testSet$Life.expectancy,
                       prediction = previsao$fit)
score_lm <- mutate(score_lm, error = prediction - actual)
ggplot(score_lm, aes(x = error)) +
  geom_histogram(binwidth = 1, fill = 'white', color = 'black')
```



O gráfico acima indica a distribuição dos resíduos, isto é, a diferença entre o valor observado e o valor previsto. Nota-se que os resíduos estão concentrados em torno do zero, embora haja algumas discrepâncias. Estas discrepâncias podem ser visualizadas de outra maneira no gráfico a seguir:

```
res <- -score_lm$error
pred <- score_lm$prediction
obs <- score_lm$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 the linear model")
abline(0,1, col = "red")
segments(obs, pred, obs, pred + res)
```



Por fim, para obter uma medida definitiva, será calculada a raiz quadrada do resíduo quadrado médio:

```
rmse <- sqrt(sum(score_lm$error^2)/nrow(score_lm))
rmse
```

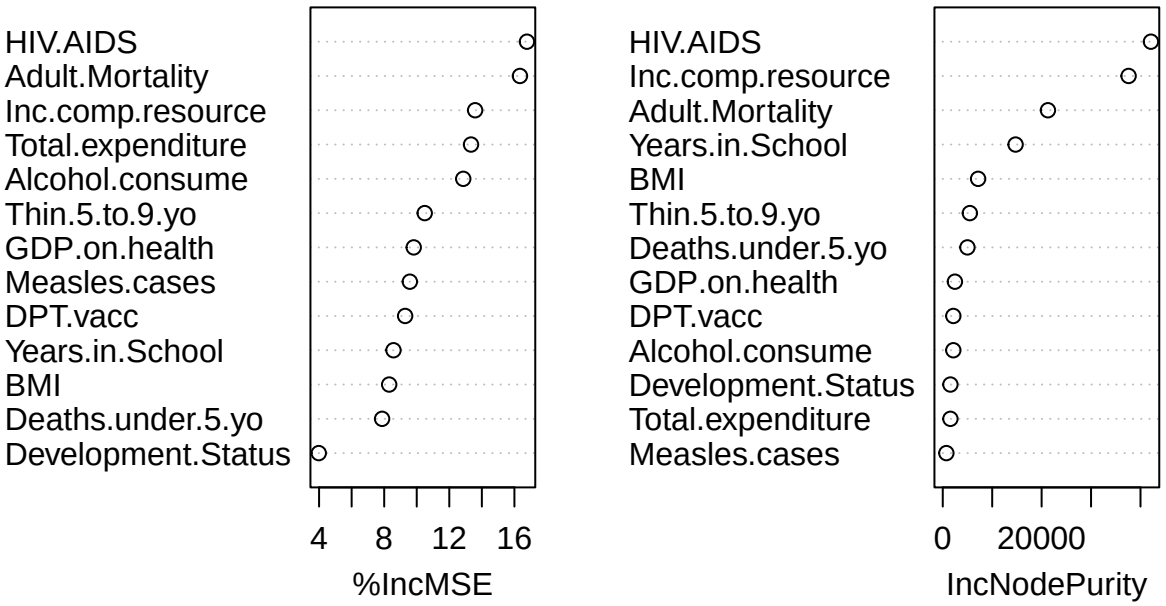
```
## [1] 3.096662
```

Isso encerra a discussão sobre o primeiro modelo de Machine Learning deste texto. Em seguida, um novo modelo será criado com base na função randomForest.

Modelo 2 - randomForest()

Como destacado anteriormente, as variáveis GDP, Thin.10.to.19.yo e Polio.immune têm um coeficiente de correlação bastante alto (próximo a 1) quando comparadas com GDP.on.health, Thin.5.to.9.yo e DPT.vacc, respectivamente. Retirando essas variáveis da análise, a importância das demais variáveis preditivas fica da seguinte forma:

```
library(randomForest)
Importância <- randomForest(Life.expectancy ~. - Country - Year - GDP
                            - Thin.10.to.19.yo - Polio.immune,
                            data = trainSet,
                            ntree = 100,
                            nodesize = 10,
                            importance = T)
varImpPlot(Importância)
```



Observando a figura acima, nota-se que, em ambos os critérios usados, as variáveis Measles.cases e Development.Status têm a menor relevância. Por esse motivo, ambas serão descartadas na construção do modelo:

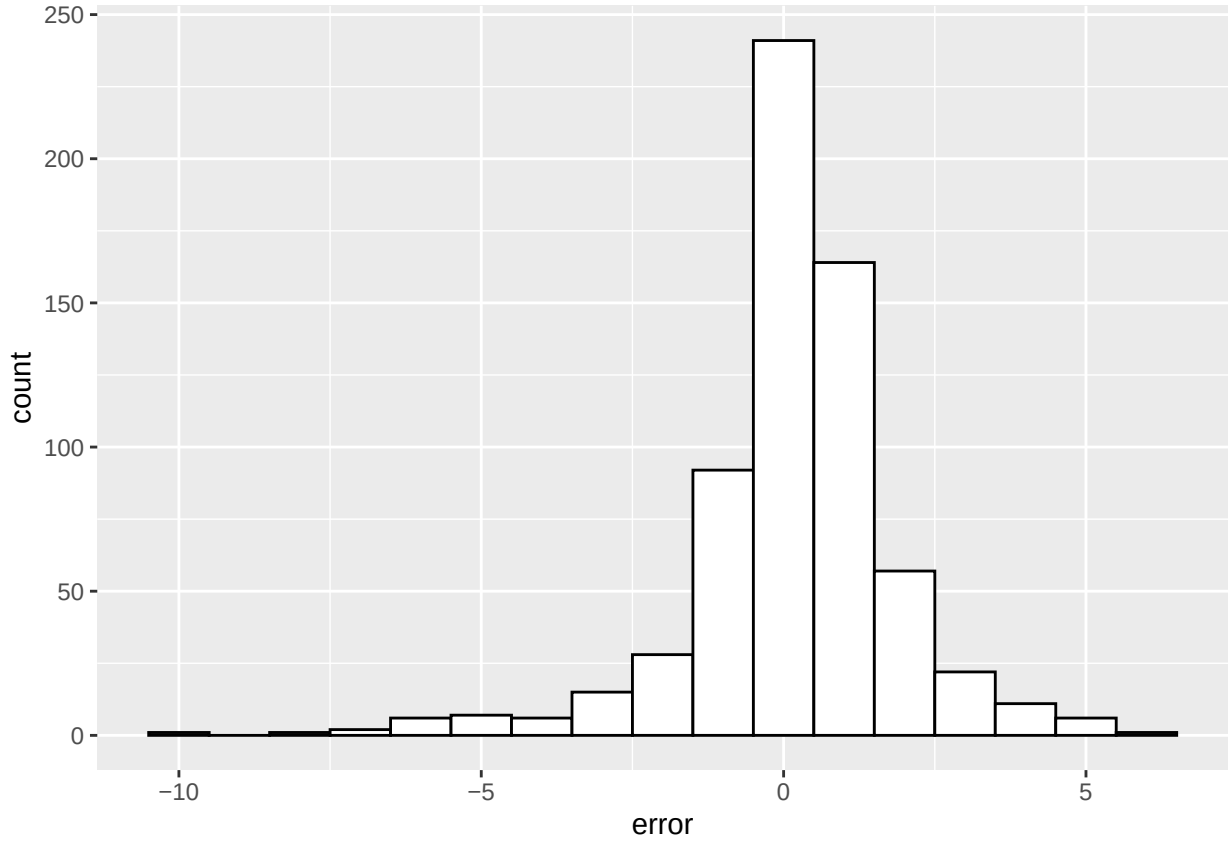
```
modelo_rf <- randomForest(Life.expectancy ~. - Country - Year
                          - GDP - Thin.10.to.19.yo - Polio.immune
                          - Development.Status - Measles.cases,
                          data = trainSet,
                          ntree = 500,
                      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

```
##
## 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: 3
##
##           Mean of squared residuals: 3.907702
##                     % Var explained: 95.88
```

O modelo criado tem `R^2` de cerca 0,96, um valor bastante alto. O passo seguinte é aplicar esse modelo ao dataset reservado para testes:

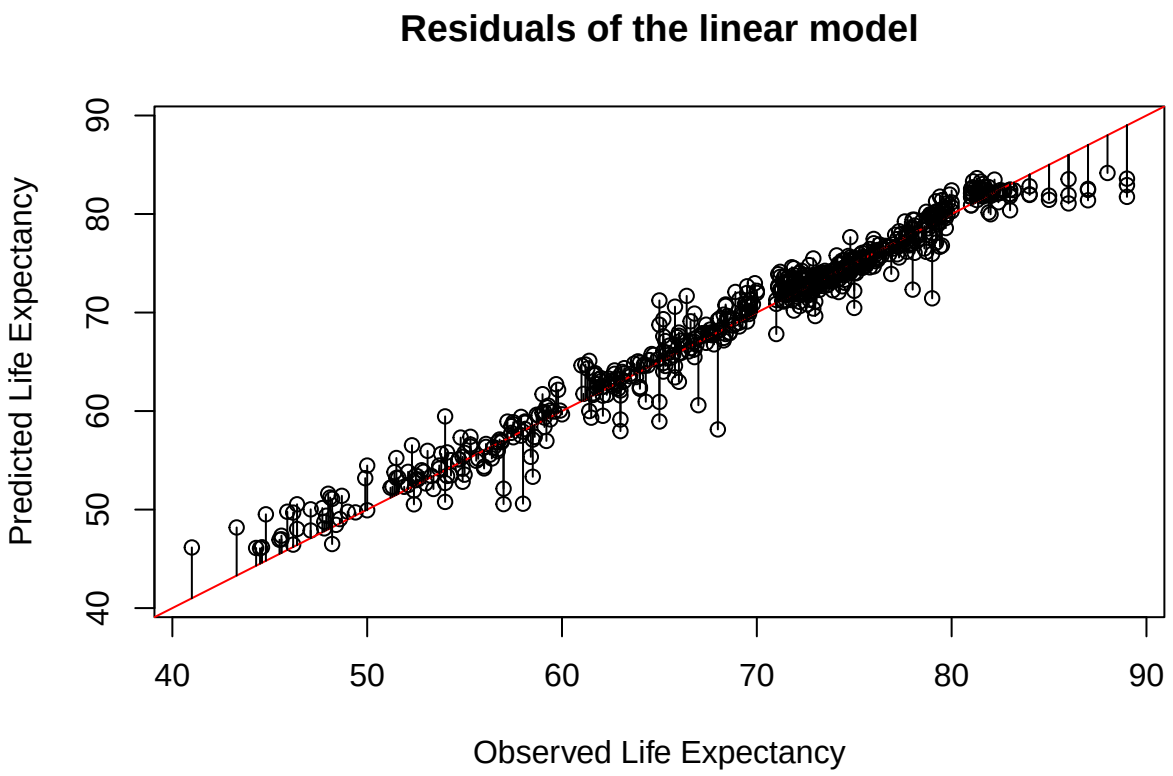
```
previsao = predict(modelo_rf, testSet)
score_rf <- data.frame(actual = testSet$Life.expectancy,
                       prediction = previsao)
score_rf <- mutate(score_rf, error = prediction - actual)
ggplot(score_rf, aes(x = error)) +
```

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



O gráfico acima mostra a distribuição de resíduos para o modelo atual. Novamente, os resíduos concentram-se na região em torno do zero. Uma outra perspectiva sobre os resíduos pode ser obtida com o gráfico a seguir:

```
res <- (-1)*score_rf$error
pred <- score_rf$prediction
obs <- score_rf$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 the linear model")
abline(0,1, col = "red")
segments(obs, pred, obs, pred + res)
```



Por fim, para que esse modelo possa ser comparado com o anterior, novamente será feito o cálculo da raiz quadrada do resíduo quadrado médio:

```
rmse <- sqrt(sum(score_rf$error^2)/nrow(score_rf))
rmse
```

```
## [1] 1.74033
```

Portanto, a raiz do resíduo médio quadrado obtido no segundo modelo é inferior ao obtido no primeiro modelo.

Conclusão

Entre as variáveis usadas neste texto, a coluna HIV.AIDS é a que tem a correlação negativa de maior intensidade com expectativa de vida em um país, enquanto as colunas Years.in.School e Inc.comp.resource são as que têm uma maior influência positiva (mais uma vez: isso não estabelece uma relação de causalidade). Outras variáveis tem efeitos menores, porém ainda consideráveis sobre a variável alvo deste estudo.

Outro ponto importante é que ambos os modelos de Machine Learning criados tiveram resultados muito bons: o primeiro teve um `R^2` de cerca de 0,89 e uma rrmq entre 3 e 3,2, enquanto para o segundo esses valores foram de (aproximadamente) `R^2 = 0,96` e rrmq entre 1,9 e 2,0 . Dessa forma, é possível afirmar que o modelo criado com o auxílio da função randomForest é mais assertivo.

É interessante notar como estudos simples como este podem ser usados para o estabelecimento de diretrizes em políticas públicas. Desde a relação entre variáveis até o entendimento do impacto que elas tem na variável alvo, a análise de dados e a posterior construção de modelos de Machine Learning mostram-se essenciais no processo de tomada de decisões.