PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Ciência de Dados e *Big Data*

Débora Anson Lima

CONSTRUÇÃO DE MODELO DE APRENDIZADO DE MÁQUINA PARA PREDIÇÃO DE INDICADOR DE FELICIDADE (*LIFE LADDER*)

Belo Horizonte
2022

Débora Anson Lima

CONSTRUÇÃO DE MODELO DE APRENDIZADO DE MÁQUINA PARA PREDIÇÃO DE INDICADOR DE FELICIDADE (*LIFE LADDER*)

Trabalho de Conclusão apresentado ao Curso de Especialização em Ciência de Dados e *Big Data* como requisito parcial à obtenção do título de especialista.

Belo Horizonte

2022

SUMÁRIO

1 Introdução

1.1 Contextualização

Diversos indicadores de considerável importância são utilizados para a mensuração do grau de desenvolvimento econômico, educacional, sanitário e social de um país, contudo, para a identificação do nível de felicidade da população com uma abrangência geográfica ampla, é possível destacar somente o *Life Ladder*.

Esse indicador é apurado e apresentado anualmente[1] pelo *World Happiness Report*[2] (Relatório da Felicidade Mundial, *WHR* doravante) da *Sustainable Development Solutions Network* (Rede de Soluções de Desenvolvimento Sustentável) da Organização das Nações Unidas.

Conforme será detalhado na próxima subseção, a apuração do *Life Ladder* advém da *Gallup World Poll* (*GWP*), realizada anualmente (desde 2005) pelo Instituto *Gallup*, tendo uma abrangência sobre os países bastante significativa[3]. A partir dessa pesquisa, uma amostra de cidadãos de cada local é questionada diretamente sobre o seu nível de felicidade, definindo-se, assim, com a média das respostas nacionais, qual o grau de felicidade da população de cada um dos países participantes.

Nesse contexto, se busca, aqui, criar modelos de aprendizado de máquina (*machine learning*) que possam predizer, com uma precisão razoável, qual o patamar de felicidade de cada país utilizando-se, para tanto, a base da *GWP* e dados relativos ao desenvolvimento socioeconômico nacionais.

1.2 O problema proposto

O objetivo deste trabalho é a criação de um modelo de aprendizado de máquina que preveja o indicador *Life Ladder* associado a cada país.

---

[1] Desde 2012.

[2] Disponível, para todas as suas edições, em http://worldhappiness.report/.

[3] Para o ano de 2020, foram ranqueados 149 países/territórios.

Conforme mencionado acima, a pontuação desse indicador advém da *GWP*, sendo ela a média da resposta nacional à principal pergunta de avaliação de vida feita na pesquisa: a da escada *Cantril*[4]. Nessa questão, os participantes são chamados a imaginar sua atual posição de vida em uma escada com degraus numerados, em ordem crescente, de 0 a 10, onde o topo representa a melhor vida possível e a parte inferior, a pior vida possível. Abaixo, segue transcrição da pergunta realizada pelo Instituto (já traduzida):

> Por favor, imagine uma escada, com degraus numerados de 0 na parte inferior a 10 na parte superior. O topo da escada representa a melhor vida possível para você e a base da escada representa a pior vida possível para você. Em qual degrau da escada você diria que pessoalmente sente que está neste momento?

Assim, o valor do indicador *Life Ladder* é baseado, exclusivamente, na percepção dos próprios indivíduos sobre suas vidas.

Para tentar compreender quais os fatores podem contribuir para a definição do nível de felicidade informado pelas pessoas (na escala de 0 a 10), o *WHR* apresenta seis variáveis-chave[5] para todo o período analisado (de 2005 a 2019[6]):

- PIB *per capita* (em forma logarítmica): a grandeza PIB (Produto Interno Bruto) se refere ao total dos bens e serviços produzidos dentro de um país em determinado ano equivalendo, assim, ao somatório dos valores adicionados pelas diversas atividades econômicas acrescidos dos respectivos tributos. O PIB *per capita*, consequentemente, é o resultado da divisão do PIB pela população do país, informando quanto dele caberia a cada indivíduo caso fosse dividido igualitariamente. Os valores constantes do *WHR* 2021 foram obtidos a partir da atualização de 14/10/2020 dos Indicadores de Desenvolvimento do Banco Mundial (*World Development Indicators*), sendo que os de Taiwan, Síria, Palestina, Venezuela, Djibuti e Iêmen provieram da tabela 9.1 da *Penn World Tables* (*PWT*[7]).

---

[4] Assim designada em razão da Escala Cantril *Self-Anchoring*, desenvolvida pelo Dr. Hadley Cantril, pioneiro pesquisador social.

[5] Cujas definições e forma de apuração constam do *Statistical Appendix for Chapter 2 of World Happiness Report 2021*, disponível em https://worldhappiness.report/ed/2021/#appendices-and-data, acesso em 03/07/2021.

[6] Dado o início da pandemia da COVID-19 em 2020, bem como a indisponibilidade de dados dos quatro indicadores adicionais descritos na sequência para este ano, optou-se por se trabalhar, aqui, somente com as informações apresentadas para o período de 2005 a 2019 (o que evita a ocorrência de vieses no *Life Ladder* decorrentes da nova situação estabelecida).

[7] A *PWT* é um banco de dados com informações sobre níveis relativos de renda, produção, insumo e produtividade desenvolvido e mantido por acadêmicos da Universidade da Califórnia em Davis e *do Groningen Growth Development Center* da Universidade de Groningen.

- Expectativa de vida saudável ao nascer: baseada nas informações extraídas do repositório de dados do Observatório de Saúde Global da Organização Mundial da Saúde (OMS). Tendo em vista que, quando da elaboração do *WHR* 2021, os anos com informações disponíveis eram somente os de 2000, 2005, 2010, 2015 e 2016, o Relatório utilizou as técnicas de interpolação e extrapolação para atingir os números necessários para todo o intervalo existente.
- Suporte social: essa variável corresponde à média nacional das respostas (binária – 0 ou 1) à outra pergunta realizada na *GWP* que procura identificar se a pessoa possui alguém em quem se apoiar em momentos de dificuldade. Expressamente, a pergunta é: "Se você estiver com problemas, você tem parentes ou amigos com quem pode contar para ajudá-lo sempre que você precisar deles, ou não?".
- Liberdade para fazer escolhas de vida: para se chegar a essa medida, é utilizada a média nacional das respostas à seguinte pergunta da *GWP*: "Você está satisfeito ou insatisfeito com sua liberdade de escolher o que fazer com sua vida?". Ou seja, a pergunta procura verificar se a pessoa entende possuir um nível de liberdade/autonomia suficiente para decidir sobre sua própria vida.
- Generosidade: essa variável é calculada pelo resíduo da regressão da média nacional de resposta à seguinte pergunta da *GWP:* "Você doou dinheiro para uma instituição de caridade no mês passado?".
- Percepções de corrupção: corresponde à média nacional das respostas (binárias – 0 ou 1) a mais duas perguntas da *GWP*: "A corrupção está disseminada por todo o governo ou não?" e "A corrupção está disseminada dentro das empresas ou não?". Para calculá-la, é utilizada apenas a média das duas respostas (0 ou 1). No caso de inexistir resposta à primeira questão, usa-se a resposta da segunda como se fosse a percepção geral.

Além dessas seis variáveis iniciais, o *WHR* traz para o rol outras duas que se referem ao afeto positivo e negativo percebidos pelas pessoas em suas vidas:

- Afeto positivo: é definido como a média de três medidas[8] de afeto positivo aferidas na *GWP*: felicidade, prazer e riso. Essas são quantificadas por intermédio das respostas às perguntas a seguir (considerando-se a primeira como duas dada sua abrangência): (1) "Você experimentou os sentimentos de felicidade e prazer/diversão durante grande parte do dia de ontem?"; e (2) "Você sorriu ou riu muito ontem?".

[8] A partir de 2013/2014, o afeto positivo passou a ser definido apenas como a média de prazer/diversão e de riso dada a limitação de dados sobre felicidade.

- Afeto negativo: é definido como a média de três medidas de afeto negativo na *GWP*: preocupação, tristeza e raiva. Tais como as de efeito positivo, essas são quantificadas por intermédio das respostas à pergunta a seguir (considerando-se ela como três dada sua abrangência sobre os sentimentos mencionados): "Você experimentou os sentimentos de preocupação, tristeza e raiva durante grande parte do dia de ontem?".

Nesse contexto, conforme se verifica da tabela apresentada no *WHR* 2020[9] e transcrita abaixo[10], apesar do afeto positivo e do afeto negativo, em geral, não poderem ser explicados pelas seis variáveis iniciais da mesma forma como o *Life Ladder* pode[11], o afeto positivo possui um impacto significativo na definição desse indicador:

Tabela 1 – Regressões *WHR* 2020

Variáveis independentes	**Variáveis dependentes**			
	Life Ladder	Afeto positivo	Afeto negativo	*Life Ladder*
PIB *per capita* (log)	0,310 (0,066)***	-0,009 (0,01)	0,008 (0,008)	0,324 (0,065)***
Suporte social	2,362 (0,363)***	0,247 (0,048)***	-0,336 (0,052)***	2,011 (0,389)***
Expectativa de vida saudável ao nascer	0,036 (0,01)***	0,001 (0,001)	0,002 (0,001)	0,033 (0,009)***
Liberdade para fazer escolhas de vida	1,199 (0,298)***	0,367 (0,041)***	0,084 (0,040)**	0,522 (0,287)*
Generosidade	0,661 (0,275)**	0,135 (0,030)***	0,024 (0,028)	0,390 (0,273)
Percepções de corrupção	-0,646 (0,297)***	0,020 (0,027)	0,097 (0,024)***	-0,720 (0,294)**
Afeto positivo	- -	- -	- -	1,944 (0,355)***
Afeto negativo	- -	- -	- -	0,379 (0,425)

Fonte: *WHR* 2020 (tabela 2.1).
Notas:
- os coeficientes foram calculados com erros padrão robustos, os quais constam discriminados entre parênteses;
- as observações ***, ** e * indicam significância estatística nos níveis de 1%, 5% e 10% respectivamente.

[9] Apesar desse tipo de análise não ter sido apresentada no *WHR* 2021 (o qual focou nas mudanças decorrentes da pandemia de COVID-19), as regressões para explicação do *Life Ladder* e dos afetos positivo e negativo foram, igualmente, apresentadas nos *WHR* de anos anteriores (2019, 2018 e 2017 por exemplo), tendo sido destacadas, neles, as mesmas conclusões que constam do *WHR* 2020 em relação aos papéis dessas duas últimas variáveis sobre o *Life Ladder*. Conforme já informado acima, todas as edições do *WHR* se encontram disponíveis na página http://worldhappiness.report/.

[10] Cujos coeficientes das regressões discriminadas foram calculados pelo método dos Mínimos Quadrados Ordinários (MQO).

[11] Conforme se pode apurar dos coeficientes e das significâncias expressas na coluna 2 da tabela (relativa à regressão explicativa do *Life Ladder*) em comparação com as informações constantes das colunas 3 e 4 (referentes às regressões que têm o afeto positivo e o afeto negativo como variáveis dependentes).

Assim, tendo em vista a magnitude do coeficiente associado ao afeto positivo e sua significância estatística (expressos na coluna 5, linhas 15 e 16, da tabela), o Relatório mostra que essa variável possui um papel importante nas avaliações feitas pelos indivíduos sobre suas vidas. Adicionalmente, informa que muito do impacto das variáveis relacionadas ao suporte social, à liberdade para fazer escolhas e à generosidade nessas avaliações é traduzido por meio de sua influência no afeto positivo (conforme se depreende das suas expressivas significâncias na explicação do afeto positivo, as quais podem ser observadas na coluna 3, linhas 6, 10 e 12, da tabela).

Paralelamente, o *WHR* destaca que o afeto negativo não possui impacto sobre a definição do *Life Ladder* de acordo com o que demonstra a insignificância da variável para a explicação deste indicador na respectiva regressão (explicitada pela observação constante da coluna 5, linha 18, da tabela).

Diante de tais conclusões do *WHR*, para a construção dos modelos de aprendizado de máquina deste trabalho são consideradas, inicialmente, todas as oito variáveis[12] por ele apresentadas, sendo, após, as relativas ao afeto positivo e negativo retiradas[13] de forma a se confirmar sua significância/insignificância na determinação do *Life Ladder*. Esses modelos constam desenvolvidos nos subitens 5.1 a 5.3 do *notebook*[14] associado a este trabalho.

Importa destacar, novamente, que as classificações de felicidade decorrentes da *GWP* não são baseadas em nenhuma dessas variáveis, sendo elas formadas, exclusivamente, pelas avaliações dos próprios indivíduos sobre suas vidas conforme suas respostas à pergunta da escada *Cantril*. Assim, no *WHR*, essas variáveis são utilizadas somente para tentar explicar a origem dos diferentes níveis de felicidade entre os países.

Além disso, na tentativa de melhorar os modelos obtidos a partir das variáveis elencadas pelo próprio *WHR*, foram buscados outros indicadores socioeconômicos para compor a base de predição. Dessa pesquisa, foram selecionadas as seguintes medidas para inserção na análise:

- Índice de Educação (*Education Index – EI)*: representa a média dos anos de escolaridade dos adultos e dos anos esperados de escolaridade das crianças, ambos expressos como um índice obtido pela escala com os máximos correspondentes. A

[12] No *dataframe* 1, discriminado na seção 5 deste trabalho.

[13] Nos *dataframes* 2 e 3, descritos, também, na seção 5.

[14] Conforme descrição constante da seção 3 apresentada na sequência.

base de dados para a geração desse índice é mantida pelo Instituto de Estatística da *United Nations Educational, Scientific and Cultural Organization – UNESCO*.

- Índice de Desenvolvimento de Gênero (*Gender Development Index – GDI*): calculado pela razão entre os valores do Índice de Desenvolvimento Humano (IDH) feminino e masculino[15]. O IDH é uma medida multidimensional do progresso social que surgiu como um contraponto ao PIB *per capita*[16], tendo sido criado no âmbito do Programa das Nações Unidas para o Desenvolvimento (PNUD). No IDH, é considerada não apenas a perspectiva da renda, mas também a da educação e a da saúde.
- Vunerabilidade do Emprego (*Vulnerable Employment – VE*): representa o percentual de pessoas que trabalham de forma autônoma ou em estabelecimentos familiares pertencentes a algum parente que resida no mesmo domicílio que elas. A fonte dos dados para definição dessa variável é a base *ILOSTAT*[17] da Organização Internacional do Trabalho (*International Labour Organization – ILO*).
- Jovens que não estudam e nem estão empregados (*Youth Not in School or Employment – YNSE*): se refere ao percentual de pessoas de 15 a 24 anos que não estudam (ou realizam qualquer treinamento), nem trabalham. A fonte dos dados para esse indicador é a mesma que a do item anterior: a base *ILOSTAT* da Organização Internacional do Trabalho.

Os dados desses quatro indicadores foram coletados no *site* dos Relatórios de Desenvolvimento Humano[18] (*Human Development Reports*) do Programa de Desenvolvimento das Nações Unidas (PNUD – *United Nations Development Programme*), tendo sido eles obtidos pelo Programa conforme descrições e fontes mencionadas acima.

Os modelos obtidos[19] a partir da inserção desses indicadores para a predição do *Life Ladder* são apresentados nos subitens 5.4 a 5.7 do *notebook* associado a este trabalho.

---

[15] Para uma melhor compreensão da forma de cálculo do GDI, consultar http://hdr.undp.org/sites/default/files/hdr2020_technical_notes.pdf.

[16] Que é uma das variáveis elencadas pelo próprio *WHR* conforme discriminado anteriormente neste Relatório.

[17] Disponível na página https://ilostat.ilo.org/data – acesso realizado em 21/07/2021.

[18] No endereço eletrônico http://hdr.undp.org/en/indicators, acessado em 01/09/2021.

[19] Com base nos *dataframes* 4, 5, 6 e 7, descritos na seção 5 deste trabalho.

2 Coleta de dados

Conforme sintetizado na seção anterior, os modelos de aprendizado de máquina a serem apresentados neste trabalho foram criados a partir da manipulação dos seguintes *datasets*[20]:

Tabela 2 – *Datasets* originais e fontes de dados

Dataset	**Nome do arquivo**	***Link***	**Data de acesso**
World Happiness Report 2021	DataPanelWHR2021C2.xls	https://worldhappiness.report/ed/2021/#appendices-and-data	14/07/2021
Education Index	Education index.csv	http://hdr.undp.org/en/indicators/103706#	01/09/2021
Gender Development Index (GDI)	Gender Development Index (GDI).csv	http://hdr.undp.org/en/indicators/137906#	01/09/2021
Vulnerable Employment (% of total employment)	Vulnerable employment (% of total employment).csv	http://hdr.undp.org/en/indicators/43006#	01/09/2021
Youth not in school or employment (% ages 15-24)	Youth not in school or employment (% ages 15-24).csv	http://hdr.undp.org/en/indicators/147906#	01/09/2021

Fonte: elaboração própria.

Preliminarmente, é necessário destacar que, apesar de se analisarem somente os dados relativos ao período de 2005 a 2019 aqui, o *WHR* utilizado como fonte de dados é o do ano de 2021 dado que, nele, constam expressos os dados de 2019 para um maior número de países[21].

Após o *download* dos *datasets* listados, foram eles carregados no *Excel* para uma análise prévia dos dados. Dessa avaliação, foram identificadas, nos quatro últimos, linhas e colunas que continham informações alheias ao indicador propriamente (tais como a fonte de dados ou a posição de cada país no IDH), que não interessavam ao trabalho. Diante disso, foram elas descartadas e, na sequência, gravados os arquivos em formato xlsx apenas com os

[20] Os quais constam descritos, também, no item 2 do *notebook* criado no ambiente *Colab*, para o desenvolvimento dos modelos deste trabalho.

[21] Na base de dados do *WHR* 2021, constam as informações de 2019 para 144 países e, na do *WHR* 2020, para 138 somente.

dados de cada país/ano em relação, propriamente, ao indicador. Por fim, os cinco arquivos abaixo foram incluídos no repositório do *github* para utilização neste trabalho:

Tabela 3 – *Datasets* adaptados no *github*

Dataset	**Nome do arquivo no *github***
World Happiness Report 2021	3_DataPanelWHR2021C2.xls
Education Index	4_Education.xlsx
Gender Development Index	5_Gender.xlsx
Vulnerable Employment	6_Vulnerable_employment.xlsx
Youth not in school or employment	7_Youth_not_in_school.xlsx

Fonte: elaboração própria.

Na próxima seção, são descritos os passos efetuados para tratamento dos dados dos *dataframes* criados a partir desses arquivos.

3 Processamento dos dados

3.1 Considerações gerais

Para construção do código de programação neste trabalho, optou-se pela linguagem *Python* e pelo uso de um *notebook* criado no ambiente *Colab* do *Google*. Todos as etapas para a criação dos *dataframes* e dos modelos de aprendizagem de máquina aqui apresentados se encontram desenvolvidas nesse *notebook* (igualmente, inserido no repositório do *github*). Relativamente às bibliotecas utilizadas, destacam-se *Pandas* e *Pycaret*.

Na sequência, são discriminadas as tarefas[22] realizadas para tratamento dos dados e para a criação de cada um dos *dataframes* utilizados neste trabalho, todas executadas no item 3 do *notebook*.

[22] Nas quais as funções executadas são apresentadas no formato `função` e os arquivos utilizados e as colunas/variáveis dos *dataframes* referenciadas são identificadas entre aspas (“arquivo/coluna/variável”).

3.2 Criação dos *dataframes*

a) *Dataframe* 1 *(df1)*

O primeiro *dataframe* do *notebook (df1)* tem como base o *dataset* do *WHR 2021*, trazendo informações relativas ao *Life Ladder* dos diversos países, bem como às outras oito variáveis discriminadas na subseção 1.2 acima (quais sejam, PIB *per capita*, expectativa de vida saudável ao nascer, suporte social, liberdade para fazer escolhas de vida, generosidade, percepções de corrupção, afeto positivo e afeto negativo).

No subitem 3.1.2 do *notebook*, após a leitura do respectivo *dataset* (com a função `read_excel`) e a criação do *df1* (realizadas no subitem 3.1.1), de forma a permitir a junção dos dados de todos os *dataframes* a serem criados ao longo do trabalho, foi realizada a padronização da grafia dos nomes dos países constantes dele em relação à verificada nos demais *dataframes* (elaborados a partir dos outros quatro *datasets* utilizados neste trabalho[23]).

Cabe destacar que a padronização foi realizada dessa forma tendo em vista que, dentre os quatro últimos *datasets*, não foram localizadas grafias diferentes para os nomes dos mesmos países. Assim, foi necessária somente a consolidação prévia dos nomes constantes deles, a exclusão dos itens repetidos e a criação, por fim, do arquivo "Countries.xlsx", no qual se encontra discriminado o rol dos nomes dos países utilizados nesses *datasets*.

Dessa forma, para a padronização dos nomes dos países entre os *dataframes* utilizados para os modelos, esse arquivo foi carregado no *notebook* (sob o nome *dfcn*) para que fosse efetivada a comparação com a grafia utilizada no *df1* por meio da função personalizada `mismatch`. Antes da execução dessa, contudo, foram eliminados os espaços em branco nos nomes dos países tanto do *df1* quanto do *dfcn* por meio da função `replace`.

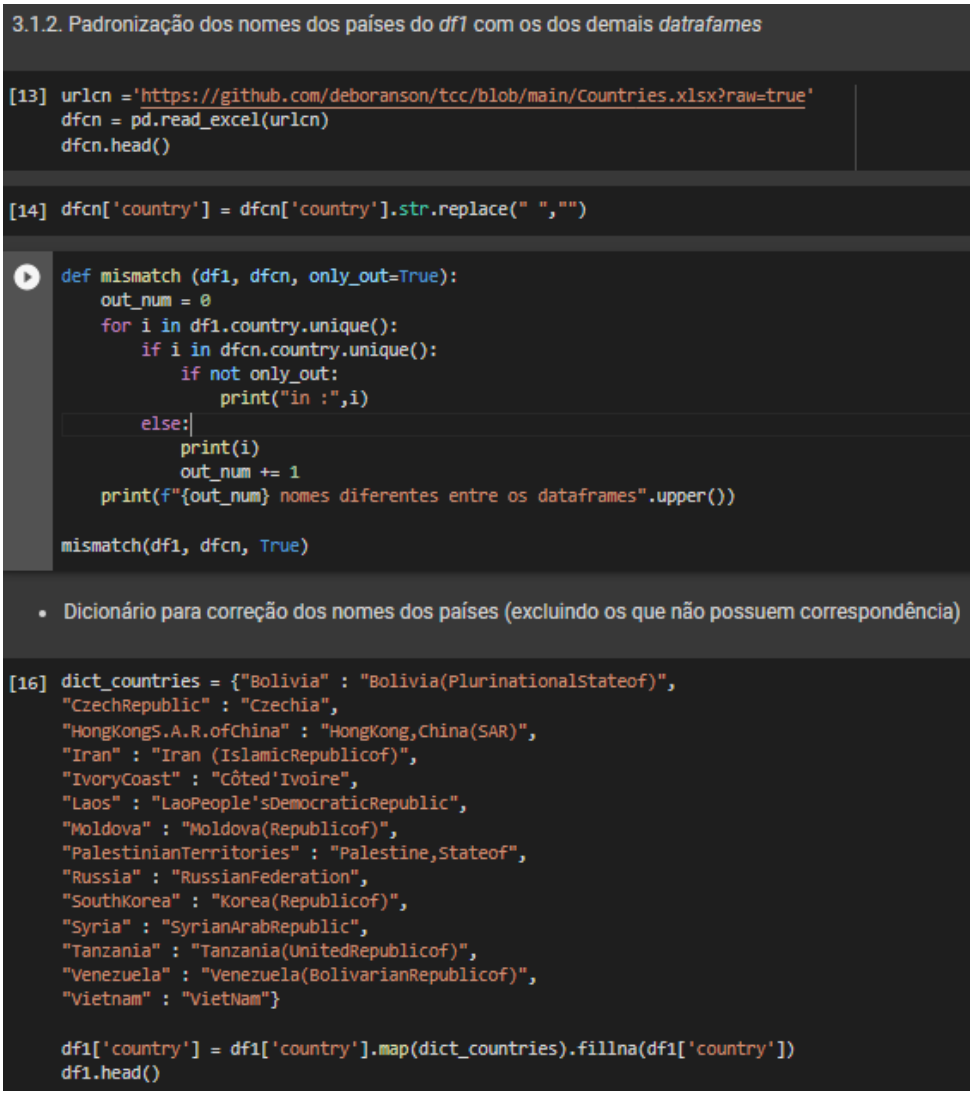
Nesse sentido, com a aplicação da função `mismatch`, foram listados os nomes dos países para os quais não se encontrou uma correspondência gráfica idêntica. Na sequência, a partir dessa lista, foi elaborado um dicionário[24] no *notebook* (*dict_countries*) relacionando-se a grafia constante do *df1* com a grafia padronizada definida no arquivo "Countries.xlsx"; por fim, o dicionário foi utilizado para adequar as divergências de grafia do *df1* por meio da função `map` do *Pandas*.

---

[23] Quais sejam, *Education Index, Gender Development Index, Vulnerable Employment* e *Youth not in school or employment.*
[24] Tendo sido descartados, nesse, os países para os quais não foi encontrada correspondência, de fato, no *dfcn*.

Abaixo, segue a relação dos comandos efetivados no *notebook* (sem as respectivas saídas) relativos às tarefas descritas acima:

Figura 1 – Comandos para padronização dos nomes dos países no *df1*



Fonte: elaboração própria conforme entradas no *notebook* associado.

Concluída essa etapa, já no subitem 3.1.3 do *notebook*, foram efetivadas as seguintes adequações no *df1:*

- eliminação das linhas com valores ausentes (`dropna`).
- criação da variável padronizada "*country-year*" para a efetivação das integrações de *dataframes* realizadas no trabalho, sendo ela, sempre, utilizada como critério das junções;

- alteração do índice do *dataframe* para a coluna "*country-year*";
- eliminação das colunas "*country"* e "*year*";

Abaixo, segue extrato do *notebook* com os comandos (sem as respectivas saídas) para a conclusão dessa fase:

Figura 2 – Comandos para adequações do *df*1

```
3.1.3. Adequações do df1

[17] df1.isna().sum()

[18] df1 = df1.dropna()

[19] df1['country-year'] = df1['country'] + df1['year'].astype(str)
     df1.head()

[20] df1.info()

[21] df1 = df1.set_index('country-year')
     df1.head()

[22] df1 = df1.drop(columns = ['country','year'])

[23] df1.info()
```

Fonte: elaboração própria conforme entradas no *notebook* associado.

Por fim, chega-se a um *dataframe* com as seguintes características:

Figura 3 – Informações do *df*1

```
df1.info()

<class 'pandas.core.frame.DataFrame'>
Index: 1708 entries, Afghanistan2008 to Zimbabwe2020
Data columns (total 9 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   Life Ladder                       1708 non-null   float64
 1   Log GDP per capita                1708 non-null   float64
 2   Social support                    1708 non-null   float64
 3   Healthy life expectancy at birth  1708 non-null   float64
 4   Freedom to make life choices      1708 non-null   float64
 5   Generosity                        1708 non-null   float64
 6   Perceptions of corruption         1708 non-null   float64
 7   Positive affect                   1708 non-null   float64
 8   Negative affect                   1708 non-null   float64
dtypes: float64(9)
memory usage: 133.4+ KB
```

Fonte: elaboração própria com base na saída do *notebook* associado.

b) *Dataframe* 2 *(df2)*

O segundo *dataframe* do *notebook (df2)* é, simplesmente, o *df1* sem uma de suas colunas, a do afeto negativo (*negative affect*). Essa transformação para a construção de uma nova modelagem é realizada de forma a se verificar a informação constante do *WHR* de que, ao contrário do afeto positivo, o negativo não teria um impacto significativo na definição do nível de felicidade traduzido pelo *Life Ladder.*

Dessa forma, no subitem 3.2 do *notebook,* foi realizada somente a eliminação da coluna "*Negative affect*" e criado o *df2*:

Figura 4 – Informações do *df2*

```
df2.info()

<class 'pandas.core.frame.DataFrame'>
Index: 1708 entries, Afghanistan2008 to Zimbabwe2020
Data columns (total 8 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   Life Ladder                       1708 non-null   float64
 1   Log GDP per capita                1708 non-null   float64
 2   Social support                    1708 non-null   float64
 3   Healthy life expectancy at birth  1708 non-null   float64
 4   Freedom to make life choices      1708 non-null   float64
 5   Generosity                        1708 non-null   float64
 6   Perceptions of corruption         1708 non-null   float64
 7   Positive affect                   1708 non-null   float64
dtypes: float64(8)
memory usage: 184.6+ KB
```

Fonte: elaboração própria com base na saída do *notebook* associado.

c) *Dataframe* 3 *(df3)*

Da mesma forma que é criado o *df2,* o *df3* surge da exclusão de mais uma coluna da base: o afeto positivo (*positive affect*). Seguindo no contexto discriminado na subseção anterior, essa alteração decorre do interesse de se averiguar a afirmação constante do *WHR* de que o afeto negativo não teria uma influência expressiva na definição do nível de felicidade representado pelo *Life Ladder*. Logo, com a eliminação de ambas as colunas dos afetos, pode ser efetivada a comparação do modelo baseado no *df1* com os modelos decorrentes do *df2* e do *df3,* verificando-se, então, se a retirada do afeto negativo melhora o modelo e, também,

qual o impacto da exclusão de ambos (concluindo-se, dessa forma, se seria favorável ou não manter somente o afeto positivo).

Assim, no subitem 3.3 do *notebook*, foi excluída a coluna "*Positive affect*" do *df2* e criado o *df3* (o qual possui as mesmas características do primeiro):

Figura 5 – Informações do *df3*

```
df3.info()

<class 'pandas.core.frame.DataFrame'>
Index: 1708 entries, Afghanistan2008 to Zimbabwe2020
Data columns (total 7 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   Life Ladder                       1708 non-null   float64
 1   Log GDP per capita                1708 non-null   float64
 2   Social support                    1708 non-null   float64
 3   Healthy life expectancy at birth  1708 non-null   float64
 4   Freedom to make life choices      1708 non-null   float64
 5   Generosity                        1708 non-null   float64
 6   Perceptions of corruption         1708 non-null   float64
dtypes: float64(7)
memory usage: 171.3+ KB
```

Fonte: elaboração própria com base na saída do *notebook* associado.

d) *Dataframe* 4 *(df4)*

Conforme descrito na subseção 1.2 acima, além das oito variáveis constantes do próprio *dataset* do *WHR* – as quais são introduzidas no relatório de forma a se verificar quais fatores podem contribuir para a definição do *Life Ladder* – neste trabalho, na tentativa de melhorar os modelos obtidos a partir dessas oito variáveis iniciais, são incluídos os seguintes indicadores (obtidos junto ao PNUD):

- Índice de Educação (*EI).*
- Índice de Desenvolvimento de Gênero (*GDI*).
- Vulnerabilidade do emprego (*VE).*
- Jovens que não estudam e nem estão empregados (*YNSE*).

Assim, a fim de se realizar um comparativo entre a influência das oito variáveis consideradas no *WHR* e a desses quatro indicadores adicionais na definição do *Life Ladder*, no subitem 3.4.1 do *notebook*, foi realizada uma adaptação do *df1* de forma a se manter somente a coluna com os dados do *Life Ladder* (criando-se o *df1_ad*) para, na sequência, com a introdução das informações referentes aos demais indicadores, ser criado o *df4.*

Após, no subitem 3.4.2 do *notebook*, são importados e manipulados os *datasets* relativos aos indicadores adicionais (provenientes do PNUD), criando-se, então, os *dataframes df_ei*, *df_gdi*, *df_ve* e *df_ynse*. Preliminarmente, tendo em vista a disposição dos dados nesses *datasets*, tiveram de ser efetivados alguns procedimentos para adequação ao padrão do *df1/df1_ad* (de forma a permitir as necessárias junções posteriores). Abaixo, segue o rol das tarefas efetuadas para a criação dos quatro *dataframes* mencionados:

- carregamento do *dataset* no *notebook* (com a função `read_excel`) e criação do respectivo *dataframe*;
- definição e execução da função `transform` com vistas a reposicionar os dados relativos aos anos (separados em colunas inicialmente) para linhas próprias, permitindo, assim, sua junção posterior com o *df1/df1_ad;*
- alteração do tipo de dado da coluna que contém o indicador propriamente para numérico (*float64)* – sendo essa executada somente quando o tipo de dado a exige;
- eliminação das linhas com valores ausentes (com a função `dropna`).
- criação da variável padronizada "*country-year"* (utilizada como critério para as junções dos *dataframes*);
- eliminação dos espaços em branco nos dados da coluna "*country-year"*;
- alteração do índice do *dataframe* para a coluna "*country-year"*;
- eliminação das colunas "*country"* e *"year"*;

Abaixo, segue extrato do *notebook* com os procedimentos acima mencionados (relativos à criação/manipulação de um dos *dataframes* criados – o *df_ei*, sem as respectivas saídas), bem como com as demais tarefas efetuadas apenas para fins de conferência das alterações realizadas*:*

Figura 6 – Comandos para criação do *df_ei*

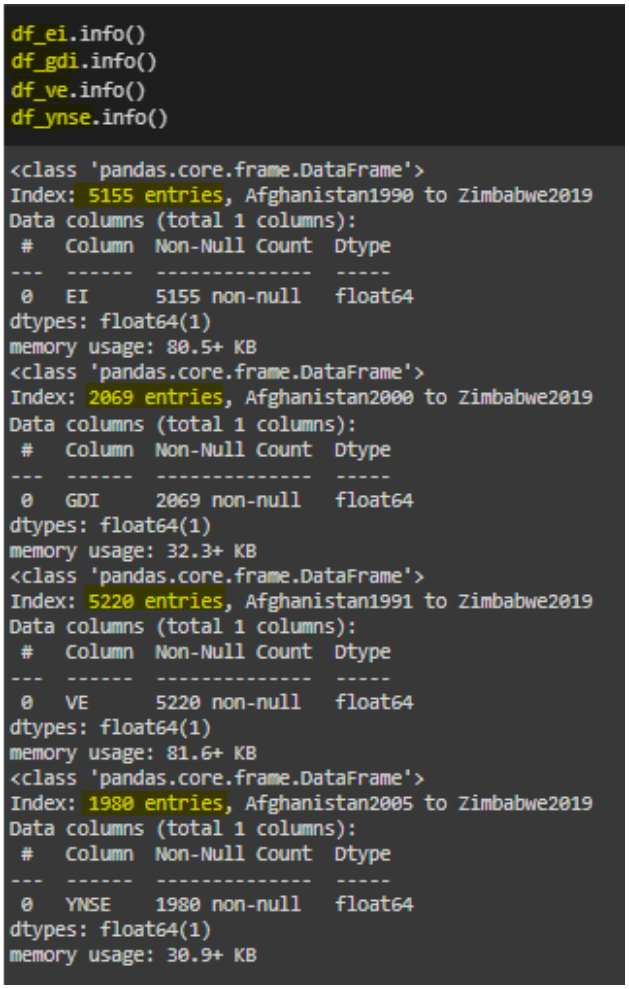
a) Criação do *dataframe Education Index (EI)* - df_ei

```
[31] url_ei ='https://github.com/deboranson/tcc/blob/main/Education.xlsx?raw=true'
     df_ei = pd.read_excel(url_ei)
     df_ei.head()

[32] columns_ei = df_ei.columns[1:]

[33] def transform_ei(df,columns):
         country = []
         year = []
         value = []
         for linha in df.iloc():
           for column in columns:
             year.append(column)
             country.append(linha['Country'])
             value.append(linha[column])
         return pd.DataFrame({'Country':country,'Year':year,'EI':value})

[34] df_ei=transform_ei(df_ei,columns_ei)
     df_ei.head()

[35] df_ei.info()

[36] df_ei['EI'] = pd.to_numeric(df_ei['EI'],errors='coerce')
     df_ei.head()

[37] df_ei.isna().sum()

[38] df_ei = df_ei.dropna()

[39] df_ei['country-year'] = df_ei['Country']+ df_ei['Year'].astype(str)
     df_ei.head()

[40] df_ei['country-year'] = df_ei['country-year'].str.replace(" ","")

[41] df_ei = df_ei.set_index('country-year')
     df_ei.head()

[42] df_ei = df_ei.drop(columns = ['Country','Year'])
     df_ei.head()
```

Fonte: elaboração própria conforme entradas no *notebook* associado.

Como resultado, foram obtidos *dataframes* que possuem as informações abaixo:

Figura 7 – Informações de *df_ei*, *df_gdi*, *df_ve* e *df_ynse*



```
df_ei.info()
df_gdi.info()
df_ve.info()
df_ynse.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 5155 entries, Afghanistan1990 to Zimbabwe2019
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   EI      5155 non-null   float64
dtypes: float64(1)
memory usage: 80.5+ KB
<class 'pandas.core.frame.DataFrame'>
Index: 2069 entries, Afghanistan2000 to Zimbabwe2019
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   GDI     2069 non-null   float64
dtypes: float64(1)
memory usage: 32.3+ KB
<class 'pandas.core.frame.DataFrame'>
Index: 5220 entries, Afghanistan1991 to Zimbabwe2019
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   VE      5220 non-null   float64
dtypes: float64(1)
memory usage: 81.6+ KB
<class 'pandas.core.frame.DataFrame'>
Index: 1980 entries, Afghanistan2005 to Zimbabwe2019
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   YNSE    1980 non-null   float64
dtypes: float64(1)
memory usage: 30.9+ KB
```

Fonte: elaboração própria com base nas saídas do *notebook* associado.

Concluída essa etapa, no subitem 3.4.3 do *notebook*, é, por fim, criado o *df4* com a junção dos *dataframes df1_ad*, *df_ei*, *df_gdi*, *df_ve* e *df_ynse*. As informações relativas ao *df4* seguem abaixo:

Figura 8 – Informações do *df*4

```
df4.info()

<class 'pandas.core.frame.DataFrame'>
Index: 1217 entries, Afghanistan2010 to Zimbabwe2019
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Life Ladder  1217 non-null   float64
 1   EI           1217 non-null   float64
 2   GDI          1217 non-null   float64
 3   VE           1217 non-null   float64
 4   YNSE         1217 non-null   float64
dtypes: float64(5)
memory usage: 57.0+ KB
```

Fonte: elaboração própria com base na saída do *notebook* associado.

e) *Dataframe* 5 *(df5)*

Para a criação do *df*5, no subitem 3.5 do *notebook*, são integrados os *dataframes df1*, *df_ei*, *df_gdi*, *df_ve* e *df_ynse*, englobando-se nele, então, todas as oito variáveis introduzidas no próprio *WHR* (constantes do *df1*) e os quatro indicadores adicionais apresentados neste trabalho (constantes dos demais *dataframes* mencionados).

Como resultado, obtém-se um *dataframe* com as seguintes informações:

Figura 9 – Informações do *df*5

```
df5.info()

<class 'pandas.core.frame.DataFrame'>
Index: 1217 entries, Afghanistan2010 to Zimbabwe2019
Data columns (total 13 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   Life Ladder                       1217 non-null   float64
 1   Log GDP per capita                1217 non-null   float64
 2   Social support                    1217 non-null   float64
 3   Healthy life expectancy at birth  1217 non-null   float64
 4   Freedom to make life choices      1217 non-null   float64
 5   Generosity                        1217 non-null   float64
 6   Perceptions of corruption         1217 non-null   float64
 7   Positive affect                   1217 non-null   float64
 8   Negative affect                   1217 non-null   float64
 9   EI                                1217 non-null   float64
 10  GDI                               1217 non-null   float64
 11  VE                                1217 non-null   float64
 12  YNSE                              1217 non-null   float64
dtypes: float64(13)
memory usage: 133.1+ KB
```

Fonte: elaboração própria com base na saída do *notebook* associado.

f) *Dataframe* 6 *(df*6*)*

O sexto *dataframe* criado para desenvolvimento dos modelos de aprendizado de máquina neste trabalho, consiste, simplesmente, no *df*5 sem os dados relativos ao afeto negativo (*negative affect*). Conforme já informado acima, essa transformação é realizada a fim de se verificar a afirmação constante do *WHR* de que o afeto negativo não teria um impacto significativo na definição do nível de felicidade traduzido pelo *Life Ladder*.

Diante do exposto, no subitem 3.6 do *notebook,* foi executada apenas a eliminação da coluna "*Negative affect*" e criado o *df*6:

Figura 10 – Informações do *df*6

```
df6.info()

<class 'pandas.core.frame.DataFrame'>
Index: 1217 entries, Afghanistan2010 to Zimbabwe2019
Data columns (total 12 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Life Ladder                     1217 non-null   float64
 1   Log GDP per capita              1217 non-null   float64
 2   Social support                  1217 non-null   float64
 3   Healthy life expectancy at birth 1217 non-null  float64
 4   Freedom to make life choices    1217 non-null   float64
 5   Generosity                      1217 non-null   float64
 6   Perceptions of corruption       1217 non-null   float64
 7   Positive affect                 1217 non-null   float64
 8   EI                              1217 non-null   float64
 9   GDI                             1217 non-null   float64
 10  VE                              1217 non-null   float64
 11  YNSE                            1217 non-null   float64
dtypes: float64(12)
memory usage: 155.9+ KB
```

Fonte: elaboração própria com base na saída do *notebook* associado.

g) *Dataframe* 7 *(df*7*)*

Por fim, foi criado o *df*7, o qual surge, também, da exclusão de mais uma das colunas do *df*5: o afeto positivo (*positive affect*). Dentro do contexto discriminado anteriormente, essa modificação decorre do interesse de se averiguar a afirmação constante do *WHR* de que o afeto negativo não teria uma influência expressiva na definição do *Life Ladder*.

Dessa forma, no subitem 3.7 do *notebook*, foi eliminada a coluna "*Positive affect*" do *df6* e criado o *df7*:

Figura 11 – Informações do *df7*

```
df7.info()

<class 'pandas.core.frame.DataFrame'>
Index: 1217 entries, Afghanistan2010 to Zimbabwe2019
Data columns (total 11 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   Life Ladder                       1217 non-null   float64
 1   Log GDP per capita                1217 non-null   float64
 2   Social support                    1217 non-null   float64
 3   Healthy life expectancy at birth  1217 non-null   float64
 4   Freedom to make life choices      1217 non-null   float64
 5   Generosity                        1217 non-null   float64
 6   Perceptions of corruption         1217 non-null   float64
 7   EI                                1217 non-null   float64
 8   GDI                               1217 non-null   float64
 9   VE                                1217 non-null   float64
 10  YNSE                              1217 non-null   float64
dtypes: float64(11)
memory usage: 146.4+ KB
```

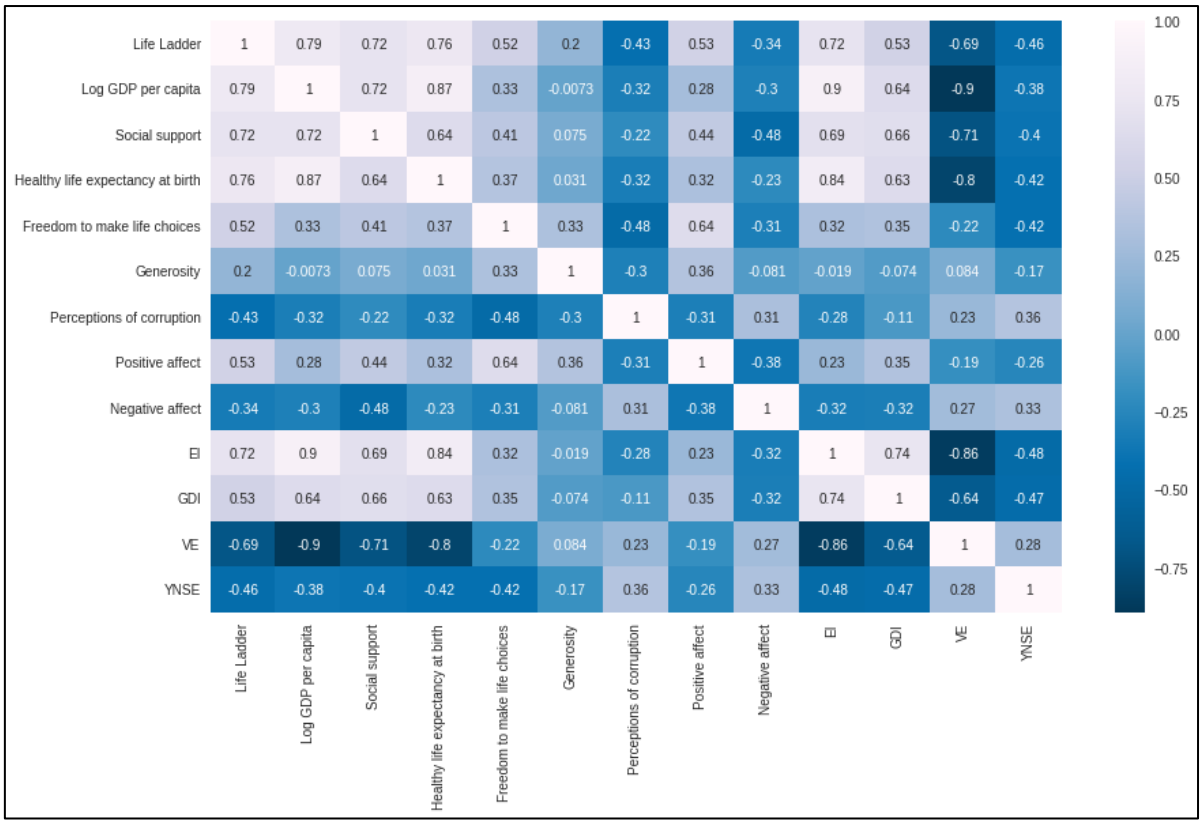
Fonte: elaboração própria com base na saída do *notebook* associado.

4 Análise e exploração dos dados

A fim de ilustrar a estrutura de dados, no item 4 do *notebook*, são executados gráficos com as funções do pacote *Seaborn* e do próprio *Pandas* que tiveram por base o *dataframe* 5 – que engloba todas as variáveis independentes consideradas neste trabalho.

Primeiramente, segue discriminada a matriz de correlação entre as variáveis apresentadas por meio de um *heatmap* obtido por meio da função `sns.heatmap`:

Figura 12 – *Heatmap* da matriz de correlação das variáveis (*df5*)

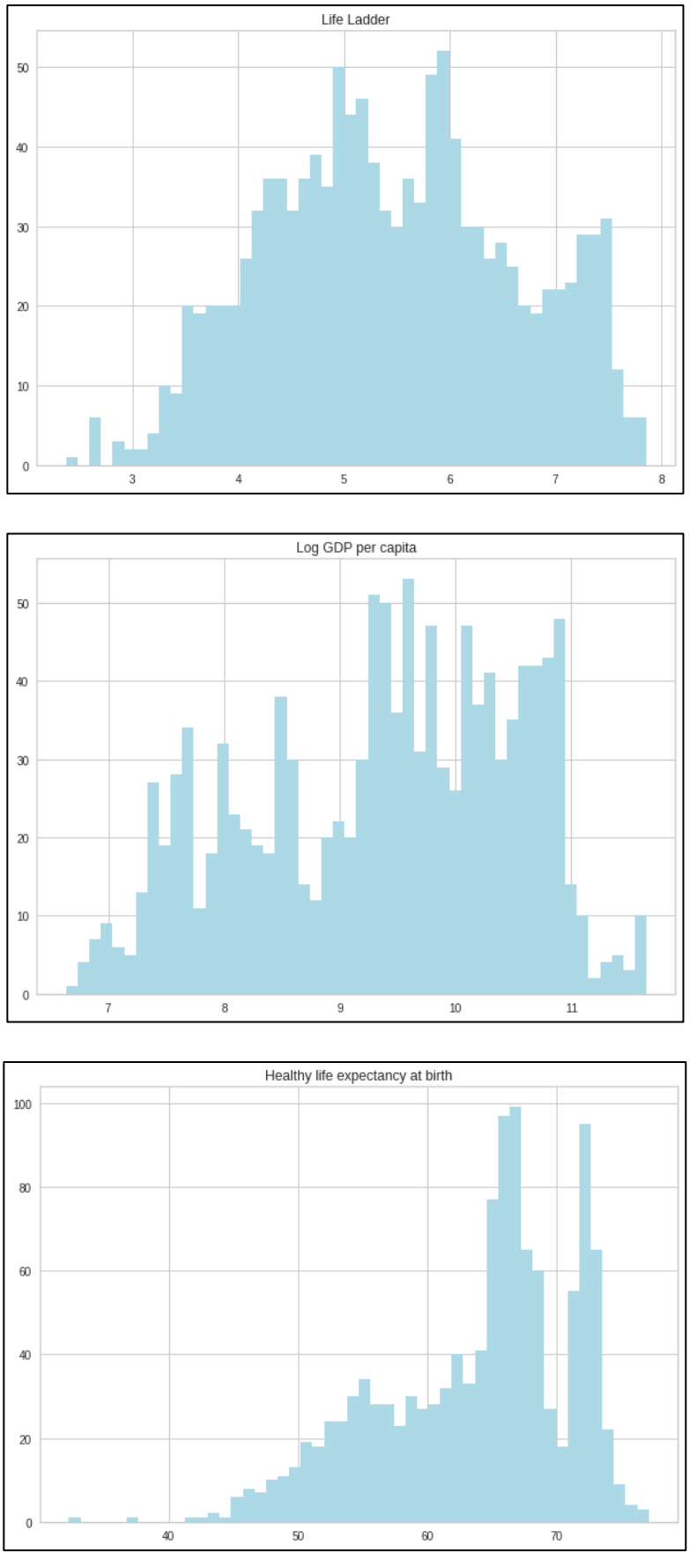


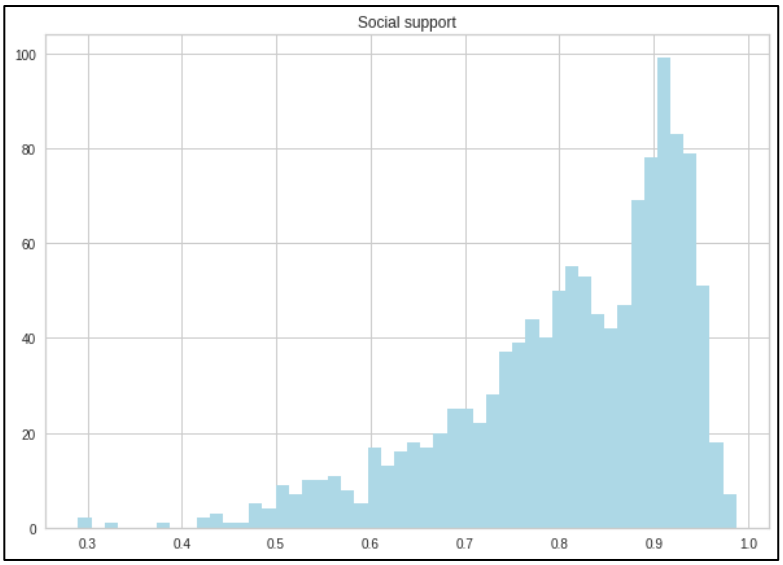
Fonte: elaboração própria conforme saída do *notebook* associado.

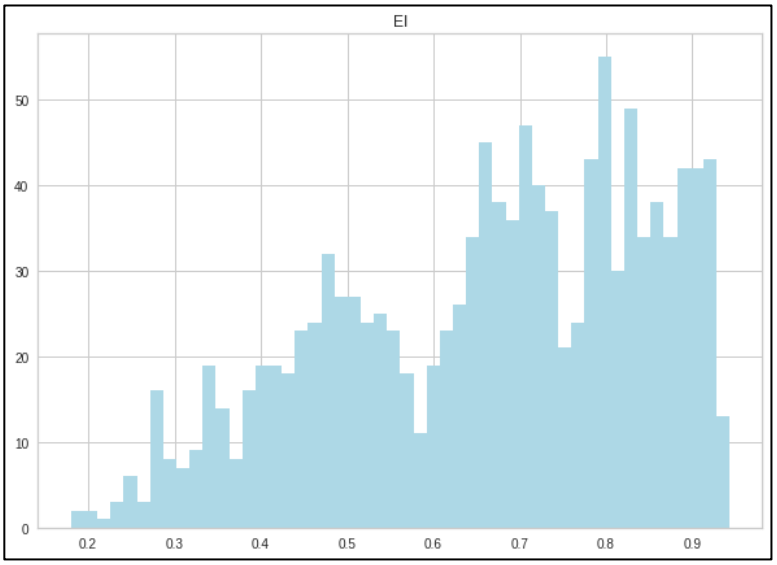
Conforme se verifica no *heatmap*, é expressiva a correlação entre as variáveis explicativas e o *Life Ladder*, sendo que um terço delas apresenta um coeficiente maior que 0,70 – PIB *per capita* (log), expectativa de vida saudável ao nascer, suporte social e *EI*.

Adicionalmente, foram executados os histogramas para o *Life Ladder* e para as quatro variáveis mencionadas acima (que possuem os maiores coeficientes de correlação associados com esse indicador):

Figura 13 – Histogramas das principais variáveis (conforme resultados do *heatmap*)





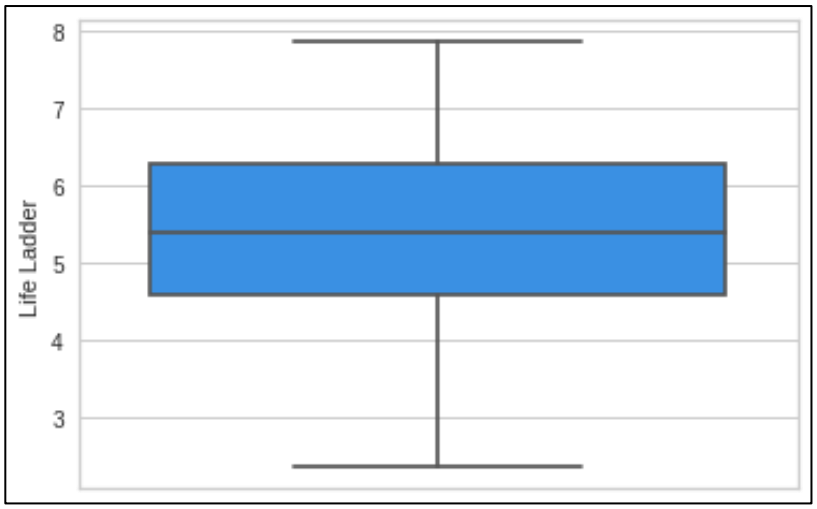


Fonte: elaboração própria com base nas saídas do *notebook* associado.

Conforme se verifica dos gráficos acima, à exceção das distribuições do *Life Ladder* e do PIB *per capita*, observa-se uma assimetria à direita nas demais, o que se mostra positivo dado que se referem a variáveis indicativas de desenvolvimento e bem-estar populacional.

Por fim, foi executado o gráfico *boxplot* (função `sns.boxplot`) a fim de se verificar a existência de *outliers* para a variável dependente *Life Ladder:*

Figura 14 – *Boxplot* da variável *Life Ladder*



Fonte: elaboração própria conforme saída do *notebook* associado.

Da análise desse gráfico, confirmando o antecipado pelo respectivo histograma, verifica-se que a variável possui uma distribuição simétrica, não se observando a presença de *outliers.*

Finalizada a criação dos *dataframes* e a análise preliminar dos dados, na próxima seção, é discriminada a forma de criação dos modelos de aprendizado de máquina utilizando-se essa base.

5 Criação dos modelos de aprendizado de máquina

5.1 *Pycaret*

Para a criação dos modelos deste trabalho, foi utilizada a biblioteca *Pycaret*, módulo *Regression*[25]. Essa é uma biblioteca de código aberto *Python* criada com o objetivo de facilitar a execução de tarefas padrão em um projeto de aprendizagem de máquina. Sua origem é o pacote de aprendizado de máquina *Caret* da linguagem R, o qual permite que os modelos

[25] Dado que o objetivo, aqui, é definir um modelo supervisionado que preveja o nível de felicidade dos diversos países (o indicador *Life Ladder*).

sejam avaliados, comparados e ajustados para um determinado conjunto de dados com o uso de poucas linhas de código.

Nesse contexto, a biblioteca *PyCaret* trouxe esses recursos para a linguagem *Python*, tornando possível a execução de um conjunto de algoritmos de aprendizagem de máquina padrão para uma base de dados de uma forma bastante simplificada.

Preliminarmente, cabe destacar que, por padrão, o *PyCaret* utiliza 70% do conjunto de dados para treinamento, sendo os 30% restantes destinados para testes. Nele, a avaliação de um modelo de aprendizado de máquina treinado e a otimização dos hiperparâmetros são realizadas por meio da *K-fold Cross Validation* somente sobre a base de treino.

A técnica de *K-fold Cross Validation*[26] divide a base de dados de forma aleatória em K subconjuntos com quantidade de observações similares. A cada iteração, um conjunto formado por (K-1) subconjuntos são utilizados para treinamento e o subconjunto restante, para teste, trazendo como resultado métricas para avaliação. Dessa forma, todos os subconjuntos (da base de treino) são usados para teste em algum momento da avaliação do modelo.

Na próxima subseção, são descritas as funções do *Pycaret* e os respectivos parâmetros utilizados para a criação dos modelos neste trabalho.

5.2 Etapas para a criação dos modelos

No item 5 do *notebook*, para o desenvolvimento dos modelos de aprendizado de máquina a partir dos *dataframes* criados (conforme discriminação constante da subseção 3.2 acima), são utilizadas as funções do *Pycaret* listadas abaixo:

a) *Setup*

Como primeira etapa para a utilização do módulo *regression* do *Pycaret*, é executada a função `setup` no *notebook* (a qual realiza as configurações iniciais e as inferências sobre os dados). Para tanto, são definidos os seguintes parâmetros nela:

- *data* (base de dados)*:* informado o *dataframe* de trabalho (do *df1* até o *df7*);
- *target* (variável-alvo): indicada a variável *Life Ladder*;

[26] Conforme discriminado na página https://scikit-learn.org/stable/modules/cross_validation.html, acessada em 18/09/2021.

- *normalize*: alterado para `normalize = True` de forma a promover a normalização dos dados;
- *silent*: modificado para `silent = True` a fim de que não seja solicitada confirmação sobre as inferências realizadas pela função.

Quanto à divisão das bases treino/teste e à definição do número de subconjuntos para a *K-fold Cross Validation*, são mantidos os padrões definidos (70%/30% e 10 *folds* respectivamente).

Assim, o comando dessa função para, por exemplo, o *df1* no *notebook* fica sintetizado conforme estrato abaixo:

Figura 15 – Comando para a função *setup*

```
df1_models = setup (data = df1, target ='Life Ladder', session_id=123, normalize = True, silent = True)
```

Fonte: elaboração própria conforme entrada no *notebook* associado.

b) *Compare_models*

Após a conclusão do `setup`, é comandada a função `compare_models` que treina todos os modelos da respectiva biblioteca se utilizando de hiperparâmetros padrão e analisando as métricas de desempenho por meio, também, da *cross-validation*.

Dado que, em uma regressão, o erro é calculado comparando-se os valores previstos com os valores reais, essa função retorna as seguintes medidas para avaliação dos modelos[27]:

- *MAE* (*Mean Absolute Error* – Erro Médio Absoluto): representa a média dos valores absolutos dos erros, ou seja, mostra a média da diferença absoluta entre os valores reais e os previstos no conjunto de dados. Em forma de equação:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} | y_i - \hat{y}_i |$$

[27] Classificando-os a partir de seus coeficientes de determinação (R^2), cuja conceituação consta na sequência deste trabalho.

- *MSE* (*Mean Squared Error* – Erro Quadrático Médio): representa a média da diferença quadrática entre os valores originais e os previstos no conjunto de dados, logo, ele mede a variância dos resíduos. Como equação:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

- *RMSE* (*Root Mean Squared Error* – Raiz do Erro Médio Quadrático): como o próprio nome diz, ele representa a raiz do *MSE*, medindo, assim, o desvio padrão dos resíduos. Sua equação segue abaixo:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

- R^2 (R quadrado ou coeficiente de determinação): apresenta quanto da variação na variável dependente é explicada pelo modelo de regressão (por meio das variáveis independentes). Assim, ele pode ser definido como uma medida que representa o grau de ajuste de um modelo aos valores observados de uma variável aleatória, estando seu valor sempre situado entre 0 e 1.

  Para se chegar a essa medida, calcula-se a razão entre a soma de quadrados da regressão (SQR, que é a diferença entre a soma de quadrados total – SQT – e a soma de quadrados do erro – SQE) e a soma de quadrados total (SQT) conforme segue:

$$R^2 = \frac{SQR}{SQT} = \frac{\sum_i (\hat{y}_i - \bar{y})^2}{\sum_i (y_i - \bar{y})^2}$$

  Neste trabalho, o R^2 será a medida utilizada para comparação da qualidade dos modelos de aprendizado de máquina criados.

- *RMSLE*: (*Root Mean Squared Logarithmic Error* – Erro logarítmico médio quadrático): representa uma versão do *RMSE* na qual são considerados os logaritmos dos valores reais e dos previstos antes de se efetivar a elevação ao quadrado:

$$RMSLE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\log(y_i + 1) - \log(\hat{y}_i + 1))^2}$$

- *MAPE* (*Mean Absolute Percentage Error* - Erro Percentual Médio Absoluto): apresenta a medida de erro percentual absoluto no modelo, para tanto, divide o erro absoluto do modelo pelos valores reais dos dados:

$$MAPE = \frac{100\%}{N}\sum_{i=0}^{N}\frac{|\, y_i - \hat{y}_i|}{y_i}$$

Abaixo, para exemplificação, segue estrato do comando dessa função e parte de sua saída para o *df1*:

Figura 16 – Comando e parte da saída da função *compare_models* (*df1*)

```
compare_models()
```

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
et	Extra Trees Regressor	0.2911	0.1474	0.3830	0.8866	0.0662	0.0593	0.702
rf	Random Forest Regressor	0.3166	0.1738	0.4154	0.8663	0.0709	0.0641	0.965
lightgbm	Light Gradient Boosting Machine	0.3228	0.1801	0.4229	0.8614	0.0723	0.0653	0.148

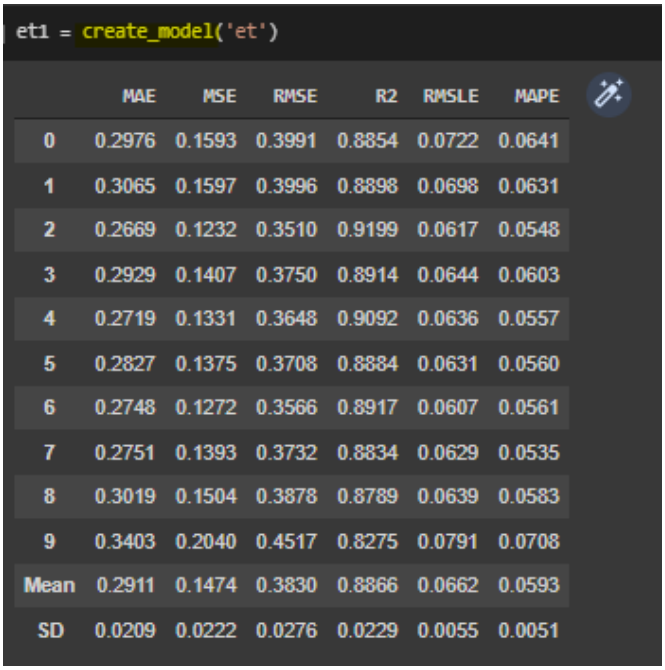
Fonte: elaboração própria com base na saída do *notebook* associado.

c) *Create_model*

Após a saída com a classificação dos modelos/algoritmos, é executada a função `create_model` para a criação dos três melhores modelos a partir dos algoritmos listados para cada *dataframe* (os que obtiveram os melhores R^2).

Abaixo, segue imagem da saída dessa função para o *df1* no *notebook* relativamente ao melhor modelo indicado pela função `compare_models` para esse *dataframe* (o *Extra Trees Regressor*):

Figura 17 – Comando e saída da função *create_model* (*df1*)



```
et1 = create_model('et')
```

	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	0.2976	0.1593	0.3991	0.8854	0.0722	0.0641
1	0.3065	0.1597	0.3996	0.8898	0.0698	0.0631
2	0.2669	0.1232	0.3510	0.9199	0.0617	0.0548
3	0.2929	0.1407	0.3750	0.8914	0.0644	0.0603
4	0.2719	0.1331	0.3648	0.9092	0.0636	0.0557
5	0.2827	0.1375	0.3708	0.8884	0.0631	0.0560
6	0.2748	0.1272	0.3566	0.8917	0.0607	0.0561
7	0.2751	0.1393	0.3732	0.8834	0.0629	0.0535
8	0.3019	0.1504	0.3878	0.8789	0.0639	0.0583
9	0.3403	0.2040	0.4517	0.8275	0.0791	0.0708
Mean	0.2911	0.1474	0.3830	0.8866	0.0662	0.0593
SD	0.0209	0.0222	0.0276	0.0229	0.0055	0.0051

Fonte: elaboração própria com base na saída do *notebook* associado.

Na próxima subseção, são discriminados os melhores modelos/algoritmos identificados e as respectivas combinações para cada um dos *dataframes* criados.

5.3 Melhores modelos e combinações realizadas

Com a execução da função `compare_model` acima discriminada (nos subitens 5.1 a 5.7 do *notebook*), foram identificados os seguintes três melhores modelos/algoritmos para os *dataframes* criados:

Tabela 4 – Melhores modelos/algoritmos

Dataframes	**Melhores modelos/algoritmos**
df1 a *df3*	*Extra Trees Regressor*
	Random Forest Regressor
	Light Gradient Boosting Machine
df4	*Extra Trees Regressor*
	Random Forest Regressor
	K Neighbors Regressor
df5 a *df7*	*Extra Trees Regressor*
	Light Gradient Boosting Machine
	Random Forest Regressor

Fonte: elaboração própria com base nas saídas do *notebook* associado.

Abaixo, seguem suas breves definições:

a) *Extra Trees Regressor* (*ET*)

O *Extra Trees Regressor* faz parte do grupo de árvores de regressão (*Regression Tree Models)*. Usualmente, as árvores de regressão são estimadas por um algoritmo que particiona recursivamente os dados de treinamento fornecidos em subconjuntos menores, visando a uma melhor previsão local de uma resposta contínua.

O algoritmo do *Extra Trees Regressor*, especificamente, funciona criando um significativo número de árvores de decisão a partir do conjunto de dados de treinamento e seus resultados advêm da média das previsões dessas árvores[28].

b) *Random Forest Regressor* (*RF*)

É, também, um algoritmo pertencente ao grupo de árvores de regressão. Assim como o *Extra Trees*, é composto por diversas árvores de decisão, sendo a decisão final obtida de acordo com a previsão delas.

As principais diferenças[29] entre eles se referem ao uso de subamostras com substituição no *Random Forest* (frente à utilização da integralidade da amostra no *Extra Trees*) e à forma de definição dos pontos de corte para a divisão dos nós (sendo selecionado um ponto de divisão ideal no *Random Forest* e um aleatório no *Extra Trees*).

c) *Light Gradient Boosting Machine* (*LightGBM*)

O *LightGBM*[30] é uma estrutura de aumento de gradiente que se utiliza de algoritmos de aprendizado baseados em árvore também. Seu diferencial se dá pela maior velocidade e menor uso de memória no processamento, apresentando considerável eficiência e precisão.

d) *K Neighbors Regressor* (*KNN*)

Com o *KNN*, são avaliados todos os elementos disponíveis para que seja efetuada a previsão de acordo com a similaridade entre as suas características e as da unidade de análise. Para tanto, é calculada a distância entre eles, a qual é baseada no conjunto de características associadas a cada um.

É um algoritmo de implementação simples, efetivo com *datasets* grandes, mas que gera uma carga computacional bastante alta dada a necessidade de cálculo das distâncias entre todos os elementos[31].

---

[28] Conforme explicação constante do *site* https://machinelearningmastery.com/extra-trees-ensemble-with-python, acesso em 18/11/2021.

[29] De acordo com o discriminado na página https://quantdare.com/what-is-the-difference-between-extra-trees-and-random-forest/, acesso em 18/11/2021.

[30] Conforme consta do *site* https://lightgbm.readthedocs.io/en/latest/, acesso em 28/02/2022.

[31] De acordo com o exposto em https://medium.com/analytics-vidhya/k-neighbors-regression-analysis-in-python-61532d56d8e4, acesso em 28/02/2022.

A partir dos modelos criados por esses algoritmos, são efetuadas combinações[32] para seu aprimoramento por meio das funções `blend_models` e `stack_models`.

A função `blend_models` pode ser utilizada para combinar todos os modelos treinados ou modelos específicos definidos por intermédio do parâmetro `estimator_list` inserido diretamente nela.

A função `stack_models` objetiva construir um metamodelo para a previsão final a partir dos diversos estimadores da base e, tal como na `blend_models`, permite a seleção dos modelos a serem utilizados (por meio do `estimator_list`) ou executa a combinação com todos os modelos treinados.

Abaixo, para ilustração, seguem os comandos e as saídas dessas funções executadas para o *df1*:

Figura 18 – Comandos e saídas das funções *blend_models* e *stack_models* (*df1*)

```
blend3_1 = blend_models(estimator_list = [et1,rf1,lightgbm1])
```

	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	0.2982	0.1639	0.4048	0.8821	0.0724	0.0641
1	0.3210	0.1715	0.4141	0.8817	0.0715	0.0656
2	0.2762	0.1293	0.3596	0.9160	0.0621	0.0565
3	0.3125	0.1666	0.4082	0.8713	0.0711	0.0652
4	0.2874	0.1470	0.3834	0.8997	0.0664	0.0584
5	0.2949	0.1452	0.3811	0.8821	0.0647	0.0584
6	0.2823	0.1242	0.3524	0.8942	0.0601	0.0572
7	0.2955	0.1598	0.3998	0.8662	0.0671	0.0576
8	0.3020	0.1546	0.3932	0.8755	0.0644	0.0579
9	0.3614	0.2240	0.4733	0.8106	0.0823	0.0750
Mean	0.3031	0.1586	0.3970	0.8779	0.0682	0.0616
SD	0.0232	0.0263	0.0319	0.0264	0.0061	0.0056

[32] Dos 2 e dos 3 melhores modelos obtidos para cada *dataframe*.

```
stacked_models3_1 = stack_models(estimator_list=[et1,rf1,lightgbm1])
```

	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	0.2967	0.1550	0.3937	0.8885	0.0714	0.0640
1	0.3024	0.1542	0.3927	0.8936	0.0689	0.0624
2	0.2665	0.1217	0.3489	0.9209	0.0618	0.0549
3	0.2927	0.1412	0.3757	0.8910	0.0644	0.0602
4	0.2707	0.1334	0.3652	0.9090	0.0629	0.0547
5	0.2777	0.1400	0.3742	0.8863	0.0638	0.0550
6	0.2719	0.1330	0.3647	0.8867	0.0621	0.0555
7	0.2729	0.1375	0.3708	0.8849	0.0625	0.0530
8	0.3006	0.1509	0.3884	0.8785	0.0641	0.0581
9	0.3395	0.2000	0.4472	0.8309	0.0786	0.0708
Mean	0.2892	0.1467	0.3822	0.8870	0.0661	0.0588
SD	0.0211	0.0203	0.0254	0.0222	0.0051	0.0053

Fonte: elaboração própria com base nas saídas do *notebook* associado.

Na próxima seção, são apresentados os resultados obtidos, detalhado o melhor modelo identificado dentre todos os *dataframes* criados e discriminada a aplicação deste na base de teste.

6 Apresentação dos resultados

6.1 Resultados dos melhores modelos

Com a execução individual dos melhores algoritmos/modelos encontrados para cada *dataframe*[33] (por intermédio da função `create_model`) e, após, das suas combinações (por meio das funções `blend_models` e `stack_models`), são obtidos os seguintes resultados:

[33] Nos subitens 5.1 a 5.7 do *notebook*, respectivamente, para os *dataframes* de 1 a 7.

Tabela 5 – Resultados dos melhores modelos – base de treinamento

(continua)

Dataframe	**Variáveis do *dataframe***	**Melhores modelos**	**R^2 do modelo**	**Melhor combinação**	**R^2 da combinação**
df1	*Life Ladder* PIB *per capita* (log) Expectativa de vida Suporte social Liberdade para escolher Generosidade Percepções de corrupção Afeto positivo Afeto negativo	*ET*	0,8866	*Stack* dos 2 melhores modelos	0,8871
		RF	0,8663		
		LightGBM	0,8614		
df2	*Life Ladder* PIB *per capita* (log) Expectativa de vida Suporte social Liberdade para escolher Generosidade Percepções de corrupção Afeto positivo	*ET*	0,8842	*Stack* dos 2 melhores modelos	0,8850
		RF	0,8632		
		LightGBM	0,8531		
df3	*Life Ladder* PIB *per capita* (log) Expectativa de vida Suporte social Liberdade para escolher Generosidade Percepções de corrupção	*ET*	0,8695	*Stack* dos 2 melhores modelos	0,8708
		RF	0,8430		
		LightGBM	0,8402		
df4	*Life Ladder* *EI* *GDI* *VE* *YNSE*	*ET*	0,8631	*Stack* dos 3 melhores modelos	0,8630
		RF	0,8468		
		KNN	0,8294		
df5	*Life Ladder* PIB *per capita* (log) Expectativa de vida Suporte social Liberdade para escolher Generosidade Percepções de corrupção Afeto positivo Afeto negativo *EI* *GDI* *VE* *YNSE*	*ET*	0,8951	*Stack* dos 2 melhores modelos	0,8964
		LightGBM	0,8767		
		RF	0,8762		

Tabela 5 – Resultados dos melhores modelos – base de treinamento

(conclusão)

Dataframe	**Variáveis do *dataframe***	**Melhores modelos**	**R^2 do modelo**	**Melhor combinação**	**R^2 da combinação**
df6	*Life Ladder* PIB *per capita* (log) Expectativa de vida Suporte social Liberdade para escolher Generosidade Percepções de corrupção Afeto positivo *EI* *GDI* *VE* *YNSE*	*ET*	0,8935	*Stack* dos 2 e dos 3 melhores modelos	0,8943
		LightGBM	0,8769		
		RF	0,8743		
df7	*Life Ladder* PIB *per capita* (log) Expectativa de vida Suporte social Liberdade para escolher Generosidade Percepções de corrupção *EI* *GDI* *VE* *YNSE*	*ET*	0,8896	*Stack* dos 3 melhores modelos	0,8910
		LightGBM	0,8694		
		RF	0,8650		

Fonte: elaboração própria com base nas saídas do *notebook* associado.

Conforme se verifica da tabela acima, o modelo que possui maior poder de previsão do indicador *Life Ladder* (ou seja, que tem o maior coeficiente de determinação – R^2) é o obtido a partir da combinação do *ET* e do *LightGBM* – realizada por meio da função `stack_models`[34] – a partir das variáveis independentes constantes do *df*5. Dado que o respectivo R^2 é de 0,8964, entende-se que as doze variáveis independentes presentes nesse modelo são capazes de explicar 89,64% da variação do *Life Ladder*.

Além desse resultado, importa destacar a combinação *stack* dos dois (ou dos três) melhores modelos obtidos a partir do *df6* cujo R^2 (de 0,8943) é somente 0,24% menor que o alcançado tendo como base o *df*5.

Nesse contexto, no que se refere ao papel do afeto negativo (e a conclusão reiterada do *WHR* pela sua irrelevância nas avaliações de vida dos indivíduos no decorrer dos anos), se verifica que ele, de fato, não possui impacto significativo na determinação do *Life Ladder*. A

[34] Executada no subitem 5.5.3 do *notebook*.

variação do R^2 entre os modelos apresentados acima para o *df5* (que o inclui como variável independente) e para *df6* (que o exclui) é ínfima: a melhora do R^2 entre os referidos *dataframes* atinge, no máximo, 0,22% (com o modelo *RF*, passando de 0,8743 no *df6* para 0,8762 no *df5*). Por outro lado, a retirada do afeto positivo da base de variáveis independentes (passando-se, assim, do *df6* para o *df7*), leva a uma variação no R^2 de até 1,08% (com o modelo *RF* também, indo de 0,8650 no *df7* para 0,8743 no *df6*).

Diante do exposto, entende-se que os resultados obtidos pelos modelos desenvolvidos neste trabalho ratificam a conclusão apresentada nos *WHR* de que o afeto negativo não possui influência para as avaliações de vida dos indivíduos (expressas pelo *Life Ladder*).

Por fim, ressalta-se que o processamento dessas funções e todos os resultados associados constam detalhados nos subitens 5.1 a 5.7 do *notebook* (respectivamente, para os *dataframes* de 1 a 7).

6.2 Análise dos resultados do melhor modelo

De forma a facilitar a compreensão do *notebook*, a análise completa do melhor modelo obtido para previsão do *Life Ladder* (o resultante da combinação `stack` do *ET* e do *LightGBM* a partir dos dados constantes do *df5*) é apresentada no seu item 6. Abaixo, para ilustração, segue tabela com todas as métricas obtidas[35] para esses dois modelos individuais executados sobre a base de treinamento do *dataframe* (70% das observações) e para a respectiva combinação `stack`:

Tabela 6 – Métricas dos melhores modelos para o *df5* – base de treinamento

Modelo/ combinação	Métricas					
	MAE	*MSE*	*RMSE*	*RMSLE*	*MAPE*	R^2
Extra Trees Regressor	0,2680	0,1363	0,3668	0,0648	0,0556	0,8951
Light Gradient Boosting Machine	0,2920	0,1595	0,3971	0,0694	0,0598	0,8767
Stack	0,2634	0,1347	0,3643	0,0644	0,0544	0,8964

Fonte: elaboração própria com base nas saídas do *notebook* associado.

[35] Discriminadas nos subitens 5.5.1 e 5.5.3 do *notebook*, bem como nos subitens 6.1 e 6.2.

Todas as métricas expostas até este ponto referem-se aos resultados obtidos somente a partir dos dados da base de treinamento (ou seja, sobre 70% das observações). Assim, de forma a se verificar as métricas para a base de testes (os 30% restantes de observações), no subitem 6.2 do *notebook*, foi executado o modelo final treinado – `stacked_models2_5` – por meio da função `predict_model`, alcançando-se, assim, os seguintes resultados:

Tabela 7 – Métricas do melhor modelo para o *df5* – base de testes

Métricas					
MAE	*MSE*	*RMSE*	*RMSLE*	*MAPE*	R^2
0,2361	0,0993	0,3151	0,0552	0,0486	0,9260

Fonte: elaboração própria com base nas saídas do *notebook* associado.

Conforme se verifica da tabela 7 acima, o R^2 obtido com a base de testes é de 0,9260, logo, 92,60% da variação do *Life Ladder* poderia ser explicada pelas variáveis independentes consideradas no modelo. Dado que esse percentual se mostra próximo ao alcançado com a base de treinamento (89,64%), pode-se concluir que não ocorreu um ajuste excessivo (*overfitting*) dos dados no conjunto de teste.

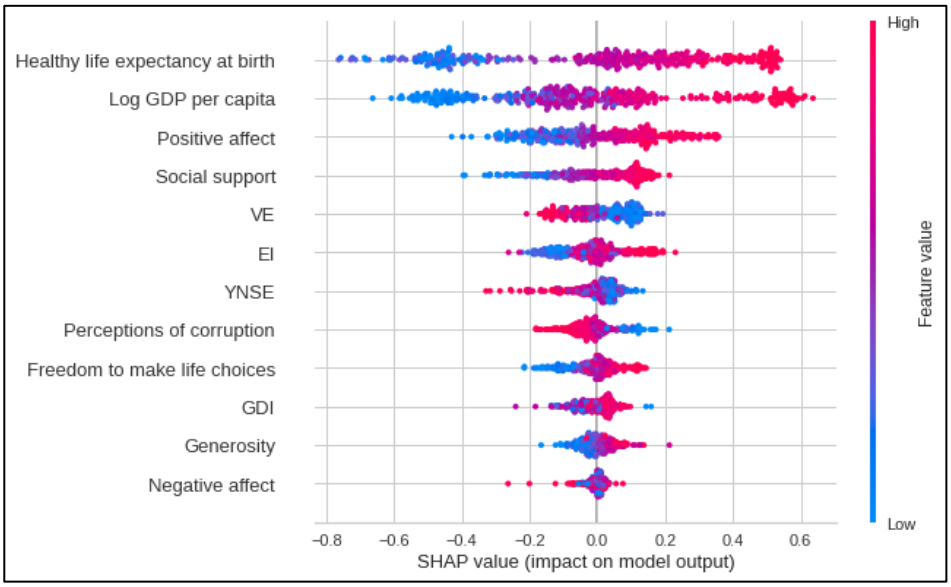
Cabe ressaltar que, na tentativa de se aprimorar o resultado obtido com o *dataframe 5*, no item 7 do *notebook*, foram testadas, ainda, alterações em alguns parâmetros opcionais da função `setup`, quais sejam: `train_size` (relativo ao tamanho da base de treino, tendo sido modificado para 60 e 80%), `remove_multicollinearity/multicollinearity_threshold` (referente à remoção das variáveis com colinearidade superior ao limite definido, tendo sido utilizados os percentuais de 70, 80 e 90%) e `remove_outliers/outliers_threshold` (concernente à remoção de *outliers*, com os percentuais de 5 e 10% para eliminação). Contudo, após a inserção individual (nos subitens 7.1 a 7.3 do *notebook*) e conjunta (no subitem 7.4) dessas modificações nos parâmetros/limites para execução dos modelos, restou verificado que o modelo obtido a partir da combinação `stack` do *ET* e do *LightGBM* sobre o *df5* com os parâmetros originalmente fixados[36] para a função `setup` ainda detinha o maior R^2 observado para a base de testes (de 0,926, tendo o maior R^2 nas demais execuções atingido 0,9193[37]).

[36] Constantes da alínea "a" da subseção 5.2 deste trabalho.
[37] Com a mesma combinação *stack* do *ET* e do *LightGBM* sobre o *df5*, mas com a base de treino ampliada para 80%.

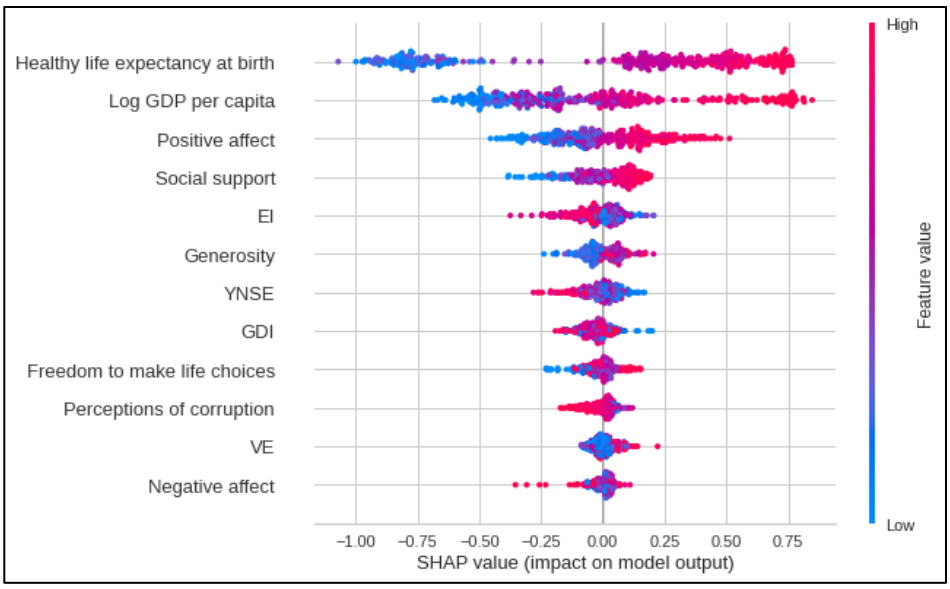
Dessa forma, para os modelos individuais que compuseram a combinação `stacked_models2_5`[38], foi executada a função `interpret_model` no subitem 6.3 do *notebook*, obtendo-se os seguintes gráficos:

Figura 19 – Gráficos de interpretação dos modelos *ET e LightGBM* para o *df5*

Extra Trees Regressor



Light Gradient Boosting Machine



Fonte: elaboração própria com base nas saídas do *notebook* associado.

[38] Ou seja, a combinação `stack` do *ET* e do *LightGBM* sobre o *df5*.

Conforme se verifica dos gráficos acima, as variáveis que possuem maior impacto[39] para a determinação do *Life Ladder* a partir dos modelos treinados são:

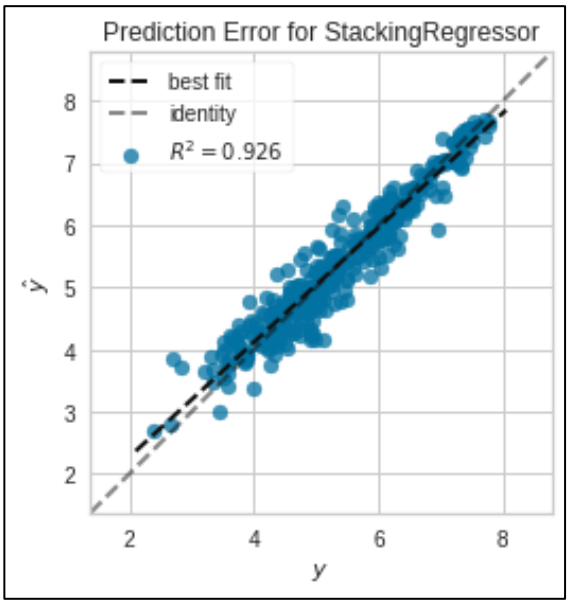
- expectativa de vida saudável ao nascer;
- PIB *per capita* (log)*;*
- afeto positivo; e
- suporte social.

Quanto às demais variáveis, suas importâncias se alternam, sendo que, dentre os quatro indicadores adicionais do PNUD inseridos neste trabalho (quais sejam, *EI*, *GDI*, *VE* e *YNSE*), destaca-se o relativo à educação, classificando-se sua influência em sexta e em quinta posição nos modelos apresentados acima.

Cabe destacar, ainda, que, para ambos os modelos, o afeto negativo possui a menor influência dentre as doze variáveis independentes utilizadas para a explicação do *Life Ladder*, confirmando, novamente as conclusões dos *WHR*.

Por fim, para visualização dos resultados obtidos com o melhor modelo sobre a base de teste, no subitem 6.4 do *notebook***,** foram gerados os gráficos abaixo com a função `plot_model`:

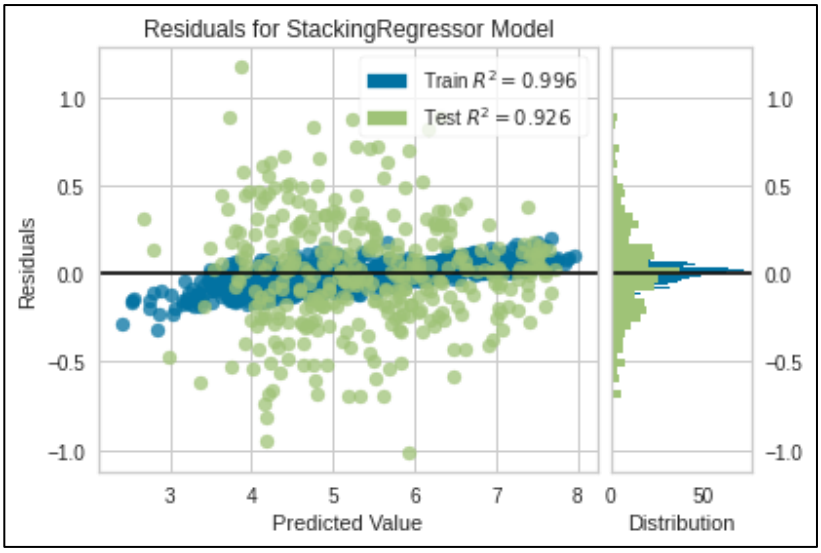
Figura 20 – Erro de predição do melhor modelo



Fonte: elaboração própria conforme saída do *notebook* associado.

[39] Em geral, pode-se afirmar que as variáveis independentes que apresentam uma maior dispersão de observações/pontos no eixo horizontal desse tipo de gráfico são as que possuem maior importância para explicação da variável-alvo.

Figura 21 – Resíduos no melhor modelo



Fonte: elaboração própria conforme saída do *notebook* associado.

7 Conclusão

O objetivo deste trabalho foi criar um modelo de aprendizado de máquina que pudesse prever o indicador *Life Ladder* – o qual busca expressar o nível de felicidade das pessoas ao redor do mundo. Para sua definição, o Instituto *Gallup* calcula a média da resposta de uma amostra de cidadãos de cada país à principal pergunta de avaliação de vida feita na *GWP*, sendo essa média da percepção dos próprios indivíduos sobre suas vidas a base para as análises do *WHR* que é divulgado anualmente.

Nesse contexto, de forma a se obter um modelo para previsão desse indicador de felicidade, foram consideradas as oito variáveis apresentadas pelo próprio *WHR* na tentativa de determinar o *Life Ladder* (PIB *per capita* – log, expectativa de vida saudável ao nascer, suporte social, liberdade para fazer escolhas de vida, generosidade, percepções de corrupção, afeto positivo e afeto negativo), e, adicionalmente, foram inseridos quatro indicadores socioeconômicos (Índice de Educação, Índice de Desenvolvimento de Gênero, Vulnerabilidade do emprego e Jovens que não estudam e nem estão empregados) para a tentativa de modelagem aqui apresentada.

A partir dessas doze variáveis, foram criados sete *dataframes* diferentes, os quais foram submetidos às funções da biblioteca *Pycaret*, tendo sido obtido, assim, com a combinação (`stack_models`) dos dois melhores modelos do *df7* (o *Extra Trees Regressor* e o *Light Gradient Boosting Machine*), um coeficiente de determinação (R^2) de 0,8964 com a base de treinamento e de 0,9260 com a de teste.

Diante de todo o exposto, conclui-se que este trabalho atingiu seu objetivo de criar um modelo que pudesse prever, de forma razoável, o *Life Ladder*. Como sugestão para trabalhos futuros, contudo, entende-se ser interessante a busca e a consideração de outras variáveis explicativas que possam aprimorar o modelo de uma forma mais robusta que a observada para os quatro indicadores adicionais inseridos, aqui, na tentativa de se alcançar o melhor modelo de previsão.

Links

- Repositório dos arquivos relativos a este trabalho no *GitHub*:

https://github.com/deboranson/tcc_puc_minas

- Vídeo da apresentação:

https://youtu.be/RCHHqRIm2xM