Projeto Final para Conclusão do Curso de Ciência de Dados

Tema: Expectativa de Vida

Aluno: Marcio Carvalho

Sumário Executivo

- Introdução
- Origem do Banco de Dados
- Hipóteses a serem estudadas
- 1ª Etapa do Estudo:
    - Coleta de Dados
    - Análise Exploratória de Dados
    - Preparação dos Dados
- 2ª Etapa do Estudo:
    - Modelagem
    - Análise Preditiva
    - Validação
- Conclusão
- Curiosidades
- Agradecimentos

Introdução

O que é Expectativa de Vida?

A expectativa de vida é número médio de anos que a população de um país viverá.

Portanto, o aumento da expectativa de vida está diretamente associado à melhoria das condições de vida da população.

Banco de Dados

- O repositório de dados é da Global Health Observatory (GHO), que é o portal de conjunto de dados estatísticos relacionados à saúde de 183 Estados Membros e gerenciado pela Organização Mundial da Saúde (OMS).

- Os dados econômicos foram coletados do site das Nações Unidas.

- Os dados foram coletados e organizados no Site: https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who. O responsável pela obtenção e organização dos dados é o Kumar Rajarshi.

Banco de Dados

- Período de 2000 a 2015
- Inclui dados da população, crescimento econômico e investimento de saúde e educação de (+/-) 180 países.
- Distribuídos em dois grupos de países, sendo: desenvolvidos e em desenvolvimento.

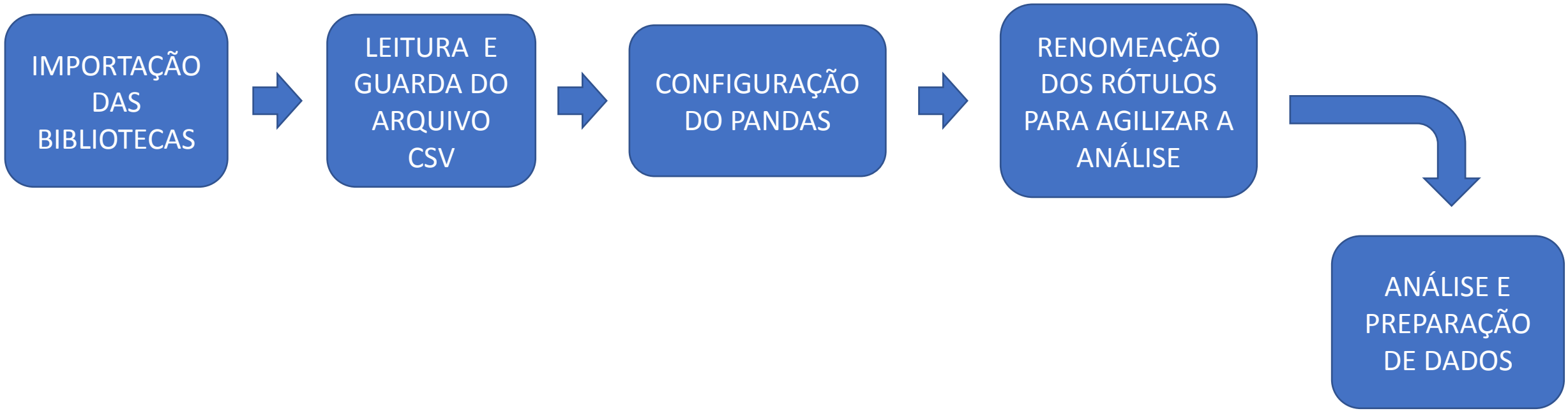
Hipóteses a Serem Estudadas

- Ajudar a sugerir quais as áreas que devem ser dado maior importância para melhorar a expectativa de vida.

- Avaliar se adotamos modelo preditivo único ou separado conforme os grupos de países.

1ª Etapa dos Estudos:

O conjunto de **dados foi baixado do arquivo csv: 'Life Expectancy Data.csv'**



Coleta de Dados

```python
import pandas as pd
import numpy as np
import missingno as msno
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import warnings
warnings.filterwarnings('ignore')
```

✓ 0.0s

```python
df = pd.read_csv('Life Expectancy Data.csv')
pd.set_option('display.max_columns',None)
df.columns = ['PaIs','Ano','Classificacao','Expectativa_de_Vida',
    'Taxa_Mortalidade_Adulto','Numero_Obito_Infantil','Consumo_Alcool',
    'Gastos_Com_Saude','Hepatite_B','Sarampo','Indice_Massa_Corporal',
    'Mortes_Menores_5anos','Cobertura_Polio','Relacao_GastosSaude/
    despesasTotais','Cobertura_Difeteria','Mortes_criancas_0-4a_Hiv_aids','PIB',
    'Populacao','Magreza_1-19anos','Magreza_5-9anos','IDH',
    'Anos_de_Escolaridade']
df.head()
```

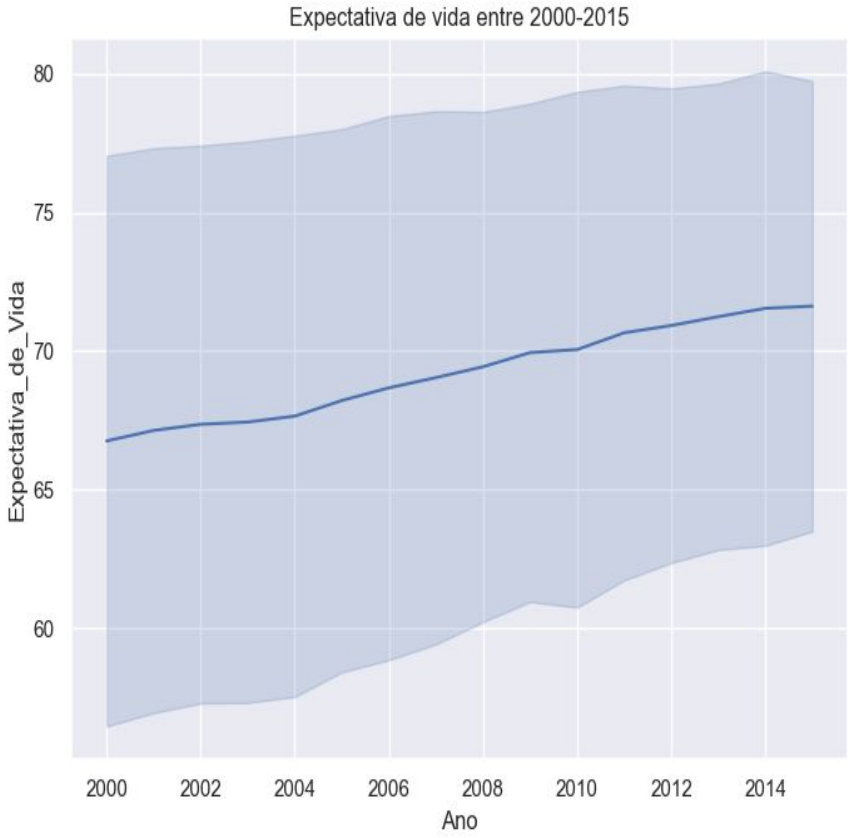
✓ 0.0s

Coleta de Dados

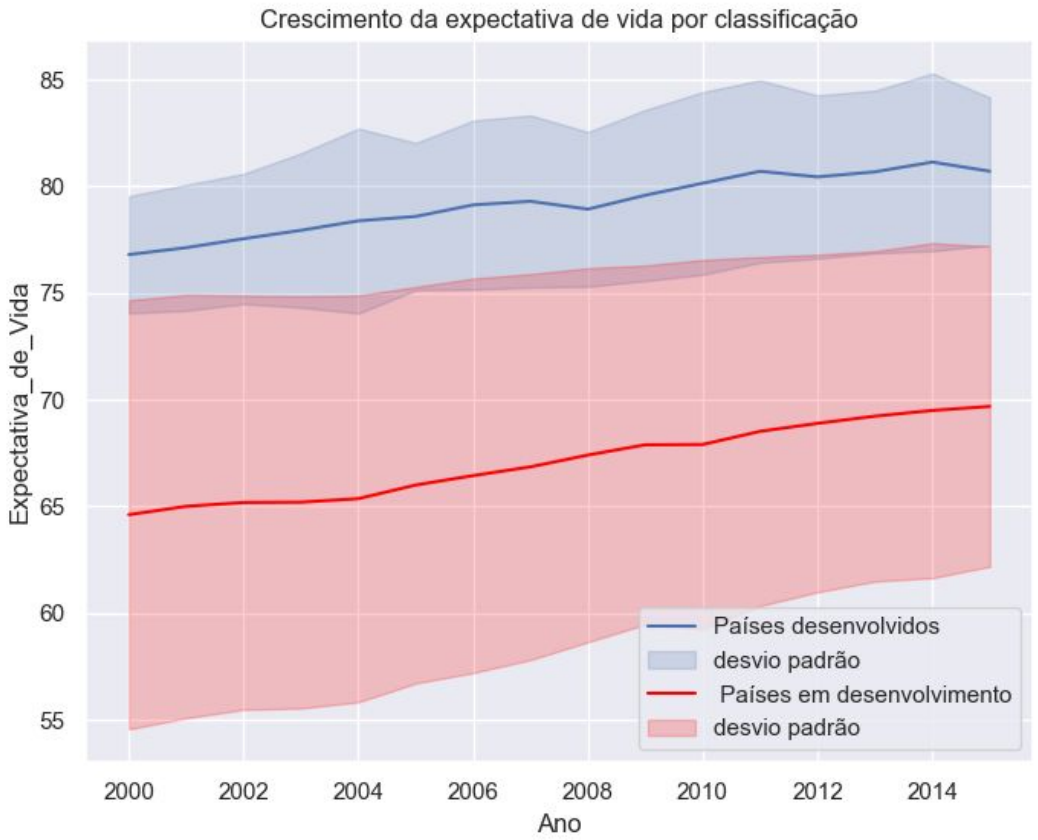
✓ 0.0s Python

	Pais	Ano	Classificacao	Expectativa_de_Vida	Taxa_Mortalidade_Adulto	Numero_Obito_Infantil	Consumo_Alcool	Gastos_Com_Sa
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.27
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.52
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.21
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.18
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.09

Análise Exploratória de Dados

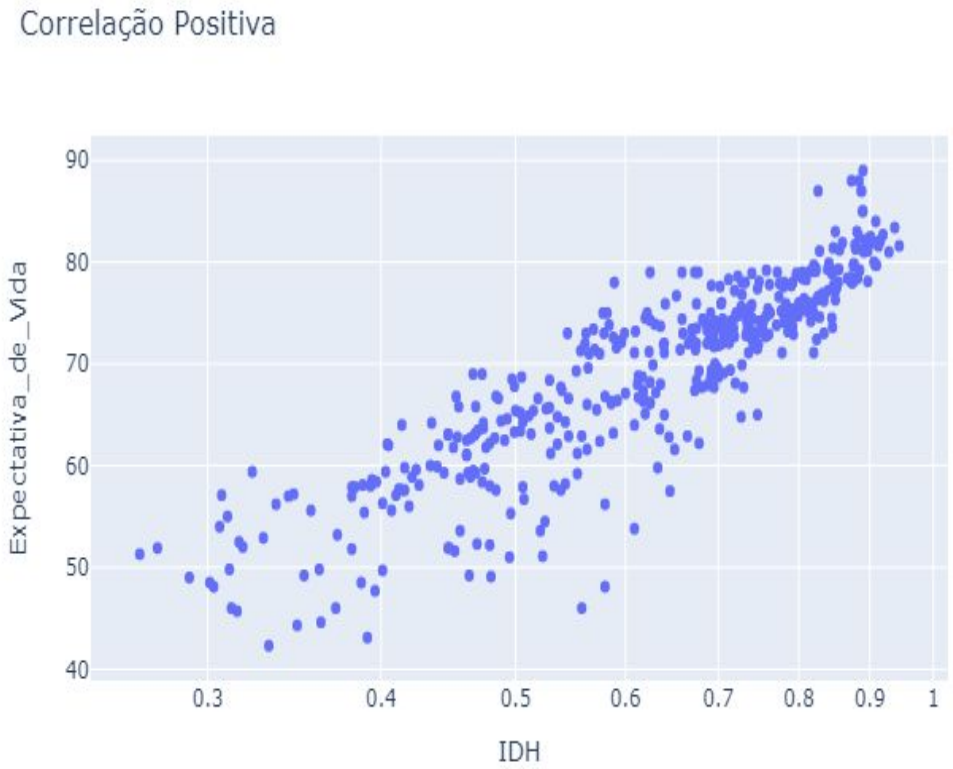


1.Conjunto de Dados Completos

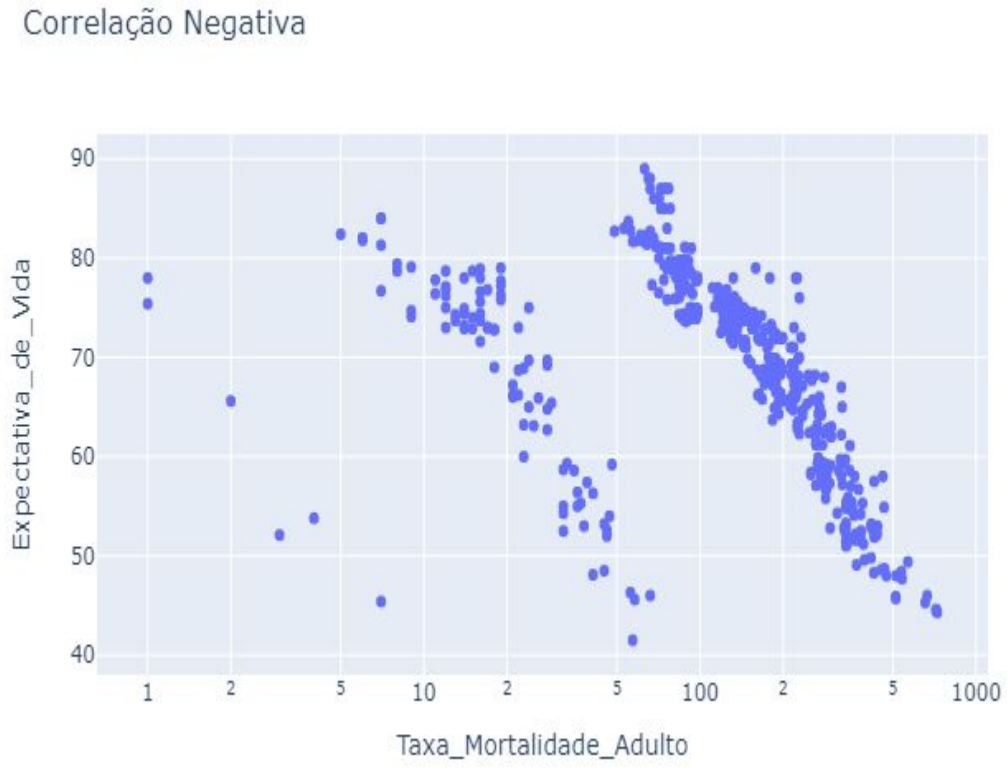


2.Conjunto de dados separados por grupo

Análise Exploratória de Dados

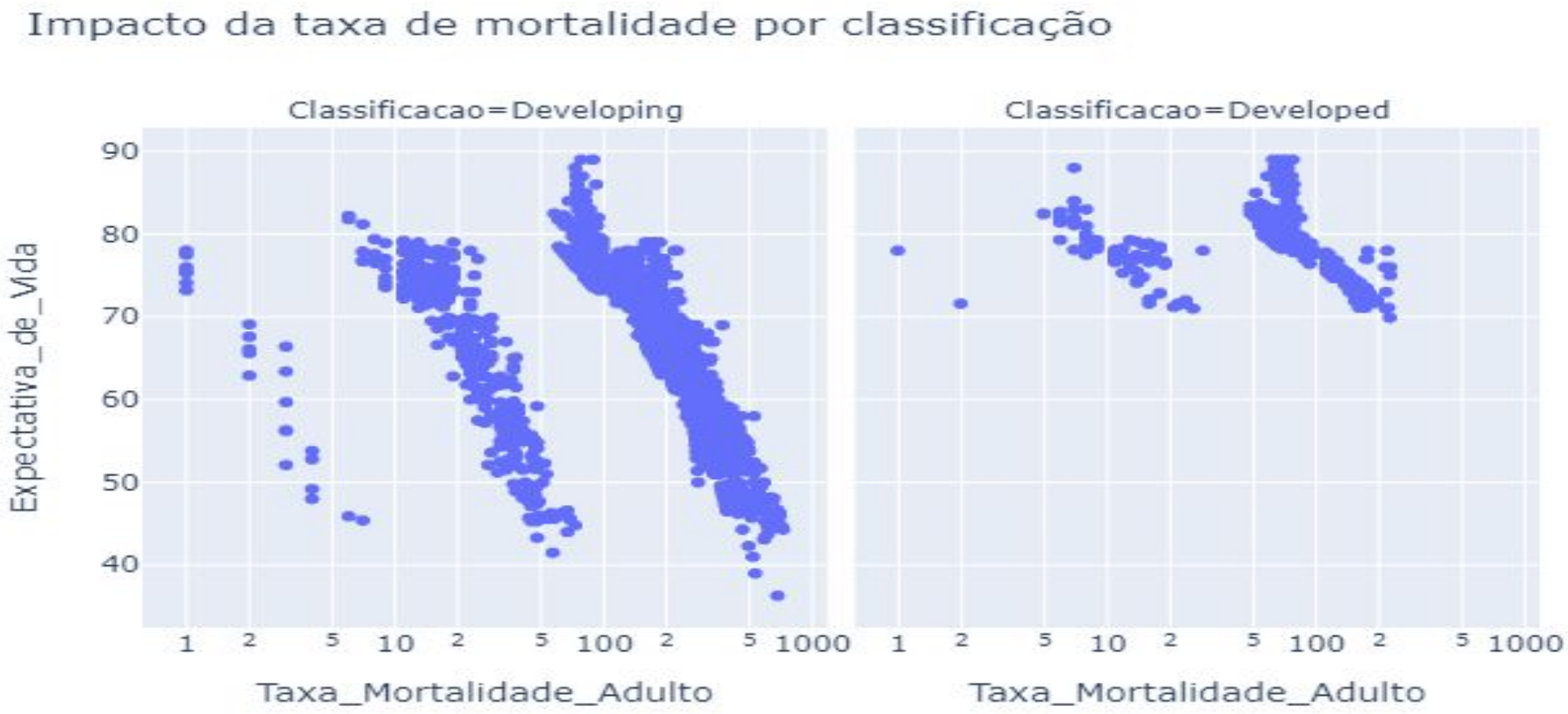


3.Grande Candidato na Importância do Modelo



4.Grande Candidato na Importância do Modelo

Análise Exploratória de Dados



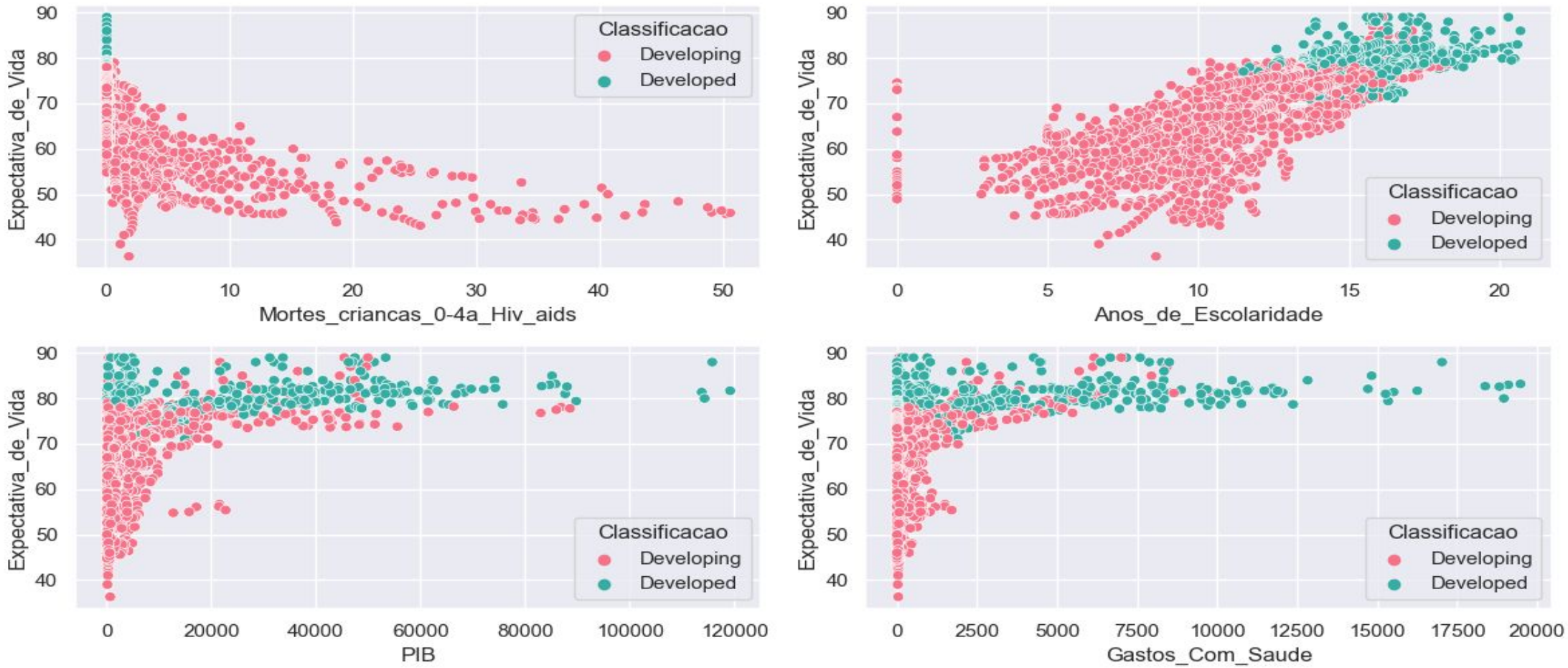
5.Taxa de Mortalidade Adulta

Análise Exploratória de Dados

```
1  coluna = ['Mortes_criancas_0-4a_Hiv_aids','Anos_de_Escolaridade','PIB',
       'Gastos_Com_Saude']
2  fig, axs = plt.subplots(2, 2, figsize=(12, 6))
3  axs = axs.flatten()
4  for i,var in enumerate(coluna):
5      ax = sns.scatterplot(x=var, y='Expectativa_de_Vida',hue='Classificacao',
           palette='husl', data=df, ax=axs[i])
6      ax.get_figure().savefig("Dispersão de variaveis.png")
7  fig.tight_layout()
```

5.Código dos Gráficos de Dispersão

Análise Exploratória de Dados



6. Gráficos de Dispersão

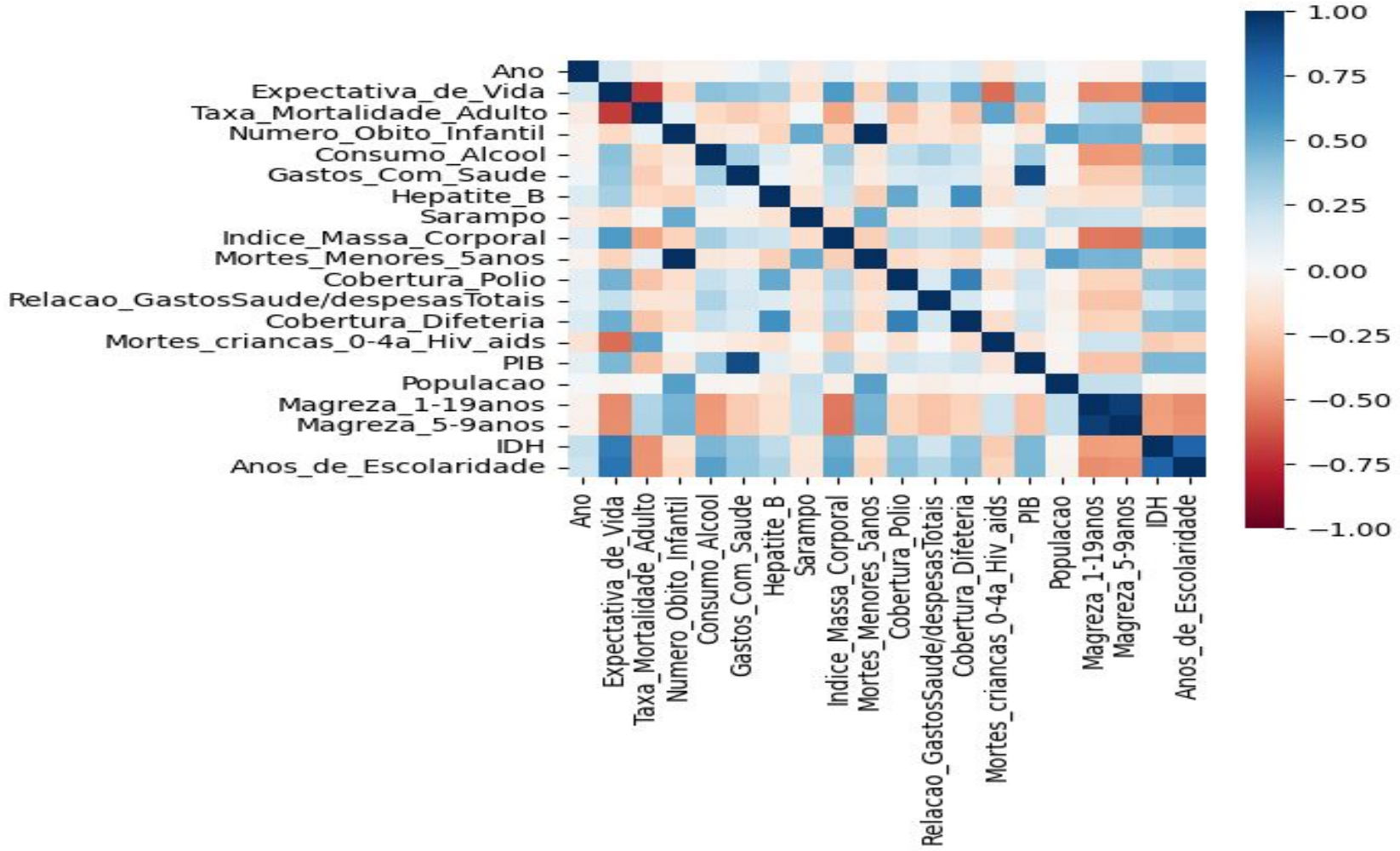
Análise Exploratória de Dados

➔ Alta correlação positiva:

- IDH,
- Anos de Escolaridade

➔ Alta correlação negativa:

- Taxa de mortalidade adulto



➔ Media Correlação positiva:

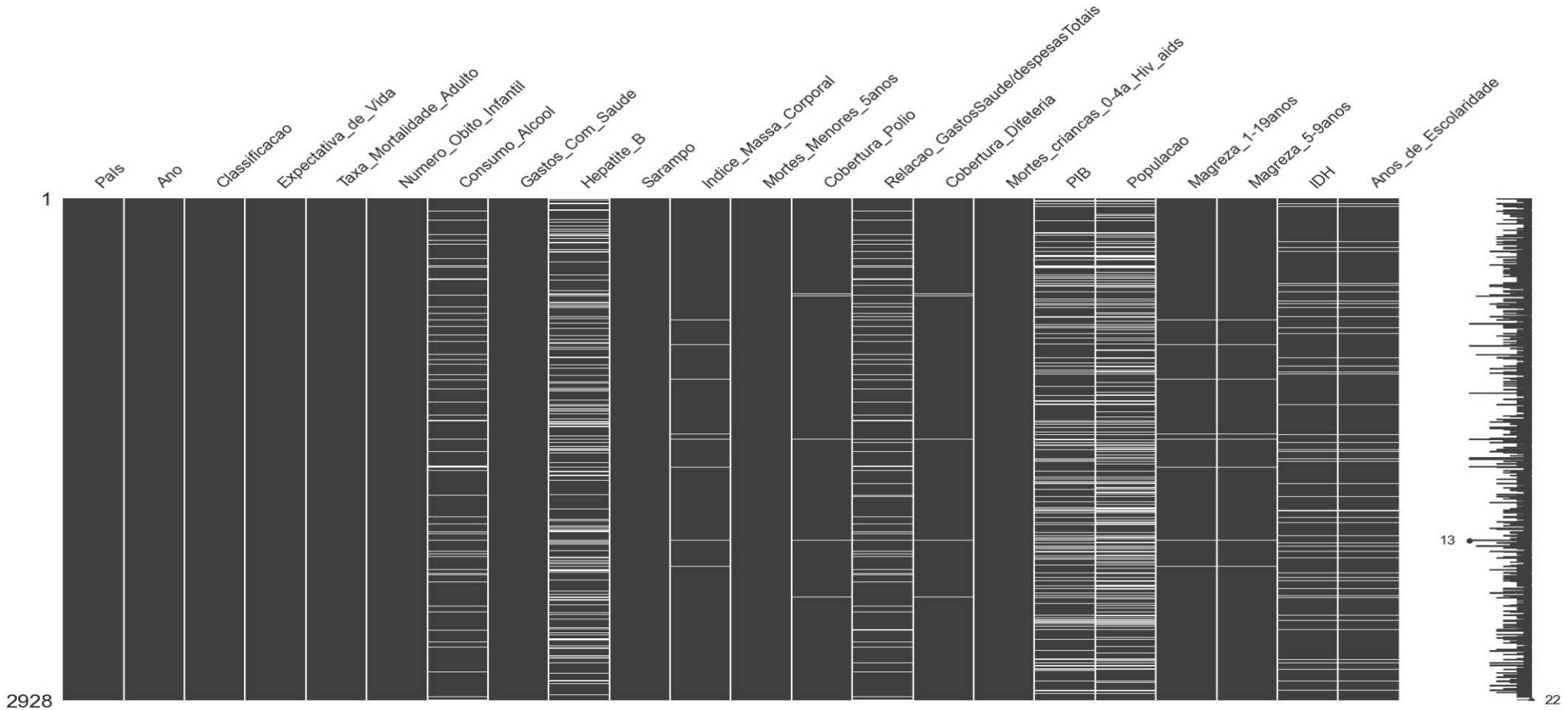
- Índice de massa corporal

➔ Média Correlação negativa:

- Morte Crianças 0-4 anos Hiv/Aids

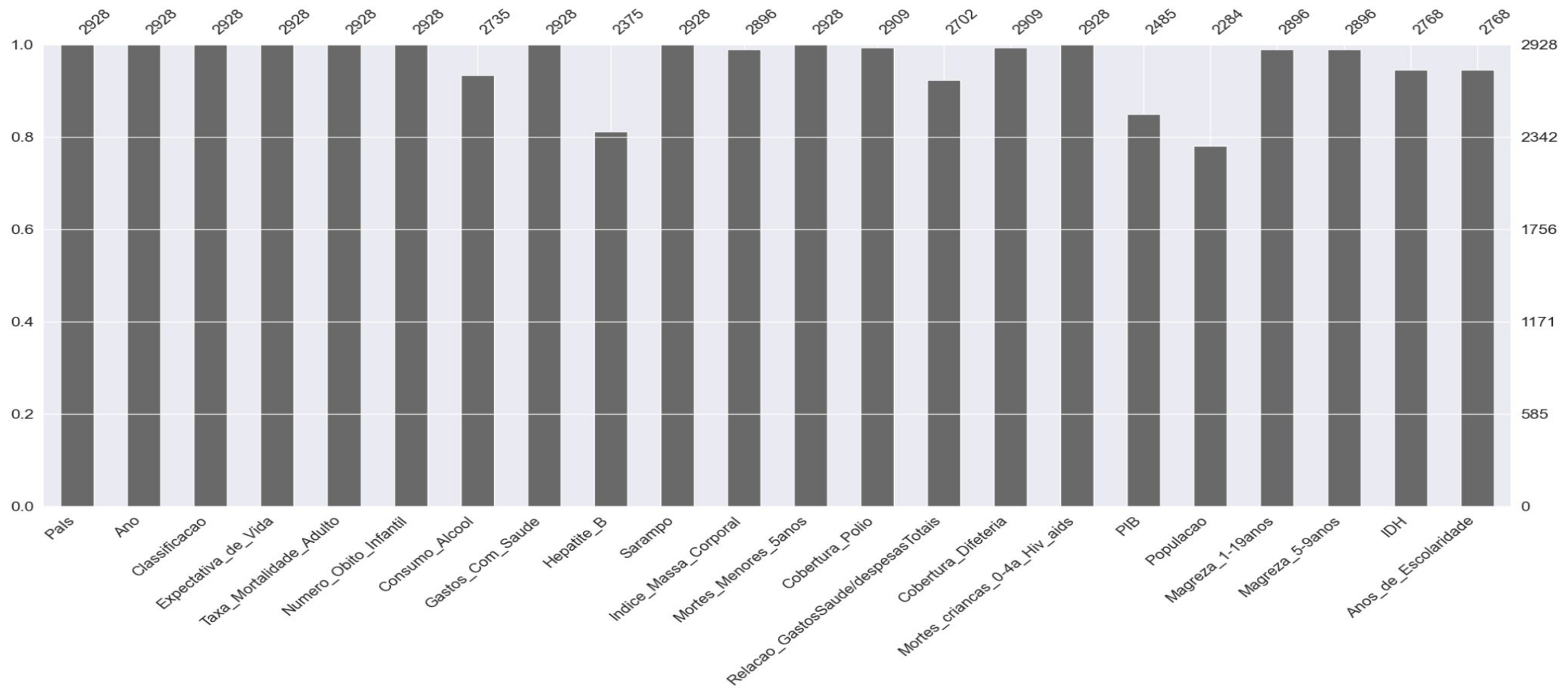
7. Matriz de Correlação

Preparação dos Dados



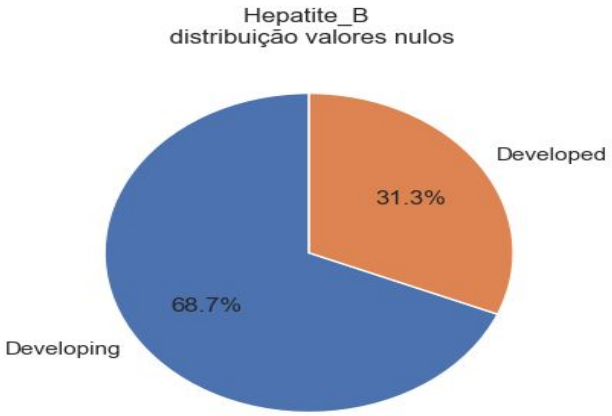
8. Distribuição por Colunas de Valores Nulos

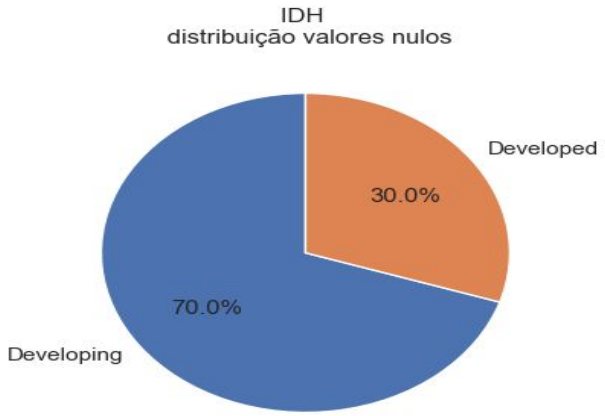
Preparação dos Dados

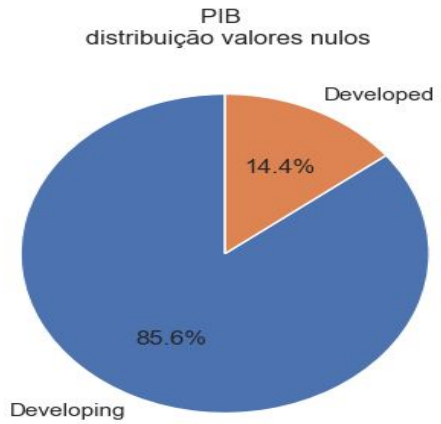


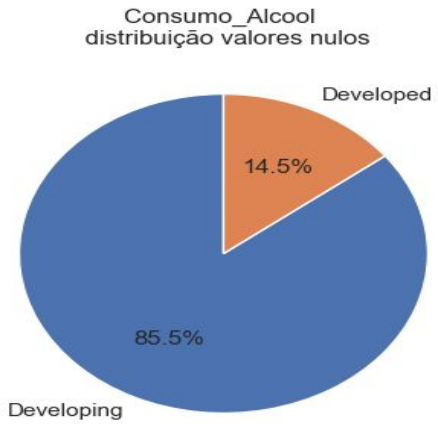
9. Gráfico de Barras Valores Nulos

Preparação dos Dados





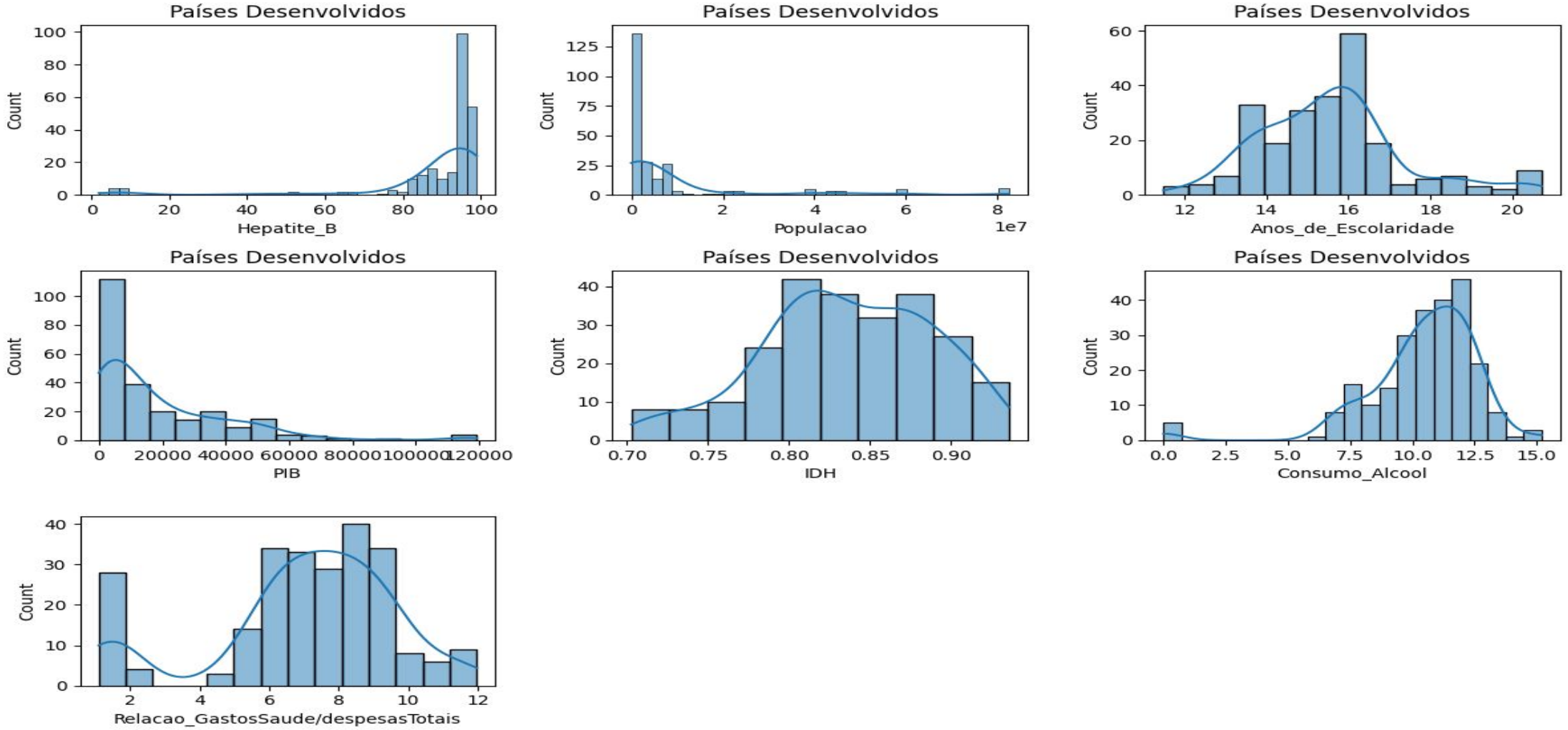




Valores nulos desbalanceados

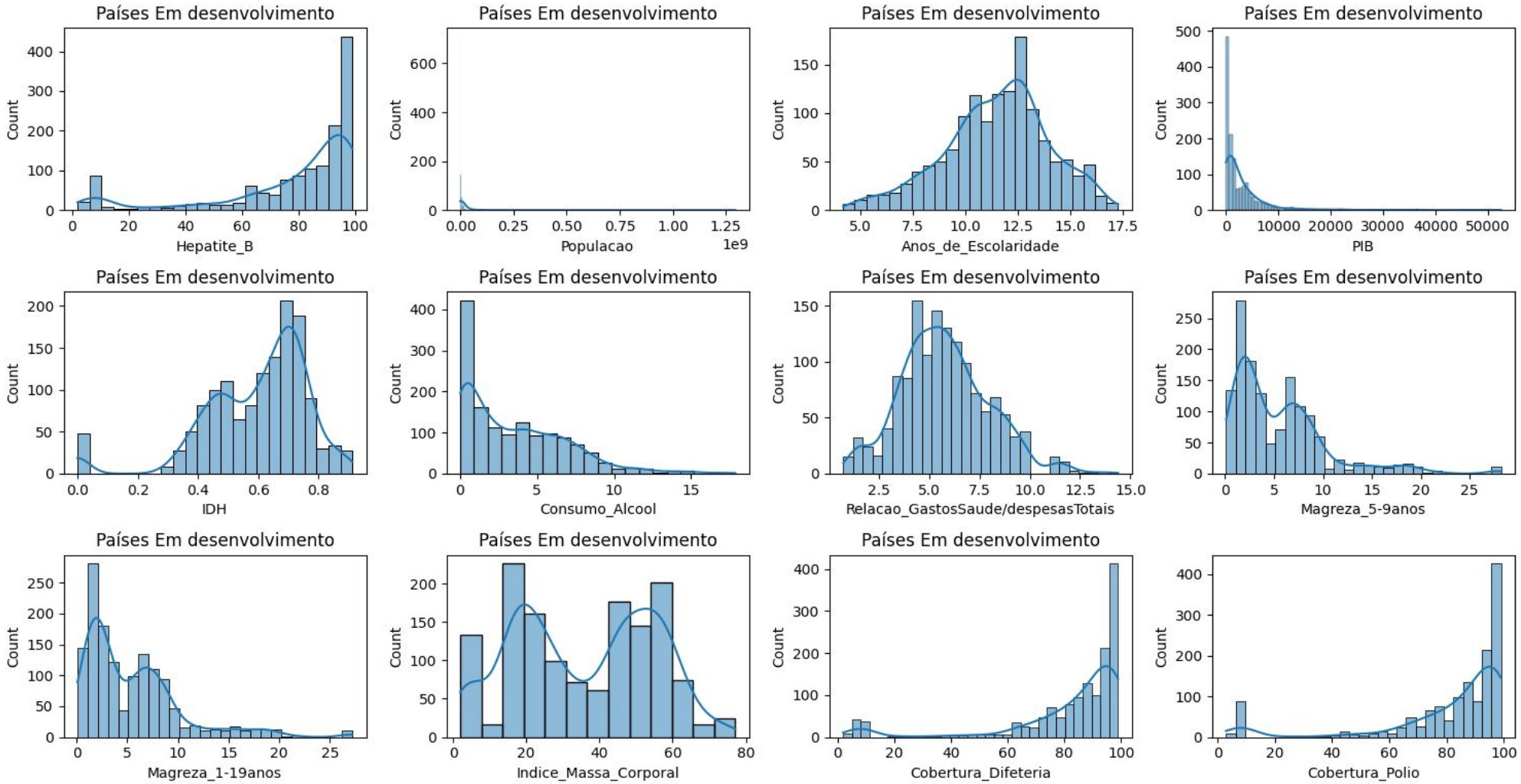
10. Valores Nulos por Classificação

Preparação dos Dados



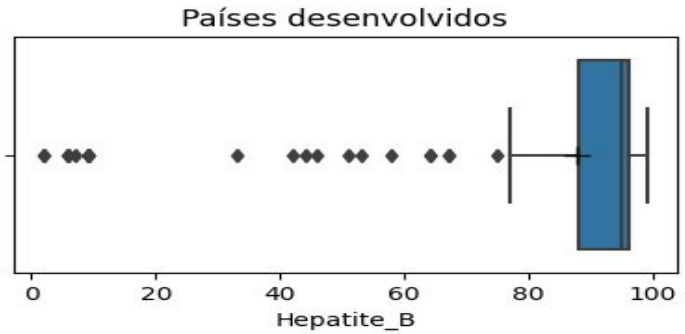
11. Histograma sem valores nulos

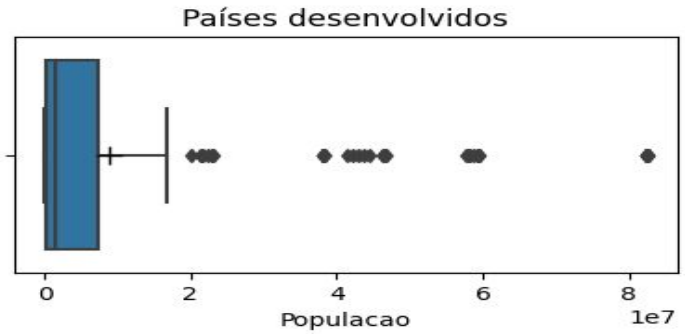
Preparação dos Dados

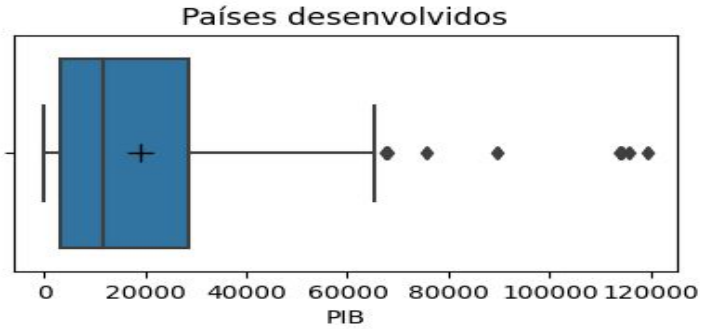


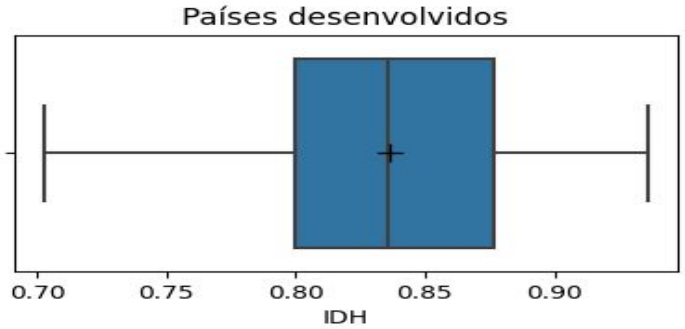
12. Atributos sem valores nulos por Classificação

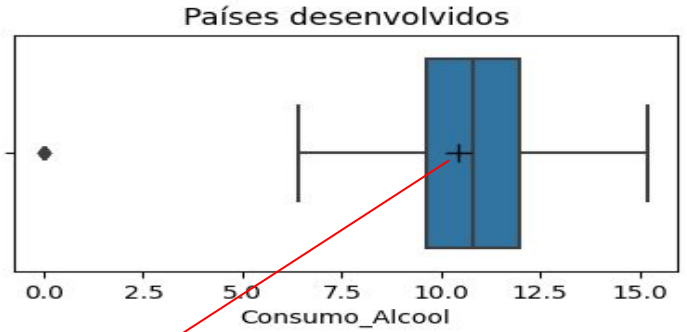
Preparação dos Dados

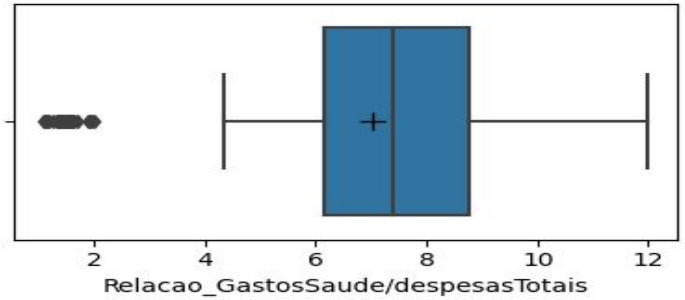








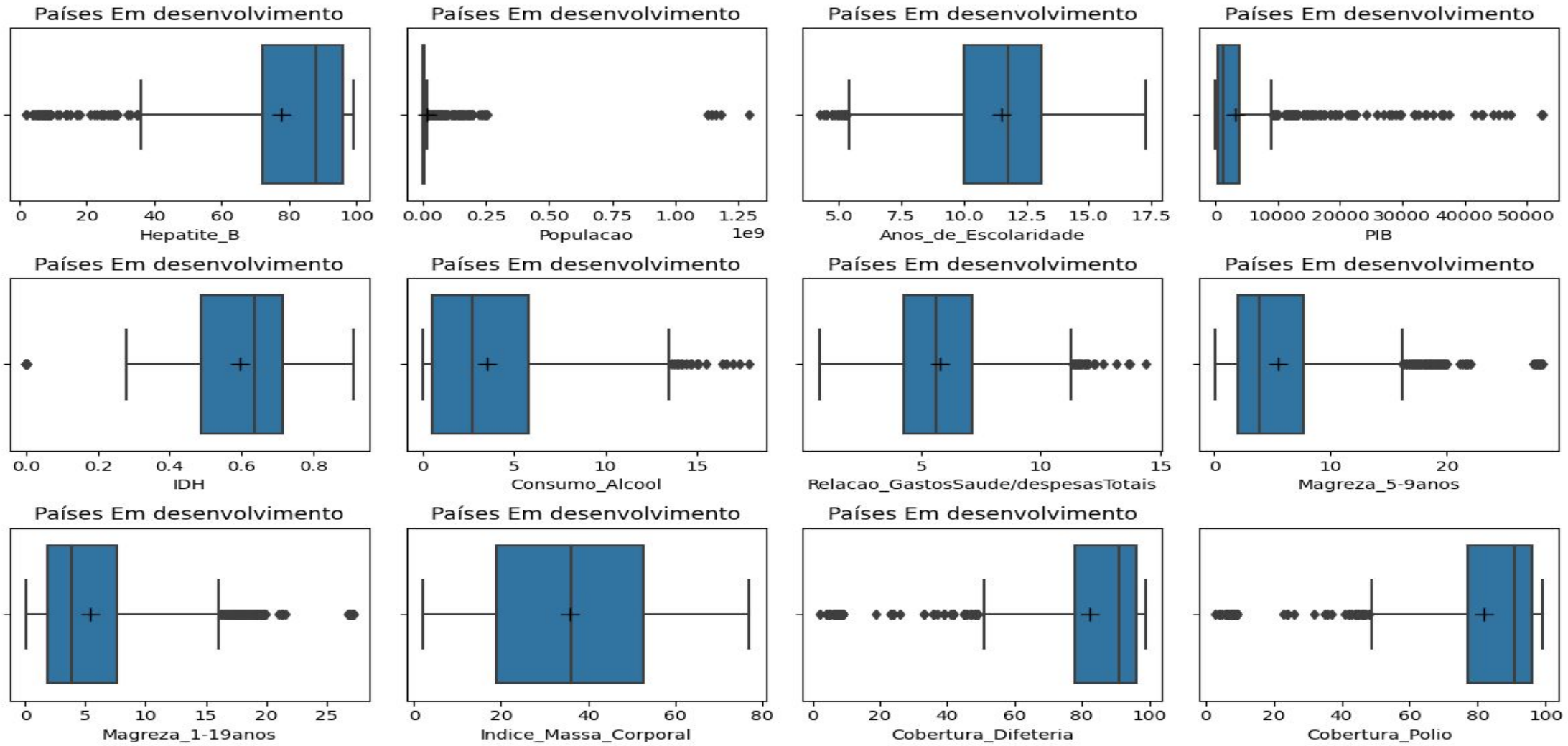




"+" MÉDIA

13. Atributos sem valores nulos

Preparação dos Dados



14. Atributos sem valores nulos

Preparação dos Dados

Código para imputação de valores parcialmente faltantes por país

```
1  # Substituir os dados parcialmente faltantes com a média 2000-2015 para cada
       país.
2  def imputar (df):
3      media = df.mean()
4      return df.fillna(media)
5  df_full = df.groupby('PaIs').apply(imputar)
```

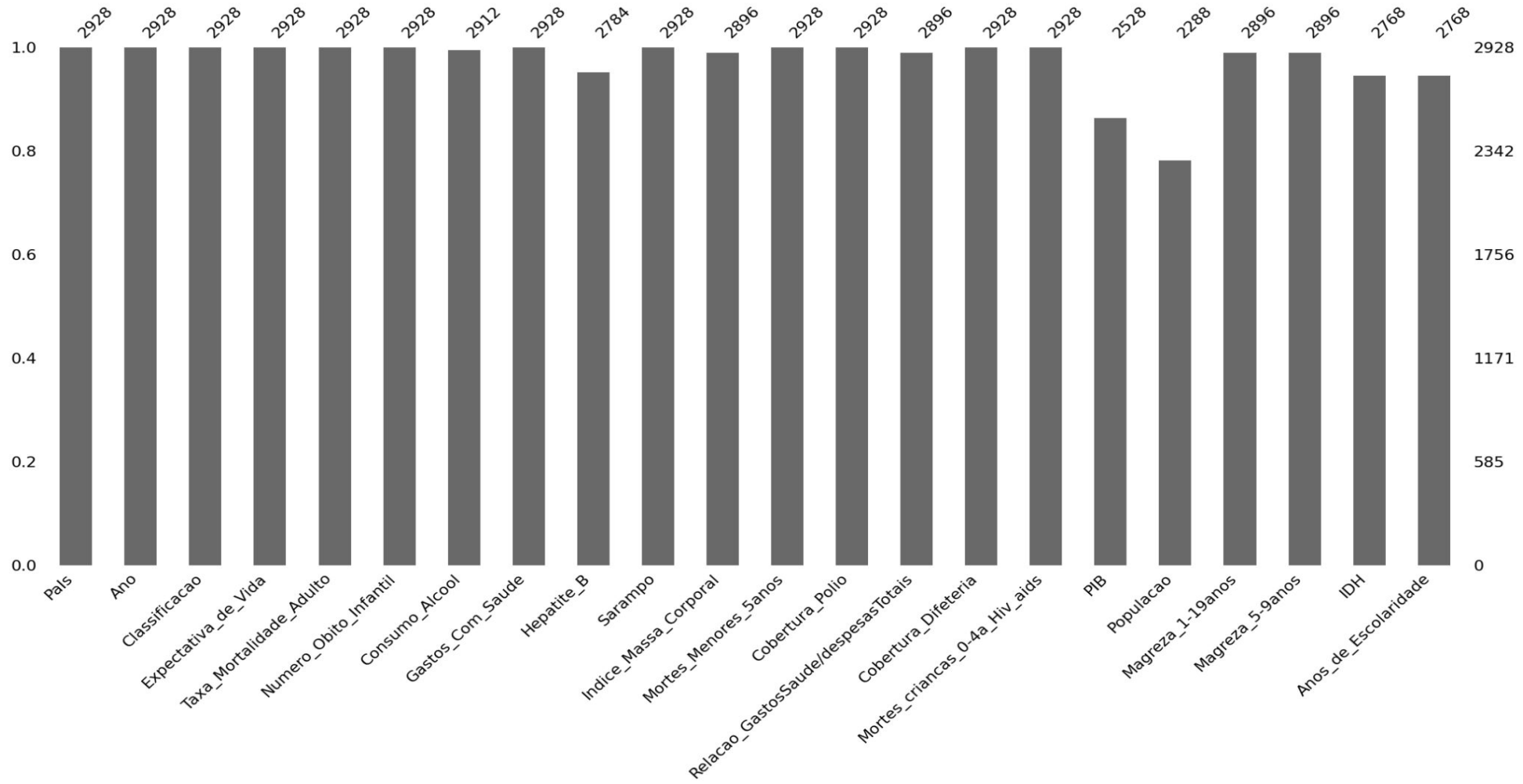
✓ 1.4s

Código para imputação de valores totalmente faltantes por país

```
1  # Substituir os dados totalmente faltantes por país considerando a mediana
       separada por países desenvolvidos e países em desenvolvimento.
2  def imputar (df):
3      mediana = df.median()
4      return df.fillna(mediana)
5  df_full_final = df_full.groupby('Classificacao').apply(imputar)
```

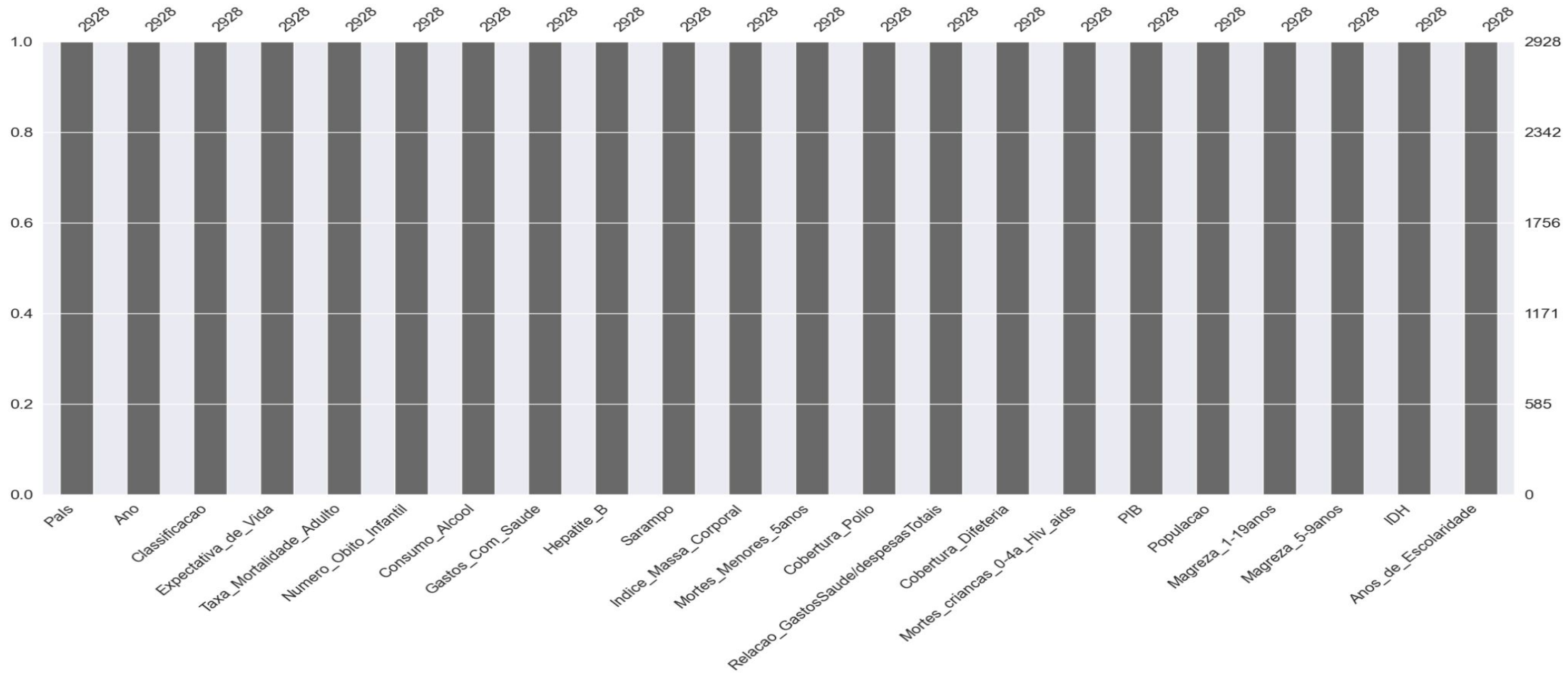
✓ 0.0s

Preparação dos Dados



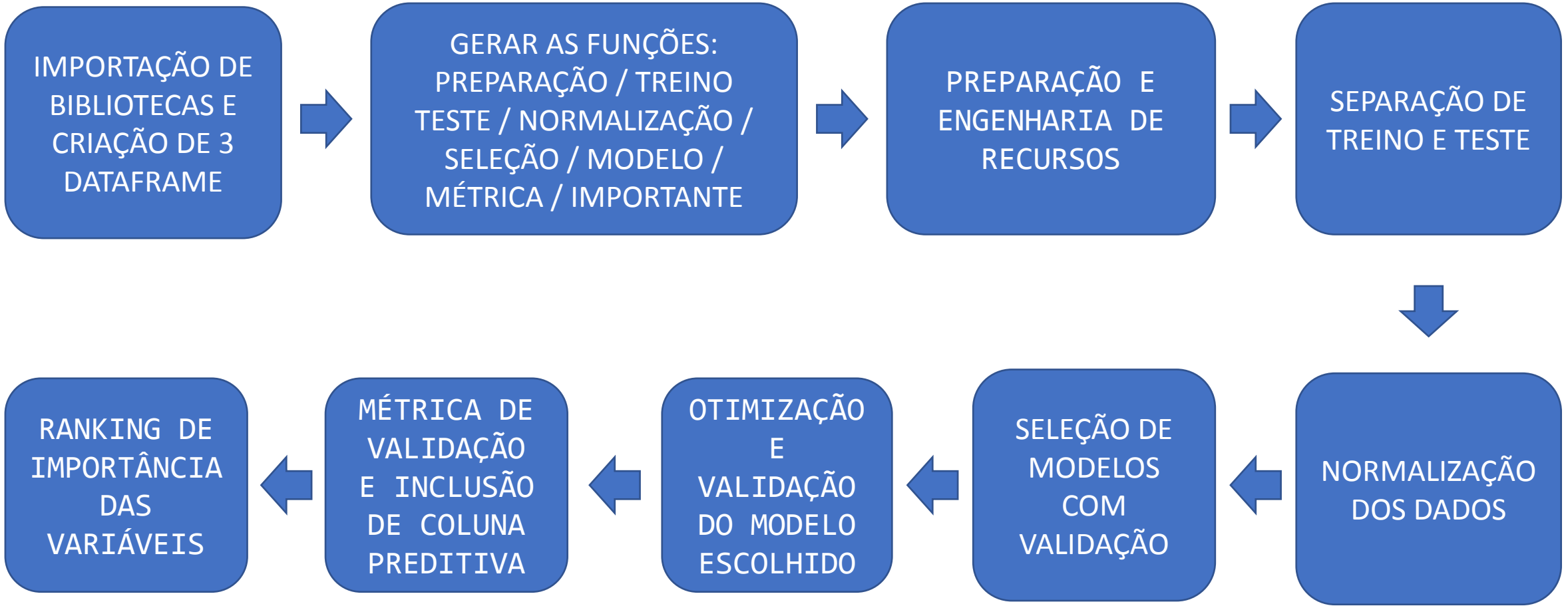
15. Conjunto de Dados com preparação agrupando os países

Preparação dos Dados



16. Conjunto de Dados com preparação agrupando por classificação

2ª Etapa do Estudo



17. Fluxograma criação de modelo de regressão

Modelagem dos Dados

```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.dummy import DummyRegressor
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.neural_network import MLPRegressor
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import KFold
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import cross_val_score
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')

```

✓ 9.9s

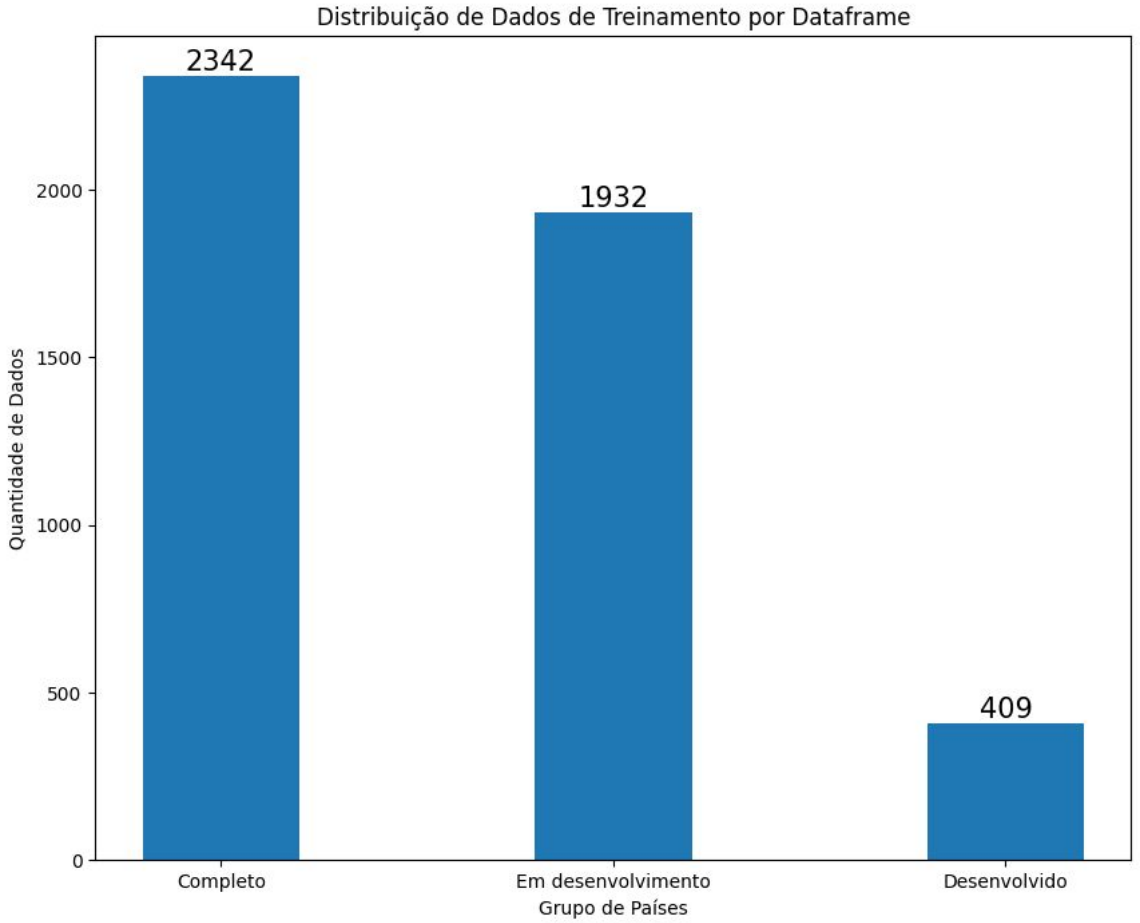
```python
def preparacao(def_model):
    df_model = def_model.drop(columns=['PaIs','Ano'])
    df_model = pd.get_dummies(df_model,columns=['Classificacao'])
    x = df_model.drop(columns='Expectativa_de_Vida')
    y = df_model.Expectativa_de_Vida
    return x , y
```

✓ 0.0s

```python
def treino_teste (X,Y):
    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,
        random_state=20)
    return X_train, X_test, Y_train, Y_test
```

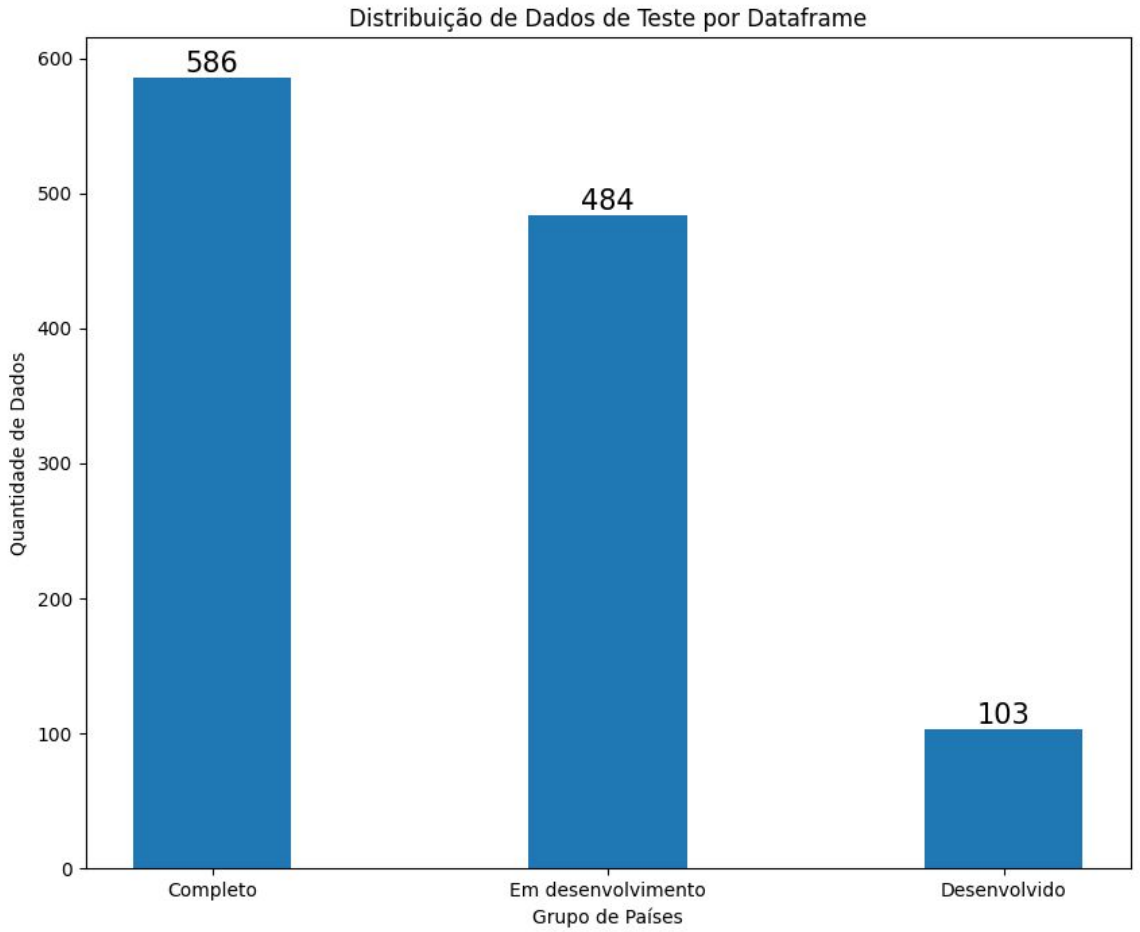
✓ 0.0s

Modelagem dos Dados



18. Conjunto de Dados de Treinamento

Modelagem dos Dados



19. Conjunto de Dados de Teste

Modelagem e Predição dos Dados

```python
def normalizacao(X_train, X_test):
    scaler = StandardScaler()
    scaler_fit = scaler.fit(X_train)
    X_train_norm = scaler_fit.transform(X_train)
    X_train_dfnorm = pd.DataFrame(X_train_norm,columns=X_train.columns)
    X_test_norm = scaler_fit.transform(X_test)
    X_test_dfnorm = pd.DataFrame(X_test_norm,columns=X_train.columns)
    return X_train_norm , X_train_dfnorm , X_test_norm , X_test_dfnorm,
        scaler_fit
```

✓ 0.0s

- Normalização = StandardScaler

- Cross Validation = Kfold

```python
def selecao(X_train , X_test , Y_train , Y_test):
    X_rank = pd.concat([X_train, X_test])
    Y_rank = pd.concat([Y_train, Y_test])
    for model in [DummyRegressor,SVR,DecisionTreeRegressor,
      RandomForestRegressor,GradientBoostingRegressor,MLPRegressor]:
        cls = model()
        kfold = KFold(n_splits=10,shuffle=True, random_state=20 )
        s = cross_val_score(cls, X_rank, Y_rank,
          scoring="neg_mean_squared_error", cv=kfold)
        print(f"{model.__name__:22} MSE: "f"{s.mean():.3f} STD: {s.std():.2f}")
```

✓ 0.0s

Predição dos Dados

```
Dataset todos os países:
DummyRegressor          MSE: -90.759 STD: 5.19
SVR                     MSE: -11.957 STD: 1.54
DecisionTreeRegressor   MSE: -6.500 STD: 0.87
RandomForestRegressor   MSE: -3.175 STD: 0.52
GradientBoostingRegressor MSE: -5.003 STD: 0.64
MLPRegressor            MSE: -14.869 STD: 1.48
 Dataset países desenvolvidos:
DummyRegressor          MSE: -15.453 STD: 3.20
SVR                     MSE: -5.359 STD: 1.41
DecisionTreeRegressor   MSE: -5.910 STD: 2.58
RandomForestRegressor   MSE: -2.952 STD: 0.72
GradientBoostingRegressor MSE: -3.496 STD: 0.78
MLPRegressor            MSE: -418.718 STD: 73.43
 Dataset países em desenvolvimento:
DummyRegressor          MSE: -81.174 STD: 3.42
SVR                     MSE: -12.973 STD: 1.57
DecisionTreeRegressor   MSE: -7.109 STD: 1.27
RandomForestRegressor   MSE: -3.318 STD: 0.60
GradientBoostingRegressor MSE: -4.956 STD: 0.73
MLPRegressor            MSE: -34.685 STD: 2.25
```

- Random Forest Regressor foi escolhido para Otimização e Validação

```python
def modelo (x_train , y_train):
    param_grid = {
     'bootstrap': [True],
     'max_depth': [20],
     'max_features': [0.4],
     'min_samples_leaf': [1],
     'min_samples_split': [2,3],
     'n_estimators': [700,800],
     'random_state': [20],
    }
    rf = RandomForestRegressor()
    grid_search = GridSearchCV(estimator = rf, param_grid = param_grid,
                               cv = 10, n_jobs = -1, verbose = 3,
                                  scoring='neg_mean_squared_error',
                                  return_train_score=True)
    grid_search.fit(x_train, y_train)
    return grid_search
```

Validação de Dados

```python
def metrica(y_test,x_test,model):
    metrica = mean_squared_error(y_test,model.predict(x_test),squared=False)
    return metrica
```

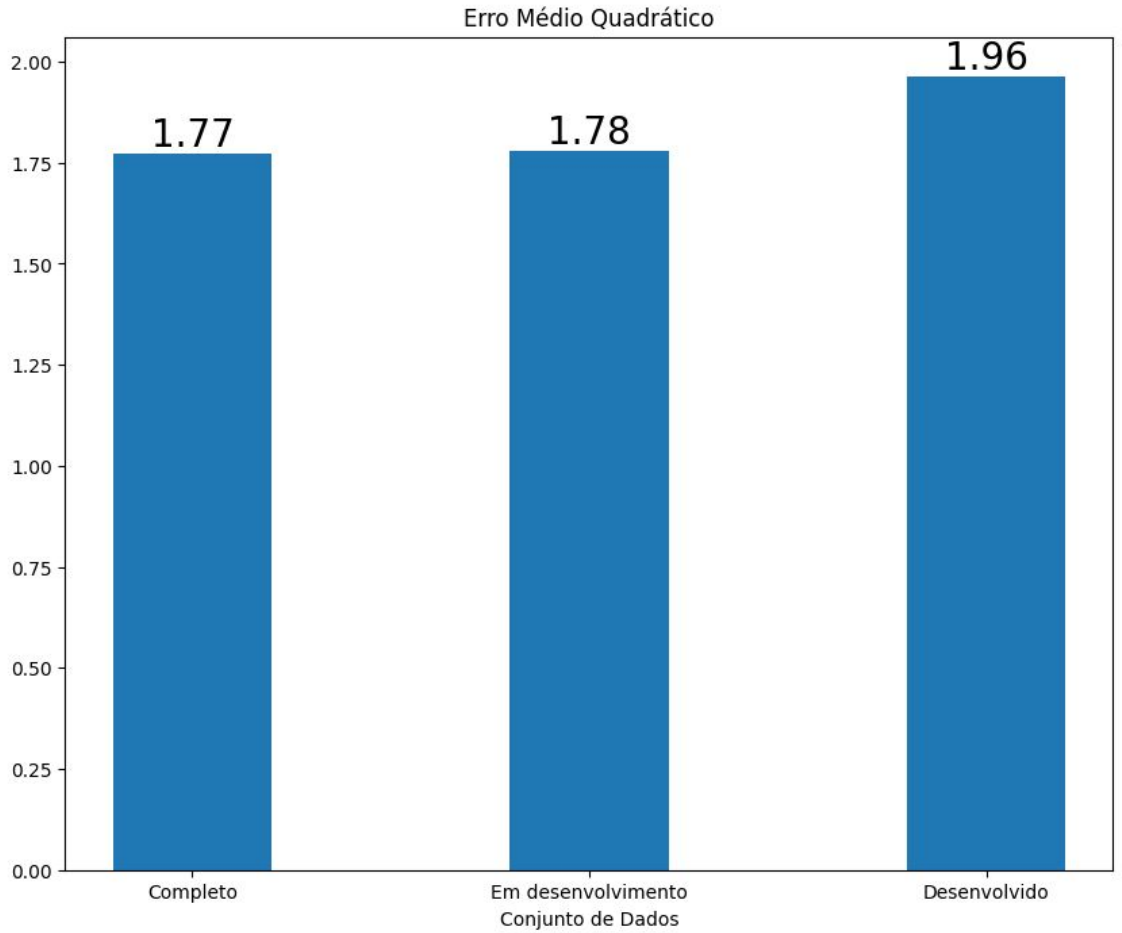
✓ 0.0s

```python
def importante (x_train,feature):
    imp = pd.DataFrame({"Variáveis": x_train.columns,"importantes": feature})
    var_imp = imp.sort_values(by='importantes',ascending=False).reset_index
        (drop=True)
    return var_imp
```


20. Código de criação do modelo

Validação de Dados

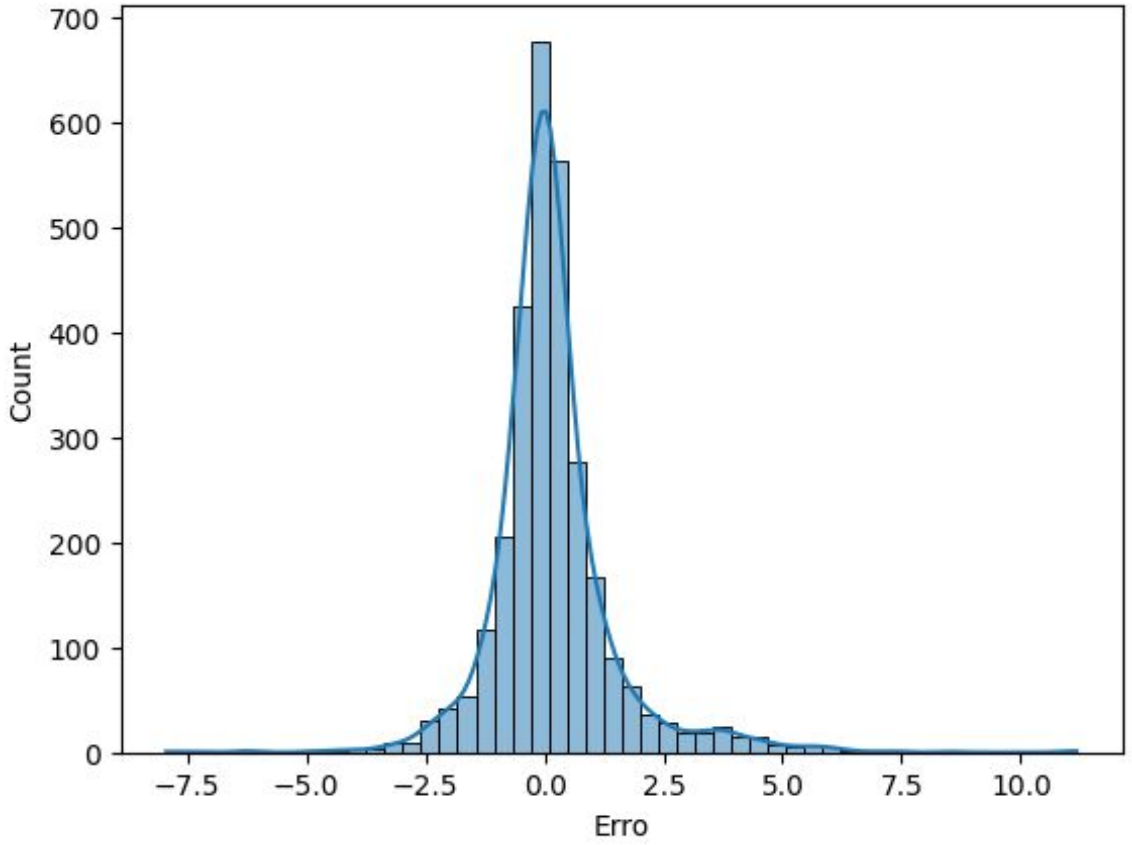
- Conjunto com todos os países, ligeiramente, teve o menor erro.



20. Erro Médio Quadrático

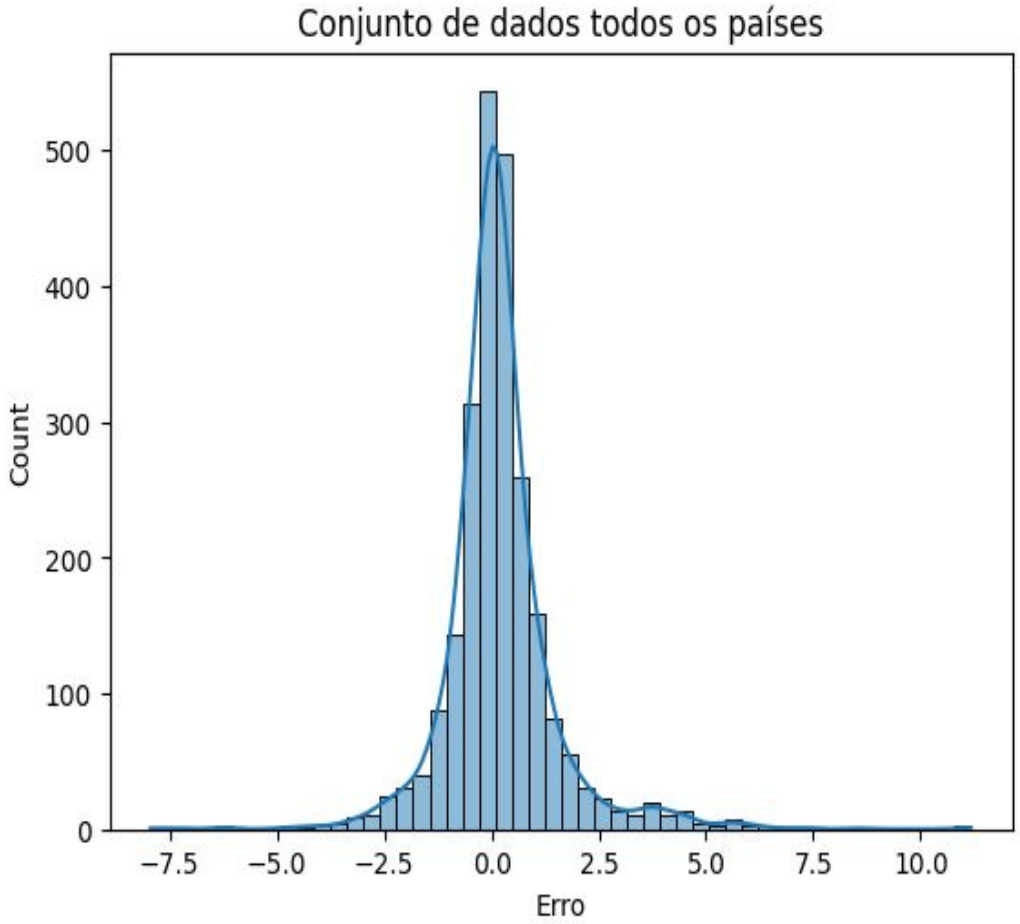
Validação dos Dados

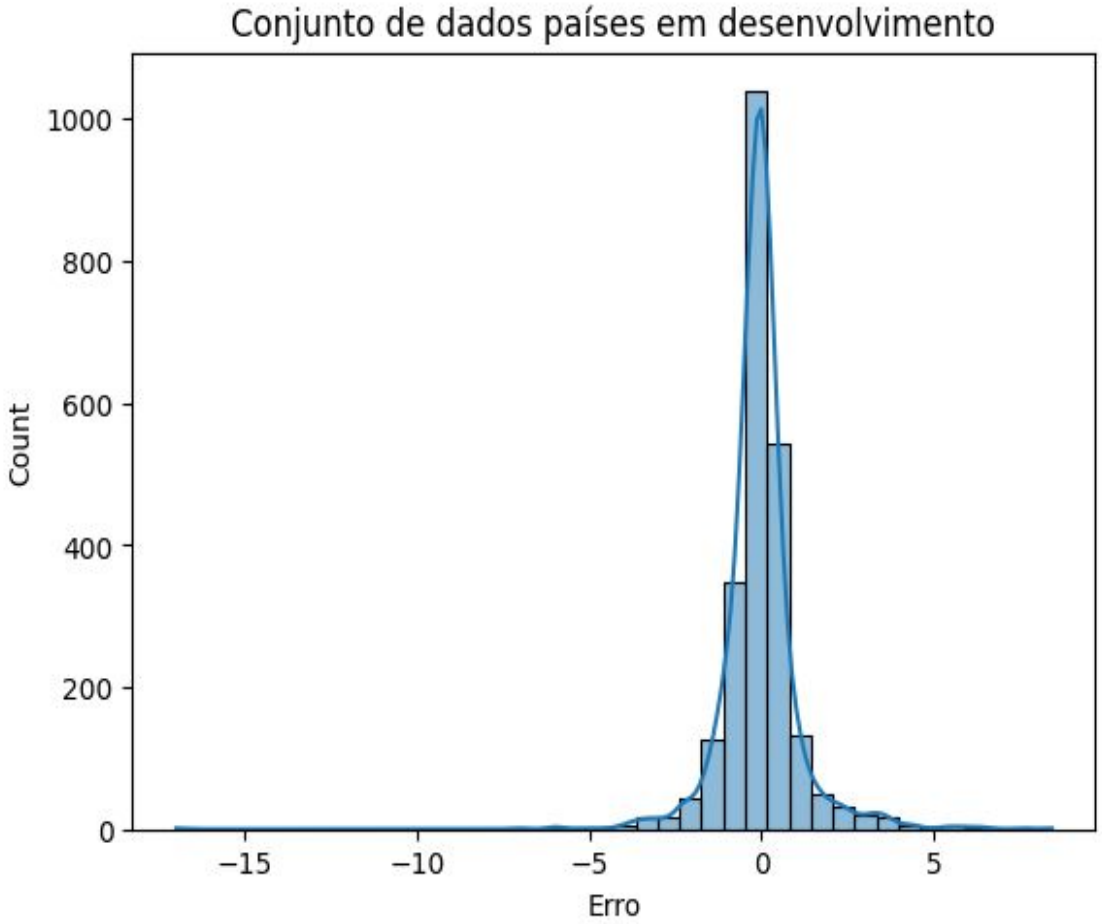
	Pais	Ano	Classificacao	Expectativa_Vida_real	Expectativa_Vida_previsto	Erro
0	Afghanistan	2015	Developing	65.0	63.348404	1.651596
20	Albania	2011	Developing	76.6	76.456180	0.143820
40	Algeria	2007	Developing	73.8	73.335380	0.464620
60	Angola	2003	Developing	46.8	47.632599	-0.832599
80	Argentina	2015	Developing	76.3	76.416321	-0.116321
...	...	...	...	...	...	...
2840	Vanuatu	2007	Developing	73.0	72.877812	0.122188
2860	Venezuela (Bolivarian Republic of)	2003	Developing	72.4	72.627280	-0.227280
2880	Yemen	2015	Developing	65.7	64.610620	1.089380
2900	Zambia	2011	Developing	58.2	57.960701	0.239299
2920	Zimbabwe	2007	Developing	46.6	48.752486	-2.152486



21. Conjunto de dados para todos os países

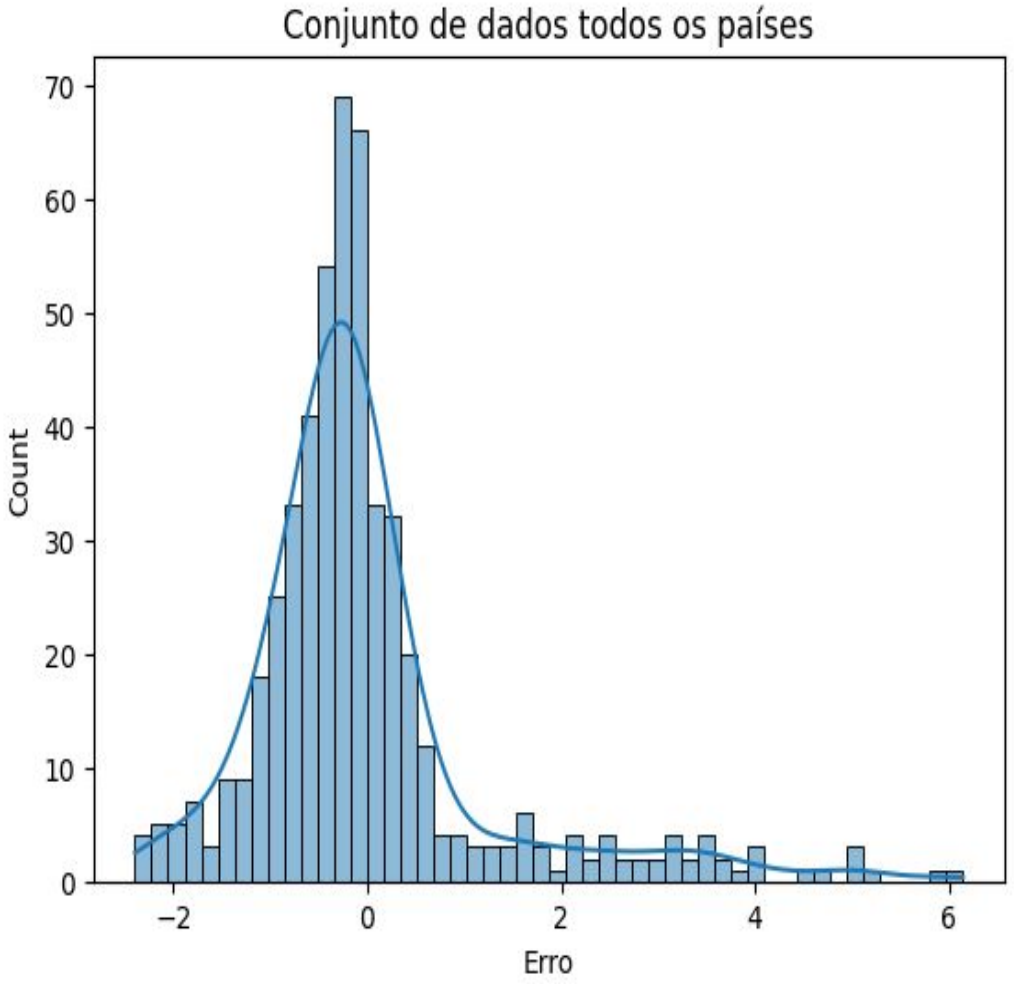
Validação de Dados

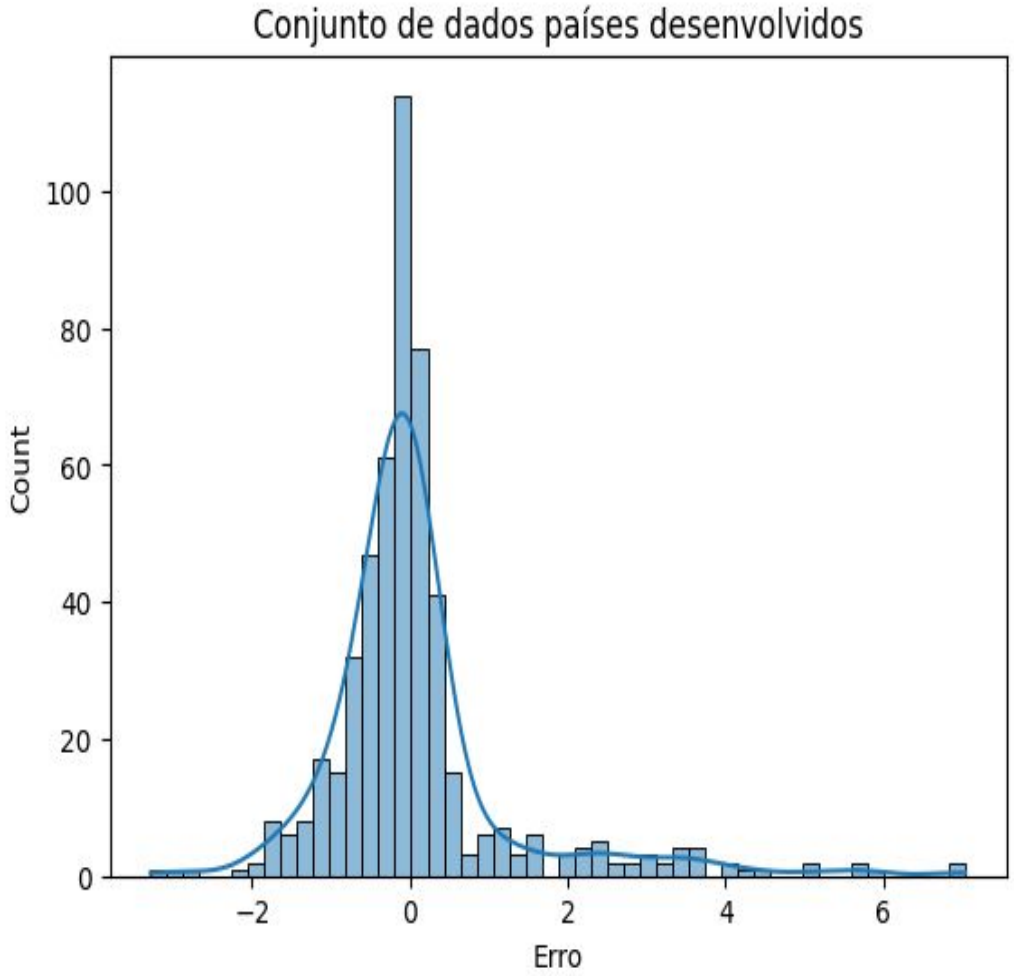




22. Países Em Desenvolvimento

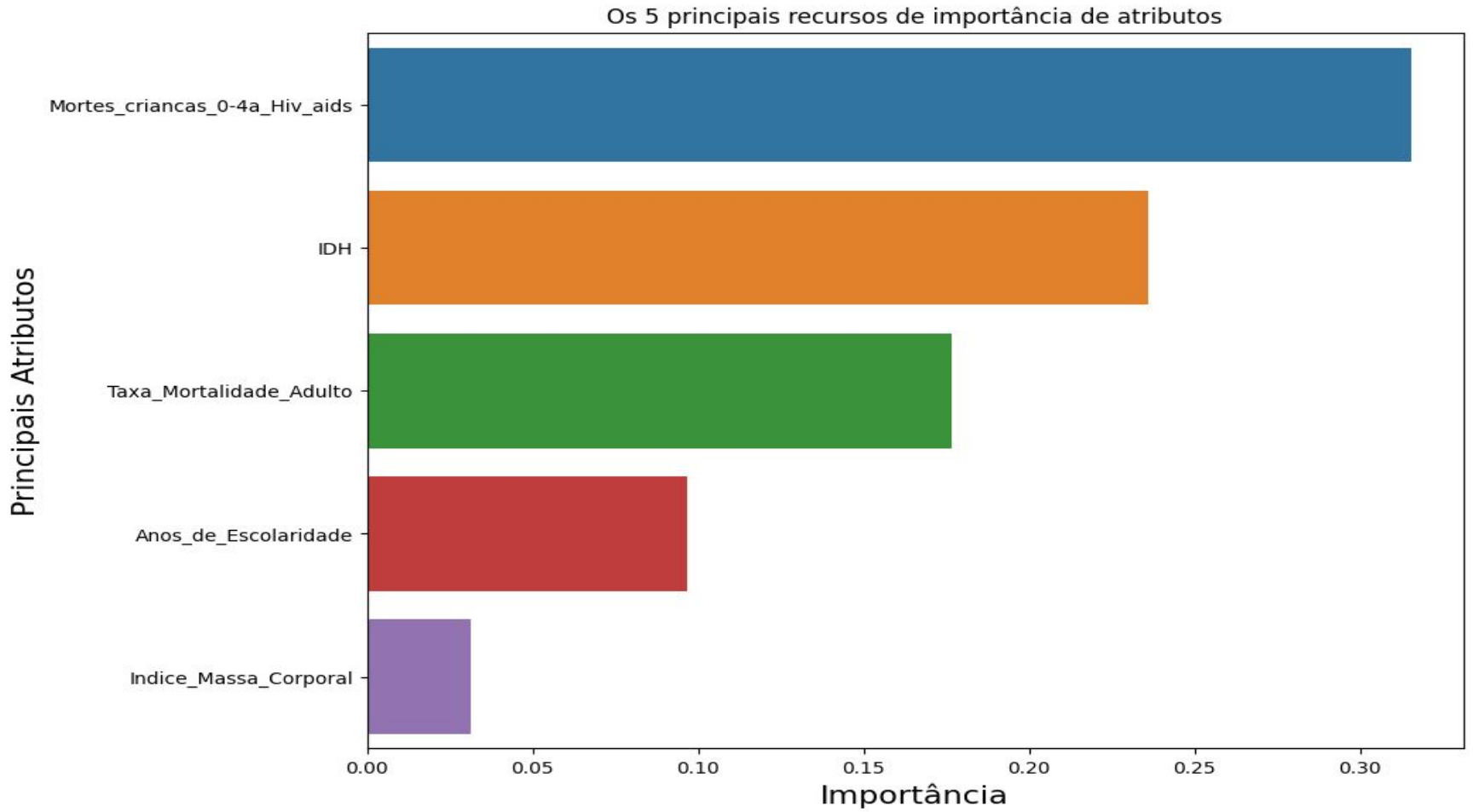
Validação de Dados





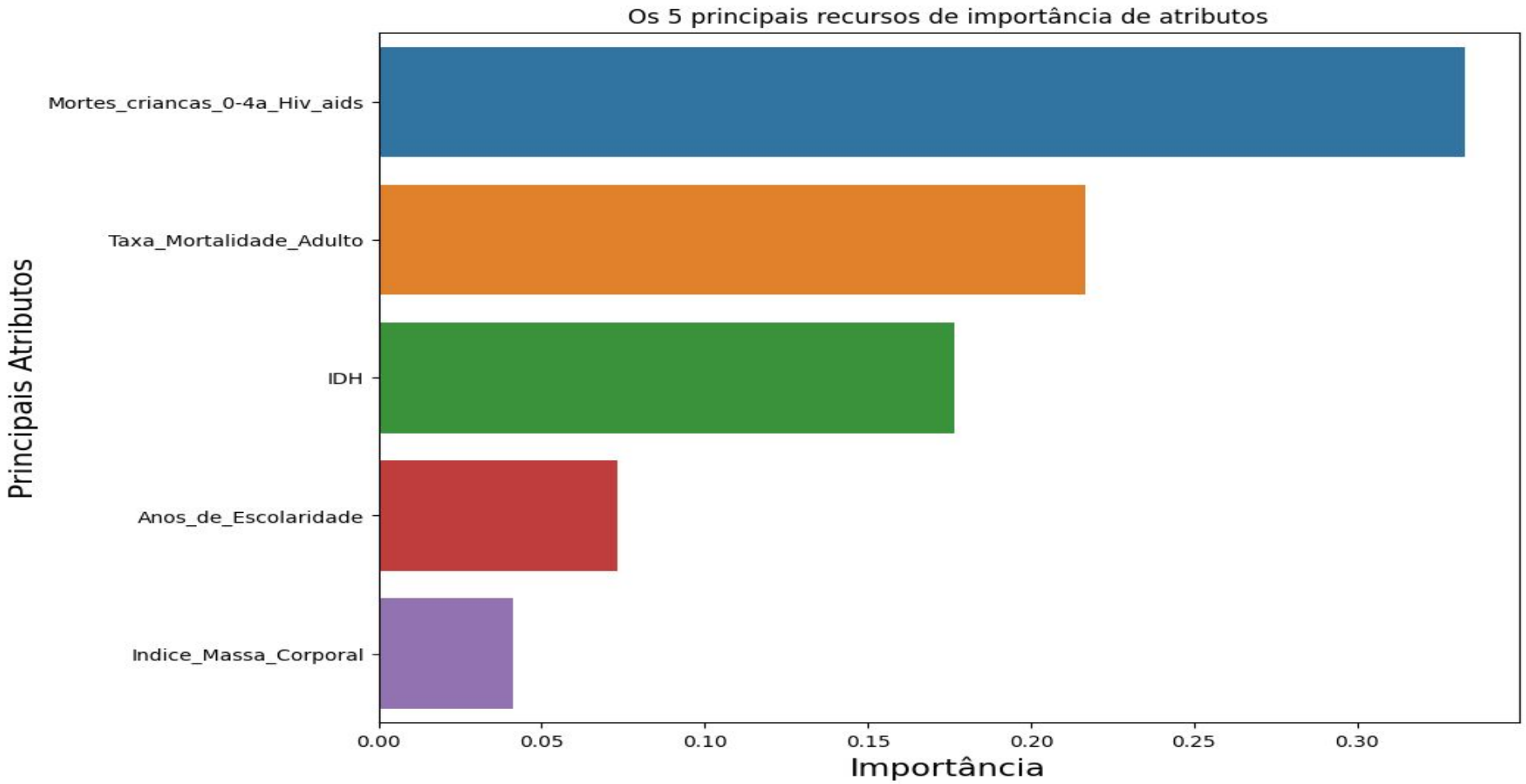
23. Países Desenvolvidos

Validação de Dados



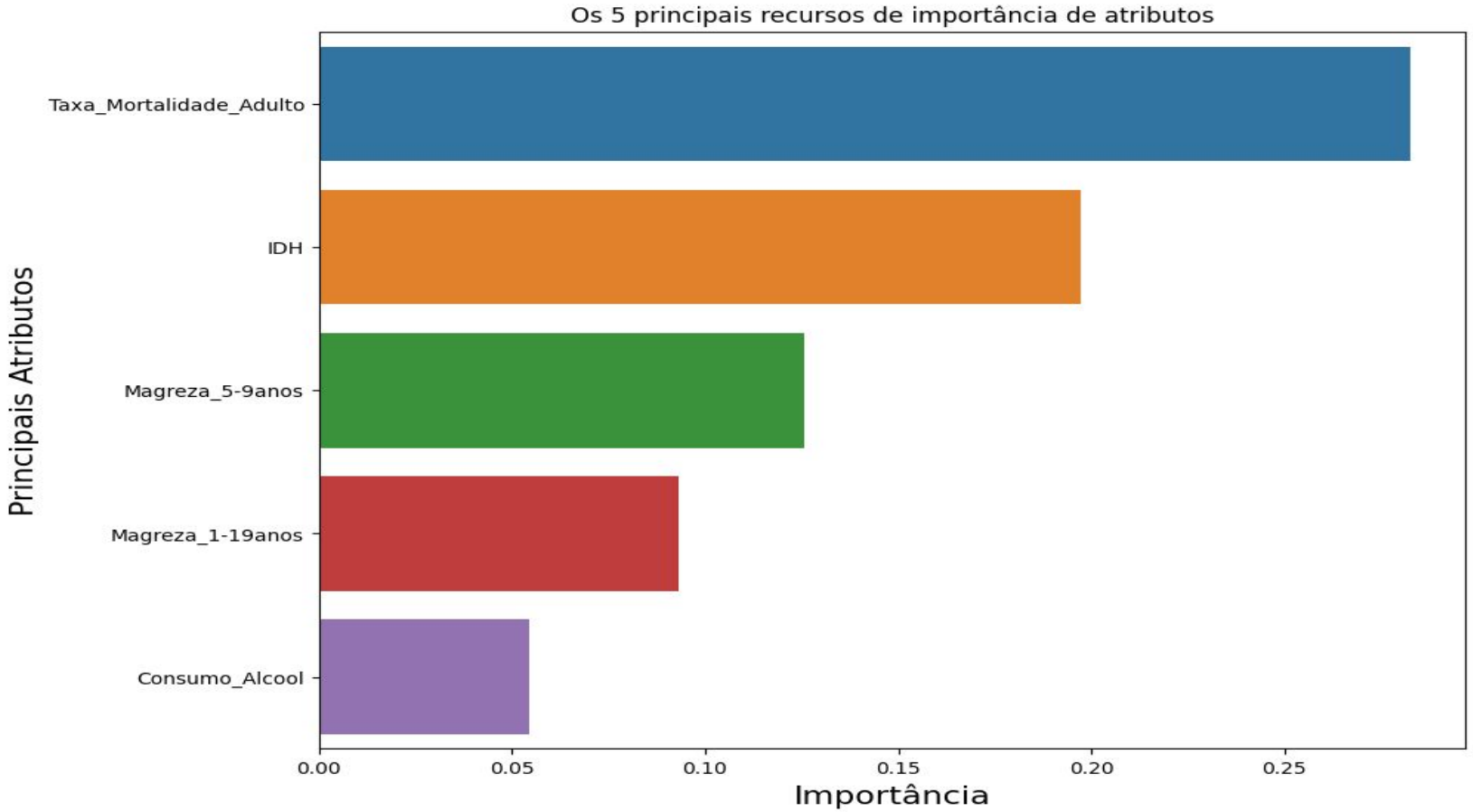
24. Ranking modelo de todos os países

Validação de Dados



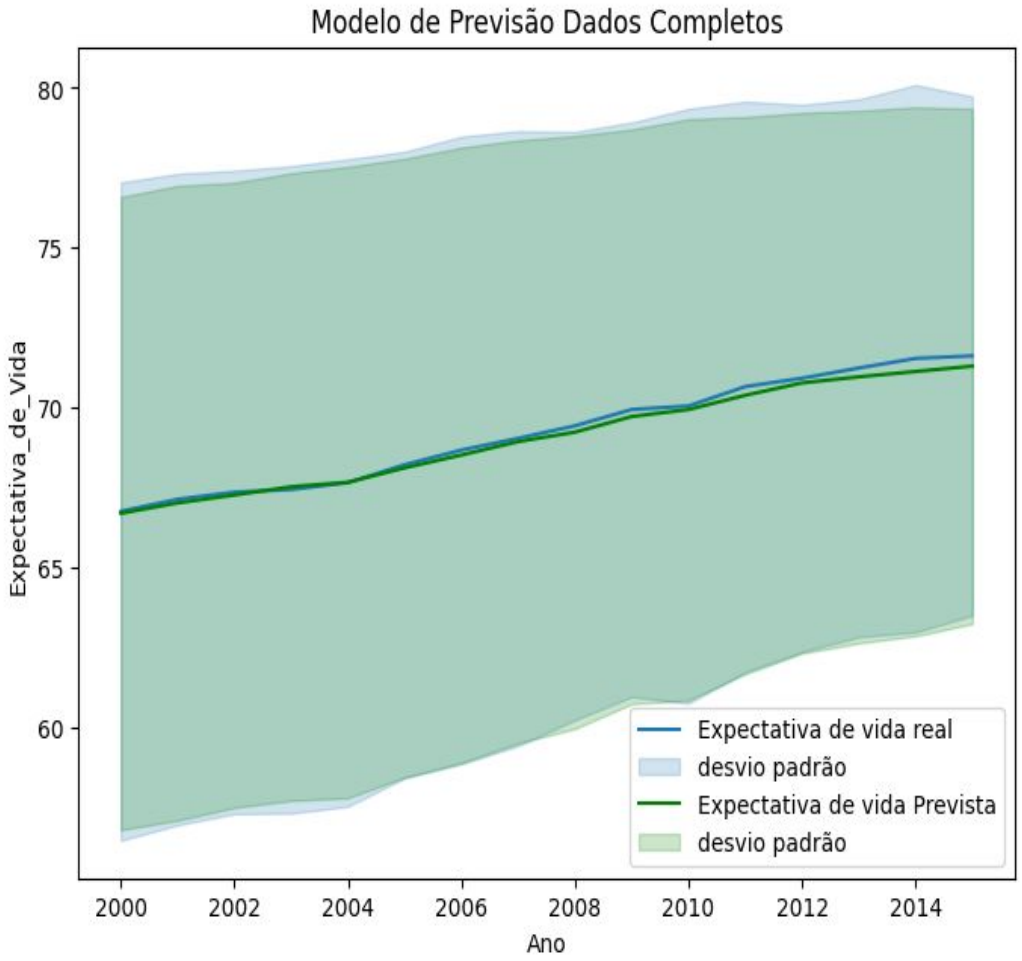
25. Ranking modelo países em desenvolvimento

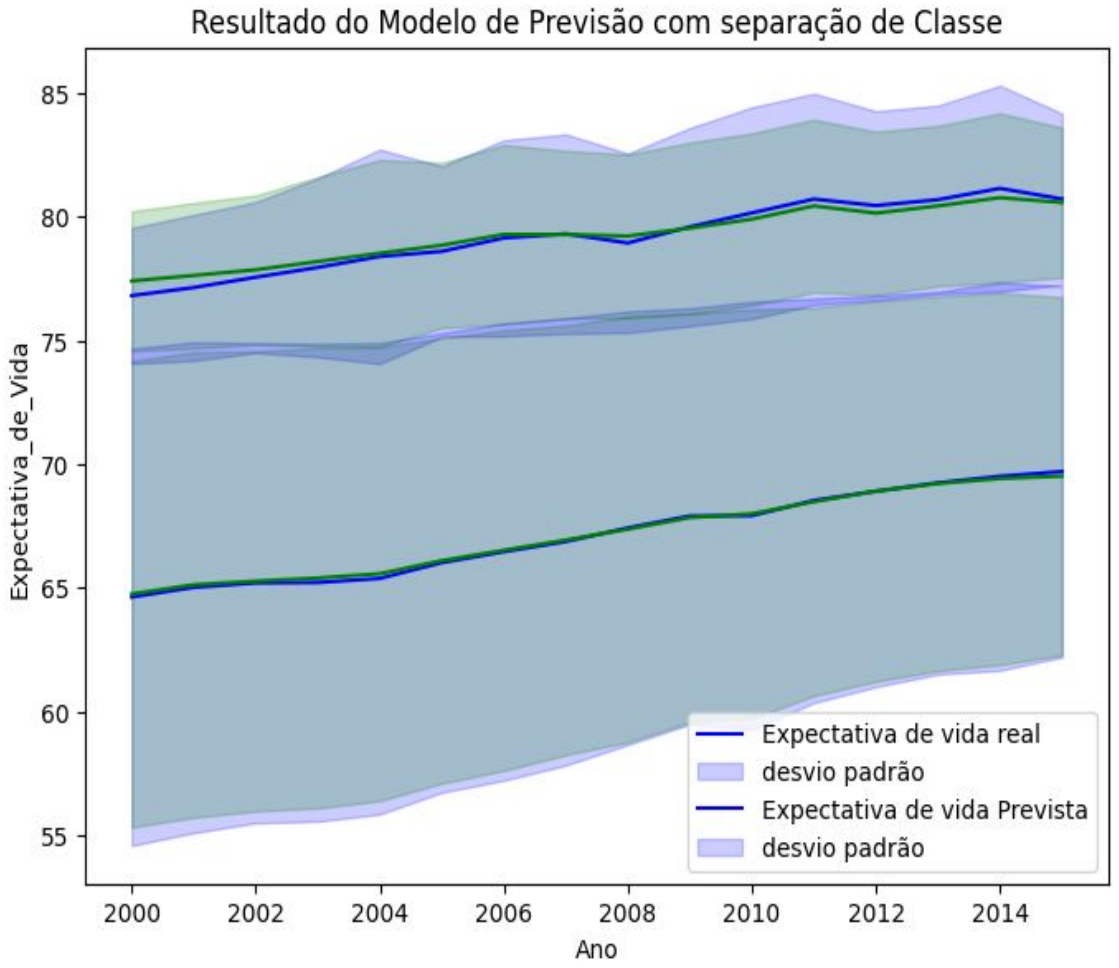
Validação de Dados



26. Ranking modelo países desenvolvidos

Validação de Dados





27. Gráfico de expectativa de vida real e previsto

Conclusão

- ✔ Claramente a quantidade de dados em cada conjunto de dados impactam na escolha de modelo.entretanto, os modelos de grupos de países separados teve uma amplitude de valores próximo de zero bem maior e uma melhor centralidade nas distribuições dos erros.
- ✔ Podemos sugerir que os países desenvolvidos devem focar os esforços em diminuir a taxa de mortalidade adulta e campanhas para diminuir a magreza de crianças 0-5 anos e jovens de 1-19 anos
- ✔ Países em desenvolvimento também deve ter preocupação com a taxa de mortalidade adulta, entretanto, o foco deve estar voltado para morte de crianças 0-4 anos com HIV/AIDS.
- ✔ Como esperado um alto nível de IDH é importante e os países precisam aumentar os investimentos na área para ter um ganho significativo na expectativa de vida.
- ✔ Random Forrest Regressor continua sendo um ótimo modelo de regressão, tendo resultados iniciais muito acima da média.

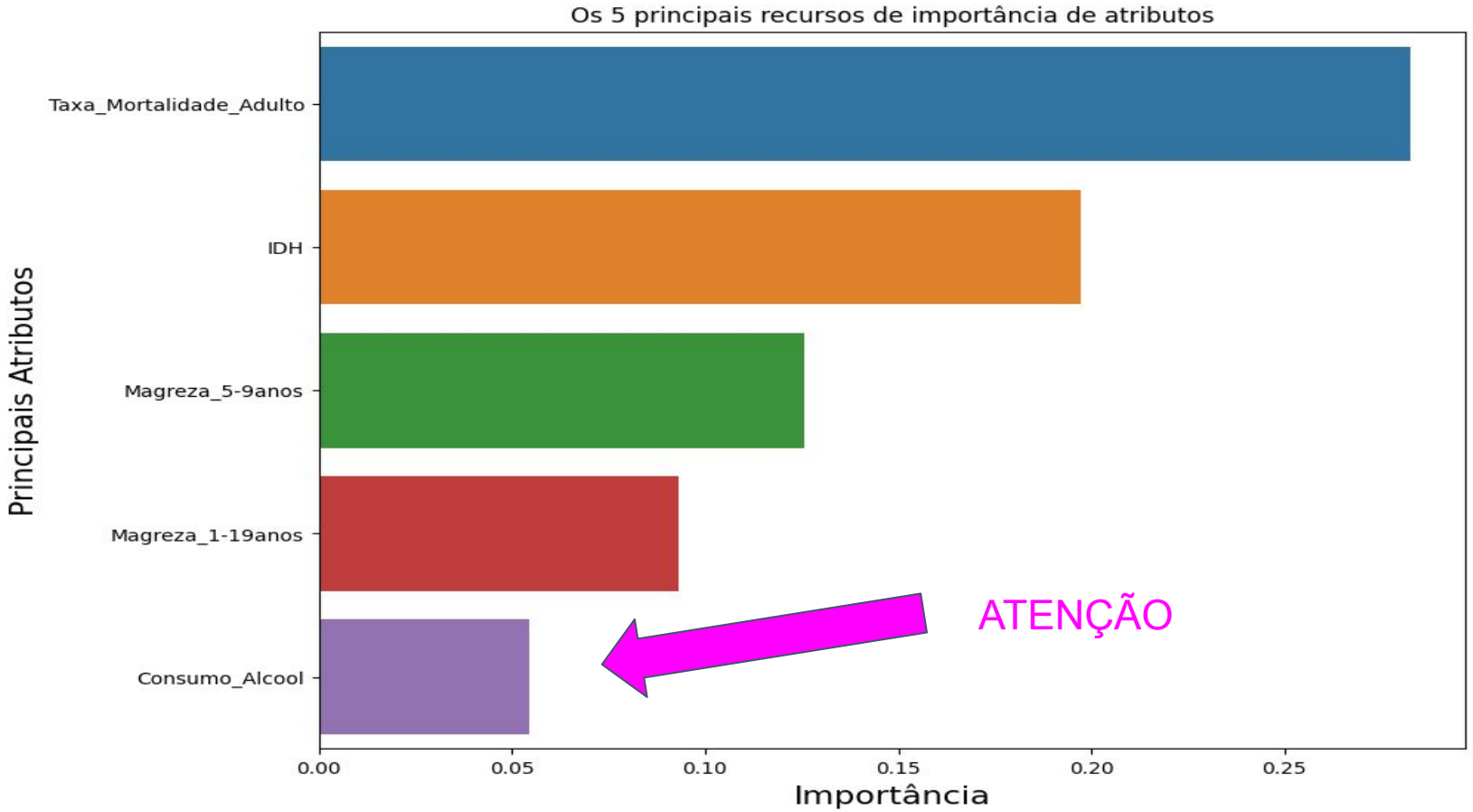
Curiosidades

A SEGUIR CENAS DOS PRÓXIMOS CAPÍTULOS…

- ✔ Coletar um banco de dados com número maior de anos ou com dados semestrais para aperfeiçoar o modelo para o grupo de países desenvolvidos.

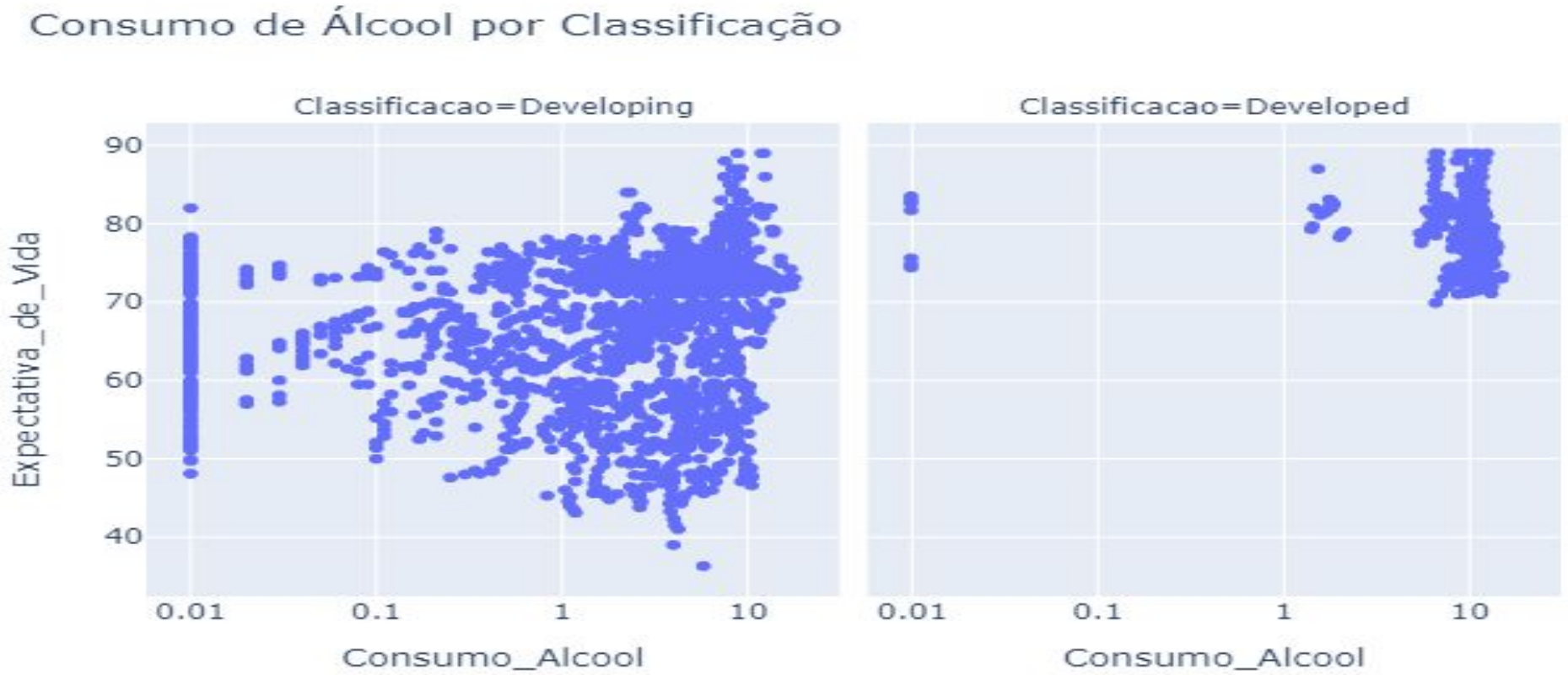
- ✔ Realizar um estudo focado exclusivamente para o Brasil e suas dificuldades no aumento da expectativa de vida.

Curiosidades



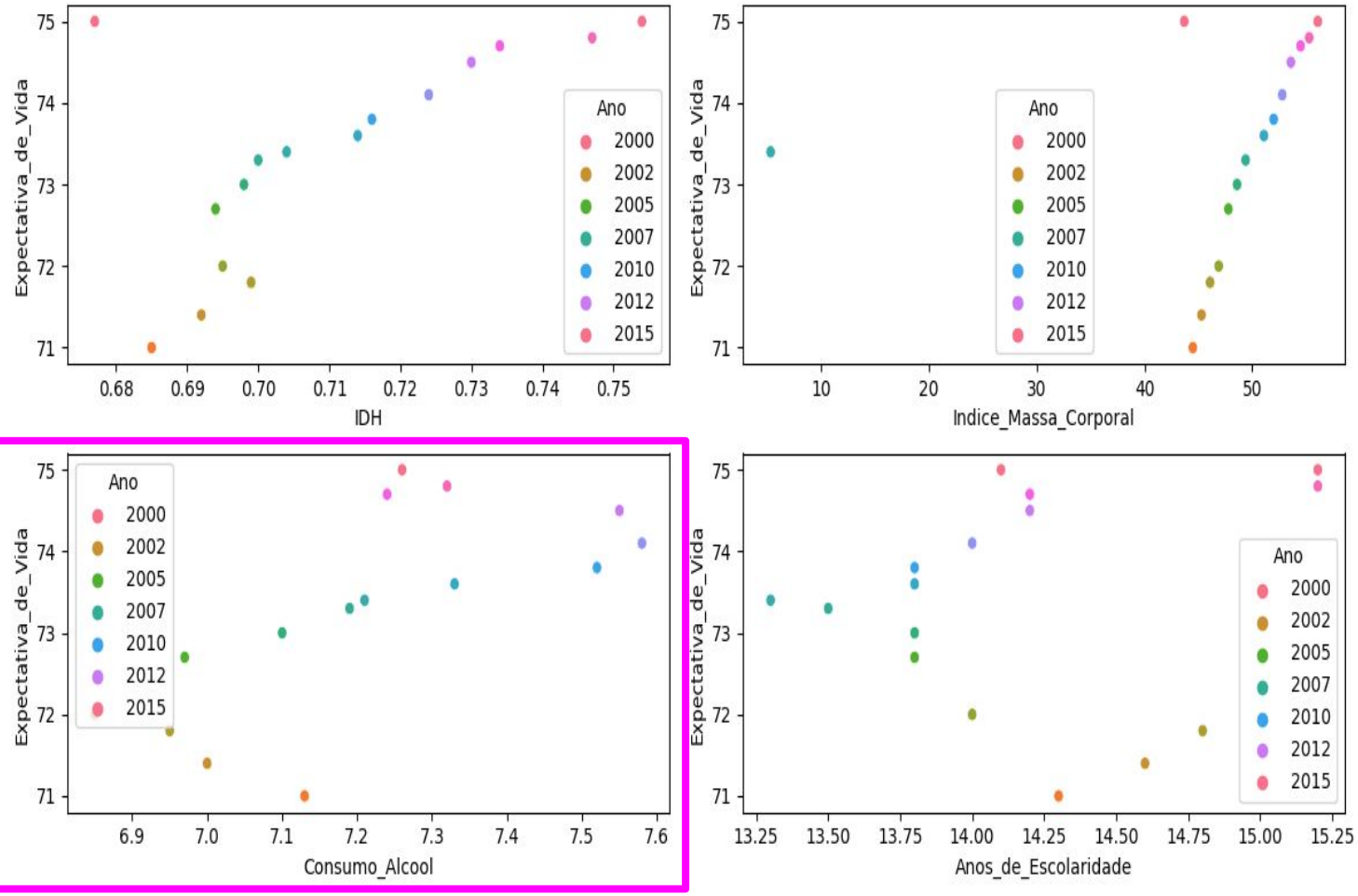
28. Atenção ao Ranking

Curiosidades



24. Análise consumo de álcool

Curiosidades



➔ Beba com Moderação

Apêndice

Material utilizado no projeto:

https://github.com/rochamarcio/Expectativa_de_vida

Agradecimento

"Agradeço a Deus e a minha família.

Há todos os professores e monitores pelos conselhos, dicas e ensinamentos em todo curso.

Aos colegas de classe pela ajuda e compreensão com esse eterno aprendiz.

A todos que trabalham nos bastidores para manter a plataforma operante. muito obrigado a todos."

- Marcio Carvalho

https://www.coursera.org/

https://cloud.ibm.com/

https://github.com/

https://www.infnet.edu.br/infnet/home/

https://www.google.com.br/

https://www.stackoverflow.com

- Machine Learning - Guia de Referência rápida - Trabalhando com dados estruturados em Python - novatec Matt Harrison

FIM