Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Departamento de Computação

Trabalho de Aprendizado de Máquina Supervisionado

Karys Cristina da Silva Barbosa - 811871
Pedro Marchi Nunes - 801053
Vitor Yuki Inumaru Ferreira - 794041

Maio 2025

Conteúdo

1 Introdução

Este trabalho tem como objetivo utilizar técnicas de aprendizado de máquina aprendidas em aula para analisar a expectativa de vida em diferentes países, a partir de um conjunto de dados disponível no Kaggle, que contempla informações do período de 2000 a 2015. Diferentemente de outros estudos disponíveis no Kaggle que utilizaram modelos de regressão linear múltipla baseados em dados de um único ano, neste trabalho optamos por deixar de lado os modelos de regressão e buscar a aplicação de técnicas de classificação, ampliando a abordagem para identificar categorias ou grupos relacionados à expectativa de vida.

A relevância do estudo está na análise conjunta de múltiplos fatores (imunização, mortalidade, econômicos, sociais e outros relacionados à saúde) que podem influenciar a expectativa de vida da população em cada país. Para isso, será aplicado um processo completo de pré-processamento, tratamento dos dados e visualização das características do dataset, com o intuito de compreender a necessidade e a aplicabilidade de cada técnica de aprendizado supervisionado.

O trabalho possui três etapas principais: justificativa da escolha do dataset, estudo e preparação dos dados, e aplicação dos métodos supervisionados. O uso dessas técnicas permitirá identificar os principais preditores que impactam a expectativa de vida, auxiliando na interpretação dos resultados e na visualização do conjunto de dados.

Acesse o notebook no Google Colab.:

2 Fundamentação Teórica

2.1 Aprendizado de Máquina e Pré-processamento de Dados

O aprendizado de máquina (machine learning) é uma área da inteligência artificial que desenvolve algoritmos capazes de aprender padrões a partir de dados, sem serem explicitamente programados para isso. Entre os tipos de aprendizado, o supervisionado é amplamente utilizado para problemas de classificação, onde o objetivo é atribuir uma categoria ou classe a cada exemplo com base em características observadas.

Antes de aplicar algoritmos de aprendizado, é fundamental realizar o pré-processamento dos dados, que inclui a limpeza, tratamento de valores ausentes, normalização e transformação das variáveis. O tratamento adequado dos dados é essencial para garantir a qualidade do modelo, pois dados incompletos, inconsistentes ou mal distribuídos podem prejudicar o desempenho do classificador.

A análise exploratória, por meio de técnicas como o cálculo da correlação entre variáveis e a visualização com histogramas, ajuda a entender as relações e distribuições dos dados. A correlação indica a força e o sentido da relação linear entre duas variáveis, o que pode orientar a seleção das características mais relevantes para o modelo, eliminando colunas que pouco influenciam. Os histogramas, por sua vez, permitem observar a distribuição das variáveis, identificando assim assimetrias, outliers e padrões que influenciam a modelagem.

A classificação, dentro do aprendizado supervisionado, consiste em construir um modelo que aprenda a partir de exemplos rotulados para prever a classe de novos dados desconhecidos.

2.2 Biblioteca Scikit-learn

A biblioteca `scikit-learn` é fundamental para o aprendizado de máquina em Python, oferecendo ferramentas para pré-processamento, modelagem e avaliação de dados. Neste trabalho, utilizamos o `KNNImputer` para tratar dados ausentes, o `StandardScaler` para normalizar os atributos e os codificadores `OrdinalEncoder` e `LabelEncoder` para transformar variáveis categóricas em numéricas. A divisão dos dados em conjuntos de treino e teste foi feita com o `train_test_split`, e a validação cruzada estratificada com o `StratifiedKFold`. Para classificação, aplicamos modelos como `DecisionTreeClassifier`, `RandomForestClassifier` e `MLPClassifier`, avaliados por métricas como acurácia e precisão, usando funções como `accuracy_score` e `classification_report`. Essas ferramentas permitiram um fluxo completo, desde o tratamento dos dados até a avaliação dos classificadores.

3 Metodologia

3.1 Base de Dados

Neste trabalho, Utilizaremos como material de estudo um dataset retirado da pataforma kaggle. O conjunto de dados, trata sobre expectativa de vida em diversos países, contendo informações com foco na expectativa de vida, fatores de imunização, sociais e fatores de saúde de 193 países por um período de 15 anos 2000 à 2015 - disponível através do link: Life Expectancy (WHO) - que são observadas em 22 variáveis. Os indicadores são:

- **Country**: Nome do país (193 países membros da OMS).
- **Year**: Ano da observação dos dados.
- **Status**: Classificação do país como *Developed* (desenvolvido) ou *Developing* (em desenvolvimento).
- **Life Expectancy**: Expectativa de vida ao nascer (em anos). Variável alvo do dataset.
- **Adult Mortality**: Taxa de mortalidade adulta (por 1000 habitantes) na faixa etária 15–60 anos.
- **Infant Deaths**: Número de mortes de crianças menores de 1 ano por 1000 nascidos vivos.
- **Under-five deaths**: Mortalidade de crianças menores de 5 anos (por 1000 nascidos vivos).
- **Alcohol**: Consumo per capita de álcool (litros puros/ano) em adultos $\geq$ 15 anos.
- **Percentage Expenditure**: Gastos totais em saúde per capita (em USD).
- **Hepatitis B**: Cobertura vacinal contra Hepatite B (%) em crianças de 1 ano.
- **Measles**: Número de casos reportados de sarampo por 1000 habitantes.
- **BMI**: Índice de Massa Corporal (IMC) médio da população adulta.
- **Polio**: Cobertura vacinal contra poliomielite (%) em crianças de 1 ano.
- **Total Expenditure**: Porcentagem do PIB gasta em saúde.
- **Diphtheria**: Cobertura vacinal contra difteria (%) em crianças de 1 ano.
- **HIV/AIDS**: Taxa de mortalidade por HIV/AIDS (por 1000 habitantes).
- **GDP**: PIB per capita (em USD ajustado por inflação).
- **Population**: População estimada do país no ano observado.
- **Thinness 1-19 years**: Prevalência de magreza em crianças/adolescentes de 1–19 anos (%).
- **Thinness 5-9 years**: Prevalência de magreza em crianças de 5–9 anos (%).

- **Income composition of resources**: Índice de renda dos recursos (combina educação, renda e saúde).
- **Schooling**: Número médio de anos de escolaridade da população adulta.

Outras informações interesssante são as fornecidas pelas funções info(). Como é possível perceber na Figura 1, existem 2938 exemplos no dataset. Além disso, apenas as colunas "Country"e "Status"não são valores numéricos.

```
Data columns (total 22 columns):
 #   Column                           Non-Null Count  Dtype
---  ------                           --------------  -----
 0   Country                          2938 non-null   object
 1   Year                             2938 non-null   int64
 2   Status                           2938 non-null   object
 3   Life expectancy                  2928 non-null   float64
 4   Adult Mortality                  2928 non-null   float64
 5   infant deaths                    2938 non-null   int64
 6   Alcohol                          2744 non-null   float64
 7   percentage expenditure           2938 non-null   float64
 8   Hepatitis B                      2385 non-null   float64
 9   Measles                          2938 non-null   int64
 10   BMI                             2904 non-null   float64
 11  under-five deaths                2938 non-null   int64
 12  Polio                            2919 non-null   float64
 13  Total expenditure                2712 non-null   float64
 14  Diphtheria                       2919 non-null   float64
 15   HIV/AIDS                        2938 non-null   float64
 16  GDP                              2490 non-null   float64
 17  Population                       2286 non-null   float64
 18   thinness  1-19 years            2904 non-null   float64
 19   thinness 5-9 years             2904 non-null   float64
 20  Income composition of resources  2771 non-null   float64
 21  Schooling                        2775 non-null   float64
```

Figura 1: Informações sobre o dataset, fornecidas pelo .info()

A partir das informações fornecidas pela função describe(), é possível identificar a presença de valores faltantes e determinar quais colunas concentram a maior quantidade desses dados ausentes. Isso evidencia a necessidade de um pré-processamento para garantir a consistência do dataset antes da aplicação de métodos de aprendizado de máquina. Além disso, embora não tenham sido encontrados dados evidentemente incorretos, observou-se que alguns países apresentaram valores anormalmente altos para determinadas doenças em comparação com os demais.

3.2 Pré-processamento dos Dados

Inicialmente, visualizou-se a matriz de dispersão dos atributos em relação a expectativa de vida, tentando encontrar padrões visuais.

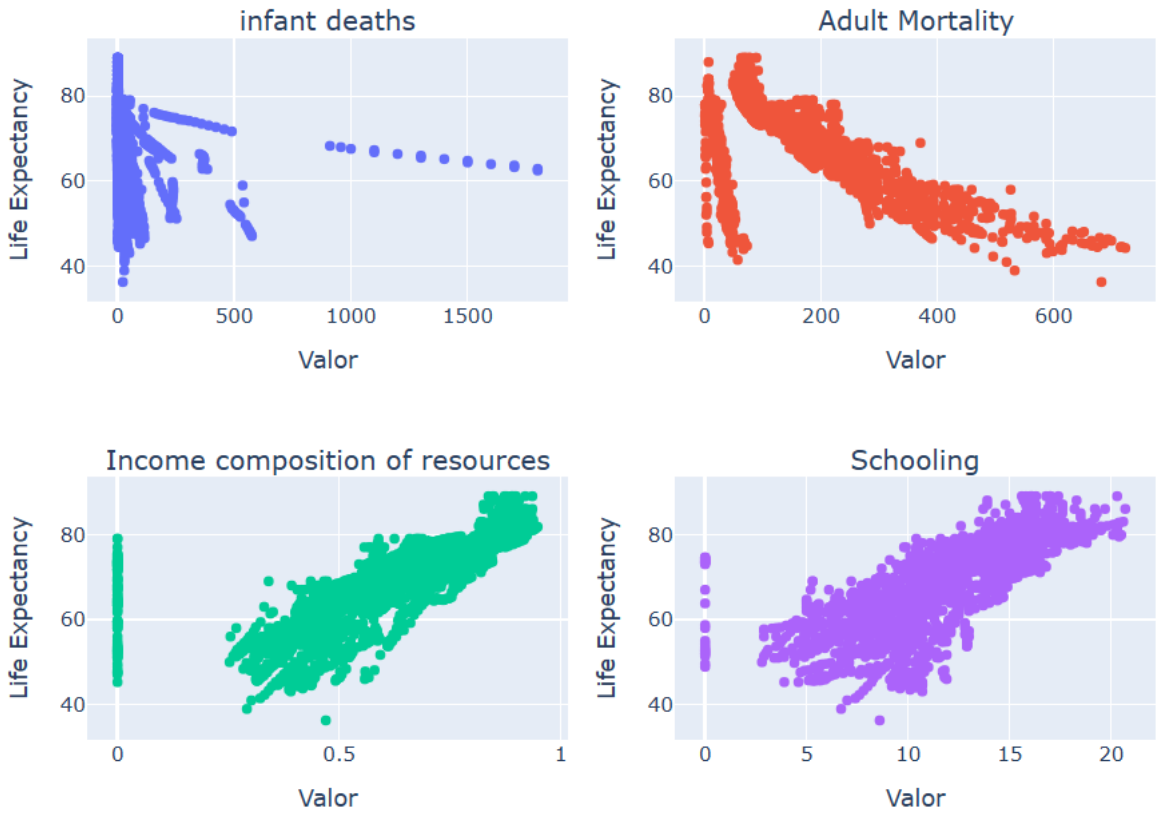

Figura 2: Matriz de Dispersão para 4 atributos

Observa-se uma tendência clara de aumento da expectativa de vida à medida que crescem o índice de composição de renda e o nível de escolaridade. Em contrapartida, a mortalidade adulta apresenta uma relação inversa, diminuindo conforme a expectativa de vida aumenta. Já a mortalidade infantil mostrou um comportamento atípico em relação aos demais dados, o que pode indicar a necessidade de um tratamento desses valores.

Como etapa de pré-processamento do conjunto de dados, realizou-se a identificação e tratamento de valores faltantes. Para imputação desses dados, aplicou-se o algoritmo KNN Imputer para variáveis numéricas, garantindo assim uma aproximação mais fidedigna dos valores originais, sendo necessário normalizar os valores através da função StandardScaler(). Ademais, a normalização é de extrema importância para evitar que certos parâmetros atrapalhem o funcionamento do algoritmo de classificação.

Em seguida, analisou-se a matriz de correlação, presente na Figura 3, com o objetivo de compreender a influência de cada variável na expectativa de vida, bem como identificar possíveis variáveis redundantes que poderiam ser agrupadas.

A partir da análise da matriz de correlação, observa-se que muitos atributos apresentam baixa correlação com a variável Life Expectancy. Por esse motivo, optou-se por remover esses atributos durante o processo de tratamento dos dados. Além disso, foi identificada uma correlação máxima entre as colunas

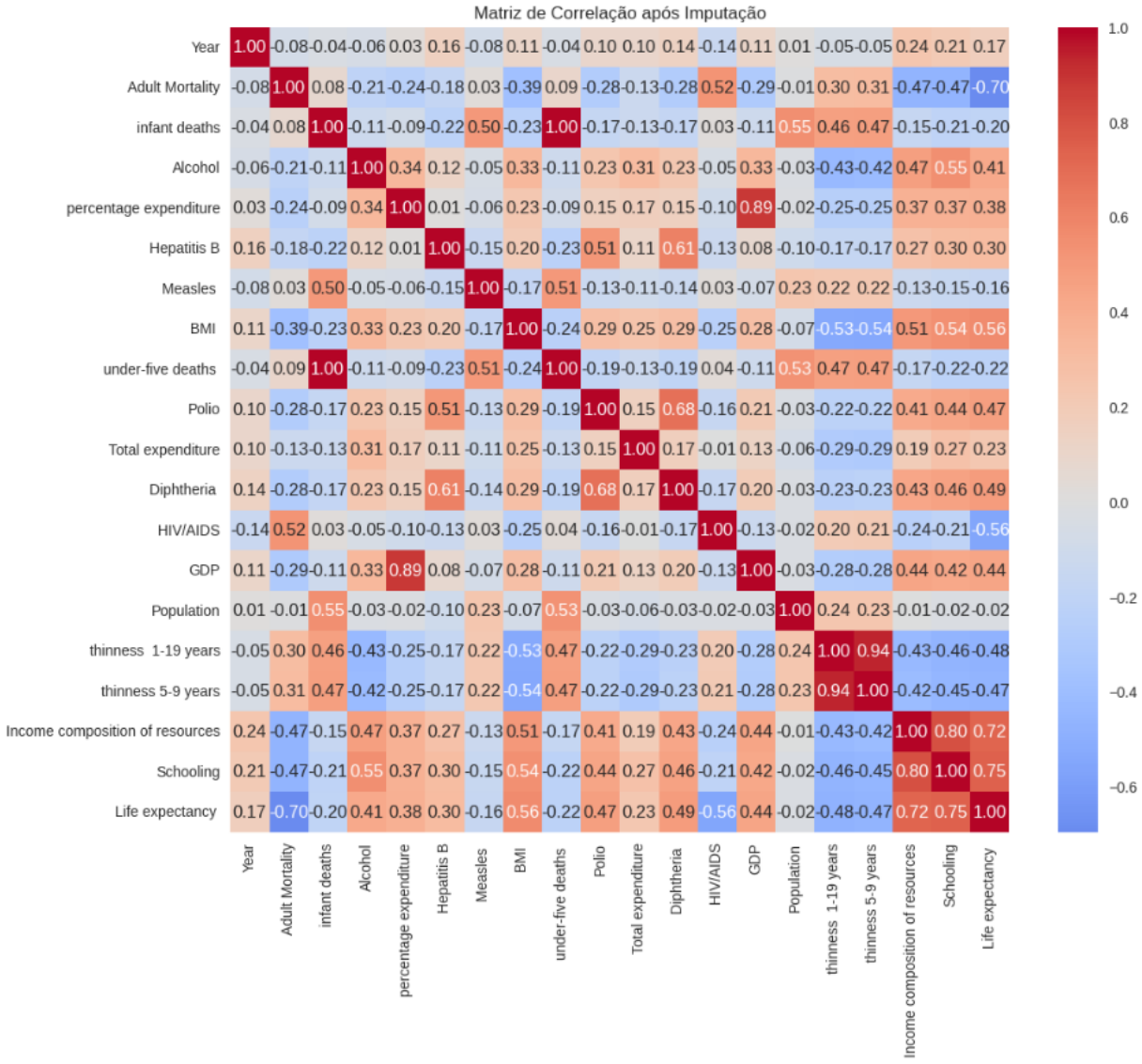

Figura 3: Matriz de Correlação Gerada após a inserção dos valores

infant deaths e under-five deaths. No entanto, essa redundância não foi tratada separadamente, pois ambas as variáveis foram eliminadas por não exercerem influência significativa sobre a expectativa de vida.

Como resultado, foram escolhidos os seguintes atributos: Adult Mortality, Alcohol, percentage expenditure, Hepatitis B, BMI, Polio, Diphtheria, HIV/AIDS, GDP, thinness 1-19 years, thinness 5-9 years, Income composition of resources e Schooling. Esses atributos foram considerados os mais relevantes para representar o comportamento da variável alvo e, consequentemente, foram utilizados nos testes com os modelos de classificação.

Outro gráfico que se mostrou de grande relevância na análise apresentada neste relatório foi o histograma da expectativa de vida. Através dele, é possível observar uma forte concentração de dados em

determinados intervalos, evidenciando padrões importantes na distribuição dos valores.

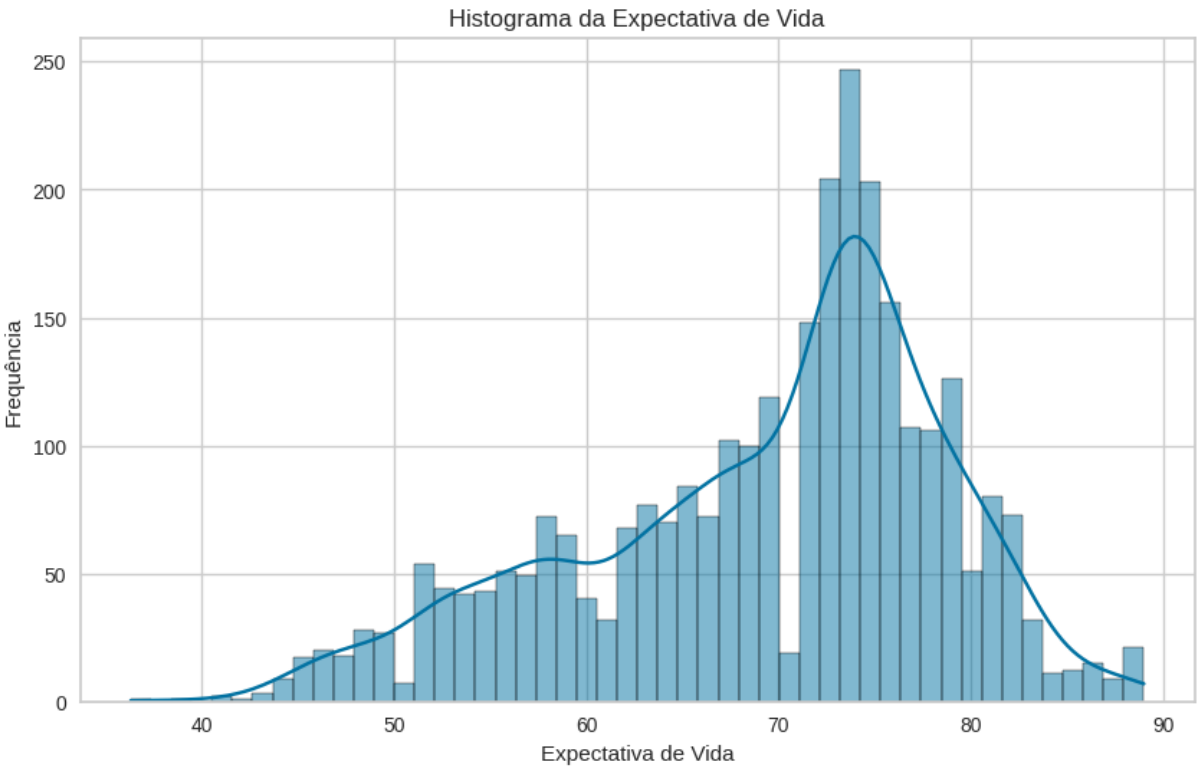

Figura 4: Histograma da Expectativa de Vida

Considerando que o dataset foi originalmente projetado para tarefas de regressão linear e não possuía classes pré-definidas, optou-se por criar uma variável categórica a partir do atributo principal "Life Expectancy". Para determinar o número ideal de grupos, foram utilizados dois métodos: o método do cotovelo, que analisa a variação da inércia com o aumento de clusters, e o índice de silhueta, que avalia a coesão e separação entre os grupos formados. Na figura abaixo é possível analisar os dados.

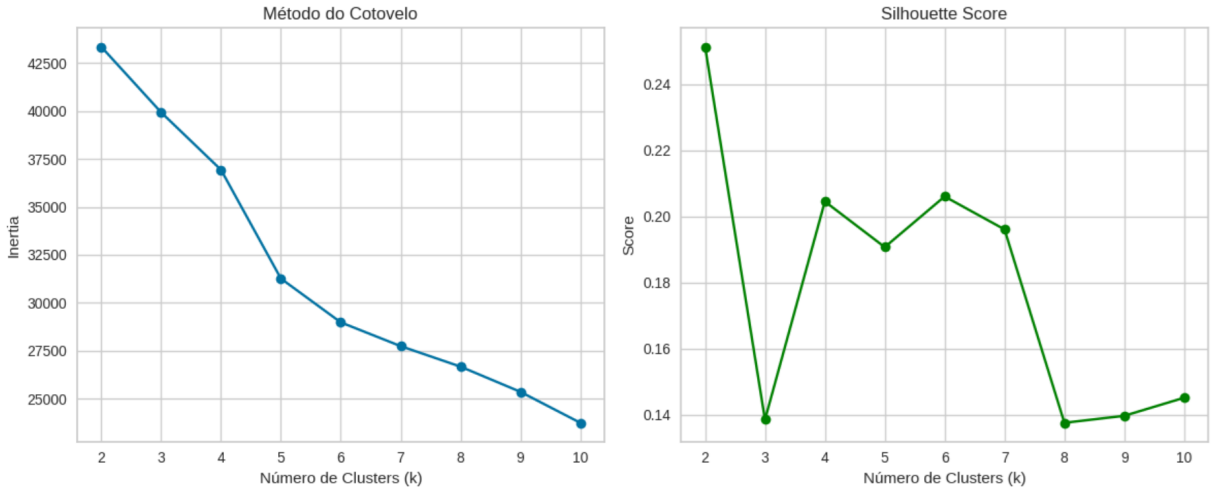

Figura 5: Método do Cotovelo e Silhouette Score

Uma análise rápida da Figura 5 indica que os valores 4, 5 ou 6 parecem ser boas opções para o número de classes. No entanto, ficou evidente que o índice de silhueta permanece baixo, independentemente do valor escolhido, indicando uma possível dificuldade de aplicar classes neste dataset.

Com base nas informações obtidas a partir do histograma, do método do cotovelo e do índice de silhueta, foi selecionado o valor 4 para o número de classes. Os intervalos dessas classes foram definidos utilizando a função qcut() do Pandas, aliados a uma análise visual do histograma. Dessa forma, foi possível estabelecer quatro classes distintas de expectativa de vida, segmentadas em faixas etárias estratificadas, conforme descrito a seguir:

- Ótima (80 à 90 anos)
- Boa (72 à 80 anos)
- Regular (60 à 72 anos)
- Ruim (36 à 60 anos)

Obteve-se, então, o gráfico de barras que mostra a quantidade de elementos em cada classe de expectativa de vida. A distribuição dos dados é a seguinte: a classe Boa contém 1.213 elementos, a classe Regular possui 878 elementos, a classe Ruim reúne 594 elementos e a classe Ótima apresenta 253 elementos.

A Figura 6 ilustra essa distribuição por meio do gráfico de barras correspondente.

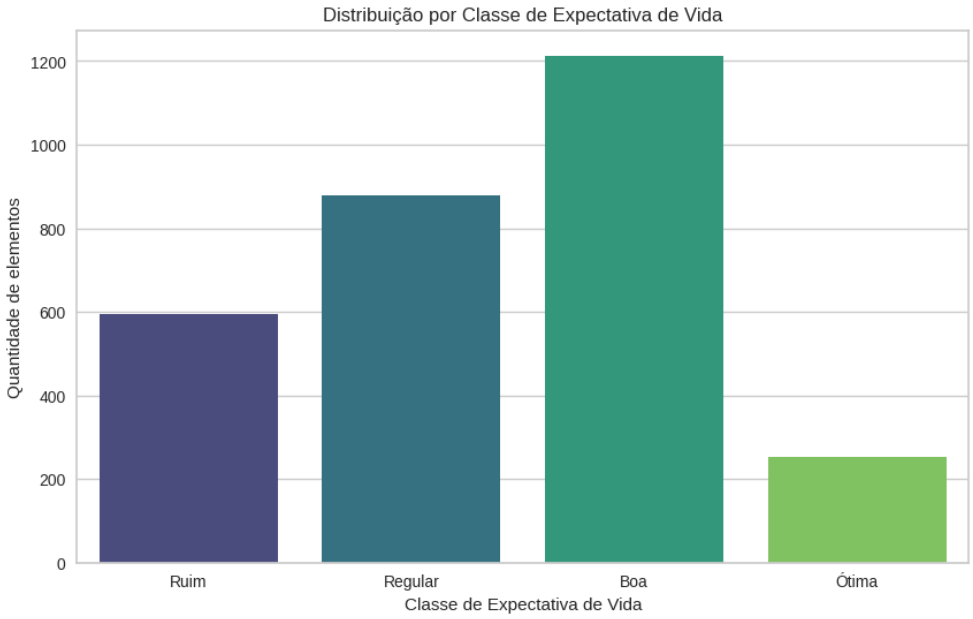

Figura 6: Distribuição por Classe de Expectativa de Vida

Além disso, utilizou-se um PCA 2D para verificar em um gráfico bidimensional as novas classes criadas. Nele, evidencia-se como as classes estão sobrepostas, um problema que será tratado durante os resultados.

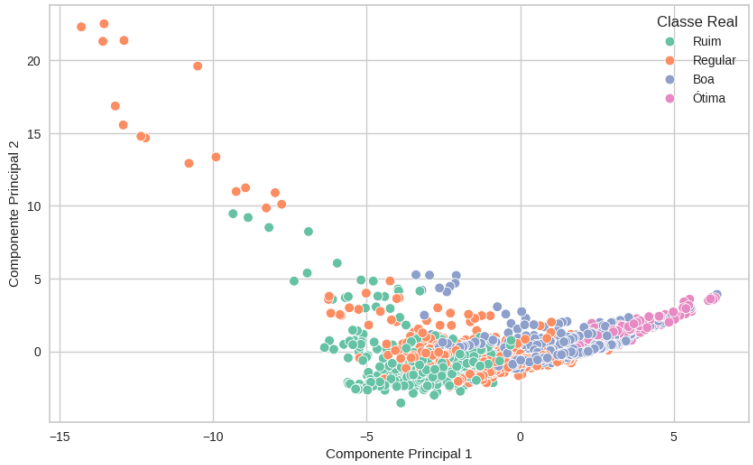

Figura 7: Visualização das Classes Reais com PCA (2D)

A partição dos dados foi realizada mediante a aplicação aninhada de algumas técnicas vistas em aula:

- Holdout - Divisão de 80% dos dados para treinamento e 20% para teste.
- Estratificação - Preservação da distribuição original das classes em ambos os subconjuntos de treino e teste.

Dessa forma, este procedimento garante que haja boa representatividade estatística dos conjuntos em

fase de treino e teste, equivalência nas proporções categóricas e minimização de vieses na avaliação do modelo.

3.3 Seleção de Modelos de Classificação

Foram testados diversos algoritmos supervisionados para a tarefa de classificação, incluindo:

- Árvore de Decisão (Decision Tree), além do uso comparativo da Floresta Aleatória (Random Forest).
- Rede Neural, implementada por meio do classificador MLPClassifier.

Para cada modelo, foram realizados múltiplos testes visando ajustar parâmetros importantes. No caso da rede neural, por exemplo, foram avaliadas diferentes configurações do número de neurônios nas camadas ocultas. Já na Árvore de Decisão, os testes focaram nas principais colunas para otimizar o desempenho e evitar overfitting.

3.3.1 Árvore de Decisão

O algoritmo de Árvore de Decisão (Decision Tree) foi implementado utilizando a biblioteca scikit-learn com o objetivo de classificar países nas 4 categorias de expectativa de vida criadas ("Ruim", "Regular", "Boa", "Ótima") com base em indicadores socioeconômicos e de saúde. O algoritmo foi testado com os atributos reduzidos e com apenas os 5 melhores atributos identificados e selecionados, tentando comparar a diferença de desempenho. Além disso, foi feita uma comparação com o algoritmo de randomforest, com o objetivo de visualizar a diferença nas medidas de desempenho. O algoritmo foi instanciado por meio da função DecisionTreeClassifier(), com os seguintes parâmetros-chave:

- Critério de Divisão: Entropia, que mede a impureza dos nós. A entropia é calculada como:

$$\text{Entropia}(S) = -\sum_{i=1}^{c} p_i \log_2 p_i \tag{1}$$

  onde é a proporção de amostras da classe i no nó. O modelo busca maximizar o ganho de informação (redução de entropia) em cada divisão.

- Profundidade Máxima: Não limitada, permitindo que a árvore cresça até que todas as folhas sejam puras ou contenham o número mínimo de amostras.

3.3.2 Rede Neural

O algoritmo de Rede Neural Artificial (Multilayer Perceptron - MLP) foi implementado utilizando a biblioteca scikit-learn com o mesmo objetivo de classificar os países nas mesmas 4 categorias de expectativa de vida ("Ruim", "Regular", "Boa", "Ótima"). A rede neural foi instanciada através da função MLPClassifier(), com os seguintes parâmetros-chave:

- Arquitetura da Rede: Composta por Camadas Ocultas (hidden_layer_sizes=(20, 20), ou seja, 2 camadas com 20 e 20 neurônios respectivamente), Função de Ativação (ReLU (activation='relu'), para introduzir não-linearidade ao modelo) e Solver (Adam (solver='adam'), um otimizador adaptativo para eficiência em datasets com tamanhos grandes).

- Treinamento: Que possui Taxa de Aprendizado (Inicial learning_rate_init=0.001 (default), com ajuste adaptativo), Máximo de Iterações (max_iter=500 para garantir convergência) e Random State: random_state=0 para reprodutibilidade)

Crtério de treino: Para problemas de classificação, o MLPClassifier usa automaticamente a Entropia Cruzada Logística (Log Loss ou Cross-Entropy Loss), que mede a divergência entre as probabilidades previstas e as classes reais, dada pela função para multiclasse:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{i,j}\log(p_{i,j}(\theta)) \tag{2}$$

Em outras palaras, a rede neural ajusta seus parâmetros θ para minimizar a função de perda $\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{i,j}\log(p_{i,j})$, que mede a diferença entre as previsões ($p_{i,j}$) e os valores reais ($y_{i,j}$): quando $p_{i,j} \approx y_{i,j}$, $\mathcal{L}(\theta)$ diminui; quando divergem, $\mathcal{L}(\theta)$ aumenta.

Um teste adicional realizado envolveu a experimentação com diferentes configurações de neurônios na arquitetura da rede neural. As combinações testadas variaram desde redes mais simples, com apenas uma camada oculta contendo 10, 20 ou 30 neurônios, até arquiteturas mais complexas, com múltiplas camadas, como (20, 20), (30, 15), (50, 20) e (50, 30, 20). O objetivo foi observar como o aumento da profundidade e da quantidade de neurônios impactava o desempenho do modelo em termos de acurácia e outras métricas.

4 Resultados e Discussão

4.1 Árvore de Decisão

Foram obtidas as seguintes métricas de desempenho para o modelo de Árvore de Decisão (DecisionTree) utilizando a biblioteca scikit-learn:

	precision	recall	f1-score	support
Boa	0.85	0.89	0.87	243
Regular	0.81	0.78	0.79	176
Ruim	0.90	0.94	0.92	119
Ótima	0.98	0.80	0.88	50
accuracy			0.86	588
macro avg	0.88	0.85	0.87	588
weighted avg	0.86	0.86	0.86	588

Figura 8: Medidas de Desempenho da Árvore de Decisão

Um fenômeno observado foi a baixa precisão na classificação das duas classes mais representativas, enquanto as classes com menor quantidade de exemplos apresentaram desempenho significativamente superior nessa medida. Isto pode ser explicado pela tendência do modelo em classificar nas classes com maior quantidade de elementos. A medida do recall demonstra uma dificuldade do modelo em indentificar os verdadeiros positivos da classe Regular e Ótima, ou seja, apesar da alta acurácia do modelo na ótima, muitos elementos acabam passando despercebidos.

Também foi feito a matriz de confusão a fim de ajudar a entender esse comportamento:

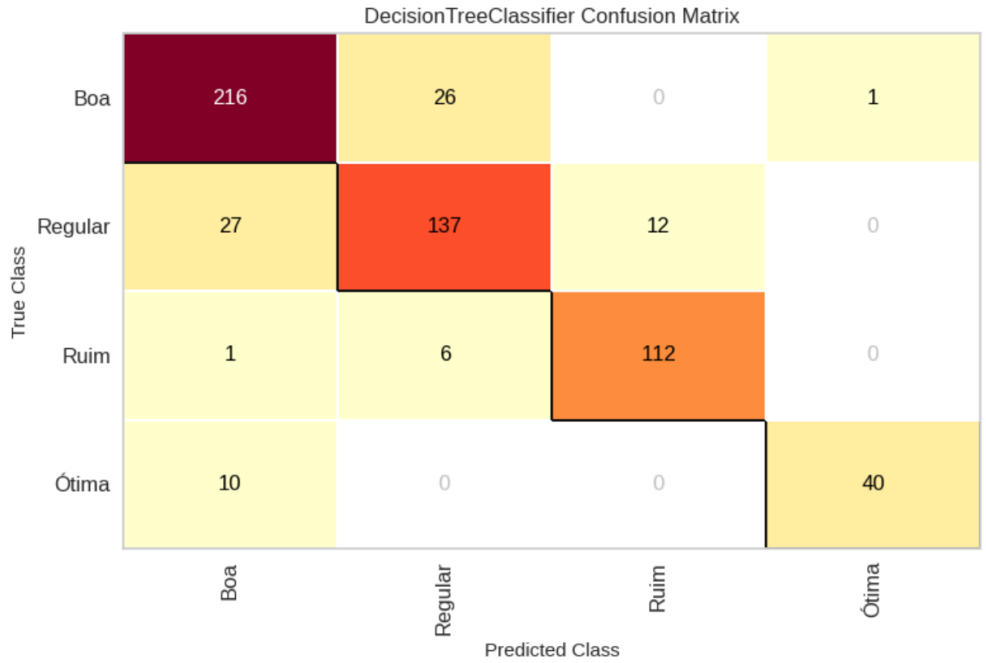

Figura 9: Matriz de confusão da árvore de decisão

Nesta figura fica ainda mais evidente a maneira com a qual o modelo se comporta, confundindo geralmente as classes Boa e Regular, visto que são as que possuem maior quantidade e são próximas. Além de deixar de identificar elementos como Ótimos, evidenciado pelo recall anterior.

Ademais, foram feitos a árvore e o 2D PCA, para compreender mais a fundo o comportamento do modelo.

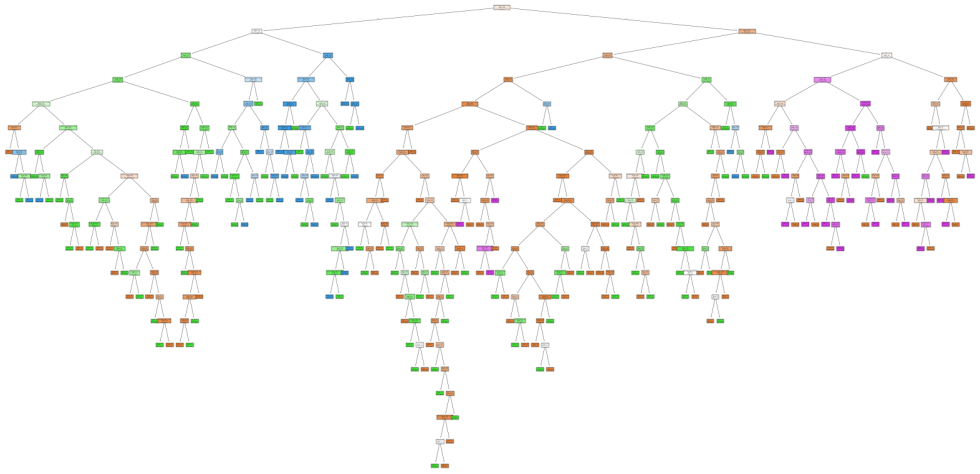

Figura 10: Árvore de decisão gerada

A árvore apresentou uma profundida de 19 e um total de 393 nós, sendo 197 folhas.

A matriz de confusão mostrou que a árvore de decisão teve maior dificuldade em distinguir entre as classes "Boa"e "Regular", além de algumas confusões com a classe "Ótima". Essa dificuldade é justificada visualmente pelo gráfico PCA 2D com cores das classes reais, que revela que as amostras dessas categorias se distribuem em regiões próximas ou até sobrepostas no espaço de maior variância dos dados. Isso indica que, mesmo com um classificador adequado, os próprios atributos numéricos disponíveis não oferecem separação clara entre essas categorias, o que limita a performance do modelo.

A seguir, são apresentados os resultados obtidos com o modelo de Árvore de Decisão utilizando apenas as cinco variáveis com maior correlação com a expectativa de vida: Adult Mortality, BMI, HIV/AIDS, Income composition of resources e Schooling.

	precision	recall	f1-score	support
Boa	0.85	0.89	0.87	243
Regular	0.81	0.78	0.79	176
Ruim	0.90	0.94	0.92	119
Ótima	0.98	0.80	0.88	50
accuracy			0.86	588
macro avg	0.88	0.85	0.87	588
weighted avg	0.86	0.86	0.86	588

Figura 11: Resultado com Apenas 5 Atributos

É possível observar que, mesmo com a redução da quantidade de variáveis utilizadas pelo modelo, a acurácia manteve-se praticamente inalterada, assim como as outras métricas de desempenho. Esse resultado indica que os demais atributos exercem pouca influência sobre o desempenho da Árvore de Decisão para este conjunto de dados.

Por fim, apresentam-se os resultados obtidos com o uso do RandomForestClassifier, aplicado com o objetivo de minimizar o possível overfitting observado no modelo anterior.

	precision	recall	f1-score	support
Boa	0.91	0.94	0.93	243
Regular	0.90	0.85	0.87	176
Ruim	0.94	0.96	0.95	119
Ótima	0.96	0.94	0.95	50
accuracy			0.92	588
macro avg	0.93	0.92	0.92	588
weighted avg	0.92	0.92	0.92	588

Figura 12: Resultado com RandomForestClassifier

Seguindo o estudado em aula, o RandomForestClassifier apresentou melhores medidas de desempenho de maneira geral do que a Decision Tree evidenciando que usar várias árvores juntas ajuda o modelo a funcionar melhor com dados novos. Isso acontece porque a Random Forest evita que o modelo fique muito preso aos detalhes do conjunto de treino, tornando as previsões mais estáveis e confiáveis. Além disso, combinando várias árvores, o modelo consegue entender melhor as relações entre as variáveis, o que melhora o resultado final.

4.2 Rede Neural

Após aplicar uma Rede Neural de duas camadas ocultas de 20 neurônios cada, obtevê-se o seguinte resultado:

	precision	recall	f1-score	support
Boa	0.83	0.87	0.85	243
Regular	0.82	0.78	0.80	176
Ruim	0.93	0.93	0.93	119
Ótima	0.83	0.76	0.79	50
accuracy			0.85	588
macro avg	0.85	0.84	0.84	588
weighted avg	0.85	0.85	0.85	588

Figura 13: Resultado com Rede Neural

Assim como observado no modelo de Árvore de Decisão, a rede neural teve dificuldade em identificar corretamente os verdadeiros positivos da classe Regular. Além disso, apresentou baixa precisão na classe Ótima, uma piora em comparação do modelo da Árvore de Decisão. Nas demais métricas, o desempenho foi semelhante ao anterior, embora de forma geral tenha sido um pouco inferior.

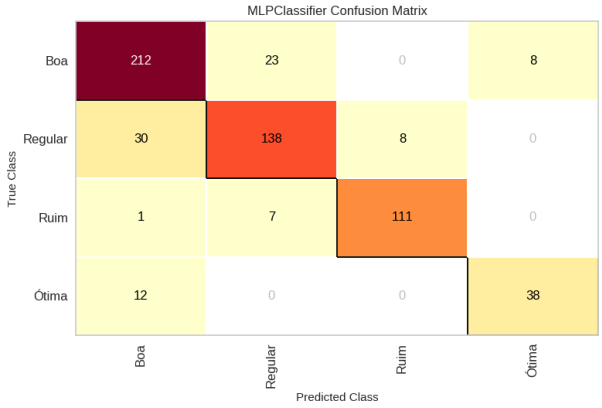

Figura 14: Resultado com Rede Neural

Na matriz de confusão é possível visualizar o mesmo problema da classe Regular, principalmente em relação à classe Boa. Além disso a classe Boa passou a ser confundida com a Ótima, fato responsável pela diminuição da precisão

Em sequida está a tabela valores comparados com diferentes configurações de redes neurais.

Tabela 1: Comparação entre diferentes configurações do MLPClassifier

Configuração (hidden layers)	**Épocas**	**Acurácia**	**Precisão (média)**	**Recall (média)**
(10,)	197	0.7993	0.80	0.78
(20,)	221	0.8231	0.83	0.79
(30,)	179	0.8163	0.82	0.81
(20, 20)	186	0.8401	0.84	0.83
(30, 15)	234	0.8537	0.86	0.85
(50, 20)	281	0.8776	0.88	0.86
(50, 30, 20)	295	0.8793	0.88	0.88

A tabela permite observar que, de modo geral, o aumento da complexidade da rede neural, representado pela maior quantidade de neurônios e camadas, resultou em uma melhora nas métricas de desempenho. Apesar disso, esse ganho veio acompanhado de um aumento no número de iterações necessárias para o treinamento. No entanto, para este conjunto de dados, esse acréscimo não representou um problema significativo em termos de tempo de processamento.

5 Conclusão

De forma geral, tanto a Rede Neural quanto a Árvore de Decisão apresentaram desempenhos semelhantes, cada uma com suas particularidades. No entanto, é importante destacar que a transformação do problema original, de regressão para classificação, pode ter impactado negativamente os resultados. Durante os testes iniciais, foram realizadas diversas variações na forma de agrupar as classes, algumas priorizando o balanceamento entre elas. No entanto, observou-se que o balanceamento não gerou melhorias significativas no desempenho dos modelos.

Além disso, mudanças aparentemente simples, como a substituição da média pelo uso do KNNImputer para preenchimento de valores ausentes, alteraram de forma significativa o comportamento dos algoritmos. Vale destacar ainda que, mesmo com o uso de imputação, alguns atributos apresentavam grande quantidade de dados faltantes, o que também pode ter influenciado as métricas finais.

Todavia, foi possível aplicar, de maneira prática, os principais conceitos trabalhados em aula, como o pré-processamento de dados, a análise e visualização de informações relevantes e a implementação de modelos de aprendizado supervisionado. A atividade proporcionou uma compreensão mais sólida de como esses elementos se interligam na construção de soluções baseadas em dados reais.

6 Referências

KAGGLE. Life Expectancy (WHO). Kaggle, 2017. Disponível em: https://www.kaggle.com/datasets/kumarajarshi/life expectancy-who. Acesso em: 17 maio 2025.

PANDAS. Pandas Documentation. Disponível em: https://pandas.pydata.org/. Acesso em: 17 maio 2025.

NUMPY. NumPy. Disponível em: https://numpy.org/. Acesso em: 17 maio 2025.

WASKOM, M. Seaborn: Statistical Data Visualization. Disponível em: https://seaborn.pydata.org/. Acesso em: 19 maio 2025.

HUNTER, J. D. et al. Matplotlib. Disponível em: https://matplotlib.org/. Acesso em: 19 maio 2025.

SCIKIT-LEARN DEVELOPERS. Scikit-learn: Machine Learning in Python. Disponível em: https://scikit-learn.org/. Acesso em: 19 maio 2025.

DISTRICT DATA LABS. Yellowbrick: Visualizing the Scikit-Learn Model Selection Process. Disponível em: https://www.scikit-yb.org/. Acesso em: 19 maio 2025.

SCIPY DEVELOPERS. SciPy: Scientific computing tools for Python. Disponível em: https://scipy.org/. Acesso em: 19 maio 2025.