UNIVERSIDADE ANHEMBI MORUMBI

CAMPUS PAULISTA

INTELIGÊNCIA ARTIFICIAL

ATIVIDADE AVALIATIVA 3

Integrantes:

Eduarda Tobias Fernandes - RA: 12522216514;

Jennifer Reis Sicherolli De Souza - RA: 125221110547;

Katerine Linda Witkoski - RA: 1252221362;

Nathalie Mori Taylor Zampieri - RA: 1252224816;

São Paulo/SP - 2024

Regressão

Histórico

A regressão foi introduzida por Sir Francis Galton no final do século XIX enquanto estudava a hereditariedade, ele percebeu que os filhos apresentavam as mesmas características dos seus pais, porém em uma intensidade menor, por exemplo: alturas dos descendentes tendiam a "regredir" para a média da população, o que levou ao termo "regressão".

Os algoritmos de regressão são importantes para extração de informação de dados, bastante utilizada quando se deseja prever valores a partir de uma ou mais variáveis explicativas.

A regressão, em termos gerais, refere-se a qualquer método que tenta modelar e compreender a relação entre uma variável dependente (ou resposta) e uma ou mais variáveis independentes (ou preditoras). Pode ser linear ou não-linear, paramétrica ou não-paramétrica, etc.

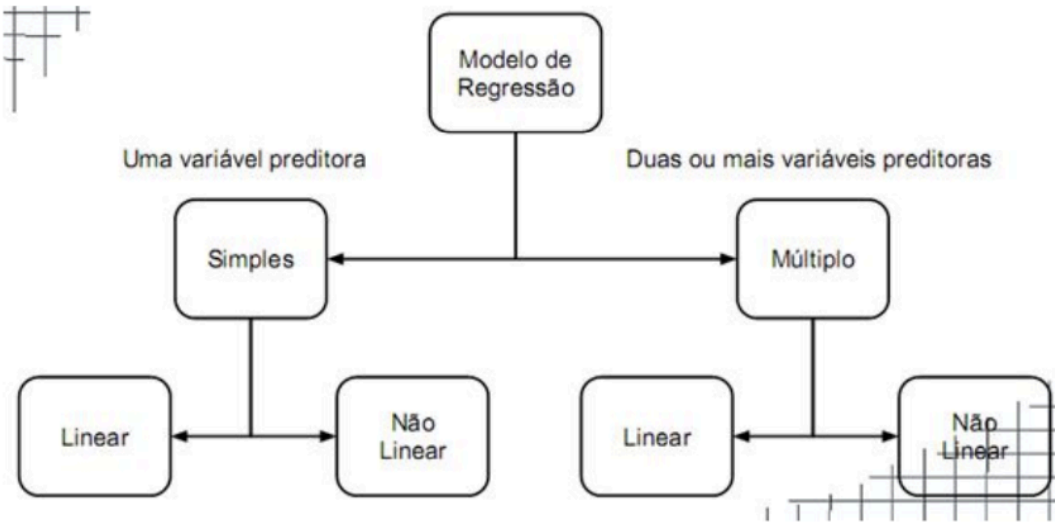


Figura 1.[1]

Principais técnicas de algoritmos de regressão:

Existem várias técnicas de regressão, como regressão linear, regressão logística, regressão polinomial, regressão de Ridge, Lasso, Elastic Net, e regressão de árvore (incluindo

[1] Fonte: OLIVEIRA, A.; ARAUJO, C.; JANAILDA, I. **Análise de Regressão Tópicos em Avaliação de Desempenho de Sistemas.** [s.l: s.n.]. Disponível em: <https://www.modcs.org/wp-content/uploads/2015/12/Analise%20de%20Regressao.pdf>.

Random Forest), cada uma com suas próprias aplicações e vantagens. Iremos comentar sobre algumas das principais técnicas abaixo:

1) **Regressão Linear:**

Regressão linear é uma técnica que assume que a relação entre a variável dependente e as variáveis independentes é linear. Sendo também um tipo de algoritmo supervisionado, portanto, antes de entender como funciona o algoritmo é importante conhecer o que seria aprendizado supervisionado.

1.1) Regressão linear Simples: temos somente uma variável independente (X) para fazermos a predição. Consiste em achar uma reta que relacione duas variáveis quantitativas.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \epsilon$$

Onde,

Y: é a variável dependente (ou explicada).

$X_1, X_1, \ldots, X_k$: são as variáveis independentes (ou explicativas).

$\beta_0, \beta_1, \ldots, \beta_k$: parâmetros da regressão ou coeficientes angulares.

ϵ: é o termo que representa o erro aleatório.

Veja a seguir um exemplo gráfico de um modelo de uma regressão linear simples (quando temos duas variáveis e a relação entre elas pode ser representada por uma linha reta):

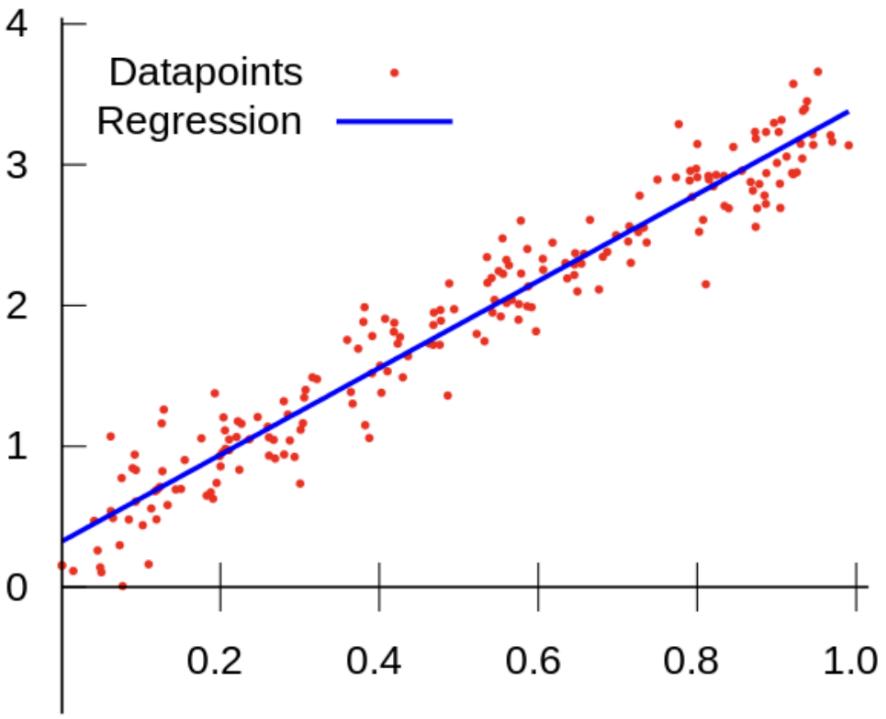


Figura 2. [2]

Podemos analisar o resultado da regressão avaliando os valores reais e os valores preditos através da reta de regressão.

1.2) Regressão Linear Múltipla: várias variáveis independentes (X) usadas para fazer a predição (envolve três ou mais variáveis). Essa abordagem mais abrangente permite modelar relações mais complexas entre as variáveis, capturando o efeito combinado e interativo de múltiplos fatores na variável dependente.

Na Regressão Linear Múltipla, a fórmula é bem parecida, só vamos acrescentar outras variáveis preditoras:

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2$$

Na fórmula acima temos duas variáveis preditoras e seus betas. Dando continuidade a este modelo, podemos ter quantas variáveis preditoras quanto quisermos:

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n$$

[2] Fonte: FAGNA, M. **6 Machine Learning - Regressão | Tutorial Shiny com Machine Learning usando Tidymodels (em desenvolvimento)**. Disponível em: <https://bookdown.org/fagna/_machine_learning_shiny_tutorial/machine-learning---regress%C3%A3o.html#regress%C3%A3o-linear>. Acesso em: 20 maio. 2024.

Na figura abaixo conseguimos comparar o algoritmo da regressão linear Simples para Regressão Linear Múltipla:

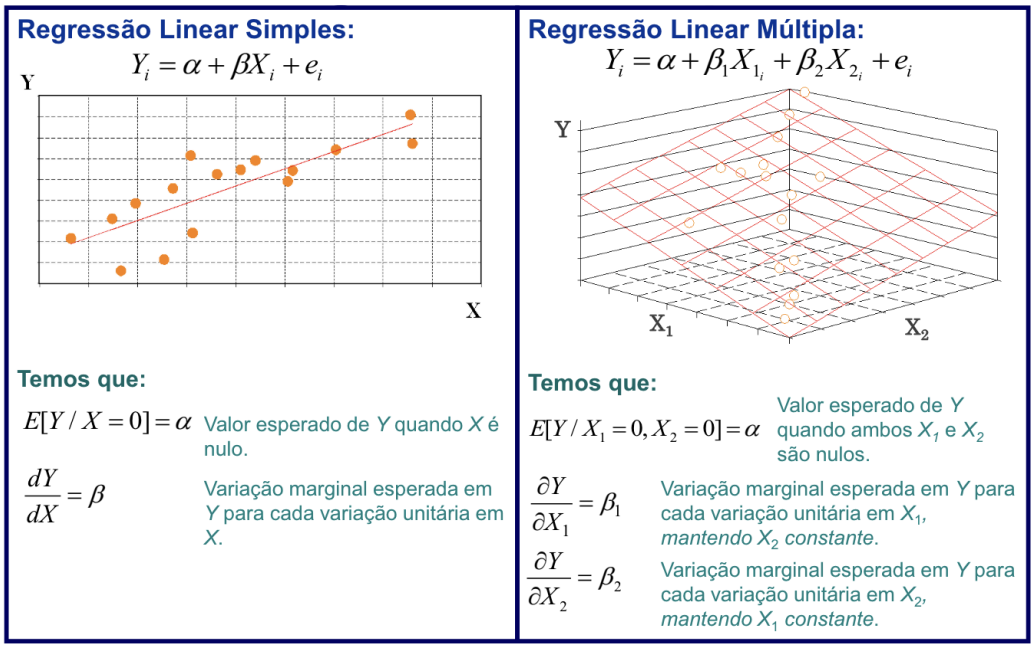


Figura 3. [3]

2) Regressão com Random Forest

Consiste em uma técnica de aprendizado de máquina que utiliza a construção de múltiplas árvores de decisão (Geralmente treinados com o método de Bagging[4]) para modelar e prever uma variável contínua. A ideia principal do método é criar combinações de modelos que aumentem o resultado do sinal. Um dos principais benefícios de se trabalhar com esse algoritmo é a possibilidade de se trabalhar com um grande conjunto de dados de maior dimensão.

---

[3] Fonte: GORI, E. **Regressão Linear Múltipla Ementa: • Definição; • Estimadores Mínimos de Mínimos Ordinários; • Teorema de Gauss-Markov.** [s.l: s.n.]. Disponível em: <https://www4.eco.unicamp.br/docentes/gori/images/arquivos/Econometrial/Econometria_RegressaoMultipla.pdf>.

[4] "A técnica de bootstrap que vimos anteriormente pode ser extremamente útil para o cálculo de desvio-padrão em situações onde isso pode não ser possível. Já aqui, vamos ver o emprego dessa técnica com o intuito de aumentar a performance de métodos de aprendizado como as árvores de decisão." FILHO, L. H. B. **Bagging, Random Forests e Boosting**. Disponível em: <https://analisemacro.com.br/economia/macroeconometria/bagging-random-forests-e-boosting/>. Acesso em: 20 maio. 2024.

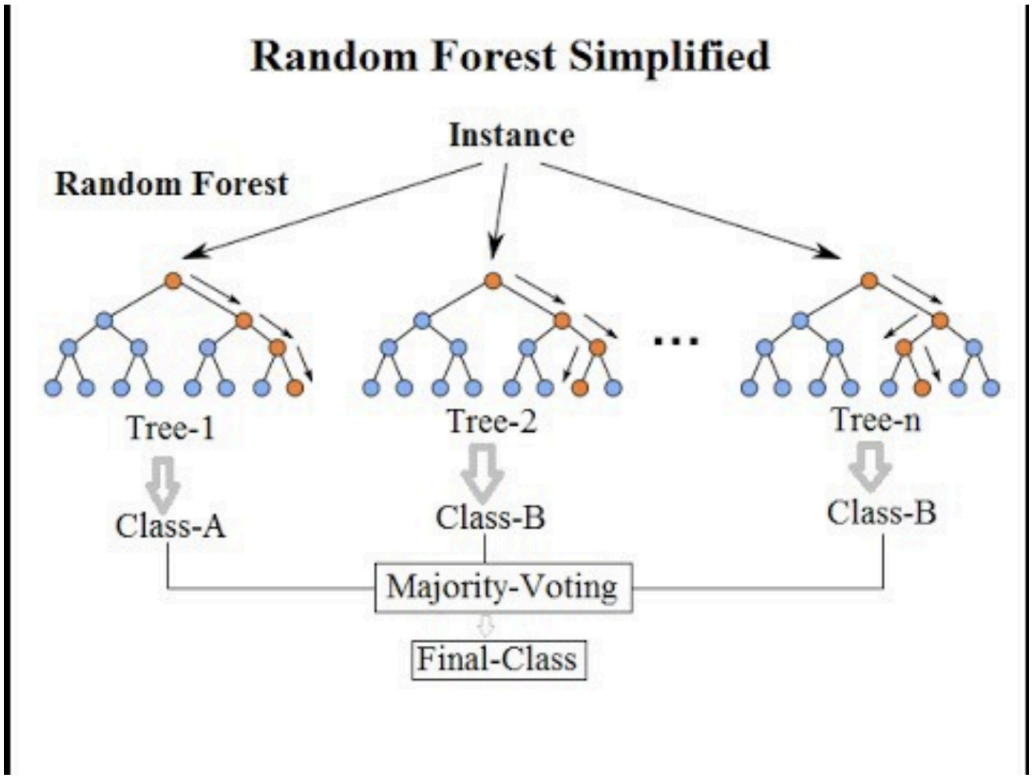


Figura 4.[5]

O Random Forest é um modelo versátil, semelhante às árvores de decisão, adequado tanto para tarefas de classificação quanto de regressão. Ele possui viés e variância reduzidos, resultando em melhores e mais consistentes resultados, além de oferecer robustez ao modelo.

3) Regressão com XGBoost

O “Extreme Gradient Boosting” (XGBoost) , é uma técnica avançada de aprendizado de máquina baseada em árvores de decisão que utiliza a estrutura de Gradient boosting[6]. Ele combina técnicas de otimização do uso do software e hardware, como, por exemplo, técnicas que ajudam a evitar *overfitting* (ajuste excessivo) e produz resultados superiores usando menos recursos de computação.

---

[5] Fonte: **ESTATSITE.COM.BR.** Disponível em: <https://estatsite.com.br/>. Acesso em: 20 maio. 2024.

[6] "O gradient boosting é uma técnica de aprendizado de máquina para problemas de regressão e classificação, que produz um modelo de previsão na forma de um ensemble de modelos de previsão fracos, geralmente árvores de decisão. Ela constrói o modelo em etapas, como outros métodos de boosting, e os generaliza, permitindo a otimização de uma função de perda diferenciável arbitrária." **Gradient boosting.** Disponível em: <https://pt.wikipedia.org/wiki/Gradient_boosting>. Acesso em: 20 maio. 2024.

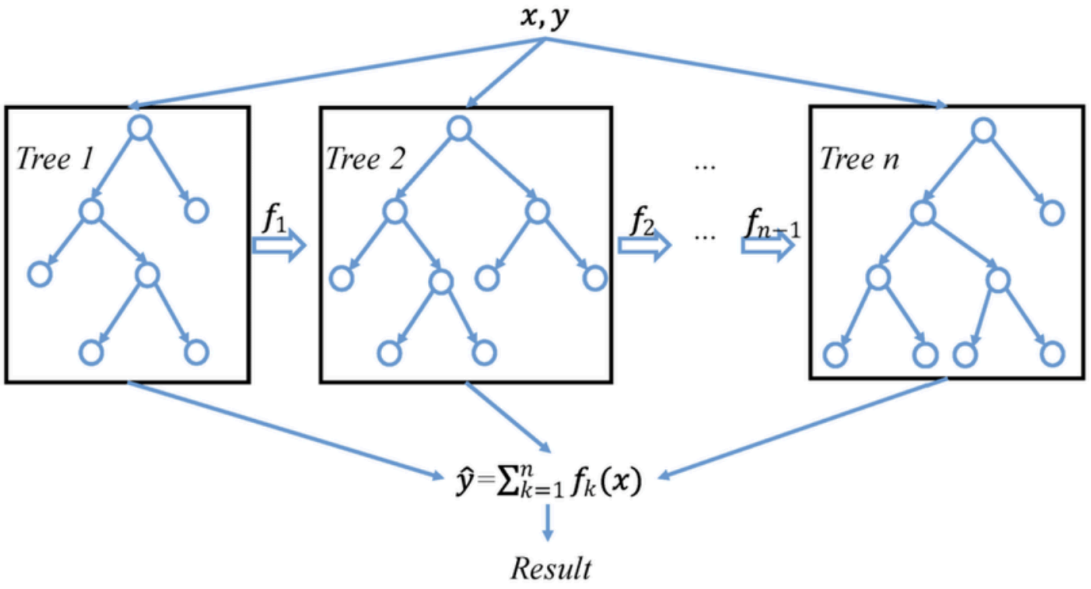


Figura 5. [7]

Descrição do funcionamento do algoritmo:

Como comentamos acima, a regressão é um conjunto de métodos estatísticos que permitem prever ou estimar um valor dependente (variável dependente) com base em um ou mais valores independentes (variáveis independentes). O objetivo principal é encontrar a relação entre as variáveis, geralmente expressa por uma função matemática.

Base de dados: Recolher dados históricos que contenham variáveis de interesse. A análise de dados pode ser feita por ferramentas estatísticas que possibilitam correlacionar de diversas maneiras as variáveis envolvidas.

Análise Exploratória: Analisar os dados para entender as relações e identificar padrões.

Modelagem: Escolher o tipo de regressão apropriado (linear, múltipla, radom forest, etc.) e ajustar o modelo aos dados. Isso envolve a determinação dos coeficientes que minimizam a diferença entre os valores observados e os valores previstos.

Previsão: Usar o modelo para prever valores futuros ou desconhecidos da variável dependente com base nos valores das variáveis independentes.
Histórico:

[7] Fonte: FAGNA, M. **6 Machine Learning - Regressão | Tutorial Shiny com Machine Learning usando Tidymodels (em desenvolvimento)**. Disponível em: <https://bookdown.org/fagna/_machine_learning_shiny_tutorial/machine-learning---regress%C3%A3o.html#regress%C3%A3o-linear>. Acesso em: 20 maio. 2024.

Para exemplificar o funcionamento do algoritmo iremos focar técnica de regressão linear simples em Python. Portanto, para implementação é necessário:

- Utilizar as bibliotecas como: numpy, matplotlib e pandas em Python e o método dos mínimos quadrados.
- Base de dados (csv ou xlsx, por exemplo)

Exemplo em código:

```python
def regressao_linear(df):
    # TRAIN TEST SPLIT
    # Para a regressão as variáveis serão x (como um conjunto de todas as colunas menos o recurso alvo a ser observado) e Y (life_expectancy)
    X = df.drop(['life_expectancy'], axis =1)
    y = df['life_expectancy']

    # . Dividir os dados em conjuntos de treinamento e teste permite avaliar o desempenho do modelo em dados não vistos, ajudando a identificar se o
    # modelo está superestimando (overfitting) ou subestimando (underfitting) os dados. No nosso caso, 30% dos dados serão utilizados para teste e 70% para treinamento.

    X_train_org, X_test_org, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 0)

    # Criando uma instância MinMaxScaler (método de normalização que transforma os dados de um intervalo específico, geralmente entre 0 e 1.)

    scaler = MinMaxScaler()
    X_train = scaler.fit_transform(X_train_org)
    X_test = scaler.transform(X_test_org)

    print("X_train shape: ",X_train.shape)
    print("y_train shape: ",y_train.shape)
    print("X_test shape: ",X_test.shape)
    print("y_test shape",y_test.shape)

    # Em resultado vimos que 2056 são X_train e 882 são X_test

    # Criando uma Instância do Modelo de Regressão Linear
    lreg = LinearRegression()
    # Ajusta o modelo de regressão linear (lreg) aos dados de treinamento normalizados (X_train e y_train)
    lreg.fit(X_train, y_train)

    print('Train Score: {:.4f}'.format(lreg.score(X_train, y_train)))
    print('Test Score:{:.4f}'.format(lreg.score(X_test, y_test)))

    ylinear_predicted = lreg.predict(X_test)

    print('MSE:', metrics.mean_squared_error(y_test,ylinear_predicted))
    print('R2_score: {:.4f}'.format(metrics.r2_score(y_test,ylinear_predicted)))

    # Plotando os valores reais versus os valores previstos pelo modelo
    plt.scatter(ylinear_predicted, y_test, color = 'blue')
    plt.plot(y_test, y_test, color='red', label='Linha de Regressão')

    # Adicionando legenda
    plt.legend()

    plt.show()
```

Figura 6.[8]

A normalização (ou padronização) de dados é uma etapa crucial no pré-processamento de dados para muitos algoritmos de aprendizado de máquina, para isso fazemos um método de divisão de teste de treinamento (Train Test Split), permitindo uma validação de modelos e simulação do desempenho do modelo com novos dados.

Isso consiste em amostragem aleatória sem substituição de cerca de 70% das linhas (isso pode variar) e colocá-las em seu conjunto de treinamento. Os 30% restantes são colocados em seu conjunto de teste. (“X_train,” “X_test,” “y_train,” “y_test”)[9]

[8]Código: https://github.com/katerine-dev/regressao/blob/main/LifeExpect.py

[9] Fonte: GALARNYK, M. **Train Test Split: What it Means and How to Use It | Built In**. Disponível em: <https://builtin.com/data-science/train-test-split>. Acesso em 25 de Maio de 2023.

Esse tipo de teste possui vários benefícios, promover a convergência mais rápida dos modelos, evitar problemas numéricos, melhorar a interpretabilidade e a comparação entre features, e reduzir a sensibilidade a outliers. Além de evitar modelos excessivamente complexos que não generalizam bem para novos dados.

Características:

Além das características de comentamos, podemos concluir que regressão são modelos que visam predizer o comportamento de uma variável dependente (Y) com uma função de uma ou mais variáveis independentes (X) e que existem diferentes tipos de técnicas de regressão, cada uma adequada a diferentes tipos de dados e relações entre variáveis.

1) Previsão e estimativa: modelos de regressão são usados para prever valores futuros ou desconhecidos da variável dependente com base em variáveis independentes conhecidas.
2) Coeficientes: os coeficientes estimados representam a magnitude e a direção do impacto das variáveis independentes sobre a variável dependente.
3) Avaliação do Modelo: a qualidade do ajuste do modelo é frequentemente avaliada por métricas como o coeficiente de determinação (R^2), erro quadrático médio (MSE), erro absoluto médio (MAE) e outros.
4) Resíduos: Os resíduos (diferença entre os valores observados e os valores previstos) são analisados para verificar a adequação do modelo e a conformidade com as suposições de regressão.

As suposições de regressão linear incluem a existência de uma relação linear entre variáveis independentes e dependentes, a ausência de outliers[10], e a distribuição normal dos termos de erro, homocedasticidade[11], independência dos erros e multicolinearidade[12].

---

[10] "Os outliers são dados que se diferenciam drasticamente de todos os outros. Em outras palavras, um outlier é um valor que foge da normalidade e que pode (e provavelmente irá) causar anomalias nos resultados obtidos por meio de algoritmos e sistemas de análise." ANALYTICS, A. A. **Outliers, o que são e como tratá-los em uma análise de dados?** Disponível em: <https://aquare.la/o-que-sao-outliers-e-como-trata-los-em-uma-analise-de-dados/>.

[11] "Homocedasticidade refere-se à variância constante dos resíduos." **HOMOCEDASTICIDADE NA ANÁLISE DE REGRESSÃO EM AVALIAÇÕES IMOBILIÁRIAS: A IMPORTÂNCIA DA VARIÂNCIA CONSTANTE DOS RESÍDUOS** – Escola do Avaliador – Cursos e Treinamentos de Avaliação de Imóveis. Disponível em: <https://escoladoavaliador.com.br/2024/02/01/homocedasticidade-na-analise-de-regressao-em-avaliacoes-imobiliarias-a-importancia-da-variancia-constante-dos-residuos/#:~:text=Import%C3%A2ncia%20da%20Vari%C3%A2ncia%20Constante%3A%20Homocedasticidade>. Acesso em: 20 maio. 2024.

[12] "Multicolinearidade consiste em um problema comum em regressões, no qual as variáveis independentes possuem relações lineares exatas ou aproximadamente exatas." Multicolinearidade. Disponível em: <https://pt.wikipedia.org/wiki/Multicolinearidade>.

Áreas de aplicação:

Os modelos de regressão são ferramentas poderosas e flexíveis para análise preditiva e descritiva, permitindo a compreensão e previsão de relações complexas entre variáveis.

Esse algoritmo pode ser utilizado em qualquer problema, onde as variáveis de entrada e saída são valores contínuos, como, por exemplo:

- ➔ Economia: Previsão de tendências de mercado, análise de investimentos, modelagem de crescimento econômico, analisar a relação entre variáveis macroeconômicas, como taxa de juros, inflação e investimento.
- ➔ Ciências Sociais: Estudos de relações sociais, previsão de comportamentos humanos.
- ➔ Saúde: Modelagem de risco, análise de sobrevivência, predição de resultados de tratamentos, análise de eficácia de tratamentos, na relação entre fatores de risco e doenças, ou na previsão de resultados médicos com base em múltiplas variáveis.
- ➔ Marketing: Previsão de vendas, análise de comportamento do consumidor, otimização de campanhas publicitárias.
- ➔ Engenharia: Modelagem de falhas, previsão de desempenho de sistemas.
- ➔ Ciências Ambientais: Previsão de mudanças climáticas, modelagem de poluição, análise de recursos naturais.
- ➔ Finanças: Precificação de ativos, análise de risco, previsão de retornos financeiros.

Implementação da regressão em Python:

A partir dos materiais disponibilizados pelo professor e pesquisas complementarem realizamos um projeto em Python para aplicar os conceitos abordados acima.

Esse projeto está disponível no github[13] e falaremos melhor sobre os gráficos gerados nos tópicos abaixo.

Banco de dados utilizado:

A base utilizada foi encontrada no site Kaggle Datasets[14], ela é composta por 22 colunas, e 2938 linhas. Para realizar as análises foi necessário uma "faxina dos dados". Limpando nomes de colunas, tirando informações de '*Null*' e vazios. Abaixo Você pode visualizar as informações das colunas e types:

[13] https://github.com/katerine-dev/regressao

[14] Fonte: **Life Expectancy Regression**. Disponível em: <https://www.kaggle.com/code/wrecked22/life-expectancy-regression/comments>. Acesso em: 25 maio. 2024.

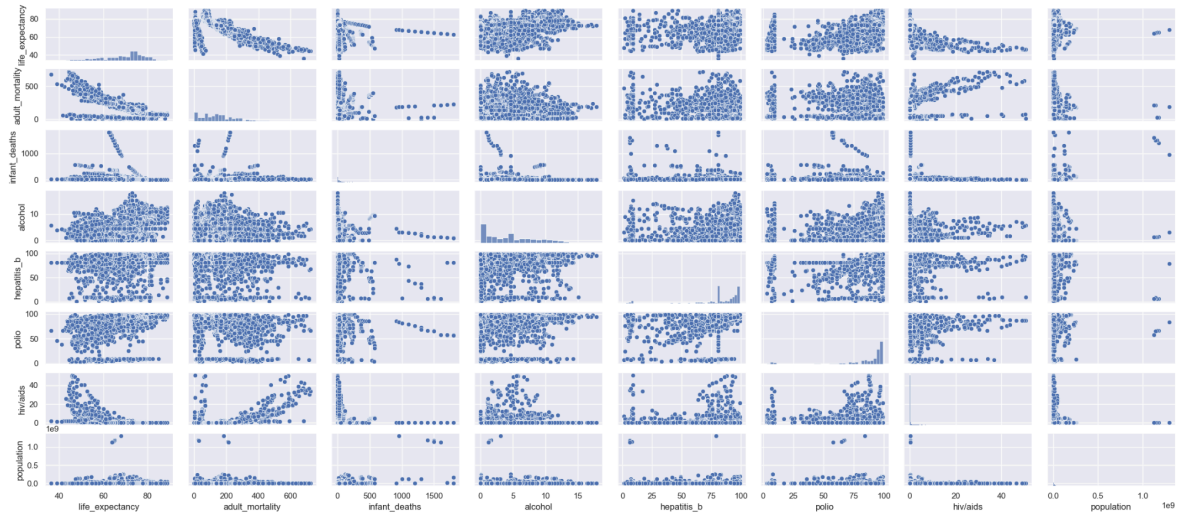
```
174    informacao =  df.info()
175    print(informacao)

PROBLEMAS   SAÍDA   CONSOLE DE DEPURAÇÃO   TERMINAL   PORTAS   SQL CONSOLE   GITLENS

 ~/Doc/regressao  on  main !1  /Users/katerinewitkoski/Documents/regressao/.venv/bin/python /Users/katerinewitkoski/Documents/regressao/LifeExpect.py
Index(['country', 'year', 'status', 'life_expectancy', 'adult_mortality',
       'infant_deaths', 'alcohol', 'percentage_expenditure', 'hepatitis_b',
       'measles', 'bmi', 'under-five_deaths', 'polio', 'total_expenditure',
       'diphtheria', 'hiv/aids', 'gdp', 'population', 'thinness__1-19_years',
       'thinness_5-9_years', 'income_composition_of_resources', 'schooling'],
      dtype='object')
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2938 entries, 0 to 2937
Data columns (total 22 columns):
 #   Column                           Non-Null Count  Dtype
---  ------                           --------------  -----
 0   country                          2938 non-null   int64
 1   year                             2938 non-null   int64
 2   status                           2938 non-null   int64
 3   life_expectancy                  2938 non-null   float64
 4   adult_mortality                  2938 non-null   float64
 5   infant_deaths                    2938 non-null   int64
 6   alcohol                          2938 non-null   float64
 7   percentage_expenditure           2938 non-null   float64
 8   hepatitis_b                      2938 non-null   float64
 9   measles                          2938 non-null   int64
 10  bmi                              2938 non-null   float64
 11  under-five_deaths                2938 non-null   int64
 12  polio                            2938 non-null   float64
 13  total_expenditure                2938 non-null   float64
 14  diphtheria                       2938 non-null   float64
 15  hiv/aids                         2938 non-null   float64
 16  gdp                              2938 non-null   float64
 17  population                       2938 non-null   float64
 18  thinness__1-19_years             2938 non-null   float64
 19  thinness_5-9_years               2938 non-null   float64
 20  income_composition_of_resources  2938 non-null   float64
 21  schooling                        2938 non-null   float64
dtypes: float64(16), int64(6)
memory usage: 505.1 KB
None
```

Figura 7.[15]

Gráficos gerados:

Todo o código dos gráficos e imagens geradas estão no repositório, já disponível como fonte desse trabalho.

Gráfico pairplot:



[15] Código: https://github.com/katerine-dev/regressao/blob/main/LifeExpect.py

Gráfico de Matriz de correlação:

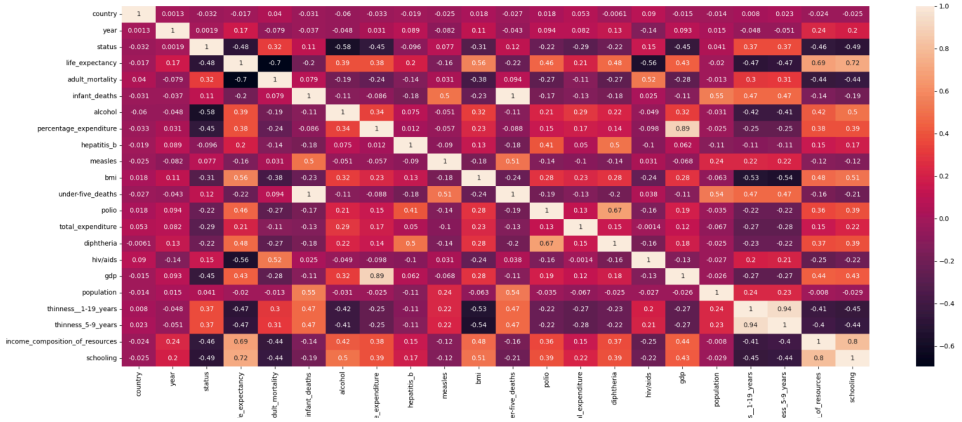


Gráfico de matriz de colunas:

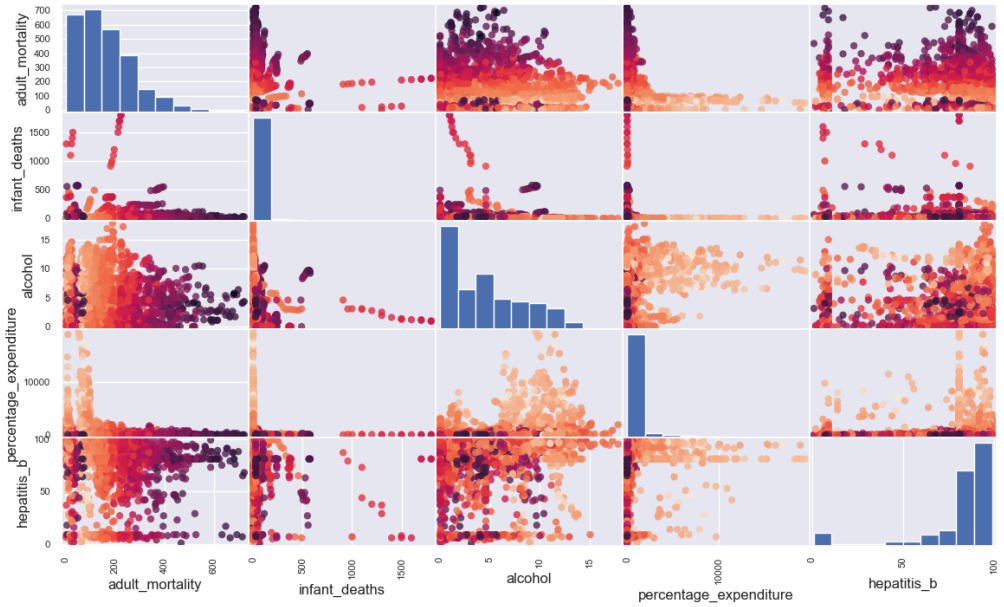


Gráfico de regressão linear:

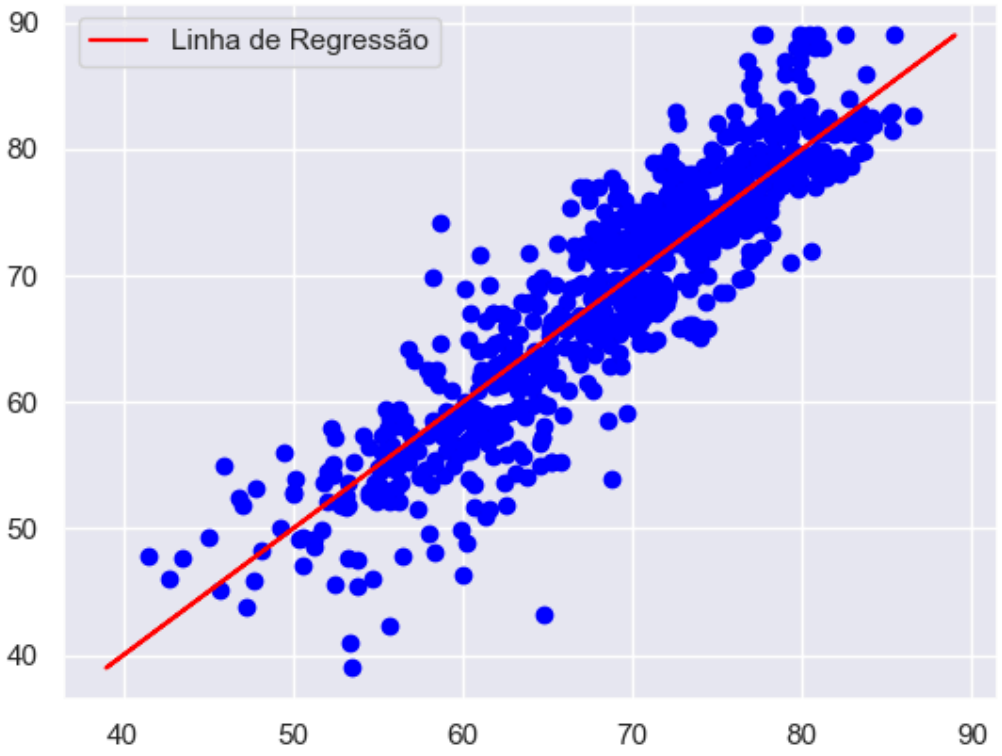
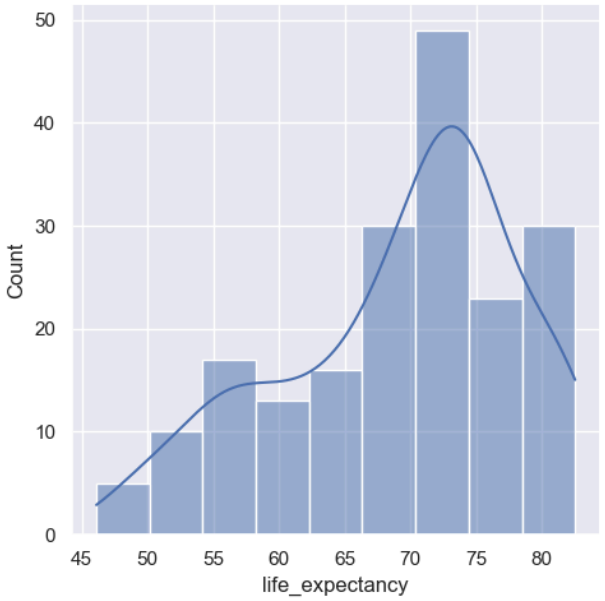


Gráfico de distribuição expectativa de vida:



Conclusão:

Utilizando uma base de dados sobre faixas etárias, foi possível realizar este projeto em python analisando por meio da regressão linear e a geração de gráficos a expectativa de vida estimada. Por meio dos gráficos gerados foi possível analisar e visualizar de forma detalhada a concentração de pessoas por faixa etária e dessa forma prever uma média de qual seria a expectativa de vida. Dessa forma, podemos concluir que a expectativa de vida estimada é em torno de 70 anos.

Referências:

BOAS, D. **Algoritmos de Regressão: Uma abordagem dos principais**. Disponível em: <https://ianaveia.com/algoritmos-de-regressao/>. Acesso em: 20 maio. 2024.

DAMACENO, L. **Regressão Linear?** Disponível em: <https://medium.com/@lauradamaceno/regress%C3%A3o-linear-6a7f247c3e29>.

FAGNA, M. **6 Machine Learning - Regressão | Tutorial Shiny com Machine Learning usando Tidymodels (em desenvolvimento)**. Disponível em: <https://bookdown.org/fagna/_machine_learning_shiny_tutorial/machine-learning---regress%C3%A3o.html#regress%C3%A3o-linear>.

Guia Completo sobre Regressão: Conceito, Tipos e Aplicações | Blog do Foco Educação Profissional. Disponível em: <https://www.focoeducacaoprofissional.com.br/blog/guia-completo-sobre-regress%C3%A3o-conceito-tipos-e-aplicacoes>. Acesso em: 20 maio. 2024.

OLIVEIRA, A.; ARAUJO, C.; JANAILDA, I. **Análise de Regressão Tópicos em Avaliação de Desempenho de Sistemas.** [s.l: s.n.]. Disponível em: <https://www.modcs.org/wp-content/uploads/2015/12/Analise%20de%20Regressao.pdf>.