

# MODELO PREDITIVO DE ENCARGOS MÉDICOS

## ANUAIS INDIVIDUAIS

Cintia Zago, Guilherme Machado, Rodrigo Vidal

[cintiazago@gmail.com](mailto:cintiazago@gmail.com) / [26guilhermemachado@gmail.com](mailto:26guilhermemachado@gmail.com) / [rodrigo.vidal@fbsbrasil.com.br](mailto:rodrigo.vidal@fbsbrasil.com.br)

FIAP – Faculdade de Informática e Administração Paulista  
São Paulo, Brasil

**Resumo** — este documento apresenta a aplicação de dois modelos de aprendizado de máquina para prever o valor dos custos médicos anuais individuais de um determinado conjunto de dados. Diferentes técnicas de pré-processamento de dados foram realizadas para utilização dos algoritmos. Foram utilizados os modelos de regressão linear e árvore de decisão.

**Keywords**— *python; machine learning; inteligência artificial;*

### I. INTRODUÇÃO

Nos últimos anos, encargos de planos de saúde tornou-se uma área de grande interesse tanto para empresas do setor médico quanto para pesquisadores. Com o aumento contínuo desses valores, é crucial desenvolver modelos que possam prever com precisão esses gastos, permitindo às pessoas uma melhor previsibilidade e gestão financeira. Este projeto tem como objetivo aplicar e comparar dois modelos de *machine learning* para prever os gastos de encargos médicos com base em um conjunto de dados que inclui atributos dos segurados, como idade, sexo, IMC, número de filhos e região.

### II. GERAÇÃO E EXPLORAÇÃO DOS DADOS

Para fins de estudo, uma base de dados fictícia será gerada utilizando um algoritmo desenvolvido em Python. O código gera um arquivo .csv com determinado número de linhas definidas pelo usuário.

O conjunto de dados é gerado com os seguintes atributos: idade, gênero, IMC, fumante, quantidade de filhos, região e encargos (atributo *target*). Cada atributo tem o seu valor gerado aleatoriamente com base em uma faixa de valores pré-definidos pelo usuário. Dependendo do valor gerado, será aplicado certo peso fixo para os cálculos dos encargos.

A implementação do algoritmo gerador de dados foi baseado em pesquisas reais e que apresenta as correlações dos atributos com os cálculos dos encargos, sendo estes relacionados diretamente com a expectativa de vida, que por sua vez, é influenciada, por exemplo, pelo consumo do cigarro, faixa etária e IMC (Índice de Massa Corporal).

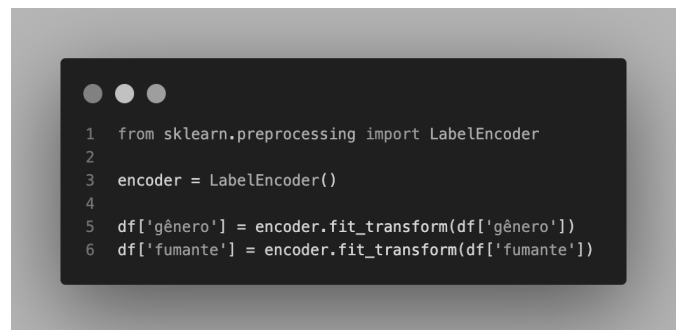
### III. PRÉ-PROCESSAMENTO

A preparação dos dados é uma etapa crucial em qualquer projeto de *machine learning*, pois garante que os dados estejam em um formato adequado para os modelos serem aplicados.

Neste projeto realizamos várias operações de pré-processamento para tratar variáveis categóricas, normalizar os dados e lidar com valores ausentes (quando necessário). Abaixo, estão as principais etapas realizadas:

#### A. Label Encoding

Convertemos as variáveis categóricas 'gênero' e 'fumante' em valores numéricos, onde cada categoria é representada por um número inteiro.



```
1 from sklearn.preprocessing import LabelEncoder
2
3 encoder = LabelEncoder()
4
5 df['gênero'] = encoder.fit_transform(df['gênero'])
6 df['fumante'] = encoder.fit_transform(df['fumante'])
```

#### B. One-Hot Encoding

Transformamos a variável categórica 'região' em múltiplas colunas binárias. Esta técnica é útil quando as categorias não possuem uma ordem intrínseca.

#### C. Normalização dos Dados

A normalização dos dados é essencial para garantir que todas as variáveis contribuam igualmente para o modelo de *machine learning*. Utilizamos a técnica de *Min-Max Scaling* para normalizar os dados numéricos, escalando-os para um intervalo de 0 a 1.

#### D. Tratamento de Valores Ausentes

Valores ausentes podem afetar o desempenho dos modelos de *machine learning*. Utilizamos o *SimpleImputer* para preencher valores ausentes com a mediana das respectivas colunas, garantindo que os dados estejam completos.

Essas etapas de pré-processamento garantem que os dados estejam adequadamente preparados para aplicação dos modelos de *machine learning*, melhorando a qualidade das previsões.

#### IV. APLICAÇÃO DOS MODELOS

Nesta seção, descreveremos os modelos de *machine learning* aplicados para prever os encargos médicos. O primeiro modelo que utilizamos foi a Regressão Linear.

Para avaliar a performance dos modelos de *machine learning*, é essencial separar os dados em dois conjuntos: um para treino e outro para teste. A função `train_test_split` da biblioteca *scikit-learn* facilita esta divisão. Ela recebe como entrada o conjunto de dados completo e retorna dois subconjuntos: um para treinar o modelo (*training set*) e outro para avaliar sua performance (*testing set*)<sup>1</sup>.

Utilizar esta função é crucial porque evita que o modelo seja avaliado nos mesmos dados nos quais foi treinado, o que levaria a uma superestimação da sua capacidade preditiva. Neste projeto, os dados foram divididos em 80% para treino e 20% para teste. Essa proporção é comum e geralmente oferece um bom equilíbrio entre ter dados suficientes para treinar o modelo e dados suficientes para avaliar sua performance.

##### A. Regressão Linear

A Regressão Linear é um dos modelos mais simples de *machine learning*. Ela assume que existe uma relação linear entre as variáveis independentes (atributos) e a variável dependente (*target*). A biblioteca Python *scikit-learn* possui a classe `LinearRegression` que foi utilizada para este projeto.

##### B. Árvore de Decisão

A Árvore de Decisão é um modelo que divide os dados em subconjuntos com base em uma série de perguntas de verdadeiro/falso sobre os atributos, resultando em uma estrutura semelhante a uma árvore. É um modelo intuitivo e fácil de interpretar. Utilizamos a biblioteca *scikit-learn*, que possui a classe `DecisionTreeRegressor`, para treinar o modelo.

#### V. AVALIAÇÃO DOS MODELOS

Para avaliar os modelos, utilizamos várias métricas, incluindo o Erro Médio Quadrático (MSE), o Erro Absoluto Médio (MAE), o coeficiente de determinação ( $R^2$ ), e o Erro Percentual Absoluto Médio (MAPE). Estas métricas nos ajudam a entender a precisão e a eficiência dos modelos na previsão dos encargos médicos.

RESULTADOS DA REGRESSÃO LINEAR

Métrica	Resultado
MAE	2627.2511
$R^2$	0.5923
MAPE	62.25%
MSE	3585.83

RESULTADOS DA ÁRVORE DE DECISÃO

Métrica	Resultado
MAE	696.3760
$R^2$	0.9533
MAPE	11.6351%
MSE	1213.95

O coeficiente de determinação varia entre 0 e 1, onde valores próximos a 1 indicam um modelo que explica bem a variabilidade dos dados.

De acordo com os gráficos de dispersão abaixo, pode-se identificar que ao utilizar-se o modelo de Regressão Linear, os valores reais e previstos possuem maior discrepância em relação ao resultado obtido utilizando-se o modelo e Árvore de Decisão, o que pode ser visto ao identificar que os pontos azuis (que indicam valores reais) e os pontos vermelhos (que indicam os valores previstos) pouco se encontram no primeiro gráfico.

Gráfico de Dispersão gerado pela Regressão Linear

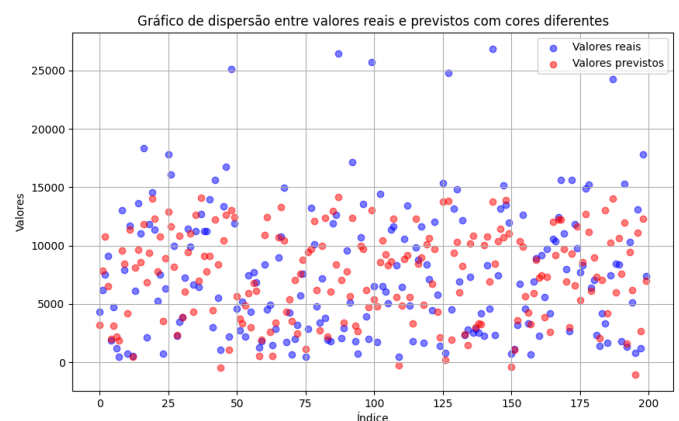
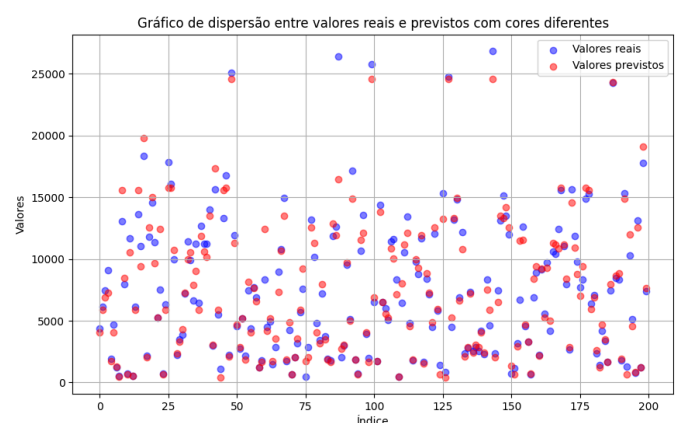


Gráfico de Dispersão gerado pela Árvore de Decisão



## VI. CONCLUSÃO

Os resultados indicam que a Regressão Linear tem uma capacidade moderada de explicar a variabilidade nos encargos médicos individuais de planos de saúde, com um coeficiente de determinação de 0.5923. No entanto, os valores relativamente altos de MAPE e MAE sugerem que o modelo pode não ser muito preciso para algumas observações.

Os resultados para a Árvore de Decisão são significativamente melhores, com um coeficiente de determinação de 0.9533, indicando uma capacidade muito alta de explicar a variabilidade nos dados. Os valores baixos de MAPE e MAE sugerem que o modelo é bastante preciso na predição dos gastos.

Assim, concluímos que:

1. A Regressão Linear pode ser mais simples e interpretável, mas pode não capturar relações não lineares nos dados.
2. A Árvore de Decisão pode capturar essas relações complexas, mas pode ser mais suscetível a *overfitting*, especialmente se a profundidade não for limitada.

## REFERÊNCIAS

- [1] Documentação oficial: *scikit-learn*. Disponível em: [https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.train\\_test\\_split.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html). Acesso em: 19 de maio de 2024.
- [2] Introduction to Machine Learning with Python. A guide for data scientists. Andreas C. Muller, Sarah Guido. O'Reilly