INTERPRETABILIDADE DE MODELOS DE APRENDIZADO DE MÁQUINA

INTERPRETABILITY OF MACHINE LEARNING MODELS

LUANA AMY NAKASUGA
Graduando em Ciência da Computação
luana.nakasuga@gmail.com

HOWARD CRUZ ROATTI
Profº. Msc. FAESA Centro Universitário
howard.cruz@faesa.br

RESUMO

A interpretabilidade dos modelos permite que os usuários identifiquem um problema que pode ocasionar desvios no comportamento previsto de um sistema e que tenham uma melhor compreensão dos resultados obtidos, entendendo como foi feita uma classificação ou como o modelo chegou a uma determinada predição para assim realizar as tomadas de decisões mais benéficas possíveis. A execução desses algoritmos favorece uma confiabilidade maior no sistema, em decorrência disso reduz diversos problemas como ineficiência do sistema, erros de funcionamento e vulnerabilidade a violações de segurança. Existem diversos modelos de aprendizado de máquina com esse intuito de auxiliar na extração de informações e tomar decisões baseadas nos dados, como a regressão linear e a regressão logística. Ambos os modelos permitem a compreensão e interpretação de seus resultados, possibilitando bons insights para análise. No caso da Regressão linear, através dos coeficientes é possível adquirir informações sobre cada variável independente do modelo, permitindo uma interpretação das funções das variáveis. Essa interpretação permite identificar qual variável tem maior influência nos resultados. Já em relação a Regressão Logística, os coeficientes estão ligados a probabilidade de pertencer a uma classe específica. É fornecido informações sobre a direção e o impacto relativo das variáveis na classificação binária, permitindo quais variáveis tem maior influência na determinação das categorias. A interpretabilidade desses modelos é de suma importância para inúmeras áreas, como em medicina, que possibilita identificar fatores de risco em doenças e condições médicas. Em conclusão, a interpretabilidade dos modelos de regressão linear e regressão logística desempenha um papel fundamental na compreensão e confiança nos resultados desses modelos. A capacidade de interpretar e explicar as relações entre variáveis independentes e dependentes permite uma compreensão mais profunda dos fenômenos em estudo, fornecendo bons insights para a tomada de decisões informadas.

Palavras-chave: Interpretabilidade de modelo. Tomada de decisões. Redução de problemas. Regressão linear. Regressão logística. Fatores de risco.

ABSTRACT

The interpretability of models allows users to identify issues that may cause deviations in the predicted behavior of a system and gain a better understanding of the obtained results, comprehending how a classification was made or how the model arrived at a specific prediction, enabling more beneficial decision-making. The execution of these algorithms enhances system reliability, reducing problems such as inefficiency, operational errors, and vulnerability to security breaches. There are several machine learning models designed to assist in extracting information and making data-driven decisions, such as linear regression and logistic regression. Both models allow for the comprehension and interpretation of their results, providing valuable insights for analysis. In the case of linear regression, information about each independent variable can be acquired through the coefficients, enabling the interpretation of variable functions. This interpretation helps identify which variable has a greater influence on the results. As for logistic regression, the coefficients are associated with the probability of belonging to a specific class. They provide information about the direction and relative impact of variables in binary classification, revealing which variables have a greater influence in determining the categories. The interpretability of these models is of utmost importance in various fields, such as medicine, where it allows for the identification of risk factors in diseases and medical conditions. In conclusion, the interpretability of linear regression and logistic regression models plays a fundamental role in understanding and having confidence in their results. The ability to interpret and explain the relationships between independent and dependent variables enables a deeper understanding of the studied phenomena, providing valuable insights for informed decision-making.

Keywords: Model interpretability. Decision-making. Problem reduction. Linear regression. Logistic regression. Risk factors.

INTRODUÇÃO

A interpretabilidade dos modelos de regressão linear e regressão logística tem sido amplamente discutida e explorada no campo do aprendizado de máquina. Esses modelos estatísticos são utilizados em diversas áreas, como medicina, finanças e ciências sociais, devido à sua capacidade de modelar e prever relações entre variáveis.

A regressão linear busca estabelecer uma relação linear entre uma variável dependente contínua e variáveis independentes, permitindo uma interpretação direta dos resultados. Já a regressão logística é usada para problemas de classificação binária, fornecendo informações sobre a probabilidade de pertencer a uma classe específica.

Neste trabalho, será apresentado um contexto, representações matemáticas e aplicações práticas desses modelos, destacando a importância da interpretabilidade na compreensão e tomada de decisões informadas. Além disso, será realizado uma explicação de um exemplo prático feito Visual Studio Code para ilustrar a interpretação dos resultados.

DESENVOLVIMENTO

1. Regressão Linear

A Regressão Linear é uma técnica estatística usada quando a variável de resposta é contínua, ou seja, pode assumir qualquer valor dentro de um intervalo específico. Então é utilizada para modelar a relação entre uma variável dependente contínua (variável de resposta) e uma ou mais variáveis independentes (variáveis preditoras), visando fazer previsões ou estimativas. O objetivo dessa regressão é encontrar uma relação linear entre os valores previstos pelo modelo e os valores reais observados nos dados.

1.1. Tipos de regressão linear

Existem dois principais tipos de Regressão Linear: a Regressão Linear Simples e a Regressão Linear Múltipla. Na Regressão Linear Simples, há uma única variável independente, enquanto na Regressão Linear Múltipla, há várias variáveis independentes.

1.2. Representação matemática

A equação da regressão linear simples é dada por:

$y = \beta0 + \beta1 * x$

Onde y é a variável dependente, x é a variável independente, β0 é o intercepto da linha de regressão e β1 é o coeficiente angular que representa a inclinação da linha.

Para a equação da regressão linear múltipla, é inserido na equação os coeficientes necessários para cada variável independente adicional.

1.3. Aplicações

A regressão linear é amplamente utilizada na área da economia, na análise de relações de variáveis econômicas, como o produto interno bruto, consumo e o investimento ou no estudo da demografia podendo projetar o crescimento populacional com base em dados demográficos anteriores.

1.4. Exemplo

Para exemplificar a Regressão Linear, apresenta-se abaixo um conjunto de dados que relaciona o número de horas de treino de uma atleta (variável independente, denotada de X) com a sua pontuação na competição (variável dependente, denotada de y).

Horas de Treino	Pontuação na Competição
2	6.3
4	12.5
5	14.6
6	15.9
7	18.2
8	19.8

Realizando o exemplo no editor de código Visual Studio Code, é necessário baixar as bibliotecas necessárias para criar o *dataframe* que será usado e construir todo o algoritmo.

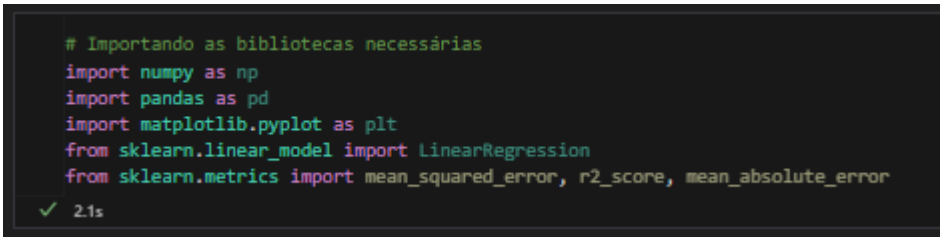

Figura 1: Importação das bibliotecas.

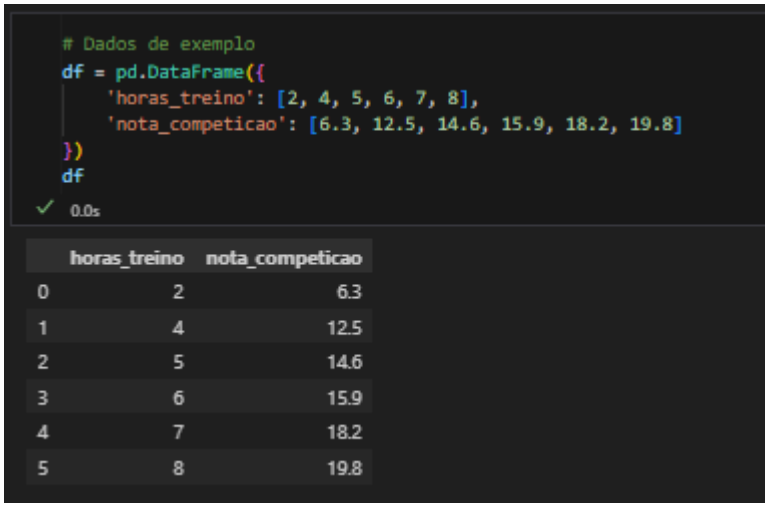

Figura 2: Inserção dos dados em um dataframe e impressão.

Como observado na descrição do exemplo a variável independente (X) é horas de treino e a dependente (y) é a pontuação na competição.

```
# Definindo as variáveis independentes (X) e dependentes (y)
X = df[['horas_treino']]
y = df['nota_competicao']
✓ 0.0s
```

Figura 3: Definição das variáveis X e y.

Após a definição das variáveis, é possível ajustar o modelo e obter os coeficientes angular e linear para assim gerar um gráfico para melhor visualização dos dados.

```
# Criando e ajustando o modelo
model = LinearRegression()
model.fit(X, y)
✓ 0.0s
```

Figura 4: Criação e ajuste do modelo.

```
# Definindo e imprimindo os coeficientes
a = model.coef_[0]
b = model.intercept_
print(f"Coeficiente angular: {a:.2f}")
print(f"Coeficiente linear: {b:.2f}")
✓ 0.0s
Coeficiente angular: 2.19
Coeficiente linear: 2.85
```

Figura 5: Definição dos coeficientes e impressão.

Abaixo mostra o gráfico de regressão linear com base nos dados apresentados, onde é possível interpretar que há um aumento da pontuação na competição à medida que o número de horas de treino aumenta.

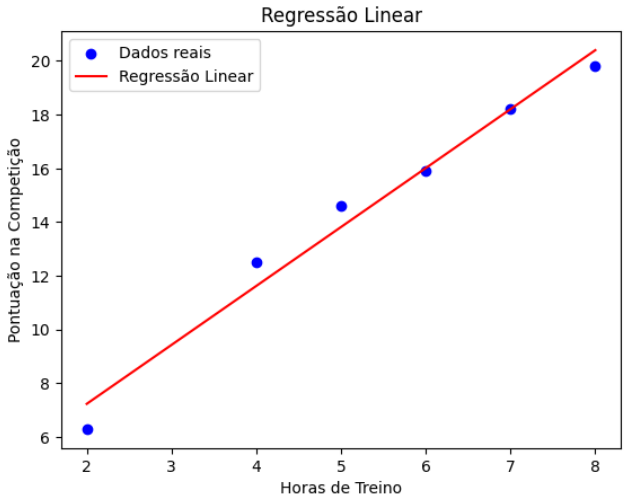

Figura 6: Projeção do gráfico.

Apresentando a equação y= 2.19 + 2.85 * x, é esperado um aumento de 2.19 pontos a cara hora adicional de treino e a nota esperada para uma ginasta sem treino nenhum é de 2.85 pontos.

1.4.1. Avaliação de desempenho do modelo

O desempenho do modelo pode ser avaliado por meio das métricas de avaliação que fornecem informações sobre a qualidade das predições ou classificações feitas pelo modelo. Os resultados dessas métricas são geradas a partir da comparação das predições do modelo com valores reais pertencentes a dados de teste ou validação. As métricas mais comuns a serem utilizadas são o RMSE, SER, MAE e o R2.

1.4.1.1. Root Mean Squared Error ou RMSE

O *Root Mean Squared Error* ou RMSE indica a raiz quadrada do erro médio quadrático das previsões, ou seja, essa métrica fornece uma medida de dispersão dos resíduos e indica o quanto o modelo se desvia, em média, dos valores reais. Portanto quanto menor for o resultado do cálculo, melhor o ajuste do modelo.

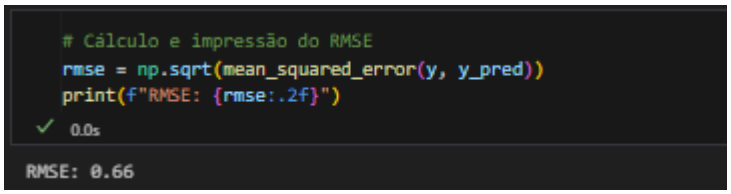

Figura 7: Cálculo e impressão do Root Mean Squared Error.

Com o RMSE em 0.66, interpretamos que essa diferença representa a precisão média do modelo nas previsões da pontuação da atleta com base no número de horas de treino.

1.4.1.2. Residual Standart Error ou RSE

O *Residual Standart Error* ou RSE indica a raiz quadrada do erro médio quadrático dos resíduos da regressão. Semelhante ao RMSE, também é medido a dispersão dos resíduos do modelo e quanto menor for o resultado do cálculo, melhor o ajuste do modelo.

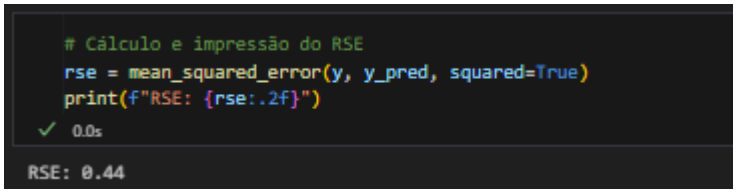

Figura 8: Cálculo e impressão do Residual Standart Error.

Apresentando RSE = 0.44, interpretamos que os resíduos do modelo de regressão estão desviando em torno de 0.44 na pontuação da competição.

1.4.1.3. Mean Absolut Error ou MAE

O *Mean Absolut Error* ou MAE indica a média dos valores absolutos dos erros, ou seja, calcula a média da diferença absoluta entre os valores previstos e os valores reais. Assim como o RMSE e o RSE, quanto menor for o resultado do cálculo do MAE, melhor o ajuste do modelo.valores reais. Assim como o RMSE e o RSE, quanto menor for o resultado do cálculo do MAE, melhor o ajuste do modelo.

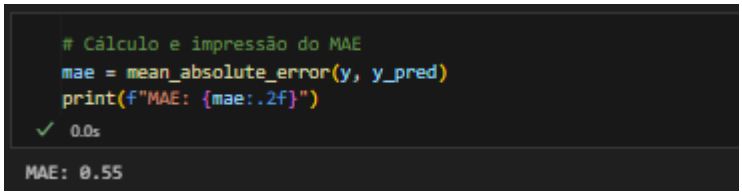

Figura 9: Cálculo e impressão do Mean Absolut Error.

Apresentando 0.55 de resultado, interpretamos que o erro médio absoluto das previsões do modelo é em torno de 0.55 pontos.

1.4.1.4. *R-Squared/R-2* ou coeficiente de determinação

O *R-squared*/R-2, também conhecido como coeficiente de determinação, indica a proporção da soma dos quadrados dos resíduos dividida pela soma dos quadrados totais. O resultado desse cálculo varia entre 0 e 1, onde 0 não explica nenhuma variedade e 1 explica toda a variedade. Portanto quanto mais próximo de 1 for o resultado, melhor o ajuste do modelo.

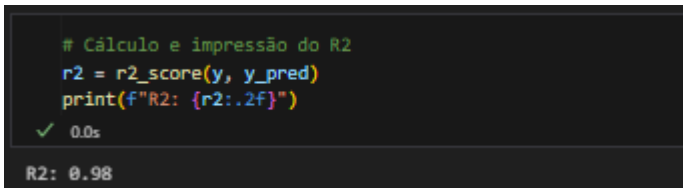

Figura 10: Cálculo e impressão do R-Squared.

Com o valor de 0.98, interpretamos que o modelo explica cerca de 98% da variação da pontuação na competição com base no número de horas de treino.

2. Regressão logística

A Regressão Logística é um modelo de classificação que também faz previsões ou estimativas de um evento acontecer. Usado quando a variável de resposta é binária/categórica, ou seja, apresenta apenas dois resultados possíveis, como 'sim' ou não', 0 ou 1. Então é utilizado para modelar a relação entre uma variável dependente binária (variável de resposta) e um conjunto de variáveis independentes (variáveis preditoras).

1.5. Representação matemática

A função utilizada nessa regressão tem a forma de:

P(y=1|x) = 1 / (1 + e^(-z))

Onde P é a probabilidade de o evento acontecer; y é a classe, podendo ser 1 ou 0; x é o conjunto de atributos; z é a soma ponderada dos atributos.

1.6. Aplicações

Os modelos de classificação são amplamente utilizados em análise de sentimentos em textos, como comentários em redes sociais; na área da saúde, no auxílio em diagnosticar a doença de um paciente; ou na área empresarial como a predição de churn que é para prever a taxa de perda de clientes. O modelo de classificação necessita de dados de treinamento que é um conjunto de dados de exemplo, dados de entrada e suas respectivas saídas, para ensinar o modelo a fazer previsões.

1.7. Exemplo

Para exemplificar a Regressão Logística, considera-se um problema de classificação binária em que será previsto se o câncer de um paciente é benigno ou maligno (variável dependente/*target*) com base em 30 atributos (variáveis independentes): *diagnosis*, radius*_mean*, *texture_mean*, *perimeter_mean*, *area_mean*, *smoothness_mean*, *compactness_mean*, *concavity_mean*, *concave_points_mean*, *symmetry_mean*, *fractal_dimension_mean*, *radius_se*, *texture_se*, *perimeter_se*, *area_se*, *smoothness_se*, *compactness_se*, *concavity_se*, *concave_points_se*, *symmetry_se*, *fractal_dimension_se*, *radius_worst*, *texture_worst*, *perimeter_worst*, *area_worst*, *smoothness_worst*, *compactness_worst*, *concavity_worst*, *concave_points_worst*, *symmetry_worst* e *fractal_dimension_worst*.

1.7.1. Descrição dos atributos

diagnosis: Diagnóstico do câncer, maligno ou benigno.

radius_mean: Distância média do centro da massa aos pontos de perímetro.

texture_mean: Desvio padrão médio dos valores da escala de cinza.

perimeter_mean: Tamanho médio do centro do tumor.

smoothness_mean: Média de variação local em comprimentos de raio.

compactness_mean: Média do perímetro ^ 2 / área – 1.0

concavity_mean: Média da severidade das porções concavas do contorno.

concave_points_mean: Média do número de porções concavas do contorno.

fractal_dimension_mean: Média da *coast approximation"* - 1

radius_se: Erro padrão do *radius_mean*.

texture_se: Erro padrão do *texture_mean.*

perimeter_se: Erro padrão do *perimeter_mean*.

area_se: Erro padrão do *area_mean*.

smoothness_se: Erro padrão do *smoothness_mean*.

compactness_se: Erro padrão do *compactness_mean*.

concavity_se: Erro padrão do *concavity_mean*.

concave_points_se: Erro padrão do *concave_points_mean*.

symmetry_se: Erro padrão do *symmetry_mean*.

fractal_dimension_se: Erro padrão do *fractal_dimension_mean*.

radius_worst: Maior valor médio para *radius_mean*.

texture_worst: Maior/pior valor médio para *texture_mean*.

perimeter_worst: Maior/pior valor médio para *perimeter_mean*.

area_worst: Maior/pior valor médio para *area_mean*.

smoothness_worst: Maior/pior valor médio para *smoothness_mean*.

compactness_worst: Maior/pior valor médio para *compactness_mean*.

concavity_worst: Maior/pior valor médio para *concavity_mean*.

concave_points_worst: Maior/pior valor médio para *concave_points_mean*.

symmetry_worst: Maior/pior valor médio para *symmetry_mean*.

fractal_dimension_worst: Maior/pior valor médio para *fractal_dimension_mean*.

Realizando novamente o exemplo no editor de código Visual Studio Code é necessário baixar as bibliotecas necessárias para criar o *dataframe* que será usado para construir todo o algoritmo.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix, classification_report, roc_curve
```

✓ 0.0s

Figura 11: Importação das bibliotecas.

```
# Carregando os dados
df = load_breast_cancer()
```

✓ 0.0s

Figura 12: Carregando os dados.

1.7.2. Divisão dos dados

É importante realizar uma divisão adequada dos dados entre os conjuntos de treinamento, validação e teste para evitar o sobreajuste (*overfitting*) do modelo aos dados de treinamento e garantir que ele generalize bem para dados não vistos. O conjunto de validação é útil para ajustar hiperparâmetros e fazer seleção de modelo, enquanto o conjunto de teste fornece uma avaliação final do desempenho do modelo antes de aplicá-lo em dados reais.

Primeiramente os dados foram divididos em dois conjuntos: dados de treinamento e dados de teste.

Os dados de treinamento são os dados usados para treinar o modelo de regressão logística. Nesse conjunto é apresentado as variáveis independentes e o *target*.

Os dados de teste são usados para avaliar o desempenho final do modelo, após o treinamento. O modelo é aplicado a esses dados para obter a previsões e analisar a relação com o *target*.

```
# Definindo o Target e separando os dados de Treinamento e Teste
data = df.data
target = df.target
data_train, data_test, target_train, target_test = train_test_split(data, target, test_size=0.3, random_state=42)
✓ 0.0s
```

Figura 13: Definição do Target e separação dos dados de Treino e Teste.

1.7.3. Matriz de confusão

A matriz de confusão é um informativo de desempenho do modelo em um conjunto de dados de testes. Apresenta-se uma contagem classificações corretas incorretas feitas pelo modelo comparando as predições com as classes verdadeiras dos dados.

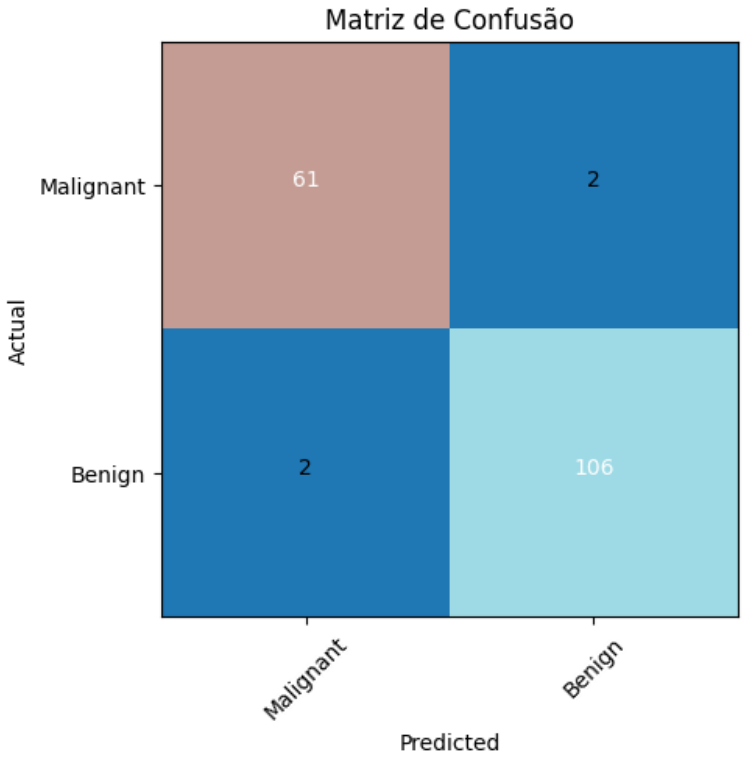

Figura 14: Matriz de confusão.

Foram apresentadas 61 amostras classificadas corretamente como câncer maligno, 2 amostras classificadas incorretamente como câncer benigno, 106 amostras classificadas corretamente como câncer benigno e 2 amostras classificadas incorretamente como câncer maligno.

1.7.4. Métricas de avaliação

As métricas de avaliação de modelos de aprendizado de máquina são utilizadas para medir a qualidade e o desempenho dos modelos em relação aos dados de teste. A mais comuns em serem usadas são *accuracy, precision, recall, F1-score* e ROC-AUC.

1.7.4.1. *Accuracy*

A Acurácia/*Accuracy* apresenta uma performance geral do modelo, apontando quantas classificações foram feitas corretamente ou incorretamente. É a proporção de previsões verdadeiras em relação ao todo. Entretanto, caso os dados estejam desequilibrados está métrica pode apresentar resultados não confiáveis.

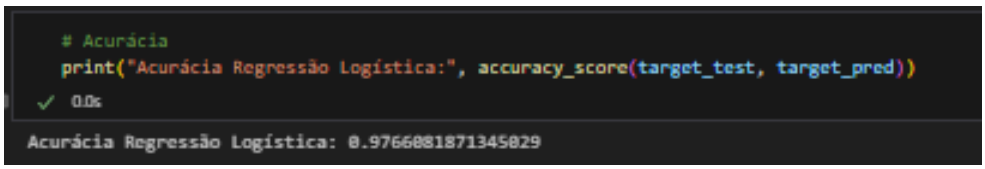

Figura 15: Resultado da acurácia.

Com a acurácia em 0.97, entende-se que o modelo classificou corretamente 97% das amostras em relação ao todo.

1.7.4.2. Precision

A Precisão/*Precision* apresenta uma performance dentre as classificações da classe Positivo, ou seja, apresenta a quantidade de classificações feitas corretamente apenas com base nos exemplos previstos como positivos. Útil para evitar classificar incorretamente exemplos negativos como positivos.

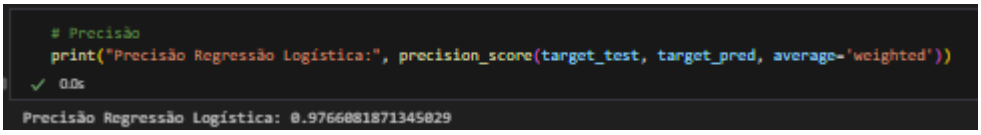

Figura 16: Resultado da precisão.

Com a precisão em 0.97, indica que 97% das amostras classificadas como positivas pelo modelo são realmente positivas.

1.7.4.3. *Recall*

A Revocação/*Recall* apresenta a proporção de exemplos positivos previstos corretamente em relação ao total de exemplos reais positivos. Útil para evitar de deixar de classificar exemplos positivos como positivos.

```
# Revocação
print("Revocação Regressão Logística:", recall_score(target_test, target_pred, average='weighted'))
✓ 0.0s
Revocação Regressão Logística: 0.9766081871345029
```

Figura 17: Resultado da revocação.

Com a revocação em 0.97, significa que o modelo identificou corretamente 97% das amostras positivas em relação ao total de amostras positivas reais

1.7.4.4. F1-score

O F1-*score* é uma combinação da Precisão e revocação de maneira equilibrada que fornece uma medida geral de equilíbrio entre as métricas.

```
# Medida F1-score
print("Medida F1_score Regressão Logística:", f1_score(target_test, target_pred, average='weighted'))
✓ 0.0s
Medida F1_score Regressão Logística: 0.9766081871345029
```

Figura 18: Resultado do F1-Score.

Um valor de 0.97 indica um bom equilíbrio entre a capacidade do modelo de classificar corretamente as amostras positivas e a capacidade de detectar corretamente os casos positivos.

1.7.4.5. ROC-AUC

O ROC-AUC representa a área sob a curva ROC. A curva ROC fornece uma medida de verdadeiros positivos em relação a taxa de falsos positivos para diferentes limiares de classificação.

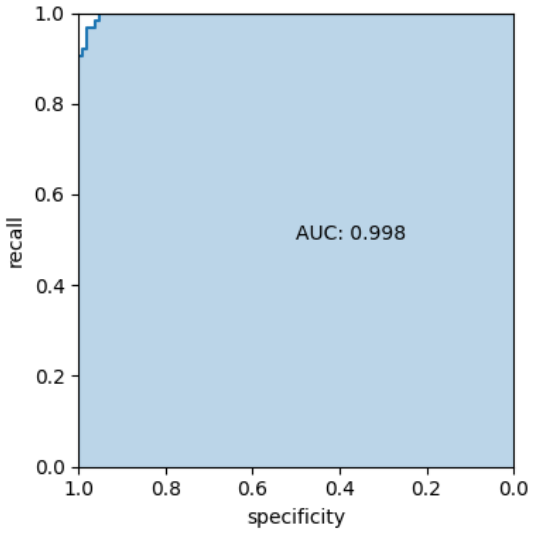

Figura 19: Resultado do ROC-AUC.

Com o valor de 0.998, a métrica sugere que o modelo tem uma excelente performance, tendo uma ótima capacidade de discriminar e classificar corretamente as classes positivas e negativas.

1.7.5. Validação cruzada

Além da divisão tradicional entre dados de treinamento e teste, também é comum utilizar técnicas de validação cruzada. A validação cruzada envolve dividir os dados em conjuntos de treinamento, teste e validação.

```
# Definindo o Target e separando os dados de Treinamento/Validação e Teste/Validação
data = df.data
target = df.target
data_train, data_val_test, target_train, target_val_test = train_test_split(data, target, test_size=0.3, random_state=42)
data_test, data_val, target_test, target_val =train_test_split(data_val_test, target_val_test, test_size=0.5, random_state=42)
✓ 0.0s
```

Figura 20: Separação dos dados em treino/validação e teste/validação.

O conjunto de dados de treinamento são usados para treinar o modelo, o conjunto de validação e teste é usada durante a fase de ajuste e validação do modelo permitindo seleção do melhor modelo como base o desempenho nos dados de validação, o conjunto de teste é usado para avaliar o desempenho final do modelo e o conjunto de validação é usado para ajustar os hiper parâmetros durante a validação do modelo, enquanto o conjunto de validação e teste é usado para avaliar o desempenho do modelo selecionado. E após a separação desses conjuntos acontece a divisão em vários *folds* fazendo diversos treinos e testes. No exemplo, foi realizado com apenas 10 folds.

```
# Criando o modelo, aplicando a validação cruzada 10-fold no modelo
model = LogisticRegression(max_iter=10000, random_state=1032)
score = cross_val_score(model, data_train, target_train, cv=10)
✓ 10.8s
```

Figura 21: Aplicação da validação cruzada de 10 folds no modelo.

As métricas de avaliação realizada na validação cruzada são as mesmas realizada na divisão de dados anterior.

REFERÊNCIA

Christoph Molnar. *Interpretable Machine Learning*. Disponível em: https://christophm.github.io/interpretable-ml-book/. Acesso em: 18 junho 2023.