Published in Parceiro de crescimento IPNET

IPNET Growth Partner
Feb 15, 2021 · 5 min read

A importância da normalização e padronização dos dados em Machine Learning

A Normalização e a padronização são técnicas frequentemente aplicadas na etapa de preparação dos dados, com o objetivo de colocá-los em um intervalo de valores comuns. Não são técnicas obrigatórias, contudo, podem impactar na acurácia (entenda-se, neste artigo, a acurácia como uma forma de avaliar o modelo, independente de que ele seja de classificação ou regressão) do modelo de alguns algoritmos de aprendizado de máquina.

Vale ressaltar que nem todos modelos de Machine Learning precisam de dados em uma escala comum.

. . .

Normalização

A normalização coloca os dados no intervalo entre 0 e 1 ou -1 e 1, caso haja valores negativos, sem distorcer as diferenças nas faixas de valores. Ou seja, ela não retira os outliers (valores extremos).

Mas, por qual motivo as colunas com escalas diferentes afetam a acurácia do modelo?

Para ilustrar a ideia, imagine uma função de regressão linear na qual queremos prever o preço de uma casa baseado no número de cômodos e no tamanho do terreno. Teríamos uma função no formato:

y = x1*w1 + x2*w2

onde,
y: Representa o preço da casa;
x1: Representa o número de cômodos;
w1: Peso referente a variável x1;
x2: Representa o tamanho do terreno (m^2);
w2: Peso referente a variável x2.

O x2, por ser uma variável que indica o tamanho de um terreno, naturalmente, terá um valor maior do que o x1, que representa o número de cômodos de uma casa. Como consequência, o valor dos pesos do modelo terão escalas diferentes, e o mesmo “aprenderá” que uma coluna terá maior relevância para a previsão do que a outra. Porém, o modelo chegará nessa conclusão sob a influência da ordem de grandeza da coluna, e não pela importância da variável em si.

A fórmula matemática abaixo nos permite fazer a normalização dos dados:

$$Xc = \frac{X - Xmin}{Xmax - Xmin}$$

Fórmula Min-Max

. . .

Padronização

A padronização tem a mesma ideia da normalização, isto é, colocar os dados em uma mesma escala. Porém, na padronização, colocamos a média dos dados em 0 e o desvio padrão em 1. Esse algoritmo é melhor utilizado quando a nossa distribuição é Gaussiana.

A fórmula z-score é uma das mais comuns para padronização:

$$z = \frac{x - \mu}{\sigma}$$

Fórmula z-score

. . .

Algumas Considerações

Nem sempre é preciso padronizar ou normalizar os dados que estão em escala diferente. Talvez, colocá-los em uma unidade de medida diferente seja o suficiente. Por exemplo, caso os dados estejam em centímetros, você pode apenas colocá-los na unidade de medida em metros (ou seja, em vez de usar 100cm, use 1m), desde que modifique também a escala das outras colunas. É importante, no entanto, voltar os dados para a escala original quando for preciso analisar ou apresentar os resultados.

É aconselhável testar as mais de uma forma para avaliar qual tem o melhor desempenho e acurácia para o seu problema.

Alguns algoritmos que precisam dos dados na mesma escala: KNN (K-Nearest Neighbours), Redes Neurais, Regressão Linear, Regressão Logística e SVM.

Alguns algoritmos que **não** precisam dos dados na mesma escala: Árvores de Decisão, Random Forest, AdaBoost, Naïve Bayes, etc. (porém, aconselho testar a normalização ou padronização).

. . .

Exemplo prático

Mostrarei um exemplo desenvolvido usando o Google Colab, uma ferramenta gratuita oferecida pelo Google para desenvolver códigos em Python com acesso grátis a uma GPU. Apesar de simples, este exemplo demonstra o impacto da normalização ou padronização na acurácia. Utilizei o dataset Boston House Prices presente na biblioteca do scikit-learn (link). O código completo encontra-se aqui.

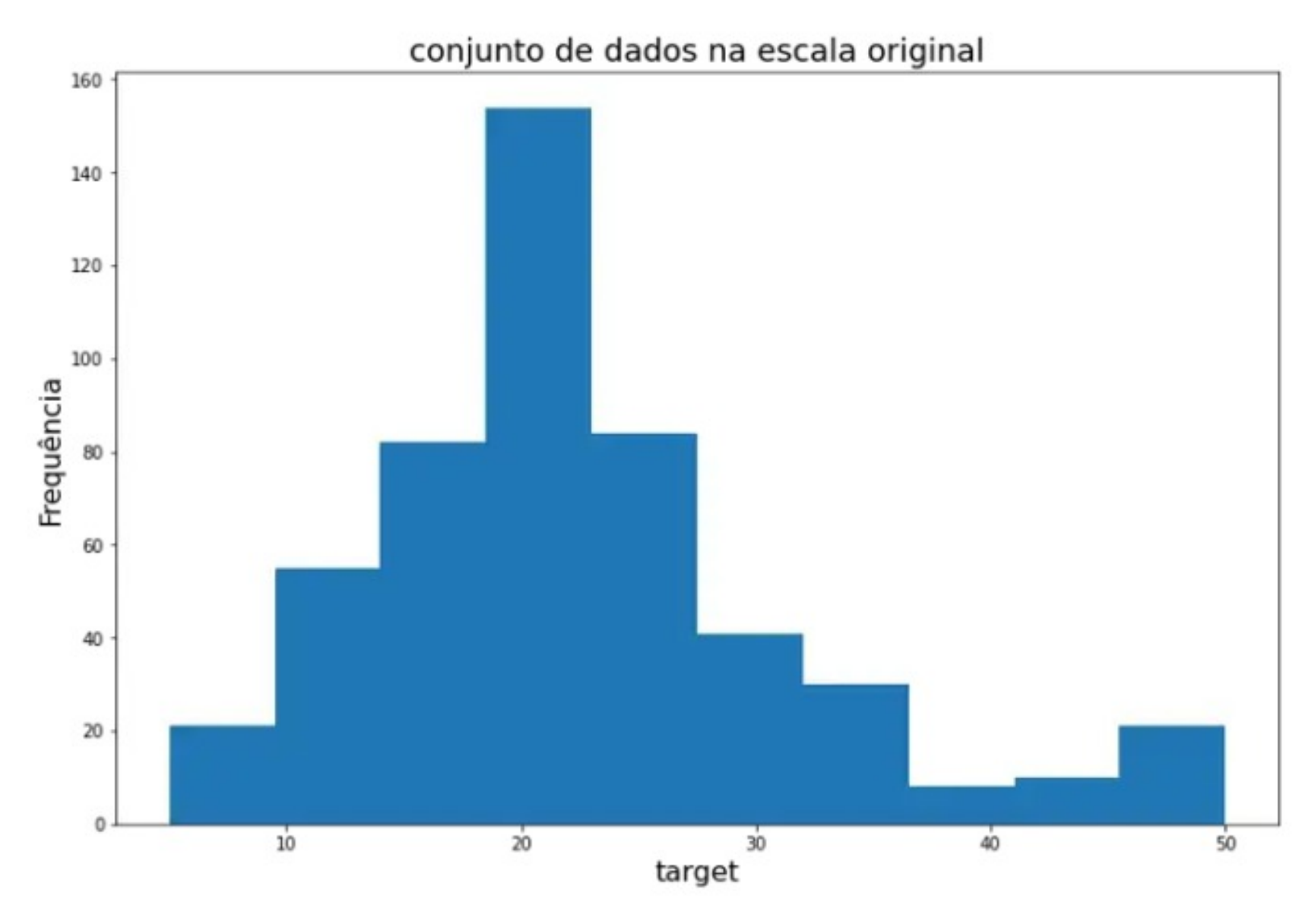
No código abaixo, os dados foram carregados e ajustados em um data frame da biblioteca Pandas.

```
# carregando o dataset "Boston house prices"
data = load_boston()

dataset = pd.DataFrame(data['data'], columns=data['feature_names'])

dataset['target'] = data['target']

# visualizando os primeiros valores
dataset.head()
```



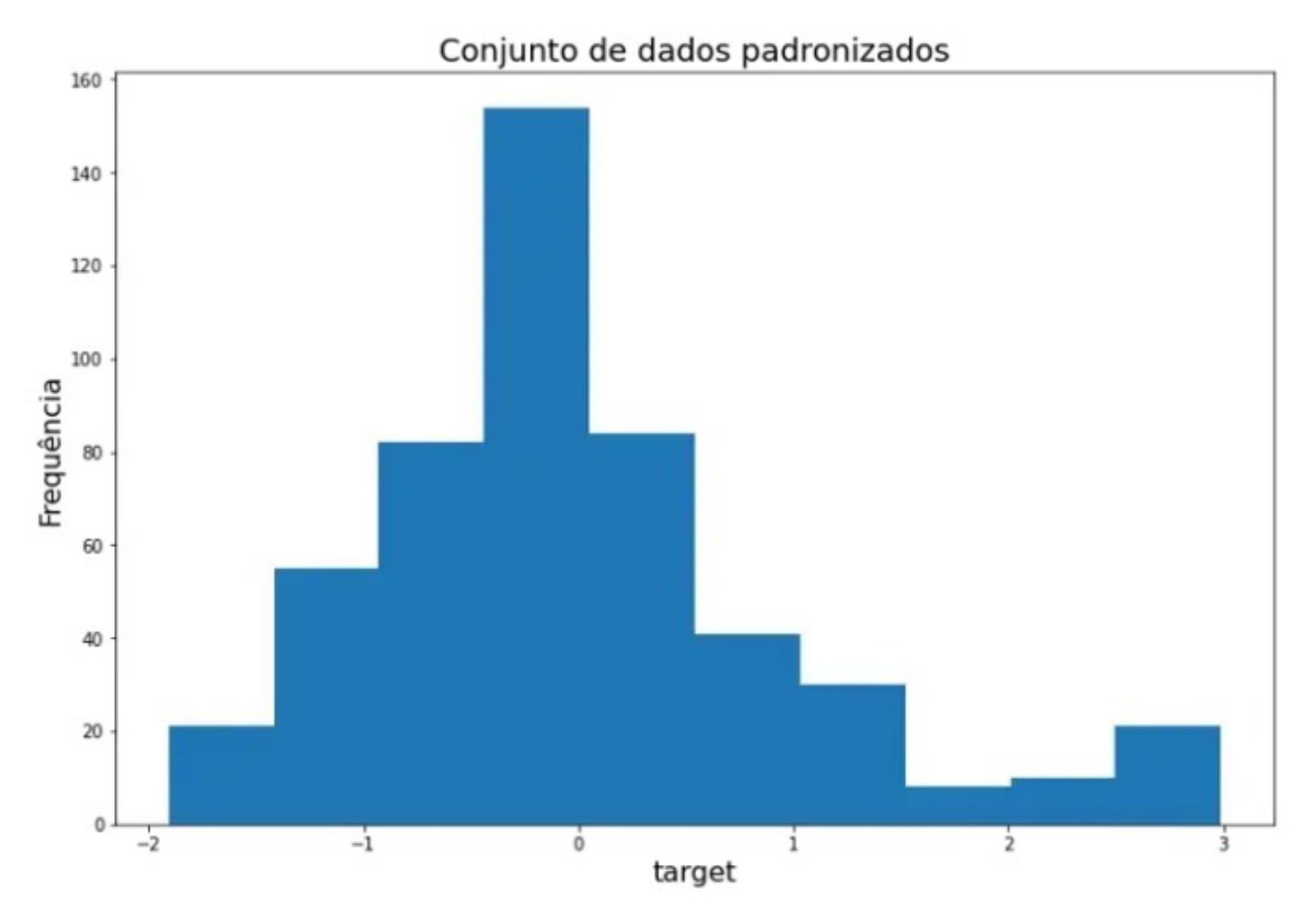
Para padronizar os dados, podemos fazer uso da biblioteca StandardScaler do Python.

```
colunas = dataset.columns

# Padronizando os dados
from sklearn.preprocessing import StandardScaler
scaler_standard = StandardScaler()

dataset_padronizado =
pd.DataFrame(scaler_standard.fit_transform(dataset), columns=colunas)

dataset_padronizado.head()
```



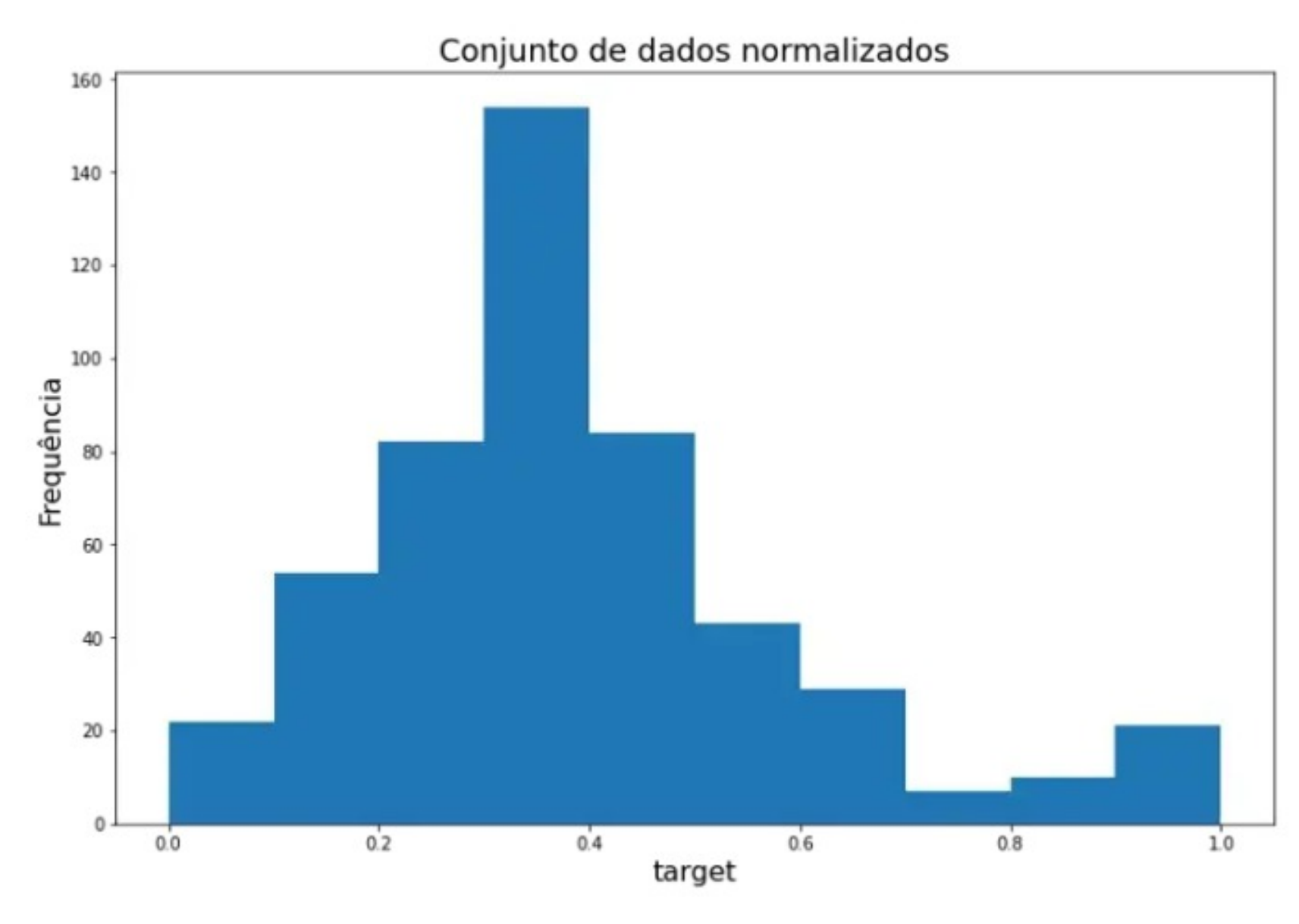
Já para normalizar os dados, usamos a biblioteca MinMaxScaler.

```
# normalizando os dados
from sklearn.preprocessing import MinMaxScaler

scaler_minMax = MinMaxScaler()

data_normalize = pd.DataFrame(scaler_minMax.fit_transform(dataset),
columns=colunas)

data_normalize.head()
```



Perceba que os dados mudaram de intervalo de acordo com a coluna x, e note que a distribuição dos dados não mudou. Iremos definir uma função para treinar e imprimir coeficiente de determinação (r^2). Finalmente, vamos avaliar a acurácia que conseguimos utilizando o algoritmo de aprendizado de máquina *support-vector machine*.

```
def ML_step(data):
    dataset_num_columns = data.shape[1]
    x = data.iloc[:, 0:dataset_num_columns-1]
    y = data['target']

    # separando em dados de treino e teste
    X_train, X_test, y_train, y_test = train_test_split(x, y,
test_size=0.30, random_state=12)

    # instanciando o modelo Support Vector Regression
    model = SVR()

    # treinando o modelo
    model.fit(X_train, y_train)

    # prevendo os valores
    y_predict = model.predict(X_test)

    # avaliando o modelo (quanto mais próximo de 1 melhor)
    print("r^2:", r2_score(y_test, y_predict))
```

```
print("Dataset não padronizado")
ML_step(dataset)
```

Dataset não padronizado
r^2: 0.1604256843627555

```
print("Dataset padronizado")
ML_step(dataset_padronizado)
```

Dataset padronizado
r^2: 0.8566091790160586

```
print("Dataset normalizado")
ML_step(data_normalize)
```

Dataset normalizado
r^2: 0.8355270748255403

Quanto mais próximo de 1 o valor do r^2 estiver, mais eficiente é a acurácia do modelo. Nota-se uma expressiva diferença entre o r^2 dos dados padronizados e não-padronizados, o que evidencia a importância da padronização e normalização.

Este artigo foi útil? Deixe suas dúvidas e sugestões nos comentários.

Autor: Breno Dutra

. . .

Referências:

How, When, and Why Should You Normalize / Standardize / Rescale Your Data?
Author(s): Swetha Lakshmanan Before diving into this topic, lets first start with some definitions.
towardsai.net

Normalizar ou padronizar as variáveis?
Um dos processos muito rotineiros para um Data Science é "colocar as variáveis na mesma página". Mas quando devemos...
medium.com

Standardization VS Normalization
Standardization
VS Normalization Standardizationmedium.com