

Aprendizado de máquina com SKLearn

Disciplina: Programação aplicada à engenharia cartográfica

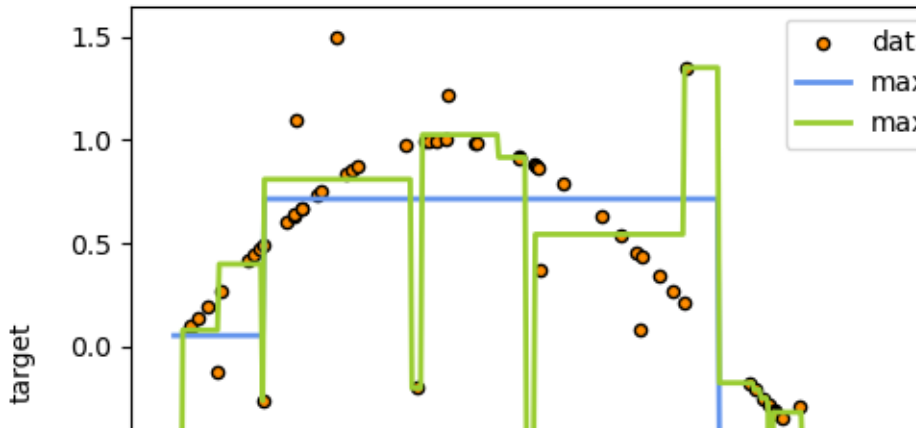
Maurício C. M. de Paulo - D.Sc.

25 de fevereiro de 2026

Árvores de decisão

<https://scikit-learn.org/stable/modules/tree.html>

Decision Tree Regression



Código Python

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt

# Step 1: Generate synthetic data
np.random.seed(42)
n_samples = 100
X = np.random.rand(n_samples, 1) * 10
y = np.sin(X).ravel() + np.random.normal(0, 0.2, size=n_samples) # non-linear pattern

# Step 2: Create a pandas DataFrame
df = pd.DataFrame({'Feature': X.flatten(), 'Target': y})
```

Código Python

```
# Step 3: Train-test split
X_train, X_test, y_train, y_test = train_test_split(df[['Feature']], df['Target'],

# Step 4: Fit a Decision Tree Regressor
tree = DecisionTreeRegressor(max_depth=4, random_state=42)
tree.fit(X_train, y_train)

# Step 5: Make predictions
y_pred = tree.predict(X_test)

# Step 6: Evaluation
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse:.3f}")
print(f"R-squared: {r2:.3f}")
```

Código Python

```
# Step 7: Plotting predictions
X_plot = np.linspace(0, 10, 500).reshape(-1, 1)
y_plot = tree.predict(X_plot)

plt.scatter(X, y, s=20, edgecolor="black", c="darkorange", label="Data")
plt.plot(X_plot, y_plot, color="cornflowerblue", linewidth=2, label="Model")
plt.xlabel("Feature")
plt.ylabel("Target")
plt.title("Decision Tree Regression")
plt.legend()
plt.show()
```