

# Aprendizado de máquina com SKLearn

## Disciplina: Programação aplicada à engenharia cartográfica

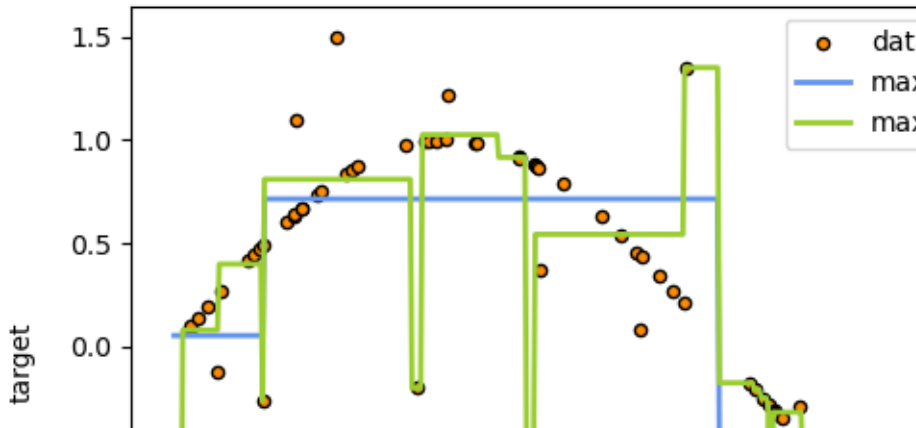
Maurício C. M. de Paulo - D.Sc.

6 de fevereiro de 2026

# Árvores de decisão

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

Decision Tree Regression



# Código Python

```
1 import numpy as np
2 import pandas as pd
3 from sklearn.model_selection import train_test_split
4 from sklearn.tree import DecisionTreeRegressor
5 from sklearn.metrics import mean_squared_error, r2_score
6 import matplotlib.pyplot as plt
7
8 # Step 1: Generate synthetic data
9 np.random.seed(42)
10 n_samples = 100
11 X = np.random.rand(n_samples, 1) * 10
12 y = np.sin(X).ravel() + np.random.normal(0, 0.2, size=n_samples)
13
14 # Step 2: Create a pandas DataFrame
15 df = pd.DataFrame({'Feature': X.flatten(), 'Target': y})
```

```
1  # Step 3: Train-test split
2  X_train, X_test, y_train, y_test = train_test_split(df[['Feature']],
3
4  # Step 4: Fit a Decision Tree Regressor
5  tree = DecisionTreeRegressor(max_depth=4, random_state=42)
6  tree.fit(X_train, y_train)
7
8  # Step 5: Make predictions
9  y_pred = tree.predict(X_test)
10
11 # Step 6: Evaluation
12 mse = mean_squared_error(y_test, y_pred)
13 r2 = r2_score(y_test, y_pred)
14
15 print(f"Mean Squared Error: {mse:.3f}")
16 print(f"R-squared: {r2:.3f}")
```

```
1  # Step 7: Plotting predictions
2  X_plot = np.linspace(0, 10, 500).reshape(-1, 1)
3  y_plot = tree.predict(X_plot)
4
5  plt.scatter(X, y, s=20, edgecolor="black", c="darkorange", label="Data")
6  plt.plot(X_plot, y_plot, color="cornflowerblue", linewidth=2, label="Predictions")
7  plt.xlabel("Feature")
8  plt.ylabel("Target")
9  plt.title("Decision Tree Regression")
10 plt.legend()
11 plt.show()
```