

# 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>

```
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="D
6  plt.plot(X_plot,_y_plot,_color="cornflowerblue",_linewidth=2,_label
7  plt.xlabel("Feature")
8  plt.ylabel("Target")
9  plt.title("Decision_Tree_Regression")
10 plt.legend()
11 plt.show()
```