

# Untitled

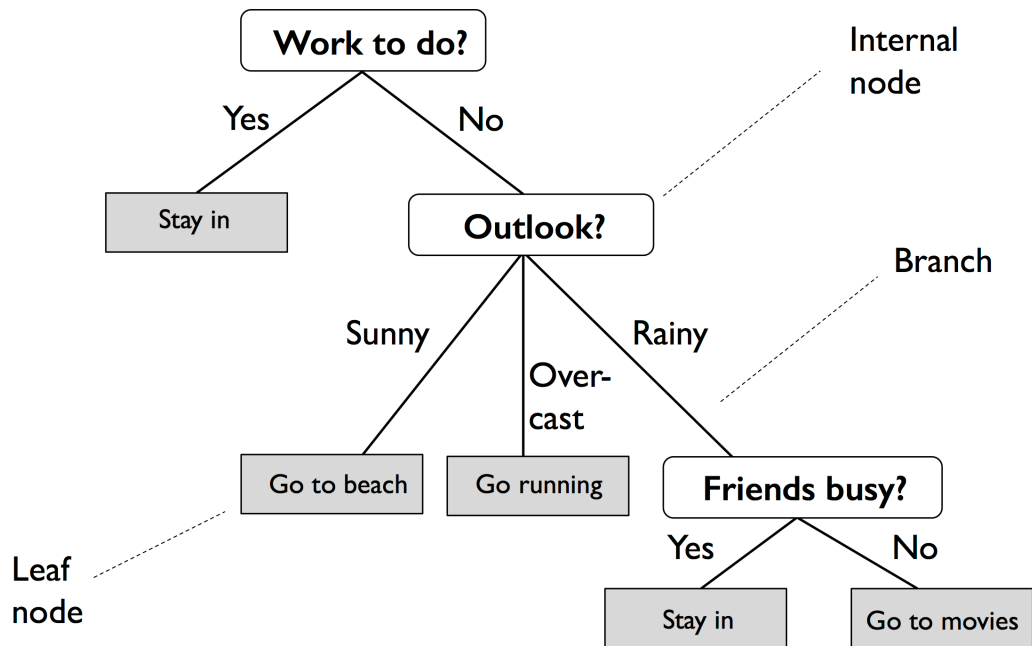
October 25, 2022

## 1 Arvore de Decisão

```
[1]: from IPython.display import Image  
     %matplotlib inline
```

```
[2]: Image(filename='images/03_17.png')
```

[2]:



```
[3]: from sklearn import datasets  
     import numpy as np  
  
     iris = datasets.load_iris()  
     X=iris.data[:, [2,3]]  
     y= iris.target
```

```
[15]: from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(  
    X, y, test_size=0.3, random_state=1, stratify=y)
```

```
[16]: from sklearn.preprocessing import StandardScaler
```

```
sc=StandardScaler()  
sc.fit(X_train)  
X_train_std = sc.transform(X_train)  
X_test_std = sc.transform(X_test)
```

```
[18]: X_combined_std = np.vstack((X_train_std, X_test_std))  
y_combined = np.hstack((y_train, y_test))
```

```
[19]: pip install mlxtend
```

```
Defaulting to user installation because normal site-packages is not writeable  
Requirement already satisfied: mlxtend in  
/home/hefesto/.local/lib/python3.10/site-packages (0.21.0)  
Requirement already satisfied: pandas>=0.24.2 in  
/home/hefesto/.local/lib/python3.10/site-packages (from mlxtend) (1.4.3)  
Requirement already satisfied: joblib>=0.13.2 in  
/home/hefesto/.local/lib/python3.10/site-packages (from mlxtend) (1.1.0)  
Requirement already satisfied: scikit-learn>=1.0.2 in  
/home/hefesto/.local/lib/python3.10/site-packages (from mlxtend) (1.1.2)  
Requirement already satisfied: setuptools in /usr/lib/python3/dist-packages  
(from mlxtend) (59.6.0)  
Requirement already satisfied: scipy>=1.2.1 in /usr/lib/python3/dist-packages  
(from mlxtend) (1.8.1)  
Requirement already satisfied: matplotlib>=3.0.0 in /usr/lib/python3/dist-  
packages (from mlxtend) (3.5.2)  
Requirement already satisfied: numpy>=1.16.2 in /usr/lib/python3/dist-packages  
(from mlxtend) (1.21.5)  
Requirement already satisfied: python-dateutil>=2.8.1 in  
/usr/local/lib/python3.10/dist-packages (from pandas>=0.24.2->mlxtend) (2.8.2)  
Requirement already satisfied: pytz>=2020.1 in /usr/lib/python3/dist-packages  
(from pandas>=0.24.2->mlxtend) (2022.4)  
Requirement already satisfied: threadpoolctl>=2.0.0 in  
/home/hefesto/.local/lib/python3.10/site-packages (from scikit-  
learn>=1.0.2->mlxtend) (3.1.0)  
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from  
python-dateutil>=2.8.1->pandas>=0.24.2->mlxtend) (1.16.0)  
Note: you may need to restart the kernel to use updated packages.
```

```
[27]: from sklearn.tree import DecisionTreeClassifier  
from mlxtend.plotting import plot_decision_regions  
import matplotlib.pyplot as plt
```

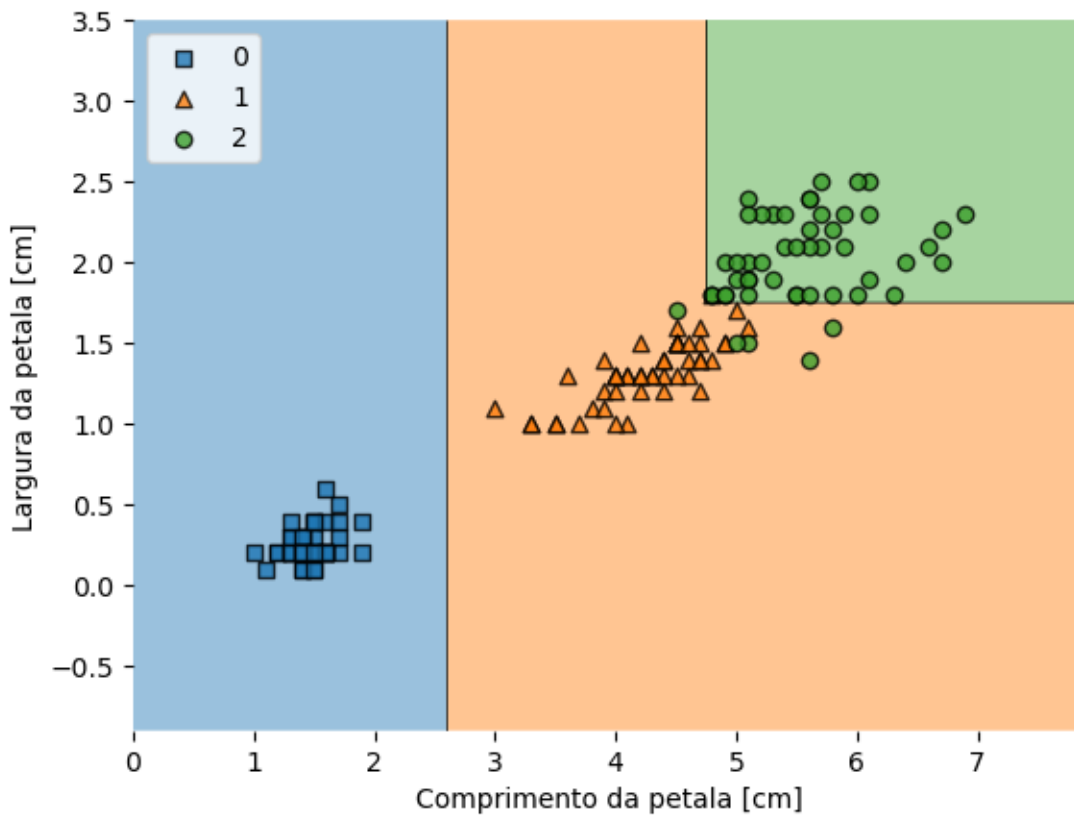
```

modelo_Arvore = DecisionTreeClassifier(
    criterion='gini', max_depth= 3, random_state=36)
modelo_Arvore.fit(X_train, y_train)
X_combined = np.vstack((X_train, X_test))
y_combined = np.hstack((y_train, y_test))

plot_decision_regions(X_combined, y_combined,
                      clf=modelo_Arvore, legend=2)
plt.xlabel('Comprimento da petala [cm]')
plt.ylabel('Largura da petala [cm]')

plt.legend(loc = 'upper left')
plt.show()

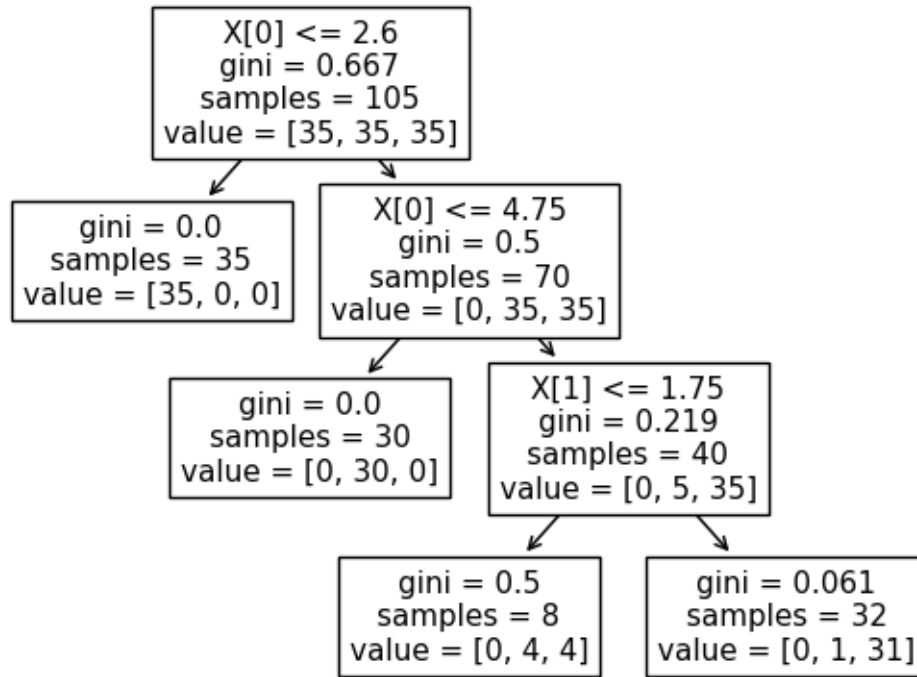
```



```

[25]: from sklearn import tree
      tree.plot_tree(modelo_Arvore)
      plt.show()

```



## 2 Floresta Aleatória

Random Florest

```
[28]: import pandas as pd
import numpy as np
wine = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/
↳wine/wine.data',
                  header=None)
wine.columns=['Labels','Alcohol','Malic acid','Ash','Alcalinity of_
↳ash','Magnesium',
              'Total phenols','Flavanoids',
              'Nonflavanoid phenols','Proanthocyanins','Color intensity',
              'Hue','OD280/OD315 of diluted wines', 'Proline']

print('Labels', np.unique(wine['Labels']))
```

Labels [1 2 3]

```
[29]: wine.head()
```

```
[29]:   Labels  Alcohol  Malic acid  Ash  Alcalinity of ash  Magnesium \
0        1    14.23         1.71  2.43                15.6        127
```

1	1	13.20	1.78	2.14	11.2	100
2	1	13.16	2.36	2.67	18.6	101
3	1	14.37	1.95	2.50	16.8	113
4	1	13.24	2.59	2.87	21.0	118

	Total phenols	Flavanoids	Nonflavanoid phenols	Proanthocyanins	\
0	2.80	3.06	0.28	2.29	
1	2.65	2.76	0.26	1.28	
2	2.80	3.24	0.30	2.81	
3	3.85	3.49	0.24	2.18	
4	2.80	2.69	0.39	1.82	

	Color intensity	Hue	OD280/OD315 of diluted wines	Proline
0	5.64	1.04	3.92	1065
1	4.38	1.05	3.40	1050
2	5.68	1.03	3.17	1185
3	7.80	0.86	3.45	1480
4	4.32	1.04	2.93	735

```
[30]: from sklearn.model_selection import train_test_split

X, y = wine.iloc[:, 1:].values, wine.iloc[:, 0].values

X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.3,
↳random_state=1, stratify=y)
```

```
[31]: from sklearn.ensemble import RandomForestClassifier

feat_labels= wine.columns[1:]

floresta = RandomForestClassifier(n_estimators=500, random_state=1)
↳#n_estimators número de arvores da floresta
```

```
[32]: floresta.fit(X_train, y_train)
```

```
[32]: RandomForestClassifier(n_estimators=500, random_state=1)
```

```
[34]: importancia = floresta.feature_importances_

indices = np.argsort(importancia)[::-1]
```

```
[36]: importancia
```

```
[36]: array([0.13822434, 0.03085574, 0.00873805, 0.0299998 , 0.02513494,
0.05823566, 0.14612287, 0.01154759, 0.0257134 , 0.15315771,
0.07752497, 0.114818 , 0.17992694])
```

```
[37]: feat_labels
```

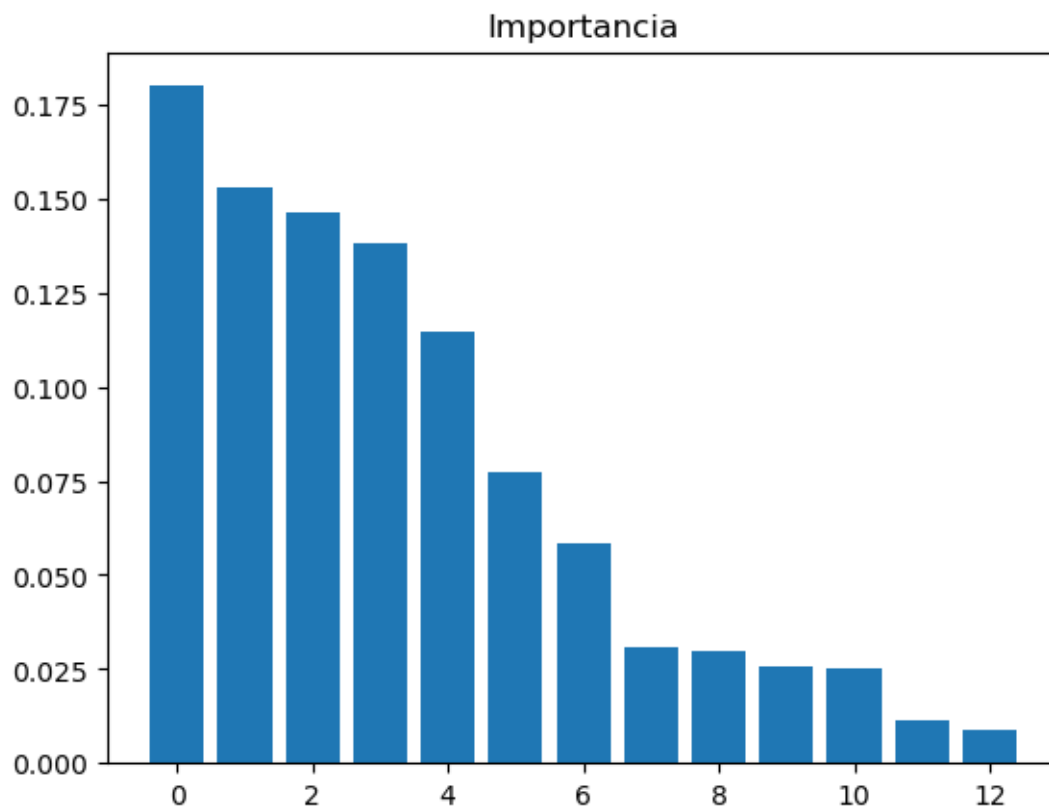
```
[37]: Index(['Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash', 'Magnesium',
          'Total phenols', 'Flavanoids', 'Nonflavanoid phenols',
          'Proanthocyanins', 'Color intensity', 'Hue',
          'OD280/OD315 of diluted wines', 'Proline'],
          dtype='object')
```

```
[39]: for f in range (X_train.shape[1]):
        print("%2d %-*s %f" %(f+1, 30, feat_labels[indices[f]], importancia[f]))
```

```
plt.title('Importancia')
plt.bar(range(X_train.shape[1]), importancia[indices], align='center')
```

1) Proline	0.138224
2) Color intensity	0.030856
3) Flavanoids	0.008738
4) Alcohol	0.030000
5) OD280/OD315 of diluted wines	0.025135
6) Hue	0.058236
7) Total phenols	0.146123
8) Malic acid	0.011548
9) Alcalinity of ash	0.025713
10) Proanthocyanins	0.153158
11) Magnesium	0.077525
12) Nonflavanoid phenols	0.114818
13) Ash	0.179927

```
[39]: <BarContainer object of 13 artists>
```



```
[38]: from sklearn.feature_selection import SelectFromModel

sfm = SelectFromModel(floresta, threshold=0.1, prefit=True)
X_selected = sfm.transform(X_train)
print('numero de recursos que atendem ao limite', X_selected.shape[1])
```

numero de recursos que atendem ao limite 5