

# Review\_decision\_Tree

## 1 Review Decision Tree

COSC 3337: Data Science I

Instructor: Nouhad Rizk

```
[2]: from sklearn import datasets
import numpy as np

iris = datasets.load_iris()
X = iris.data[:, [2, 3]]
y = iris.target

print('Class labels:', np.unique(y))
```

Class labels: [0 1 2]

```
[3]: 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)
```

```
[4]: print('Labels counts in y:', np.bincount(y))
print('Labels counts in y_train:', np.bincount(y_train))
print('Labels counts in y_test:', np.bincount(y_test))
```

Labels counts in y: [50 50 50]

Labels counts in y\_train: [35 35 35]

Labels counts in y\_test: [15 15 15]

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

tree = DecisionTreeClassifier(criterion='entropy',
```

```

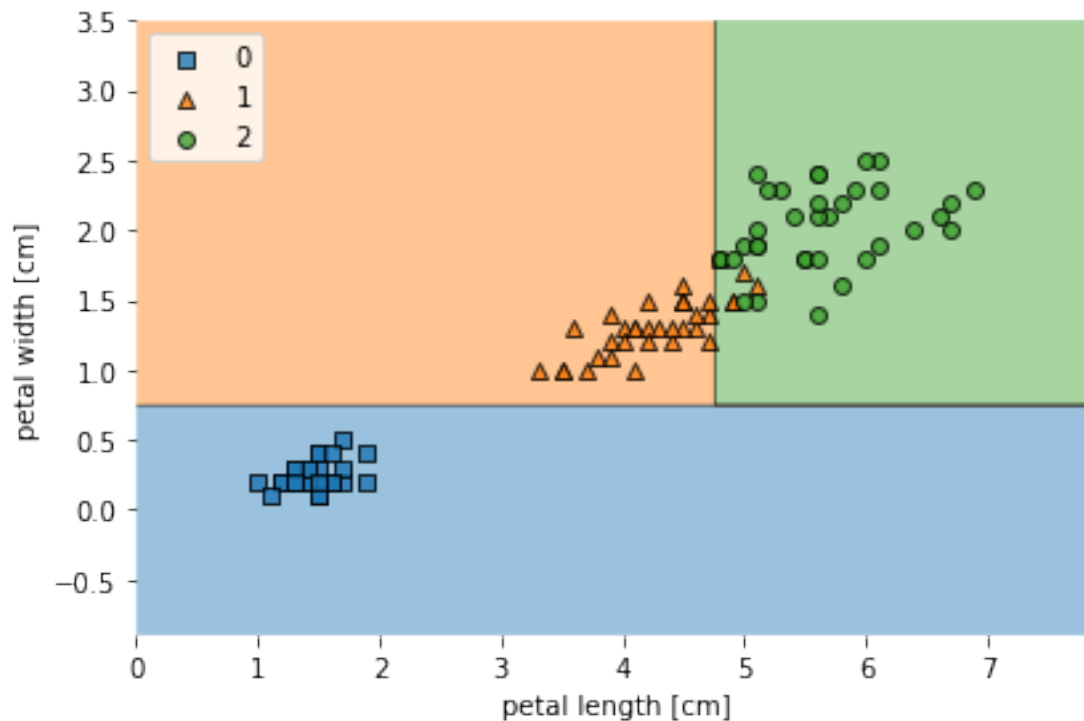
max_depth=2,
random_state=1)

tree.fit(X_train, y_train)

plot_decision_regions(X_train, y_train, tree)

plt.xlabel('petal length [cm]')
plt.ylabel('petal width [cm]')
plt.legend(loc='upper left')
plt.tight_layout()
plt.savefig('images/sklearn_tree.png', dpi=300)
plt.savefig('images/sklearn_tree.pdf')
plt.show()

```



```

[6]: from pydotplus import graph_from_dot_data
from sklearn.tree import export_graphviz

dot_data = export_graphviz(tree,
                           filled=True,
                           rounded=True,
                           class_names=['Setosa'],

```

```

        'Versicolor',
        'Virginica'],
        feature_names=['petal length',
                        'petal width'],
        out_file=None)
graph = graph_from_dot_data(dot_data)
graph.write_png('images/tree.png')

```

[6]: True

[7]: `from IPython.display import Image`

`Image('images/tree.png')`

[7]:

