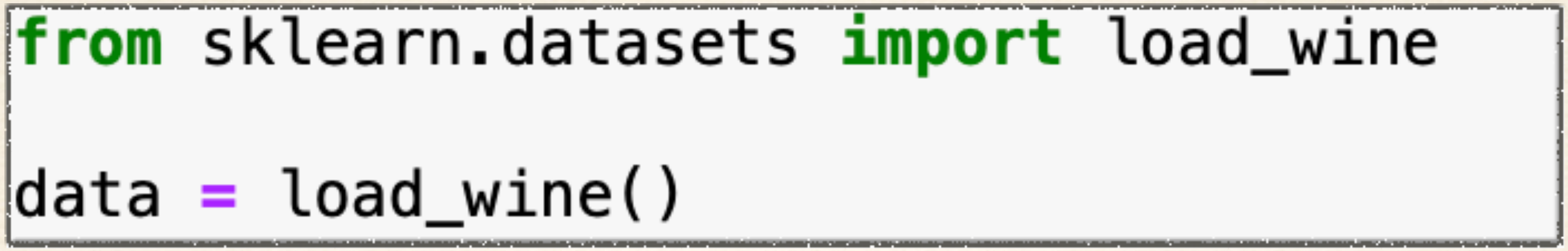
自主練習

- 導入葡萄酒數據集



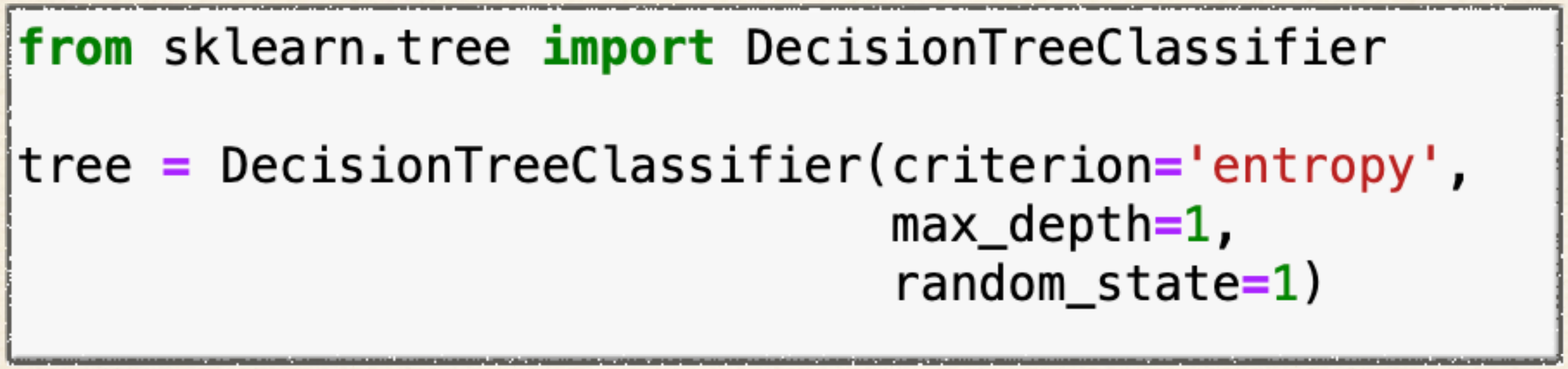
```
from sklearn.datasets import load_wine

data = load_wine()
```

  - 只考慮類別1跟類別2，以及其中的兩個特徵「酒精」(Alcohol) 和「稀釋葡萄酒的OD280/OD315」(OD280/OD315 of diluted wines)

- 分割數據集 (train: test = 8 : 2)

自主練習

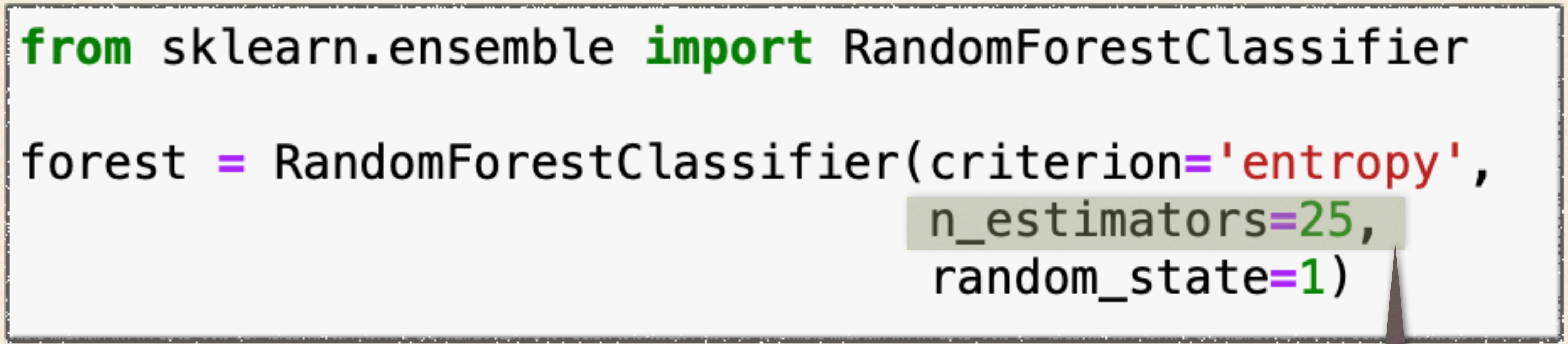
- 訓練「決策樹」模型



```
from sklearn.tree import DecisionTreeClassifier

tree = DecisionTreeClassifier(criterion='entropy',
                              max_depth=1,
                              random_state=1)
```

自主練習

- 訓練「隨機森林」模型

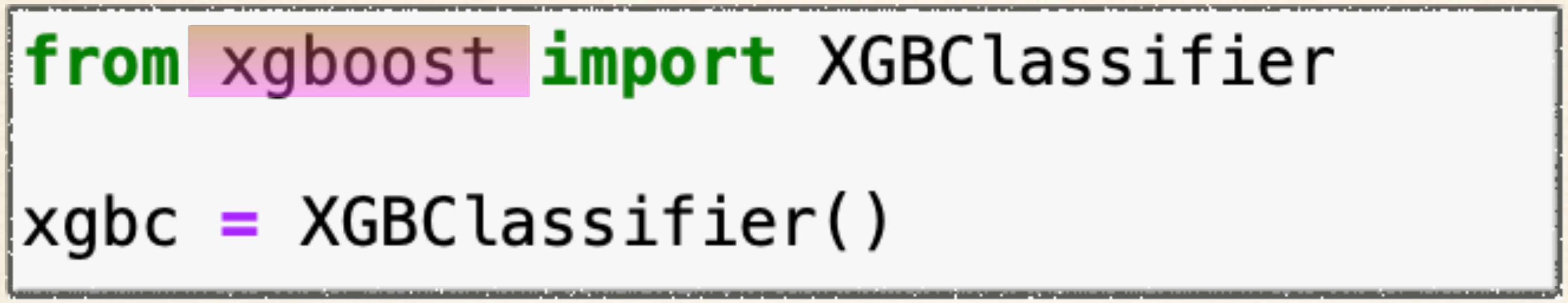


```
from sklearn.ensemble import RandomForestClassifier

forest = RandomForestClassifier(criterion='entropy',
                                n_estimators=25,
                                random_state=1)
```

隨機森林裡有25棵決策樹

自主練習

- 訓練「XGBoost」模型



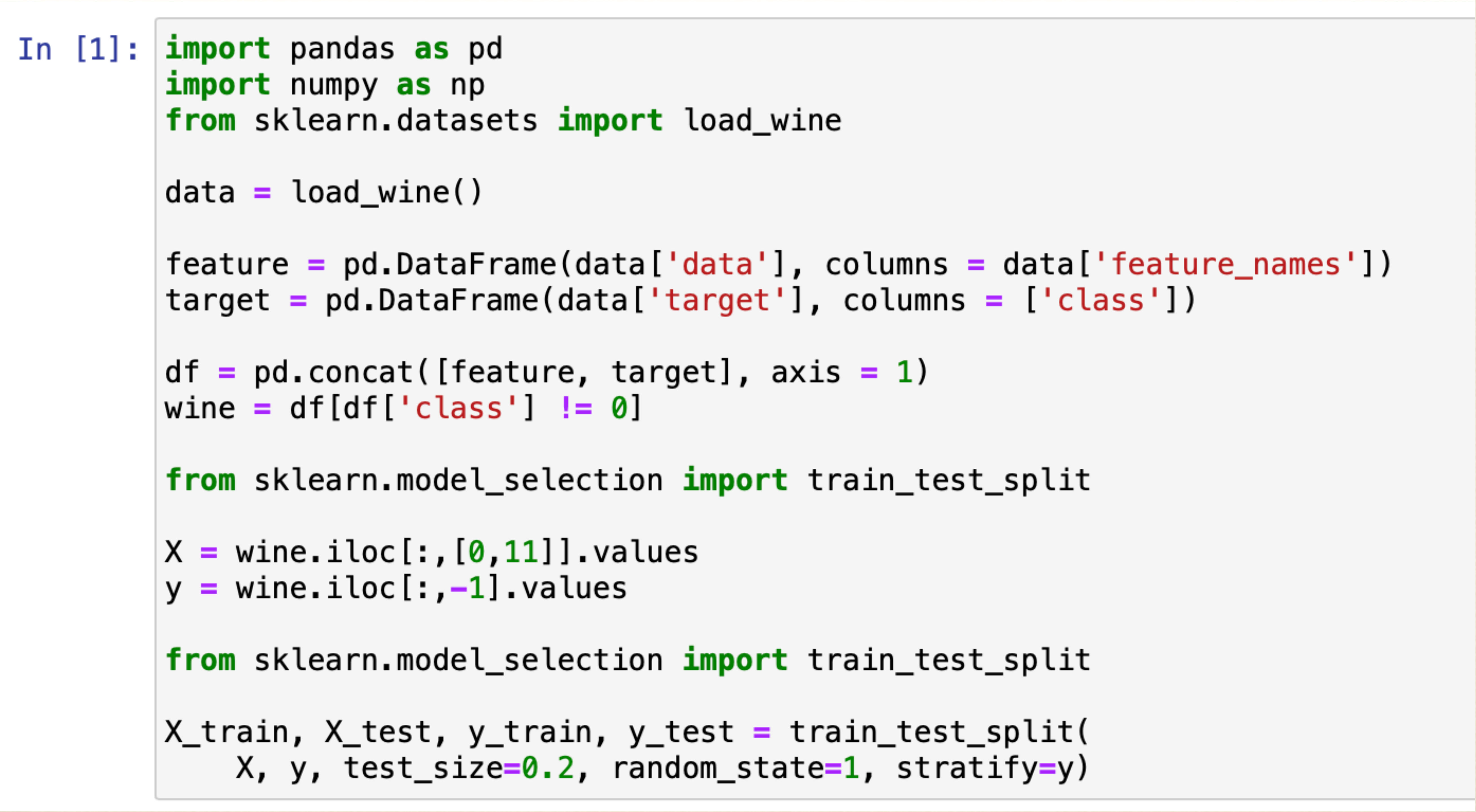
```
from xgboost import XGBClassifier

xgbc = XGBClassifier()
```

XGBoost 是 GBDT 演算法的一種工程實作，同時進行若干改進，如加入正則項、自動學習缺失值的處理策略等，因此於Kaggle 競賽及其他機器學習專案，普遍能取得不錯的成績

自主練習

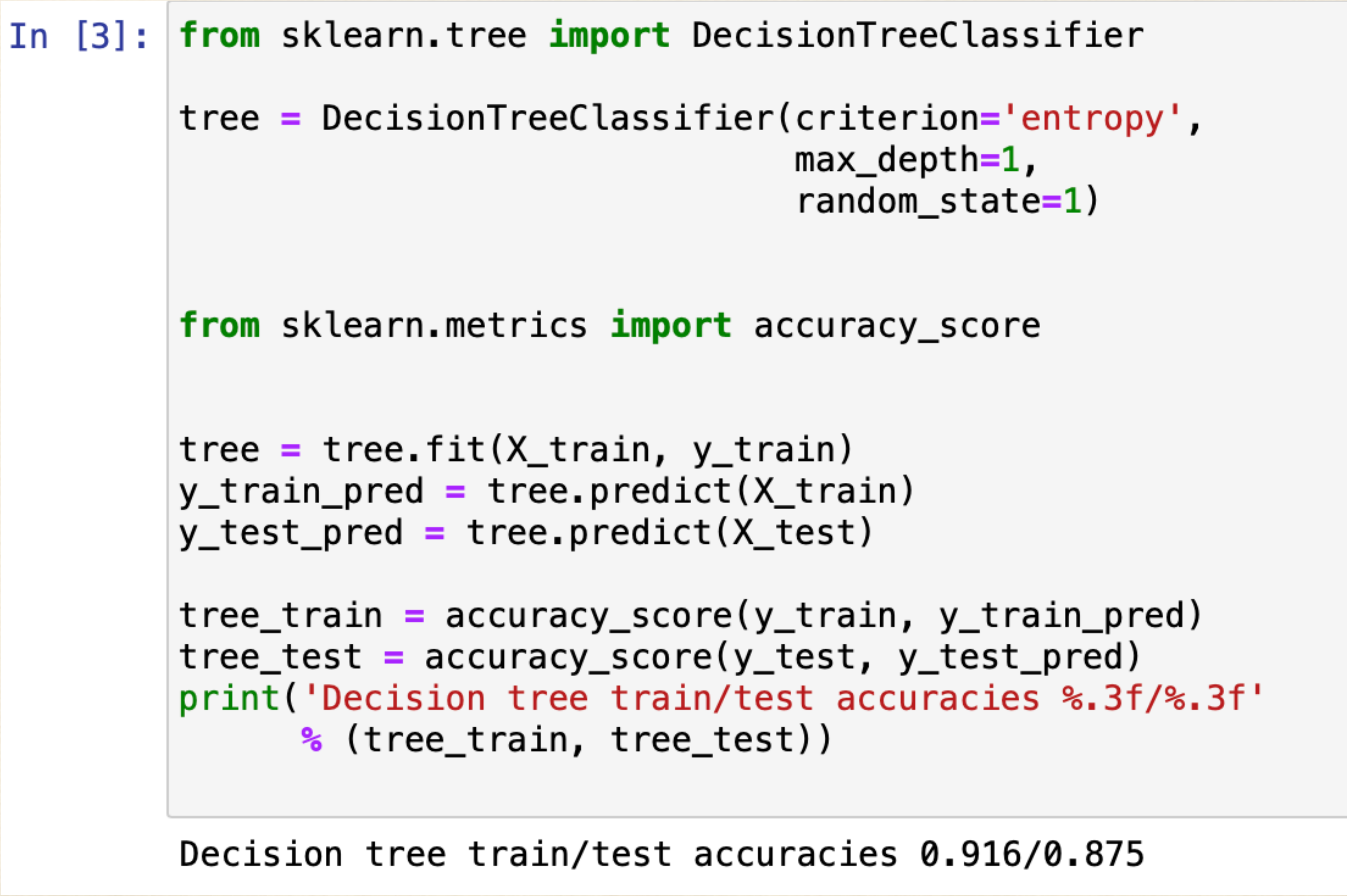
- 畫出各個不同演算法的決策區域圖並算出正確率
- 比較各個決策區域圖
- 試著改變不同的參數，看看決策區域圖會有什麼變化

自主練習解答



```python
import pandas as pd
import numpy as np
from sklearn.datasets import load_wine

data = load_wine()

feature = pd.DataFrame(data['data'], columns = data['feature_names'])
target = pd.DataFrame(data['target'], columns = ['class'])

df = pd.concat([feature, target], axis = 1)
wine = df[df['class'] != 0]

from sklearn.model_selection import train_test_split

X = wine.iloc[:,[0,11]].values
y = wine.iloc[:,-1].values

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=1, stratify=y)
```

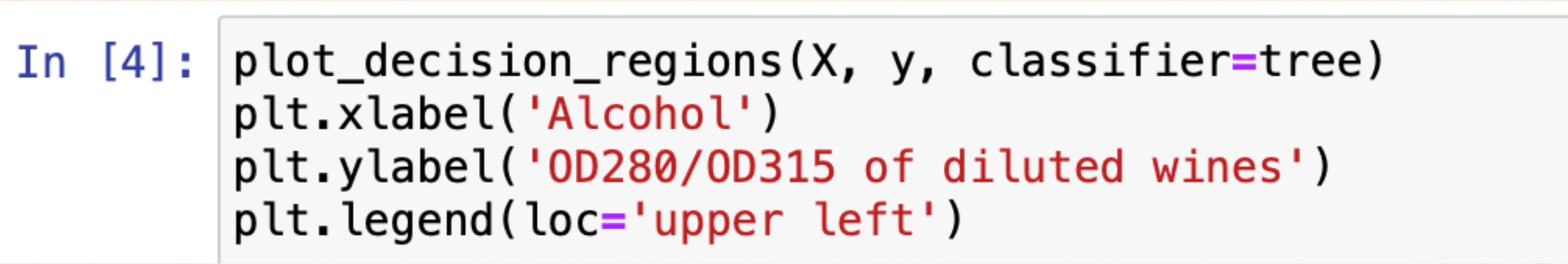
自主練習解答



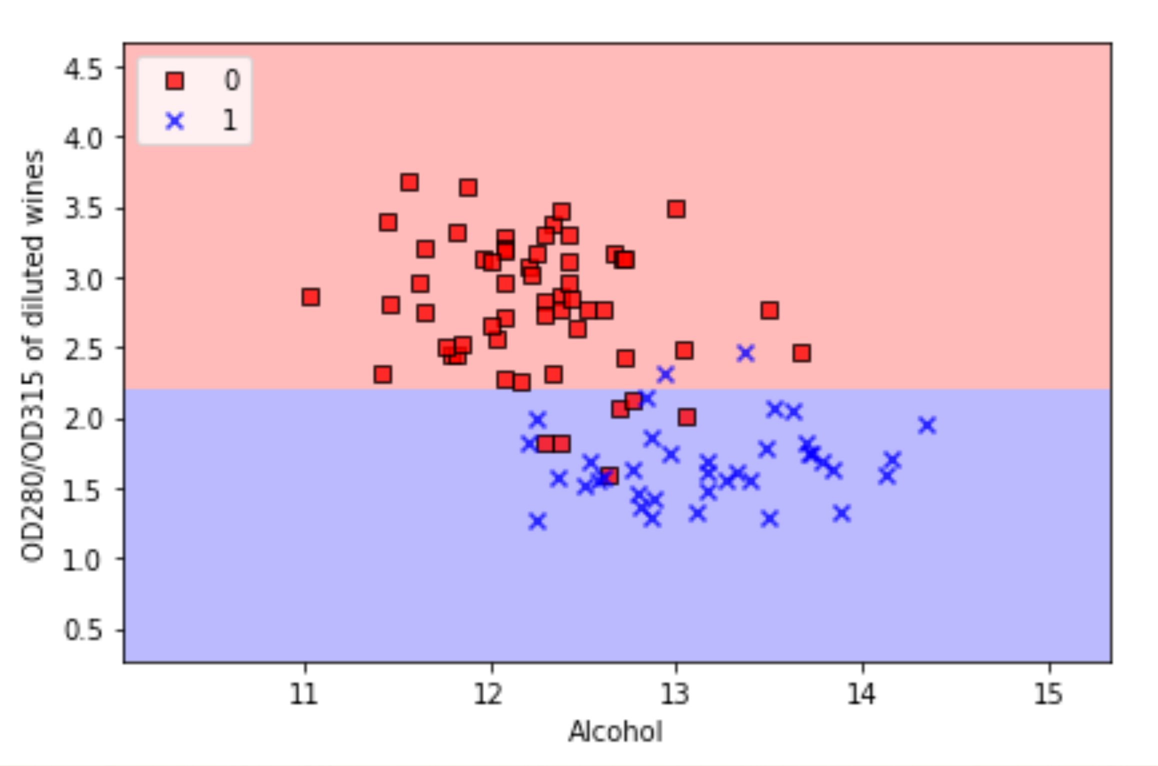
```
In [3]: from sklearn.tree import DecisionTreeClassifier

        tree = DecisionTreeClassifier(criterion='entropy',
                                      max_depth=1,
                                      random_state=1)



        from sklearn.metrics import accuracy_score


        tree = tree.fit(X_train, y_train)
        y_train_pred = tree.predict(X_train)
        y_test_pred = tree.predict(X_test)

        tree_train = accuracy_score(y_train, y_train_pred)
        tree_test = accuracy_score(y_test, y_test_pred)
        print('Decision tree train/test accuracies %.3f/%.3f'
              % (tree_train, tree_test))
```

```
Decision tree train/test accuracies 0.916/0.875
```

自主練習解答

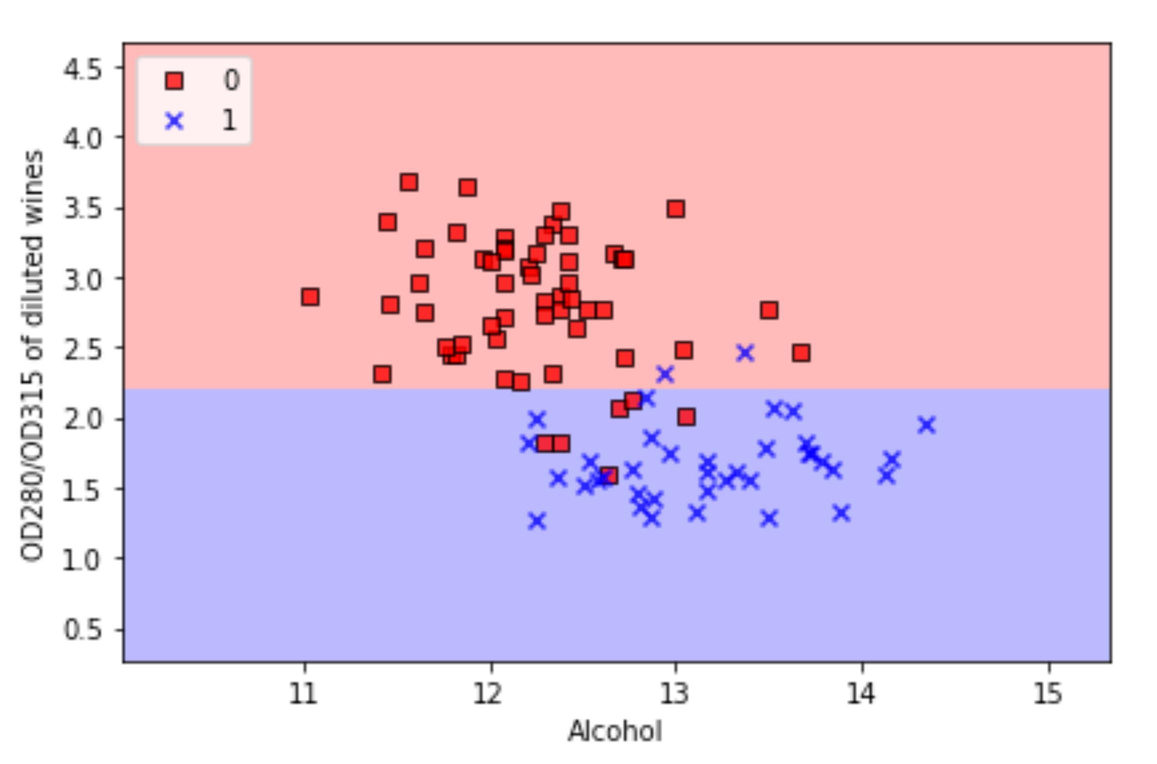


```
In [4]: plot_decision_regions(X, y, classifier=tree)
        plt.xlabel('Alcohol')
        plt.ylabel('OD280/OD315 of diluted wines')
        plt.legend(loc='upper left')
```

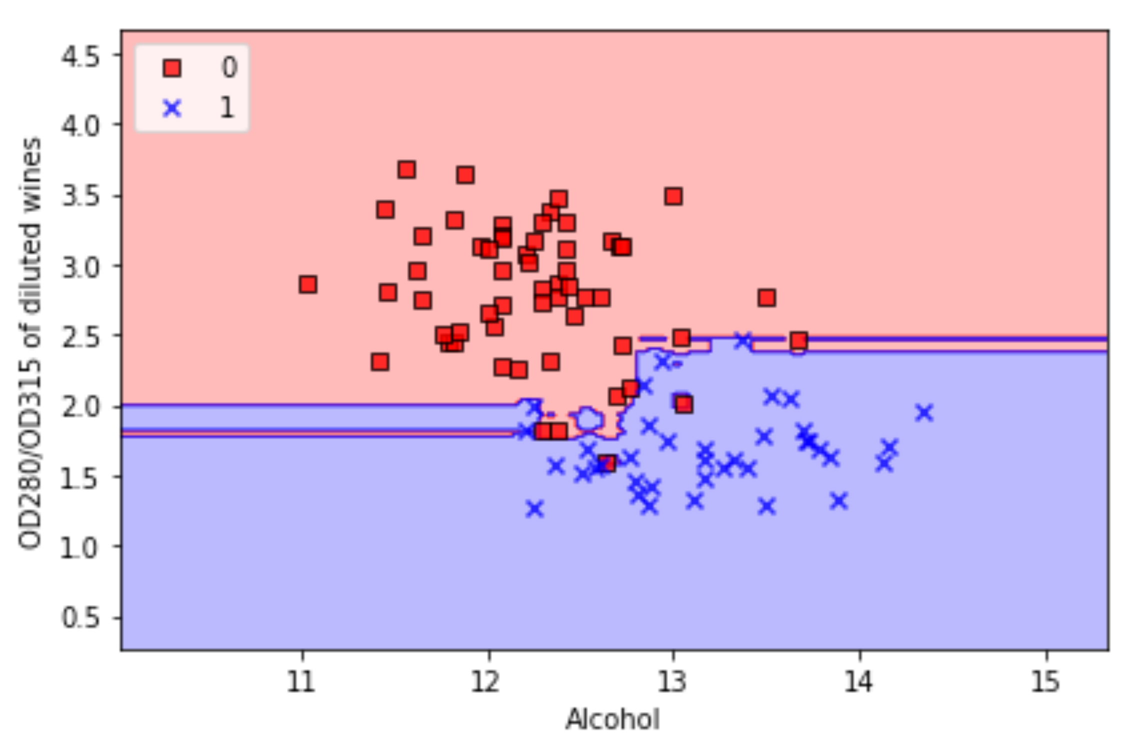


自主練習

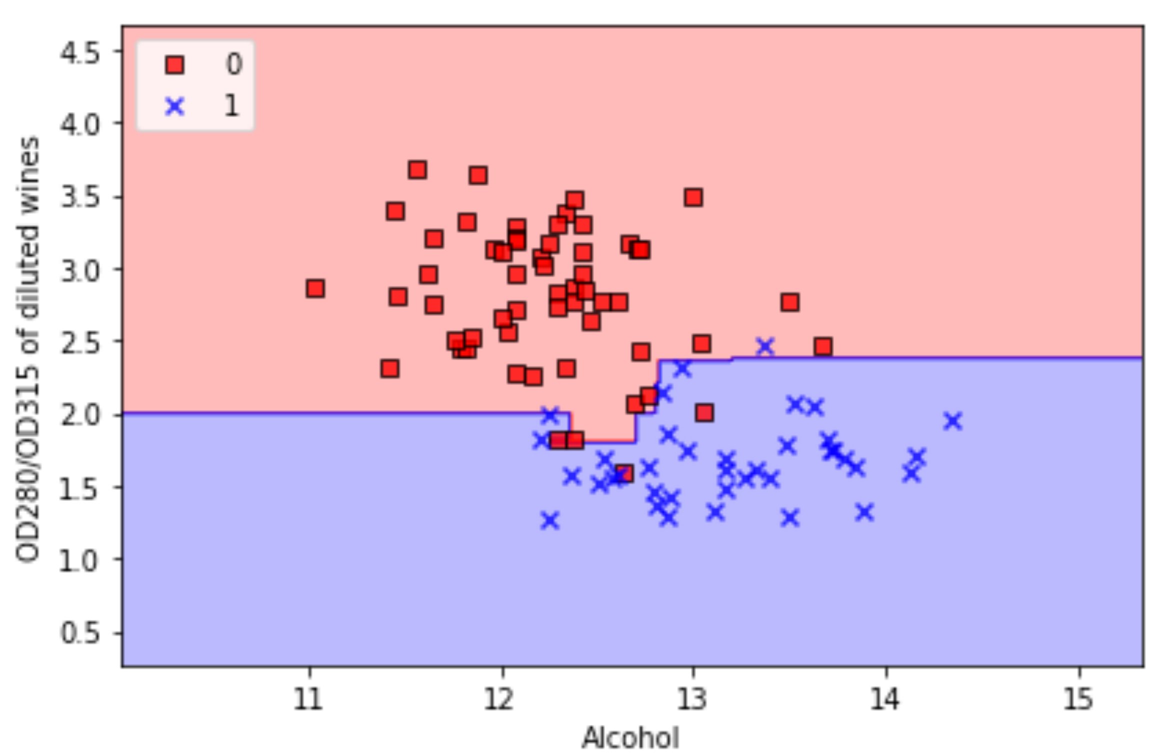
Decision tree train/test: 0.916/0.875



Random Forest train/test: 1.000/0.917



XGBoost train/test: 0.968/0.917


OD280/OD315 of diluted wines
Alcohol