姓名：李宇豪

学号：21305412

周数：6

成绩：

程序：

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report
import time

# Data preprocessing.
filename ='D://Program//pythonProject//
           Assignment_of_Numerical_Calculation//ad.data'
data = pd.read_csv(filename, header=None, low_memory=False)
cid = data.shape[1] - 1
data[cid] = (data[cid] == 'ad.').astype('int')
y = data[cid].values
data_1 = data.applymap((lambda x: "".join(x.split()) if type(x)
is str else x))
data_1.replace('?', -1, inplace=True)
X = data_1.drop(columns=cid).values
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=0)

# Test decision tree without parameter optimization.
start = time.process_time()
clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)
train_score = clf.score(X_train, y_train)
test_score = clf.score(X_test, y_test)
print('train score: {0}; test score: {1}\n'.format(train_score,
test_score))
predictions = clf.predict(X_test)
print(classification_report(y_test, predictions))
print("默认参数搜索用时：", time.process_time() - start)

# Test decision tree with grid search .
start2 = time.process_time()
entropy_thresholds = np.linspace(0, 0.1, 5)
param_grid = {'criterion': ['gini'],
              'min_impurity_decrease': entropy_thresholds,
              'max_depth': range(13, 19, 1),
              'min_samples_split': range(2, 7)}
```

```
clf = GridSearchCV(DecisionTreeClassifier(random_state=0),
param_grid, cv=5)
clf.fit(X, y)
best_parameters = clf.best_params_
print('grid search best param:\n {0} '.format(best_parameters))
print('grid search best score: {0}\n'.format(clf.best_score_))
predictions = clf.predict(X_test)
print(classification_report(y_test, predictions))

# Visualize the best decision tree.
clf_best =
DecisionTreeClassifier(criterion=list(best_parameters.values())[0
],

max_depth=list(best_parameters.values())[1],

min_impurity_decrease=list(best_parameters.values())[2],

min_samples_split=list(best_parameters.values())[3])
clf_best.fit(X_train, y_train)
fig = plt.figure(figsize=(35, 20), dpi=200)
plot_tree(clf_best, filled=True)
plt.show()
print("网格搜索用时：", time.process_time() - start2)
```

输出：

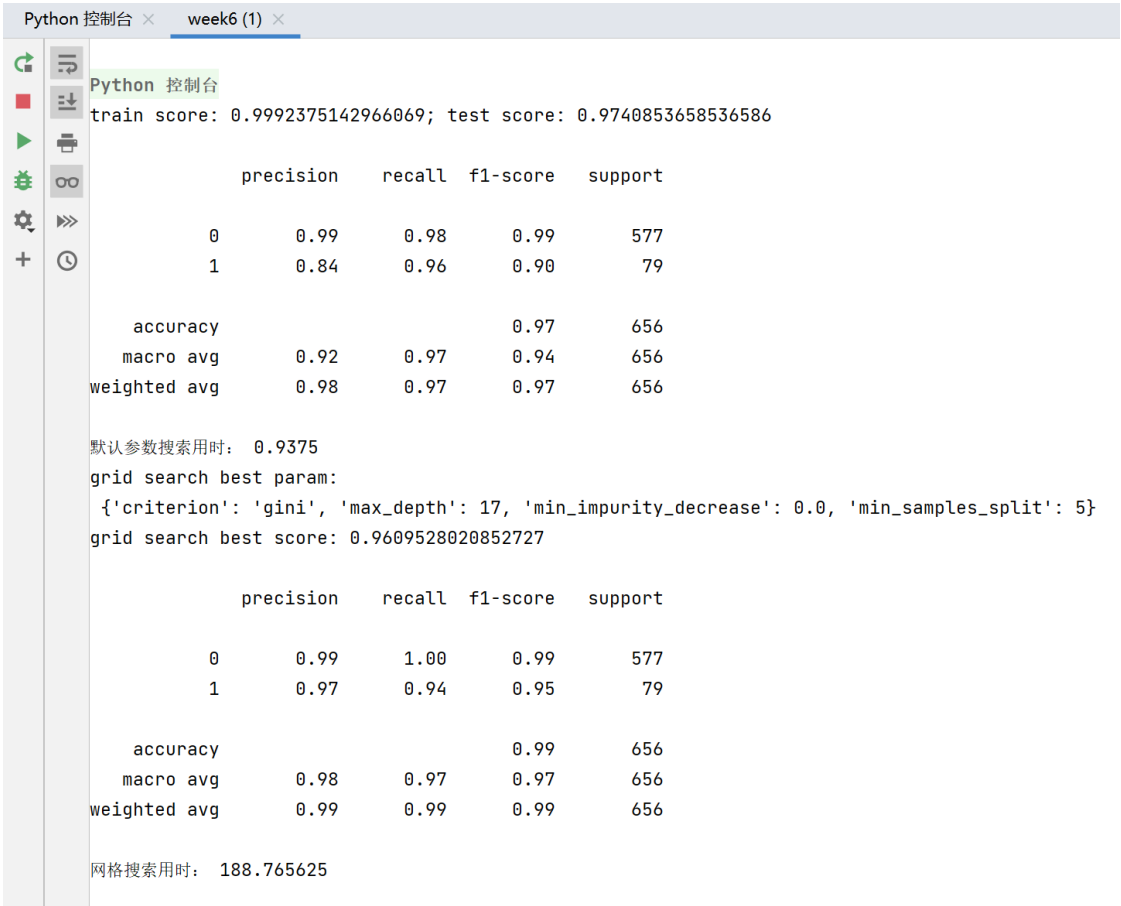

图: