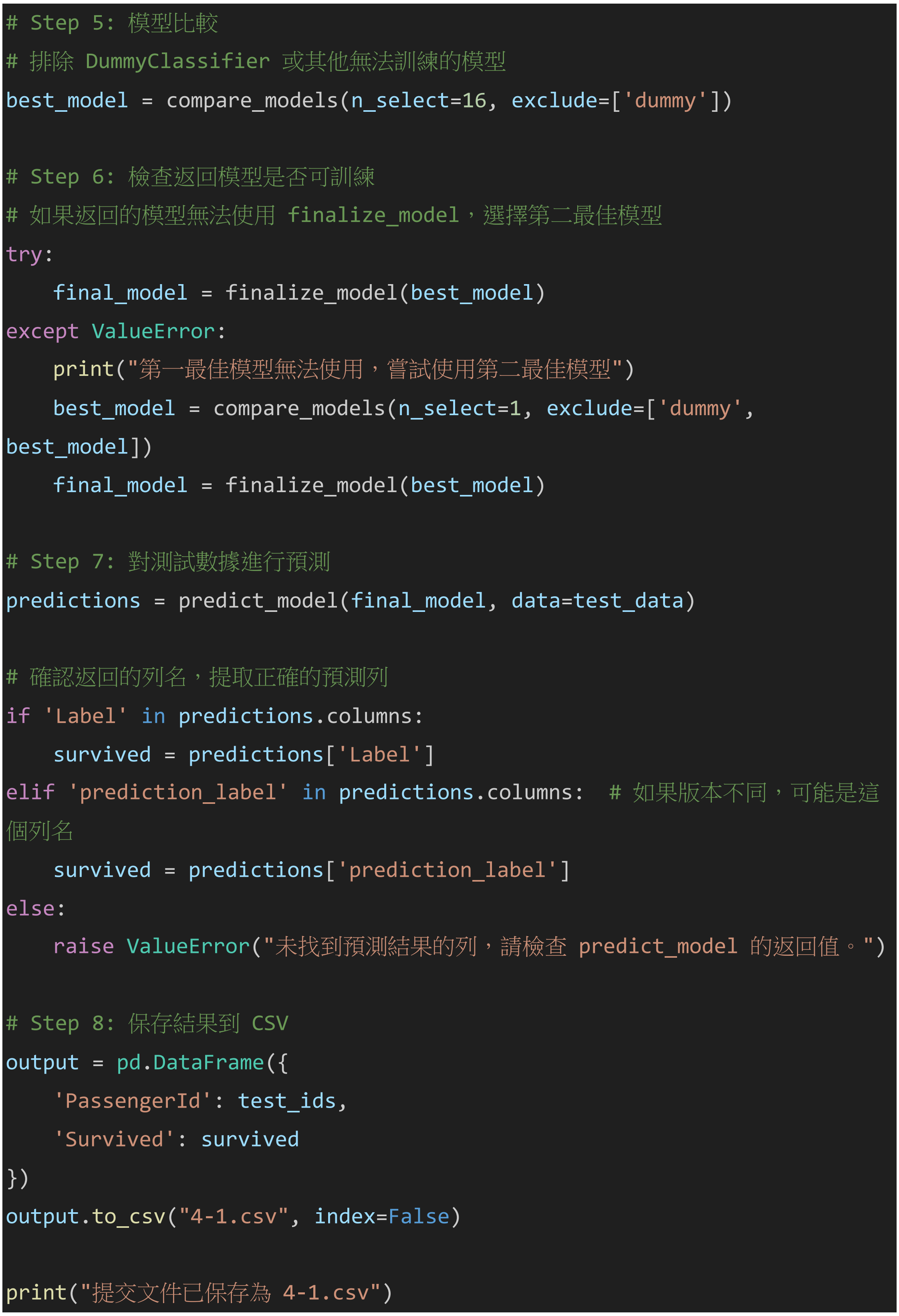
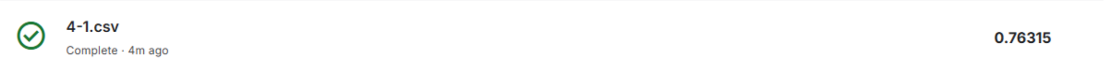
HW4-1 Pycarat to compare ML agorithms on classification problem 16 Model (multi-features titanic is ok ,)

```
# Step 1: 導入所需套件
import pandas as pd
from pycaret.classification import setup, compare_models, finalize_model, predict_model

# Step 2: 加載 Titanic 數據集
train_data = pd.read_csv("train.csv")
test_data = pd.read_csv("test.csv")

# Step 3: 數據預處理
# 對訓練數據進行處理
train_data = train_data.drop(columns=['PassengerId', 'Name', 'Ticket', 'Cabin'])
train_data['Age'] = train_data['Age'].fillna(train_data['Age'].median())
train_data['Embarked'] = train_data['Embarked'].fillna(train_data['Embarked'].mode()[0])

# 對測試數據進行相同的處理
test_ids = test_data['PassengerId']  # 保存測試數據的 PassengerId 用於生成輸出
test_data = test_data.drop(columns=['PassengerId', 'Name', 'Ticket', 'Cabin'])
test_data['Age'] = test_data['Age'].fillna(train_data['Age'].median())
test_data['Embarked'] = test_data['Embarked'].fillna(train_data['Embarked'].mode()[0])
test_data['Fare'] = test_data['Fare'].fillna(train_data['Fare'].median())

# Step 4: 建立 PyCaret 環境
classification_setup = setup(
    data=train_data,
    target='Survived',
    session_id=123  # 保持隨機性一致
)
```



```
# Step 5: 模型比較
# 排除 DummyClassifier 或其他無法訓練的模型
best_model = compare_models(n_select=16, exclude=['dummy'])

# Step 6: 檢查返回模型是否可訓練
# 如果返回的模型無法使用 finalize_model，選擇第二最佳模型
try:
    final_model = finalize_model(best_model)
except ValueError:
    print("第一最佳模型無法使用，嘗試使用第二最佳模型")
    best_model = compare_models(n_select=1, exclude=['dummy',
best_model])
    final_model = finalize_model(best_model)

# Step 7: 對測試數據進行預測
predictions = predict_model(final_model, data=test_data)

# 確認返回的列名，提取正確的預測列
if 'Label' in predictions.columns:
    survived = predictions['Label']
elif 'prediction_label' in predictions.columns:  # 如果版本不同，可能是這
個列名
    survived = predictions['prediction_label']
else:
    raise ValueError("未找到預測結果的列，請檢查 predict_model 的返回值。")

# Step 8: 保存結果到 CSV
output = pd.DataFrame({
    'PassengerId': test_ids,
    'Survived': survived
})
output.to_csv("4-1.csv", index=False)

print("提交文件已保存為 4-1.csv")
```



4-1.csv

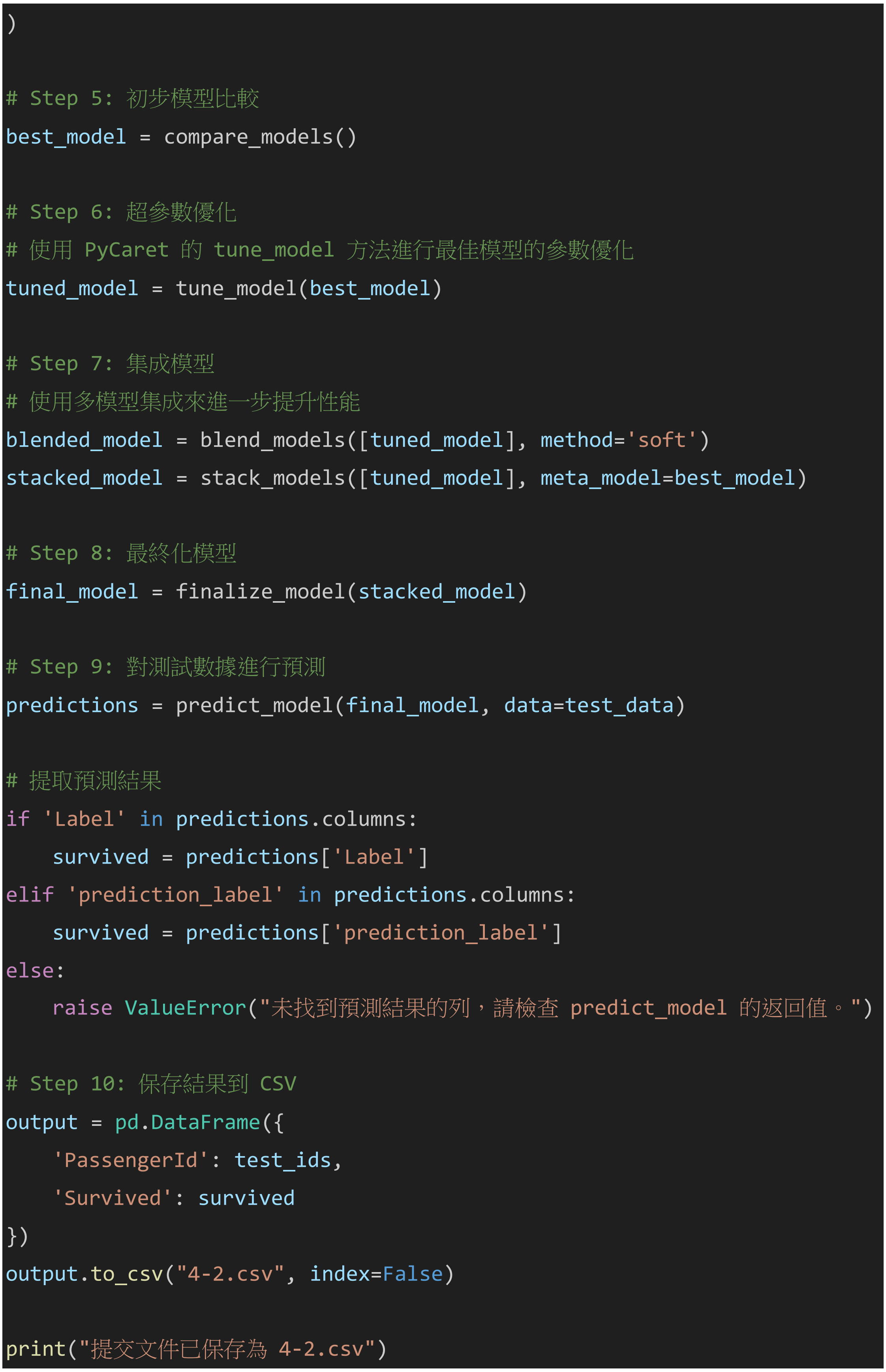
Complete · 4m ago

0.76315

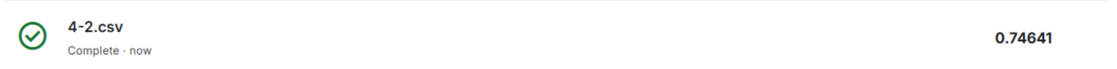
HW4-2 對 model optimization using pycarat or optuna or other AutoML,meta-heuristic

1. Feature engineering, 2. model selection (ensemble), 3 . training 超參數優化

```
# Step 1: 導入所需套件
import pandas as pd
from pycaret.classification import (
    setup, compare_models, blend_models, stack_models, tune_model,
finalize_model, predict_model
)

# Step 2: 加載 Titanic 數據集
train_data = pd.read_csv("train.csv")
test_data = pd.read_csv("test.csv")

# Step 3: 數據預處理
# 對訓練數據進行處理
train_data = train_data.drop(columns=['PassengerId', 'Name', 'Ticket',
'Cabin'])
train_data['Age'] =
train_data['Age'].fillna(train_data['Age'].median())
train_data['Embarked'] =
train_data['Embarked'].fillna(train_data['Embarked'].mode()[0])

# 對測試數據進行相同的處理
test_ids = test_data['PassengerId']
test_data = test_data.drop(columns=['PassengerId', 'Name', 'Ticket',
'Cabin'])
test_data['Age'] = test_data['Age'].fillna(train_data['Age'].median())
test_data['Embarked'] =
test_data['Embarked'].fillna(train_data['Embarked'].mode()[0])
test_data['Fare'] =
test_data['Fare'].fillna(train_data['Fare'].median())

# Step 4: 建立 PyCaret 環境
classification_setup = setup(
    data=train_data,
    target='Survived',
    session_id=123  # 保持隨機性一致
```



```
)

# Step 5: 初步模型比較
best_model = compare_models()

# Step 6: 超參數優化
# 使用 PyCaret 的 tune_model 方法進行最佳模型的參數優化
tuned_model = tune_model(best_model)

# Step 7: 集成模型
# 使用多模型集成來進一步提升性能
blended_model = blend_models([tuned_model], method='soft')
stacked_model = stack_models([tuned_model], meta_model=best_model)

# Step 8: 最終化模型
final_model = finalize_model(stacked_model)

# Step 9: 對測試數據進行預測
predictions = predict_model(final_model, data=test_data)

# 提取預測結果
if 'Label' in predictions.columns:
    survived = predictions['Label']
elif 'prediction_label' in predictions.columns:
    survived = predictions['prediction_label']
else:
    raise ValueError("未找到預測結果的列，請檢查 predict_model 的返回值。")

# Step 10: 保存結果到 CSV
output = pd.DataFrame({
    'PassengerId': test_ids,
    'Survived': survived
})
output.to_csv("4-2.csv", index=False)

print("提交文件已保存為 4-2.csv")
```



4-2.csv

Complete · now

0.74641