

100天搞定機器學習番外：使用FastAPI構建機器學習API

原創 張北海 機器學習算法與Python實戰 2021-11-22 16:26

↓↓↓點擊關注，回復**資料**，10個G的驚喜



機器學習算法與Python實戰

長期跟踪關注統計學、數據挖掘、機器學習算法、深度學習、人工智能技術與行業發展...

249篇原創內容

公眾號

100天搞定機器學習：模型訓練好了，然後呢？

FastAPI

FastAPI 是一個高性能Web 框架，用於構建API。



FastAPI 建立在 `Starlette` 和 `Pydantic` 之上。

- `Starlette`：輕量級的ASGI 框架和工具包，用來構建高性能的asncio 服務
- `Pydantic`：Python中用於數據接口schema定義與檢查的庫。通過它，可以更為規範地定義和使用數據接口。

想要深入學習這兩個庫，可以移步對應 [官方文档](#)：

<https://pydantic-docs.helpmanual.io/>
<https://www.starlette.io/>

實際應用中，FastAPI 需要與 **Uvicorn** 一起使用，Uvicorn主要用於加載和提供應用程序的服務器。

FastAPI和Uvicorn的使用方法

使用之前先安裝

```
pip install fastapi
pip install uvicorn
```

看一个小例子，感受一下FastAPI 多方便，多简单：

```
from typing import Optional
from fastapi import FastAPI
import uvicorn

#创建FastAPI实例
app = FastAPI()

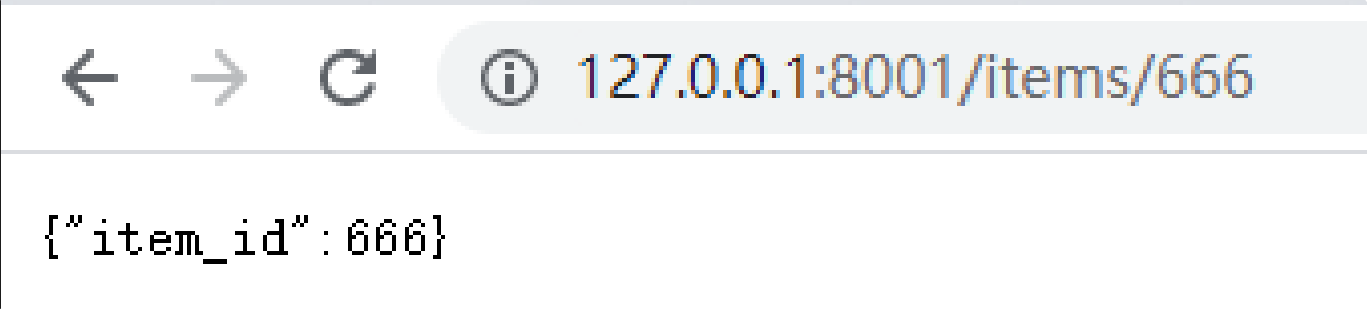
#创建访问路径
@app.get("/")
def read_root():#定义根目录方法
    return {"message": "Hello World"}#返回响应信息

#定义方法 · 处理请求
@app.get("/items/{item_id}")
async def read_item(item_id: int):
    return {"item_id": item_id}

#运行
if __name__ == '__main__':
    uvicorn.run(app, host="127.0.0.1", port=8000)
```

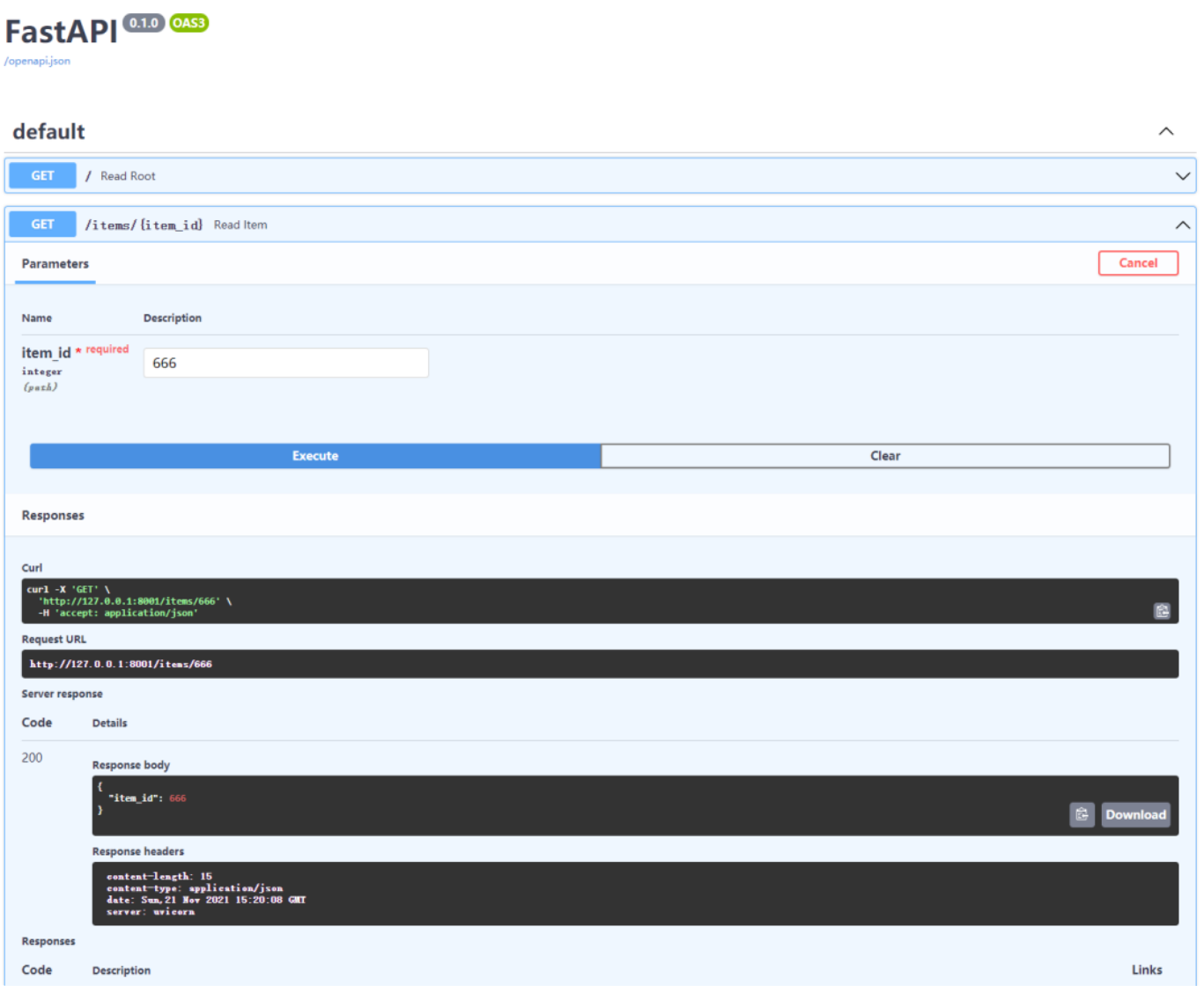
uvicorn服务器运行之后可以尝试访问

http://127.0.0.1:8000/items/666，返回：



也可进在文档中调试

打开交互文档(Swagger UI)：`http://127.0.0.1:8000/docs`
也可以访问API文档(ReDoc)：`http://127.0.0.1:8080/redoc`



FastAPI部署机器学习模型

第一步：准备模型

```
import numpy as np
import os
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
import joblib
from sklearn import datasets

def main():
    clf = LogisticRegression()
    p = Pipeline([('clf', clf)])
    print('Training model...')
    p.fit(X, y)
    print('Model trained!')

    filename_p = 'IrisClassifier.pkl'
    print('Saving model in %s' % filename_p)
    joblib.dump(p, filename_p)
    print('Model saved!')

if __name__ == "__main__":
    print('Loading iris data set...')
    iris = datasets.load_iris()
    X, y = iris.data, iris.target
    print('Dataset loaded!')
    main()
```

第二步：创建FastAPI实例

```
import uvicorn
from fastapi import FastAPI
import joblib
from os.path import dirname, join, realpath
from typing import List

app = FastAPI(
    title="Iris Prediction Model API",
    description="A simple API that use LogisticRegression model to predict the Iris species",
    version="0.1",
)

# load model
```

```

with open(
    join(dirname(realpath(__file__)), "models/IrisClassifier.pkl"), "rb"
) as f:
    model = joblib.load(f)

def data_clean(str):
    arr = str.split(',')
    arr = list(map(float, arr))
    return arr

# Create Prediction Endpoint
@app.get("/predict-result")
def predict_iris(request):
    # perform prediction
    request = data_clean(request)
    prediction = model.predict([request])
    output = int(prediction[0])
    probas = model.predict_proba([request])
    output_probability = "{:.2f}".format(float(probas[:, output]))

    # output dictionary
    species = {0: "Setosa", 1: "Versicolour", 2: "Virginica"}

    # show results
    result = {"prediction": species[output], "Probability": output_probability}
    return result

if __name__ == '__main__':
    uvicorn.run(app, host="127.0.0.1", port=8001)

```

第三步：传入参数

我们用模型预测属性为以下值时Iris应该属于哪一类，并输出预测概率。

sepal_length=7.233

sepal_width=4.652

petal_length=7.39

petal_width=0.324

打开网址，传入参数

<http://127.0.0.1:8001/predict-result?request=7.233%2C4.652%2C7.39%2C0.324>

```
{"prediction": "Virginica", "Probability": "0.89"}
```

bingo!

推荐阅读

1. 用Python学线性代数：自动拟合数据分布
2. Python 用一行代码搞事情，机器学习通吃
3. Github 上最大的开源算法库，还能学机器学习！
4. JupyterLab 这插件太强了，Excel灵魂附体
5. 终于把 jupyter notebook 玩明白了
6. 一个超好用的 Python 标准库，666
7. 几百本编程中文书籍（含Python）持续更新
8. 100天搞定机器学习|Day63 彻底掌握 LightGBM



【欢迎围观朋友圈】

备注：见面礼，领取资料

[好文点个在看吧！](#)

喜欢此内容的人还喜欢

开发软件 | Pyinstaller+Pipenv打包Python文件，简直不要太香！

大鄧和他的Python

一文讀懂異常檢測LOF 算法（Python代碼）

Python數據科學

【進階】嫌棄Python慢，試試這幾個方法？

深度學習算法與計算機視覺