

## PROMPT

撰寫一個完整的 Python 程式碼來完成以下任務，確保其能夠正確運行並符合語法要求：

從 "C:\Users\EE715\Desktop\2330-training.csv" 讀取 CSV 資料。

資料規格：

日期欄位 Date，格式為 %m/%d/%Y。

股票價格欄位 y，數字格式包含逗號。

進行資料處理：

將 Date 欄位轉換成 datetime 格式。

移除 y 欄位中的逗號，並將其轉換為浮點數。

使用 Prophet 進行預測：

模型的目標是預測 y 值（股票價格）。

設定變化點靈敏度為 0.5。

不確定性區間設置為 95%。

加入每月的季節性，並將 Fourier Order 設為 5。

預測未來 60 天的股票價格，並繪製預測圖：

繪製圖表，顯示實際的數據（黑色線條）。

預測數據顯示為藍色線條，不確定性區間使用淺藍色陰影。

添加水平虛線表示「Historical Average」。

在圖表中添加紅色虛線標記預測的初始化時間點，並標註「Forecast Initialization」。

添加一個綠色箭頭指向上升趨勢的區域，標註「Upward Trend」。

程式碼需要符合 Python 語法，無錯誤執行，且包含清晰的註解。

確保該程式碼適用於 Python 3.6 或以上版本。


把upward trend放在forecast initialization之後

用flask做web implementation deployment

準備工作：

- 安裝 Flask、Prophet、Matplotlib 等必要的套件：

```
bash
```

 複製程式碼

```
pip install flask prophet matplotlib pandas
```

## HTML Templates

你需要創建兩個簡單的 HTML 檔案來支援這個 Flask 應用。

## 應用的運作流程：

1. 首頁：使用者可以上傳一個包含 `Date` 和 `y`（價格）的 CSV 檔案。
2. 處理上傳文件：伺服器端讀取上傳的 CSV 檔案，使用 Prophet 進行股票價格預測，並生成預測圖表。
3. 顯示結果：預測結果以圖表形式顯示在網頁上，並允許使用者上傳另一個文件進行新的預測。

## 部署 Flask 應用：

1. 本地運行：直接運行 `app.py` 文件來啟動 Flask 伺服器：

```
bash

python app.py
```

複製程式碼

然後在瀏覽器中訪問 `http://127.0.0.1:5000/`。

2. 部署到雲端伺服器：

- 你可以將 Flask 應用部署到 Heroku、PythonAnywhere、AWS 等伺服器。
- 具體的部署方式會依據平台而異，例如：
  - Heroku 部署：需要安裝 Heroku CLI，並使用 `git push` 將 Flask 應用部署到 Heroku。
  - PythonAnywhere：允許簡單地上傳 Flask 應用，並設置伺服器來運行。



生成這個專案的目錄架構



要生成這個 Flask 預測專案的目錄架構，你可以使用以下結構。這將包含所有需要的檔案與資料夾來正確地運行專案。

## 項目目錄架構：

```
csharp

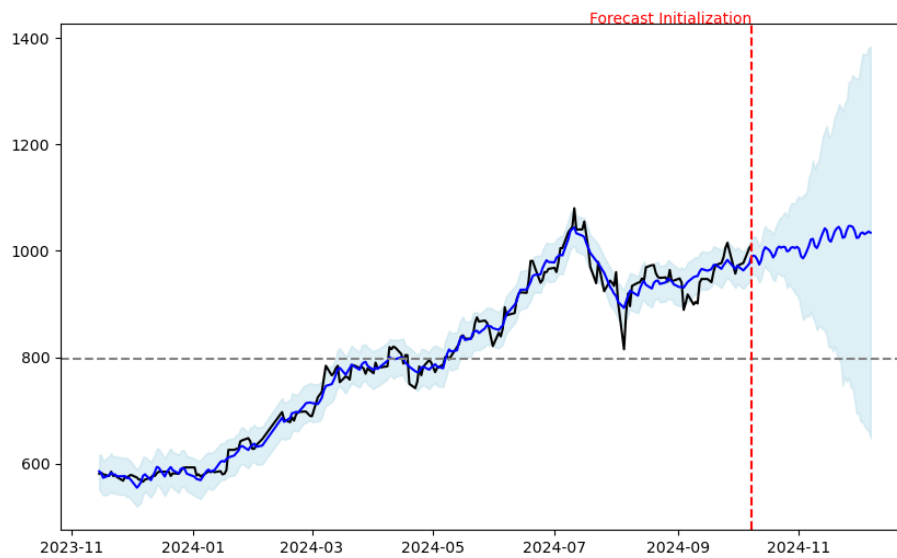
flask_prophet_project/
|
├─ app.py                # 主 Flask 應用的 Python 腳本
├─ requirements.txt      # 需要安裝的 Python 依賴庫
├─ static/              # 靜態文件夾（儲存生成的圖表）
│   └─ forecast_plot.png # 生成的預測圖（每次運行應用後生成）
|
├─ templates/           # HTML 模板文件夾
│   └─ index.html       # 上傳 CSV 文件的頁面
│   └─ result.html      # 顯示預測結果的頁面
└─ README.md            # 專案說明文件
```

複製程式碼

1. 創建一個名為 `flask_prophet_project` 的資料夾。
2. 在該資料夾中創建上述的檔案和資料夾結構。
3. 確保 `requirements.txt` 列出所有必要的依賴，並且可以通過 `pip install -r requirements.txt` 來安裝依賴。
4. 啟動 Flask 應用，並在瀏覽器中測試上傳和預測功能。

← → ↻ ⓘ 127.0.0.1:5000/upload

## Stock Price Prediction Result



[Upload Another File](#)