Midterm Project Report
Advanced Computer Programming

Web Scraping with Python

Student Name : Filbert Owen Susanto
Student ID : 112021184
Teacher : DINH-TRUNG VU

2024-04

Chapter 1 Introduction

1.1 Github

1) **Personal Github Account**: https://github.com/FOwen123
2) **Group Github Account**: https://github.com/112021179
3) **Group Project Repository**: https://github.com/112021179/1
4) **List of submitted files**:
   - 112021184.ipynb

1.2 Topic

Build a web scraper to extract information about stock historical data from a website page:

https://github.com/112021179/1/blob/main/midterm/112021184.ipynb

1.3 Project Overview

This program have used, numpy, pandas, matplotlib, time, datetime, sklearn, RandomForestClassifier, requests, BeautifulSoup, and csv libraries. It has extracted information about the stock's name and symbol. It has also extracted stock's historical data to develop a machine learning model to predict future stock prices.

Chapter 2 Implementation

2.1 Function 1

First function(scrape_stock_info) takes one parameter, which is a stock symbol. It uses the stock symbol to get a response from the stock website(finance yahoo). It uses the ‘requests’ library to send an http get request to the generated URL. If the response is successful, then it starts to scrape the webpage. The ‘BeautifulSoup’ library is then used to parse the HTML content of the website. It is trying to find specific elements on the webpage using BeautifulSoup’s ‘find()’ method. Then, it returns the stock name and sprice. However, if the request fails, it returns an error.

2.2 Function 2

‘Save_to_csv’ takes two parameters, data and the file name. Using the with open method, the function opens the file specified by the ‘filename’ parameter in write mode. This function creates a ‘DictWriter’ object from the ‘csv’ module to write the csv files. Then, it write the header and the data from the web scraper. All of this is written within a ‘with’ block to make sure the file is properly closed after it finished writing.

2.3 Function 3

The ‘predict’ function takes four parameters, training data, test data, a list of column names from the dataframe that are used for prediction, and the machine learning model. It starts to train the machine learning model using the training data and fits the model to the predictors and the target result. Then, it makes some predictions for the test data using the trained model. It uses the ‘predict_proba’ method which returns the probability estimates for each class. It sets predictions with a probability greater than or equal to 0.6 to 1, which means the price will go up, and

with a probability less than 0.6 to 0, which means the price will go down. This will make the model think twice, so when it predicts the stock will go up, it needs to be 60% sure. After that, it converts the predictions into a panda Series object, sets the index of the Series to match the index of the test data and concatenates the actual target values from the test data and the predicted values. It will return the dataframe ‘combined’ that contains both the actual and the predicted values.

2.4 Function 4

The “backtesting” function takes several parameters, historical data, machine learning model, predictors, starting index with a default value of 1250, and step size to move through the historical data with a default value of 250. It uses a for loop to iterate the historical data, taking a small portion of the data, in this case every 10 years using the step of 250. The historical data will be divided into train and test dataset using the ‘.iloc’ method to select rows by index and ‘.copy()’ to ensure a new copy of the data is created to avoid modifying the original dataframe. Then, it calls the ‘predict’ function to generate predictions. After that, it append all of the predictions into an empty list, ‘all_preds’. Finally, it concatenates all the predictions in ‘all_preds’ into a single dataframe and returns it.

Chapter 3 Results

3.1 Result 1

The first result will return a csv file of the stock's name and price that the user has inputted. The user can input more than one symbol and the program will scrape all of the stock's information.

```
[ ]  symbol = str(input("Enter the stock symbol(): "))
     symbols = (symbol.upper()).split(" ")
     for s in symbols:
         stock_data = scrape_stock_info(s)
         if stock_data:
             save_to_csv(stock_data, f"{s}_data.csv")
             print(f"Stock information for {s} has been saved to {s}_data.csv")
         else:
             print(f"There is no stock information for {s}")
```

```
Enter the stock symbol(): tsla aapl
Stock information for TSLA has been saved to TSLA_data.csv
Stock information for AAPL has been saved to AAPL_data.csv
```

3.2 Result 2

The second result will return a classification report on how well the machine learning model could work. It also make a prediction on whether tomorrow's stock price will go up or go down.

```
print(classification_report(predictions["Target"], predictions["Predictions"]))
```

```
              precision    recall  f1-score   support

           0       0.69      0.89      0.77       593
           1       0.85      0.61      0.71       620

    accuracy                           0.75      1213
   macro avg       0.77      0.75      0.74      1213
weighted avg       0.77      0.75      0.74      1213
```

```
tomorrow_data = df.iloc[[-1]]
tomorrow_prediction = RFC_model.predict(tomorrow_data[new_predictors])
if tomorrow_prediction == 1:
    print("Tomorrow's price is predicted to go up.")
else:
    print("Tomorrow's price is predicted to go down.")
```

```
Tomorrow's price is predicted to go down.
```

Chapter 4 Conclusions

This program can fetch the information of a stock and give it to the user in a csv file. Using the historical stock price information, it uses panda and numpy to clean up the data, changing it into a dataframe that is ready to be process. It develop a machine learning model using the dataframe and is expected to be able to predict future stock prices.