Botus 0.0 (@BOTUS)

Introduction

All across Wall Street, humans are being replaced by computers. Even the people who make decisions about which stocks to buy and sell are being replaced by computer programs, by bots.

Bots are cheaper than stock-picking humans. They're less emotional and more disciplined. They can process more information at once. They are doing things like scanning social media for consumer trends and counting the number of cars parked in Wal-Mart parking lots, then using that to trade.

To understand what goes on inside a stock-picking bot, the host of NPR's Planet Money built their own stock trading bot. The bot is doing something seemingly simple: It looks at President Trump's Twitter feed, and when he tweets about a company, it trades stocks, with real money. Because the official Twitter handle of the president of the United States is @POTUS, they named our bot @BOTUS, bot of the United States. [1]

In this assignment you will practice web scraping and using APIs with Python. You will build a prototype of Botus -- a program that retrieves tweets and news about a company, conducts a simple sentimental analysis, and make a simple "YES" or "NO" investment decision

You will be provided with one skeleton files and a function stored in a file called utilities.py that you will import into your skeleton file to fetch HTML from the web:

Assignment 2 Botus-skeleton.ipynb

utilities.py

[1] See more of the story and listen to the episode, http://www.npr.org/sections/money/2017/04/07/522897876/meet-botus-planet-money-s-stock-trading-twitter-bot)

Deliverable:

Answer_2.zip should contain the following:

1. A filled-in skeleton file:
    a. DELETE YOUR TWITTER CREDENTIAL BEFORE YOU UPLOAD THE FILE. WE DO NOT WANT YOUR PERSONAL INFORMATION.
    b. You **must** run the code in the result section and print out your object -- this allows us to verify your code can be run correctly.
2. Discussion.txt: Describe some ways you can improve the sentimental analysis and investment decision making. This can be one or two paragraphs. (*Hint: combine this with looking up actual stock price?)

Grading:

The documentation in the files should provide you with enough information on what the function expects (parameters) and what it should return (outputs). Please follow the instructions and you will be graded for:

Code (90%):

80% - completion: all functions are completed and can be run correctly.

10% - documentation: improve the current documentation

Discussion (10%)

10% - discussion.txt

Instruction

1. Install necessary packages

The Python packages you will need to install and use include:

Tweepy: for interacting with Twitter API

BeautifulSoup: for parsing HTML texts

Requests:for opening and reading URLs

vaderSentiment: for conducting sentiment analysis

NLTK: for conducting natural language processing

2. Import Utilities

Open utilities.py

This file contains one function called fetch that takes in a url and returns the html content of the page. It also saves the output to a local file so it will retrieve the content locally if available, aka caching it. This allows us to avoid calling the same website over and over again. Import the fetch function from utility into your main code. Use it whenever you need to get the html content of a website.

3. Inspect Object

We will hold all the tweets/articles/sentimental score/investment decision for each company in a company object. Upon initiation, it has the following attributes:

name = name

articles = None

tweets = None

articles_sentiments = None

tweets_sentiments = None

investment = None

We will fill in the value for each attribute in the Result section.

4. Twitter API:

To start with, you will need to have a Twitter account and obtain credentials (i.e. API key, API secret, Access token and Access token secret) on the Twitter developer site to access the Twitter API, following these steps:

- Create a Twitter user account if you do not already have one.
- Go to https://apps.twitter.com/ and log in with your Twitter user account. This step gives you a Twitter dev account under the same name as your user account.
- Click “Create New App”
- Fill out the form, agree to the terms, and click “Create your Twitter application”
- In the next page, click on “Keys and Access Tokens” tab, and copy your “API key” and “API secret”. Scroll down and click “Create my access token”, and copy your “Access token” and “Access token secret”.

Now we are ready to retrieve some tweets!

For this, you are provided with 3 functions:

get_tweets(company_name, n=10): DO NOT MODIFY THIS FUNCTION
create_twitter_api()
get_tweet_bag_of_words(tweets)

get_tweets takes a name of a company, retrieves n tweets about the company, and returns a list of words in the tweets. It also writes out the tweets you obtained.

Getting tweets is accomplished in two steps:

create_twitter_api takes no parameters but creates and returns a twitter API wrapper that you can use to interact with Twitter. You will copy and paste your credentials obtained in the first step here. ERASE THEM WHEN YOU TURN IN THE ASSIGNMENT.

We will use the search method to get search results of the company, see the documentation about the method here:
http://docs.tweepy.org/en/v3.5.0/api.html#tweepy-api-twitter-api-wrapper

get_tweet_bag_of_words takes the tweet search results and returns a list of words. You will need to parse the tweet search results in order to get the twitter status texts, get rid of the hashtag and other punctuations, and find a list of words. See what’s being returned:
https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets

5. Web Scraping:
You will get data from USA News Today about the company. You are provided with 4 functions:

get_article(company_name, n=3): NOT MODIFY THIS FUNCTION
get_search_results(company_name, n=3)
get_articles_from_search_results(results)

get_articles_bag_of_words(articles)

Essentially, in the get_article function, we will first get the search result page of the USA News Today about the company. From the result age we can obtain all the links to the news about the company. We will then open the links to access each piece of news and store the texts. Finally we will process the texts into a list of individual words.

get_search_results takes the name of the company and returns the search result page of USA News Today of the company. Below are some search result links for different terms.
In technical terms: observe the pattern, recreate your own search link, and use fetch function we imported to get the html content.
https://www.usatoday.com/search/facebook/

get_articles_from_search_results takes the search result page, get all the links to the news articles, and opens n links (default to 3 articles: for testing we suggest you use a small number because you will get your results faster and you are not aggressively scraping content and risking getting banned). Not all links returned from the page will be links to articles, so your function should ensure the links selected are in fact articles from the company search. **In technical terms: find all the links to articles on the page (<a> using BeautifulSoup), and use fetch function we imported to get the html content of each link, store the article htmls, return a list of html.**

get_articles_bag_of_words(articles) takes the html from the articles and return a list of words. **In technical terms: for each article, find all the texts on the page (<p> using BeautifulSoup), extract words, clean words, return a list of words.**

5. Get the sentiments:
get_sentiment(text) takes a list of words and return a sentimental score. DO NOT MODIFY THIS FUNCTION
You will want to use the function to get the sentiment scores for tweets and articles and store them in company.tweets_sentiments and company.articles_sentiments respectively.

6. Make a decision:
get_decision(tweets_sentiments, articles_sentiments) takes sentiment scores from tweets and articles and return True if the total sentimental score is larger than 0, aka positive. DO NOT MODIFY THIS FUNCTION.

Note: Those who have negative articles in the press should have a negative sentiment score. Verify that your solution effectively characterizes both positive and negative subject matter.

7. Print result:

YOU MUST PRINT YOUR RESULT.

Run the code below Result section. Substitute the company name with the company you are interested in and print the result -- this will allow us to verify that your code can be run correctly.

8. Documentation: You will be graded for documentation. See the proper way to document here: https://github.com/numpy/numpy/blob/master/doc/HOWTO_DOCUMENT.rst.txt#method-docstrings