**Problem:**

Financial markets are very fast-moving and affected by a variety of factors, including public policy, taxes, current events, consumer sentiment, price changes and more. All of these factors are reported in the **news**, and the **news** (and other information obtained by traders) serves as the basis for high-stakes **trading activity**.

Currently, traders have to rely on their own **intuition** and personally judge **impact** when evaluating news articles and deciding whether or not to **trade based on the news**. It is hard to spend time analyzing any individual news piece for **accuracy, timeliness and potential impact on asset prices** when you have to make trades on a second-by-second basis.

**Solution:**

Create a bot that **supplements traders’ workflows** by **alerting them** to **relevant, timely and impactful news** that should **inform their trading activity**. Our bot hopes to analyze and interpret newly published news articles to determine whether or not the article—and the topic on which it is reporting—will **significantly affect asset prices**.

We will analyze and correlate the past publication of news articles with the price history of assets in order to teach the bot how to **identify significant news articles** based on factors such as **reputation of the site, historical accuracy of the site, verification of the topic being reported on, and article content**.

Then, as new articles are published, the bot will algorithmically **determine the relevance and significance of the piece** and **alert traders** to articles that have a **high likelihood of affecting asset prices**. Traders will also be able to research publications and their reputation/significance in terms of **high-impact news** **that results in trading activity**. The bot should be **configurable to any asset**, including cryptocurrency, stocks and commodities.

Basically, we are trying to solve this problem: **How do traders know which news articles/publications/news beats they should listen to and base their trading decisions on?**

1. What asset is the trader working with?

* Get a list of past news articles on the topic from every available publication
* Establish a listener to keep track of newly published articles

2. Which of these news articles are “relevant” to the price of the asset being traded (i.e. which of these news items directly affect prices and lead to trading activity?)

* Regress article timing and “attributes” to prices to establish a relationship
* Generate a “Relevance Score” based on a number scale or (1-5) or Low, Medium and High

3. How reputable are the sources from which these articles are published?

* "Reputation Score” can be initially assigned i.e. Reuters, AP, PRN etc. can be whitelisted with a high score
* Eventually, “Reputation Score” can be crowdsourced/algorithmically determined and trustworthiness can be established using several variables.
  + How long has the publisher been around?
  + Historically, how accurate and relevant to trading activity have their articles been?

FINAL Article Score will be a function of Relevance Score + Reputation Score of the publisher. This score will be assigned to future articles and displayed to the trader based on the total score. Traders can then get updates in real-time with our bot predicting how impactful the news will be.

**List of News Services APIs**

<https://www.programmableweb.com/category/news%20services/apis?category=20250>

General individual news media official APIs from wikipedia:

<https://en.wikipedia.org/wiki/List_of_news_media_APIs>

<https://www.quora.com/Anyone-familiar-with-a-RSS-news-feed-or-API-that-gives-you-the-content-of-news-articles> (this one has some good links)

<https://newsapi.org/sources>

<http://eventregistry.org/search?type=articles>

**1. For what (tickers, companies, topics, sectors, regions, people) collectively called “themes” does the trader want news?**

**General resources:**

<https://blog.algorithmia.com/create-your-own-machine-learning-powered-rss/>

<https://www.wired.com/2010/02/rss_for_beginnners/>

<http://techpp.com/2009/04/27/top-10-free-tools-to-create-rss-for-any-website-2/>

<https://feedity.com/>

<https://www.howtogeek.com/318401/how-to-find-or-create-an-rss-feed-for-any-website/>

<https://www.codeinwp.com/blog/best-wordpress-rss-feed-plugins/>

<https://www.michaelhartzell.com/marketing-insights-ezine/how-to-embed-rss-feed-into-website-for-diy-entrepreneurs>

<https://github.com/spatie/laravel-feed>

<https://www.linux.com/news/building-command-line-generator-rss-feeds>

<https://www.eclipse.org/>

\* <https://www.cam.ac.uk/news-feed-generator>

**Tools for Building RSS Reader (aggregator) Using Python:**

**Basic overview:**

<https://www.thoughtco.com/build-an-rss-reader-with-python-2813529>

**Building with native libraries:**

<https://www.blog.pythonlibrary.org/2014/01/09/wxpython-create-rss-reader/>

**About libraries:**

<https://wiki.python.org/moin/RssLibraries>

**Intro tutorials:**

1. <http://www.pythonforbeginners.com/feedparser/using-feedparser-in-python>
2. <https://jcutrer.com/howto/dev/python/python-tutorial-howto-parse-rss-headlines>
3. <https://alvinalexander.com/python/python-script-read-rss-feeds-database>
4. <https://www.pubnub.com/blog/2014-06-11-the-easiest-way-to-create-a-realtime-rss-feed-using-python-and-javascript/>

**A good model:**

<https://indico.io/blog/create-customized-rss-feed-with-indico-python-html/>

**Options for loading RSS Feed (default):**

<https://www.google.com/search?q=data+base+of+RSS+feeds+for+finance&oq=data+base+of+RSS+feeds+for+finance&aqs=chrome..69i57.7728j0j4&sourceid=chrome&ie=UTF-8>

**Research paper on building a Feed-based aggregator**

[**https://arxiv.org/pdf/1207.2596.pdf**](https://arxiv.org/pdf/1207.2596.pdf)

**3) (2 people will work on this together) A tool (open-source/free is preferred) that allows us to search articles/databases/the web/news feeds based on certain user-defined parameters. Basically, we don't want to have to rebuild Google. If we can find a tool/Python library that lets us a) input parameters/search terms, and b) submit a body of text/a feed/a list of items to search, that would be great. We can build on it and use it as the base for our whole value proposition.**

* Web spider/ crawler to get articles/databases/the web/news based on user-defined parameters
* Analyze scraped data to get useful info
* NLP, and ML to generate predict model

**Web spider/crawler**

<https://github.com/scrapy/scrapy>

**Scrap useful information from web (e.g. stock price)**

<https://github.com/rahulrrixe/finance-news-crawler> (bloomberg)

<https://github.com/jackstine/stockAnalysis/blob/83f74c5fc1c5216baa28fccf6e5a6511dfd6c11b/pycharmCode/stock/scrape/scrapycrawls/bloomberg/financials/financials/spiders/BloombergFinancials.py> (bloomberg, stock)

<https://github.com/tjfrahme/sfdat26-frahme/blob/818001d2889f72563e0cae4327129c57c220b21b/notebooks/.ipynb_checkpoints/03_getting_data_from_web-checkpoint.ipynb> (google finance get stock price)

**Some related Python tutorials**

<https://pythonprogramming.net/getting-stock-prices-python-programming-for-finance/>

BeautifulSoup

<https://www.pythoncentral.io/python-beautiful-soup-example-yahoo-finance-scraper/>

**Sentimental analysis on financial news**

<https://github.com/WayneDW/Sentiment-Analysis-in-Event-Driven-Stock-Price-Movement-Prediction>

<https://github.com/VincentTatan/PythonAnalytics/blob/9a184eca76f6f4a013e3964a552702597c0c0f74/Youtube/.ipynb_checkpoints/Lesson%207%20Intermediate%20Python%20for%20Data%20Analytics%20(Finance%20Performance%20and%20Fraudulent%20Detection)-checkpoint.ipynb>

[**https://www.geeksforgeeks.org/performing-google-search-using-python-code/**](https://www.geeksforgeeks.org/performing-google-search-using-python-code/)

* Performs a google search but with python code, uses beautiful soup

[**https://opensource.com/resources/python/web-scraper-crawler**](https://opensource.com/resources/python/web-scraper-crawler)

* Libraries for web scraping - might be helpful

<https://www.twilio.com/blog/2016/12/http-requests-in-python-3.html>

* More python libraries to work with web

<https://www.datacamp.com/courses/building-chatbots-in-python>

* For creating a chatbot

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4988639/> (Predicting Fluctuations in Cryptocurrency Transactions Based on User Comments and Replies)