

# Cómo construir una herramienta de análisis de sentimiento para el comercio de acciones

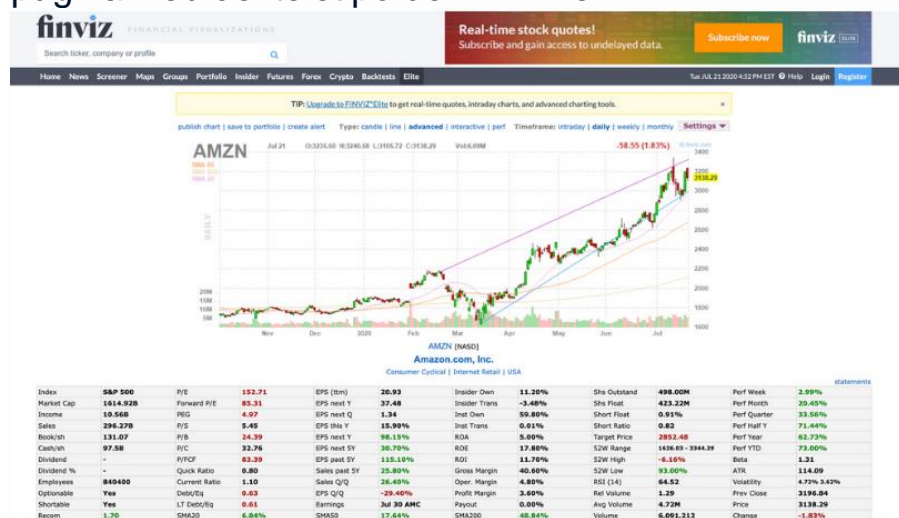
construiremos una herramienta de análisis de sentimientos para los titulares de operaciones bursátiles. Este proyecto le permitirá perfeccionar sus habilidades de extracción web, análisis y manipulación de datos y visualización para crear una herramienta completa de análisis de sentimientos.

Aquí hay una hoja de ruta para el proyecto de hoy:

1. Usaremos **Beautifulsoup en Python** para raspar los titulares de los artículos de FinViz
2. Luego, usaremos **Pandas** (Biblioteca de análisis de datos de Python) para analizar y ejecutar **análisis de opinión** en los títulos de los artículos.
3. Finalmente, usaremos **Matplotlib para la visualización** de nuestros resultados.

## Paso 1: Recopilación y análisis de datos de FinViz

FinViz es un sitio web gratuito que hace que los comerciantes e inversores puedan acceder fácilmente a los datos bursátiles. Recopilaremos los datos bursátiles de FinViz para un indicador bursátil específico. Por ejemplo, así es como se ve la página web del teletipo de AMZN en FinViz



Datos de FinViz para Amazon

Si se desplaza hacia abajo, verá los artículos de stock que estamos tratando de analizar. Ver código fuente muestra el código HTML exacto que contiene el nombre del artículo de stock y la fecha en que se publicó:



The screenshot shows a table of stock news articles. The first article is 'Amazon Prime Day Officially Postponed Due To Coronavirus Impact' published on Jul-21-20 at 04:14PM. The HTML code on the right shows the structure of the table, with the article title and date being part of a link container.

Date	Time	Article Title
Jul-21-20	04:14PM	Amazon Prime Day Officially Postponed Due To Coronavirus Impact
03:38PM		S&P 500 turns positive for 2020, but most stocks
02:59PM		Microsoft Makes First Climate Fund Investment, J
02:38PM		Only a Few Stocks Are Fueling the Markets Rise.
02:20PM		France's Macron: we want an EU-wide digital tax
01:44PM		Amazon Prime Day Officially Postponed Due To Coronavirus Impact
01:25PM		US STOCKS-S&P 500, Dow rise after positive earnings
01:14PM		We need to back more female founders, 'not just
01:05PM		Amazon Sets New Lobbying Record as Tech Antitrust
12:29PM		Prime day is not "a requirement for Amazon base
12:16PM		Amazon Small Business Sales Surge 60% Amid Pandemic

```
<tr><td width="130" align="right" style="white-space:nowrap">Jul-21-20 04:14PM&nbsp;&nbsp;&nbsp;</td><td align="left"><div class="news-link-container"><div class="news-link-left"><a href="https://www.investors.com/news/technology/amazon-prime-day-officially-postponed-due-to-coronavirus-impact/?src=A00220" target="_blank" class="tab-link-news">Amazon Prime Day Officially Postponed Due To Coronavirus Impact</a></div><div class="news-link-right"><span style="color:#aa6dc0;font-size:9px"> Investor's Business Daily</span></div></div></td></tr>
```

escribamos código BeautifulSoup para guardar esta tabla de artículos en un conjunto de datos. En un nuevo proyecto de Python, cree un archivo *main.py* y escriba el siguiente código:

```
from urllib.request import urlopen, Request
from bs4 import BeautifulSoup
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import pandas as pd
import matplotlib.pyplot as plt

finviz_url = 'https://finviz.com/quote.ashx?t='
tickers = ['AMZN', 'GOOG', 'FB']

news_tables = {}
for ticker in tickers:
    url = finviz_url + ticker

    req = Request(url=url, headers={'user-agent': 'my-app'})
    response = urlopen(req)

    html = BeautifulSoup(response, features='html.parser')
    news_table = html.find(id='news-table')
    news_tables[ticker] = news_table
```

**Asegúrese de tener todos los módulos anteriores instalados a través de pip.**

Aquí, hemos creado una serie de tickers, y para cada uno creamos la URL de FinViz completa para analizar los datos. Usando el

módulo Request en Python, obtenemos la respuesta html del sitio web y la lanzamos a BeautifulSoup para que podamos analizarla fácilmente. El elemento HTML con id 'tabla de noticias' contiene todos nuestros artículos de noticias, por lo que estamos guardando ese elemento BeautifulSoup en un diccionario. Analicemos ese diccionario ahora mismo:

```
parsed_data = []

for ticker, news_table in news_tables.items():
    for row in news_table.findAll('tr'):
        title = row.a.text
        date_data = row.td.text.split(' ')

        if len(date_data) == 1:
            time = date_data[0]
        else:
            date = date_data[0]
            time = date_data[1]

        parsed_data.append([ticker, date, time, title])

df = pd.DataFrame(parsed_data, columns=['ticker', 'date', 'time',
'title'])
df['date'] = pd.to_datetime(df.date).dt.date
```

Nuestro código de análisis simplemente manipula la tabla de noticias que guardamos mientras recopilamos los resultados y analiza los valores específicos que necesitamos. Estamos buscando todas las filas de la tabla en la tabla de artículos de noticias y reuniendo el título, la fecha y la hora de cada artículo publicado. Una vez que tengamos esos valores, podemos guardar cada dato como un objeto de matriz en nuestra matriz `parsed_data`.

En las últimas 2 líneas, convertimos nuestra matriz `parsed_data` en un marco de datos de Pandas y configuramos la columna Fecha para que tenga el formato Python Datetime. Esto nos permitirá aplicar fácilmente el análisis de sentimientos y visualizar los datos con Matplotlib.

## Paso 2: Aplicar el Análisis de Sentimiento

Aplicar el análisis de sentimientos en los títulos es en realidad la parte más fácil de todo el proyecto. Con NLTK (Natural Language Toolkit) viene un hermoso submódulo llamado vader que nos permite pasar una cadena a su función y obtener un resultado de aspecto original como este:

```
Very bad movie.  
compound: -0.5849, neg: 0.655, neu: 0.345, pos: 0.0,
```

Podemos ver que la cadena "Muy mala película". devuelve la respuesta de 4 variables, compuesta, negativa, neutra y positiva. El resultado compuesto es un rango entre -1 y 1, siendo -1 abrumadoramente negativo y +1 positivo respectivamente. Este será el resultado del que deduciremos si un artículo en stock es positivo o negativo.

Avancemos y apliquemos el análisis de sentimiento en nuestro marco de datos:

```
vader = SentimentIntensityAnalyzer()  
  
f = lambda title: vader.polarity_scores(title)['compound']  
df['compound'] = df['title'].apply(f)
```

Inicializamos el `SentimentIntensityAnalyzer` y luego creamos una función lambda que toma una cadena de título, le aplicamos la función `vader.polarity_scores()` para obtener los resultados en la imagen de arriba y luego solo devolvemos la puntuación compuesta. Usando la función de aplicación en Pandas, podemos crear una nueva columna 'compuesta' en el marco de datos con todas las puntuaciones compuestas de cada título.

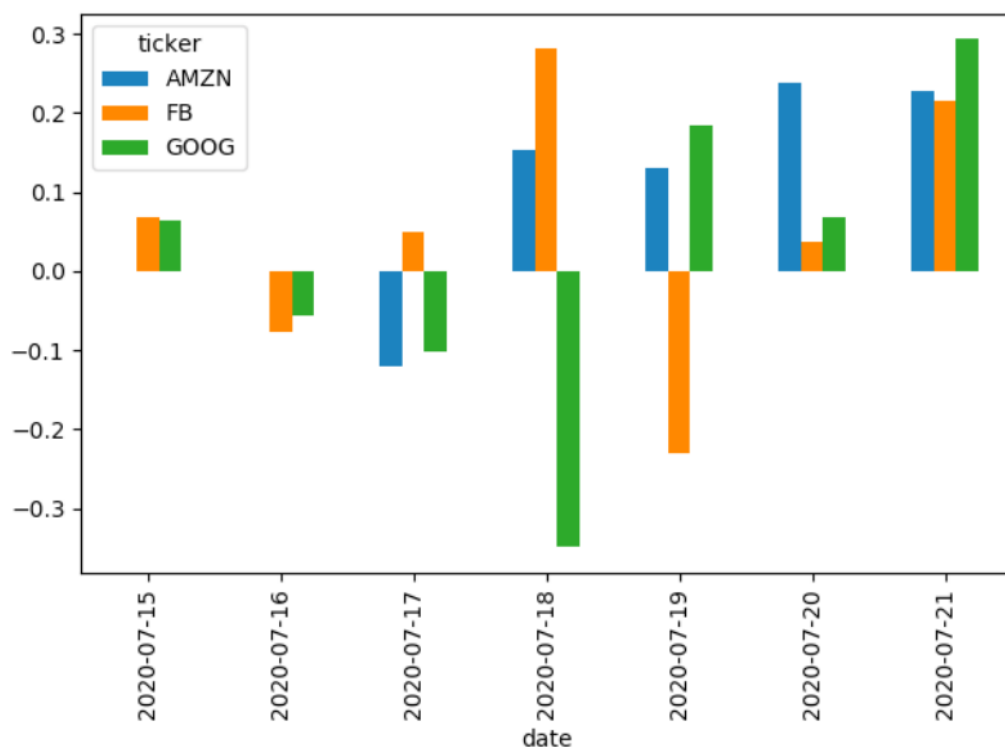
## Paso 3: Visualización de los resultados en Matplotlib

Por último, pero no menos importante, necesitamos visualizar este marco de datos en Matplotlib para ver cómo les fue a nuestras acciones todos los días según la percepción pública en los artículos de noticias.

Visualicemos los resultados en un gráfico de barras, agrupando los datos según los tickers y las fechas:

```
plt.figure(figsize=(10,8))
mean_df = df.groupby(['ticker', 'date']).mean().unstack()
mean_df = mean_df.xs('compound', axis="columns").transpose()
mean_df.plot(kind='bar')
plt.show()
```

El código de visualización anterior agrupa nuestro conjunto de datos según el ticker y las fechas de cada fila, y luego visualiza el puntaje compuesto promedio de cada día. Tomamos la sección transversal de las filas 'compuestas', volteamos el marco de datos para que tengamos las fechas como el eje x y luego lo trazamos como un gráfico de barras.



Y ¡Acaba de crear una herramienta de análisis de sentimiento para el comercio de acciones!