Use Finnhub API data with Plotly Library for Python

Technical Summary of Jupyter Notebook for Stock Data Analysis

Setting Up the Environment

- **Imports**: Essential libraries are imported:
- **plotly** for interactive plotting.
- **pandas** for data manipulation.
- **jproperties** for configuration management.
- **finnhub** to access Finnhub's API.
- **datetime** for handling date and time data.

```
pip install panda plotly jproperties


Requirement already satisfied: panda in /opt/conda/lib/python3.10/site-packages (0.3.1)
Requirement already satisfied: plotly in /opt/conda/lib/python3.10/site-packages (5.22.0)
Requirement already satisfied: jproperties in /opt/conda/lib/python3.10/site-packages (2.1.1)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.10/site-packages (from
panda) (67.7.2)
Requirement already satisfied: requests in /opt/conda/lib/python3.10/site-packages (from
panda) (2.31.0)
Requirement already satisfied: tenacity>=6.2.0 in /opt/conda/lib/python3.10/site-packages
(from plotly) (8.2.2)
Requirement already satisfied: packaging in /opt/conda/lib/python3.10/site-packages (from
plotly) (23.1)
Requirement already satisfied: six~=1.13 in /opt/conda/lib/python3.10/site-packages (from
jproperties) (1.16.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /opt/conda/lib/python3.10/site-
packages (from requests->panda) (3.1.0)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.10/site-packages (from
requests->panda) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /opt/conda/lib/python3.10/site-packages
```

```
(from requests->panda) (2.0.2)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/lib/python3.10/site-packages
(from requests->panda) (2023.5.7)
Note: you may need to restart the kernel to use updated packages
```

```
pip install finnhub-python

Collecting finnhub-python
  Downloading finnhub_python-2.4.19-py3-none-any.whl (11 kB)
Requirement already satisfied: requests>=2.22.0 in /opt/conda/lib/python3.10/site-packages
(from finnhub-python) (2.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /opt/conda/lib/python3.10/site-
packages (from requests>=2.22.0->finnhub-python) (3.1.0)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.10/site-packages (from
requests>=2.22.0->finnhub-python) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /opt/conda/lib/python3.10/site-packages
(from requests>=2.22.0->finnhub-python) (2.0.2)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/lib/python3.10/site-packages
(from requests>=2.22.0->finnhub-python) (2023.5.7)
Installing collected packages: finnhub-python
Successfully installed finnhub-python-2.4.19
Note: you may need to restart the kernel to use updated packages.
```

Load Properties

[2]:
```
# Initialize from property file
configs = Properties()

with open('fh_analyst_trends.properties', 'rb') as config_file:
     configs.load(config_file)

TICKER = configs.get('TICKER').data
API_KEY = configs.get('FH_API_KEY').data
TEMPLATE  = configs.get('TEMPLATE').data
```

Load Data

[3]:
```
# Setup client
finnhub_client = finnhub.Client(api_key=API_KEY)

# Stock recommendations
res = finnhub_client.recommendation_trends(TICKER)

#Convert to Pandas Dataframe
df = pd.DataFrame(res)
df
```

[3]:

	buy	hold	period	sell	strongBuy	strongSell	symbol
0	36	5	2024-05-01	0	22	0	NVDA
1	36	5	2024-04-01	0	22	0	NVDA
2	35	5	2024-03-01	0	21	0	NVDA
3	33	5	2024-02-01	0	20	0	NVDA

Utility method to convert date to month year format

[4]:
```
# Apply this method to convert dates to month year format
def add_month_year_col(x):
    date_obj = datetime.datetime.strptime(x, '%Y-%m-%d')
    return date_obj.strftime('%b') + ' ' + date_obj.strftime('%y')
```

Add label column

[5]:
```
df['label'] = df['period'].apply(add_month_year_col)
df
```

[5]:

	buy	hold	period	sell	strongBuy	strongSell	symbol	label
0	36	5	2024-05-01	0	22	0	NVDA	May 24
1	36	5	2024-04-01	0	22	0	NVDA	Apr 24
2	35	5	2024-03-01	0	21	0	NVDA	Mar 24
3	33	5	2024-02-01	0	20	0	NVDA	Feb 24

Sort by period

[6]:
```
# Sort by dates - we plot chart in ascending order
df = df.sort_values('period')
df
```

[6]:

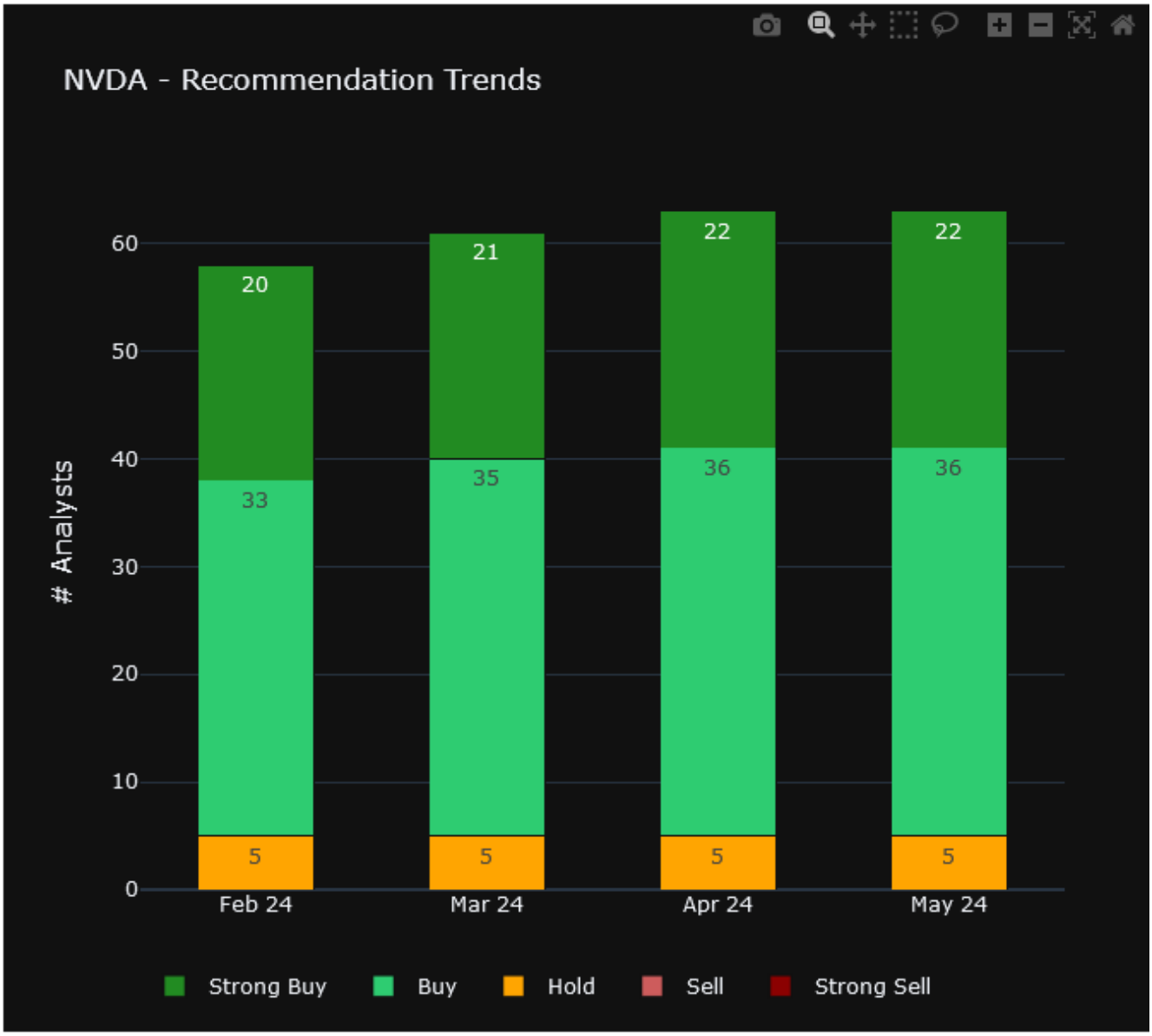
	buy	hold	period	sell	strongBuy	strongSell	symbol	label
3	33	5	2024-02-01	0	20	0	NVDA	Feb 24
2	35	5	2024-03-01	0	21	0	NVDA	Mar 24
1	36	5	2024-04-01	0	22	0	NVDA	Apr 24
0	36	5	2024-05-01	0	22	0	NVDA	May 24

Plot Analyst Trends

```
# Create bar charts
fig = go.Figure(data=[
    go.Bar(name='Strong Sell', x=df.label, y=df.strongSell, text=df.strongSell, marker_color='darkred'),
    go.Bar(name='Sell', x=df.label, y=df.sell, text=df.sell, marker_color='indianred'),
    go.Bar(name='Hold', x=df.label, y=df.hold, text=df.hold, marker_color='orange'),
    go.Bar(name='Buy', x=df.label, y=df.buy, text=df.buy, marker_color='#2ECC71'),
    go.Bar(name='Strong Buy', x=df.label, y=df.strongBuy, text=df.strongBuy, marker_color='forestgreen')
])
# Layout for the plot
layout = go.Layout(
    template=TEMPLATE,
    title=TICKER + ' - Recommendation Trends',
    # Stack bar
    barmode='stack',
    width=700, height=600,
    uniformtext=dict(
        minsize=10,
        mode='hide'
    ),
    xaxis=dict(
        tickfont_size=12
    ),
    yaxis=dict(
        title='# Analysts',
        titlefont_size=16,
        tickfont_size=12,
    ),
    # Horizontal ledger
    legend=dict(
        orientation="h"
    ),
    font=dict(
        size=12
    )
)
fig.update_layout(layout)

# Zero angle for the text inside bar and set the width between bars
fig.update_traces(textangle=0, width=0.5)

fig.show()
```


NVDA - Recommendation Trends
Analysts
60
50
40
30
20
10
0
20
21
22
22
33
35
36
36
5
5
5
5
Feb 24
Mar 24
Apr 24
May 24
Strong Buy
Buy
Hold
Sell
Strong Sell

Compute the Ratings

Reference:Understanding Buy, Sell, and Hold Ratings of Stock Analysts

[9]:
```
df['rating'] = round((df.strongBuy * 1 + df.buy * 2 + df.hold * 3 + df.sell * 4 + df.strongSell * 5)
                     / (df.strongBuy + df.buy + df.hold + df.sell + df.strongSell),2)
df
```

[9]:

	buy	hold	period	sell	strongBuy	strongSell	symbol	label	rating
3	33	5	2024-02-01	0	20	0	NVDA	Feb 24	1.74
2	35	5	2024-03-01	0	21	0	NVDA	Mar 24	1.74
1	36	5	2024-04-01	0	22	0	NVDA	Apr 24	1.73
0	36	5	2024-05-01	0	22	0	NVDA	May 24	1.73

Utlity method to add an horizontal line

10]:
```
# Adds a horizontal dotted line with given annotation text in color
def add_hline(fig, pos, color, text):
    fig.add_hline(y=pos, line_color=color, line_dash='dot',
                  annotation_text=text, annotation_position='top right', annotation_font_size=8, annotation_font_color=cc
```

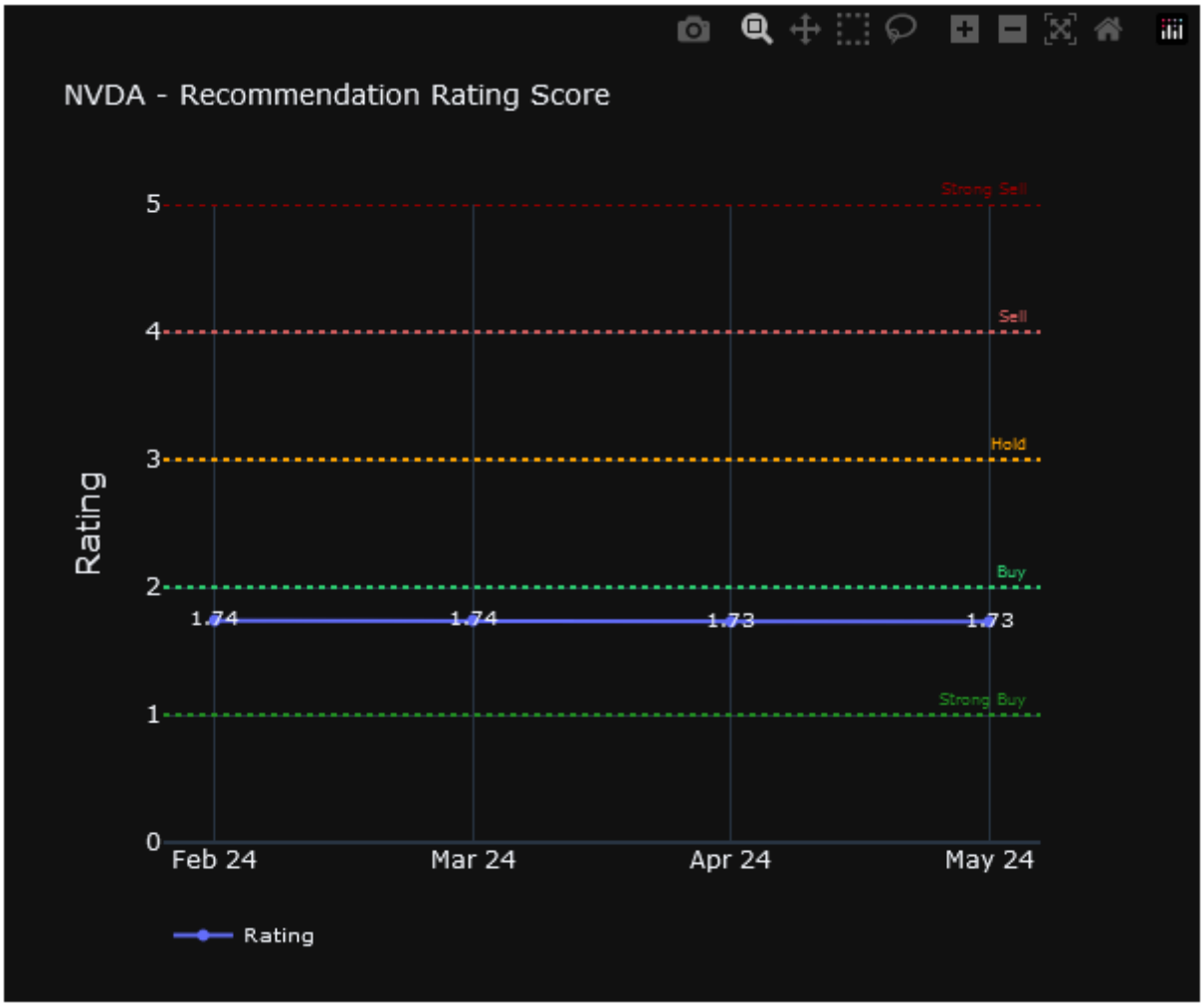
Plot Ratings

[11]:
```
fig = go.Figure()
# Add horizontal lines
add_hline(fig, 5, 'darkred', 'Strong Sell')
add_hline(fig, 4, 'indianred', 'Sell')
add_hline(fig, 3, 'orange', 'Hold')
add_hline(fig, 2, '#2ECC71', 'Buy')
add_hline(fig, 1, 'forestgreen', 'Strong Buy')

# Plot the scores
fig.add_trace(go.Scatter(x=df.label, y=df.rating, name='Rating', showlegend=True,
                         mode='lines+markers+text', text=df.rating))

layout=go.Layout(
    template=TEMPLATE,
    title=TICKER + ' - Recommendation Rating Score',
    xaxis=dict(
        tickfont_size=12
    ),
    width=600, height=500,
    yaxis=dict(
        title='Rating',
        titlefont_size=16,
        tickfont_size=12,
        # Set our own range for y axis
        range=[0,5]
    ),
    legend=dict(
        orientation="h"
    ),
    font=dict(
        size=10
    )
)
fig.update_layout(layout)

fig.show()
```



Files

- Jupyler Notebook file :
- Config File :

Source: https://medium.com/@sugath.mudali/plot-recommendation-trends-from-finnhub-using-plotly-library-for-python-6487a9c9e4ec

---

Revision #8
Created 2024-05-13 10:15:26 UTC by bookstack
Updated 2024-05-13 14:20:10 UTC by bookstack