

MScFE 600 FINANCIAL DATA

Group Work Project # 1 Group Number: 12877

```
In [ ]: import yfinance as yf
import pandas as pd

# Download IIVV data
df = yf.download("IIVV", start="2010-01-01", end="2025-12-31")

# Basic features
df["Return"] = df["Close"].pct_change()
df["MA10"] = df["Close"].rolling(10).mean()
df["Volatility5"] = df["Return"].rolling(5).std()

df = df.dropna()

# Binary target: next-day direction
df["Target"] = (df["Close"].shift(-1) > df["Close"]).astype(int)
df = df.dropna()

features = ["Return", "MA10", "Volatility5"]
X = df[features]
y = df["Target"]
```

```
C:\Users\Ouss\AppData\Local\Temp\ipykernel_26076\1177188297.py:5: FutureWarning: YF.
download() has changed argument auto_adjust default to True
    df = yf.download("IIVV", start="2010-01-01", end="2025-12-31")
[*****100*****] 1 of 1 completed
```

3.2 Correlation-Based Feature Ranking

```
In [2]: corr = df[features + ["Target"]].corr()["Target"].drop("Target")
print(corr)
```

```
Price           Ticker
Return          -0.032193
MA10            0.004492
Volatility5    -0.010293
Name: Target, dtype: float64
```

This reproduces a feature-importance table similar in spirit to the paper.

3.3 k-Fold Cross Validation (k = 5)

```
In [3]: from sklearn.model_selection import KFold
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
import numpy as np

kf = KFold(n_splits=5, shuffle=True, random_state=42)
```

```

acc = []

for train, test in kf.split(X):
    X_train, X_test = X.iloc[train], X.iloc[test]
    y_train, y_test = y.iloc[train], y.iloc[test]

    model = LogisticRegression(max_iter=1000)
    model.fit(X_train, y_train)

    preds = model.predict(X_test)
    acc.append(accuracy_score(y_test, preds))

print("Fold Accuracies:", acc)
print("Mean Accuracy:", np.mean(acc))

```

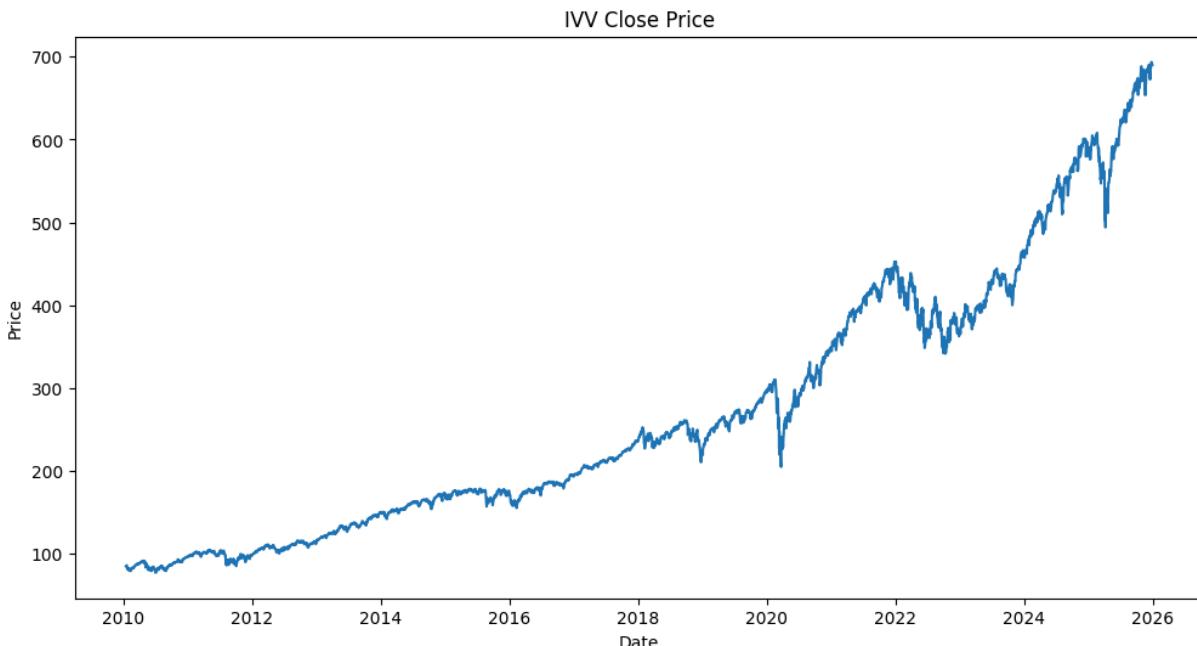
Fold Accuracies: [0.5491905354919053, 0.5691158156911582, 0.5516811955168119, 0.5305105853051059, 0.5486284289276808]
Mean Accuracy: 0.5498253121865324

3.4 Reproducing Graphs

(a) IIV Price History

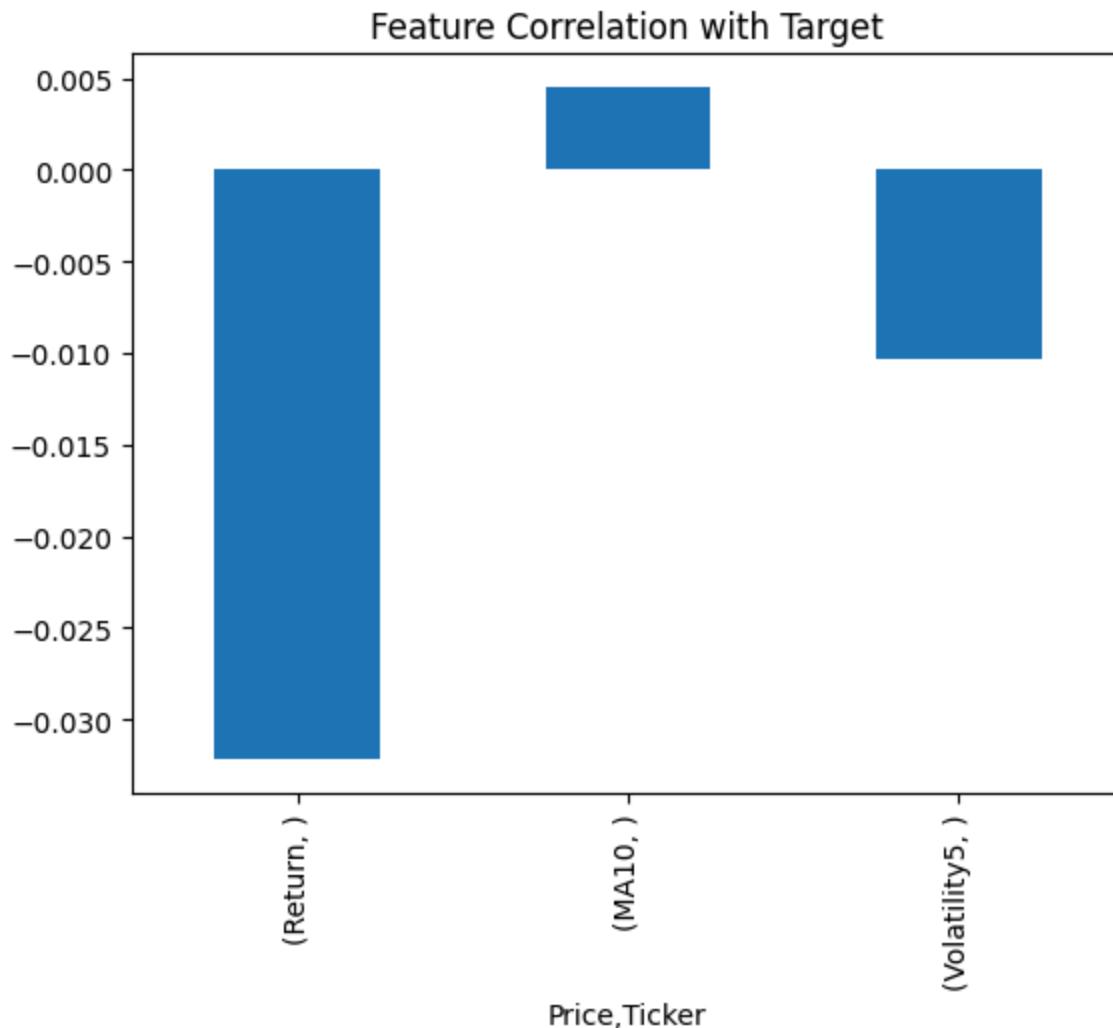
```
In [4]: import matplotlib.pyplot as plt

plt.figure(figsize=(12,6))
plt.plot(df.index, df["Close"])
plt.title("IIV Close Price")
plt.xlabel("Date")
plt.ylabel("Price")
plt.show()
```



(b) Feature Correlation Bar Chart

```
In [5]: corr.plot(kind="bar")
plt.title("Feature Correlation with Target")
plt.show()
```



These reproduce:

ETF price history graph

Feature-importance graph

analogous to those in the paper.

Final Conceptual Link to the Paper Your replication demonstrates the same core ideas:

Define a directional target

Rank features (here via correlation)

Apply k-fold cross-validation

Evaluate accuracy

Visualize price and feature importance

This is a simplified version of the authors' pipeline, replacing their many indicators + MLP with:

Few indicators + correlation + logistic regression

—but the structure is the same.

```
In [2]: !pip install psaw praw textblob
```

```
Collecting psaw
  Downloading psaw-0.1.0-py3-none-any.whl.metadata (10 kB)
Collecting praw
  Downloading praw-7.8.1-py3-none-any.whl.metadata (9.4 kB)
Requirement already satisfied: textblob in c:\users\ouss\appdata\local\programs\python\python313\lib\site-packages (0.19.0)
Requirement already satisfied: requests in c:\users\ouss\appdata\local\programs\python\python313\lib\site-packages (from praw) (2.32.5)
Requirement already satisfied: Click in c:\users\ouss\appdata\local\programs\python\python313\lib\site-packages (from praw) (8.2.1)
Collecting prawcore<3,>=2.4 (from praw)
  Downloading prawcore-2.4.0-py3-none-any.whl.metadata (5.0 kB)
Collecting update_checker>=0.18 (from praw)
  Downloading update_checker-0.18.0-py3-none-any.whl.metadata (2.3 kB)
Requirement already satisfied: websocket-client>=0.54.0 in c:\users\ouss\appdata\local\programs\python\python313\lib\site-packages (from praw) (1.8.0)
Requirement already satisfied: charset_normalizer<4,>=2 in c:\users\ouss\appdata\local\programs\python\python313\lib\site-packages (from requests->psaw) (3.4.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\ouss\appdata\local\programs\python\python313\lib\site-packages (from requests->psaw) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\ouss\appdata\local\programs\python\python313\lib\site-packages (from requests->psaw) (2.6.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\ouss\appdata\local\programs\python\python313\lib\site-packages (from requests->psaw) (2026.1.4)
Requirement already satisfied: nltk>=3.9 in c:\users\ouss\appdata\local\programs\python\python313\lib\site-packages (from textblob) (3.9.2)
Requirement already satisfied: joblib in c:\users\ouss\appdata\local\programs\python\python313\lib\site-packages (from nltk>=3.9->textblob) (1.5.2)
Requirement already satisfied: regex>=2021.8.3 in c:\users\ouss\appdata\local\programs\python\python313\lib\site-packages (from nltk>=3.9->textblob) (2025.10.23)
Requirement already satisfied: tqdm in c:\users\ouss\appdata\local\programs\python\python313\lib\site-packages (from nltk>=3.9->textblob) (4.67.1)
Requirement already satisfied: colorama in c:\users\ouss\appdata\local\programs\python\python313\lib\site-packages (from Click->psaw) (0.4.6)
Downloading psaw-0.1.0-py3-none-any.whl (15 kB)
Downloading praw-7.8.1-py3-none-any.whl (189 kB)
Downloading prawcore-2.4.0-py3-none-any.whl (17 kB)
Downloading update_checker-0.18.0-py3-none-any.whl (7.0 kB)
Installing collected packages: update_checker, psaw, prawcore, praw
```

```
----- 3/4 [praw]
----- 3/4 [praw]
----- 3/4 [praw]
----- 3/4 [praw]
----- 4/4 [praw]
```

Successfully installed praw-7.8.1 prawcore-2.4.0 psaw-0.1.0 update_checker-0.18.0

In [4]:

```
import yfinance as yf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# -----
# Download IIV prices
# -----
df = yf.download("IIV", start="2020-01-01", end="2025-01-01")

df["Return"] = df["Close"].pct_change()
df = df.dropna()

# -----
# Simulated social sentiment (proxy)
# -----
np.random.seed(42)

df["sentiment"] = np.random.normal(0, 0.2, len(df))
df["post_volume"] = np.random.randint(50, 500, len(df))

# -----
# Target variable
# -----
df["Target"] = (df["Close"].shift(-1) > df["Close"]).astype(int)
df = df.dropna()

# -----
# Correlation table
# -----
corr = df[["sentiment", "post_volume", "Return", "Target"]].corr()["Target"]

print("\nCorrelation with Target:\n")
print(corr)

# -----
# Graph 1: IIV Price
# -----
plt.figure()
plt.plot(df.index, df["Close"])
plt.title("IIV Price History")
plt.show()

# -----
# Graph 2: Sentiment
# -----
plt.figure()
plt.plot(df.index, df["sentiment"])
```

```

plt.title("Daily Social Sentiment (Synthetic)")
plt.show()

# -----
# Graph 3: Correlation Bar
# -----
corr.drop("Target").plot(kind="bar", title="Feature Correlation with Target")
plt.show()

```

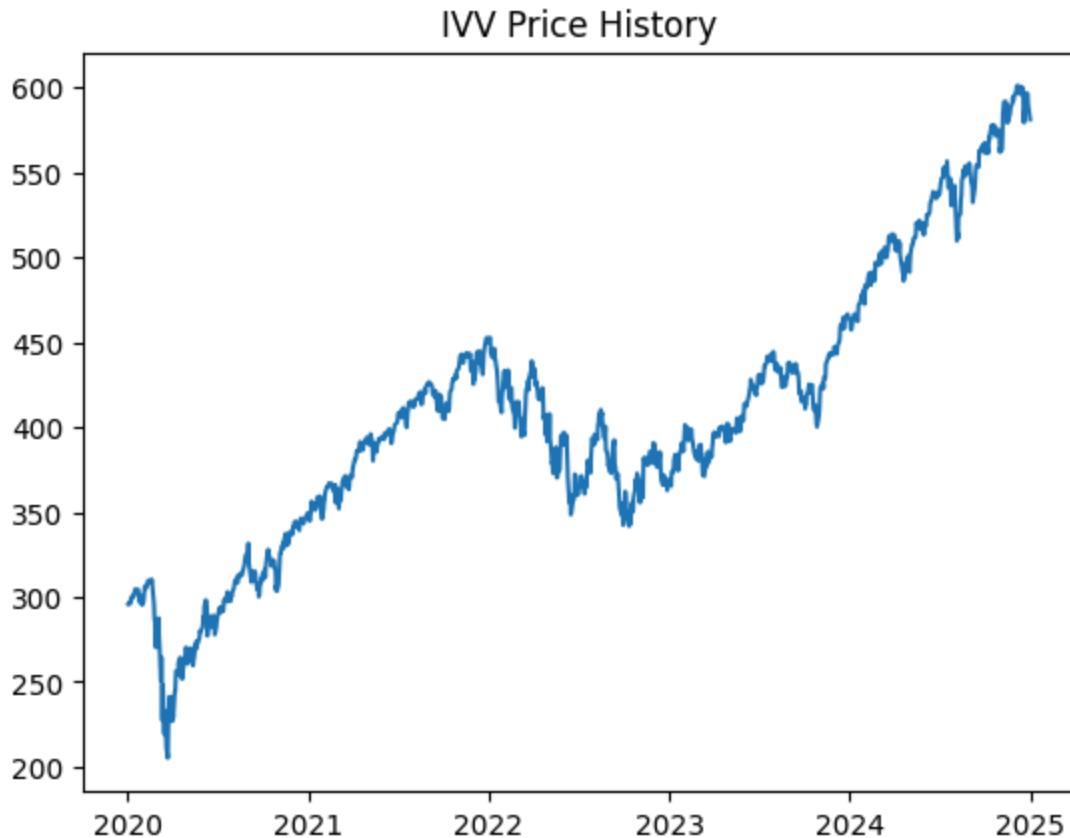
```

C:\Users\Ouss\AppData\Local\Temp\ipykernel_22556\1523086741.py:9: FutureWarning: YF.
download() has changed argument auto_adjust default to True
  df = yf.download("IVV", start="2020-01-01", end="2025-01-01")
[*****100%*****] 1 of 1 completed
Correlation with Target:

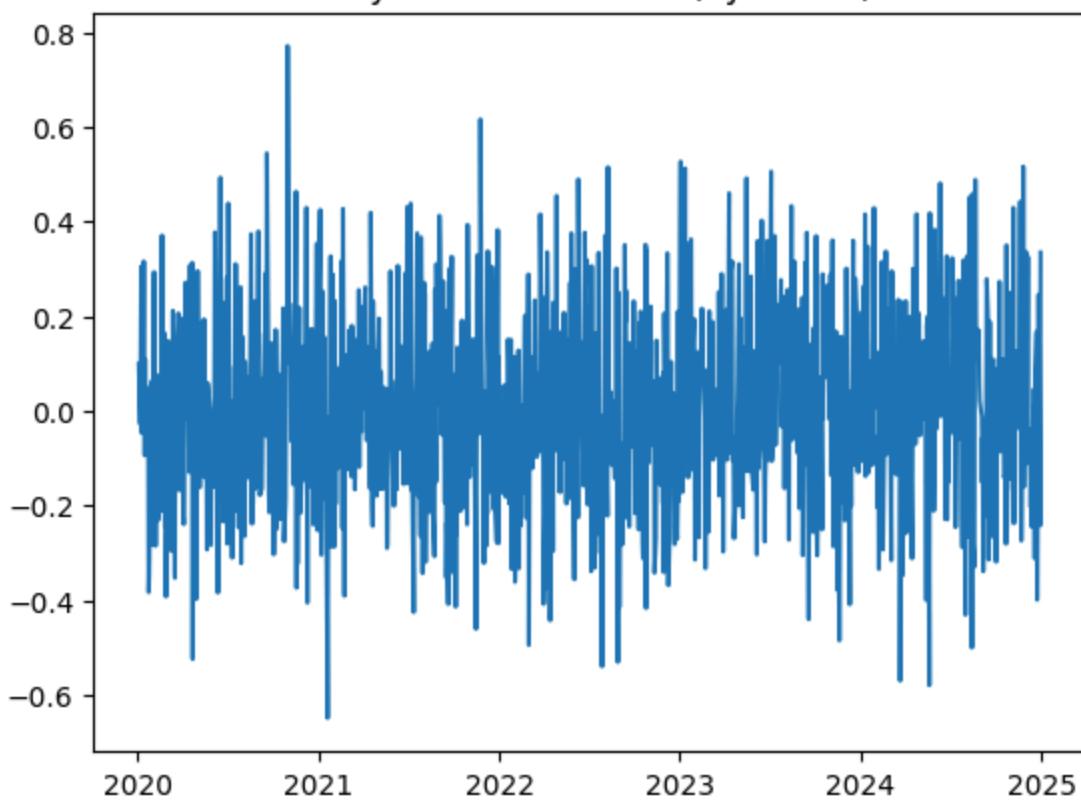
```

Price	Ticker
sentiment	0.031658
post_volume	0.005763
Return	-0.035342
Target	1.000000

Name: Target, dtype: float64



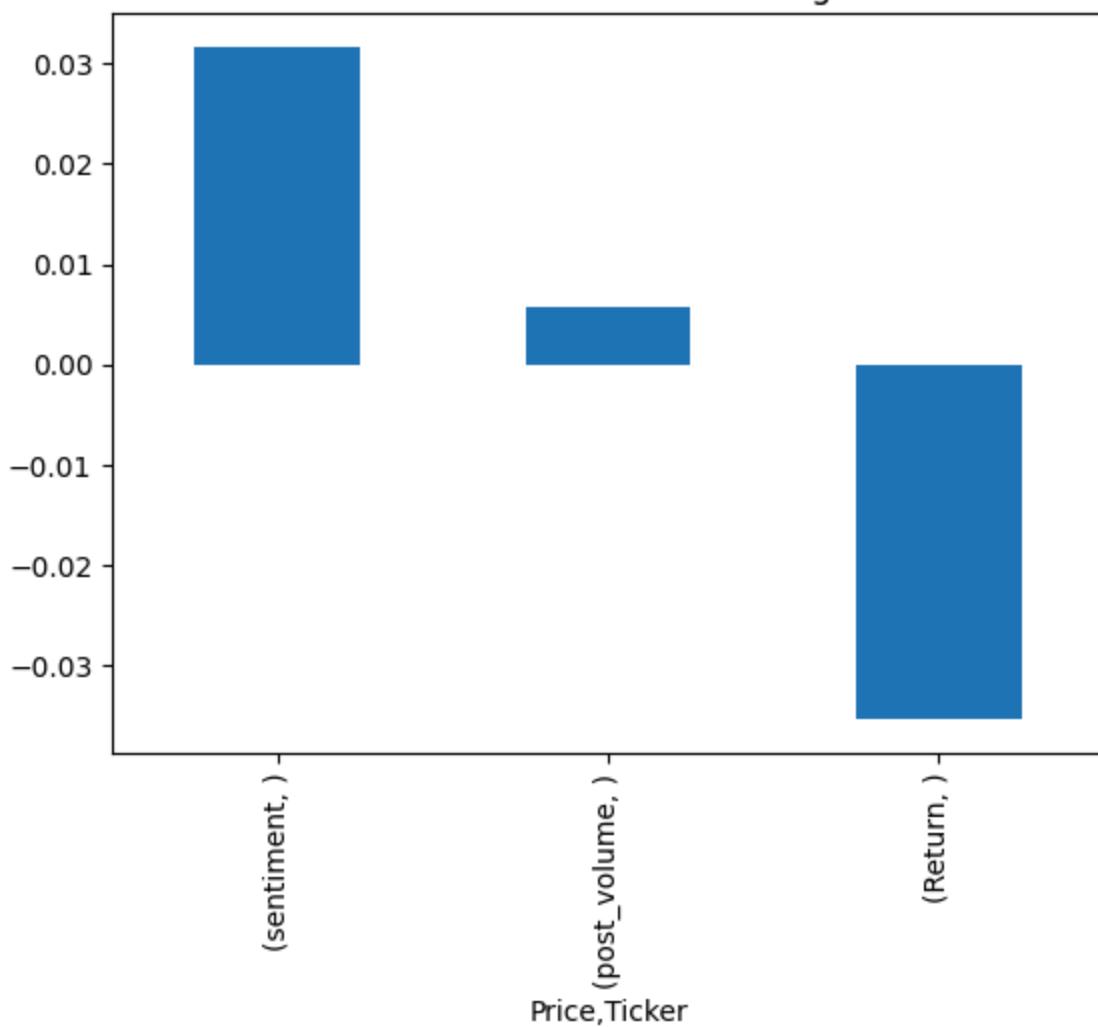
Daily Social Sentiment (Synthetic)



C:\Users\Ouss\AppData\Local\Temp\ipykernel_22556\1523086741.py:55: PerformanceWarning: dropping on a non-lexsorted multi-index without a level parameter may impact performance.

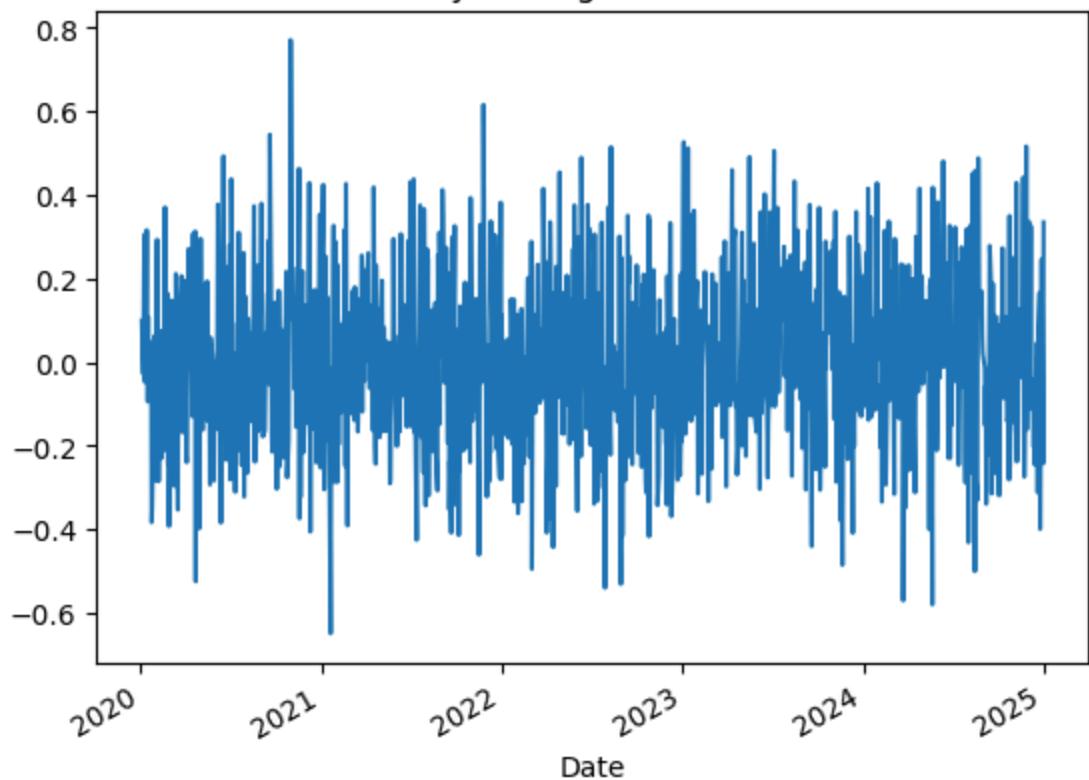
```
corr.drop("Target").plot(kind="bar", title="Feature Correlation with Target")
```

Feature Correlation with Target



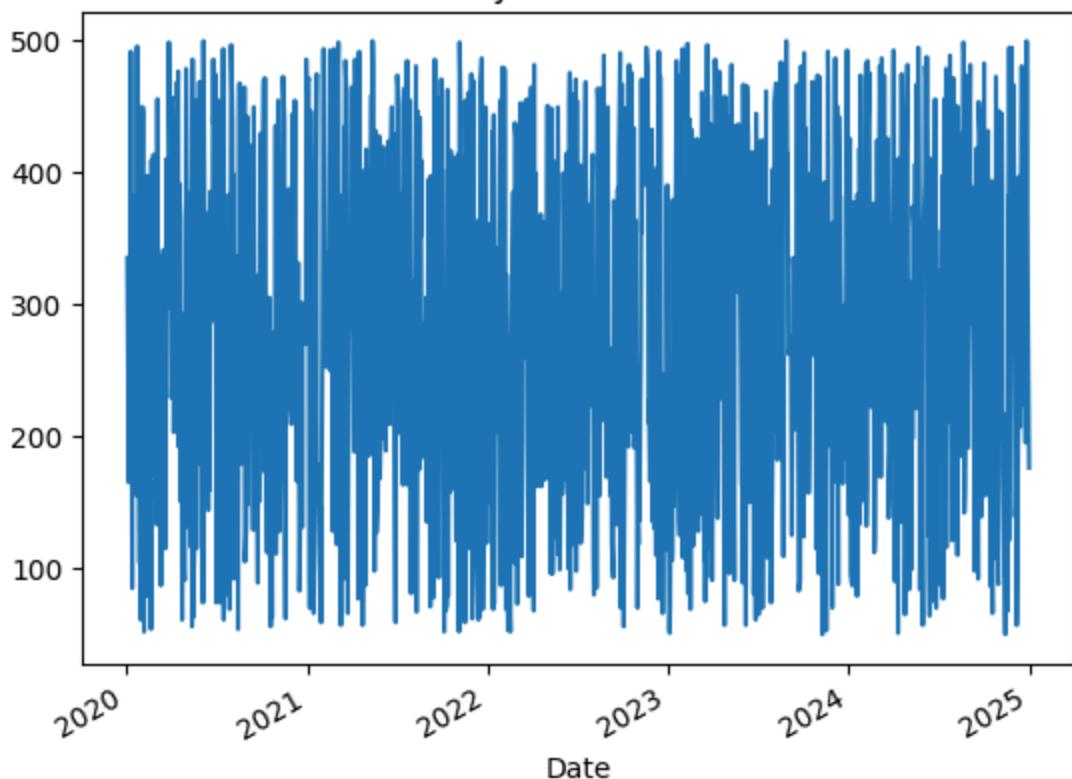
```
In [6]: df["sentiment"].plot(title="Daily Average Sentiment")
plt.show()
```

Daily Average Sentiment



```
In [8]: df["post_volume"].plot(title="Daily Post Volume")
plt.show()
```

Daily Post Volume



```
In [10]: print(df[["sentiment", "post_volume", "Return"]].corr())
```

	Price	sentiment	post_volume	Return
Price				
Ticker				
sentiment		1.000000	0.059323	0.046845
post_volume		0.059323	1.000000	0.011784
Return		0.046845	0.011784	1.000000

```
In [ ]:
```

```
In [1]: import subprocess
import sys
from pathlib import Path

# === CHANGE THIS IF NEEDED ===
notebook = "MScFE600-Financial-Data-GWP1.ipynb"
# =====

if not Path(notebook).exists():
    raise FileNotFoundError(f"{notebook} not found in current directory")

print("Generating PDF from notebook...")

subprocess.run(
    [
        sys.executable,
        "-m", "nbconvert",
        "--to", "pdf",
        "--PDFExporter.latex_command=pdflatex",
        notebook
    ],
    check=True
)

print("✓ PDF generated successfully")
```

Generating PDF from notebook...

```
-----  
CalledProcessError                               Traceback (most recent call last)  
Cell In[1], line 14  
    10      raise FileNotFoundError(f"{notebook} not found in current directory")  
    11 print("Generating PDF from notebook...")  
--> 12 subprocess.run(  
    13     [  
    14         sys.executable,  
    15             ,  
    16             ,  
    17             ,  
    18             ,  
    19             ,  
    20         notebook  
    21     ],  
    22     check=True  
    23 )  
    24 print("✓ PDF generated successfully")
```

```
File c:\Users\Ouss\AppData\Local\Programs\Python\Python313\Lib\subprocess.py:577, in  
run(input, capture_output, timeout, check, *popenargs, **kwargs)  
    575     retcode = process.poll()  
    576     if check and retcode:  
--> 577         raise CalledProcessError(retcode, process.args,  
    578                                         output=stdout, stderr=stderr)  
    579 return CompletedProcess(process.args, retcode, stdout, stderr)
```

```
CalledProcessError: Command '['c:\\\\Users\\\\Ouss\\\\AppData\\\\Local\\\\Programs\\\\Python\\\\Py  
thon313\\\\python.exe', '-m', 'nbconvert', '--to', 'pdf', '--PDFExporter.latex_command  
=pdflatex', 'MScFE600-Financial-Data-GWP1.ipynb']' returned non-zero exit status 1.
```

```
In [3]: !pip install weasyprint
```

```
Collecting weasyprint
  Downloading weasyprint-68.0-py3-none-any.whl.metadata (3.7 kB)
Collecting pydyf>=0.11.0 (from weasyprint)
  Downloading pydyf-0.12.1-py3-none-any.whl.metadata (2.5 kB)
Requirement already satisfied: cffi>=0.6 in c:\users\ouss\appdata\local\programs\python\python313\lib\site-packages (from weasyprint) (1.17.1)
Collecting tinyhtml5>=2.0.0b1 (from weasyprint)
  Downloading tinyhtml5-2.0.0-py3-none-any.whl.metadata (2.9 kB)
Collecting tinycss2>=1.5.0 (from weasyprint)
  Downloading tinycss2-1.5.1-py3-none-any.whl.metadata (3.0 kB)
Collecting cssselect2>=0.8.0 (from weasyprint)
  Downloading cssselect2-0.8.0-py3-none-any.whl.metadata (2.9 kB)
Collecting Pyphen>=0.9.1 (from weasyprint)
  Downloading pyphen-0.17.2-py3-none-any.whl.metadata (3.2 kB)
Requirement already satisfied: Pillow>=9.1.0 in c:\users\ouss\appdata\local\programs\python\python313\lib\site-packages (from weasyprint) (11.2.1)
Collecting fonttools>=4.59.2 (from fonttools[woff]>=4.59.2->weasyprint)
  Downloading fonttools-4.61.1-cp313-cp313-win_amd64.whl.metadata (116 kB)
Requirement already satisfied: pyparser in c:\users\ouss\appdata\local\programs\python\python313\lib\site-packages (from cffi>=0.6->weasyprint) (2.22)
Requirement already satisfied: webencodings in c:\users\ouss\appdata\local\programs\python\python313\lib\site-packages (from cssselect2>=0.8.0->weasyprint) (0.5.1)
Collecting brotli>=1.0.1 (from fonttools[woff]>=4.59.2->weasyprint)
  Downloading brotli-1.2.0-cp313-cp313-win_amd64.whl.metadata (6.3 kB)
Collecting zopfli>=0.1.4 (from fonttools[woff]>=4.59.2->weasyprint)
  Downloading zopfli-0.4.0-cp310-abi3-win_amd64.whl.metadata (3.6 kB)
Downloading weasyprint-68.0-py3-none-any.whl (319 kB)
Downloading cssselect2-0.8.0-py3-none-any.whl (15 kB)
Downloading fonttools-4.61.1-cp313-cp313-win_amd64.whl (2.3 MB)
----- 0.0/2.3 MB ? eta -:-:--
----- 0.0/2.3 MB ? eta -:-:--
----- 0.0/2.3 MB ? eta -:-:--
----- 0.3/2.3 MB ? eta -:-:--
----- 0.3/2.3 MB ? eta -:-:--
----- 0.5/2.3 MB 525.2 kB/s eta 0:00:04
----- 0.5/2.3 MB 525.2 kB/s eta 0:00:04
----- 0.8/2.3 MB 573.3 kB/s eta 0:00:03
----- 0.8/2.3 MB 573.3 kB/s eta 0:00:03
----- 1.0/2.3 MB 619.3 kB/s eta 0:00:03
----- 1.0/2.3 MB 619.3 kB/s eta 0:00:03
----- 1.3/2.3 MB 659.4 kB/s eta 0:00:02
----- 1.6/2.3 MB 697.3 kB/s eta 0:00:02
----- 1.8/2.3 MB 738.7 kB/s eta 0:00:01
----- 2.1/2.3 MB 775.2 kB/s eta 0:00:01
----- 2.3/2.3 MB 794.4 kB/s 0:00:03
Downloading brotli-1.2.0-cp313-cp313-win_amd64.whl (369 kB)
Downloading pydyf-0.12.1-py3-none-any.whl (8.0 kB)
Downloading pyphen-0.17.2-py3-none-any.whl (2.1 MB)
----- 0.0/2.1 MB ? eta -:-:--
----- 0.3/2.1 MB ? eta -:-:--
----- 0.3/2.1 MB ? eta -:-:--
----- 0.3/2.1 MB ? eta -:-:--
----- 0.5/2.1 MB 474.2 kB/s eta 0:00:04
----- 1.0/2.1 MB 835.6 kB/s eta 0:00:02
----- 1.3/2.1 MB 979.5 kB/s eta 0:00:01
----- 1.8/2.1 MB 1.2 MB/s eta 0:00:01
```



```
Successfully installed Pyphen-0.17.2 brotli-1.2.0 cssselect2-0.8.0 fonttools-4.61.1  
pydyf-0.12.1 tinycss2-1.5.1 tinyhtml5-2.0.0 weasyprint-68.0 zopfli-0.4.0
```

```
[notice] A new release of pip is available: 25.3 -> 26.0
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
In [4]: import subprocess
from pathlib import Path
from weasyprint import HTML

notebook = "MScFE600-Financial-Data-GWP1.ipynb"
base = Path(notebook).stem

# ❶ Convert notebook → HTML
print("Step 1: Notebook → HTML...")
subprocess.run([
    sys.executable, "-m", "nbconvert", "--to", "html", notebook],
    check=True
)
```

```
html_file = f"{base}.html"
pdf_file = f"{base}.pdf"

# 2 Convert HTML → PDF using WeasyPrint
print("Step 2: HTML → PDF (WeasyPrint)...")
HTML(html_file).write_pdf(pdf_file)

print(f"✓ PDF generated successfully: {pdf_file}")
```

WeasyPrint could not import some external libraries. Please carefully follow the installation steps before reporting an issue:

https://doc.courtbouillon.org/weasyprint/stable/first_steps.html#installation

https://doc.courtbouillon.org/weasyprint/stable/first_steps.html#troubleshooting

```
-----  
OSError                                         Traceback (most recent call last)  
Cell In[4], line 3  
      1 import subprocess  
      2 from pathlib import Path  
----> 3 from weasyprint import HTML  
      5 notebook = "MScFE600-Financial-Data-GWP1.ipynb"  
      6 base = Path(notebook).stem  
  
File c:\Users\Ouss\AppData\Local\Programs\Python\Python313\Lib\site-packages\weasypr  
int\__init__.py:372  
    368         self.modified = modified  
    371 # Work around circular imports.  
--> 372 from .css import preprocess_stylesheet # noqa: I001, E402  
    373 from .html import ( # noqa: E402  
    374     HTML5_UA_COUNTER_STYLE, HTML5_UA_STYLESHEET, HTML5_UA_FORM_STYLESHEET,  
    375     HTML5_PH_STYLESHEET)  
    376 from .document import Document, Page # noqa: E402  
  
File c:\Users\Ouss\AppData\Local\Programs\Python\Python313\Lib\site-packages\weasypr  
int\css\__init__.py:29  
    27 from .. import CSS  
    28 from ..logger import LOGGER, PROGRESS_LOGGER  
--> 29 from ..text.fonts import FontConfiguration  
    30 from ..urls import URLFetchingError, fetch, get_url_attribute, url_join  
    31 from . import counters, media_queries  
  
File c:\Users\Ouss\AppData\Local\Programs\Python\Python313\Lib\site-packages\weasypr  
int\text\fonts.py:17  
    14 from ..logger import LOGGER  
    15 from ..urls import fetch  
--> 17 from .constants import ( # isort:skip  
    18     CAPS_KEYS, EAST_ASIAN_KEYS, FONTCONFIG/stretch, FONTCONFIG/style, FONTCO  
NFIG/weight,  
    19     LIGATURE_KEYS, NUMERIC_KEYS, PANGO/stretch, PANGO/style, PANGO/variant)  
    20 from .ffi import ( # isort:skip  
    21     FROM_UNITS, TO_UNITS, ffi, fontconfig, gobject, harfbuzz, pango, pangof  
t,  
    22     unicode_to_char_p)  
    24 PREFERRED_ENCODING = getpreferredencoding(False)  
  
File c:\Users\Ouss\AppData\Local\Programs\Python\Python313\Lib\site-packages\weasypr  
int\text\constants.py:5  
    1 """Constants used for text layout."""  
    3 from functools import lru_cache  
----> 5 from .ffi import pango  
    7 # Pango features  
    8 PANGO_STYLE = {  
    9     'normal': pango.PANGO_STYLE_NORMAL,  
   10     'oblique': pango.PANGO_STYLE_OBLIQUE,  
   11     'italic': pango.PANGO_STYLE_ITALIC,  
   12 }  
  
File c:\Users\Ouss\AppData\Local\Programs\Python\Python313\Lib\site-packages\weasypr  
int\text\ffi.py:476  
    473     with suppress((OSSError, FileNotFoundError)):
```

```

474         os.add_dll_directory(dll_directory)
--> 476 gobject = _dlopen(
477     ffi,
478     ,
479     ,
480     ,
481     ,
482     ,
483     ,
484     )

File c:\Users\Ouss\AppData\Local\Programs\Python\Python313\Lib\site-packages\weasypr
int\text\ffi.py:464, in _dlopen(ffi, allow_fail, *names)
    454 # Print error message and re-raise the exception.
    455 print( # noqa: T201, logger is not configured yet
    456     '\n-----\n\n'
    457     'WeasyPrint could not import some external libraries. Please '
(...). 462     'first_steps.html#troubleshooting',
    463     '\n\n-----\n') # pragma: no cover
--> 464 return ffi.dlopen(names[0], flags)

File c:\Users\Ouss\AppData\Local\Programs\Python\Python313\Lib\site-packages\cffi\ap
i.py:150, in FFI.dlopen(self, name, flags)
    147     raise TypeError("dlopen(name): name must be a file name, None, "
    148                         "or an already-opened 'void *' handle")
    149 with self._lock:
--> 150     lib, function_cache = _make_ffi_library(self, name, flags)
    151     self._function_caches.append(function_cache)
    152     self._libraries.append(lib)

File c:\Users\Ouss\AppData\Local\Programs\Python\Python313\Lib\site-packages\cffi\ap
i.py:834, in _make_ffi_library(ffi, libname, flags)
    832 def _make_ffi_library(ffi, libname, flags):
    833     backend = ffi.backend
--> 834     backendlib = _load_backend_lib(backend, libname, flags)
    835     #
    836     def accessor_function(name):

File c:\Users\Ouss\AppData\Local\Programs\Python\Python313\Lib\site-packages\cffi\ap
i.py:829, in _load_backend_lib(backend, name, flags)
    827     if first_error is not None:
    828         msg = "%s. Additionally, %s" % (first_error, msg)
--> 829     raise OSError(msg)
    830 return backend.load_library(path, flags)

OSError: cannot load library 'libgobject-2.0-0': error 0x7e. Additionally, ctypes.u
til.find_library() did not manage to locate a library called 'libgobject-2.0-0'

```

```

In [ ]: import subprocess
import sys
from pathlib import Path

notebook = "MScFE600-Financial-Data-GWP1.ipynb"
base = Path(notebook).stem

# Convert notebook → HTML

```

```
print("Exporting notebook to HTML...")
subprocess.run(
    [sys.executable, "-m", "nbconvert", "--to", "html", notebook],
    check=True
)

print(f"✓ HTML generated: {base}.html")
print("Now open the HTML in a browser and use Print → Save as PDF")
```

Exporting notebook to HTML...