**Objective**

Use historical price trends and recent news sentiment to predict which **5 Dow Jones stocks** are most likely to go **up** and which are likely to go **down** tomorrow (or over a short-term period).

**🧩 Step-by-Step Blueprint**

**1. Get the DOW Jones Stocks**

You can hard-code or scrape the 30 DOW tickers (like AAPL, MSFT, JPM, etc.).

python

CopyEdit

dow\_tickers = ['AAPL', 'MSFT', 'JPM', 'JNJ', 'V', 'PG', 'KO', 'INTC', 'HD', 'IBM', 'DIS', ...]

**2. Pull Stock Data**

Use yfinance for historical prices + calculate technical indicators:

python

CopyEdit

import yfinance as yf

import ta

def get\_stock\_data(ticker):

df = yf.download(ticker, period="6mo", interval="1d")

df['Return'] = df['Close'].pct\_change()

df['RSI'] = ta.momentum.RSIIndicator(df['Close']).rsi()

df['MACD'] = ta.trend.MACD(df['Close']).macd\_diff()

return df

**3. Pull News Sentiment**

Use **Finnhub** or **NewsAPI** to get recent news headlines for each stock, then analyze sentiment.

python

CopyEdit

import requests

def get\_news\_sentiment(ticker):

url = f"https://finnhub.io/api/v1/company-news?symbol={ticker}&from=2024-04-01&to=2024-04-10&token=YOUR\_API\_KEY"

news = requests.get(url).json()

# Basic sentiment scoring using keywords or pre-trained model

sentiment\_score = 0

for article in news:

text = (article['headline'] + " " + article['summary']).lower()

if "beat" in text or "strong" in text:

sentiment\_score += 1

elif "miss" in text or "weak" in text:

sentiment\_score -= 1

return sentiment\_score

Or use transformers for more accurate sentiment analysis.

**4. Combine Features**

For each ticker, create a feature vector:

* RSI
* MACD
* 5-day return
* News sentiment

python

CopyEdit

def generate\_features(ticker):

df = get\_stock\_data(ticker)

sentiment = get\_news\_sentiment(ticker)

latest = df.dropna().iloc[-1]

return {

'Ticker': ticker,

'RSI': latest['RSI'],

'MACD': latest['MACD'],

'5d\_return': df['Close'].pct\_change(5).iloc[-1],

'Sentiment': sentiment

}

**5. Build a Model**

You can either:

* Train a **classification model** to predict up/down (if you have labeled data)
* Or use a **scoring approach**: rank stocks based on feature-weighted scores

**6. Rank & Predict**

python

CopyEdit

import pandas as pd

feature\_list = [generate\_features(t) for t in dow\_tickers]

df = pd.DataFrame(feature\_list)

# Simple score: you can refine this with ML

df['Score'] = df['5d\_return'] + df['MACD'] + 0.01 \* df['Sentiment']

top\_5\_up = df.sort\_values(by='Score', ascending=False).head(5)

top\_5\_down = df.sort\_values(by='Score', ascending=True).head(5)

**✅ Output**

Print or visualize your predicted top 5 gainers and losers.

python

CopyEdit

print("📈 Top 5 Predicted Gainers:")

print(top\_5\_up[['Ticker', 'Score']])

print("\n📉 Top 5 Predicted Losers:")

print(top\_5\_down[['Ticker', 'Score']])