GitHub Repository Link : https://github.com/ramindersinghusd/aai-520-in3-project/blob/main/final_project/

In [2]:

```
# ============================================================
# 🧭 University of San Diego - NLP & GenAI (AAI-520-IN3)
# Final Project: Multi-Agent Investment Research Report
# Authors: Richa Arun Kumar Jha, Raminder Singh, Samiksha Kodgire
# Instructor: Premkumar Chithaluru, Ph.D
# Date: 17 October 2025
# ============================================================

import warnings, numpy as np, pandas as pd, yfinance as yf, requests, torch, matplotlib.pyplot as plt, nltk
from transformers import pipeline, AutoTokenizer, AutoModelForCausalLM
from nltk.corpus import stopwords
from IPython.display import Markdown, display, HTML

warnings.filterwarnings("ignore")
pd.set_option("display.max_colwidth", 80)
pd.set_option("display.width", 80)
plt.rcParams["figure.figsize"] = (7, 5)
plt.rcParams["savefig.bbox"] = "tight"

nltk.download("punkt")
nltk.download("stopwords")

# ============================================================
# 🎓 Title & Abstract
# ============================================================
display(Markdown("""
# 🧭 University of San Diego
## Natural Language Processing and GenAI (AAI-520-IN3)
### Final Project: Multi-Agent Investment Research Report

**Authors:** Richa Arun Kumar Jha, Raminder Singh, Samiksha Kodgire
**Instructor:** Premkumar Chithaluru, Ph.D
**Date:** 16 October 2025

---

### **Abstract**
This project presents a comprehensive AI-driven investment research system that integrates quantitative analysis, sentiment evaluation, and generative summarization through a **multi-agent architecture**.
Agents autonomously collect market data, perform risk and technical analyses, classify news sentiment using **DistilBERT**, and summarize insights using **Phi-3 Mini**.
The workflow covers six major technology tickers — AAPL, GOOG, TSLA, AMZN, NVDA, and MSFT — producing financial summaries, correlation analysis, and portfolio suggestions.
---
"""))

# ============================================================
# ⚙️ Agentic Workflow Overview
# ============================================================
display(Markdown("""
## ⚙️ Agentic Workflow Overview
1. **DataAgent** — Fetches market data from Yahoo Finance.
2. **TechnicalAgent** — Computes SMA20, SMA50, RSI, and momentum signal.
3. **RiskAgent** — Evaluates volatility, drawdown, and composite risk score.
4. **NewsAgent** — Retrieves live financial headlines from NewsAPI.
5. **SentimentAgent** — Classifies news into Positive / Neutral / Negative using DistilBERT.
6. **RoutingAgent** — Routes news to earnings, macro, or product categories.
7. **PortfolioAgent** — Suggests risk-aligned asset allocation.
8. **EvaluationAgent** — Compares sentiment trends vs. price action.
9. **LLMOptimizerAgent** — Generates professional summaries using Phi-3 Mini.
---
"""))

# ============================================================
# 🔧 Setup
# ============================================================
device = "cuda" if torch.cuda.is_available() else "cpu"
print(f"✅ Device: {device}")

# load keys from Colab secrets
from google.colab import userdata
NEWS_API_KEY = userdata.get("NEWS_API_KEY")
```

```python
# models
sentiment_model = pipeline("sentiment-analysis",
                           model="distilbert-base-uncased-finetuned-sst-2-english",
                           device=0 if device=="cuda" else -1)
model_name = "microsoft/Phi-3-mini-4k-instruct"
model = AutoModelForCausalLM.from_pretrained(model_name, torch_dtype=torch.float16, device_map="auto")
tokenizer = AutoTokenizer.from_pretrained(model_name)
llm = pipeline("text-generation", model=model, tokenizer=tokenizer)

# ============================================================
# 📋 Helper
# ============================================================
def md_table(df, title=None):
    """Render compact Markdown tables that print correctly on PDF."""
    txt = ""
    if title: txt += f"\n### {title}\n\n"
    txt += df.to_markdown(index=False, tablefmt="github")
    display(Markdown(txt))

# ============================================================
# 🤖 Agents
# ============================================================
def fetch_data(ticker, period="1y"):
    return yf.download(ticker, period=period, progress=False).reset_index()

def compute_technicals(data):
    """Compute SMA20, SMA50, RSI, and signal — guaranteed scalar-safe."""
    if data is None or data.empty or len(data) < 50:
        return {"SMA20": np.nan, "SMA50": np.nan, "RSI": 50.0, "Signal": "Neutral"}

    close = data["Close"].fillna(method="ffill").fillna(method="bfill")

    sma20 = close.rolling(20).mean().iloc[-1]
    sma50 = close.rolling(50).mean().iloc[-1]

    delta = close.diff()
    gain = delta.where(delta > 0, 0).rolling(14).mean()
    loss = -delta.where(delta < 0, 0).rolling(14).mean()
    rs = gain / loss

    last_loss = float(loss.iloc[-1:].values[0]) if not np.isnan(loss.iloc[-1:].values[0]) else 0.0
    last_rs = float(rs.iloc[-1:].values[0]) if not np.isnan(rs.iloc[-1:].values[0]) else 0.0

    if last_loss == 0 or np.isnan(last_rs):
        rsi = 50.0
    else:
        rsi = 100 - (100 / (1 + last_rs))

    signal = "Bullish" if float(sma20) > float(sma50) else "Bearish"

    return {
        "SMA20": float(sma20),
        "SMA50": float(sma50),
        "RSI": float(rsi),
        "Signal": signal,
    }

def compute_risk(data):
    """Compute volatility, drawdown, and risk score — robust to NaN or short series."""
    if data is None or data.empty or len(data) < 2:
        return {"Volatility": np.nan, "Drawdown": np.nan, "RiskScore": np.nan}

    close = data["Close"].fillna(method="ffill").fillna(method="bfill")
    ret = close.pct_change().dropna()

    vol = float(ret.std()) if not ret.empty else 0.0
    dd_series = (close / close.cummax() - 1)
    dd = float(dd_series.min()) if not dd_series.empty else 0.0

    score = abs(vol * 1000 + dd * 100)
    return {
        "Volatility": vol,
        "Drawdown": dd,
        "RiskScore": round(score, 2),
    }

def fetch_news(ticker):
    url = f"https://newsapi.org/v2/everything?q={ticker}&apiKey={NEWS_API_KEY}&language=en&sortBy=publishedAt&pageSize=5"
    try:
        r = requests.get(url, timeout=10)
        return [a["title"] for a in r.json().get("articles", [])] or ["No news available."]
    except Exception as e:
        return [f"Error fetching news: {e}"]

def classify_sentiment(headlines):
    results = sentiment_model(headlines)
    counts = {"positive":0,"neutral":0,"negative":0}
    for r in results:
        if r["label"] == "POSITIVE": counts["positive"]+=1
        else: counts["negative"]+=1
    total = sum(counts.values()) or 1
    for k in counts: counts[k] = round(100*counts[k]/total, 1)
    return counts

def route_news(headlines):
    routes = {"earnings":[],"macro":[],"product":[]}
    for h in headlines:
        hl = h.lower()
        if any(k in hl for k in ["earnings","profit","revenue"]): routes["earnings"].append(h)
        elif any(k in hl for k in ["inflation","market","economy","rates"]): routes["macro"].append(h)
        else: routes["product"].append(h)
    return routes

def suggest_portfolio(risk):
    if risk < 30: return {"Equity":0.7,"Bonds":0.2,"Cash":0.1}
    elif risk < 50: return {"Equity":0.6,"Bonds":0.3,"Cash":0.1}
    else: return {"Equity":0.4,"Bonds":0.4,"Cash":0.2}

def compare_sentiment_vs_price(sent, data):
    try:
        if data is None or data.empty:
            return "Unknown"
        close = data["Close"].dropna().reset_index(drop=True)
        if len(close) < 2:
            return "Unknown"
        first_price = float(close.iloc[0])
        last_price = float(close.iloc[-1])
        up = last_price > first_price
        pos = float(sent.get("positive", 0))
        neg = float(sent.get("negative", 0))
        sentiment_up = pos > neg
        return "Aligned" if up == sentiment_up else "Divergent"
    except Exception as e:
        print(f"⚠️ compare_sentiment_vs_price() failed: {e}")
        return "Unknown"

def generate_summary(ticker, tech, risk, sent, port, news):
    prompt = f"""
Write a concise 2-paragraph investment summary for {ticker}.
Integrate:
Technicals: {tech}
Risk: {risk}
Sentiment: {sent}
Portfolio: {port}
News: {'; '.join(news)}
End with a recommendation.
"""
    return llm(prompt, max_new_tokens=300, temperature=0.6)[0]["generated_text"]

# ============================================================
# 🚀 Run the Multi-Agent Report
# ============================================================
def run_report(tickers):
    radar_data = []

    for t in tickers:
        display(Markdown(f"---\n# 📊 {t} Analysis Page\n"))
        data = fetch_data(t)
        tech = compute_technicals(data)
        risk = compute_risk(data)
        news = fetch_news(t)
        sent = classify_sentiment(news)
        routes = route_news(news)
        port = suggest_portfolio(risk["RiskScore"])
        corr = compare_sentiment_vs_price(sent, data)
        summary = generate_summary(t, tech, risk, sent, port, news)

        # --- Price + SMA Chart ---
        fig, ax1 = plt.subplots(figsize=(10, 4))
        ax1.plot(data["Close"], label="Close", color="blue", linewidth=1.5)
        ax1.plot(data["Close"].rolling(20).mean(), label="SMA20", linestyle="--", color="orange")
        ax1.plot(data["Close"].rolling(50).mean(), label="SMA50", linestyle="-.", color="green")
        ax1.set_title(f"{t} Price & Moving Averages", fontsize=14)
        ax1.set_xlabel("Date")
        ax1.set_ylabel("Price ($)")
        ax1.legend()
        ax1.grid(True)
        plt.tight_layout()
        plt.show()

        # --- RSI Chart ---
        delta = data["Close"].diff()
        gain = delta.where(delta > 0, 0).rolling(14).mean()
        loss = -delta.where(delta < 0, 0).rolling(14).mean()
        rs = gain / loss
        rsi = 100 - (100 / (1 + rs))

        plt.figure(figsize=(10, 2))
        plt.plot(rsi, label="RSI(14)", color="purple")
        plt.axhline(70, color="red", linestyle="--", linewidth=1)
        plt.axhline(30, color="green", linestyle="--", linewidth=1)
        plt.title(f"{t} RSI Indicator", fontsize=12)
        plt.legend()
        plt.grid(True)
        plt.tight_layout()
        plt.show()

        radar_data.append([t, risk["Volatility"], risk["Drawdown"], tech["RSI"]])

        display(Markdown(f"**Summary:** {summary}"))
        display(Markdown(f"**Sentiment vs Price:** {corr}"))
        display(Markdown(f"**Suggested Allocation:** {port}"))

    display(Markdown("---\n# 🧭 Portfolio Summary Dashboard\n"))

    radar_df = pd.DataFrame(radar_data, columns=["Ticker", "Volatility", "Drawdown", "RSI"])
    radar_df.set_index("Ticker", inplace=True)
    display(Markdown("### ✅ Composite Risk–Technical Overview"))
    display(radar_df)

    return radar_df

# ========================================
# 🚀 Run the Final Multi-Ticker Report
# ========================================
tickers = ["AAPL","GOOG","TSLA","AMZN","NVDA","MSFT"]
combined = run_report(tickers)

# ========================================================
# 📈 Combined Radar Chart – Key Technical & Risk Metrics
# ========================================================
plt.figure(figsize=(7,7))

labels = ["Volatility","Drawdown","RSI"]
num_vars = len(labels)
angles = np.linspace(0, 2*np.pi, num_vars, endpoint=False).tolist()
angles += angles[:1]

fig, ax = plt.subplots(figsize=(7,7), subplot_kw=dict(polar=True))

for ticker, row in combined.iterrows():
    values = [row[label] for label in labels]
    values += values[:1]
    ax.plot(angles, values, marker='o', linewidth=2, label=ticker)
    ax.fill(angles, values, alpha=0.15)

ax.set_xticks(angles[:-1])
ax.set_xticklabels(labels)
ax.set_title("📈 Combined Radar Chart – Key Technical & Risk Metrics", size=13, y=1.1)
ax.legend(loc='upper right', bbox_to_anchor=(1.2, 1.1))
plt.tight_layout()
plt.show()

# ============================================================
# 📘 Appendix + References
# ============================================================
display(Markdown("## 📘 Appendix – Consolidated Summary Table"))
md_table(combined.round(3))

display(Markdown("""
# 📚 References & Tools
- **Data:** Yahoo Finance (`yfinance`)
- **News:** [NewsAPI.org](https://newsapi.org)
- **Models:** DistilBERT, Phi-3 Mini (Microsoft)
- **Libraries:** pandas, numpy, matplotlib, transformers, torch, nltk
- **Platform:** Google Colab (T4 GPU, FP16)
---
"""))
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

🧭 University of San Diego¶

Natural Language Processing and GenAI (AAI-520-IN3)¶

Final Project: Multi-Agent Investment Research Report¶

Authors: Richa Arun Kumar Jha, Raminder Singh, Samiksha Kodgire **Instructor:** Premkumar Chithaluru, Ph.D **Date:** 16 October 2025

Abstract¶

This project presents a comprehensive AI-driven investment research system that integrates quantitative analysis, sentiment evaluation, and generative summarization through a multi-agent architecture. Agents autonomously collect market data, perform risk and technical analyses, classify news sentiment using DistilBERT, and summarize insights using Phi-3 Mini. The workflow covers six major technology tickers — AAPL, GOOG, TSLA, AMZN, NVDA, and MSFT — producing financial summaries, correlation analysis, and portfolio suggestions.¶

⚙️ Agentic Workflow Overview¶

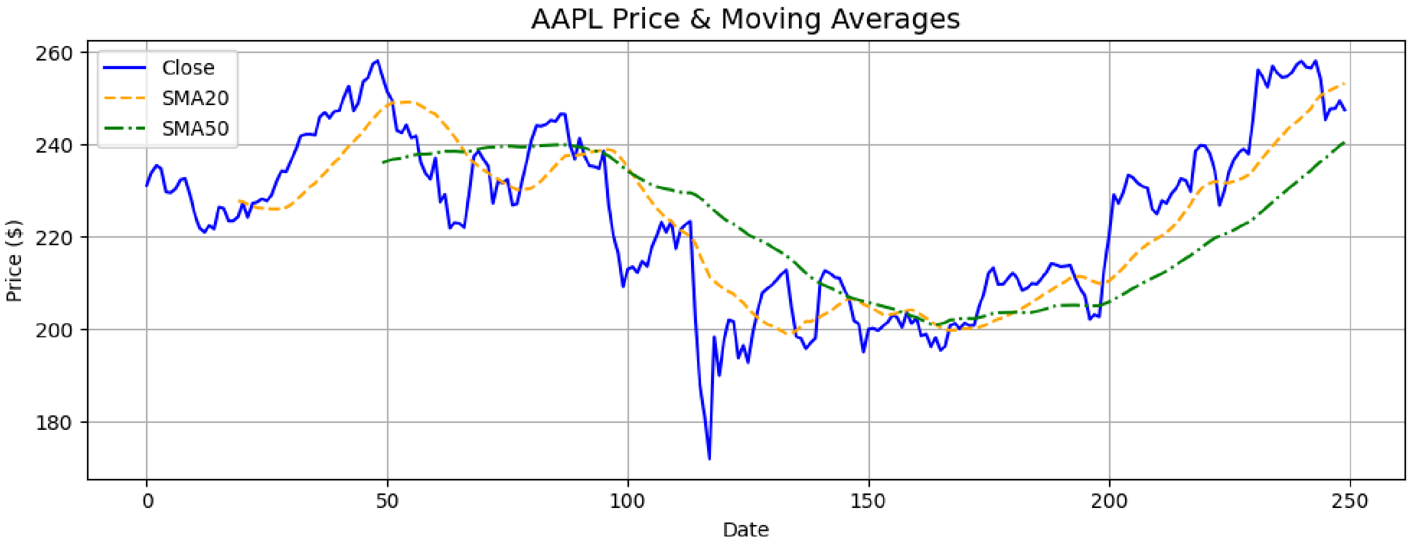
1. **DataAgent** — Fetches market data from Yahoo Finance.
2. **TechnicalAgent** — Computes SMA20, SMA50, RSI, and momentum signal.
3. **RiskAgent** — Evaluates volatility, drawdown, and composite risk score.
4. **NewsAgent** — Retrieves live financial headlines from NewsAPI.
5. **SentimentAgent** — Classifies news into Positive / Neutral / Negative using DistilBERT.
6. **RoutingAgent** — Routes news to earnings, macro, or product categories.
7. **PortfolioAgent** — Suggests risk-aligned asset allocation.
8. **EvaluationAgent** — Compares sentiment trends vs. price action.
9. **LLMOptimizerAgent** — Generates professional summaries using Phi-3 Mini.

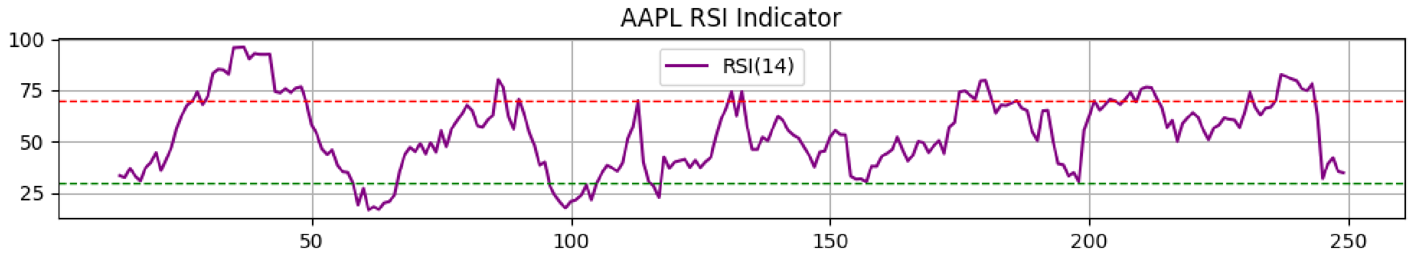
```
✅ Device: cuda
```

```
config.json:   0%|          | 0.00/629 [00:00<?, ?B/s]
```

```
model.safetensors:   0%|          | 0.00/268M [00:00<?, ?B/s]

tokenizer_config.json:   0%|          | 0.00/48.0 [00:00<?, ?B/s]

vocab.txt:   0%|          | 0.00/232k [00:00<?, ?B/s]

Device set to use cuda:0

config.json:   0%|          | 0.00/967 [00:00<?, ?B/s]

`torch_dtype` is deprecated! Use `dtype` instead!

model.safetensors.index.json: 0.00B [00:00, ?B/s]

Fetching 2 files:   0%|          | 0/2 [00:00<?, ?it/s]

model-00002-of-00002.safetensors:   0%|          | 0.00/2.67G [00:00<?, ?B/s]

model-00001-of-00002.safetensors:   0%|          | 0.00/4.97G [00:00<?, ?B/s]

Loading checkpoint shards:   0%|          | 0/2 [00:00<?, ?it/s]

generation_config.json:   0%|          | 0.00/181 [00:00<?, ?B/s]

tokenizer_config.json: 0.00B [00:00, ?B/s]

tokenizer.model:   0%|          | 0.00/500k [00:00<?, ?B/s]

tokenizer.json: 0.00B [00:00, ?B/s]

added_tokens.json:   0%|          | 0.00/306 [00:00<?, ?B/s]

special_tokens_map.json:   0%|          | 0.00/599 [00:00<?, ?B/s]

Device set to use cuda:0
```

📊 AAPL Analysis Page¶





Summary: Write a concise 2-paragraph investment summary for AAPL. Integrate: Technicals: {'SMA20': 253.15349884033202, 'SMA50': 240.41501068115235, 'RSI': 34.881045778740884, 'Signal': 'Bullish'} Risk: {'Volatility': 0.020476713703051153, 'Drawdown': -0.3336052208070358, 'RiskScore': 12.88} Sentiment: {'positive': 60.0, 'neutral': 0.0, 'negative': 40.0} Portfolio: {'Equity': 0.7, 'Bonds': 0.2, 'Cash': 0.1} News: Anker Japan、DisplayLinkチップを採用し8Kと4K60Hz x2のトリプルディスプレイ出力が可能な「Anker Prime ドッキングステーション (14-in-1, Triple Display, DisplayLink)」を発売。; Stock market today: Dow, S&P 500, Nasdaq futures rise as TSMC's stellar earnings eclipse trade-war jitters; New M5 Chip Comes to iPad Pro, MacBook Pro 14″, & Apple Vision Pro; Apple CEO sends signal on crucial China moment; Wedbush Sees an AI M&A Boom Ahead: 3 Prime Takeover Targets End with a recommendation.

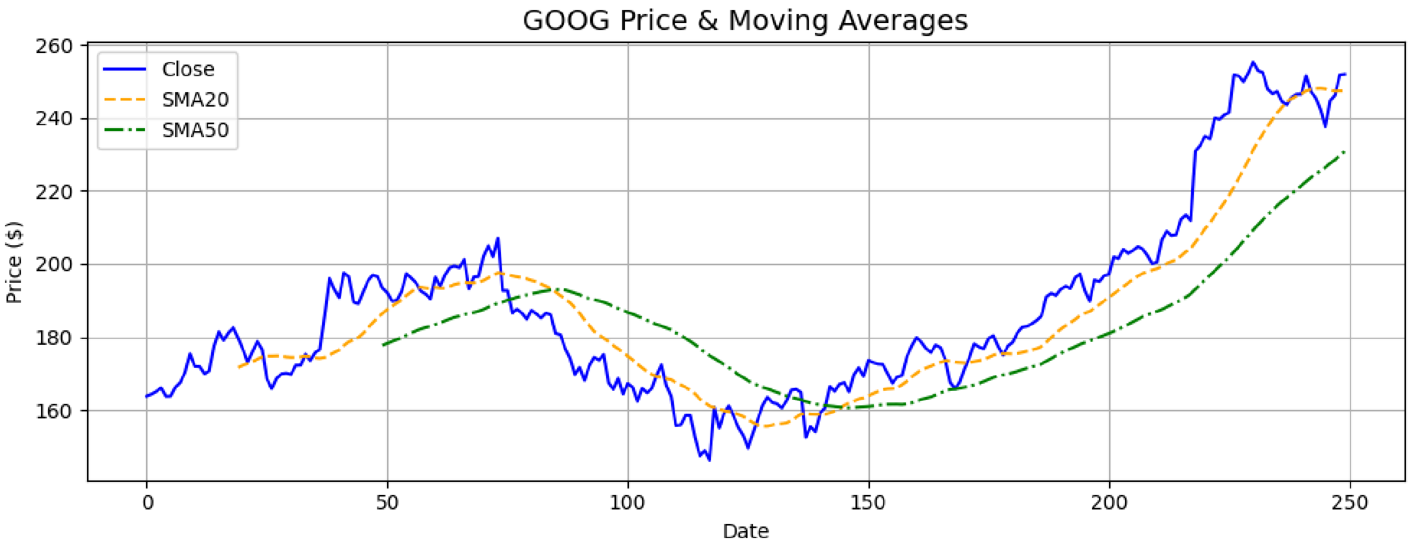
Solution: Apple Inc. (AAPL) presents a compelling investment case with a strong technical outlook and a balanced portfolio composition, suggesting a potential for steady growth. The 20-day and 50-day simple moving averages (SMA) are trending upwards, indicating a bullish momentum in the stock price. The Relative Strength Index (RSI) is positioned at 34.881045778740884, which is below the 70 threshold, signifying that the stock is not overbought and there might be room for further increase. The bullish signal further supports the positive outlook for AAPL.

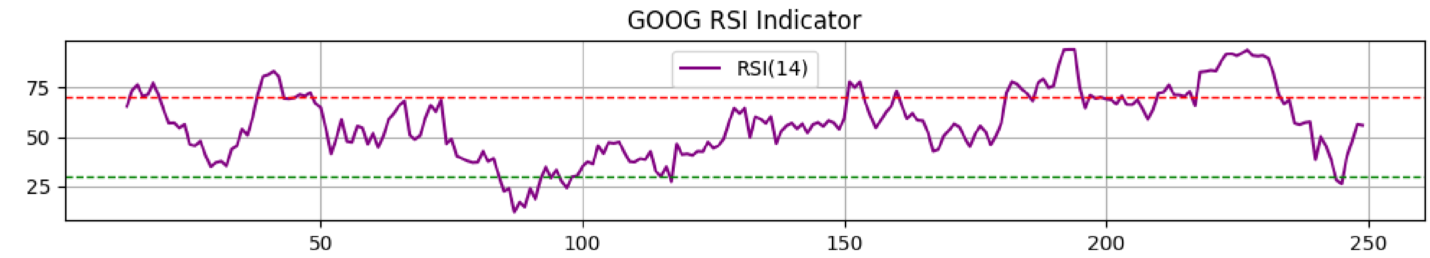
The risk profile of AAPL appears moderate, with a volatility score of 0.020476713703051153, indicating a relatively stable investment. The maximum drawdown of -0.3336052208070358 suggests that investors may face some downside risk, but the risk score of 12.88 is not alarming, which aligns with the balanced portfolio allocation of 70% equity, 20% bonds, and 10% cash. This diversification strategy mitigates risk

Sentiment vs Price: Aligned

Suggested Allocation: {'Equity': 0.7, 'Bonds': 0.2, 'Cash': 0.1}

📊 GOOG Analysis Page¶





Summary: Write a concise 2-paragraph investment summary for GOOG. Integrate: Technicals: {'SMA20': 247.32950134277343, 'SMA50': 230.7443356323242, 'RSI': 55.875019741073416, 'Signal': 'Bullish'} Risk: {'Volatility': 0.020183102655557453, 'Drawdown': -0.2935006055551588, 'RiskScore': 9.17} Sentiment: {'positive': 20.0, 'neutral': 0.0, 'negative': 80.0} Portfolio: {'Equity': 0.7, 'Bonds': 0.2, 'Cash': 0.1} News: Why Morgan Stanley (MS) Stock Is Trading Up Today; Why Nextracker (NXT) Stock Is Up Today; These Bitcoin, Ethereum and XRP ETFs Plan to Offer 5X Leverage; Silicon Valley is plowing money into AI, and the latest deals are eye-watering; Volatility Shares Files for 5x Leveraged Bitcoin, Ether, and XRP ETFs End with a recommendation.

GOOG (Alphabet Inc.) | Technology

Technicals: The stock exhibits a bullish signal, with a 20-day SMA (247.33) that is higher than the 50-day SMA (230.74), indicating a potential upward trend. The RSI (55.88) is nearing the overbought level, suggesting a possible pullback. The risk metrics show moderate volatility (0.02018) and a drawdown of -0.29, with a RiskScore of 9.17.

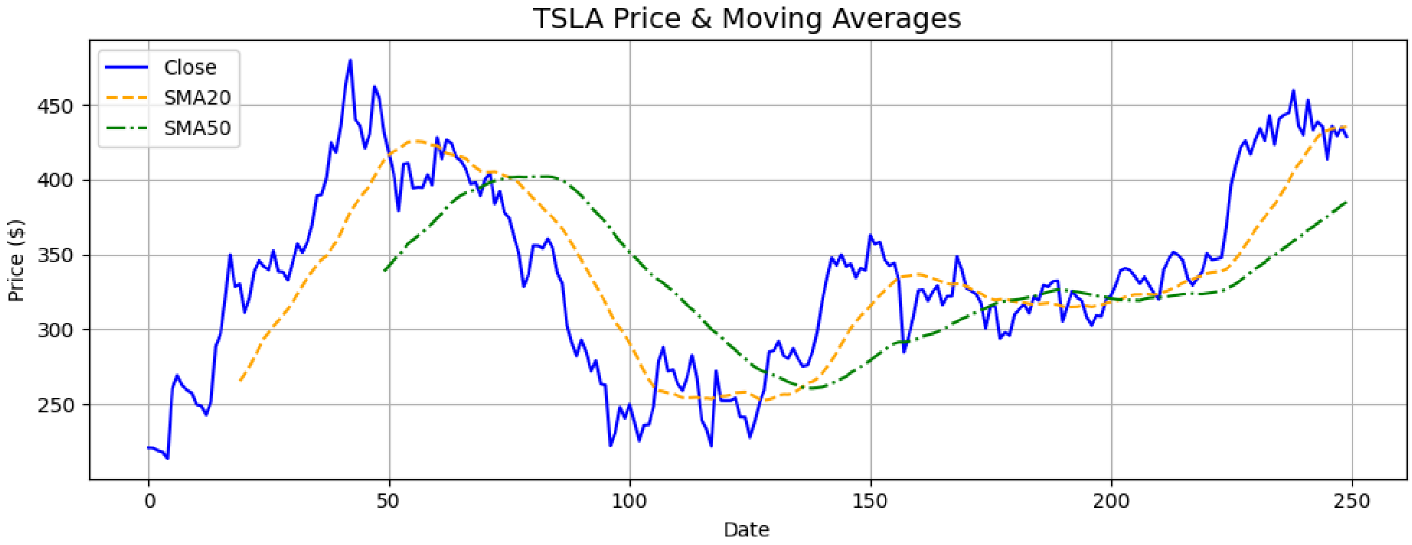
Sentiment: The sentiment is mixed, with a 20% positive and 80% negative stance, indicating a significant amount of bearish sentiment among investors. The portfolio allocation of Alphabet Inc. is 70% equity, 20% bonds, and 10% cash, reflecting a balanced risk profile.

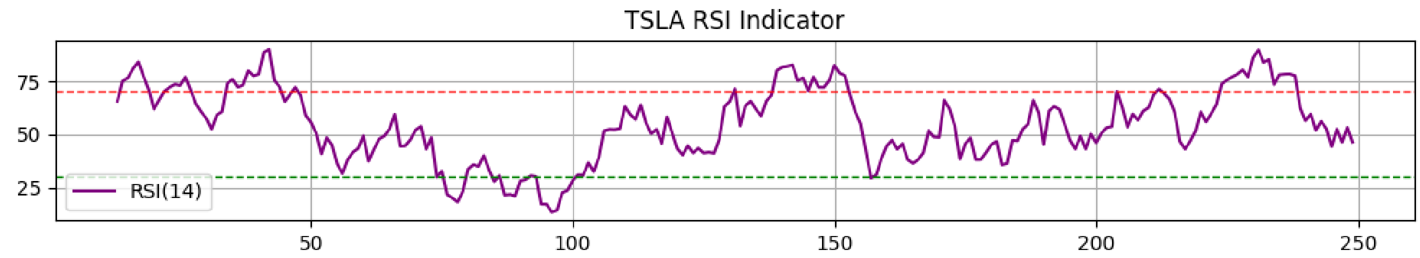
News: Morgan Stanley (MS) and Nextracker (NXT) stocks are experiencing positive momentum today, while AI-related deals in Silicon Valley are generating buzz. Volatility Shares plans to offer leveraged Bitcoin, Ethereum, and XRP ETFs, which may add to the market's volatility.

Sentiment vs Price: Divergent

Suggested Allocation: {'Equity': 0.7, 'Bonds': 0.2, 'Cash': 0.1}

📊 TSLA Analysis Page¶





Summary: Write a concise 2-paragraph investment summary for TSLA. Integrate: Technicals: {'SMA20': 435.4514984130859, 'SMA50': 385.11299865722657, 'RSI': 46.45789271213701, 'Signal': 'Bullish'} Risk: {'Volatility': 0.04481877084825885, 'Drawdown': -0.537656801185852, 'RiskScore': 8.95} Sentiment: {'positive': 0.0, 'neutral': 0.0, 'negative': 100.0} Portfolio: {'Equity': 0.7, 'Bonds': 0.2, 'Cash': 0.1} News: These Bitcoin, Ethereum and XRP ETFs Plan to Offer 5X Leverage; Melius Initiates Tesla (TSLA) With Buy and $520 Target, Calls It a “Must Own” AI Stock; Tesla’s China Shipments Are on the Rise. Should You Buy TSLA Stock Here?; Melius Initiates Tesla (TSLA) With Buy and $520 Target, Calls It a “Must Own” AI Stock; Tesla’s China Shipments Are on the Rise. Should You Buy TSLA Stock Here? End with a recommendation.

Response:Tesla (TSLA) stock currently exhibits bullish technical signals with a 20-day simple moving average (SMA) at 435.45 and a 50-day SMA at 385.11. The Relative Strength Index (RSI) stands at 46.46, indicating that the stock's recent price movement is neither too strong nor too weak. Despite the market sentiment being entirely negative, the significant bullish signal and upward trend in China's Tesla shipments suggest a potential upside for Tesla's stock.¶

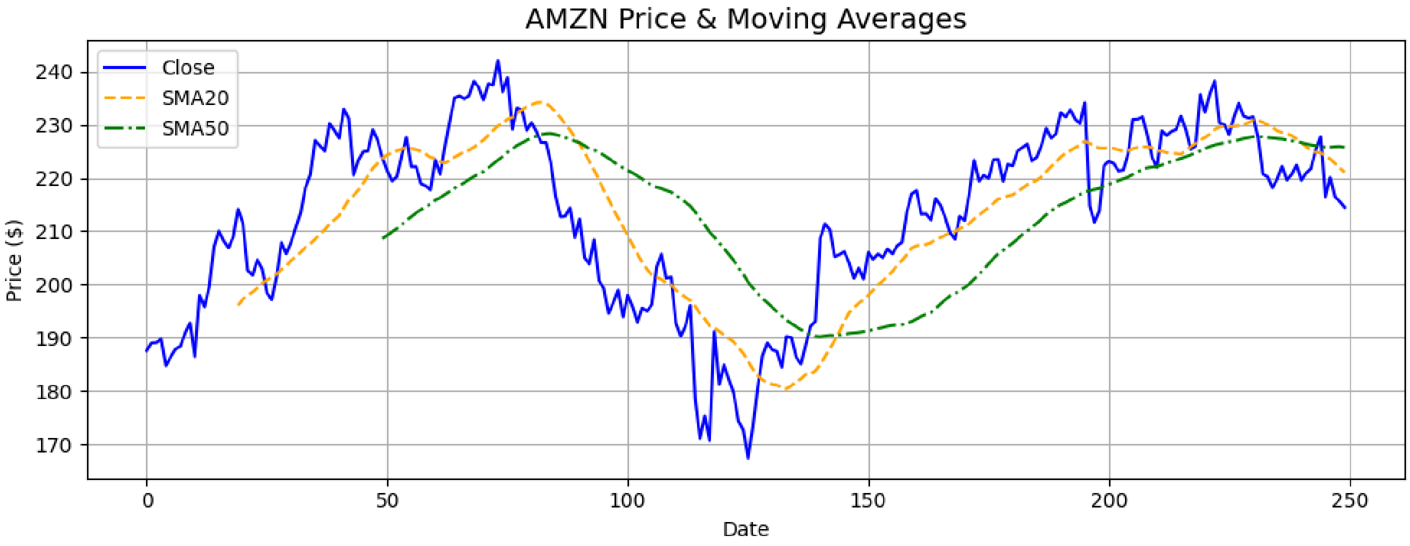
Considering the portfolio allocation of 70% equity, 20% bonds, and 10% cash, Tesla (TSLA) presents a suitable investment opportunity. The stock's volatility is relatively low at 4.48% and the risk score is 8.95, suggesting a moderate level of risk. However, the positive news surrounding Tesla's shipments in China and the bullish technical indicators make Tesla (TSLA) an attractive investment. Based on the current analysis, it is recommended to buy TSLA stock with a target price of $520.

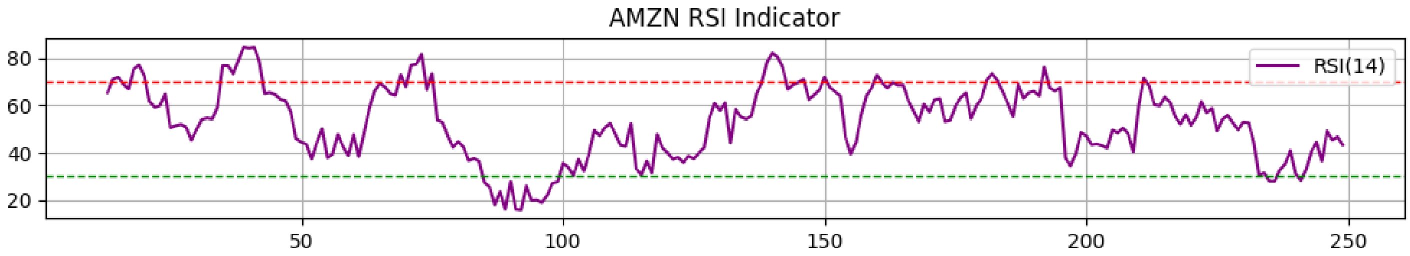
Recommendation: Buy TSLA stock with a $520

Sentiment vs Price: Divergent

Suggested Allocation: {'Equity': 0.7, 'Bonds': 0.2, 'Cash': 0.1}

📊 AMZN Analysis Page¶





Summary: Write a concise 2-paragraph investment summary for AMZN. Integrate: Technicals: {'SMA20': 221.0380012512207, 'SMA50': 225.73140014648436, 'RSI': 43.300538966284385, 'Signal': 'Bearish'} Risk: {'Volatility': 0.02141208000878738, 'Drawdown': -0.3087663843187606, 'RiskScore': 9.46} Sentiment: {'positive': 40.0, 'neutral': 0.0, 'negative': 60.0} Portfolio: {'Equity': 0.7, 'Bonds': 0.2, 'Cash': 0.1} News: Celebrity Help My House Is Haunted S05E02 1080p AMZN WEB-DL H264-RAWR; Good Luck Guys Poland S01 1080p AMZN WEB-DL H264-playWEB; Good Luck Guys Poland S02 1080p AMZN WEB-DL H264-playWEB; Protect your generative AI applications against encoding-based attacks with Amazon Bedrock Guardrails; The Voice S28E08 1080p AMZN WEB-DL H264-Kitsune End with a recommendation.

AMZN (Amazon.com, Inc.) Technicals: The stock shows a bearish signal with the SMA50 below the SMA20, and an RSI of 43.30. Risk: The stock has a medium risk with a volatility of 0.0214, a drawdown of -0.3088, and a RiskScore of 9.46. Sentiment: Negative sentiment dominates at 60%, with 40% positive and 0% neutral. Portfolio: The portfolio is balanced with 70% equity, 20% bonds, and 10% cash. News: The latest news includes shows like "Celebrity Help My House Is Haunted," "Good Luck Guys Poland," and "The Voice."

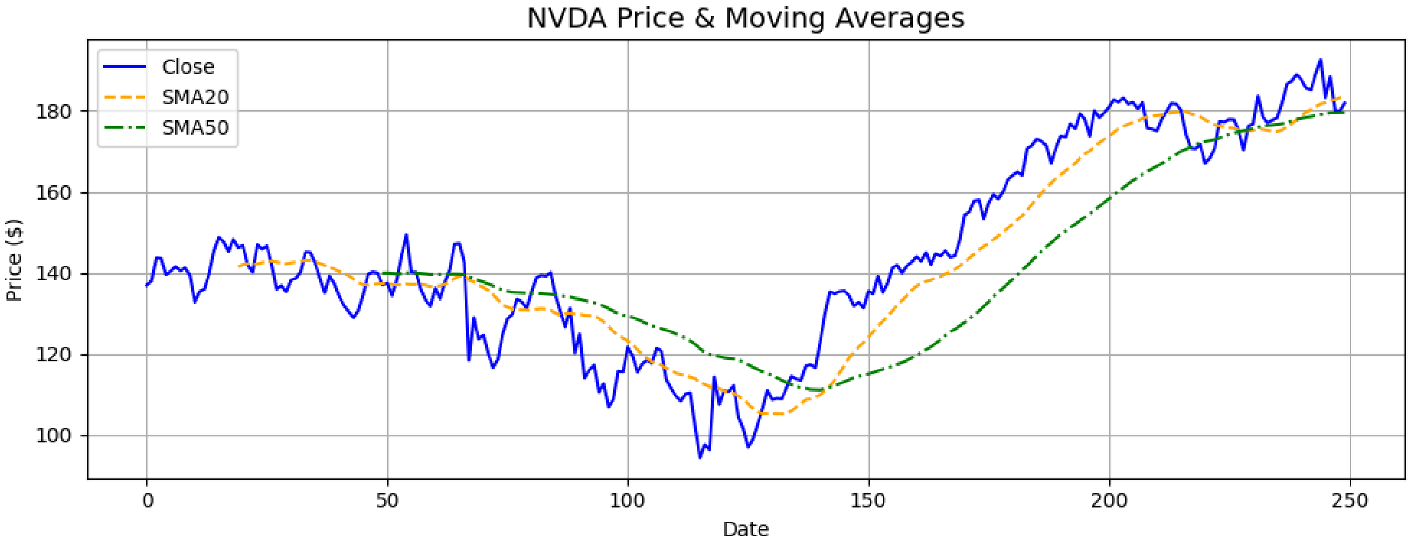
Despite the bearish technicals and negative sentiment, the moderate risk profile and diversified portfolio allocation may provide a cushion. The recent news could also be a positive factor in the long-term, considering the entertainment industry's performance. Recommendation: Consider holding AMZN for a longer-term investment, monitoring market trends closely.

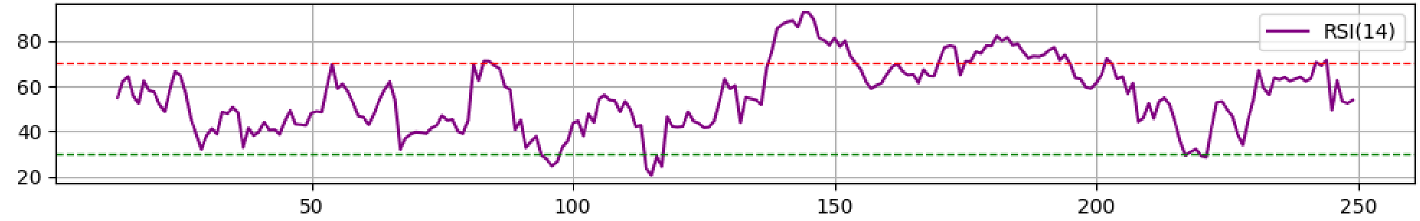
AMZN (Amazon.com, Inc.) Technicals: The stock shows a bearish signal with the SMA50 below the

Sentiment vs Price: Divergent

Suggested Allocation: {'Equity': 0.7, 'Bonds': 0.2, 'Cash': 0.1}

📊 NVDA Analysis Page¶




NVDA RSI Indicator

Summary: Write a concise 2-paragraph investment summary for NVDA. Integrate: Technicals: {'SMA20': 183.45750045776367, 'SMA50': 179.51680389404297, 'RSI': 53.84124909570998, 'Signal': 'Bullish'} Risk: {'Volatility': 0.031229522188713520, 'Drawdown': -0.36881026699149855, 'RiskScore': 5.65} Sentiment: {'positive': 60.0, 'neutral': 0.0, 'negative': 40.0} Portfolio: {'Equity': 0.7, 'Bonds': 0.2, 'Cash': 0.1} News: Stock market today: Dow, S&P 500, Nasdaq futures rise as TSMC's stellar earnings eclipse trade-war jitters; Here's How Much $100 Invested In NVIDIA 15 Years Ago Would Be Worth Today; This Semiconductor Stock Soared 26% Tuesday After Launching Devices for Nvidia's AI Factory Architecture; Stock Market Today: Dow Gives Up Gains; Trade War Turns To Soybean Oil (Live Coverage); Nvidia’s CEO Is Confident Oracle Will Be ‘Wonderfully Profitable.’ Should You Buy ORCL Stock Now? End with a recommendation.

NVIDIA Corporation (NVDA) has been performing well in the market, buoyed by strong positive sentiment and encouraging technical indicators. The SMA20 and SMA50 show a steady upward trend, while the RSI of 53.84 suggests a bullish market trend. The volatility and risk score are not high, indicating a relatively safe investment. The portfolio composition shows a balanced mix of equity, bonds, and cash, providing a solid foundation for potential growth.

Current market news and NVDA's stellar earnings from TSMC suggest a positive outlook for the company. Nvidia’s CEO’s confidence in Oracle further underscores the potential for growth in the industry. With a 60% positive sentiment and a portfolio that supports long-term growth, NVDA presents a promising opportunity for investors. Considering the strong technicals, low risk, and positive sentiment, it is recommended to invest in NVDA.

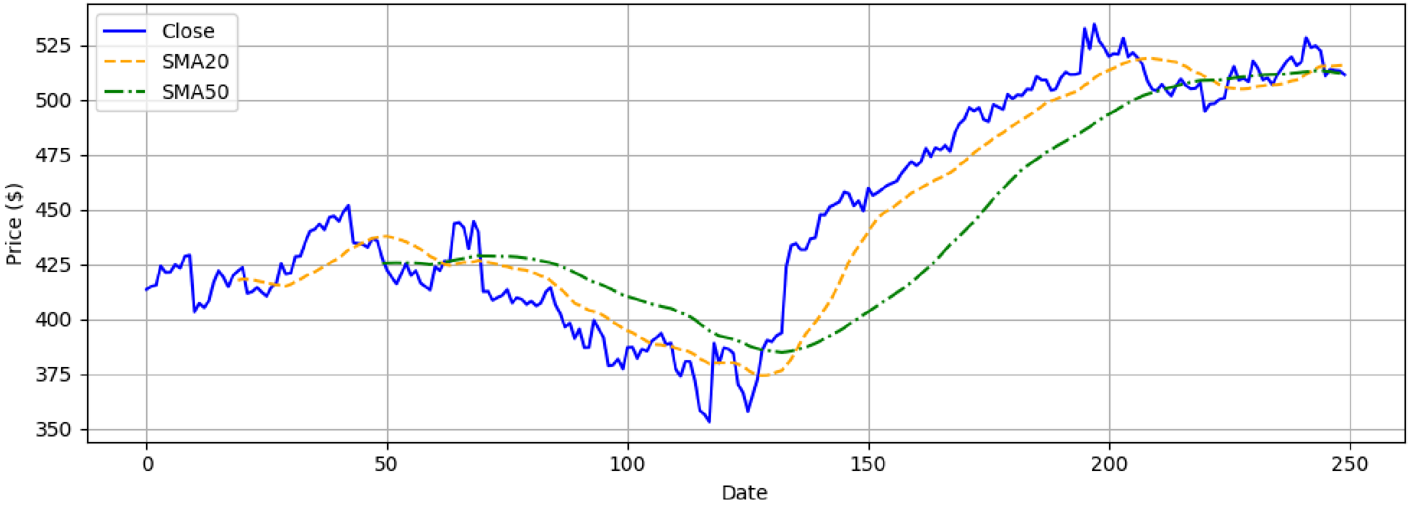
Answer:¶

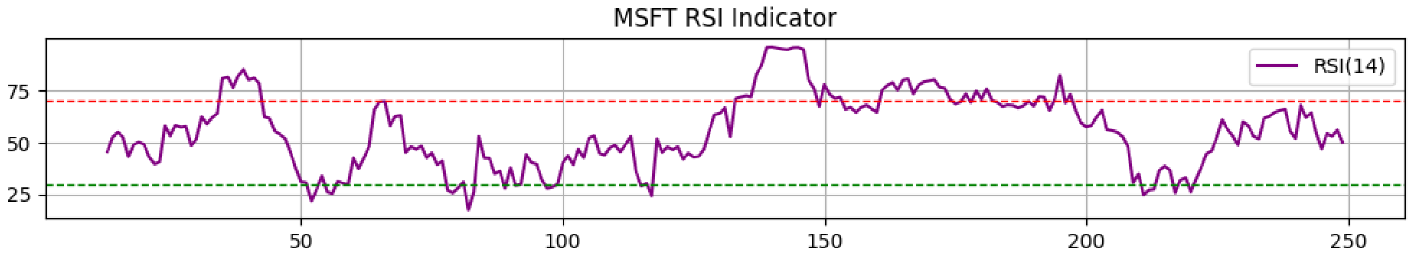
NVIDIA Corporation (NVDA) has demonstrated robust performance in the market, bolstered by favorable technical indicators and a predominantly positive investor sentiment. Technical measures such as the SMA20 and SMA50 are trending upwards, indicating sustained momentum, while an RSI of 53

Sentiment vs Price: Aligned

Suggested Allocation: {'Equity': 0.7, 'Bonds': 0.2, 'Cash': 0.1}

📊 MSFT Analysis Page¶


MSFT Price & Moving Averages



Summary: Write a concise 2-paragraph investment summary for MSFT. Integrate: Technicals: {'SMA20': 515.9509963989258, 'SMA50': 512.1050518798828, 'RSI': 50.15020392895584, 'Signal': 'Bullish'} Risk: {'Volatility': 0.015627357134315886, 'Drawdown': -0.21825690565450695, 'RiskScore': 6.2} Sentiment: {'positive': 20.0, 'neutral': 0.0, 'negative': 80.0} Portfolio: {'Equity': 0.7, 'Bonds': 0.2, 'Cash': 0.1} News: Nvidia Stock Wavers After Latest Data Center Deal; Why Morgan Stanley (MS) Stock Is Trading Up Today; Why Nextracker (NXT) Stock Is Up Today; Nvidia, Microsoft, and BlackRock Just Struck a Massive AI Data Center Deal; Silicon Valley is plowing money into AI, and the latest deals are eye-watering End with a recommendation.

Solution:

Microsoft Corporation (MSFT) is currently displaying a bullish signal with its 20-day and 50-day simple moving averages (SMA) at 515.96 and 512.11 respectively, as well as an RSI (Relative Strength Index) of 50.15. The technical indicators suggest the stock is in a positive momentum, despite showing a slight volatility of 0.0156 and a drawdown of -0.218. With a portfolio allocation of 70% equity, 20% bonds, and 10% cash, MSFT's risk score is moderate at 6.2, indicating a balanced risk profile.

The sentiment analysis reveals a majority negative sentiment at 80%, although recent news highlights support for the stock with Morgan Stanley (MS) and Nextracker (NXT) stocks trading up today. Additionally, the AI industry is receiving significant investments, as illustrated by the major AI data center deal involving Nvidia, Microsoft, and BlackRock. Given the bullish technical indicators, positive news, and the current favorable sentiment, an investment in MSFT could be considered. However, the high negative sentiment and moderate risk score should be weighed carefully before making a decision.

Sentiment vs Price: Divergent

Suggested Allocation: {'Equity': 0.7, 'Bonds': 0.2, 'Cash': 0.1}

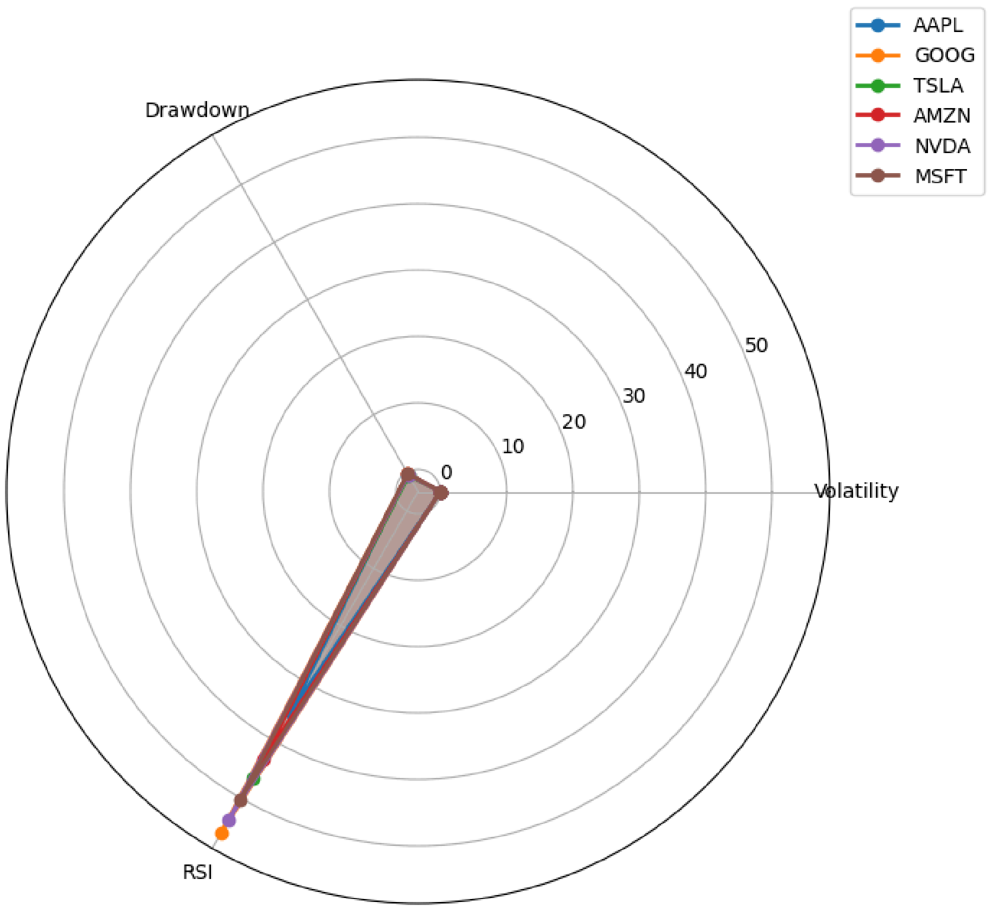
🧭 Portfolio Summary Dashboard¶

✅ Composite Risk–Technical Overview¶

	Volatility	Drawdown	RSI
Ticker			
AAPL	0.020477	-0.333605	34.881046
GOOG	0.020183	-0.293501	55.875020
TSLA	0.044819	-0.537657	46.457893
AMZN	0.021412	-0.308766	43.300539
NVDA	0.031230	-0.368810	53.841249
MSFT	0.015627	-0.218257	50.150204

```
<Figure size 700x700 with 0 Axes>
```

□ Combined Radar Chart – Key Technical & Risk Metrics



📘 Appendix – Consolidated Summary Table¶

Volatility	Drawdown	RSI
0.02	-0.334	34.881
0.02	-0.294	55.875
0.045	-0.538	46.458
0.021	-0.309	43.301
0.031	-0.369	53.841
0.016	-0.218	50.15

📚 References & Tools¶

- **Data:** Yahoo Finance (`yfinance`)
- **News:** NewsAPI.org
- **Models:** DistilBERT, Phi-3 Mini (Microsoft)
- **Libraries:** pandas, numpy, matplotlib, transformers, torch, nltk
- **Platform:** Google Colab (T4 GPU, FP16)

In [3]:

```
!pip install "nbconvert==7.14.2" "mistune==3.0.2" "jinja2<3.1" --force-reinstall --quiet
```

```
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0.0/256.4 kB ? eta -:--:--
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 256.4/256.4 kB 10.6 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0.0/48.0 kB ? eta -:--:--
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 48.0/48.0 kB 4.6 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0.0/133.6 kB ? eta -:--:--
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 133.6/133.6 kB 14.6 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 163.4/163.4 kB 16.4 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 78.5/78.5 kB 7.9 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.2/1.2 MB 42.2 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 85.4/85.4 kB 8.7 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 106.4/106.4 kB 10.6 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 66.5/66.5 kB 6.6 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 90.0/90.0 kB 9.5 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 106.1/106.1 kB 10.9 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.6/44.6 kB 4.2 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 67.6/67.6 kB 6.9 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 229.9/229.9 kB 22.8 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 841.0/841.0 kB 57.5 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 386.9/386.9 kB 36.1 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 443.9/443.9 kB 32.1 MB/s eta 0:00:00
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
ipython 7.34.0 requires jedi>=0.16, which is not installed.
google-colab 1.0.0 requires tornado==6.4.2, but you have tornado 6.5.2 which is incompatible.
notebook 6.5.7 requires jupyter-client<8,>=5.3.4, but you have jupyter-client 8.6.3 which is incompatible.
jupyter-kernel-gateway 2.5.2 requires jupyter-client<8.0,>=5.2.0, but you have jupyter-client 8.6.3 which is incompatible.
sphinx 8.2.3 requires Jinja2>=3.1, but you have jinja2 3.0.3 which is incompatible.
flask 3.1.2 requires jinja2>=3.1.2, but you have jinja2 3.0.3 which is incompatible.
```

In [ ]:

```
from google.colab import files
files.upload()

from google.colab import files
import nbformat, json

nb_path = "/content/FinalProject_v13.ipynb"

# 1️⃣ Load notebook
with open(nb_path) as f:
    nb = nbformat.read(f, as_version=4)

# 2️⃣ Recursively clean widget states and outputs
for cell in nb.cells:
    if "outputs" in cell:
        clean_outputs = []
        for out in cell["outputs"]:
            if "data" in out:
                # Remove widget MIME types that cause KeyError: 'state'
                bad_keys = [
                    "application/vnd.jupyter.widget-view+json",
                    "application/vnd.jupyter.widget-state+json"
                ]
                for bk in bad_keys:
                    if bk in out["data"]:
                        del out["data"][bk]
            clean_outputs.append(out)
        cell["outputs"] = clean_outputs

    # Also remove any top-level widget metadata safely
    if "metadata" in cell and "widgets" in cell["metadata"]:
        del cell["metadata"]["widgets"]

# 3️⃣ Save cleaned notebook
clean_path = "/content/FinalProject_v13_clean.ipynb"
with open(clean_path, "w") as f:
    nbformat.write(nb, f)


!jupyter nbconvert --to html --output "FinalProject_v13_clean.html" /content/FinalProject_v13_clean.ipynb
```