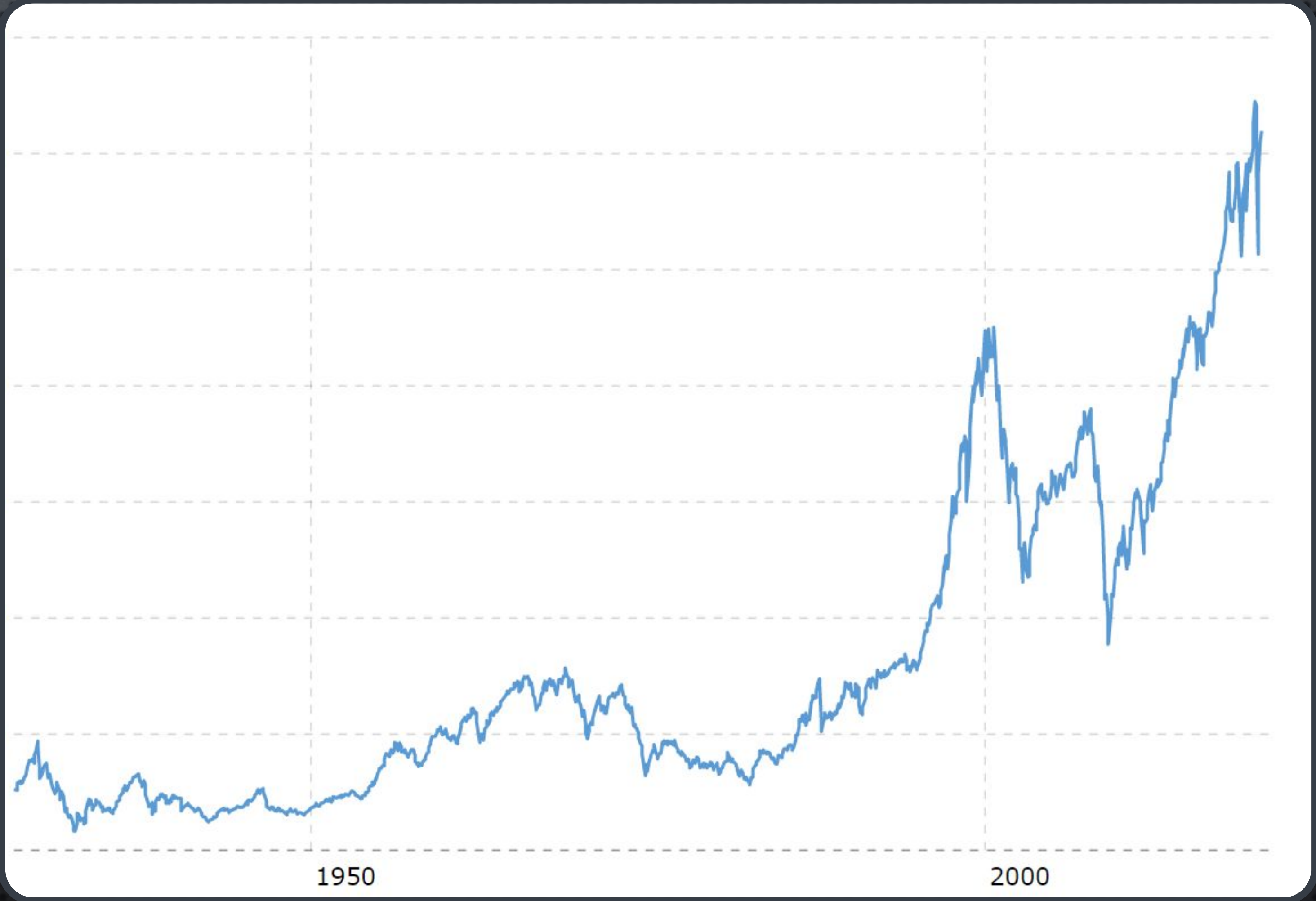
DATA-DRIVEN METHODS FOR FINANCE

MEDALLION

Ilan Gabsi, Nathan Tuil, Noame De Boerdere

STRATEGY 1 - S&P 500 FOCUS

- Strategy Overview: Emphasis on investing heavily in the S&P 500.
- Rationale: Stability and historically consistent returns of the S&P 500.
- Risk Profile: Low-risk approach aiming for steady growth.
- Allocation: Breakdown of investment percentages in the S&P 500 and other minor assets.



STRATEGY 2 - MILITARY, OIL, AND CRYPTOCURRENCY INVESTMENTS

- Strategy Overview: Investments based on geopolitical context lately.
- Sector Focus: Long positions in military security, army, defense, and the oil industry.
- Cryptocurrency Inclusion: Weights on BTC and ETH due to BlackRock's crypto announcement.
- Short Positions: Strategy to short the remaining assets.
- Analysis: Rationale behind each sector choice and expected market impact.

PERFORMANCE ANALYSIS

	group_name	mean_forecast	mean_decision	forecasts_rank	decisions_rank	overall_rank
0	HIREUS	0.157279	5.012409	3.0	1.0	2.0
1	MEDALLION	0.167839	3.135784	6.0	3.0	4.5
2	SP500	0.121401	-1.565712	1.0	9.0	5.0
3	RANDOM	0.170230	0.769636	7.0	5.0	6.0
4	4SIGMA	0.247037	4.356688	13.0	2.0	7.5
5	EW_LONG	0.160000	-2.650069	4.0	11.0	7.5
6	LAMM	0.180820	0.040650	9.0	7.0	8.0
7	GAMBLING	0.147344	-5.475511	2.0	15.0	8.5
8	HELLO_WORLD	0.205163	0.644338	11.0	6.0	8.5
9	UHCAKIP	0.171820	-2.285424	8.0	10.0	9.0
10	CITADELSPINOFFS	0.286411	2.452745	15.0	4.0	9.5
11	JAYSTREET	0.160209	-4.983883	5.0	14.0	9.5
12	DYF	0.253402	-1.434781	14.0	8.0	11.0
13	ALPHA	0.182299	-4.034960	10.0	13.0	11.5
14	MSG	0.219520	-3.072616	12.0	12.0	12.0

STRATEGY 3 - OIL, SELECT ETFS, AND CRYPTOCURRENCIES FOCUS

- Strategy Overview: Heavy investment in OIL (Com, XOP) and ETFs (XLE, XLI).
- Cryptocurrency Inclusion: Substantial allocation in ETH and BNB.
- Zero Weight: No investment in other assets.
- Strategy Rationale: Insights into choosing these particular sectors and assets.

STRATEGY 4 – SENTIMENT ANALYSIS - TWITTER DATA COLLECTION: CRYPTOCURRENCY MARKET INSIGHTS

- Gathering tweets from key cryptocurrency influencers to get market insights.
- Explanation of the data collection process: Using Twitter API to fetch recent tweets from influencers like CoinMarketCap, Cointelegraph, etc.
- Overview of the extended list of crypto keywords used to filter relevant tweets.
- technical environment: Python, requests library for API calls, and pandas for data manipulation.

```python
# Twitter accounts you want to fetch tweets from
influencers = [
    "CoinMarketCap",
    "Cointelegraph",
    "coinmetrics",
    "coinbase",
    "DefiantNews",
    "WeeklyCrypto",
    "decryptmedia",
    "MessariCrypto",
    "StackerSatoshi",
]

# Extended list of crypto keywords
crypto_keywords = ["crypto", "bitcoin", "ethereum", "eth", "blockchain", "cryptocurrenc
                   "ripple", "litecoin", "cardano", "polkadot", "dogecoin", "uniswap",
                   "solana", "binance", "cryptomining", "nft", "non-fungible tokens",
                   "defi", "staking",
                   "wallet", "exchange", "ICO", "token", "coin",
                   "satoshi", "hodl", "hold", "altcoin", "alt", "gas",
                   "mining", "work", "stake",
                   "fork", "airdrop", "whale", "bear", "bull", "pump", "dump",
                   "moon", "fiat", "liquidity", "ERC20",
                   "stablecoin", "decentralized", ]
# Initializes an empty list to store the tweet data
data = []

# Headers for the API request
headers = {
    "Authorization": f"Bearer {BEARER_TOKEN}",
    "Content-type": "application/json",
}

# Loop through each influencer
for influencer in influencers:
    # Create a query string that includes the influencer's handle and the crypto keywor
    query_string = f"from:{influencer} ({' OR '.join([f'{keyword}' for keyword in crypt

    # Fetch recent tweets from the influencer (last 100 tweets as per Twitter API v2 li
    params = {
        'query': query_string,
        'max_results': 100,
        'tweet.fields': 'text,created_at',  # Include 'created_at' to get the date of t
    }

    response = requests.get(endpoint, headers=headers, params=params)

    if response.status_code != 200:
        raise Exception(f"Request returned an error: {response.status_code}, {response.

    tweets = response.json().get('data', [])

    # Process the response
    for tweet in tweets:
        text = tweet.get('text', '')
        created_at = tweet.get('created_at', '')

        # Store the result in a dictionary
        tweet_data = {
            "Influencer": influencer,
            "Tweet": text,
            "Date": created_at,
        }

        # Append the dictionary to the data list
        data.append(tweet_data)

# Create a pandas DataFrame from the data list
df1 = pd.DataFrame(data)

# Save the DataFrame as a CSV file
df1.to_csv("twitter_post_analysis_newletter.csv", index=False)
```

ENHANCING DATA QUALITY: CLEANING AND TEXT ANALYSIS

- Detailed explanation of the data cleaning steps: Removing URLs, special characters, digits, and extra whitespaces; converting text to lowercase.
- Overview of text processing techniques: Stopword removal, tokenization, and lemmatization using NLTK.
- Purpose of these steps: Improving data quality for more accurate analysis of tweet sentiments, trends, and keywords.
- Getting the sentiment analysis score thanks to VADER library

```python
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer

nltk.download('stopwords')
nltk.download('wordnet')

def clean_text(text):
    # Remove URLs
    text = re.sub(r"http\S+|www\S+|https\S+", "", text, flags=re.MULTILINE)
    # Remove special characters and digits
    text = re.sub(r"[^\w\s]", "", text)
    text = re.sub(r"\d+", "", text)
    # Remove extra whitespace
    text = re.sub(r"\s+", " ", text)
    # Convert to lowercase
    text = text.lower()
    return text

df['Cleaned_Tweet'] = df['Tweet'].apply(clean_text)
df1['Cleaned_Tweet'] = df1['Tweet'].apply(clean_text)
```

	group_name	mean_forecast	mean_decision	forecasts_rank	decisions_rank	overall_rank
0	HIREUS	0.156724	5.911400	3.0	3.0	3.0
1	SP500	0.124043	4.355636	1.0	5.0	3.0
2	EW_LONG	0.160000	4.617711	5.0	4.0	4.5
3	GAMBLING	0.154019	2.684527	2.0	8.0	5.0
4	4SIGMA	0.231644	7.446886	13.0	1.0	7.0
5	ALPHA	0.182340	3.109273	9.0	7.0	8.0
6	CITADELSPINOFFS	0.232236	6.910374	14.0	2.0	8.0
7	HELLO_WORLD	0.186241	3.856796	10.0	6.0	8.0
8	RANDOM	0.171278	1.983802	7.0	9.0	8.0
9	MEDALLION	0.169759	0.977094	6.0	11.0	8.5
10	JAYSTREET	0.156935	-3.422412	4.0	14.0	9.0
11	LAMM	0.187100	-1.374405	11.0	12.0	11.5
12	UHCAKIP	0.174986	-9.378341	8.0	15.0	11.5
13	DYF	0.235294	1.317655	15.0	10.0	12.5
14	MSG	0.229598	-1.867828	12.0	13.0	12.5