Anomaly Detection and Variability Analysis – S&P 500 Index

Kyle Mollard (301560261)

Department of Computing Science, Simon Fraser University

CMPT 459: Data Mining Fall 2024

Martin Ester, TA: Arash Khoeini

November 27, 2024

1 Introduction

The objective of this project is to identify and analyze anomalies in the S&P 500 Index using a combination of data mining techniques. The S&P 500 is one of the most widely monitored financial indices, and anomalies in its performance often coincide with major economic events. Detecting these anomalies provides valuable insights into market risk, investor behavior, and economic conditions. For this purpose, we used three models—GARCH, ARIMA, and Isolation Forest—to detect significant deviations in market performance.

The dataset comprises daily closing values of the S&P 500 Index from 2015 to 2025. To evaluate the models' performance, we divided the dataset into two periods: pre-2019 for training and post-2019 for testing. This division allowed us to validate the models on unseen data while observing their ability to capture anomalies during known periods of market stress, such as the COVID-19 pandemic. This report details the methodologies, findings, and lessons learned from implementing these approaches.

2 Data Preprocessing and EDA

2.1 Problem Definition

The raw dataset, containing daily closing values for the S&P 500 Index, required preprocessing to prepare it for analysis. Daily returns were calculated as the percentage change in closing values to reflect market variability. The dataset was then split into a training set (2015–2018) and a testing set (2019–2025) to allow model evaluation on unseen data. Handling stationarity was critical for ARIMA modeling, and differencing techniques were applied to ensure that the data met the stationarity assumption. Additionally, the daily returns were standardized before applying the GARCH and Isolation Forest models to ensure consistency in scaling.

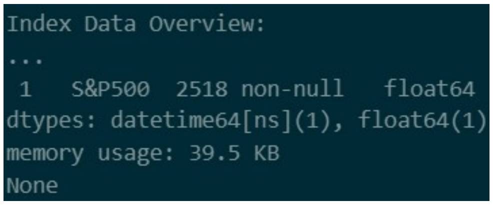
Through EDA, the datasets could be evaluated. The most updated versions of the datasets were found to be missing whole stock information, especially some of the top 15 weighted stocks to the S&P 500 index which led to much confusion and issues with continuing with the project. Further details to follow, but an older version of the data sets seemed to be complete.

2.2 Datasets

Version 960 of the datasets were eventually used and its results will be displayed below. Version 991 was initially used for EDA, but was found to not provide enough information. This was explored in section 2.3.4.

2.2.1 sp500_index.csv

This file contains historical daily values of the S&P 500 index. It has two columns: Date, representing the trading date, and S&P500, representing the index value on that day. This dataset serves as a benchmark for analyzing individual stock behavior and market-wide trends. It is particularly valuable for detecting periods of high volatility, identifying macroeconomic trends, and comparing the aggregated performance of the market with the performance of individual stocks. Time-series analysis techniques, such as ARIMA and GARCH, can be applied to forecast future index movements or assess historical volatility.



```
Index Data Overview:
...
 1   S&P500  2518 non-null   float64
dtypes: datetime64[ns](1), float64(1)
memory usage: 39.5 KB
None
```

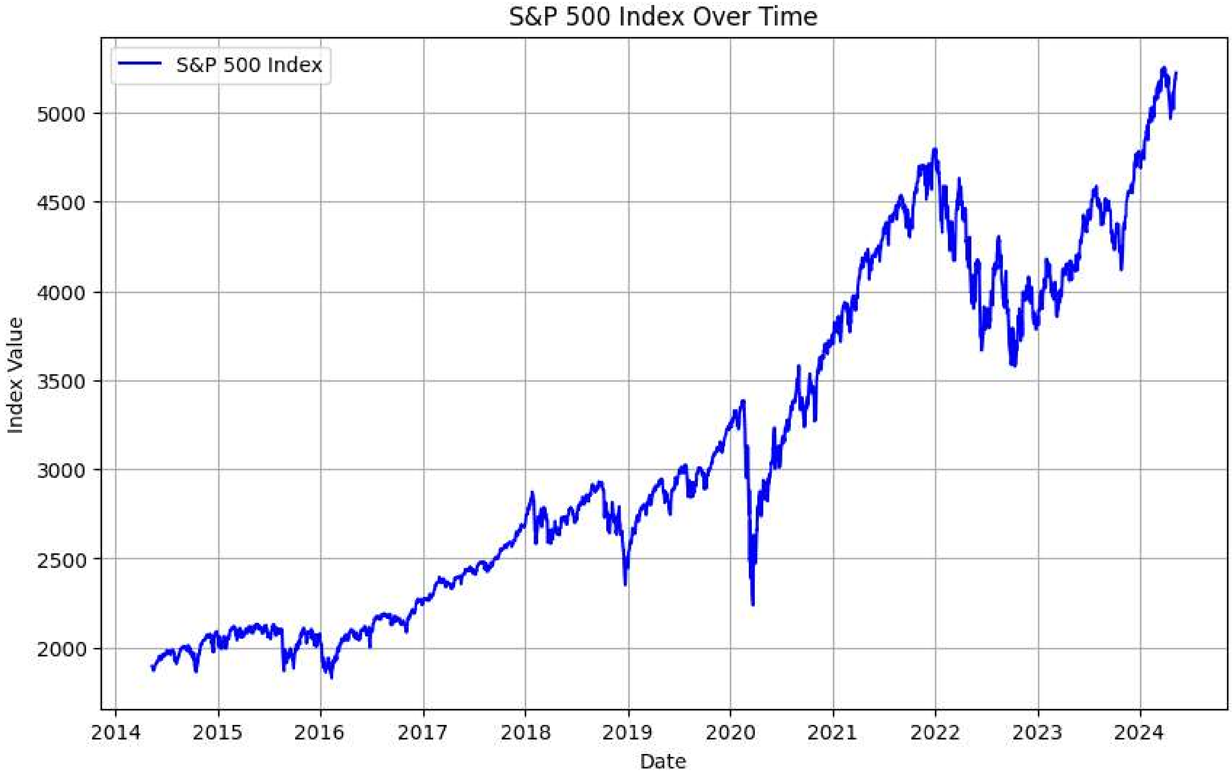
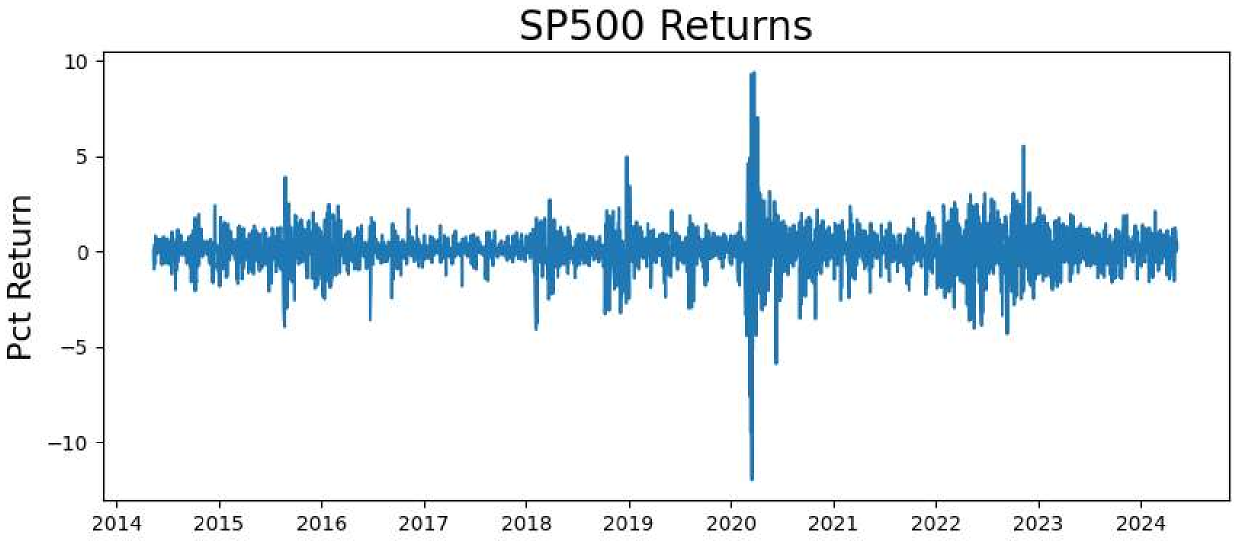
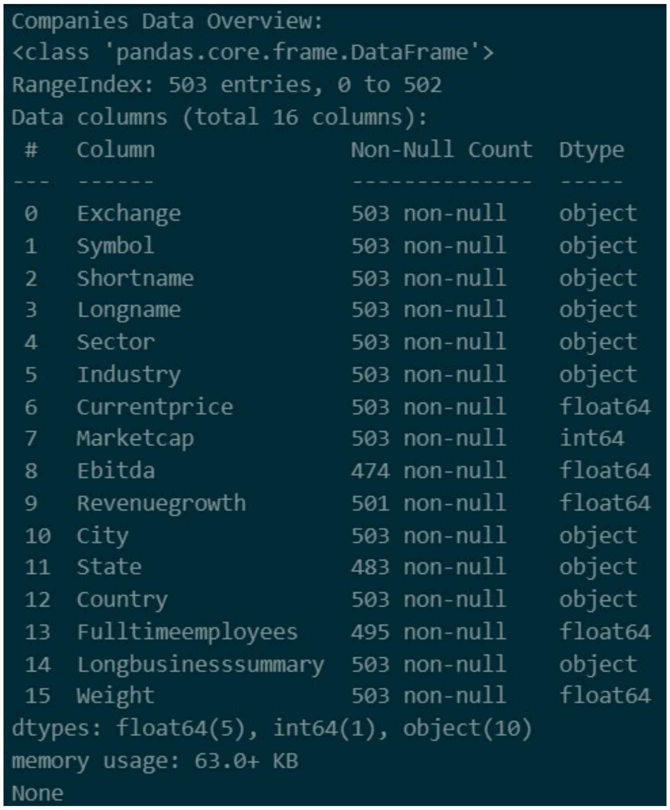


Figure 1 – S&P 500 Index Value from ~2015 to ~2025

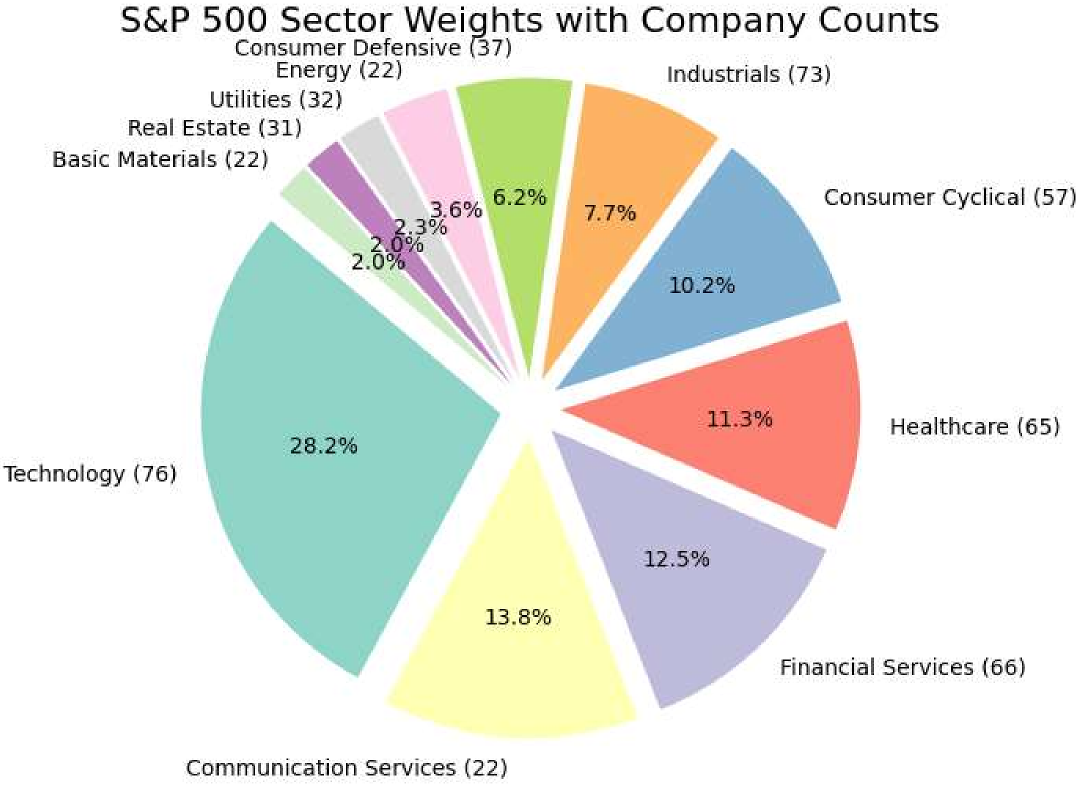


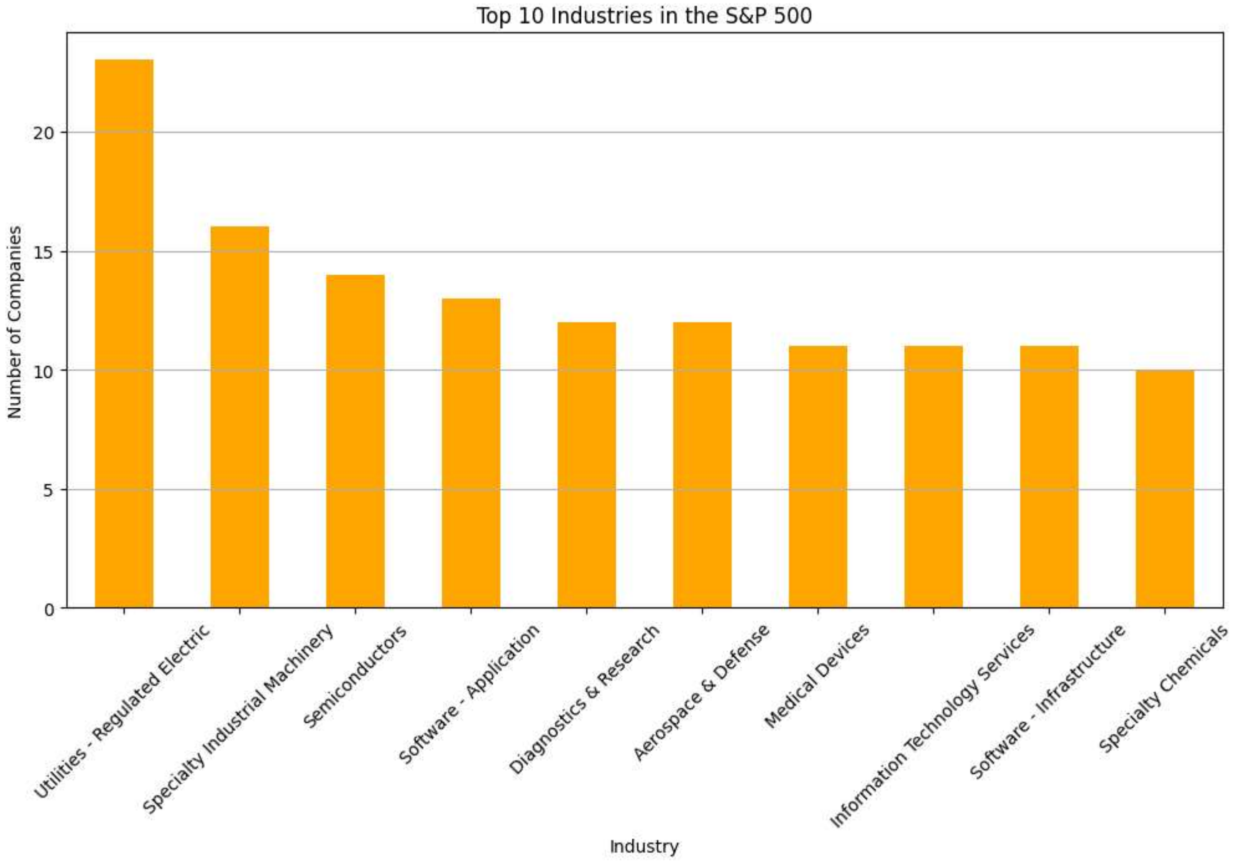
2.2.2 sp500_companies.csv

This file contains metadata about the companies that make up the S&P 500 index. Key columns include the company symbol (Symbol), sector (Sector), industry (Industry), market capitalization (Marketcap), and the weight of the company in the index (Weight). Additional details, such as geographical location (City, State, Country) and financial metrics like revenue growth and EBITDA, provide a comprehensive snapshot of each company's profile. This dataset is crucial for contextualizing the behavior of stocks within their sectors and industries and analyzing how different sectors contribute to the overall index. It also allows for grouping and clustering tasks by sector or industry to uncover patterns or anomalies within specific business domains.



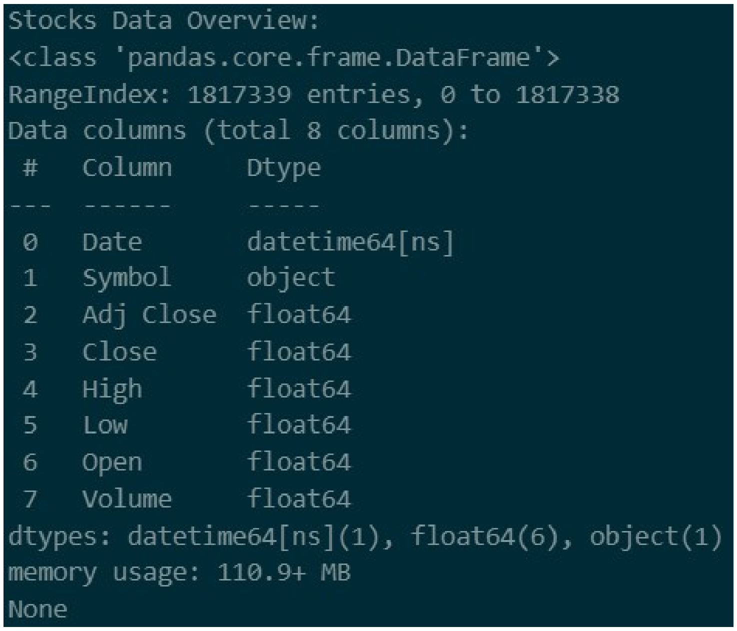
```
Companies Data Overview:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 503 entries, 0 to 502
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Exchange             503 non-null    object
 1   Symbol               503 non-null    object
 2   Shortname            503 non-null    object
 3   Longname             503 non-null    object
 4   Sector               503 non-null    object
 5   Industry             503 non-null    object
 6   Currentprice         503 non-null    float64
 7   Marketcap            503 non-null    int64
 8   Ebitda               474 non-null    float64
 9   Revenuegrowth        501 non-null    float64
 10  City                 503 non-null    object
 11  State                483 non-null    object
 12  Country              503 non-null    object
 13  Fulltimeemployees    495 non-null    float64
 14  Longbusinesssummary  503 non-null    object
 15  Weight               503 non-null    float64
dtypes: float64(5), int64(1), object(10)
memory usage: 63.0+ KB
None
```



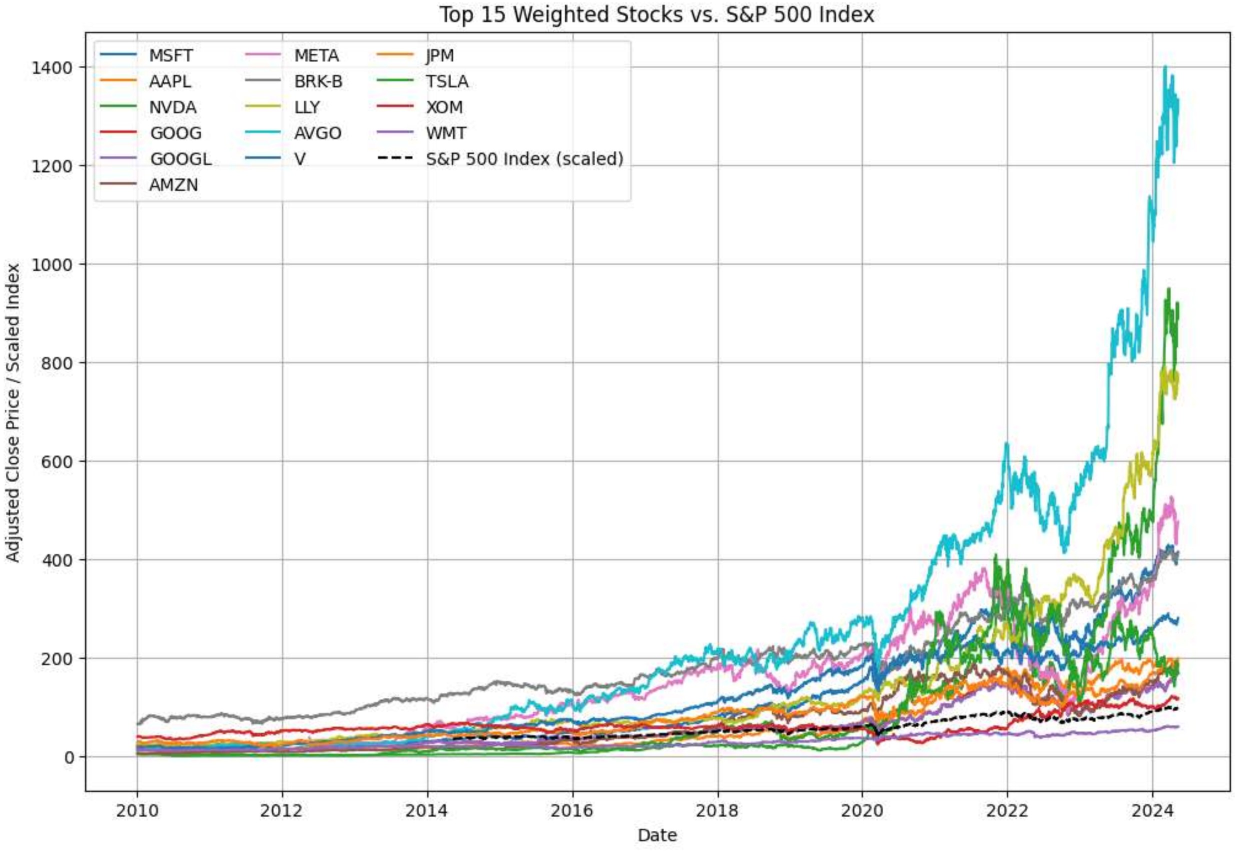


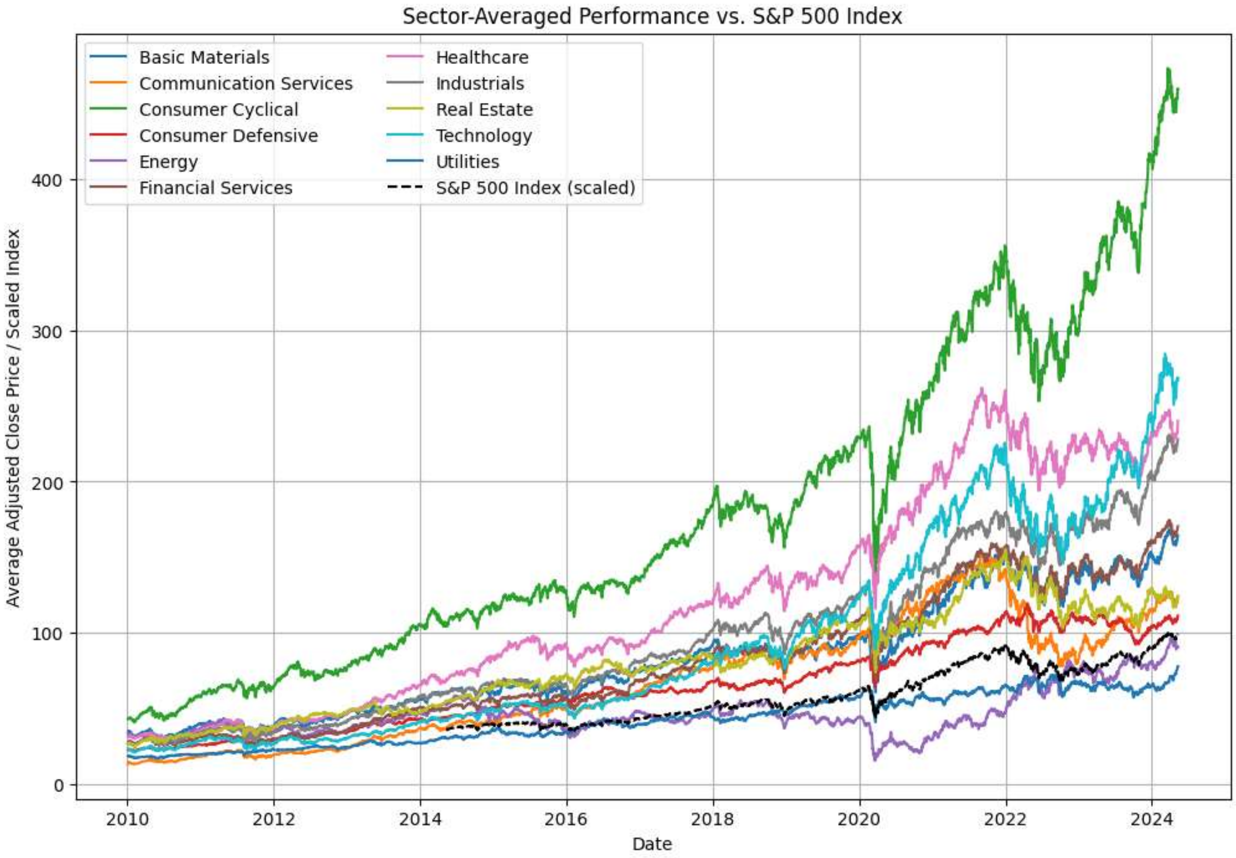
2.2.3 sp500_stocks.csv

This working dataset, version 960, contains detailed daily trading data for individual stocks in the S&P 500. Key columns include Date (trading date), Symbol (stock ticker), and price-related fields (Adj Close, Close, High, Low, Open). The Volume column indicates the number of shares traded on a given day. This dataset is the most granular of the three, enabling in-depth analysis of individual stock performance over time. It is ideal for applying data mining techniques such as outlier detection with Isolation Forest, clustering based on trading behavior, and time-series forecasting for individual stocks. When combined with sp500_companies.csv, this dataset allows for sectoral analysis, and when benchmarked against sp500_index.csv, it enables the detection of stock-specific anomalies relative to overall market trends.



```
Stocks Data Overview:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1817339 entries, 0 to 1817338
Data columns (total 8 columns):
 #   Column     Dtype
---  ------     -----
 0   Date       datetime64[ns]
 1   Symbol     object
 2   Adj Close  float64
 3   Close      float64
 4   High       float64
 5   Low        float64
 6   Open       float64
 7   Volume     float64
dtypes: datetime64[ns](1), float64(6), object(1)
memory usage: 110.9+ MB
None
```



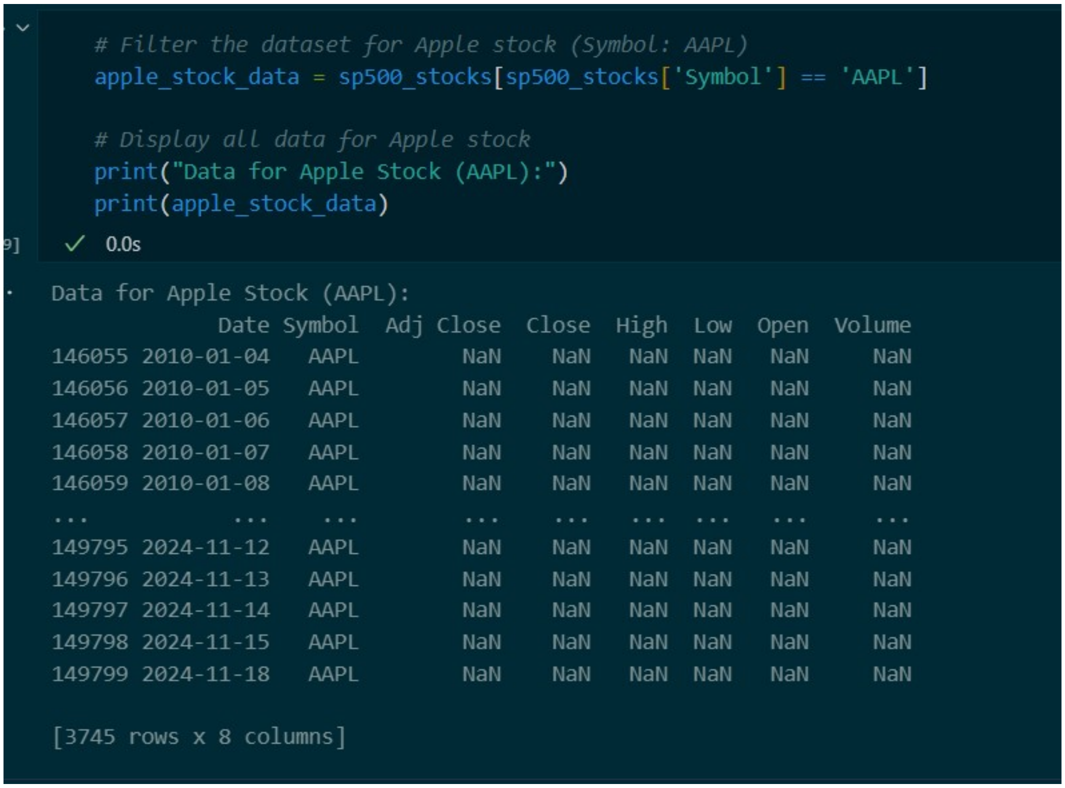


2.2.4 Missing Values Investigated Across Versions

Initially, the datasets were updated after the project proposal submission however, latest version of sp500_stocks.csv had an overwhelming number of missing values from important stocks such as APPL, AMZN, and NVDA to name a few. This was discovered quite late into the project as much of the model building was completed as a proof of concept with the sp500_index.csv with only the S&P 500 Index value.

Unfortunately, much time was wasted investigating other potential sources of the problem:

- Accidental overwriting of the csv file or data frame.
- Incorrect visualization code.
- Values were always missing
  - Installed the yfinance package, a yahoo finance stock data grabber package



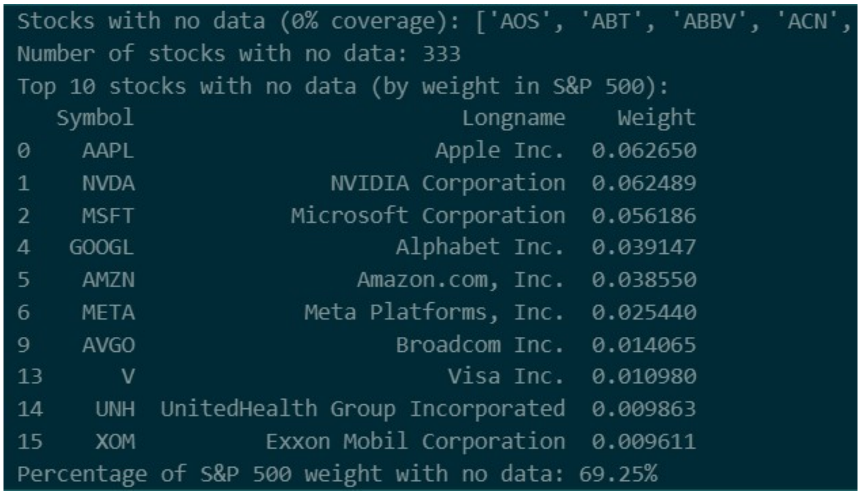
```
# Filter the dataset for Apple stock (Symbol: AAPL)
apple_stock_data = sp500_stocks[sp500_stocks['Symbol'] == 'AAPL']

# Display all data for Apple stock
print("Data for Apple Stock (AAPL):")
print(apple_stock_data)
```

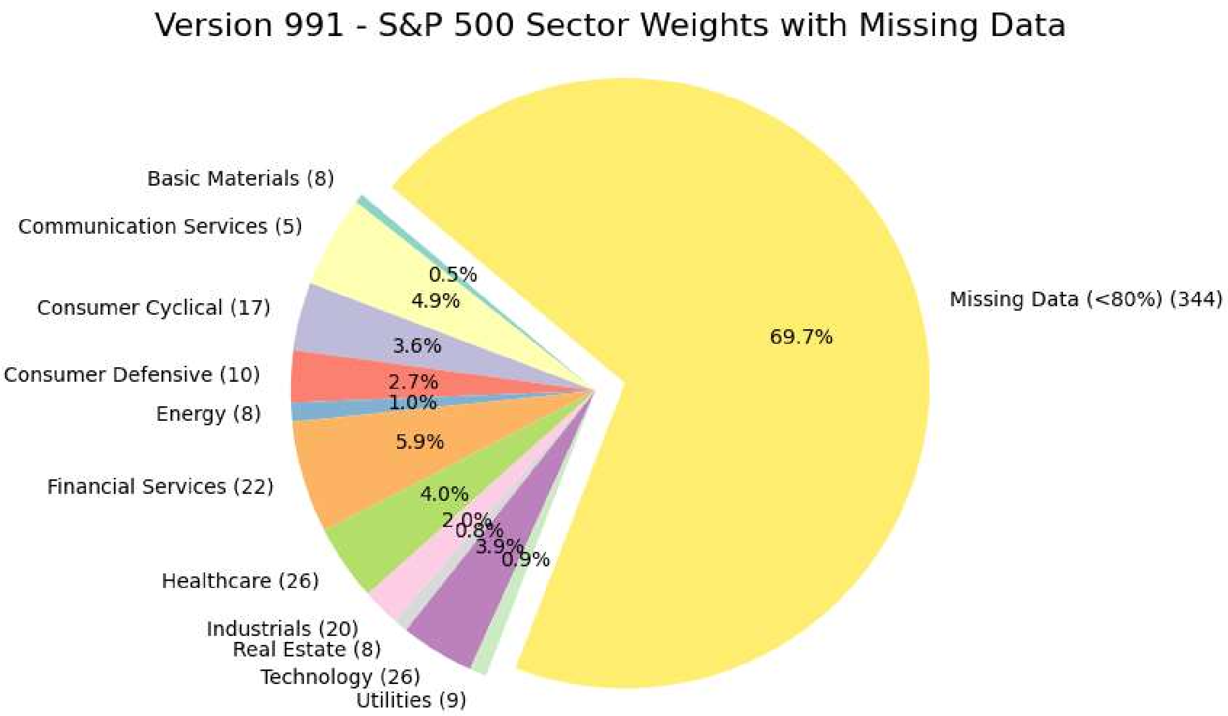
✓ 0.0s

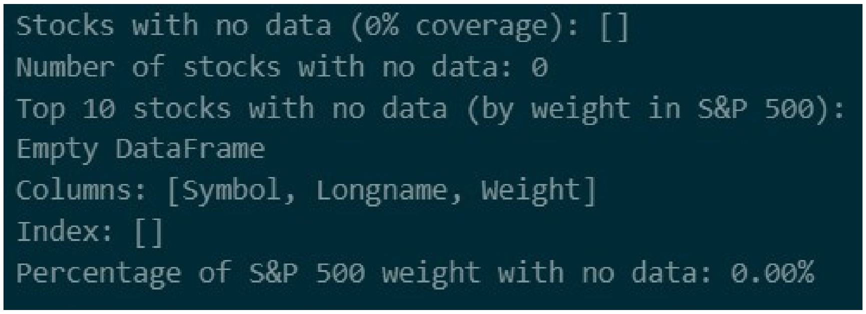
```
Data for Apple Stock (AAPL):
             Date Symbol  Adj Close  Close  High  Low  Open  Volume
146055 2010-01-04   AAPL        NaN    NaN   NaN  NaN   NaN     NaN
146056 2010-01-05   AAPL        NaN    NaN   NaN  NaN   NaN     NaN
146057 2010-01-06   AAPL        NaN    NaN   NaN  NaN   NaN     NaN
146058 2010-01-07   AAPL        NaN    NaN   NaN  NaN   NaN     NaN
146059 2010-01-08   AAPL        NaN    NaN   NaN  NaN   NaN     NaN
...           ...    ...        ...    ...   ...  ...   ...     ...
149795 2024-11-12   AAPL        NaN    NaN   NaN  NaN   NaN     NaN
149796 2024-11-13   AAPL        NaN    NaN   NaN  NaN   NaN     NaN
149797 2024-11-14   AAPL        NaN    NaN   NaN  NaN   NaN     NaN
149798 2024-11-15   AAPL        NaN    NaN   NaN  NaN   NaN     NaN
149799 2024-11-18   AAPL        NaN    NaN   NaN  NaN   NaN     NaN

[3745 rows x 8 columns]
```

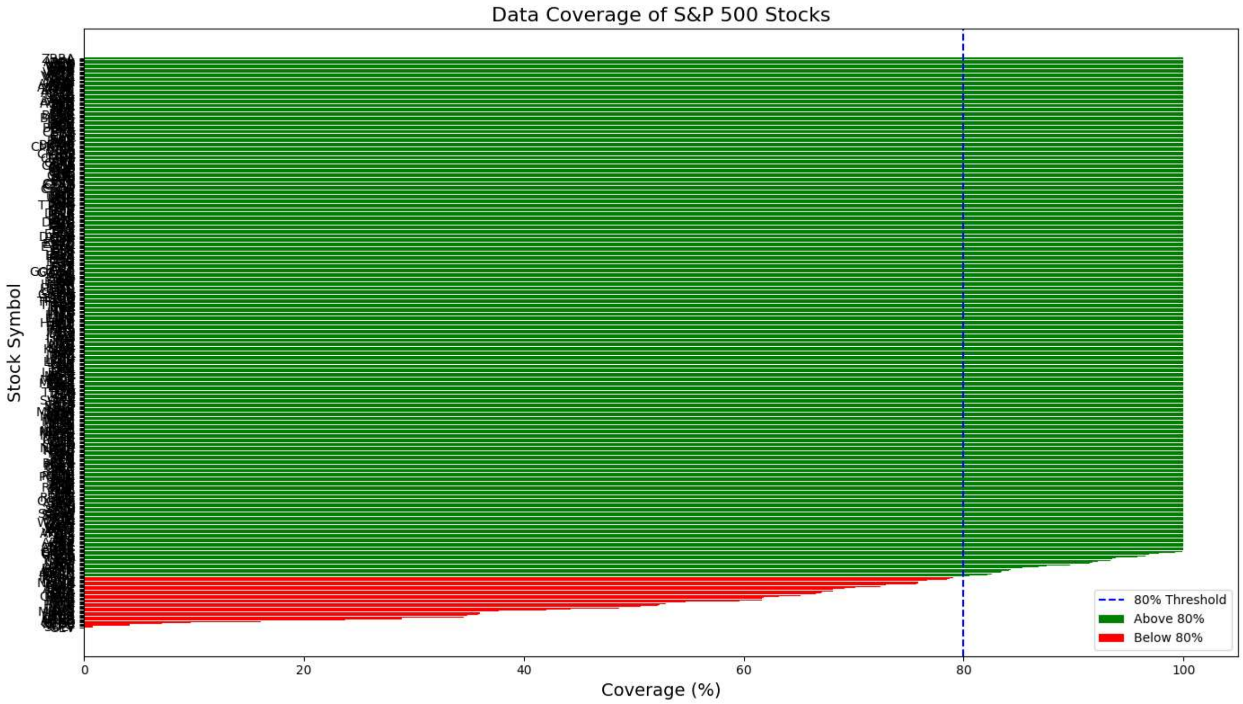


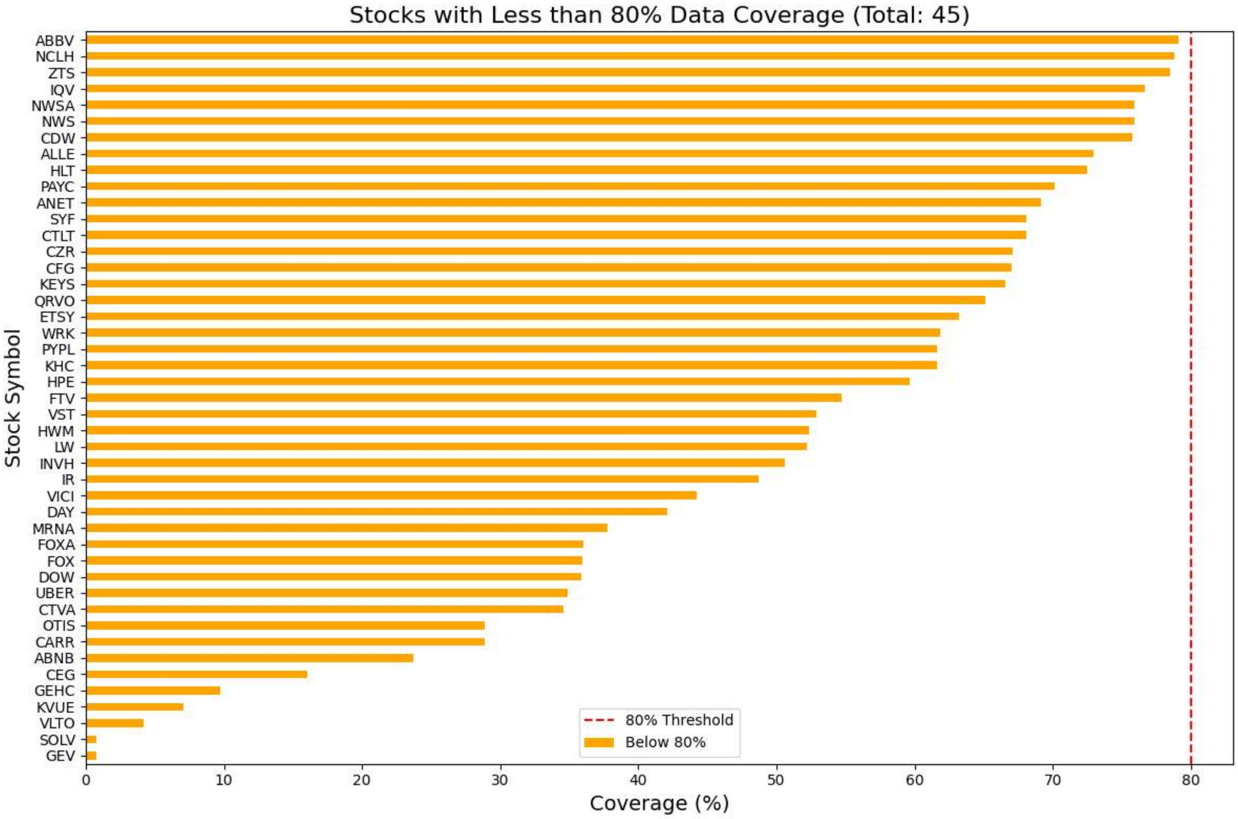
```
Stocks with no data (0% coverage): ['AOS', 'ABT', 'ABBV', 'ACN',
Number of stocks with no data: 333
Top 10 stocks with no data (by weight in S&P 500):
   Symbol                         Longname    Weight
0    AAPL                       Apple Inc.  0.062650
1    NVDA               NVIDIA Corporation  0.062489
2    MSFT            Microsoft Corporation  0.056186
4   GOOGL                    Alphabet Inc.  0.039147
5    AMZN                 Amazon.com, Inc.  0.038550
6    META            Meta Platforms, Inc.  0.025440
9    AVGO                    Broadcom Inc.  0.014065
13      V                        Visa Inc.  0.010980
14    UNH  UnitedHealth Group Incorporated  0.009863
15    XOM          Exxon Mobil Corporation  0.009611
Percentage of S&P 500 weight with no data: 69.25%
```

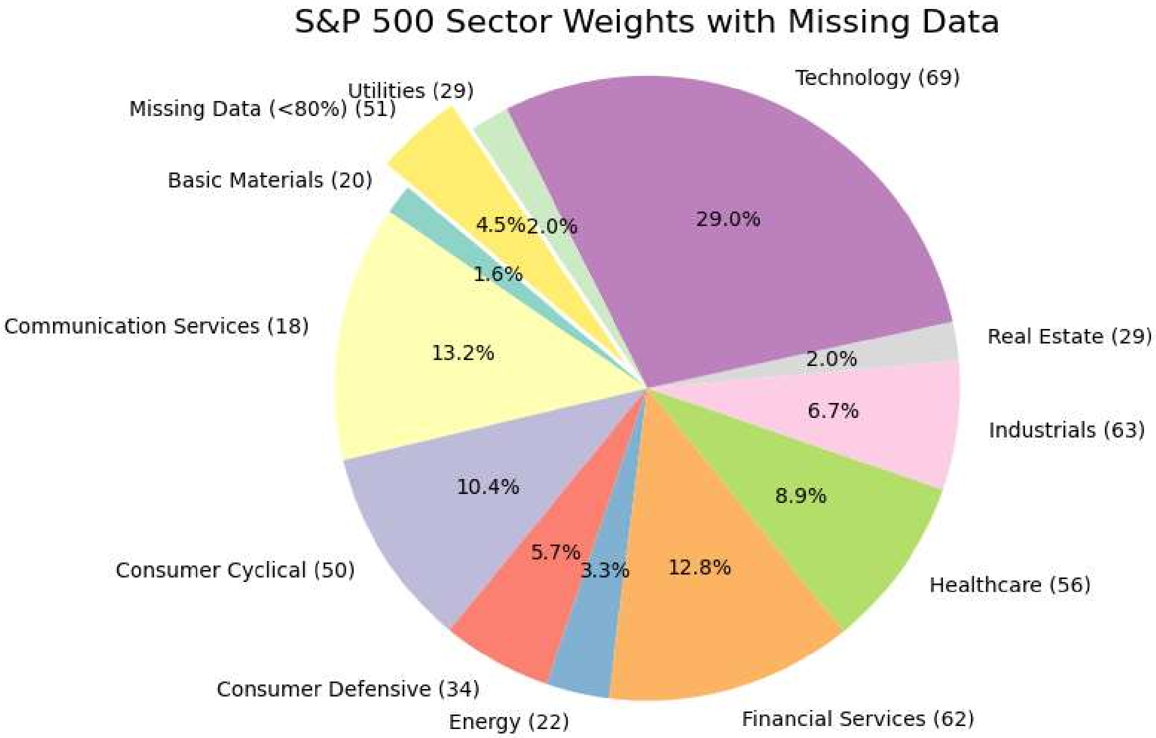




```
Stocks with no data (0% coverage): []
Number of stocks with no data: 0
Top 10 stocks with no data (by weight in S&P 500):
Empty DataFrame
Columns: [Symbol, Longname, Weight]
Index: []
Percentage of S&P 500 weight with no data: 0.00%
```

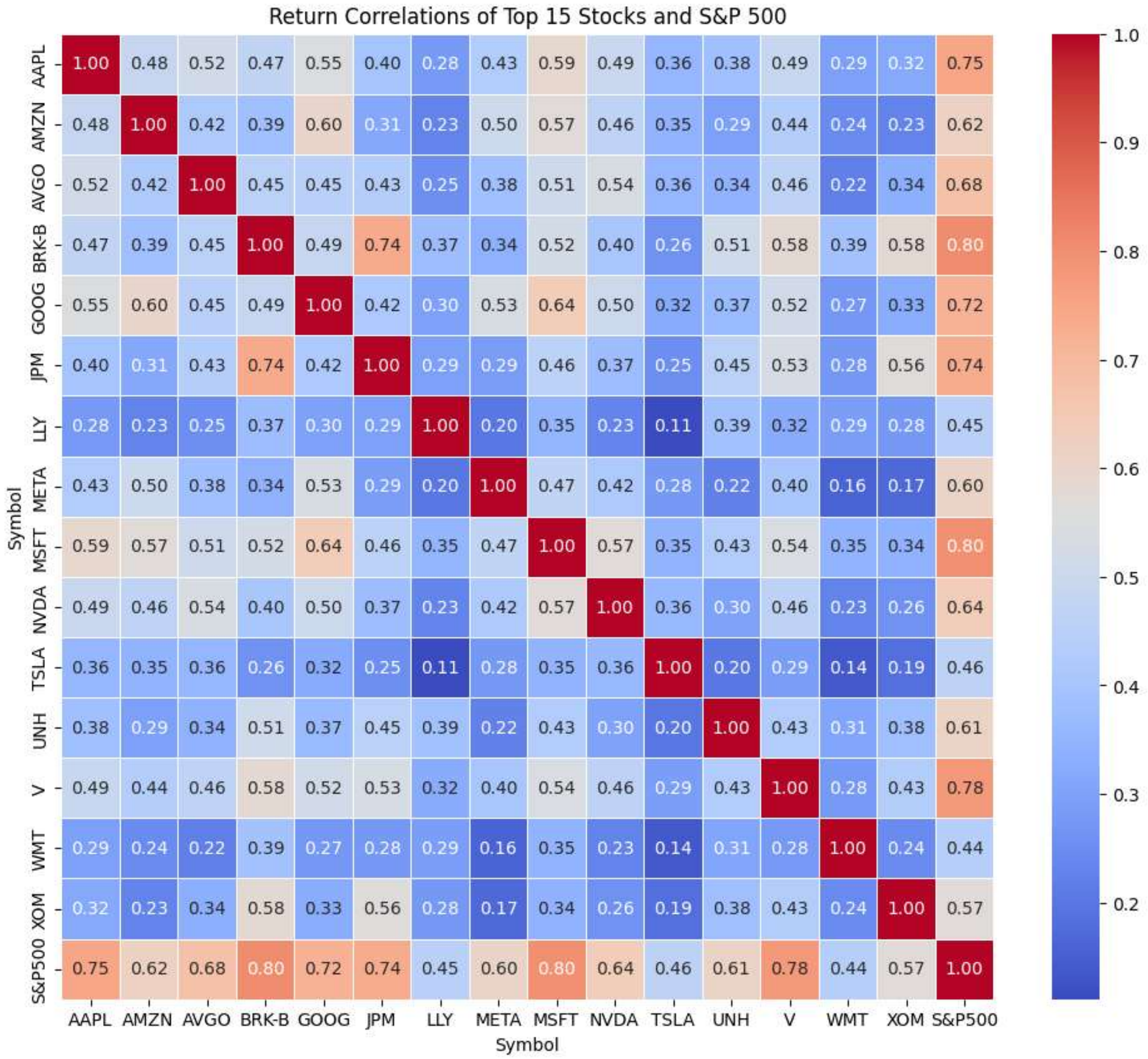

Data Coverage of S&P 500 Stocks
Stock Symbol
Coverage (%)
0
20
40
60
80
100
80% Threshold
Above 80%
Below 80%


Stocks with Less than 80% Data Coverage (Total: 45)
Stock Symbol
ABBV
NCLH
ZTS
IQV
NWSA
NWS
CDW
ALLE
HLT
PAYC
ANET
SYF
CTLT
CZR
CFG
KEYS
QRVO
ETSY
WRK
PYPL
KHC
HPE
FTV
VST
HWM
LW
INVH
IR
VICI
DAY
MRNA
FOXA
FOX
DOW
UBER
CTVA
OTIS
CARR
ABNB
CEG
GEHC
KVUE
VLTO
SOLV
GEV
0
10
20
30
40
50
60
70
80
Coverage (%)
80% Threshold
Below 80%



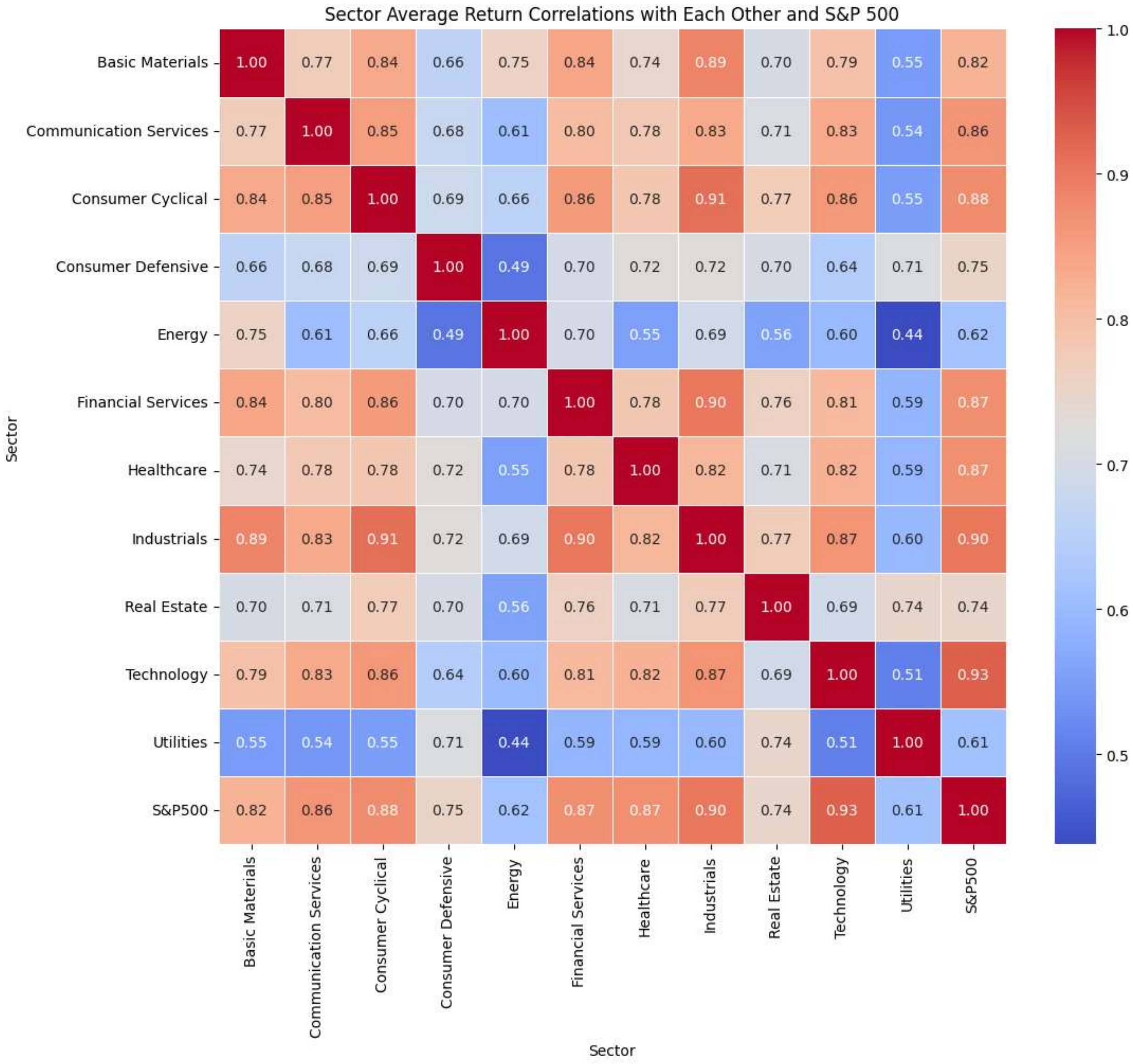
2.3 Heatmaps

This heatmap reveals the pairwise correlations among the top 15 weighted stocks in the S&P 500 index and their relationship with the S&P 500 index itself. The strong correlations (e.g., BRK-B with JPM and BRK-B with the S&P 500 index) suggest that certain stocks, such as those in similar industries or sectors, tend to move together. Stocks like LLY (Eli Lilly) show weaker correlations with others, likely because of its presence in a sector (Healthcare) that is less cyclical or driven by independent dynamics compared to the broader market. The overall correlation of the S&P 500 index with individual stocks varies but tends to be stronger with stocks like BRK-B, which have significant market influence due to their weight and diversification.



Symbol	AAPL	AMZN	AVGO	BRK-B	GOOG	JPM	LLY	META	MSFT	NVDA	TSLA	UNH	V	WMT	XOM	S&P500
AAPL	1.00	0.48	0.52	0.47	0.55	0.40	0.28	0.43	0.59	0.49	0.36	0.38	0.49	0.29	0.32	0.75
AMZN	0.48	1.00	0.42	0.39	0.60	0.31	0.23	0.50	0.57	0.46	0.35	0.29	0.44	0.24	0.23	0.62
AVGO	0.52	0.42	1.00	0.45	0.45	0.43	0.25	0.38	0.51	0.54	0.36	0.34	0.46	0.22	0.34	0.68
BRK-B	0.47	0.39	0.45	1.00	0.49	0.74	0.37	0.34	0.52	0.40	0.26	0.51	0.58	0.39	0.58	0.80
GOOG	0.55	0.60	0.45	0.49	1.00	0.42	0.30	0.53	0.64	0.50	0.32	0.37	0.52	0.27	0.33	0.72
JPM	0.40	0.31	0.43	0.74	0.42	1.00	0.29	0.29	0.46	0.37	0.25	0.45	0.53	0.28	0.56	0.74
LLY	0.28	0.23	0.25	0.37	0.30	0.29	1.00	0.20	0.35	0.23	0.11	0.39	0.32	0.29	0.28	0.45
META	0.43	0.50	0.38	0.34	0.53	0.29	0.20	1.00	0.47	0.42	0.28	0.22	0.40	0.16	0.17	0.60
MSFT	0.59	0.57	0.51	0.52	0.64	0.46	0.35	0.47	1.00	0.57	0.35	0.43	0.54	0.35	0.34	0.80
NVDA	0.49	0.46	0.54	0.40	0.50	0.37	0.23	0.42	0.57	1.00	0.36	0.30	0.46	0.23	0.26	0.64
TSLA	0.36	0.35	0.36	0.26	0.32	0.25	0.11	0.28	0.35	0.36	1.00	0.20	0.29	0.14	0.19	0.46
UNH	0.38	0.29	0.34	0.51	0.37	0.45	0.39	0.22	0.43	0.30	0.20	1.00	0.43	0.31	0.38	0.61
V	0.49	0.44	0.46	0.58	0.52	0.53	0.32	0.40	0.54	0.46	0.29	0.43	1.00	0.28	0.43	0.78
WMT	0.29	0.24	0.22	0.39	0.27	0.28	0.29	0.16	0.35	0.23	0.14	0.31	0.28	1.00	0.24	0.44
XOM	0.32	0.23	0.34	0.58	0.33	0.56	0.28	0.17	0.34	0.26	0.19	0.38	0.43	0.24	1.00	0.57
S&P500	0.75	0.62	0.68	0.80	0.72	0.74	0.45	0.60	0.80	0.64	0.46	0.61	0.78	0.44	0.57	1.00

This heatmap focuses on the correlations between average sector returns and the S&P 500 index. Sectors like Technology and Industrials exhibit strong correlations with the index, reflecting their significant contribution to overall market movements. Energy and Utilities, on the other hand, show weaker correlations with other sectors and the S&P 500, indicating that their performance is often influenced by sector-specific factors (e.g., oil prices, interest rates). This highlights the more defensive or isolated nature of these sectors compared to others like Financial Services or Technology, which are deeply integrated into market dynamics.



Sector Average Return Correlations with Each Other and S&P 500

Sector	Basic Materials	Communication Services	Consumer Cyclical	Consumer Defensive	Energy	Financial Services	Healthcare	Industrials	Real Estate	Technology	Utilities	S&P500
Basic Materials	1.00	0.77	0.84	0.66	0.75	0.84	0.74	0.89	0.70	0.79	0.55	0.82
Communication Services	0.77	1.00	0.85	0.68	0.61	0.80	0.78	0.83	0.71	0.83	0.54	0.86
Consumer Cyclical	0.84	0.85	1.00	0.69	0.66	0.86	0.78	0.91	0.77	0.86	0.55	0.88
Consumer Defensive	0.66	0.68	0.69	1.00	0.49	0.70	0.72	0.72	0.70	0.64	0.71	0.75
Energy	0.75	0.61	0.66	0.49	1.00	0.70	0.55	0.69	0.56	0.60	0.44	0.62
Financial Services	0.84	0.80	0.86	0.70	0.70	1.00	0.78	0.90	0.76	0.81	0.59	0.87
Healthcare	0.74	0.78	0.78	0.72	0.55	0.78	1.00	0.82	0.71	0.82	0.59	0.87
Industrials	0.89	0.83	0.91	0.72	0.69	0.90	0.82	1.00	0.77	0.87	0.60	0.90
Real Estate	0.70	0.71	0.77	0.70	0.56	0.76	0.71	0.77	1.00	0.69	0.74	0.74
Technology	0.79	0.83	0.86	0.64	0.60	0.81	0.82	0.87	0.69	1.00	0.51	0.93
Utilities	0.55	0.54	0.55	0.71	0.44	0.59	0.59	0.60	0.74	0.51	1.00	0.61
S&P500	0.82	0.86	0.88	0.75	0.62	0.87	0.87	0.90	0.74	0.93	0.61	1.00

3 Clustering

The methods applied to your financial time-series stock data provide significant insights by addressing core challenges in analyzing and interpreting such datasets. Clustering techniques, like DBSCAN and K-Means, help uncover patterns and relationships within the stock data. DBSCAN is particularly valuable for identifying outliers and handling irregular clusters, which can highlight unique opportunities or underperforming stocks that deviate from general trends. For example, DBSCAN can pinpoint hidden gems—stocks with extreme returns but low trading volumes—or identify consistent underperformers that may warrant attention. On the other hand, K-Means offers a simpler but effective way to group stocks, providing clear and interpretable clusters that can serve as a foundation for further analyses, such as differentiating between growth and value stocks. Evaluating clustering quality using metrics like the Silhouette Score ensures these groupings are meaningful and actionable.

4 Outlier Detection

4.1 Isolation Forest

Formula: Isolation Forest is based on the concept of isolating anomalies through recursive random splits in the data. It uses the **path length** from the root node to a leaf in a tree to measure the isolation of a data point. Anomalies are easier to isolate, resulting in shorter average path lengths. The anomaly score is calculated as:

$$Score = 2 - E\big(h(x)\big)c(n)$$

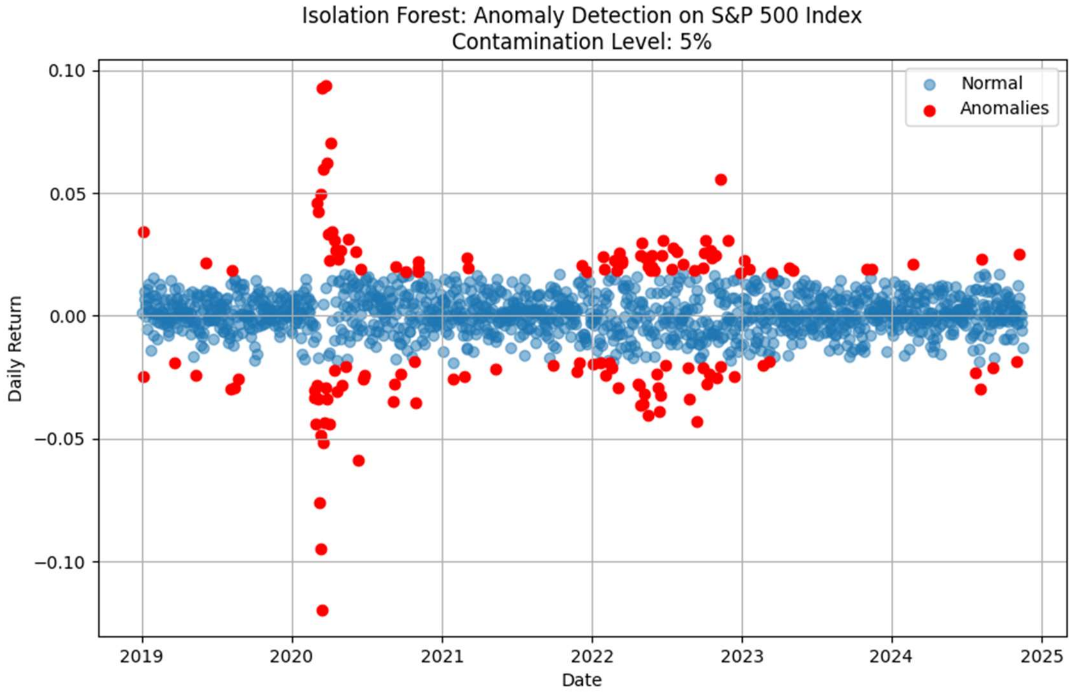
where E(h(x)) is the average path length for point x, and c(n) is the expected path length for a random binary tree of n observations.

Purpose: Designed for unsupervised anomaly detection, Isolation Forest excels at identifying rare, abnormal data points by isolating them through random splits. It is computationally efficient and handles high-dimensional datasets effectively.

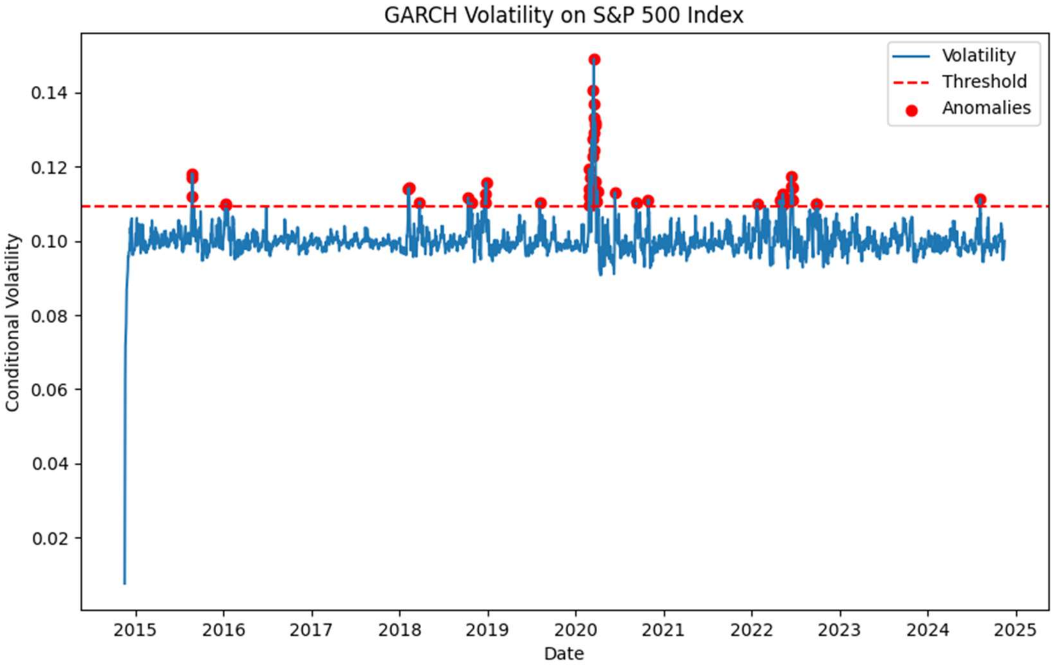
Constraints:

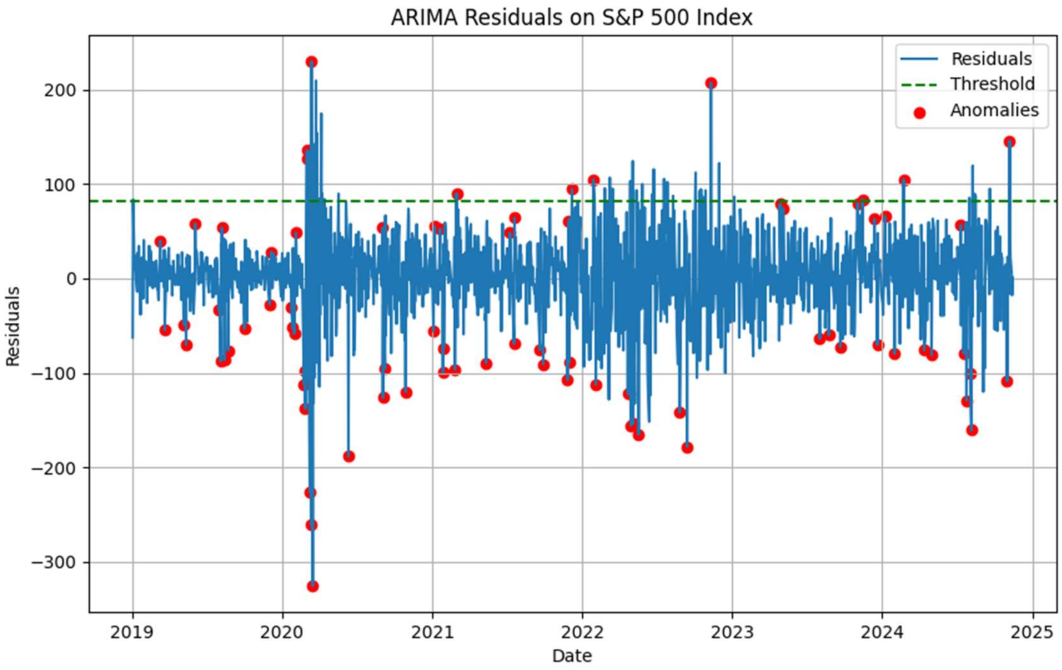
- **Insensitive to Feature Scaling**: While this can be an advantage, it may not capture relationships where scaling is critical.
- **Not a Forecasting Model**: It only detects anomalies and doesn’t predict future behavior.
- **Data Quality**: Requires clean data; it doesn’t handle missing values natively.

- **Assumes Anomalies are Few**: It works best when anomalies are a small proportion of the dataset.



4.2 Outlier Detection with ARIMA and GARCH





5 Feature Selection

Feature selection methods like Recursive Feature Elimination (RFE) and Lasso Regression help reduce noise and focus on the most impactful predictors in the dataset. RFE's ranking of features, for instance, aligns with financial intuition by identifying Return as a critical driver of stock behavior over Volume. This ranking allows you to prioritize features that matter most, making subsequent analyses and model building more efficient and interpretable. Lasso Regression, with its built-in regularization, further prevents overfitting and simplifies the dataset by highlighting the most predictive feature—in this case, Volume. Validating these feature selections through cross-validation ensures the selected features contribute effectively to prediction accuracy.

6 Classification

Classification techniques like Random Forest and k-Nearest Neighbors (k-NN) provide predictive power and interpretability, crucial for developing actionable insights. Random Forest stands out for its ability to handle complex, non-linear relationships and its robustness to noise, making it particularly suited to the volatile nature of financial data. Additionally, its feature importance metric offers transparency, showing exactly which variables drive predictions and enabling refined decision-making. Meanwhile, k-NN serves as a baseline method, useful for assessing whether simple models can deliver adequate predictive accuracy. Metrics such as precision, recall, and F1-score further evaluate model performance, ensuring predictions are reliable and minimizing costly false positives or missed opportunities.

7 Forecasting

This study employs three distinct methodologies to detect anomalies in the S&P 500 Index. The first approach, GARCH, focuses on modeling volatility to capture periods of high variability. The second, ARIMA, identifies anomalies in residuals by modeling trends and seasonality. Finally, Isolation Forest uses an unsupervised learning approach to detect outliers in daily returns.

The GARCH model was trained on the standardized daily returns of the training set to estimate conditional volatility. Anomalies in the test set were identified using a threshold of two standard deviations above the mean volatility. For ARIMA, differencing was applied to achieve stationarity before fitting the ARIMA(1,1,1) model. Anomalies were detected by comparing residuals to a rolling standard deviation threshold. Isolation Forest, trained on the training set, identified anomalies in the test set using a contamination parameter of 5%. The models were then evaluated based on their ability to capture known periods of high variability, such as the COVID-19 pandemic.

7.1 ARIMA

Formula: ARIMA combines three components:

1. **AutoRegressive (AR)**: Captures the influence of past values.

$$Xt = c + \sum i = 1p\phi iXt - i + \epsilon t$$

2. **Differencing (I)**: Makes the data stationary.

$$Xt' = Xt - Xt - 1$$

3. **Moving Average (MA)**: Models errors from previous forecasts.

$$Xt = \mu + \sum i = 1q\theta i\epsilon t - i + \epsilon t$$

Purpose: ARIMA is used for time series forecasting. It models linear relationships in stationary time series data, making it effective for short-term prediction of trends or patterns.

Constraints:

- **Stationarity Requirement**: Data must be stationary, requiring preprocessing (e.g., differencing or log transformation).

- **Linear Assumptions**: It cannot model nonlinear relationships.
- **Limited to Historical Data**: Does not account for external variables unless extended (e.g., ARIMAX).
- **Short-Term Focus**: Struggles with long-term predictions, especially in highly volatile or random data.

7.2 GARCH and ARCH

Formula: GARCH models the conditional variance of time series data as a function of past squared residuals and variances:

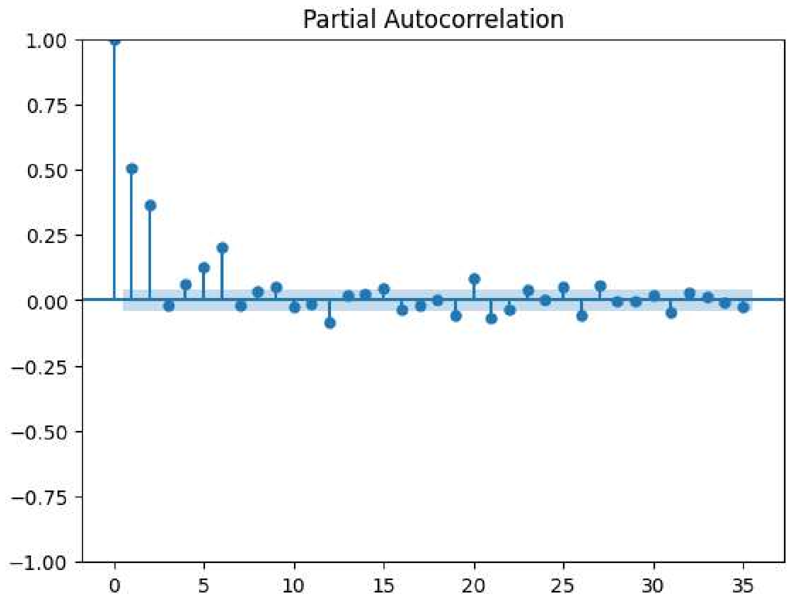
$$\sigma t2=\alpha 0+\sum i=1p\alpha i\epsilon t-i2+\sum j=1q\beta j\sigma t-j2$$

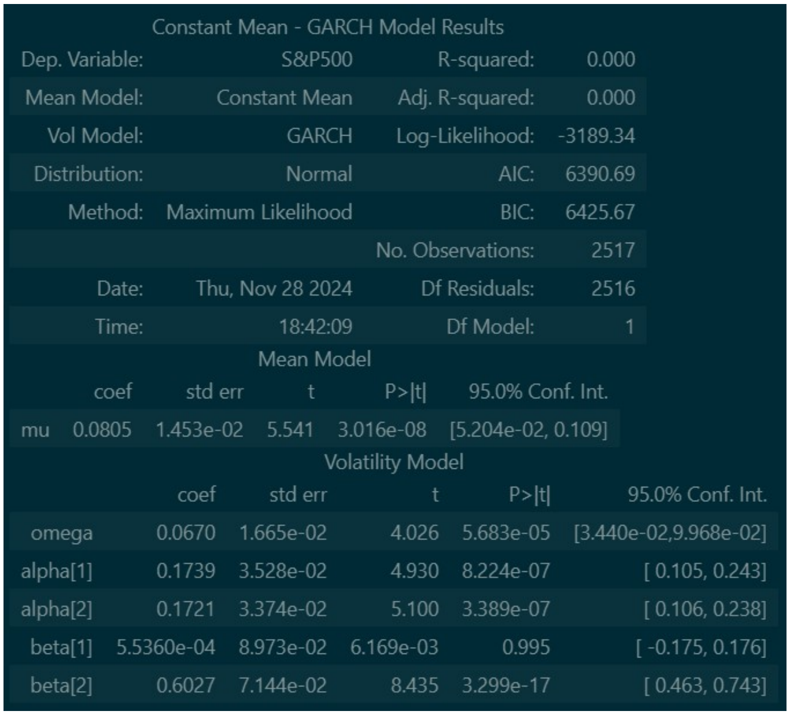
Where $\sigma t2$ is the conditional variance, ϵt is the residual, and αi,βj are coefficients.

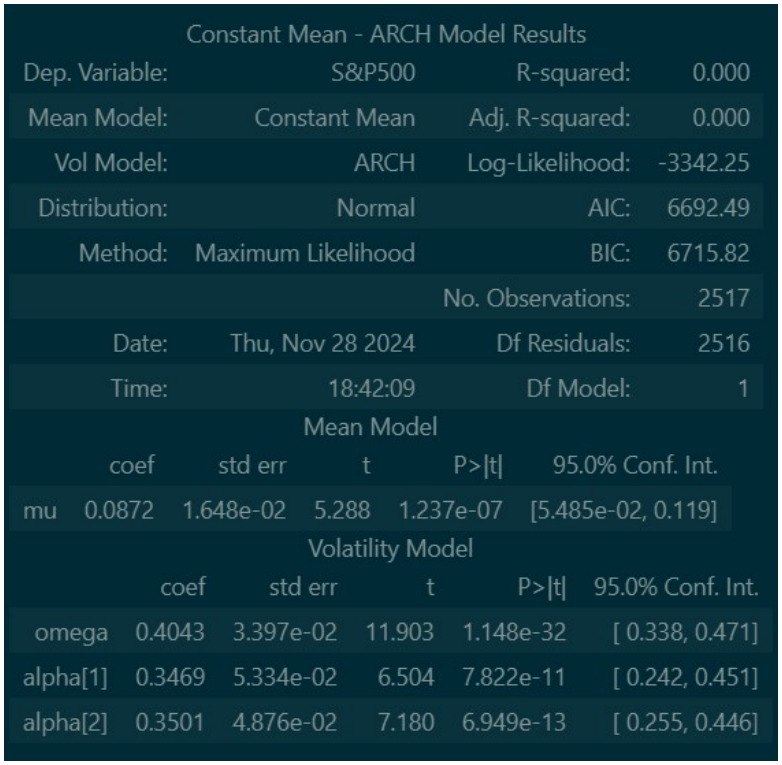
Purpose: GARCH is designed to model and forecast **volatility**, which is crucial in financial applications. It captures periods of high and low volatility (volatility clustering) and is commonly used for risk assessment and options pricing.

Constraints:

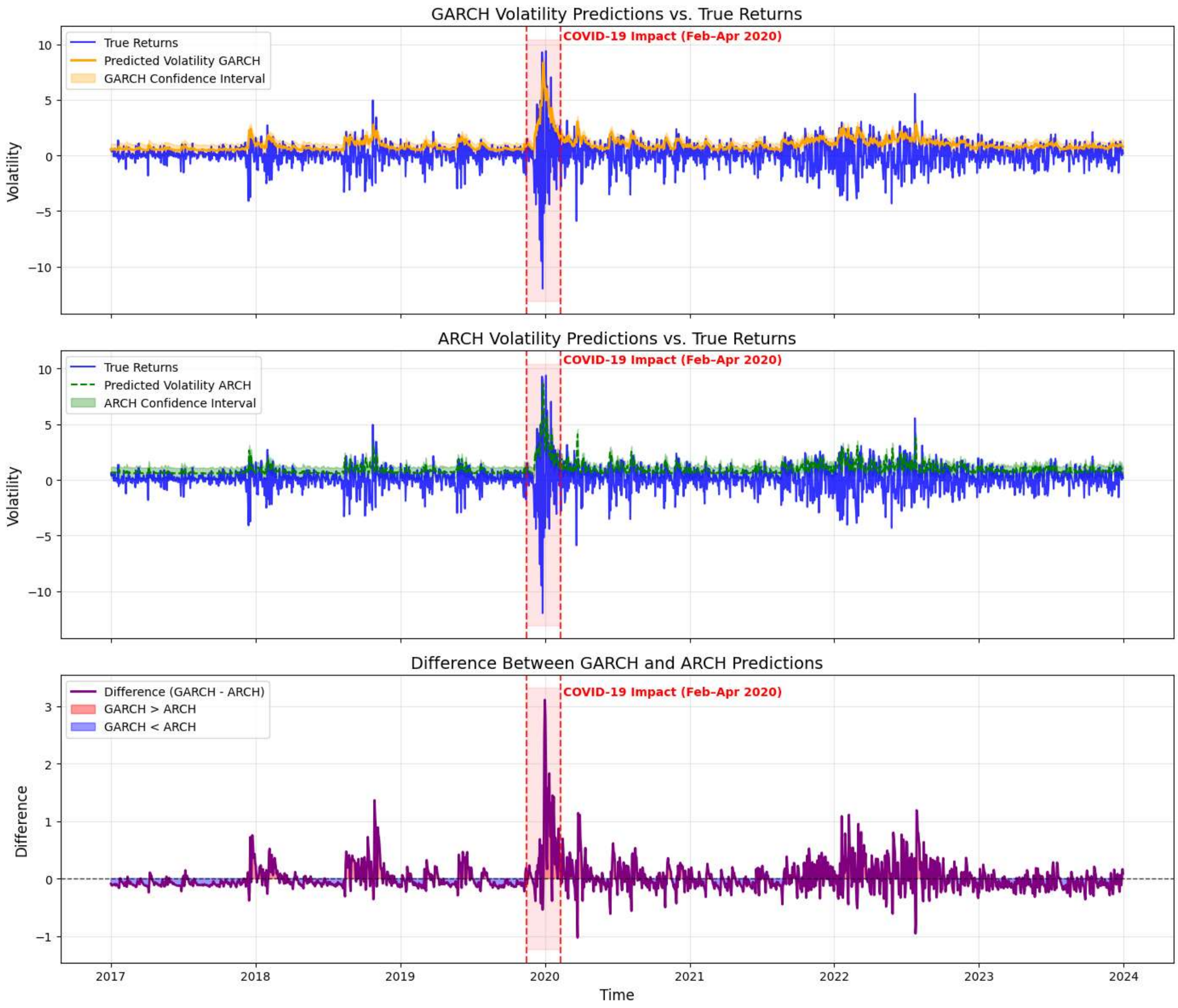
- **Assumes Stationarity**: Like ARIMA, requires stationary input data.
- **Sensitivity to Model Specification**: Incorrect selection of parameters pp and qq can lead to poor performance.
- **Does Not Model Mean**: It focuses exclusively on variance (volatility) and doesn’t predict actual values.
- **Limited for Sudden Jumps**: Struggles to capture abrupt, large shocks in volatility.

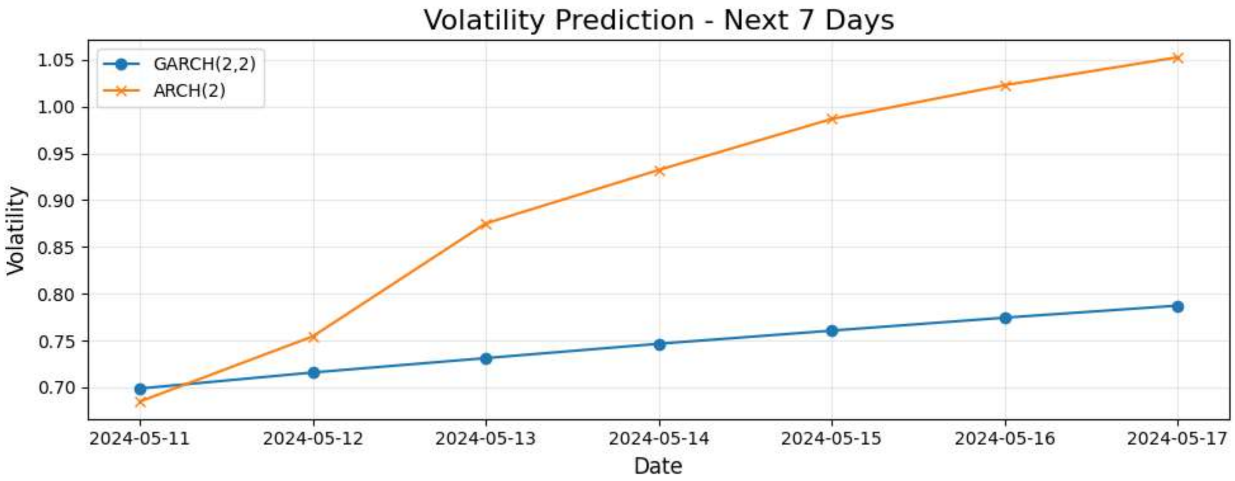



```
                    Constant Mean - GARCH Model Results
 Dep. Variable:              S&P500        R-squared:     0.000
   Mean Model:        Constant Mean    Adj. R-squared:     0.000
    Vol Model:                GARCH    Log-Likelihood:  -3189.34
 Distribution:               Normal              AIC:   6390.69
       Method:   Maximum Likelihood              BIC:   6425.67
                                    No. Observations:      2517
         Date:      Thu, Nov 28 2024    Df Residuals:      2516
         Time:              18:42:09        Df Model:         1
                              Mean Model
           coef     std err      t       P>|t|     95.0% Conf. Int.
 mu     0.0805   1.453e-02   5.541   3.016e-08   [5.204e-02, 0.109]
                                   Volatility Model
                   coef     std err          t        P>|t|       95.0% Conf. Int.
  omega          0.0670   1.665e-02      4.026    5.683e-05   [3.440e-02,9.968e-02]
 alpha[1]        0.1739   3.528e-02      4.930    8.224e-07          [ 0.105, 0.243]
 alpha[2]        0.1721   3.374e-02      5.100    3.389e-07          [ 0.106, 0.238]
  beta[1]    5.5360e-04   8.973e-02  6.169e-03        0.995         [ -0.175, 0.176]
  beta[2]        0.6027   7.144e-02      8.435    3.299e-17          [ 0.463, 0.743]
```



```
                    Constant Mean - ARCH Model Results
 Dep. Variable:                 S&P500        R-squared:     0.000
   Mean Model:           Constant Mean   Adj. R-squared:     0.000
    Vol Model:                    ARCH   Log-Likelihood:  -3342.25
 Distribution:                  Normal              AIC:   6692.49
       Method:      Maximum Likelihood              BIC:   6715.82
                                      No. Observations:      2517
         Date:        Thu, Nov 28 2024     Df Residuals:      2516
         Time:                18:42:09         Df Model:         1
                              Mean Model
         coef      std err        t       P>|t|     95.0% Conf. Int.
 mu   0.0872    1.648e-02    5.288   1.237e-07   [5.485e-02, 0.119]
                           Volatility Model
              coef     std err        t       P>|t|    95.0% Conf. Int.
   omega    0.4043   3.397e-02   11.903   1.148e-32     [ 0.338, 0.471]
 alpha[1]   0.3469   5.334e-02    6.504   7.822e-11     [ 0.242, 0.451]
 alpha[2]   0.3501   4.876e-02    7.180   6.949e-13     [ 0.255, 0.446]
```





The volatility forecast for the next 7 days reveals a clear difference between the predictions from the GARCH(2,2) and ARCH(2) models. The GARCH(2,2) model predicts a gradual increase in volatility, stabilizing at lower levels, which reflects its ability to capture both short-term market shocks and the long-term persistence of volatility. In contrast, the ARCH(2) model forecasts a sharper and more pronounced rise in volatility over the same period, indicating that it reacts more strongly to recent changes in returns but lacks the ability to smooth these effects over time. This divergence highlights the distinct strengths of each model: GARCH(2,2) provides a more conservative and stable forecast, suitable for long-term risk management, while ARCH(2) is more reactive and better suited for short-term market analysis. The differences suggest that incorporating past volatility persistence, as GARCH does, can lead to more tempered predictions in periods of heightened market uncertainty.

8 Hyperparameter Tuning

5. Results

5.1 Initial Testing of S&P 500 Alone

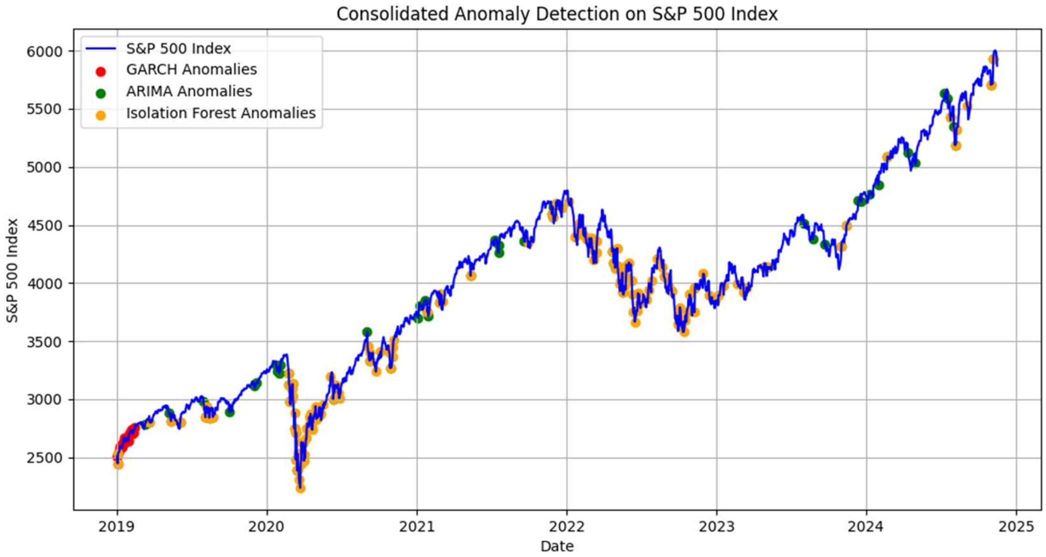
In The results from each methodology highlight their unique strengths and limitations.

The GARCH model successfully identified high-volatility periods, such as the market crash in March 2020. However, its estimates quickly stabilized after periods of high variability, potentially missing anomalies during slower market movements. Experimentation with higher-order GARCH models may improve its sensitivity to subtle changes.

The ARIMA model performed well in capturing anomalies in the residuals, particularly during significant market events. Rolling thresholds allowed for adaptive anomaly detection; however, residual noise introduced false positives, requiring further refinement in threshold selection and preprocessing.

The Isolation Forest model provided a robust unsupervised approach to detecting outliers in daily returns. Anomalies were concentrated around known periods of market stress, demonstrating the model's effectiveness. However, the contamination parameter influenced the frequency of anomalies, highlighting the need for parameter tuning.

A consolidated anomaly detection plot combined the outputs of all three models, revealing overlaps and inconsistencies. Overlapping anomalies suggest consistent detection of major events, while discrepancies indicate that each model captures unique aspects of market behavior.



5.2 Discussion

These methods collectively address critical questions in financial time-series analysis. Clustering organizes stocks into actionable groups, revealing market dynamics and segmentation opportunities. Feature selection ensures focus on the most impactful variables, enhancing interpretability and efficiency. Classification, particularly with Random Forest, provides robust and explainable predictions that can guide investment strategies or trading decisions. Together, these approaches offer a comprehensive framework for uncovering insights and building data-driven strategies in financial markets.

The findings demonstrate the utility of combining multiple methods for financial anomaly detection. GARCH is well-suited for capturing volatility, ARIMA excels in trend-based anomaly detection, and Isolation Forest effectively identifies outliers. Together, they provide a comprehensive view of market anomalies.

Challenges encountered include ensuring stationarity for ARIMA, stabilizing volatility estimates for GARCH, and tuning the contamination parameter for Isolation Forest. Additionally, visualizing anomalies across methods required standardization of timeframes and scales to ensure consistency.

Future work could incorporate additional features, such as macroeconomic indicators or sector-level data, to enhance anomaly detection. Ensemble methods that combine the strengths of all three approaches may also improve overall performance.

6 Conclusion

This project highlights the strengths and weaknesses of GARCH, ARIMA, and Isolation Forest for detecting anomalies in the S&P 500 Index. While each model has limitations, their combined use provides a robust framework for identifying periods of high variability. These insights can inform risk management strategies and improve our understanding of market behavior during periods of stress.

6.1 Future Investigations

- Greater resolution with Isolated Forest if possible
    - To see if it could predict just as well, during a regular trading day.
- Domain knowledge to offer advice for observer other than noting the expected and observed change and range of volatility to identify outliers

7 References

[1] pandas, Available: https://pandas.pydata.org/

[2] NumPy, Available: https://numpy.org/

[3] Matplotlib: Visualization with Python, Available: https://matplotlib.org/

[4] anndata, Available: https://anndata.readthedocs.io/

[5] Scanpy, Available: https://scanpy.readthedocs.io/

[6] scikit-learn, Available: https://scikit-learn.org/

[7] "K-means Clustering From Scratch In Python [Machine Learning Tutorial]," YouTube, Available: https://www.youtube.com/watch?v=lX-3nGHDhQg

[8] "StatQuest: K-means clustering**,”** YouTube, Available: https://www.youtube.com/watch?v=4b5d3muPQmA