Covid19 Impact on US Stocks

Teja Potu
tp22o@fsu.edu
Florida state university
Tallahassee, Florida, USA

Puneeth Reddy Motukuru Damodar
pm22p@fsu.edu
Florida state university
Tallahassee, Florida, USA

Venkata Sivasai Phani Praveen Mamillapalli
vm22d@fsu.edu
Florida state university
Tallahassee, Florida, USA

MD. Masum Al Masba
ma22be@fsu.edu
Florida state university
Tallahassee, Florida, USA

Abstract

The COVID-19 pandemic has had a profound impact on the global economy, causing unprecedented levels of volatility and uncertainty in the US stock market. As a result, data-driven analyses have been instrumental in understanding the impact of COVID-19 on US stocks. This research paper provides a comprehensive overview of the techniques utilized for data analysis, including exploratory data analysis, visualization, and machine learning algorithms. By leveraging these tools, researchers can gain insights into the changes in the stock market during the pandemic, helping investors and policymakers make informed decisions. The findings from this research provide a deeper understanding of the impact of COVID-19 on the US economy and can inform future economic policies aimed at mitigating the negative effects of similar crises. Ultimately, this paper aims to contribute to the body of knowledge on the impact of global pandemics on the stock market and the economy.

ACM Reference Format:
Teja Potu, Puneeth Reddy Motukuru Damodar, Venkata Sivasai Phani Praveen Mamillapalli, and MD. Masum Al Masba. 2018. Covid19 Impact on US Stocks. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

The stock markets and the global economy have both been significantly impacted by the COVID-19 epidemic. Since the epidemic started, there has been a great amount of volatility and uncertainty in the financial market, particularly in the US. We'll look at some of the ways that COVID-19 has affected the US stock market in this response.

First off, the pandemic sparked a general state of anxiety and fear that significantly lowered market prices in February and March 2020. During this time, the Dow Jones Industrial Average (DJIA) and the SP 500 both saw some of their biggest point declines in a single day in history. Stocks across all sectors were sold off as a result of investors' worries about how the epidemic will affect companies and the economy as a whole.

Certain industries have been significantly impacted by the epidemic, which has had an effect on the stock market. For instance, the travel and tourism sector has been severely impacted by the pandemic as a result of the ban on public gatherings and travel, which has resulted in a sharp drop in income for airlines, hotels, and other related enterprises. The drop in economic activity has resulted in a decrease in demand for oil and other energy products, which has had an effect on the energy industry as well. Due to the fact that numerous businesses in these sectors are included in the SP 500, this has had an impact on the entire economy.

The stock market has been impacted by how governments and central banks have responded to the pandemic. Governments all across the world have created fiscal stimulus programs to aid those impacted by the pandemic, both personally and professionally. To help the economy, central banks have also slashed interest rates and conducted quantitative easing programs. Whilst there is still a lot of uncertainty about the future, these actions have helped to stabilize the stock market to some extent.

2 LITERATURE SURVEY

The COVID-19 pandemic has caused a significant impact on the global economy, with various industries and sectors experiencing unprecedented changes. One area that has been significantly affected is the stock market. Several studies have examined the impact of the pandemic on different stock markets, including the Japanese and Egyptian markets. These studies have explored factors affecting stock returns, such as ownership structure, investor behavior, and dividend policy. The findings suggest that the COVID-19 pandemic has

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

had a significant impact on stock markets worldwide, and investors should carefully consider various factors before making investment decisions during these challenging times. In this response, we will review and analyze the findings of several studies on the impact of the COVID-19 pandemic on various stock markets, providing insights on factors that influence stock returns during this time.

The paper[5], examines the factors affecting the Japanese stock market during the COVID-19 pandemic period. It focuses on three main factors. First, it examines the ownership structure of listed companies in Japan. The research identifies that indirect ownership through the exchange-traded fund purchasing program by the Bank of Japan (BOJ) has a positive impact on abnormal returns. Traditional business group ownership is positively associated with abnormal returns, while foreign ownership is negatively associated with abnormal returns. Second, the article analyzes the impact of global value chains and finds that stock returns are lower for companies with China and U.S. exposure. Third, in terms of environmental, social, and governance (ESG) engagement, there is no evidence that firms with highly rated ESG scores have higher abnormal returns, but firms with ESG funds outperform those without. The study suggests that the existence of long-term investors, such as the BOJ, can help to curb downward pressure on stock prices during a crisis.

The paper[3] aims to explore the relationship between investors' demographic characteristics (age, gender, education level, and experience) and their investment decisions through behavioral factors (sentiment, overconfidence, overreaction and underreaction, and herd behavior) as mediator variables in the Egyptian stock market. The paper collects data from a structured questionnaire survey carried out among 384 local Egyptian, foreign, institutional, and individual investors. The paper finds that investor sentiment, overreaction and underreaction, overconfidence, and herd behavior significantly affect investment decisions. Moreover, age, gender, and the level of education have significant positive effects on investment decisions by investors. Experience does not play a significant role in investment decisions, but as investors gain experience, they tend to overlook the emotional factors. The findings of this paper would help to understand common behavioral patterns of investors and indicate a path toward the growth of the Egyptian stock market. The paper concludes by stating that there is a lack of research in behavioral finance covering Middle East and North African markets, and this paper attempts to fulfill the gap by analyzing behavioral factors in the Egyptian market.

The COVID-19 pandemic has had a significant impact on the global economy, and this impact is reflected in the stock prices of companies. A study [2] shows that there is a significant difference in daily closing stock prices and volume of stock trade before and after the COVID-19 pandemic, strengthening the efficient market hypothesis . Dr Tedros Adhanom Ghebreyesus, Director-General of the World Health Organization, stresses the importance of data and science in building resilient health systems and accelerating towards global health goals, emphasizing that all countries must have the necessary capacity and resources to collect and use health data even in the midst of an ongoing crisis like the COVID-19 pandemic. The pandemic has caused a dramatic loss of human life worldwide and has had devastating economic and social disruptions, with tens of millions of people at risk of falling into extreme poverty . The economic effects of the pandemic are reflected in the stock prices of companies, and investors should be careful when choosing to invest. Investors should choose customer goods sector companies that provide essential products like pharmacy, food, and beverages.

This paper [7] analyzes the variables that significantly affect dividend policy in manufacturing companies in Indonesia's stock exchange. The research uses a comparative causal research approach and investigates collateralizable assets, growth in net assets, liquidity, leverage, and profitability as independent variables. The results show that growth in net assets, liquidity, and leverage have a negative and significant effect on dividend policy, while collateralizable assets and profitability have a negative but not significant effect on dividend policy. The study provides empirical evidence on the impact of these variables on dividend policy in an emerging market.

This article [1] discusses the impact of the COVID-19 outbreak on China's economy and Asian stock markets using an event study method. The study found that both markets experienced significant declines, with negative cumulative abnormal returns in all examined event windows. The article also analyzes industry index responses to the epidemic, finding that pharmaceutical manufacturing, software, and IT services had positive CAR, while transportation, lodging, and catering had negative CAR. These results reflect investors' expectations for different industries and the economy under the outbreak of COVID-19.

This article [4] discusses the importance of risk management and transparency in commercial banks, using the example of Navibank (later NCB) in Vietnam. The article analyzes the impacts of various macroeconomic factors on Navibank's stock price using econometric methods, finding that GDP growth, lending rate, and VNIndex have a significant positive effect on stock price, while CPI and exchange rate have a negative effect. The article also proposes risk management plans and recommends improving transparency and compliance with international accounting standards. The study has implications for commercial bank management and policy in developing countries.

This article [6] presents a comprehensive survey of fusion techniques used in stock market prediction from 2011-2020. The financial market is complex and influenced by numerous events, making prediction challenging. The article categorizes the fusion techniques into information, feature, and

model fusion. The major applications of stock market prediction include stock price and trend prediction, risk analysis, return forecasting, index prediction, and portfolio management. The article also provides an infographic overview of fusion in stock market prediction and discusses potential future directions.

Overall, the studies suggest that the COVID-19 pandemic has had a significant impact on stock markets around the world, and various factors, such as ownership structure, investor behavior, and dividend policy, have influenced stock returns in different ways.

3 DATA COLLECTION

3.1 COVID-19 Dataset description:

We gathered our COVID-19 dataset from the GitHub repository of the Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE), which provides daily updated data on the COVID-19 pandemic worldwide. The data includes confirmed cases, deaths, and recoveries at the country and regional level, as well as some demographic information like population size and density.

The JHU CSSE COVID-19 data is widely used by researchers, journalists, and the general public to track the progress of the pandemic and to inform public health policies. The data is available in various formats, including CSV, JSON, and SQL, and can be downloaded directly from the repository or accessed through APIs provided by third-party services like Worldometer and Covid19api.com.

The JHU CSSE COVID-19 data is considered one of the most comprehensive and reliable sources of COVID-19 data available, and is often used as a benchmark for other COVID-19 data sources. The data is regularly updated with the latest available information and is accompanied by detailed documentation and data dictionaries to ensure transparency and reproducibility of analyses.

3.2 SP 500 data collection strategy

We used Yahoo Finance to web scrape the SP 500 dataset, Yahoo Finance is a popular financial news and information website that provides a wide range of financial data, news, and insights for investors. One of the features that makes Yahoo Finance popular among investors is the ability to access a wide range of financial data for free, including historical stock prices, financial statements, and news articles.

To extract SP 500 data, we used a Python library called yfinance, which allows us to easily download historical price data for any ticker symbol listed on Yahoo Finance. First, we defined the ticker symbol for the SP 500 index as GSPC. Then, we specified the start and end dates for which we wanted to download the data. We passed these parameters to the yf.download() function, which downloads the historical price data from Yahoo Finance for the specified ticker symbol and date range. Finally, we saved the downloaded data to a CSV file using the function provided by pandas. This data can then be used for further analysis or modeling.

3.3 Sector wise individual stocks

Our project includes the analysis of six different sectors: Clothing, Consumer Discretionary, Consumer Staples, Health Care, Information Technology, and Transport.

To represent the Clothing sector, we have selected companies such as Levi Strauss Co. (LEVI), Adidas AG (ADS.DE), DICK'S Sporting Goods, Inc. (DKS), Ralph Lauren Corporation (RL), Nike, Inc. (NKE), Under Armour, Inc. (UA), L Brands, Inc. (LB), and The Gap, Inc. (GPS).

In the Consumer Discretionary sector, we have included Toyota Motor Corporation (TM), Ford Motor Company (F), Fuji Heavy Industries Ltd. (FUJHY), General Motors Company (GM), Bayerische Motoren Werke Aktiengesellschaft (BMWYY), Fiat Chrysler Automobiles N.V. (FCAU), Tesla, Inc. (TSLA), and Honda Motor Co., Ltd. (HMC).

The Consumer Staples sector features Tyson Foods (TSN), PepsiCo (PEP), Grubhub (GRUB), McDonald's (MCD), Coca-Cola (KO), Yum! Brands (YUM), JBS S.A. (JBSAY), General Mills (GIS), Kellogg's (K), and Nestle (NSRGY).

The Health Care sector comprises Walgreens Boots Alliance (WBA), Cardinal Health (CAH), CVS Health (CVS), AmerisourceBergen (ABC), Centene Corporation (CNC), Pfizer (PFE), BioNTech SE (BNTX), UnitedHealth Group (UNH), Cigna Corporation (CI), and Johnson Johnson (JNJ).

In the Information Technology sector, we have selected Snap Inc. (SNAP), Amazon.com, Inc. (AMZN), Intel Corporation (INTC), Microsoft Corporation (MSFT), Facebook Inc. (META), Twitter Inc. (TWTR), Alphabet Inc. (GOOG), Zoom Video Communications Inc. (ZM), and Apple Inc. (AAPL).

Lastly, the Transport sector comprises United Airlines Holdings (UAL), Lyft Inc. (LYFT), XPO Logistics Inc. (XPO), Uber Technologies Inc. (UBER), FedEx Corporation (FDX), Deutsche Lufthansa AG (LHA.DE), FirstGroup plc (FGP.L), American Airlines Group Inc. (AAL), Air France-KLM SA (AF.PA), and Delta Air Lines Inc. (DAL).

We wrote a web scraping script to gather sector wise individual stocks of different companies. We are using the yfinance package to download the stock data for several ticker symbols. The ticker symbols are defined in a list, and we specify the start and end dates for the data that we want to download. We then use a for loop to download the stock data for each ticker symbol in the list, and store it in a dictionary with the ticker symbol as the key. We specify the interval as weekly ('1wk'). Finally, we save the data for each ticker in a separate CSV file. We use the "tocsv" method to save the data as a CSV file, with the ticker symbol as the file name. This allows us to easily analyze the stock data for each ticker separately in subsequent analyses.

4 DATA CLEANING AND PREPROCESSING

4.1 Fixing dates

At first we clean and transform a CSV file containing data on COVID-19 cases worldwide using the pandas and numpy libraries. Specifically, we wrote a script that reads in a CSV file, fixes the year in the column names using a custom function, selects data for the US only, and transposes the data to create a new dataframe since the csv file that we got contains each individual date as a column. The script also adds a new column for dates and converts the date column to a datetime data type to maintain uniformity between the covid dataset and the SP 500 dataset.

Secondly, we extract the total deaths in the US from a dataset containing COVID-19 death statistics worldwide. The pandas library is used to read in the dataset, select only the rows for the US, drop the first four columns which contain information like Provenance/State, Country/Region, Latitude, and Longitude which is not important for us since we are concentrating only on US data and then we transpose the resulting dataframe. We set the column names, add a new column for dates, and drop the date column, leaving only the death count. Finally, we concatenate the resulting death count dataframe with the previously created dataframe containing the confirmed case counts for the US.

Finally, we extract the total recovered information in the US from a dataset containing COVID-19 recovered statistics worldwide, and we perform the same cleaning steps as previously done on the COVD-19 deaths dataset. We then combine the resulting dataframe to previously created dataframe by concatenating them along the columns. The resulting dataset contains the number of confirmed cases, deaths, and recovered cases in the US for each date in the dataset.

4.2 Creating daily new cases, deaths, and active cases:

Previously extracted dataframe consisting of the number of confirmed cases, deaths, and recovered cases in the US is daily accumulated and doesn't represent the number of daily recorded cases, daily new deaths, and daily active cases. So we wrote a code that creates new columns in the dataframe that correspond to daily new cases, daily deaths, and active cases in the US during the COVID-19 pandemic. First we convert the 'Date' column to a datetime format to ensure it can perform date-based calculations. Then, we calculate daily new cases and deaths by taking the difference between the current day and the previous day's values. We fill any missing values with zero and convert them to integers. Finally, we calculate the active cases by subtracting the number of deaths and recovered cases from the total confirmed cases. The updated dataframe is then printed using the tabulate library to create a formatted table to verify the final dataframe. The resulting dataframe contains columns for date, confirmed cases, deaths, recovered cases, daily new cases, daily deaths, and active cases.

4.3 Combining COVID-19 dataset and SP 500 dataset

We open the dataset containing historical SP 500 stock market data in a dataframe and another dataset containing COVID-19 cases, deaths, and recoveries in the US. We then extract data from the SP dataset between January 22, 2020, and March 9, 2023, set the date as the index, and merge it with the COVID-19 dataset based on the date column using an inner join because the SP 500 dataset doesn't contain entries on weekends. Finally, we save the combined dataset as a new CSV file called "covidstockscombined.csv". The resulting dataset contains information about COVID-19 cases, deaths,, recovered cases, daily new cases, daily deaths, and active cases in the US as well as the performance of the SP 500 stock market index on the same dates.

4.4 Preprocessing

Our preprocessing code reads in the covidstockscombined.csv dataset and drops the unnecessary columns such as Adj Close, Open, High, Low, and Volume columns using the drop() method. Then, the Date column is converted to a datetime format using the pd.todatetime() method. After that, the DataFrame is filtered to keep only the rows before the specified date 2020-12-14 using the dfcs[dfcs['Date'] < cutoffdate] code, this done because of covid-19 dataset does not contain zeroes in the Recovered cases column which implicitly affects the active cases column so, we removed these outliers before modeling the data. Finally we create a new DataFrame with only the data up to the cutoff date. Overall, our preprocessing code drops some unnecessary columns and filters the data to only include the data up to a certain date.

5 EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis, is a critical step in data science that involves exploring and analyzing the data to gain insights and better understand the patterns, trends, and relationships within the data. In the context of the question about the impact of COVID-19 on US stocks, EDA could involve analyzing various datasets related to the stock market, such as historical stock prices, trading volumes, and economic indicators, to identify the impact of the pandemic on the stock market. Steps performed by us as a part of EDA are:

- Descriptive statistics: We used summary statistics, such as mean, median, and standard deviation, to understand the distribution of stock prices, trading volumes, and other variables over time.
- Time-series analysis: we analyzed trends and patterns in the stock market over time, including the impact of the pandemic on stock prices and trading volumes.

- Data visualization: We created visual representations of the data, such as line charts, scatter plots, and heat maps, to identify patterns and relationships between variables.

By performing EDA on the relevant data, we did gain a better understanding of how the COVID-19 pandemic has impacted the US stock market, which can help inform investment strategies and policy decisions. Please find the code snippet (fig1) of EDA performed on both Covid dataset and SnP500 dataset.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np


# Load the dataset from CSV file
df = pd.read_csv("C:\\Users\\prave\\Desktop\\IDS\\ds_project_proposal\\covid_stocks_combined.csv")

# Check the data types and non-null values of each column
print(df.info())

# Check the descriptive statistics of numeric columns
print(df.describe())

# Check the first and last 5 rows of the dataset
print(df.head())
print(df.tail())

# Check for missing values
print(df.isnull().sum())

# Check for duplicated rows
print(df.duplicated().sum())

# Check the correlation matrix of numeric columns
#corr_matrix = df.corr()
#print(corr_matrix)

# Visualize the correlation matrix using a heatmap

df.describe(percentiles = [.20, .40, .60, .80], include = ['object', 'float', 'int'])

plt.plot(df['Date'], df['Close'])
plt.show()
```

Figure 1. Code snippet of EDA performed on both Covid dataset and SnP500 dataset

```
print(df_transposed.describe())
```

	Date	0	1	2	3	4	5	6	7	8	...	\
count	4	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	...	
unique	4	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	...	
top	Unnamed: 0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
freq	1	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	...	

	1136	1137	1138	1139	1140	\
count	4	4	4	4	4	
unique	4	4	4	4	4	
top	03-03-2023	03-04-2023	03-05-2023	03-06-2023	03-07-2023	
freq	1	1	1	1	1	

	1141	1142	1143	1144	1145
count	4	4	1	1	1
unique	4	4	1	1	1
top	03-08-2023	03-09-2023	Unnamed: 1144	Unnamed: 1145	Unnamed: 1146
freq	1	1	1	1	1

[4 rows x 1147 columns]

Figure 2. describe function on covid dataset

5.0.1 EDA for Covid dataset.

5.1 EDA for snp500 dataset

image is screenshot of snp500 dataset

```
import matplotlib.pyplot as plt
print(df_transposed.head())
```

	Date	0	1	2	3	4	5	6	\
0	Unnamed: 0	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	
1	Confirmed	1.0	1.0	2.0	2.0	5.0	5.0	5.0	
2	Deaths	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	Recovered	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

	7	8	...	1136	1137	1138	1139	\
0	1/29/20	1/30/20	...	03-03-2023	03-04-2023	03-05-2023	03-06-2023	
1	6.0	6.0	...	103648690.0	103650837.0	103646975.0	103655539.0	
2	0.0	0.0	...	1122165.0	1122172.0	1122134.0	1122181.0	
3	0.0	0.0	...	0.0	0.0	0.0	0.0	

	1140	1141	1142	1143	1144	\
0	03-07-2023	03-08-2023	03-09-2023	Unnamed: 1144	Unnamed: 1145	
1	103690910.0	103755771.0	103802702.0	NaN	NaN	
2	1122516.0	1123246.0	1123836.0	NaN	NaN	
3	0.0	0.0	0.0	NaN	NaN	

	1145
0	Unnamed: 1146
1	NaN
2	NaN
3	NaN

[4 rows x 1147 columns]

```
print(df_transposed.shape)
```

(4, 1147)

Figure 3. head of covid dataset

Output exceeds the size limit. Open the full output data in a text editor

	Province/State	Country/Region	Lat	Long	1-22-2020	\
0	NaN	Afghanistan	33.939110	67.709953	0	
1	NaN	Albania	41.153300	20.168300	0	
2	NaN	Algeria	28.033900	1.659600	0	
3	NaN	Andorra	42.506300	1.521800	0	
4	NaN	Angola	-11.202700	17.873900	0	
..	...	...	...	...	...	
284	NaN	West Bank and Gaza	31.952200	35.233200	0	
285	NaN	Winter Olympics 2022	39.904200	116.407400	0	
286	NaN	Yemen	15.552727	48.516388	0	
287	NaN	Zambia	-13.133897	27.849332	0	
288	NaN	Zimbabwe	-19.015438	29.154857	0	

	1-23-2020	1-24-2020	1-25-2020	1-26-2020	1-27-2020	...	2-28-2023	\
0	0	0	0	0	0	...	209322	
1	0	0	0	0	0	...	334391	
2	0	0	0	0	0	...	271441	
3	0	0	0	0	0	...	47866	
4	0	0	0	0	0	...	105255	
..	...	...	...	...	...	...	...	
284	0	0	0	0	0	...	703228	
285	0	0	0	0	0	...	535	
286	0	0	0	0	0	...	11945	
287	0	0	0	0	0	...	343012	
288	0	0	0	0	0	...	263921	

...

1141	2023-03-08	103755771
1142	2023-03-09	103802702

[1143 rows x 2 columns]

Figure 4. covid dataset dataframe

5.2 SP500 index V/S date in a time frame of six months from start of covid to till date

We plotted the graph on snp500 dataset by taking snp index on Y axis and dates on X axis for every six month interval to find how different covid waves have impact on SP index.

	Date	Open	High	Low	Close	Adj Close	Volume
0	2018-03-12	2790.540039	2796.979980	2779.260010	2783.020020	2783.020020	3216960000
1	2018-03-13	2792.310059	2801.899902	2758.679932	2765.310059	2765.310059	3324290000
2	2018-03-14	2774.060059	2777.110107	2744.379883	2749.479980	2749.479980	3394630000
3	2018-03-15	2754.270020	2763.030029	2741.469971	2747.330078	2747.330078	3543710000
4	2018-03-16	2750.570068	2761.850098	2749.969971	2752.010010	2752.010010	5429140000
...	...	...	...	...	...	...	...
1253	2023-03-03	3998.020020	4048.290039	3995.169922	4045.639893	4045.639893	4084730000
1254	2023-03-06	4055.149902	4078.489990	4044.610107	4048.419922	4048.419922	4000870000
1255	2023-03-07	4048.260010	4050.000000	3980.310059	3986.370117	3986.370117	3922500000
1256	2023-03-08	3987.550049	4000.409912	3969.760010	3992.010010	3992.010010	3535570000
1257	2023-03-09	3998.659912	4017.810059	3908.699951	3918.320068	3918.320068	4445260000

[1258 rows x 7 columns]

Figure 5. sp500 dataset

```
print(df.describe())
```

	Open	High	Low	Close	Adj Close	Volume
count	1258.000000	1258.000000	1258.000000	1258.000000	1258.000000	1.258000e+03
mean	3496.527727	3518.091246	3472.824220	3496.661343	3496.661343	4.239659e+09
std	662.158234	665.817851	658.617785	662.461152	662.461152	1.076383e+09
min	2290.709961	2300.729980	2191.860107	2237.399902	2237.399902	1.296530e+09
25%	2882.565002	2893.472412	2864.847473	2881.242432	2881.242432	3.536050e+09
50%	3370.420044	3389.319946	3358.065063	3374.395020	3374.395020	4.010000e+09
75%	4080.309998	4112.937622	4060.007507	4081.152527	4081.152527	4.683920e+09
max	4804.509766	4818.620117	4780.040039	4796.560059	4796.560059	9.976520e+09

```
print(df.isnull().sum())
```

```
Date         0
Open         0
High         0
Low          0
Close        0
Adj Close    0
```

Figure 6. describe and null values

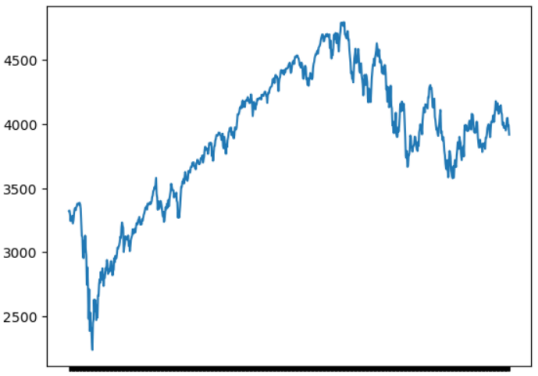

Figure 7. graph of sp500 for covid start to till date

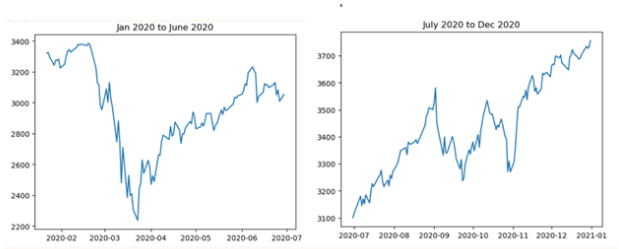

Figure 8. months time frame jan 2020 to Dec 2020

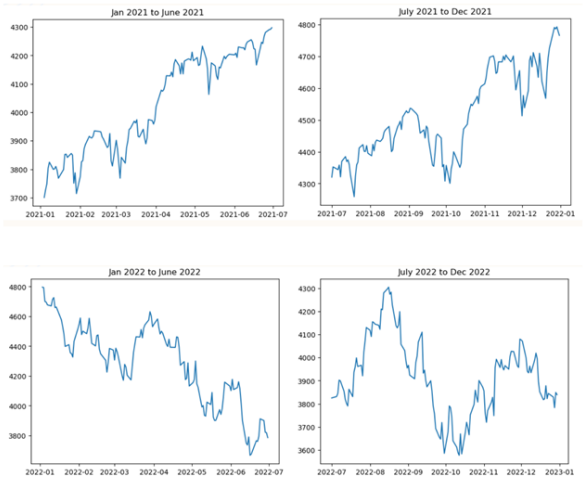

Figure 9. months time frame jan 2021 to Dec 2022

6 MODELING THE DATA

6.1 Linear Regression

First, we loaded our preprocessed dataset, which was a combination of COVID-19 cases data and SP 500 stock market data, using Pandas. Then, we split our data into independent variables, which were daily new cases and daily deaths, and our dependent variable, which was the stock market closing price. We used the traintestsplit function from Scikit-learn to split our data into training and testing datasets. We used 80 percent of the data for training and 20 percent for testing, with a random state of 42 to ensure reproducibility.

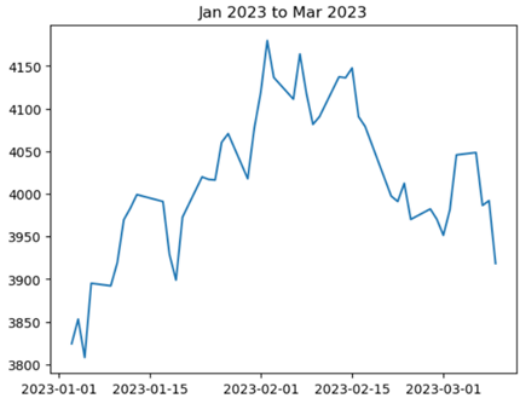

Figure 10. months time frame jan 2023 to mar 2023

Next, we created a Linear Regression model using Scikit-learn's LinearRegression class and fit it to our training data. We then used the model to make predictions on our testing data and calculated the accuracy of our model. We printed the model's accuracy, coefficients, and intercept to analyze the model's performance and understand how each independent variable impacted the stock market's closing price.

6.2 Logistic Regression

In the previous code, we first clipped the columns 'Daily New Cases' and 'Daily Deaths' to a minimum value of 0 using the clip() method to remove negative values. Next, we created a new binary column called 'Stock Increase', which indicates whether the stock prices increased or decreased on the next day relative to the current day. This column was created by comparing the 'Close' values of the current day and the next day using the shift() method, and then converting the resulting boolean array to an integer array with 1s indicating an increase and 0s indicating a decrease. We then filtered the data to keep only the rows before the cutoff date of December 14, 2020. We defined the features as the columns 'Daily New Cases', 'Daily Deaths', and 'Active Cases', and the target as the 'Stock Increase' column. We then split the data into training and test sets with a test size of 0.2 and a random state of 42.

Next, we initialized a logistic regression model and fit it to the training data using the fit() method. We then made predictions on the test data using the predict() method and calculated the accuracy score using the accuracyscore() method.

7 Data Visualization

7.1 Impact of COVID on individual companies in the SP 500 based on the sector:

our code(fig 11) reads CSV files from subdirectories of a parent folder, extracts the "Date" and "Close" columns, converts the "Date" column to datetime objects, and plots the "Date" vs "Close" values for each CSV file. The plot for each CSV file is displayed separately, and the name of each file is used as a label for the legend.

The code uses the following libraries:

- os: for interacting with the file system
- pandas: for reading and manipulating CSV files
- matplotlib: for plotting the data

The main steps of the code are:

- Get a list of all directories in the parent folder.
- Loop through each directory.
- For each directory, loop through each file in the directory.
- If the file is a CSV file, read it into a Pandas DataFrame.
- Extract the "Date" and "Close" columns from the DataFrame.
- Convert the "Date" column to datetime objects.
- Plot the "Date" vs "Close" values for the file.
- Set the plot title and labels, and show the legend and plot.

```
import os
import pandas as pd
import matplotlib.pyplot as plt


# Get a list of all directories in the parent folder
directories = [d for d in os.listdir("./") if os.path.isdir(os.path.join("./", d))]
parent_folder_path="./"
# Loop through all directories
for directory in directories:
    if(directory !=".ipynb_checkpoints"):
        folder_path = os.path.join(parent_folder_path, directory)  # Get folder path for current directory

        # Loop through all files in the current directory
        for file_name in os.listdir(folder_path):
            if file_name.endswith(".csv"):  # Assuming files are in CSV format
                file_path = os.path.join(folder_path, file_name)

                # Read CSV file into a Pandas DataFrame
                df = pd.read_csv(file_path)

                # Extract "Date" and "Close" columns
                date = df["Date"]
                close = df["Close"]
                # Convert "Date" column to datetime objects
                date = pd.to_datetime(date)

                # Plot "Date" vs "Close"
                plt.plot(date, close, label=file_name)  # Use file name as label for legend

                # Set plot title and labels

        plt.title("Time Frame vs Closing Values")
        plt.xlabel("Date")
        plt.ylabel("Close")
        plt.xticks( rotation=90)
        plt.suptitle(directory, y=1.05, fontsize=16, fontweight='bold')
        plt.legend()  # Show legend
        plt.show()  # Display the plot
```

Figure 11. Code snippet to generate the graphs

We took sp500 dataset and in that dataset I took some of the individual stocks and divide them into the sectors . the sectors are clothing, consumer discretionary, consumer staples, health care, information technology, transport. When it comes to clothing we consider ADS.DE, DKS,GPS, LB, LEVI, NKE, RL, UA.(fig 12) The clothing industry has seen drastic decline and some of the companies couldn't able to reach the previous top. Next when it comes to consumer Discretionary We took BMWYY,F,FCAU,FUJHY,GM,HMC,TM,TSLA company stock.(fig 13) The consumer discretionary means the companies handles the non essential goods for the people. Except Tesla all the non essential goods stock prices are declined drastically.

Next when it comes to consumer staples we tool GIS,GRUB,JBSAY,K,KO,M company stock.(fig 14) This is one of the sector where there was an intitial decline but it most of the stocks increased

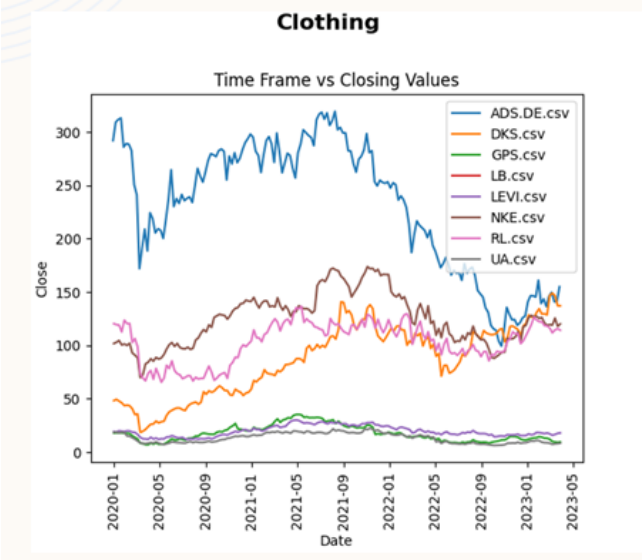

Figure 12. Clothing sector

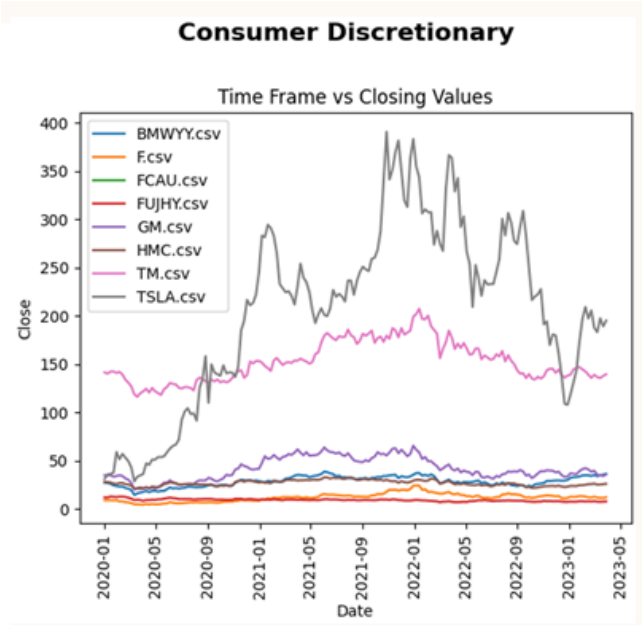

Figure 13. consumer Discretionary

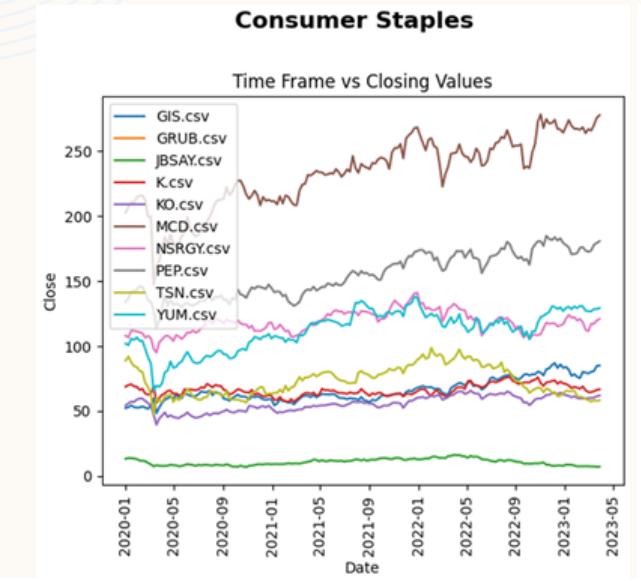

Figure 14. consumer staples

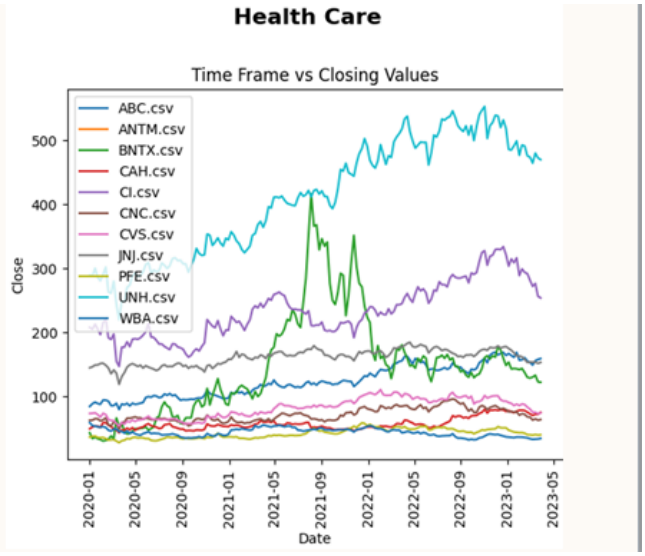

Figure 15. Health care

steadily. Consumer staples stocks are generally the companies handles the essential goods for the people.

Next when it comes to health care. This is one of the sectors where it saw high trajectories because of the medicines, etc., which are very important than any other goods. Here we considered ABC, ANTM, BNTM, CAH, CI, CNC, CVS, JNJ, PFE, UNH, WBA. (fig 15) as we can see all the stocks have good trajectories.

After healthcare another sector which saw a wonderful trajectories is information technology as most of the companies changed from offline to online. The IT sector saw a boom because of the increased user base. Here we considered AAPL,AMZN,FB, GOOG, INTC, META, MSFT, SNAP, TWTR, ZM.(fig 16) All the IT company stock values are increased of which ZM stock value increased so much. Transport sector is one sector where it saw huge declines because of the lockdowns and the people stopped moving around. The stock prices got reduced so much. we considered AAL,AF.PA,DAL,FDX,FGP.L,LHA.DE,LYFT,UAL,UBER, XPO. (fig 17) All the stocks got huge declines and some of the stocks couldnt even reach it previous heights.

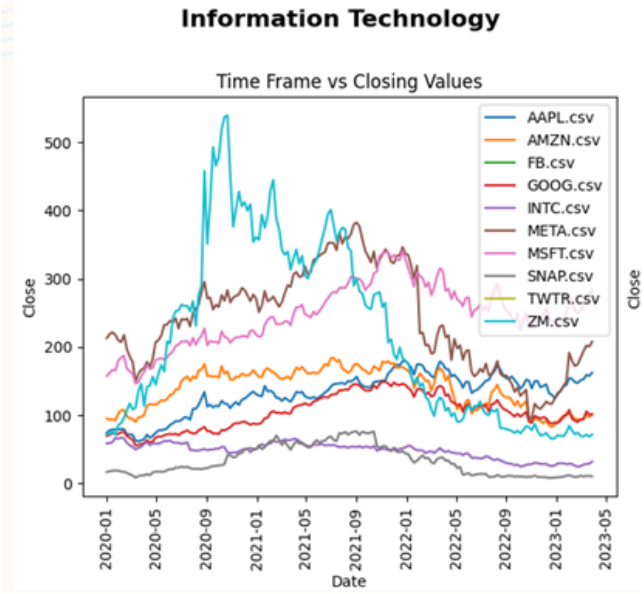

Figure 16. Information Technology

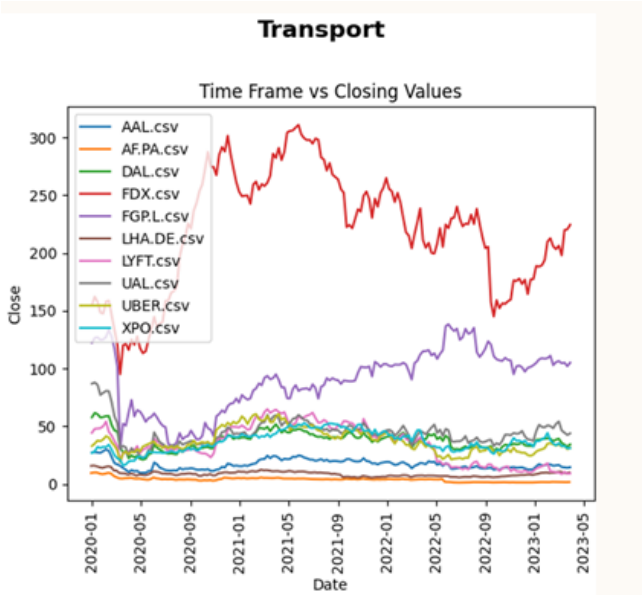

Figure 17. Transport

7.2 Impact of the SP 500 on COVID:

The code (fig 18) that reads in a CSV file containing data on the SP 500 closing price and COVID-19 cases and deaths, and then creates a plot of the SP 500 closing price and COVID-19 cases over time. The code extracts the relevant columns from the CSV file and converts the date column to datetime format . It then creates a plot with two y-axes, one for the SP 500 closing price and one for COVID-19 cases. The SP 500 closing price is plotted on the left y-axis, and COVID-19 cases are plotted on the right y-axis using ax.twinx() to

```python
import pandas as pd
import matplotlib.pyplot as plt

# Read in the data
df = pd.read_csv("covid_stocks_combined (1).csv")
# Extract relevant columns
date = pd.to_datetime(df['Date']) # Convert date to datetime format
sp500_close = df['Close']
covid_cases = df['Confirmed cases']
covid_deaths = df['Deaths']
# Create a plot
fig, ax = plt.subplots(figsize=(12, 6))
ax.plot(date, sp500_close, label='S&P 500 Closing Price', color='blue')
ax2 = ax.twinx()  # Create a secondary y-axis
ax2.plot(date, covid_cases, label='COVID-19 Cases', color='red')
ax.set_xlabel('Date')
ax.set_ylabel('S&P 500 Closing Price')
ax2.set_ylabel('COVID-19 Cases')
ax.legend(loc='upper left')
ax2.legend(loc='upper right')
plt.title('S&P 500 Closing Price vs. COVID-19 Cases')
plt.show()
```

Figure 18. Code snippet for Impact of the sp500 on covid

create a secondary y-axis. The plot (fig 19) includes axis labels and legends for both lines. Finally, the plot title is set using plt.title() and the plot is displayed using plt.show().

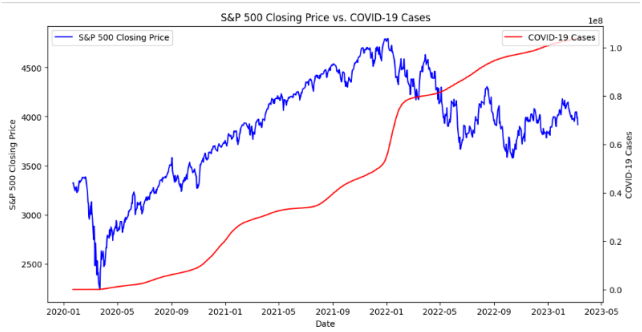

Figure 19. Impact of the SP 500 on COVID

7.3 Finding correlation

```python
import matplotlib.pyplot as plt
# Read in the data
df = pd.read_csv("covid_stocks_combined (1).csv")
# Extract relevant columns
sp500_close = df['Close']
covid_cases = df['Confirmed cases']
covid_deaths = df['Deaths']
# Calculate correlation
correlation_cases = np.corrcoef(sp500_close, covid_cases)[0, 1]
correlation_deaths = np.corrcoef(sp500_close, covid_deaths)[0, 1]

print("Correlation between S&P 500 Closing Prices and COVID-19 Cases: {:.2f}".format(correlation_cases))
print("Correlation between S&P 500 Closing Prices and COVID-19 Deaths: {:.2f}".format(correlation_deaths))
# Create scatter plots
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 6))
ax1.scatter(sp500_close, covid_cases, color='red')
ax1.set_xlabel('S&P 500 Closing Price')
ax1.set_ylabel('COVID-19 Cases')
ax1.set_title('S&P 500 Closing Price vs. COVID-19 Cases')

ax2.scatter(sp500_close, covid_deaths, color='blue')
ax2.set_xlabel('S&P 500 Closing Price')
ax2.set_ylabel('COVID-19 Deaths')
ax2.set_title('S&P 500 Closing Price vs. COVID-19 Deaths')

plt.show()
```

Figure 20. Code snippet for Finding correlation

Our code (fig 20) reads in a CSV file containing data on the SP 500 closing price and COVID-19 cases and deaths, and then calculates the correlation between the SP 500 closing price and COVID-19 cases and deaths using np.corrcoef(). The code then creates scatter plots of the SP 500 closing price vs. COVID-19 cases and deaths using plt.subplots() to create two subplots side-by-side. In the first subplot, the SP 500

closing price is plotted on the x-axis and COVID-19 cases are plotted on the y-axis using ax1.scatter(). In the second subplot, the SP 500 closing price is plotted on the x-axis and COVID-19 deaths are plotted on the y-axis using ax2.scatter(). Each subplot(fig 21) includes axis labels and a title. Finally, the plots are displayed using plt.show().

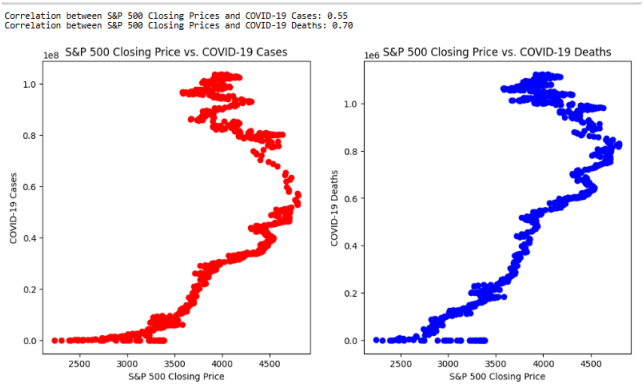

Figure 21. Impact of the SP 500 on COVID

7.4 Modelling and prediction

```
sp500_close = df['Close']
confirmed_cases = df['Confirmed cases']

# Perform linear regression
X = sp500_close.values.reshape(-1, 1) # Reshape the predictor to a 2D array
y = confirmed_cases.values.reshape(-1, 1) # Reshape the target variable to a 2D array
model = LinearRegression()
model.fit(X, y)
y_pred = model.predict(X)

# Plot scatter plot and linear regression line
plt.scatter(sp500_close, confirmed_cases, color='b', label='Data')
plt.plot(sp500_close, y_pred, color='r', label='Linear Regression')
plt.xlabel('S&P 500 Closing Prices')
plt.ylabel('COVID-19 Confirmed Cases')
plt.title('S&P 500 Closing Prices vs. COVID-19 Confirmed Cases')
plt.legend()
plt.show()
```

Figure 22. Code snippet for modelling and prediction

our code(fig 22) demonstrates a basic machine learning workflow for predicting the number of confirmed COVID-19 cases based on the SP 500 stock market closing prices using linear regression.

- Loading the dataset: The first step is to load the dataset containing the necessary variables for training and testing the linear regression model. The dataset is loaded using Pandas function.
- Data preprocessing: Next, any unnecessary columns from the dataset are dropped using the Pandas drop() function. Any missing values in the dataset are also removed using the Pandas dropna() function.
- Feature engineering: Feature engineering is the process of creating new features or selecting relevant features from the dataset. In this case, the only relevant feature is the SP 500 closing prices, which is selected as the predictor variable. The target variable is the number of confirmed COVID-19 cases.(fig 23)
- Model selection: The data is split into training and testing sets from the scikit-learn library. The training set is used to train the linear regression model using the LinearRegression() function. Once the model is trained, it is used to make predictions on the testing set.
- Model evaluation: The mean squared error (MSE)(fig 24) is used as the evaluation metric for the linear regression model. The MSE is calculated using the function from the scikit-learn library.

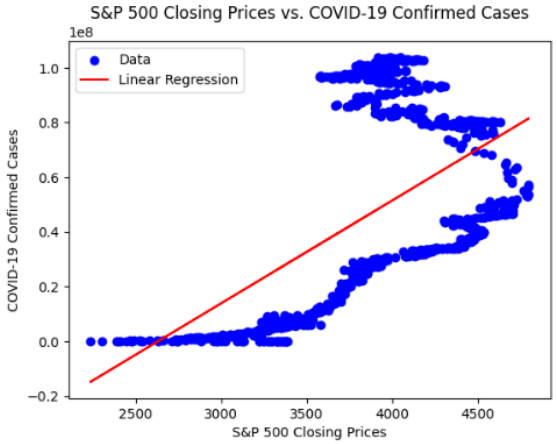

Figure 23. Model using linear regression

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error


# Load the dataset
df = pd.read_csv("covid_stocks_combined (1).csv")

# Data Preprocessing
# Assuming the dataset has columns 'Date', 'SP500_Close', 'Confirmed_cases', 'Deaths', and other relevant columns
# Drop any unnecessary columns
df = df[['Date', 'Close', 'Confirmed cases', 'Deaths']]
# Handle missing values
df = df.dropna()
# Normalize data if needed

# Feature Engineering
# Extract relevant features or create lagged variables if needed

# Model Selection
# Split data into training and testing sets
X = df[['Close']] # Use SP500_Close as predictor
y = df['Confirmed cases'] # Use Confirmed_cases as target variable
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
# Choose a machine learning algorithm, for example, Linear Regression
model = LinearRegression()
# Train the model
model.fit(X_train, y_train)
# Make predictions
y_pred = model.predict(X_test)

# Model Evaluation
mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error (MSE):", mse)
```

Figure 24. Code snippet for prediction accuracy

The mean squared error (MSE) is a commonly used metric to evaluate the performance of a regression model. It measures the average squared difference between the predicted values and the actual values. The MSE is always non-negative and a lower value indicates better performance. In this case, the calculated MSE value of (fig 25) 899164794402931.2 indicates that the model's predictions are not very accurate. Since the MSE is very high, it means that the model's predictions are

often far from the actual values. In other words, the model is not able to accurately capture the relationship between the SP 500 closing prices and the number of confirmed COVID-19 cases.

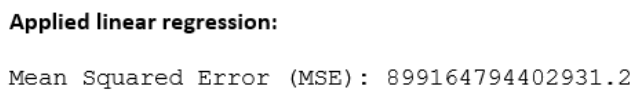

Figure 25. Model mean squared error

8 Results

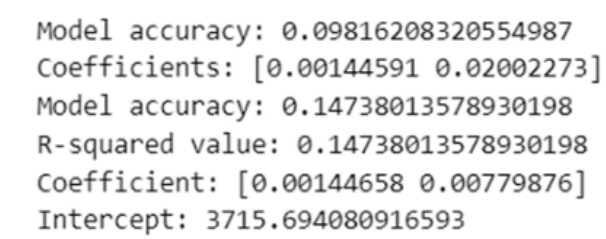

Figure 26. Results

The results of the Linear Regression model show that the model has a very low accuracy of 0.098(fig 26), indicating that the model is not able to predict the stock prices well based on the given features. The coefficients of the model indicate that the feature 'Daily New Cases' has a coefficient of 0.00144591 and the feature 'Daily Deaths' has a coefficient of 0.02002273. This means that the feature 'Daily Deaths' has a slightly higher impact on predicting the stock prices compared to 'Daily New Cases' but it may not be significant in predicting the closing price of the SP 500 stock index. The

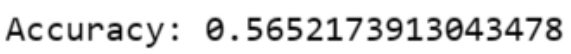

Figure 27. Model Accuracy

logistic regression model achieved an accuracy of 0.5652,(fig 27) which means that it correctly predicted whether the stock prices would increase or decrease in around 56.5 percent of the cases in the test set. While this accuracy score is not very high, it does show that there is some correlation between the daily new cases, daily deaths, and active cases of COVID-19 and the movement of the stock prices. It is important to note that this model is based on a limited set of features and a relatively short time frame, and it is possible that additional features and a longer time frame could improve the model's accuracy.

In contrast, the Logistic Regression model has an accuracy of 0.565, which is much higher than that of the Linear Regression model. The improvement in accuracy could be due to the fact that the COVID-19 data is correlated with the daily change in the stock price which is a binary outcome (stock prices increasing or decreasing). In this case, the model was trained to predict whether the stock prices would increase or decrease based on the given features, which may have been an accurate way for the model to learn compared to predicting the exact stock prices.

9 Future Work

- Analyzing the impact of government regulations on the stock market: further investigate how government regulations, such as lockdown measures, stimulus packages, and fiscal policies, have impacted the SP 500. This could involve analyzing how changes in regulations influenced market trends, investor sentiment, and stock performance.
- Exploring the global impact of the US stock market on other countries' stock markets: investigate how the SP 500's performance has influenced stock markets in other countries, particularly in different sectors. This could involve analyzing cross-country correlations, identifying patterns of spillover effects, and examining how global events and economic policies have influenced stock market interdependencies.
- Conducting sector-specific analysis: dive deeper into the impact of COVID-19 on specific sectors within the SP 500, such as healthcare, technology, finance, and consumer goods. This could involve assessing how different sectors have been affected differently by the pandemic, analyzing sector-specific trends, and identifying opportunities or risks for investors.
- Incorporating external factors: incorporating external factors, such as macroeconomic indicators, sentiment analysis of news and social media, or changes in consumer behavior, into your analysis. This could provide a more comprehensive understanding of the factors driving stock market performance during the pandemic.
- Exploring alternative modeling approaches: explore alternative modeling approaches, such as machine learning algorithms or time-series analysis techniques, to potentially improve the accuracy of your predictions. Experimenting with different modeling techniques could help identify the most suitable approach for your data and research question.

10 conclusion

we utilized various data science techniques such as data gathering, visualization, and analysis to examine the interplay between the SP 500 and COVID-19. The accuracy of

our model's predictions was around 0.5, indicating room for improvement. Through our analysis, we observed that COVID-19 had a significant impact on the SP 500 in the initial phases, but the impact reduced in later phases. We also attempted to analyze the reverse relationship of SP 500's impact on COVID-19 cases. However, the prediction error rate was high, indicating challenges in accurately predicting the relationship between the two variables. Overall, our analysis provides insights into the complex relationship between the SP 500 and COVID-19, indicating the need for further research to better understand their interplay [**?**].

References

[1] HaiYue Liu, Yile Wang, Dongmei He, and Cangyu Wang. 2020. Short term response of Chinese stock markets to the outbreak of COVID-19. *Applied Economics* 52, 53 (2020), 5859–5872.

[2] Zaky Machmuddah, St Dwiarso Utomo, Entot Suhartono, Shujahat Ali, and Wajahat Ali Ghulam. 2020. Stock market reaction to COVID-19: Evidence in customer goods sector with the implication for open innovation. *Journal of Open Innovation: Technology, Market, and Complexity* 6, 4 (2020), 99.

[3] Noura Metawa, M Kabir Hassan, Saad Metawa, and M Faisal Safa. 2019. Impact of behavioral factors on investors' financial decisions: case of the Egyptian stock market. *International Journal of Islamic and Middle Eastern Finance and Management* 12, 1 (2019), 30–55.

[4] Nguyen Thi Thanh Phuong, Dinh Tran Ngoc Huy, and Pham Van Tuan. 2020. The evaluation of impacts of a seven factor model on nvb stock price in commercial banking industry in vietnam-and roles of Discolosure of Accounting Policy In Risk Management. *International Journal of Entrepreneurship* 24 (2020), 1–13.

[5] Hidenori Takahashi and Kazuo Yamada. 2021. When the Japanese stock market meets COVID-19: Impact of ownership, China and US exposure, and ESG channels. *International Review of Financial Analysis* 74 (2021), 101670. https://doi.org/10.1016/j.irfa.2021.101670

[6] Ankit Thakkar and Kinjal Chaudhari. 2021. Fusion in stock market prediction: a decade survey on the necessity, recent developments, and potential future directions. *Information Fusion* 65 (2021), 95–107.

[7] Eko Wahjudi. 2020. Factors affecting dividend policy in manufacturing companies in Indonesia Stock Exchange. *Journal of Management Development* (2020).