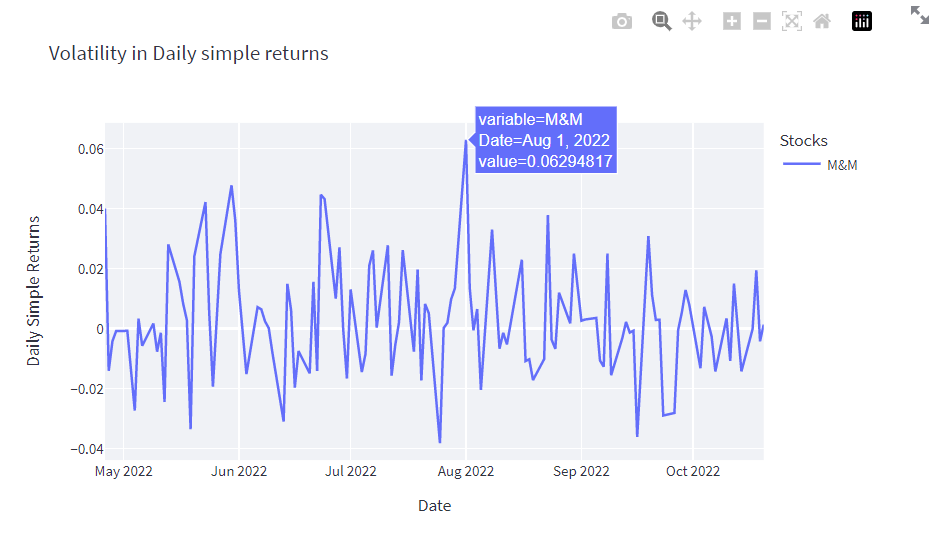
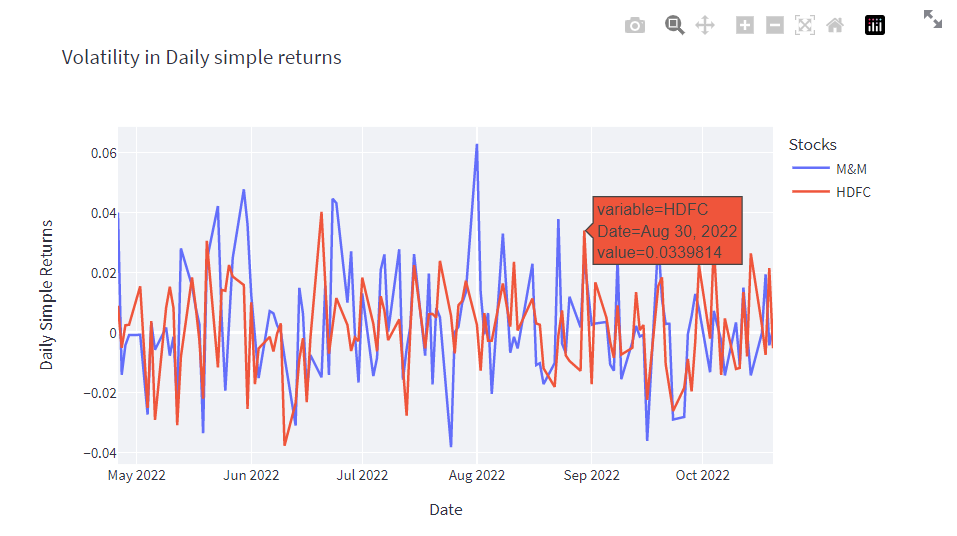
SYNOPSIS

1. Stockfolio is a multi-module application which aims to provide a variety of services for people who wish to gain knowledge to invest in the stock market. This application aims to provide various details like stock trends, sentiment analysis, portfolio optimization, stock movement visualization etc. Stockfolio is a modular application which aims to provide its users with various stock information on demand. Stockfolio uses an open-source API to track stock information and process them to create meaningful information for the end-user. By considering various angles to interpret stocks Stockfolio aims to create new insights for users to understand the movement of stocks in the market. Built with open-source software, stockfolio is stable and scalable due to modularity. All this packed together in a deployable application can be used on any system provided the necessary libraries are included.
2. There are various current applications which aim to provide similar services like stock tracking, portfolio optimisation, back testing, stock tracking indicators, brokerage deduction and many more. But, most of the features for a product currently available are limited either by having a paid version or by the nature of the application where the feature is absent. Most of the applications that provide this service are online web applications and they mostly require premium service. Also, the main drawback of most of the applications is that they are region specific and hence will not work outside their regions or will not work for foreign markets. This makes compatibility (compatibility in terms of availability of desirable service)a very big issue.
3. Stockfolio aims to overcome most of these drawbacks, mostly by being modular and open source. Since the application was designed to be modular, the datasource for the application can be any relevant form of data. Since the application computes the required indices ad hoc, we can design it to compute only necessary indices. Since the application is open source, incorporating newer modules for the project also shouldn’t be a problem. As the application is open-source, anyone can use it for free of cost to manage their personal portfolios.
4. Stockfolio is designed to be a locally hosted application mostly dependent on certain data files to lookup information online using browsers. Stockfolio is built using streamlit to provide a seamless and beautiful user interface which is both interactable and customisable. There is also the option of deploying the application in streamlit cloud which is a hosting service provided by streamlit to host their applications online. // once the application is hosted locally, the user can surf through the functions and interact with the  specifications provided in the application to make further decisions.
5. Stockfolio, which has been built so far, provides the end user the ability to track a set of stocks they select using a few tracking metrics, candlestick charts and daily returns charts. Stockfolio also comes with a module to optimize a given set of stocks to maximize their returns. Stockfolio also has a sentiment analyser module which looks up the internet to analyze the sentiment seen towards a particular stock in the market.
6. Stockfolio uses two main algorithms to provide its services. The first one being the portfolio optimiser, which determines the maximum sharpe ratio (sharpe ratio is a technical indicator used mainly to measure the amount of return subject to a certain amount of risk) that a portfolio can have with a corresponding level of risk. The second algorithm is the sentiment analyser which uses VADER (Valence Aware Dictionary and sEntiment Reasoner) to calculate the sentiment expressed by users in social media and other online sources. VADER uses rule based sentiment analysis to compute the sentiment score.
7. Stockfolio was built entirely with python. Stockfolio uses a number of libraries like pandas, numpy, pyportfolioopt, plotly, streamlit and much more to achieve its working efficiency. The libraries *pandas* and *numpy* help with data collection and data transformation while libraries like *pyportfolioopt, plotly, nltk, word cloud* etc are primarily used for performing necessary analysis on the collected data. All of this put into streamlit provides the necessary graphical user interface(GUI) required for the application to operate.**Pandas** is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language. **NumPy** is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The **Plotly** Python library is an interactive, open-source plotting library that supports over 40 unique chart types covering a wide range of statistical, financial, geographic, scientific, and 3-dimensional use-cases. The Natural Language Toolkit, or more commonly **NLTK**, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language. **Streamlit** is an open-source app framework for creating and deploying data science applications. **PyPortfolioOpt** is a library that implements portfolio optimization methods, including classical mean-variance optimization techniques and Black-Litterman allocation, as well as more recent developments in the field like shrinkage and Hierarchical Risk Parity. **Snscrape** is a scraper for social networking services (SNS). It scrapes things like user profiles, hashtags, or searches and returns the discovered items, e.g. the relevant posts. **NSEPy** is a Python library to get publicly available data on the current NSEIndia and NIFTY Indices site by communicating with their REST APIs.
8. The project yielded expected results based on the mathematics involved. The tools used in the project confirms that there is a definitive relationship between risk and return but not to the extent to say that all risky stocks pay-off.

Figure 5.1 A figure depicting a stock with high risk and relatively high return

Figure 5.2 is a similar figure as 5.1 but this show cases 2 stocks rather than 1

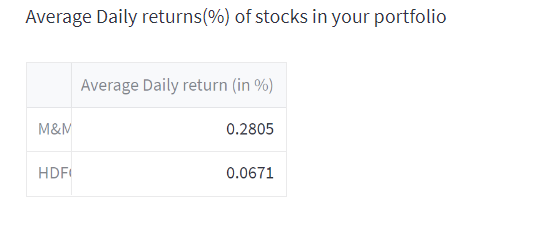
Figure 5.3 Depicts the average daily return seen on the selected stocks

Figure 5.4 Depicts the price change seen in the stock in the observed time for a rupee’s worth of investment

Average daily return of a stock is the average price increase seen in the stock in the selected period of time. It can be considered as the gain in the price of stock for a rupee’s investment.

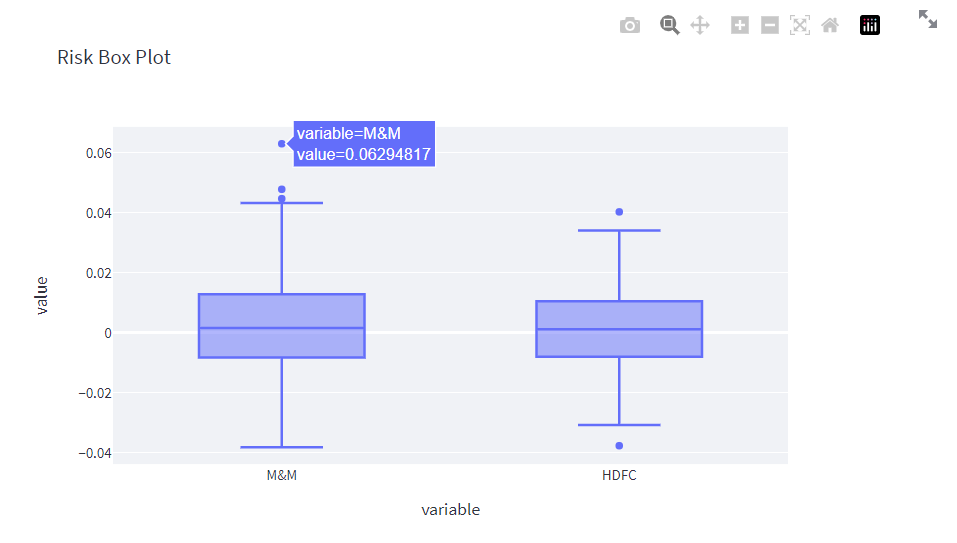


Figure 5.5 Shows the volatility of selected stocks in the form of a box plot

This figure shows-cases the maximum and minimum volatility seen in the stock visualised in the form of a box pot which shows the distribution on price volatility among the selected stocks.

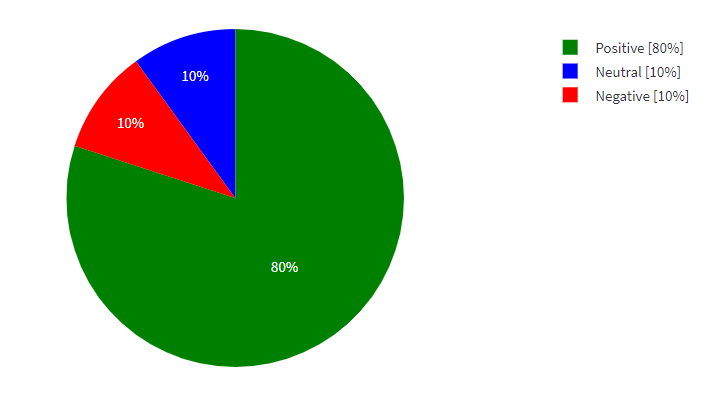
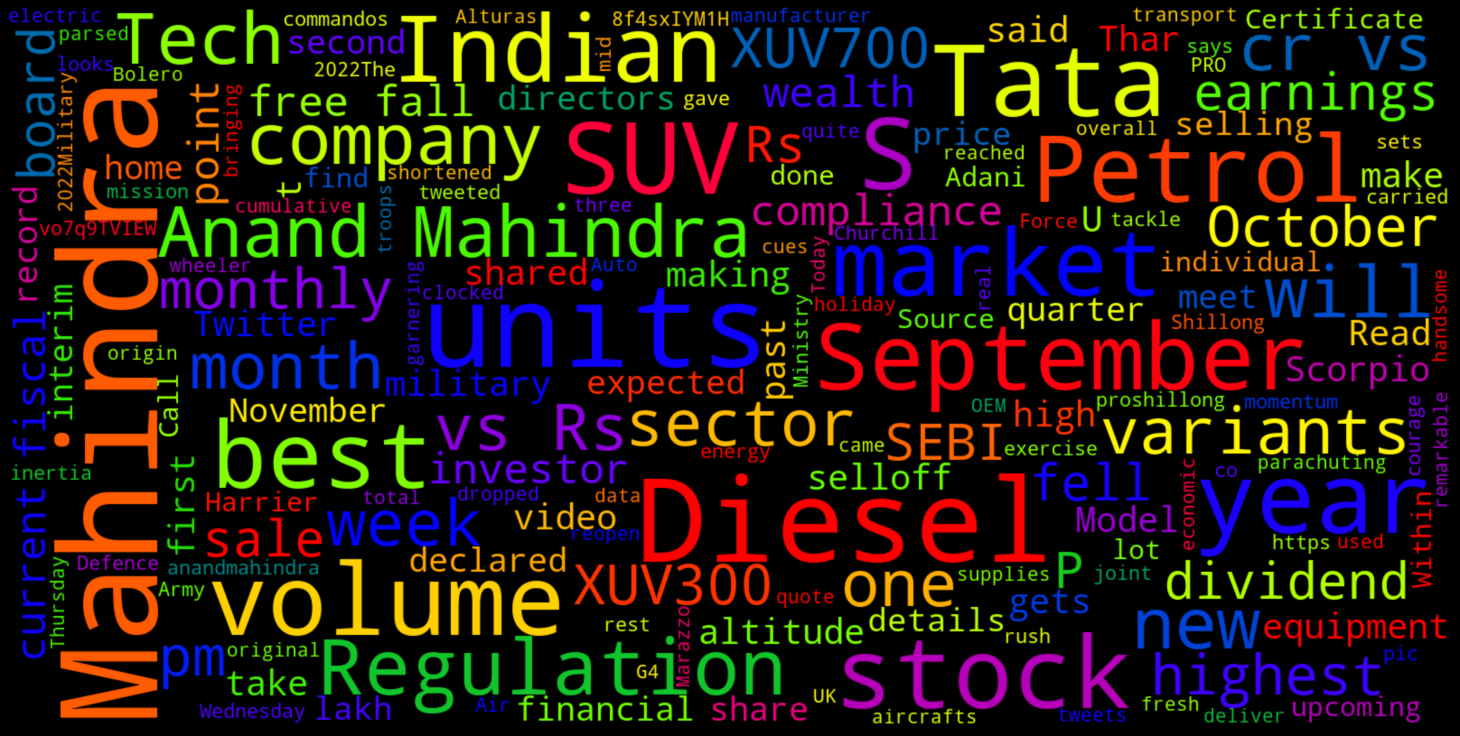
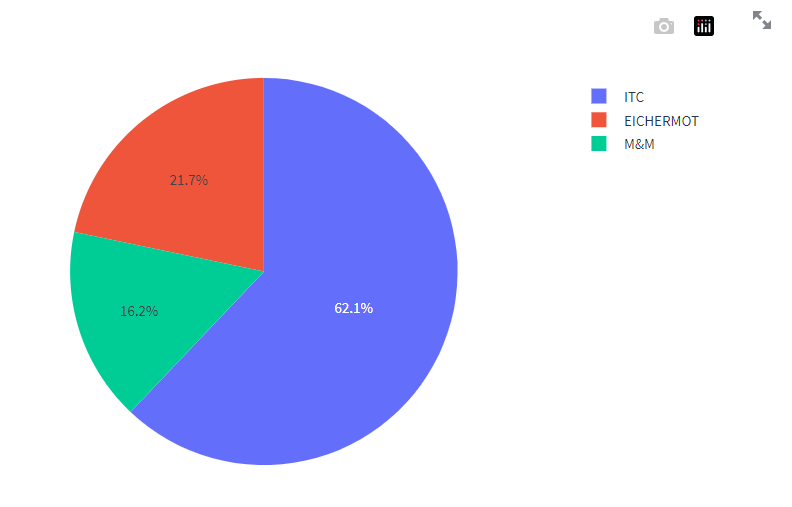


Figure 5.6 Show the sentiment score for a company

There were also some relationship between the sentiment of the user and stock prices and the sentiment of the users involved in the market. A positive trend among the users shows signs of a bullish trend and vice versa.

Figure 5.7 Depicts the word cloud generated for a particular company

Sentiment analysis aims to find the sentiment of users online, but in order to know more about the actuality of things, a word cloud visual comes in handy by displaying the words people mostly use to describe the things they search.

Figure 5.8 Shows how the portfolio allocation is done for a trio of stocks selected

Portfolio Optimisation done using the pyportfolioopt module yields a result very much mathematically. The module prioritises a stocks return over the nature of choice and it mainly used the Sharpe ratio to calculate the portfolio allocation.

1. Future changes / updates for this project includes creating more / different visualizations of collected data. Creating more modularity to enable a less cluttered user interface, more independent modules to facilitate more features. Option to save a portfolio set.