PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

100-ft Ring Road, Bengaluru – 560 085, Karnataka, India

A Project Report On

“Stock Forecasting and Markowitz’s Portfolio Optimization”

Submitted in fulfilment of the requirements for the Project Phase-II

Submitted by

Rajath Hegde
SRN: PES2PGE20DS320

Under the guidance of

Vivekananda
Dean of Research, Professor

February- June 2022

FACULTY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

PROGRAM M. TECH

FACULTY OF ENGINEERING DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
PROGRAM M. TECH

CERTIFICATE

This is to certify that the Dissertation entitled

"Stock Forecasting and Markowitz's Portfolio Optimization"

is a bonafide work carried out by

Rajath Hegde
SRN:PES2PGE20DS320

In partial fulfillment for the completion of 3rd semester course work in the Program of Study MTECH in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period Feb. 2022 – June. 2022. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the 3rd semester academic requirements in respect of project work.

Signature with date & Seal	*Signature with date & Seal*	*Signature with date & Seal*
Internal Guide	*Chairperson*	*Dean of Faculty*

Name and Signatures of the Examiners

1.

2.

3.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

ABSTRACT

Successfully predicting the stock market returns is a herculean task due to the volatility, non-linear nature and the dynamics of global scenarios in the financial markets. The success of a particular portfolio depends primarily on the future performance of the stock market. The advancements in the field of technology and especially in machine learning has significantly helped investors in finding and identifying opportunities to incorporate predictions in their analysis and build a portfolio which can deliver positive returns on a consistent basis.

This project has been divided it into two parts. In the first part, I have attempted to perform the stock price prediction using Facebook Prophet, ARIMA and L.S.T.M and the optimum model would be taken into consideration for designing a Diversified Portfolio consisting of 10 stocks in total using RMSE and MAPE. Here 8 stocks from various sectors (Cement, IT etc) and 2 Exchange Traded Mutual Fund (Niftybees, Bankbees) is being taken into consideration.In the second half of the project I will perform Markowitz's Optimization technique i.e Efficient Frontier Model and other techniques in order to obtain a portfolio with the highest Sharpe Ratio.

ACKNOWLEDGEMENTS

I would like to express my profound appreciation to every single person who supported me through my to auspicious journey at Great Learning and PES University. I would like thank my Guide Vivekananda and Yogesh Agarwal for their valuable inputs and guidance during my project. Foremost, I would like to express my sincerest gratitude to my beloved parents for being the best support system I have had since day one as well as for the sacrifices they made to see me the person I am today.

Finally, my warmest gratitude expands to everyone who contributed either directly or indirectly to the implementation of this project, namely my friends who have always pushed me to give my best and believe in myself.

1 INTRODUCTION

1.1 Motivations and Report Structure

Successfully predicting the stock market price or returns for that matter is a herculean task due to the volatile, unpredictable nature of the market. The success of a particular portfolio depends particularly on the future performance of the stocks. The advancement in the field of technology has significantly helped investors in identifying opportunities using various prediction methods and derive positive returns to the portfolio.

We can define Machine Learning as "Programming the computers to optimize a performance criterion using training data or past experience".[2] It can be used to optimize the construction of an investment portfolio, which is defined as an ensemble of investments in different assets aiming at earning certain returns in the future. Investment strategies have made a considerable progress, especially since the Modern Portfolio Theory, pioneered by Harry Markowitz in his paper "Portfolio Selection" (1952) [3]. Putting it in simpler terms, this theory addresses mathematically the process of selecting the investment instruments and assigning to each a part of the initial wealth. Quantitative investment strategies have also the advantages of not being impacted by the human emotions and bias given different market situations.

Investors and traders mainly use two approaches in predicting the trend of the stock.The first one is the technical analysis method which uses historical price of the stocks like closing and opening price, volume traded, adjacent close values for speculating the future stock price.The other approach is performed on the basis of external factors like company profile,market situation, political and economic factors, textual information in the form of news, articles, social media information etc. The stock market data is usually huge and non-linear in nature. To deal with this huge amounts of data machine learning techniques are used which have proven to improve efficiency by 60—80 percent as compared to earlier methods.[5]

This paper proposes an algorithmic method for designing a good portfolio by considering 10 stocks from different sectors of the National Stock Exchange(NSE) of India. Later on, using different machine learning models, stock prices are predicted. From there using Efficient Frontier Model, a portfolio is build thereby helping us to obtain an optimum portfolio with the highest sharpe ratio.

Motivation and also about the report structure is given in Section 1.1. Forecasting of the stock market is done in Section 1.2. Literature review is done in Section 1.3.

1.2 Forecasting of the stock.

Our attempt is to predict stock direction raises the question of the possibility of beating the market: this refers to the ”Efficient Market Theory” . This theory is summarized by Eugene F. Fama [13] by ”the statement that security prices fully reflect all available information.” Assuming this, fundamental and historical analysis shouldn't enable investors to predict the future behaviour and obtain higher rate of returns.

Stocks always trade at their fair value on exchanges, according to the EMH, making it impossible for investors to buy cheap stocks or sell at inflated prices. As a result, skilled stock selection or market timing should be impossible to outperform the entire market, and the only method for an investor to earn larger returns is to buy riskier stocks.

1.3 Literature Review.

'Stock Closing Price Prediction using Machine Learning Techniques'[6]. In this paper predicting of the closing price for the following day using Artificial Neural Network for 5 NYSE stocks were considered. A comparative analysis using Artificial Neural Network based on Root Mean Square Error, Mean Absolute Percentage Error and MBE (Mean Bias Error) in comparison to Random Forest.

'A Machine Learning Approach to Stock Price Prediction and Portfolio Optimization'[7].Initially chose S&P500 stocks(500) and finalized with 382 stocks after feature engineering. Worked primarily on 3 predictive models: 1) Moving Average Prediction.2) Multiple Linear Regression. 3) LSTM.

Drawbacks for this paper was Tangency portfolio and Minimum Variance portfolio are quadratic programming problems and solving them with the number of variables we had was computationally unfeasible.

'Portfolio Generation for Indian Stock Markets using Unsupervised Machine Learning'[8]. Have proposed a simple k-means based clustering strategy for an optimal portfolio. BSE100 stocks are represented by their fundamental financial ratios.

At the end of the financial year in March 2018, Sensex offered a return of 11.3%, BSE 100 offered a rate of return of 10.62% while the above portfolio gave a return of 27.51%.

"Portfolio Optimization on NIFTY Thematic Sector Stocks Using an LSTM Model"[9].This paper presents an algorithmic approach for designing optimum risk and eigen portfolios (PCA) for five thematic sectors of the NSE of India.The chosen sectors are as follows; (i) NIFTY services, (ii) NIFTY public sector enterprises (PSE), (iii) NIFTY multinational corporation (MNC), and (iv) NIFTY manufacturing, and (v) NIFTY commodities.Optimum risk and eigen portfolios for each sector are designed based on ten critical stocks from the sector.

"Hierarchical Risk Parity and Minimum Variance Portfolio Design on NIFTY 50 Stocks"[10].This paper proposes a systematic approach to designing portfolios using two algorithms, the critical line algorithm, and the hierarchical risk parity algorithm on eight sectors of the Indian stock market.Eight important sectors of NSE are first chosen. The selected sectors are: (i) auto, (ii) banking, (iii) FMCG, (iv)information technology (IT), (v) metal, (vi) pharmaceuticals, (vii) realty, and (viii) NIFTY 50. NIFTY 50 contains the 50 most critical stock from several sectors of the Indian stock market.

2. History of Indian Stock Market.

The Indian stock advertise follows its history back to the late 18th century when the exchanging floor was beneath the shade of a sprawling banyan tree inverse the Town Corridor in Mumbai. A number of individuals would meet beneath this tree to casually exchange in cotton. This was basically due to the reality that Mumbai was a active exchanging harbour and basic commodities were exchanged here often. The Companies Act was presented in 1850 taking after which financial specialists begun appearing an intrigued in corporate securities. The concept of restricted obligation moreover put in an appearance around this time. By 1875, an organization known as 'The Native Share and Stock Broker's Association' came into being. This was the predecessor of the BSE . In 1894, the Ahmedabad Stock Trade came into being primarily with the objective of empowering dealing within the offers of textile mills within the city.

The Calcutta Stock Trade was shaped in 1908 with the deliberate of encouraging a showcase for offers of ranches and jute mills. It was in 1920 that the Madras Stock Trade took shape. In 1957, the BSE was the primary stock trade to be recognized by the Government of India beneath the Securities Contracts Control Act. The SENSEX was launched in 1986 taken after by the BSE National Index in 1989. The Securities and Trade Board of India (SEBI) was constituted in 1988 to screen and direct the securities industry and stock trades. In 1992 it got to be an independent body with totally free powers. In 1992, the NSE was shaped as the primary demutualised electronic trade within the country with the purposeful of guaranteeing straightforwardness within the markets. NSE started operations within the Wholesale Debt Market (WDM) section in 1994, the values portion in 1994, and the subsidiaries portion in 2000. It was in 1995 that the BSE made the switch to an electronic framework of exchanging from the open-floor system. In 2015, SEBI was consolidated with the Forward Markets Commission (FMC) with the point of reinforcing direction of the commodities market, facilitating domestic and foreign institutional participation, and launch of new products.

Nowadays, the BSE is measured as the world's 11th biggest stock trade and the advertise capitalization is likely to be around \$1.7 trillion. The showcase capitalization of the NSE is evaluated to be over \$1.65 trillion. Over 5,000 companies are listed on the BSE and 1,500 figure on the NSE. In terms of share exchanging volumes, still, both the trades are on equality. These days individuals are able to conduct online exchanging sitting within the comfort of their home. Offices such as zero brokerage demat and live updates are all accessible with the assistance of internet.

3. Prediction of the Stock Direction.

Below I have shown the various stocks that I have used for the purpose of forecasting. Basically I have used three methods for the price prediction. Namely:

1) Autoregressive integrated moving average.
2) Long Short-Term Memory.
3) Facebook Prophet.

COMPANIES	SECTORS
Reliance Industries Ltd.	Diversified business.
Cipla.	Pharmaceutical.
Maruti Suzuki.	Automobile.
Hindustan Unilever.	Fast-moving consumer goods.
ICICI Bank.	Financial services.
Nestlé India.	Fast-moving consumer goods.
Tata Consultancy Services.	Information technology.
Ramco Cements.	Cement.
Nifty BeES.	E.T.F.
Bank BeES.	E.T.F.

Table 3.1 List of Stocks used in the portfolio.

These are the 10 diversified stocks from various sectors that have been taken into consideration. This includes 8 equity stocks and 2 exchange traded funds which is a Benchmark exchange traded scheme.

3.1 Note on Nifty BeES and Bank BeES :

Nifty BeES(Benchmark Trade Exchanged Scheme)the first trade exchanged fund (ETF) in India—seeks to supply investment returns that closely compare to the whole returns of securities as represented by the S&P CNX Nifty File. Clever BeES, a combination of a share and a mutual fund unit, trades on the capital market segment of NSE (National Stock Trade). Each Clever BeES unit is 1/10th of the S&P CNX Nifty File value. Nifty BeES units are traded and settled in dematerialised frame like all other share within the rolling settlement. In this way, it permits you to exchange real-time on NSE and gives you real-time indicative NAV (net resource esteem) NiftyBeES offers the benefits of diversification, index following and low expenses. Nifty BeES can be bought / sold like a share through any NSE terminal at costs accessible on the screen. The fundamental portfolio of Nifty BeES exceptionally closely duplicates that of the S&P CNX Nifty. Consequently, Nifty BeES tracks the development of S&P CNX Nifty.

Bank BeEs is an trade exchanged fund which contributes in as it were banking stocks. It was propelled by Nippon India Mutual Fund (formerly Reliance Nippon Shared Support) on 27th May 2004. The complete frame of Bank BeEs is Nippon India ETF Bank BeEs. It is benchmarked to the Nifty Bank Index which contains 12 of the biggest most fluid banking stocks recorded on the National Stock Trade (NSE). The aim of Bank BeEs is to reflect the returns of the country's best ten banks at 1/100th the fetched. For example: The cost of 1 unit of Bank Nifty is Rs 34,800 as on 9th June 2021. On the same day, the NAV of Bank BeEs is Rs 351.40 (generally 1/100th of Rs 34,800). 95% of Bank BeEs corpus is contributed as per Nifty Bank Index. The remaining 5% is contributed in cash showcase disobedient like Treasury Bills, Certificates of Deposits.

3.2 Dataset

Below are the first five rows and the last 5 rows of the various stocks in the portfolio.
It consists of Date, Open, High, Low, Close, Adj Close and Volume as the variables and 1515 observations.

3.2.1 RELIANCE

	Date	Open	High	Low	Close	Adj Close	Volume
0	01-01-2016	500.158997	504.666260	499.366516	502.907928	484.996368	2499742
1	04-01-2016	497.781525	502.140198	488.717438	492.977081	475.419189	13923887
2	05-01-2016	495.453583	500.258057	493.819092	497.855835	480.124176	6897687
3	06-01-2016	499.069336	514.324707	495.503113	511.253815	493.044983	12349673
4	07-01-2016	505.731171	509.173553	499.292206	501.867798	483.993286	9109980
...	...	...	...	...	...	...	...
1510	11-02-2022	2373.250000	2384.500000	2344.100098	2376.399902	2376.399902	7357863
1511	14-02-2022	2340.250000	2354.949951	2313.000000	2338.550049	2338.550049	4947109
1512	15-02-2022	2351.199951	2423.899902	2335.699951	2417.949951	2417.949951	5174646
1513	16-02-2022	2430.000000	2433.000000	2395.949951	2412.949951	2412.949951	4796294
1514	17-02-2022	2420.100098	2454.899902	2410.000000	2443.500000	2443.500000	5931774

1515 rows × 7 columns

3.2.2 CIPLA

	Date	Open	High	Low	Close	Adj Close	Volume
0	2016-01-01	653.200012	658.450012	648.500000	655.349976	638.705994	556012
1	2016-01-04	652.049988	653.799988	639.400024	645.650024	629.252380	631534
2	2016-01-05	646.200012	649.950012	638.049988	640.849976	624.574280	642269
3	2016-01-06	639.700012	655.950012	638.700012	652.099976	635.538513	1234096
4	2016-01-07	648.000000	650.000000	629.000000	637.450012	621.260620	732335
...	...	...	...	...	...	...	...
1510	2022-02-11	956.500000	976.049988	953.000000	958.500000	958.500000	1292417
1511	2022-02-14	947.000000	962.900024	938.849976	954.900024	954.900024	2346622
1512	2022-02-15	916.549988	938.799988	912.299988	921.849976	921.849976	8304774
1513	2022-02-16	938.000000	940.250000	920.099976	933.099976	933.099976	2480569
1514	2022-02-17	934.799988	955.500000	925.700012	928.200012	928.200012	2843010

1515 rows × 7 columns

3.2.3 MARUTI

	Date	Open	High	Low	Close	Adj Close	Volume
0	01-01-2016	4621.000000	4668.000000	4602.350098	4638.500000	4352.271973	243597
1	04-01-2016	4635.000000	4657.000000	4571.750000	4580.649902	4297.991699	405501
2	05-01-2016	4599.950195	4600.750000	4557.250000	4566.950195	4285.137207	509285
3	06-01-2016	4592.000000	4593.000000	4468.000000	4480.799805	4204.302734	593742
4	07-01-2016	4449.700195	4449.700195	4251.850098	4267.899902	4004.540283	1200069
...	...	...	...	...	...	...	...
1510	11-02-2022	8738.000000	8865.000000	8673.299805	8737.150391	8737.150391	671572
1511	14-02-2022	8500.000000	8615.049805	8325.000000	8366.400391	8366.400391	995540
1512	15-02-2022	8410.000000	8664.250000	8356.500000	8622.700195	8622.700195	682766
1513	16-02-2022	8720.000000	8736.400391	8544.000000	8582.950195	8582.950195	422704
1514	17-02-2022	8575.000000	8658.299805	8525.000000	8552.450195	8552.450195	305313

1515 rows × 7 columns

3.2.4 HINDUSTAN UNILEVER

	Date	Open	High	Low	Close	Adj Close	Volume
0	2016-01-01	860.000000	862.000000	853.500000	856.549988	781.844116	230366
1	2016-01-04	856.549988	868.700012	851.150024	859.000000	784.080383	828876
2	2016-01-05	857.299988	859.049988	845.500000	847.950012	773.994080	1118709
3	2016-01-06	850.000000	850.000000	840.950012	843.049988	769.521606	2135092
4	2016-01-07	840.000000	840.950012	817.500000	820.250000	748.710022	1353969
...	...	...	...	...	...	...	...
1510	2022-02-11	2279.000000	2279.000000	2249.000000	2258.000000	2258.000000	2100036
1511	2022-02-14	2220.000000	2236.250000	2207.000000	2228.550049	2228.550049	1661501
1512	2022-02-15	2247.949951	2304.149902	2222.550049	2290.100098	2290.100098	1533070
1513	2022-02-16	2290.100098	2303.750000	2275.000000	2286.500000	2286.500000	995668
1514	2022-02-17	2282.000000	2313.000000	2276.000000	2307.550049	2307.550049	869374

1515 rows × 7 columns

3.2.5 ICICI BANK

	Date	Open	High	Low	Close	Adj Close	Volume
0	2016-01-01	237.545456	239.636368	234.545456	239.090912	227.072784	5998096
1	2016-01-04	237.272720	237.590912	231.500000	232.318176	220.640518	9435792
2	2016-01-05	232.954544	234.090912	228.818176	233.363632	221.633408	8966977
3	2016-01-06	232.181824	233.409088	226.545456	227.363632	215.934998	17416181
4	2016-01-07	224.000000	225.181824	221.000000	224.318176	213.042603	18240712
...	...	...	...	...	...	...	...
1510	2022-02-11	796.849976	797.400024	787.900024	790.799988	790.799988	13138408
1511	2022-02-14	772.000000	773.950012	751.150024	753.700012	753.700012	16170656
1512	2022-02-15	752.950012	778.000000	743.200012	776.049988	776.049988	18880659
1513	2022-02-16	775.700012	777.549988	760.400024	764.049988	764.049988	10549784
1514	2022-02-17	766.500000	767.250000	746.299988	750.349976	750.349976	16683808

1515 rows × 7 columns

3.2.6 NESTLE INDUSTRIES

	Date	Open	High	Low	Close	Adj Close	Volume
0	2016-01-01	237.545456	239.636368	234.545456	239.090912	227.072784	5998096
1	2016-01-04	237.272720	237.590912	231.500000	232.318176	220.640518	9435792
2	2016-01-05	232.954544	234.090912	228.818176	233.363632	221.633408	8966977
3	2016-01-06	232.181824	233.409088	226.545456	227.363632	215.934998	17416181
4	2016-01-07	224.000000	225.181824	221.000000	224.318176	213.042603	18240712
...	...	...	...	...	...	...	...
1510	2022-02-11	796.849976	797.400024	787.900024	790.799988	790.799988	13138408
1511	2022-02-14	772.000000	773.950012	751.150024	753.700012	753.700012	16170656
1512	2022-02-15	752.950012	778.000000	743.200012	776.049988	776.049988	18880659
1513	2022-02-16	775.700012	777.549988	760.400024	764.049988	764.049988	10549784
1514	2022-02-17	766.500000	767.250000	746.299988	750.349976	750.349976	16683808

1515 rows × 7 columns

3.2.7 TATA CONSULTANCY SERVICES

	Date	Open	High	Low	Close	Adj Close	Volume
0	2016-01-01	1219.500000	1219.500000	1206.125000	1208.199951	1075.699341	712262
1	2016-01-04	1205.074951	1207.000000	1183.025024	1184.800049	1054.865723	1870184
2	2016-01-05	1192.500000	1193.300049	1170.500000	1174.474976	1045.672974	2678020
3	2016-01-06	1175.099976	1193.074951	1175.099976	1190.800049	1060.207886	2653228
4	2016-01-07	1185.000000	1191.449951	1180.000000	1185.625000	1055.600464	3199580
...	...	...	...	...	...	...	...
1510	2022-02-11	3752.500000	3752.500000	3690.000000	3694.949951	3694.949951	3851488
1511	2022-02-14	3724.000000	3793.250000	3710.000000	3733.750000	3733.750000	5951745
1512	2022-02-15	3786.000000	3835.000000	3748.000000	3817.800049	3817.800049	3931683
1513	2022-02-16	3844.000000	3854.100098	3806.000000	3813.100098	3813.100098	3256906
1514	2022-02-17	3825.000000	3835.000000	3779.000000	3784.199951	3784.199951	3134372

1515 rows × 7 columns

3.2.8 RAMCO CEMENT

	Date	Open	High	Low	Close	Adj Close	Volume
0	2016-01-01	391.649994	402.000000	382.350006	389.799988	379.417603	60581
1	2016-01-04	393.000000	393.600006	376.100006	389.200012	378.833557	79920
2	2016-01-05	383.250000	400.000000	383.250000	398.350006	387.739868	118475
3	2016-01-06	399.950012	404.000000	388.350006	391.000000	380.585602	525730
4	2016-01-07	389.000000	392.100006	386.000000	390.000000	379.612213	410722
...	...	...	...	...	...	...	...
1510	2022-02-11	885.000000	886.900024	868.250000	872.900024	872.900024	107865
1511	2022-02-14	855.000000	862.349976	833.049988	836.500000	836.500000	155690
1512	2022-02-15	840.000000	888.000000	827.000000	881.250000	881.250000	393474
1513	2022-02-16	881.000000	882.299988	855.099976	861.349976	861.349976	662795
1514	2022-02-17	865.900024	868.000000	851.400024	859.650024	859.650024	261514

1515 rows × 7 columns

3.2.9 NIFTYBEES

	Date	Open	High	Low	Close	Adj Close	Volume
0	2016-01-01	80.199997	80.750000	80.000000	80.505997	80.505997	273470
1	2016-01-04	80.294998	80.349998	78.900002	79.030998	79.030998	403970
2	2016-01-05	79.387001	79.394997	78.794998	78.981003	78.981003	347970
3	2016-01-06	78.980003	79.087997	78.370003	78.515999	78.515999	486200
4	2016-01-07	78.200996	78.300003	76.724998	76.891998	76.891998	1563600
...	...	...	...	...	...	...	...
1510	2022-02-11	192.500000	192.500000	187.000000	188.000000	188.000000	2152018
1511	2022-02-14	187.500000	187.500000	182.300003	182.589996	182.589996	7366629
1512	2022-02-15	185.000000	187.880005	182.589996	187.589996	187.589996	2687788
1513	2022-02-16	190.000000	190.000000	186.449997	187.139999	187.139999	2043037
1514	2022-02-17	190.000000	190.000000	186.169998	186.929993	186.929993	1415247

1515 rows × 7 columns

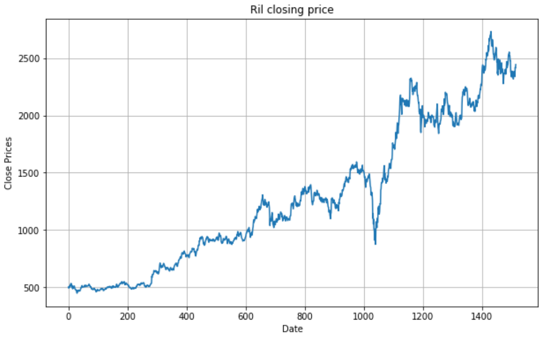
3.2.10 BANKBEES

	Date	Open	High	Low	Close	Adj Close	Volume
0	2016-01-01	170.199997	171.800003	169.259995	171.347000	171.347000	75270
1	2016-01-04	170.350006	170.399994	167.014999	167.294998	167.294998	67670
2	2016-01-05	167.500000	167.750000	166.251007	166.813995	166.813995	35430
3	2016-01-06	166.800003	167.199997	165.100006	165.123993	165.123993	94420
4	2016-01-07	165.000000	165.000000	161.600006	161.979996	161.979996	1622460
...	...	...	...	...	...	...	...
1510	2022-02-11	392.000000	392.700012	386.510010	388.000000	388.000000	771155
1511	2022-02-14	382.000000	387.000000	371.239990	372.200012	372.200012	1808412
1512	2022-02-15	375.489990	385.000000	369.769989	384.279999	384.279999	831057
1513	2022-02-16	383.600006	386.700012	380.500000	381.549988	381.549988	621344
1514	2022-02-17	385.000000	385.000000	377.010010	377.929993	377.929993	540092

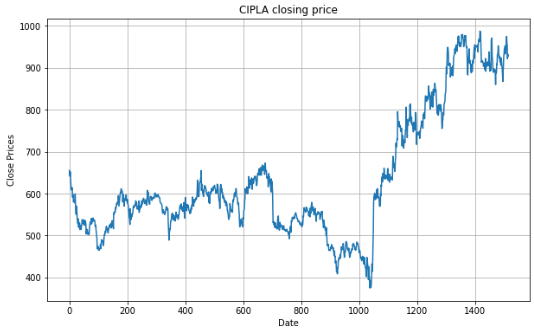
1515 rows × 7 columns

Figure 3.2. Figures of the datasets in the portfolio.

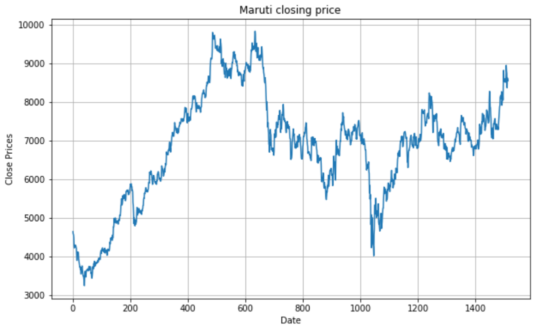
3.3 Line Graph of closing price of various stocks



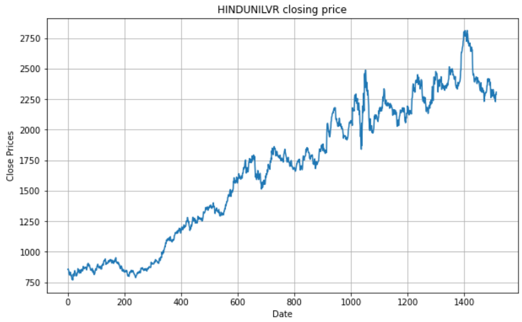
RELIANCE



CIPLA

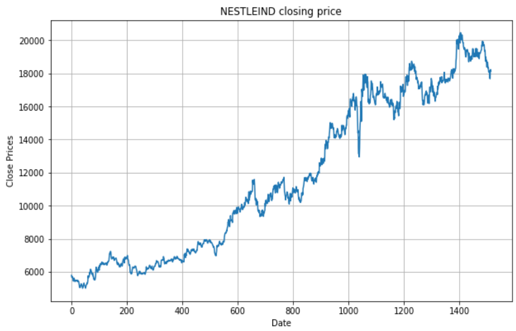


MARUTI



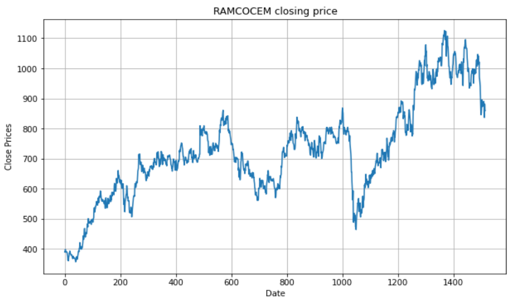
HINDUSTAN UNILEVER

ICICI BANK

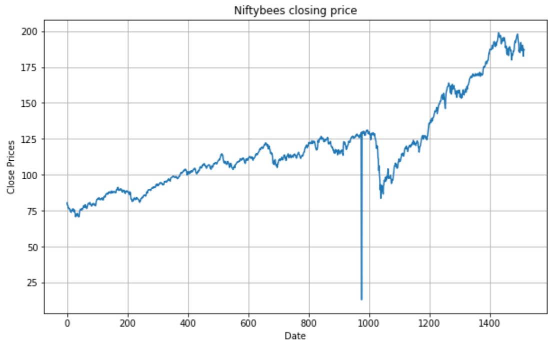


NESTLE INDUSTRIES

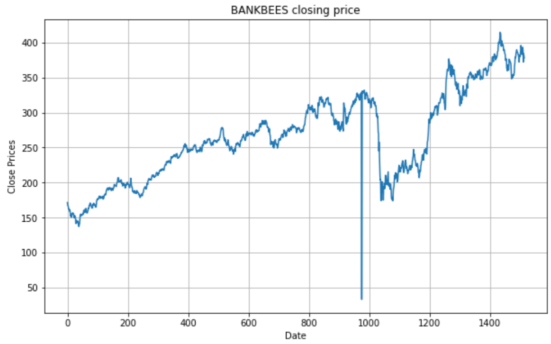
TATA CONSUTANCY SERVICES



RAMCO CEMENTS



NIFTYBEES



BANKBEES

Figure 3.3. Figures of the Line Graph of closing price of various stocks.

3.4 Machine Learning, Deep Learning and Finance

Amid the past few decades, machine learning and deep learning has been accomplishing marvelling results in a plethora of applications. In Finance for instance, deep learning has altogether made a difference in discovering the complex covered up designs in information. This tool has made the decision-making handle much easier and quicker compared to the utilize of ordinary and conventional strategies. Artificial Intelligence, in common, has been playing a notable part within the resource administration industry. Some of the critical applications of profound learning within the field of finance include technical analysis, risk management, automated trading, robo-advisors, index replication, and more[11].

Previously, the financial sector checked primarily on organized information, which is represented by prices and volumes, to conduct the statistical analysis. All things considered, the examination of unstructured data, like news stream, was essentially cleared out to human judgment. Presently, with the headway of AI innovation, different modern tools, such as sentiment and expression analysis, have been able to get it the disposition of diverse markets all over the world. Therefore, these sophisticated means have made the work with delicate and unstructured information very handy and machine-accessible.

The capacity of machine and deep learning to understand and learn from various sorts of data, like market order book and financial information, has without a doubt enhanced productivity and brought gigantic changes to the financial industry. Deep Learning comes very helpful, particularly in errands that people have no intrinsic capacity to perform. For occurrence, it can be very challenging for people, indeed with broad involvement within the field, to accurately anticipate and select which set of stocks are likely to outperform in a future time interval. Subsequently, devices like deep learning can be exceptionally invaluable in optimizing costs, scaling up services, and improving client experiences.

Deep learning is a subset of machine learning that works mainly with networks in order to learn from unlabelled and unstructured data without any human supervision. This AI function utilizes a set of Artificial Neural Networks (ANN) that are built to process data in a similar way to the human brain. It primarily consists of a set of neural nodes that are connected in hierarchical manner, like a web, to simulate the data by using a non-linear approach [14].

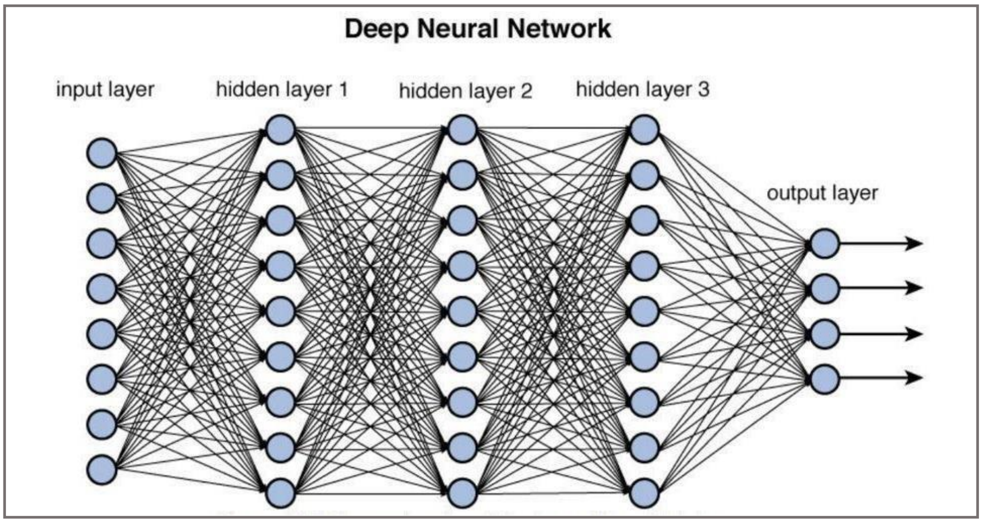


Figure 3.4. Deep Neural Network Architecture

By applying deep learning, we are mainly aiming at using a high-dimensional input X to get a predictor of an output Y. The learning process is treated as an input-output that maps Y = F(X) where X = (X1,...,Xp) such that the predictor is denoted as follows: Ŷ (X) := F(X). What distinguishes deep learning is its ability to pass the learned features through multiple layers. It utilizes hierarchical predictors that consist of L non-linear iterations that are applied to the input X. Each one of these L transformations represent a whole layer. Through the different layers, the constructed algorithm is able to learn various attributes and extract them as factors. f1, fL represent the activation functions of the L layers that transform weighted data through the non-linear activation functions. The activation rule is defined by: $fl\ W,b$:= fl ($\sum WljXj + bl$) = fl ($WlXl + bl$), $Nl\ j$=1 1 ≤ l ≤ L

such that l ∈ {1, . . . , L} depict the hidden layers that have Nl hidden units. Given the number of layers L, the deep predictor becomes a composite map represented by: Ŷ (X) : = F(X) = ($f1\ W1,b1$o … . . o$fL\ WL,bL$)(X)

Z (l) designates the l-th layer of the neural networks, which means that the input X = Z (0) and the last layer represents the response Y which is either categorical or numerical. The prediction rule can be explicitly represented by the following structure:

$$
\begin{aligned}
Z^{(1)} &= f^{(1)}\left(W^{(0)}X + b^{(0)}\right), \\
Z^{(2)} &= f^{(2)}\left(W^{(1)}Z^{(1)} + b^{(1)}\right), \\
&\ldots \\
Z^{(L)} &= f^{(L)}\left(W^{(L-1)}Z^{(L-1)} + b^{(L-1)}\right), \\
\hat{Y}(X) &= W^{(L)}Z^{(L)} + b^{(L)}.
\end{aligned}
$$

Such that W(l) represent the weight matrices, b(l) refers to the activation layers, and Z(l) are the hidden features extracted by the algorithm. The better the activation function f(l), the better is the designed predictor [15].

Training a Neural Network

To construct a deep learner, the data is commonly split into training, validation, and testing datasets. The validation set is not mandatory; its major role is to minimize the over-fitting. On the other hand, the training set adjusts the different weights of the network, whereas the testing set is used to evaluate the predictive power of the model's learner. After the identification of the activation function, the depth, and the size of the learner, the learning parameters ($\hat{W}$, $\hat{b}$) are computed such that:

$$\hat{W} = (\hat{W}0 , \dots , \hat{W}L) \text{ and } \hat{b} = (\hat{b}\ 0 , \dots , \hat{b}\ L)$$

To be able to retrieve these parameters, it is imperative to have a training dataset $D = \{Y\ (i)\ , X\ (i)\ \}\ i{=}1\ T$ that consists of input-output pairs. A loss function $\mathcal{L}(Y,\hat{Y})$ is needed as well at the output level. Generally, without any consideration of the regularization penalty, which is used to avoid overfitting, we are striving to solve:

$$\text{Arg min W,b } 1\ T \sum \mathcal{L}(Yi\ T\ i{=}1 , \hat{Y}W,b\ (Xi)).$$

By minimizing the loss function, we get the mean-squared error over the training set D as follows:

$$\mathcal{L}(Yi , \hat{Y}(Xi)) = ||Yi - \hat{Y}(Xi)||_2^2$$

4.Time Series Econometrics

4.1 Time Series Data

Time series data is characterized as a collection of values of a variable that contrasts over time. The intervals between observations of a time series can change. In any case, the range of the intervals ought to be consistent all through the observed period e.g. daily, weekly, monthly etc. In common, the time series is accepted to be stationary in empirical work based on time series (Gujarati & Watchman 2008).[12]

4.2 Stochastic Processes

A prepare is said to be stochastic, or random, in the event that the collection of a variable is accumulated over a sequence of time. A stochastic process can be either stationary or nonstationary (Gujarati & Porter 2008).

4.3 Autoregressive Model

An autoregressive model could be a demonstrate where the dependent variable is regressed on at least one lagged period of itself. On the off chance that an autoregressive model incorporates one lagged period of itself, it takes after a first-order autoregressive stochastic prepare, denoted AR(1). Besides, in the event that the model includes p number of lagged periods of the dependent variable, it takes after a pth-order autoregressive process, denoted AR(p).

4.4 Stationary Process

There are different types of stationarity. Second order stationary, commonly known as weakly stationary, is considered to be adequate in most empirical works. A stochastic process is weakly stationary in the event that it has constant mean and variance and the covariance is time invariant, i.e. the statistics do not alter over time (Gujarati & Watchman 2008). A white noise process could be a special sort of stationary stochastic process. A stochastic process is considered to be white noise on the off chance that the mean is equal to zero, the variance is constant, and the observations are serially uncorrelated (Gujarati & Watchman 2008).

4.5 Nonstationary Process

A stochastic process that has a time-varying mean, variance, or covariance is said to be nonstationary. Financial data usually takes after a random walk which is a sort of nonstationary stochastic handle. A random walk is either with or without drift, showing the presence of an intercept, and is an AR(1) process. Regressing Y_t on Y_{t-1} estimates the following

$$Y_t = \rho Y_{t-1} + \mu_t$$

and in the event that ρ equals 1, the model gets to be what is known as a random walk (Gujarati & Watchman 2008). A random walk without drift is a process where the dependent variable can be estimated on one lagged period of itself plus an error term, assumed to be white noise, known as a random shock. The formula for a random walk without drift excludes the intercept. The mean is constant over time in a random walk without drift, be that as it may, the variance is expanding indefinitely over time, making it a nonstationary stochastic process (Gujarati & Porter 2008).

Random walk without drift:

$$Y_t = Y_{t-1} + u_t$$

Similar to a random walk without drift, a random walk with drift may be a process where the variable is dependent on its own lagged values and a random shock. In any case, the demonstrate that will be utilized to estimate a random walk with drift includes an intercept known as the drift parameter, signified by δ. This parameter indicates in the event that the time series is trending upwards or downwards, depending on whether δ is positive or negative. A irregular walk with drift may be a nonstationary stochastic process since the mean and variance are increasing over time (Gujarati & Watchman 2008).

Random walk with drift:

$$Y_t = \delta + Y_{t-1} + u_t$$

The preceding random walks have infinite memory which suggests that the effects of random shocks continue all through the complete time period. The random walks are known as difference stationary processes, meaning that even in spite of the fact that the stochastic processes are nonstationary, they ended up stationary through the first order difference (Gujarati & Doorman 2008).

4.6 Integrated Process

A nonstationary stochastic process that has got to be differenced one time to ended up stationary, is said to be integrated of the first order, signified $I(1)$. Moreover, a nonstationary stochastic process that has to be differenced twice to become stationary, is said to be integrated of the second order, denoted $I(2)$. Furthermore, this implies that a nonstationary stochastic process that needs to be differenced d times, is said to be integrated of order d, indicated $Y_t \sim I(d)$. A time series that's stationary without any differencing is integrated of order zero, indicated $Y_t \sim I(0)$ (Gujarati & Doorman 2008).

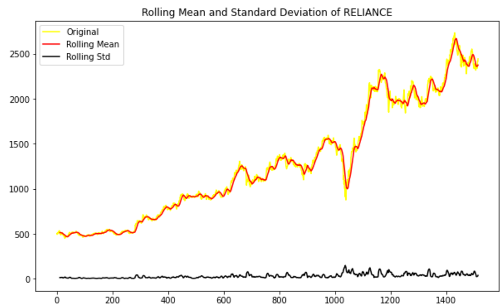
4.7 Deterministic Trend.

A time series that's deterministic can be superbly forecasted. Be that as it may, most time series are mostly deterministic and in part stochastic, making them inconceivable to predict perfectly due to the probability distribution of future values (Chatfield 2003). If a variable is dependent on its past values and a time variable, it is assessed by the following;

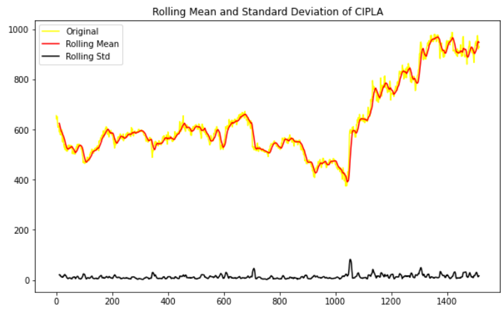
$$Yt = \beta_1 + \beta_2 t + Y_{t\text{-}1} + u_t$$

where t could be a variable that measures time chronologically and ut is an error term, assumed to be white noise. The condition is known as a random walk with drift and deterministic trend and is stochastic but moreover in part deterministic, due to the time trend t (Gujarati & Doorman 2008).

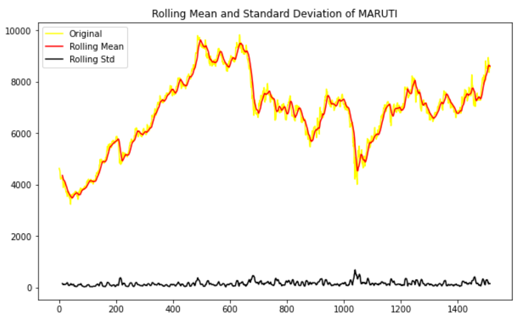
4.8. Rolling Mean and Standard Deviation in our dataset



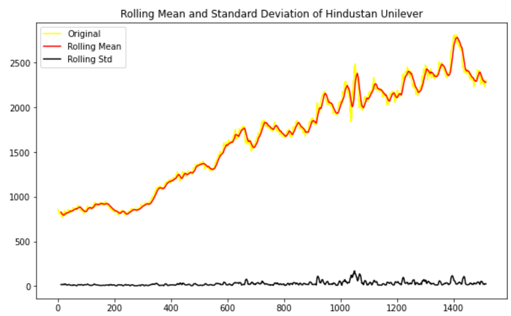
RELIANCE



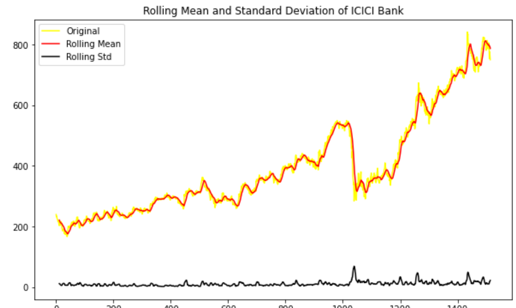
CIPLA



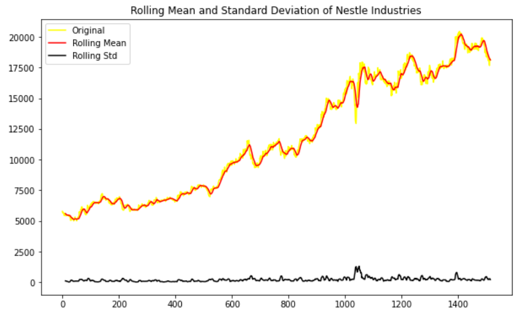
MARUTI



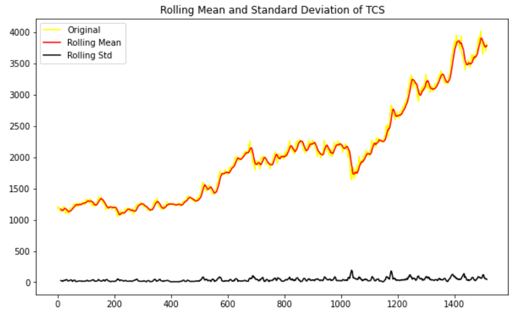
HINDUSTAN UNILEVER



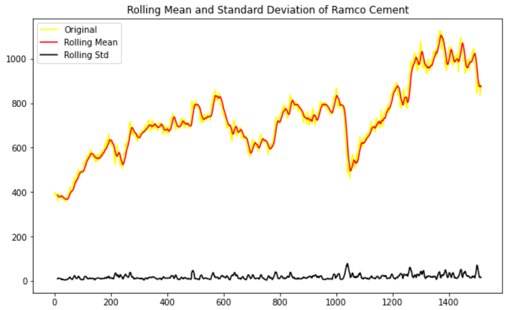
ICICI BANK



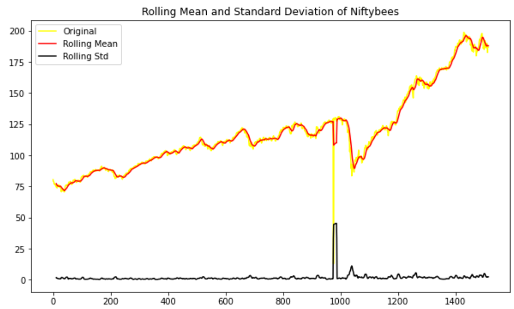
NESTLE INDUSTRIES



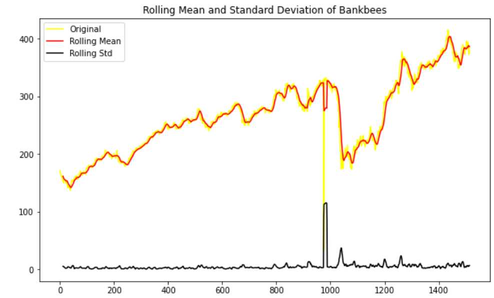
TATA CONSULTANCY SERVICES



RAMCO CEMENT



NIFTYBEES



BANKBEES

5. Modelling of Time Series Data

When working with estimating of time series data, the basic time series is accepted to be stationary. Assuming stationarity, there are a few distinctive approaches to develop forecasting models, for illustration an autoregressive prepare, a moving average process, an autoregressive and moving average process, and an autoregressive integrated moving average process (Gujarati & Porter 2008).

5.1 Autoregressive Integrated Moving Average Process

On the off chance that the time series of an ARMA model must be differenced a certain number of times to become stationary, the model gets to be what is known as an autoregressive integrated moving average model, or an ARIMA model. As mentioned previously, a time series which must be differenced d number of times in order to ended up stationary, is integrated of order d, signified *I(d).* In its common frame, the ARIMA show is indicated ARIMA(p, d, q) which suggests that the AR is of the pth-order, the time series is integrated d number of times, and the moving average is of the qth-order. This further means that on the off chance that the basic AR and MA models are of the first-order, and the time arrangement is stationary at the first difference, the ARIMA model is indicated ARIMA(1, 1, 1). It is vital to note that an ARIMA model isn't inferred from any economic hypothesis, that is, it is an atheoretic model.

Seasonal Decomposition

Time series decomposition generally involves partitioning a signal into *seasonal*, *trend*, *residual* and sometimes *level*, *holiday* etc. components, which assumes additive or multiplicative relationships. seasonal_decompose method in statsmodel.tsa library is a simple application of this. A more advanced version, which applies Box-Cox transformation beforehand automatically and accepts multiple seasonality frequencies, is mstl in R. Facebook's Prophet also employs seasonal decomposition in itself. And, usually, a ARMA model on residuals followed by a decomposition step is a typical approach in analysis.

In ARIMA, there isn't a decomposition of such type. It's a generalization of ARMA models, in which we first difference the series and fit an ARMA model. The differencing step is applied to make the signal *more* stationary, by eliminating trend and seasonality components. ARIMA models considering seasonality are called SARIMA, but it also doesn't involve decomposition.

Dicky Fuller Test

Before going into ADF test, let's first get it what is the Dickey-Fuller test. A Dickey-Fuller test could be a unit root test that tests the mull hypothesis that α=1 within the following model equation. alpha is the coefficient of the primary lag on Y. Null Hypothesis (H0): alpha=1

$$y_t = c + \beta t + \alpha y_{t-1} + \phi \Delta Y_{t-1} + e_t$$

where

- y(t-1) = lag 1 of time series.
- delta Y(t-1) = first difference of the series at time (t-1)
  Fundamentally, it has similar null hypothesis as the unit root test. That is, the coefficient of Y(t-1) is 1, implying of a unit root. If not rejected, the series is said to be non-stationary. The Augmented Dickey-Fuller test evolved based on the above equation and is one of the most common form of Unit Root test.

As the name suggest, the ADF test is an 'augmented' version of the Dickey Fuller test. The ADF test expands the Dickey-Fuller test equation to include high order regressive process in the model.

$$y_t = c + \beta t + \alpha y_{t-1} + \phi_1 \Delta Y_{t-1} + \phi_2 \Delta Y_{t-2} .. + \phi_p \Delta Y_{t-p} + e_t$$

If you notice, we have only added more differencing terms, while the rest of the equation remains the same. This adds more thoroughness to the test. The null hypothesis however is still the same as the Dickey Fuller test. A key point to remember here is: Since the null hypothesis assumes the presence of unit root, that is α=1, the p-value obtained should be less than the significance level (say 0.05) in order to reject the null hypothesis. Thereby, inferring that the series is stationary. However, this is a very common mistake analysts commit with this test. That is, if the p-value is less than significance level, people mistakenly take the series to be non-stationary.

These are the Dicky fuller test results obtained from the dataset

```
Results of dickey fuller test
Test Statistics                  -0.225584
p-value                           0.935403
No. of lags used                  0.000000
Number of observations used    1514.000000
critical value (1%)              -3.434677
critical value (5%)              -2.863451
critical value (10%)             -2.567787
dtype: float64
```

RELIANCE

```
Results of dickey fuller test
Test Statistics                  -0.621817
p-value                           0.865976
No. of lags used                  1.000000
Number of observations used    1513.000000
critical value (1%)              -3.434679
critical value (5%)              -2.863452
critical value (10%)             -2.567788
dtype: float64
```

CIPLA

```
Results of dickey fuller test
Test Statistics                  -1.776535
p-value                           0.392172
No. of lags used                  0.000000
Number of observations used    1514.000000
critical value (1%)              -3.434677
critical value (5%)              -2.863451
critical value (10%)             -2.567787
dtype: float64
```

MARUTI

```
Results of dickey fuller test
Test Statistics                  -1.211951
p-value                           0.668434
No. of lags used                  7.000000
Number of observations used    1507.000000
critical value (1%)              -3.434697
critical value (5%)              -2.863460
critical value (10%)             -2.567792
dtype: float64
```

HINDUSTAN UNILEVER

```
Results of dickey fuller test
Test Statistics                  -0.532761
p-value                           0.885463
No. of lags used                 24.000000
Number of observations used    1490.000000
critical value (1%)              -3.434746
critical value (5%)              -2.863482
critical value (10%)             -2.567804
dtype: float64
```

ICICI BANK

```
Results of dickey fuller test
Test Statistics                  -0.532761
p-value                           0.885463
No. of lags used                 24.000000
Number of observations used    1490.000000
critical value (1%)              -3.434746
critical value (5%)              -2.863482
critical value (10%)             -2.567804
dtype: float64
```

NESTLE INDUSTRIES

```
Results of dickey fuller test
Test Statistics                  0.453322
p-value                          0.983383
No. of lags used                 2.000000
Number of observations used   1512.000000
critical value (1%)             -3.434682
critical value (5%)             -2.863453
critical value (10%)            -2.567789
dtype: float64
```

TATA CONSUTANCY SERVICES

```
Results of dickey fuller test
Test Statistics                 -2.210937
p-value                          0.202300
No. of lags used                 0.000000
Number of observations used   1514.000000
critical value (1%)             -3.434677
critical value (5%)             -2.863451
critical value (10%)            -2.567787
dtype: float64
```

RAMCO CEMENT

```
Results of dickey fuller test
Test Statistics                 -0.068644
p-value                          0.952474
No. of lags used                10.000000
Number of observations used   1504.000000
critical value (1%)             -3.434705
critical value (5%)             -2.863464
critical value (10%)            -2.567794
dtype: float64
```

NIFTYBEES

```
Results of dickey fuller test
Test Statistics                 -1.307638
p-value                          0.625671
No. of lags used                10.000000
Number of observations used   1504.000000
critical value (1%)             -3.434705
critical value (5%)             -2.863464
critical value (10%)            -2.567794
dtype: float64
```

BANKBEES

Figure 5. The Dicky Fuller test results obtained from the dataset.

As you can see from the above results the p-value>0.05 so we cannot reject the Null Hypothesis and Test Statistics is greater than the critical value. So the data is not stationary.

ARIMA with Python

The statsmodels library provides the capability to fit an ARIMA model.An ARIMA model can be created using the statsmodels library as follows:

1. Defining the model by calling ARIMA() and passing in the *p*, *d*, and *q* parameters.
2. The model is prepared on the training data by calling the fit() function.
3. Predictions can be made by calling the predict() function and specifying the index of the time or times to be predicted.

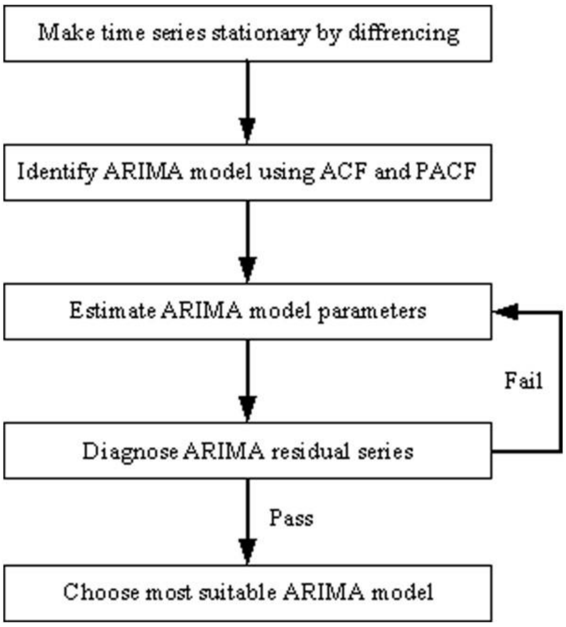
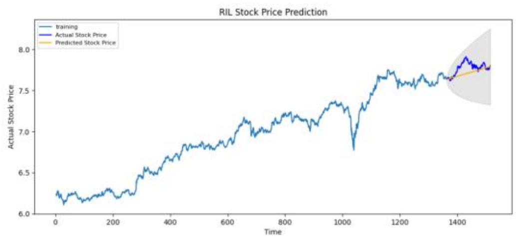
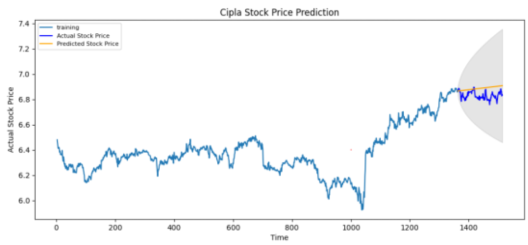


Figure 5.1 A pictorial representation of ARIMA process

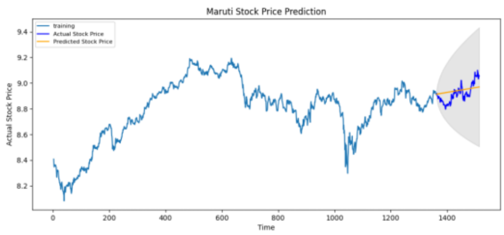
Below are some of the Price Predictions of ARIMA model for various stocks



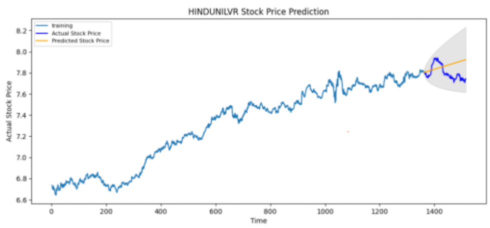
RELIANCE



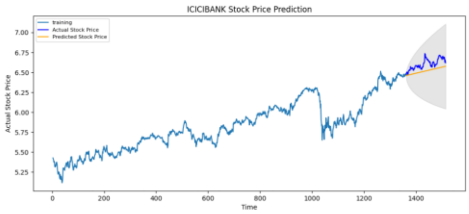
CIPLA



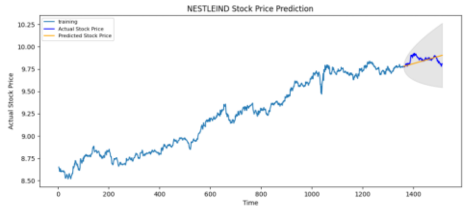
MARUTI



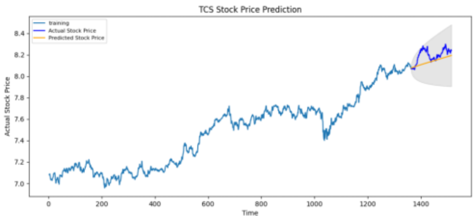
HINDUSTAN UNILEVER



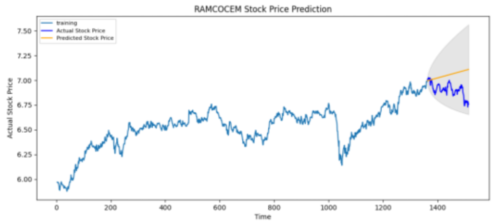
ICICI BANK



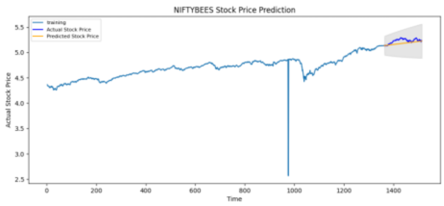
NESTLE INDUSTRIES



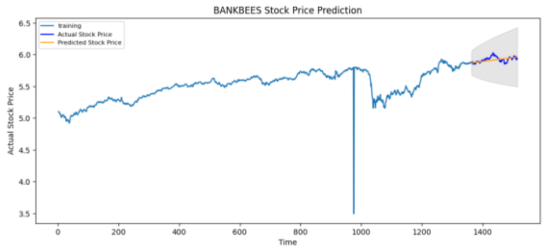
TATA CONSUTANCY SERVICES



RAMCO CEMENT



NIFTYBEES



BANKBEES

Figure5.2 Price Predictions of ARIMA model for various stocks

Below are the RMSE and MAPE values for the ARIMA model:

	RMSE-(ARIMA Model)	MAPE-(ARIMA Model)
RELIANCE	0.087198	0.836956
CIPLA	0.065674	0.848858
MARUTI	0.063869	0.609825
HUL	0.109032	1.229528
ICICI BANK	0.096046	1.310109
NESTLE IND	0.051939	0.376995
TCS	0.075441	0.723739
RAMCO CEMENT	0.173366	2.180935
NIFTYBEES	0.058234	0.916014
BANKBEES	0.039547	0.500408

Figure 5.3 RMSE and MAPE values for the ARIMA model.

6. The Long Short-Term Memory(LSTM)

The Long short-term memory (LSTM) is a special type of Repetitive Neural Systems (RNN) architecture that's primarily utilized in deep learning. LSTM employments a set of feedback connections to process sequences of data. This design is known for its productivity in making predictions, processing, and classifying large-scale time-series data in spite of the presence of a few lags between events. Other than, LSTMs were outlined in such a way that they are able to overcome the issues that can possibly emerge due to the vanishing gradient. It was named long short-term memory since its cell unit has the capacity to forget a portion of already stored data and can, at the same time, memorize extra new pieces of data.

6.1 LSTM Structure

An LSTM unit has the following elements:

• Cell: represents the memory part of the LSTM that monitors the dependencies between different elements constituting the input sequence (Figure 1).

• Input gate: regulates the information flowing into the cell.

• Output gate: regulates the information flowing out of the cell.

• Forget gate: remembers the different values over a specific time interval. In the presence of a forget gate, the forward pass is represented by the following equations:

$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
\tilde{c}_t &= \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}
$$

Such that c0 = 0, h0 = 0, and the subscript t represents the time step. The notation o refers to the Hadamard product.

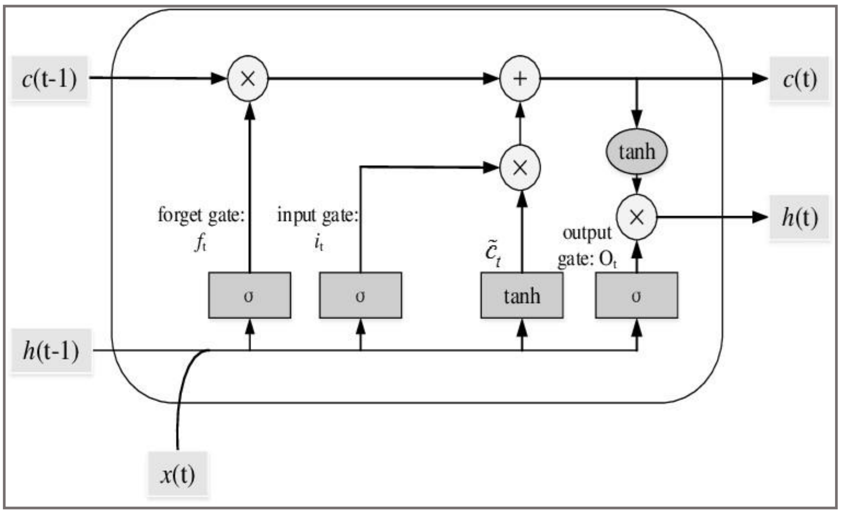


Figure 6.1 The structure of an LSTM cell

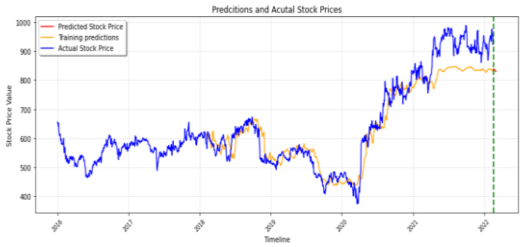
6.2. The Activation Function of the LSTM

For most of the LSTM problems, the logistic sigmoid functions are used as gate activators for the artificial neurons [10]. This mathematical function is characterized by its curve shaped like an “S” (Sigmoid curve), which is defined as:

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^{x}}{e^{x}+1}$$

Below are some of the Price Predictions of L.S.T.M model for various stocks

RELIANCE



CIPLA

MARUTI

HINDUSTAN UNILEVER

ICICI BANK

NESTLE INDUSTRIES

TATA CONSUTANCY SERVICES

RAMCO CEMENT

NIFTYBEES

BANKBEES

Figure 6.2. Price Predictions of L.S.T.M model for various stocks.

Below are the RMSE and MAPE values for 0.01 and 0.001 learning rate

RMSE and MAPE values for 0.01 learning rate

	RMSE-(LSTM)	MAPE-(LSTM)
RELIANCE	162.026482	35.243742
CIPLA	52.621882	27.462020
MARUTI	452.042989	15.361119
HUL	145.906470	18.213694
ICICI BANK	70.161319	31.863781
NESTLE IND	834.447368	30.128489
TCS	191.710740	28.695352
RAMCO CEMENT	86.442738	17.704830
NIFTYBEES	9.789166	22.378130
BANKBEES	26.668177	19.953055

RMSE and MAPE values for 0.001 learning rate

	RMSE-(LSTM)	MAPE-(LSTM)
RELIANCE	137.452160	37.291918
CIPLA	48.965489	28.138875
MARUTI	483.250810	15.688190
HUL	125.524711	19.143855
ICICI BANK	132.642433	27.367794
NESTLE IND	884.986979	30.460640
TCS	176.229420	28.252813
RAMCO CEMENT	87.447337	17.447590
NIFTYBEES	23.433939	18.070080
BANKBEES	26.668177	20.561935

Figure 6.3. RMSE and MAPE values of L.S.T.M model.

As you can see there is a slight difference in the values when the learning rate is varied.

7. The Facebook Prophet

Facebook Prophet is an open-source algorithm for creating time-series models that employments a few old ideas with a few new twists. It is especially great at modelling time series that have multiple seasonality and doesn't confront a few of the over disadvantages of other algorithms. At its core is the sum of three functions of time plus an error term: growth g(t), seasonality s(t), holidays h(t) , and error e_t :

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

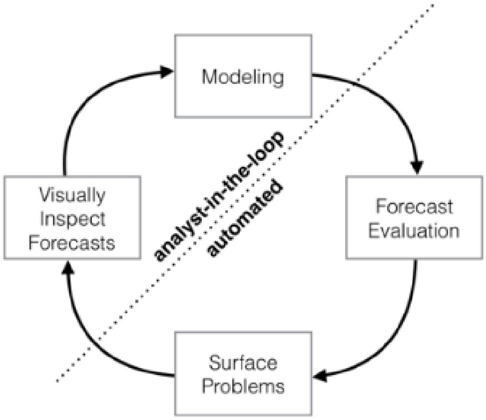


Figure 7. The Facebook Prophet Architecture

7.1. The Growth Function (and change points):

The growth function models the overall trend of the data. The old idea ought to be recognizable to anybody with a basic information of linear and logistic functions. The new idea consolidated into Facebook prophet is that the growth trend can be present at all points within the data or can be changed at what Prophet calls "change points".

Change points are moments in the data where the data shifts direction.
Prophet can automatically detect change points or you can set them yourself. You can also adjust the power the change points have in altering the growth function and the amount of data taken into account in automatic change point detection.

The growth function has three main options:

- **Linear Growth:** This is the default setting for Prophet. It uses a set of piecewise linear equations with differing slopes between change points. When linear growth is used, the growth term will look similar to the classic y=mx+b from middle school, except the slope(m) and offset(b) are variable and will change value at each change point.
- **Logistic Growth:** This setting is useful when your time series has a cap or a floor in which the values you are modelling becomes saturated and can't surpass a maximum or minimum value (think carrying capacity). When logistic growth is used, the growth term will look similar to a typical equation for a logistic curve (see below), except it

the carrying capacity (C) will vary as a function of time and the growth rate (k) and the offset(m) are variable and will change value at each change point.

$$g(t) = \frac{C(t)}{1 + x^{-k(t-m)}}$$

- **Flat:** Lastly, you can choose a flat trend when there is no growth over time (but there still may be seasonality). If set to flat the growth function will be a constant value.

The Holiday/Event Function:

The holiday function permits Facebook Prophet to adjust forecasting when a holiday or major event may change the forecast. It takes a list of dates and when each date is present in the figure adds or subtracts value from the forecast from the growth and seasonality terms based on historical data on the identified holiday dates.

7.3 Advantages of Facebook Prophet

1). It is accurate and fast. Prophet is used in many applications across Facebook for producing reliable forecasts for planning and goal setting. We've found it to perform better than any other approach in the most of the cases. We fit the models so that you can get forecasts in a few seconds.

2). Fully automatic. Get a decent forecast on the messy data with no manual effort. Prophet is robust to outliers, missing data, and dramatic changes in your time series.

3).Tunable forecasts. Prophet procedure includes many possibilities for users to tweak and adjust forecasts. You can use human interpretable parameters to improve your forecast by adding your domain knowledge.

4).Available in R or Python. We can implement the Prophet procedure in R as well as in Python but they share the same underlying Stan code for fitting. Use whatever language which you are comfortable with to get the forecasts.

	RMSE-(Facebook Prophet)	MAPE-(Facebook Prophet)
RELIANCE	131.310244	7.345378
CIPLA	30.207312	4.144978
MARUTI	504.948712	6.210224
HUL	89.982457	3.645064
ICICI BANK	25.057282	4.966212
NESTLE IND	523.323360	3.374548
TCS	90.388834	3.494713
RAMCO CEMENT	50.049060	4.952475
NIFTYBEES	6.444114	4.325292
BANKBEES	19.617812	6.114020

Figure 7.1 RMSE and MAPE values for the Facebook Prophet model.

Above are the RMSE and MAPE values for the various stocks found in the portfolio.

8. Portfolio Theory.

The Modern Portfolio Theory (MPT), also known as the mean-variance analysis, represents the theory that deals with how risk-averse individuals and investors construct portfolios with an intent to maximize the expected return given a certain level of risk. In 1952, Harry Markowitz developed this theory and published his paper "Portfolio Selection" for which he was awarded the prestigious Nobel prize.
Markowitz's concept argues that portfolios are commonly assessed and evaluated primarily based totally on how risk and return effect the whole set of assets. This concept in particular is based on using statistical measures, consisting of the variance and correlation, to assess the overall performance of the general portfolio. It contends that investors are broadly speaking risk-averse, that's why they need to spend money on more than one asset training so that you can attain better returns for an extremely low risk degree. This concept of proudly owning specific varieties of property is what's referred to as diversification. The adoption of the diversification principle makes portfolios' risk degree decrease as compared to the funding on a single asset. Therefore, the mean-variance optimizer theory is based on specific mathematical and statistical measures consisting of the anticipated returns, standard deviations, and variance-covariance matrix to be able to allocate the pertinent and efficient array of assets with a view to represent the portfolio.

8.1. Portfolio Features and Metrics

8.1.1 Portfolio Return

The return on a portfolio represents the gain/loss succeeding from the investment in varied asset classes. Investors rely all on the historical returns to draw expectations concerning the long term performance, and therefore predict the expected returns.

The portfolio' overall return are typically computed using**:**

rp = wt x r

such as rp represents the return generated by the portfolio, r is that the column vector containing returns of each asset ri, and wt is that the transpose of w, that's the vector containing the portfolio weights.

8.1.2 Portfolio Variance

The portfolio variance is a measure of the dispersion between the portfolio's returns. This variance mostly represents the weighted average of the portfolio's individual securities. It can be calculated by using:

$$\sigma_p^2 = \sum_i w_i^2 \sigma_i^2 + \sum_i \sum_{j \neq i} w_i w_j \sigma_i \sigma_j \rho_{ij},$$

Where σ is the standard deviation of the returns on asset I and ij is the correlation coefficient between the returns on assets I and j.

8.1.2 The Covariance Matrix

The covariance is the mean of the directional relationship between the assets in a portfolio. We use the following formulas to compute the covariance matrix:

$$\boldsymbol{Cov_{Matrix}} = \begin{bmatrix} cov_{11} & cov_{12} & \dots & cov_{1N} \\ cov_{12} & cov_{22} & \dots & cov_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ cov_{N1} & cov_{N2} & cov_{N2} & cov_{NN} \end{bmatrix}$$

Such that

$$cov_{ij} = cov_{ji} = E[(R_i - r_i) * (R_j - r_j)]$$

$$w = [w_1, w_2, \dots, w_N]^T$$

The variation of the individual assets is represented by the matrix's diagonals. When the covariance is positive, it suggests that the assets in the portfolio tend to move in the same direction. A negative correlation, on the other hand, means that the assets often move in the opposite direction. In general, the lower the correlation across assets, the lower the volatility of the portfolio as a whole. That is the primary reason why professional fund managers diversify portfolios by selecting uncorrelated stocks, understanding that the link between these assets can have a major impact on the overall riskiness of the portfolio.

8.1.3 Portfolio Volatility

The volatility of a portfolio is quantified by the standard deviation, which is calculated as the square root of the variance. In fact, the standard deviation and variance can be used interchangeably in most circumstances because they both depict the level of risk.

$$\sigma_p = \sqrt{\sigma_p^2}$$

The basic goal is to choose a combination of assets that has a lower standard deviation and thus a lower risk than the individual assets.

8.1.4 The Sharpe Ratio

The Sharpe Ratio is one of the most often used methods by investors to calculate the excess return on an investment above the risk-free rate (rf) per total risk. The Sharpe ratio is calculated using the formula:

$$Sharpe\ Ratio = \frac{R_p - R_f}{\sigma_p}$$

Such that: R_p: The portfolio return.

R_f: The risk-free rate.

σ_p: The standard deviation of the portfolio's return.

In general, the higher the Sharpe ratio of a portfolio, the higher its risk-adjusted return. A negative Sharpe ratio, on the other hand, indicates that the portfolio's return is likely to be negative or less than the risk-free rate. The risk-free rate was set to zero in this study since the minimum an investor may get from a risk-free investment is 0%. So for instance for the Indian market, using Bloomberg, as of August 12th, 2020, the 10 Year Government Bond Yield is 5.85%. The probability of default backed out by Sovereign CDS Spreads is 1.61%. The real risk-free rate is 4.24%. (5.85%-1.61%)[16].

9. The Efficient Frontier

By adjusting the weights associated with each stock, an investor can create several portfolios using a single set of assets. The efficient frontier is obtained by charting all feasible portfolios with various weighting configurations. This strategy, pioneered by Harry Markowitz, aids in the discovery of investment portfolios that provide the highest returns at the lowest risk. To illustrate, the black line in Figure below illustrates the efficient frontier of

the portfolio, which explicitly highlights the portfolios with the highest projected return for a given risk level. This plot can be used to generate various weight combinations based on the requirements and risk tolerance of each investor's requirements and risk aversion. For example, we can readily differentiate between the portfolio with the lowest risk and the one with the highest Sharpe ratio. Portfolios on the plot's right are defined by a high level of risk. Despite this, portfolios based on the efficient frontier tend to deliver lower returns.
As a result, using this plot aids in determining the ideal and efficient portfolio that corresponds to each investor's level of risk tolerance.

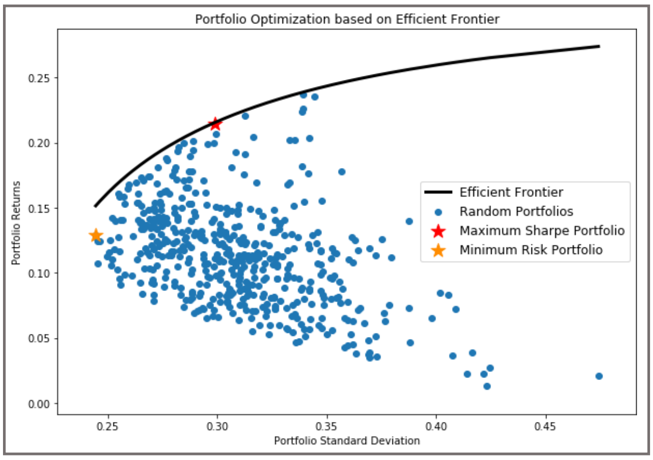


Figure 9. The Efficient Frontier Model.

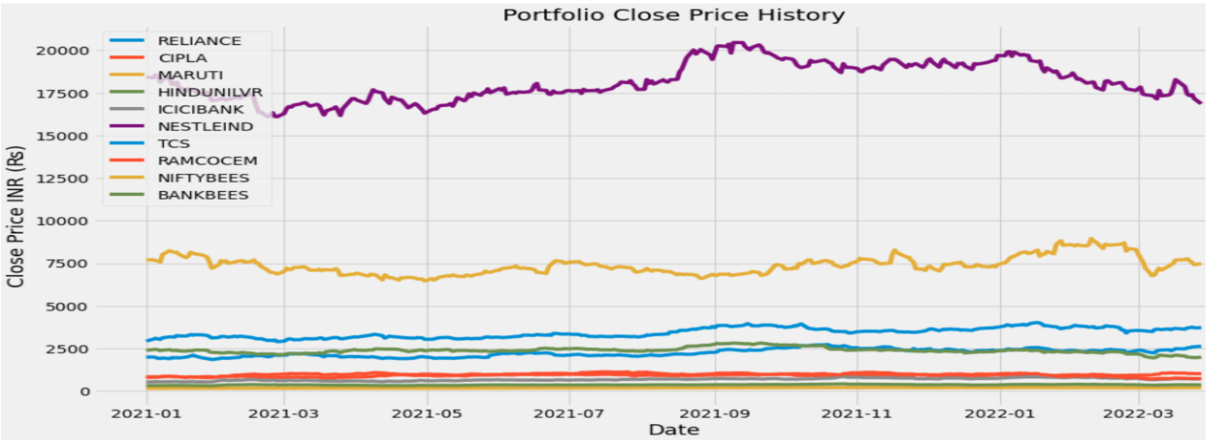
As a result, the model that outlines asset allocation and optimization for a risk-averse investor is as follows:

$$
\begin{aligned}
\underset{\omega}{\text{minimize}} \quad & \omega^t C \omega \\
\text{subject to:} \quad & r_p = \omega^t r, \\
& \sum_{i=1}^{n} \omega_i = 1, \\
& \text{No short selling constraint, thus } \omega_i \geq 0, \\
& a < \omega_i < b
\end{aligned}
$$

C is the covariance matrix for the asset returns, and a and b are the lowest and maximum weights assigned to these assets, respectively.
Furthermore, in our scenario, we assume that no short selling occurs.

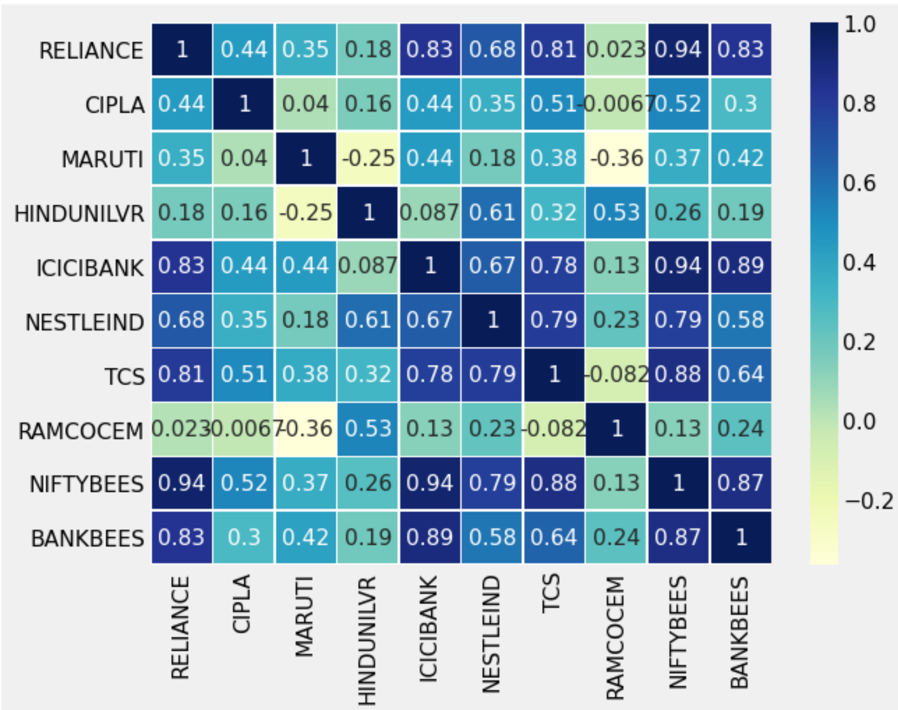
Below is the dataset that is used for the portfolio optimization which starts from 01-01-2021 to 28-03-2022.

	RELIANCE	CIPLA	MARUTI	HINDUNILVR	ICICIBANK	NESTLEIND	TCS	RAMCOCEM	NIFTYBEES	BANKBEES
Date										
2021-01-01	1987.50	826.60	7691.30	2387.55	527.50	18450.70	2928.25	795.00	149.57	313.65
2021-01-04	1990.85	832.25	7702.30	2426.50	531.70	18377.95	3039.45	798.35	150.71	314.12
2021-01-05	1966.10	827.25	7655.45	2450.55	537.25	18558.25	3093.00	793.90	151.30	318.73
2021-01-06	1914.25	824.80	7628.60	2417.30	546.70	18515.25	3051.50	802.45	151.22	319.67
2021-01-07	1911.15	826.55	7566.05	2368.85	541.10	18127.30	3032.80	832.20	151.20	321.08
...	...	...	...	...	...	...	...	...	...	...
2022-03-22	2531.15	1032.60	7766.65	1993.50	718.20	17429.25	3700.95	739.05	187.34	365.95
2022-03-23	2539.20	1014.45	7644.85	1978.55	718.30	17344.20	3712.40	730.40	186.34	363.38
2022-03-24	2578.65	1029.00	7556.95	1965.90	704.20	17408.10	3749.85	729.55	186.09	358.20
2022-03-25	2595.85	1013.50	7415.45	1953.00	699.25	17161.20	3707.45	727.90	185.82	356.90
2022-03-28	2621.95	1017.10	7477.35	1975.10	710.35	16860.05	3707.70	728.50	186.32	359.75

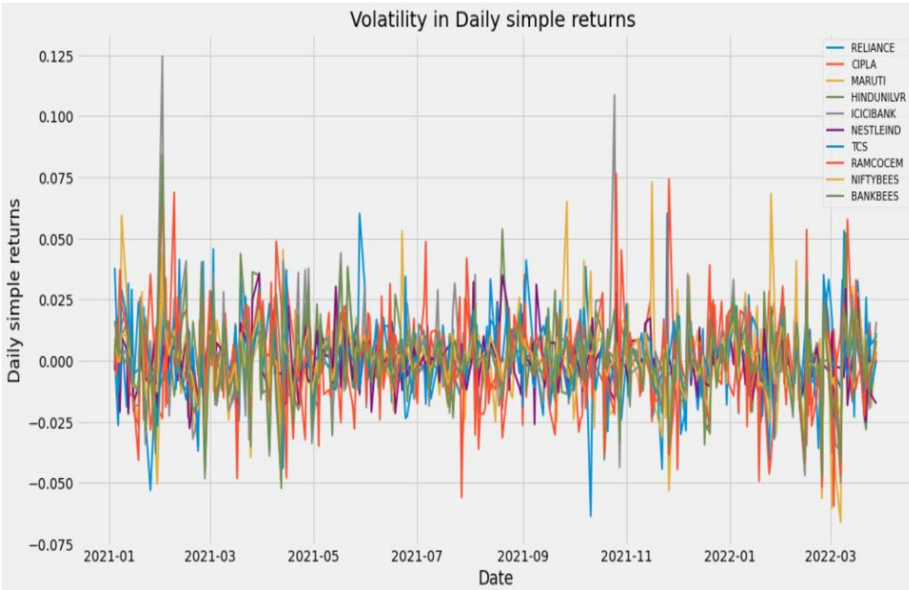


These are the closing prices of the various stocks in the portfolio with Nestle Industries at the top with a purple colour and followed by the rest.

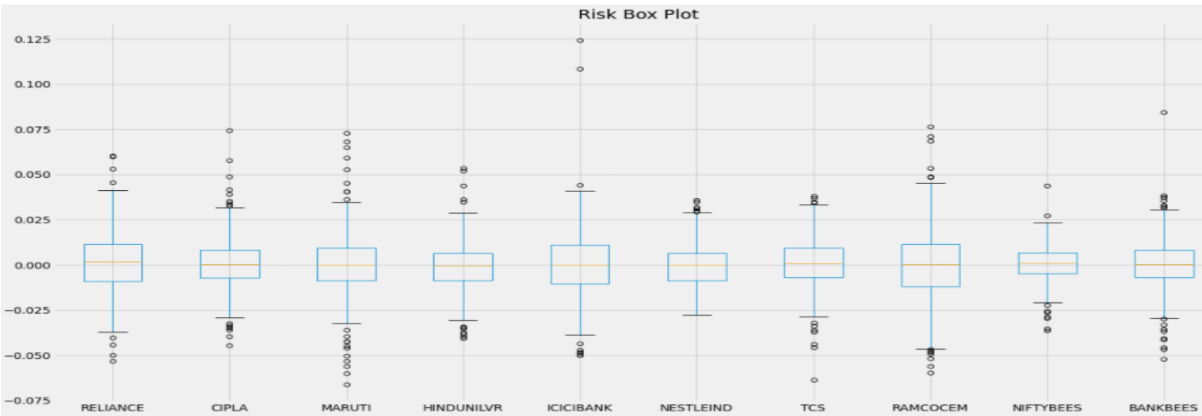
Correlation between Stocks in your portfolio



Here Pearson Correlation is used.With this matrix we can see that Reliance and Niftybees have the highest correlation with 0.94. Ramco Cement and Maruti seems to have the lowest correlation with -0.36.Hence it is wise to have them in our portfolio for efficient diversification which ensures that stock moves in opposite direction.



The above figure is a representation of Volatility in Daily simple returns for the various time period in the dataset.



As you can see from the above Risk Box Plot, ICICI Bank has the highest spread followed BankBees with the outlier present and the Nestle Industries has the least spread.

The above figure shows that for every one-rupee investment made ICICI Bank seems to have given the highest returns and followed by the rest of the stock. However, Hindustan Unilever has given a lower return for the same time period.

```
Enter the amount you want to invest: 100000
Number of stocks to buy with the amount of ₨ 100000.0
              Ticker   Number of stocks to buy
RELIANCE    RELIANCE                         7
CIPLA          CIPLA                        26
ICICIBANK  ICICIBANK                        18
TCS              TCS                         5
NIFTYBEES  NIFTYBEES                       128
Funds remaining with you will be: ₨ 27
```

```
ef.portfolio_performance(verbose=True)
```

```
Expected annual return: 21.9%
Annual volatility: 14.7%
Sharpe Ratio: 1.36

(0.21946321410227423, 0.1470295991637178, 1.3566194510274865)
```

With a capital of ₹100000, the investments made in Reliance (7 shares), Cipla (26 shares), ICICI Bank (18 shares), TCS (5 shares) and Niftybees (128 shares) it gives an expected annual return of 21.9% with an annual volatility of 14.7%. The Sharpe Ratio is 1.36 which is nothing but a risk adjusted return where there is higher returns than the returns taken.

```
cov_matrix = df.pct_change().apply(lambda x: np.log(1+x)).cov()
cov_matrix
```

Symbols Symbols	RELIANCE.NS	CIPLA.NS	MARUTI.NS	HINDUNILVR.NS	ICICIBANK.NS	NESTLEIND.NS	TCS.NS	RAMCOCEM.NS	NIFTYBEES.NS	BANKBEES.I
RELIANCE.NS	0.000352	0.000049	0.000141	0.000078	0.000151	0.000074	0.000077	0.000109	0.000060	0.0000
CIPLA.NS	0.000049	0.000298	0.000064	0.000044	0.000067	0.000043	0.000042	0.000049	0.000060	0.0000
MARUTI.NS	0.000141	0.000064	0.000377	0.000095	0.000193	0.000082	0.000071	0.000151	0.000148	0.0001
HINDUNILVR.NS	0.000078	0.000044	0.000095	0.000205	0.000084	0.000099	0.000059	0.000063	0.000055	0.0000
ICICIBANK.NS	0.000151	0.000067	0.000193	0.000084	0.000504	0.000074	0.000072	0.000172	0.000153	0.0002
NESTLEIND.NS	0.000074	0.000043	0.000082	0.000099	0.000074	0.000225	0.000051	0.000067	0.000014	0.0000
TCS.NS	0.000077	0.000042	0.000071	0.000059	0.000072	0.000051	0.000250	0.000051	0.000033	0.0000
RAMCOCEM.NS	0.000109	0.000049	0.000151	0.000063	0.000172	0.000067	0.000051	0.000371	0.000104	0.0001
NIFTYBEES.NS	0.000060	0.000060	0.000148	0.000055	0.000153	0.000014	0.000033	0.000104	0.007081	0.0071
BANKBEES.NS	0.000065	0.000060	0.000179	0.000058	0.000271	0.000017	0.000022	0.000137	0.007121	0.0072

```
rf = 0.01 # risk factor
optimal_risky_port = portfolios.iloc[((portfolios['Returns']-rf)/portfolios['Volatility']).idxmax()]
optimal_risky_port
```

```
Returns                0.235700
Volatility             0.168492
RELIANCE.NS weight     0.229030
CIPLA.NS weight        0.020617
MARUTI.NS weight       0.052173
HINDUNILVR.NS weight   0.191204
ICICIBANK.NS weight    0.023929
NESTLEIND.NS weight    0.250749
TCS.NS weight          0.175330
RAMCOCEM.NS weight     0.038619
NIFTYBEES.NS weight    0.001862
BANKBEES.NS weight     0.016487
Name: 97415, dtype: float64
```

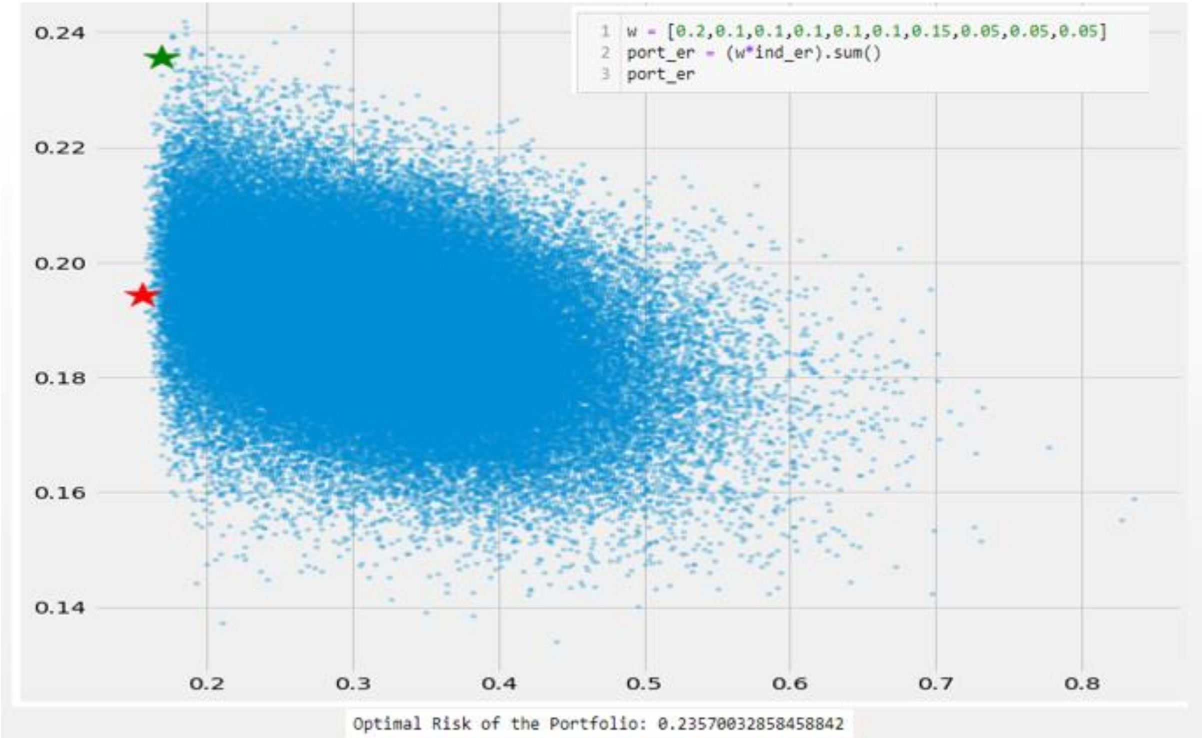


Figure 9.1.a Efficient Frontier Model with different weightage

The weightage given here is Reliance (20%), TCS (15%), Cipla(10%) , Maruti (10%), Hindustan Unilever (10%), ICICI Bank (10%), Nestle Industries(10%), Ramco Cement(5%), Niftybees(5%) and Bankbees(5%). The Red Star represents the minimum volatility and the Green star represents the highest Sharpe Ratio. The optimal risk of the Portfolio is 23.5%.

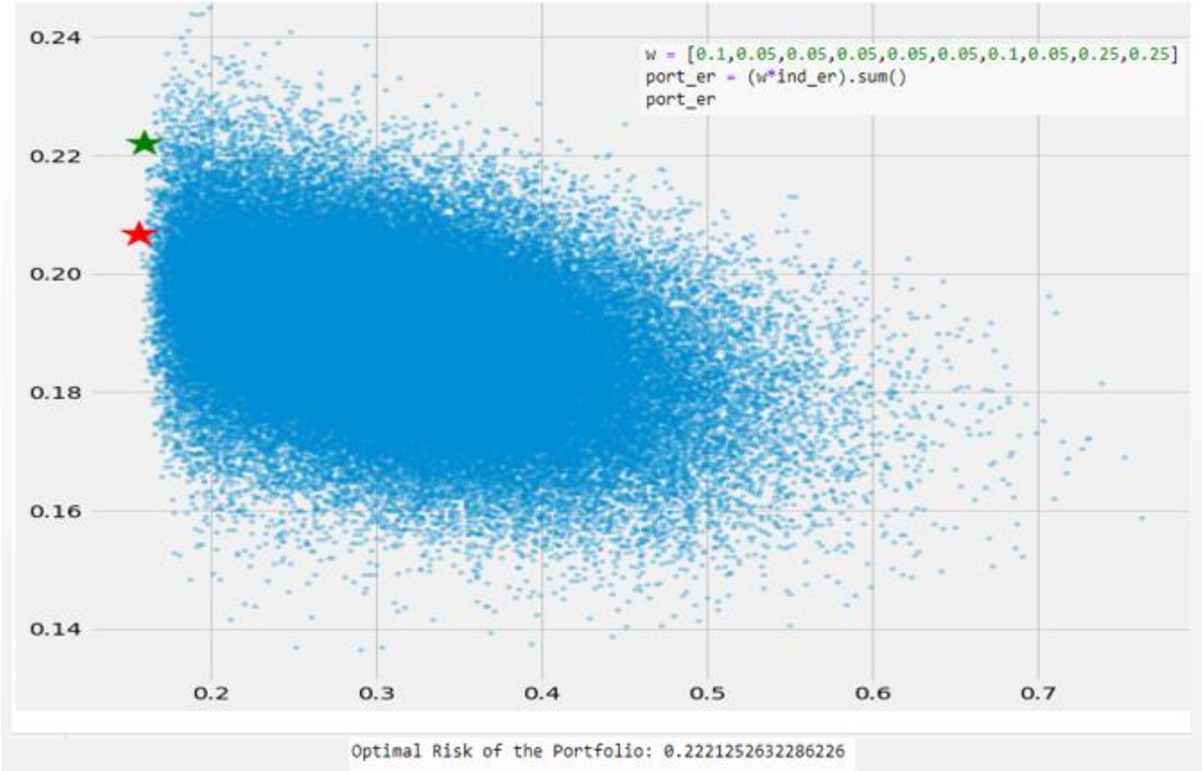


Figure 9.1.b Efficient Frontier Model with different weightage

The weightage given here is Reliance (10%), TCS (10%), Cipla(5%) , Maruti (5%), Hindustan Unilever (5%), ICICI Bank (5%), Nestle Industries(5%), Ramco Cement(5%), Niftybees(25%) and Bankbees(25%). The Red Star represents the minimum volatility and the Green star represents the highest Sharpe Ratio. The optimal risk of the Portfolio is 23.5%.

10. Implementation Details

This chapter describes the details of the implementation for the animal identification part of the system. The tracking of motion and notifications are part the next phase of the project, and are not described here. The software and tools used are described in 10.1.
Different modules used in the project are described in 10.2.

10.1 Software and Tools

The project runs on a Laptop that has an Intel Core i7 Processor. The system has 8GB RAM. The entire project has been developed in Python 3.7. The tools and libraries used in python for various tasks are summarized in Table below.

Task	Library/Framework	Version
Predicting the stock price	Facebook Prophet	v0.7.1
Prediction of stock price	Tensor Flow 2 (Keras)	v2.8.0
Portfolio optimization	PyPortfolioOpt	v1.5.2

Table10.1 Table of Software and Tools.

10.2 Modules

10.2.1 Facebook Prophet

Prophet is a method for forecasting time series data that uses an additive model to accommodate non-linear trends with yearly, weekly, and daily seasonality, as well as holiday impacts. It works well with time series that have substantial seasonal impacts and historical data from multiple seasons. Prophet is resistant to missing data and trend alterations, and it typically handles outliers well. The Core Data Science team of Facebook released Prophet as open source software.

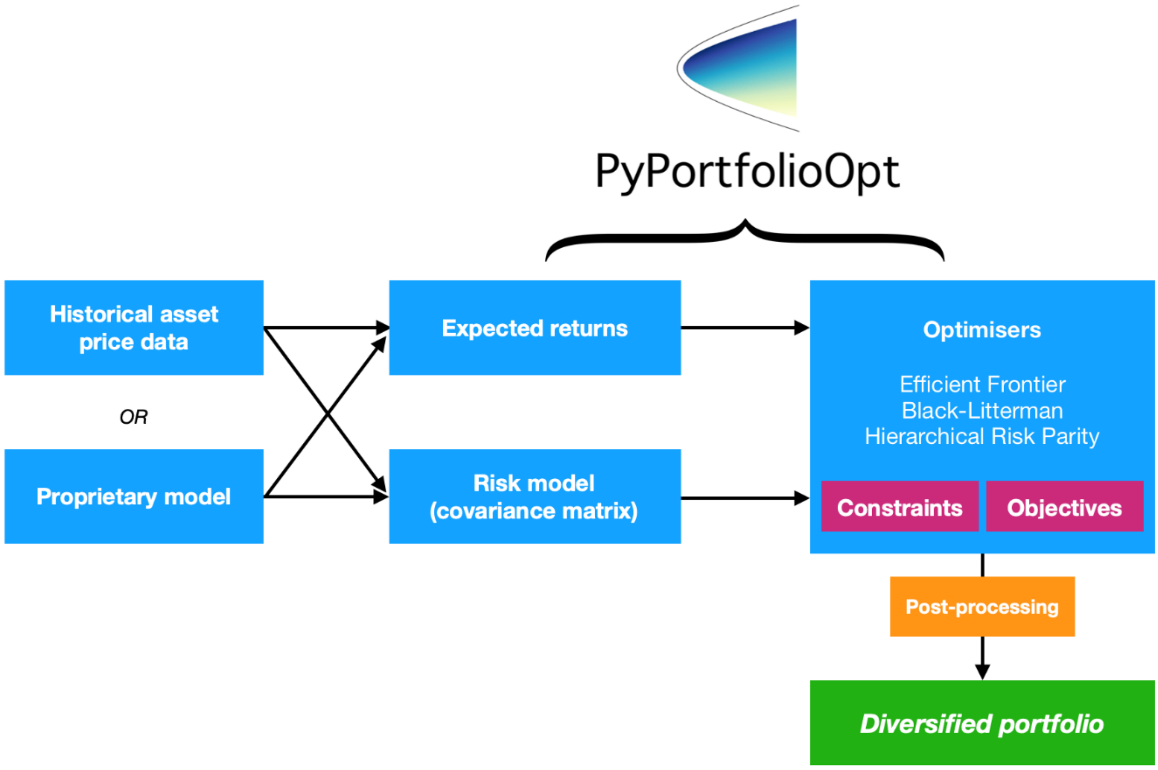
10.2.2 TensorFlow 2 and Keras

TensorFlow 2 is an open-source machine learning platform that runs from start to end. Consider it an infrastructure layer for differentiable programming. It combines four essential abilities:

- Executing low-level tensor operations efficiently on CPU, GPU, or TPU.
- Calculating the gradient of any differentiable expression.
- Scaling computation to a large number of devices, such as hundreds of GPU clusters.
- Exporting programmes ("graphs") to external runtimes like servers, browsers, mobile devices, and embedded devices.

Keras is TensorFlow 2's high-level API: a user-friendly, highly productive interface for addressing machine learning issues, with a focus on current deep learning. It provides fundamental abstractions and building blocks for rapidly designing and shipping machine learning applications.

10.2.3 PyPortfolioOpt



10.2.3 PyPortfolioOpt

PyPortfolioOpt is a library that implements portfolio optimization methods such as efficient frontier techniques and Black-Litterman allocation, as well as more recent advances in the field such as shrinkage and Hierarchical Risk Parity, as well as some novel experimental features such as exponentially-weighted covariance matrices.
It is comprehensive yet easily expandable, and it may benefit both the casual investor and the serious practitioner. PyPortfolioOpt can help you integrate your alpha sources in a risk-efficient manner, whether you are a fundamentals-oriented investor with a handful of undervalued picks or an algorithmic trader with a basket of techniques. [17].

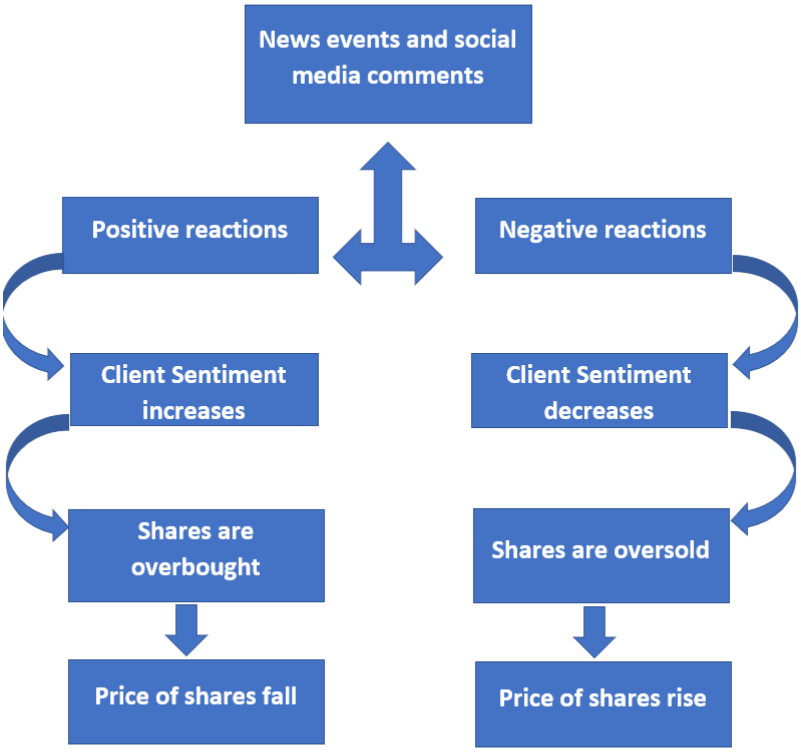
11. Twitter Sentiment Analysis of stocks.

Introduction:

Nowadays, social media has evolved into a mirror that reflects people's ideas and opinions on every given event or topic. Any good or negative public opinion regarding a piece of news the stock values of a specific company can be affected. We attempt to forecast the stock market prices of numerous companies using social media sentiment analysis. Data from the media, such as tweets on the respective companies. The paper describes how Twitter can help you become a better writer by making prudent investment selections with its funds and understanding of market sentiment Tweets gathered with. The Twitter API would be associated with a variety of topics. The primary difficulty in data processing would be sorting through these tweets and filter out the ones that are important to us. [18]

11.1 Problem Statement Data Collection:

The goal of the project is to understand the sentiments of people using the twitter sentiment analysis and categorizing whether the sentiments are positive, negative or neutral and their impact on the stock prices.



11.1 Figure on how twitter sentiment work on stocks.

Here the stock prices have been extracted from Yahoo Finance API and tweets from Twitter API. From Yahoo Finance the historical data of the various stocks have been taken and using tweets I have calculated their sentiments.

Below are some of the Twitter Sentiment Analysis of a few stocks.

11.2 Twitter Sentiment Analysis of Reliance.

	Date	Tweets	Prices
0	2022-05-09	RT EnergyTidbits Elevated LNG prices to contin...	2518
1	2022-05-08	RT BiIndia The oiltotelecom conglomerate comp...	
2	2022-05-07	Reliance Industries Limited contribute 20 of ...	
3	2022-05-06	RT nandtara Modi regularly accuses the UPA go...	2620
4	2022-05-05	Reliance RelianceIndustries reliancebazar rel...	2640
5	2022-05-04	RT BeatTheStreet10 Reliances demerged busines...	2693
6	2022-05-03	RT EuremO8 6 stocks that gave me multibagger ...	
7	2022-05-02	RT mpparimal Shri MukeshAmbani imparts a grea...	2780
8	2022-05-01	RT StocksCall Only Stocks Looking To Buy For ...	
9	2022-04-30	Which district became countrys 1st district w...	

There are a total of 570 tweets with **RelianceIndustries** as the keyword from April 30th 2022 to 9th May 2022. The closing price of the stocks have been shown.

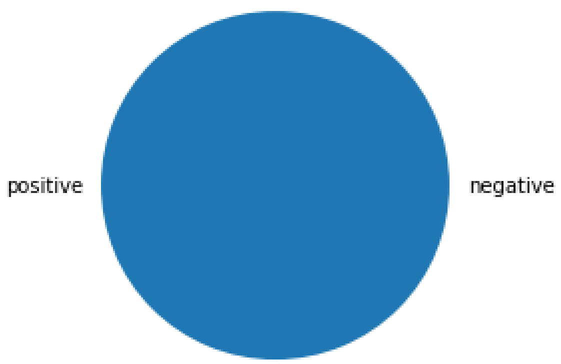
	Date	Tweets	Prices
0	2022-05-09	RT EnergyTidbits Elevated LNG prices to contin...	2518
1	2022-05-08	RT BiIndia The oiltotelecom conglomerate comp...	2650
2	2022-05-07	Reliance Industries Limited contribute 20 of ...	2650
3	2022-05-06	RT nandtara Modi regularly accuses the UPA go...	2620
4	2022-05-05	Reliance RelianceIndustries reliancebazar rel...	2640
5	2022-05-04	RT BeatTheStreet10 Reliances demerged busines...	2693
6	2022-05-03	RT EuremO8 6 stocks that gave me multibagger ...	2650
7	2022-05-02	RT mpparimal Shri MukeshAmbani imparts a grea...	2780
8	2022-05-01	RT StocksCall Only Stocks Looking To Buy For ...	2650
9	2022-04-30	Which district became countrys 1st district w...	2650

For the days which have holidays I have taken Prices as mean.

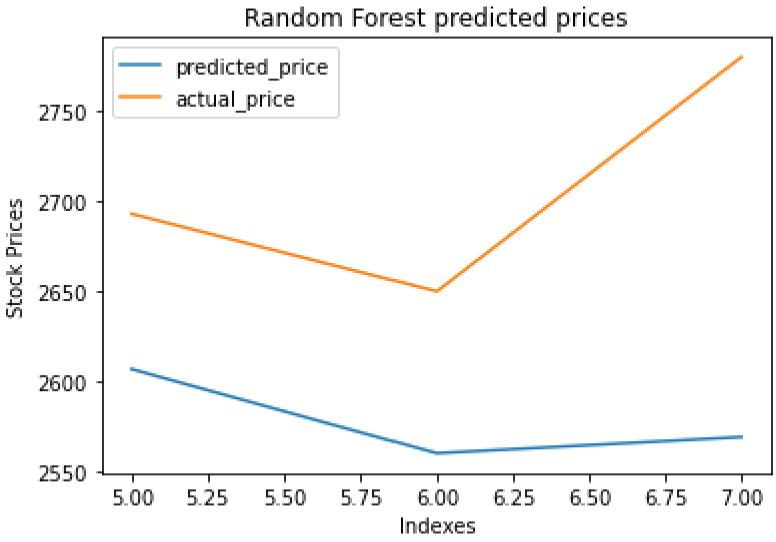
	Date	Tweets	Prices	Comp	Negative	Neutral	Positive
0	2022-05-09	RT EnergyTidbits Elevated LNG prices to contin...	2518	0.9996	0.038	0.866	0.9996
1	2022-05-08	RT BiIndia The oiltotelecom conglomerate comp...	2650	0.9656	0.046	0.886	0.9656
2	2022-05-07	Reliance Industries Limited contribute 20 of ...	2650	0.9959	0.087	0.808	0.9959
3	2022-05-06	RT nandtara Modi regularly accuses the UPA go...	2620	0.9999	0.059	0.824	0.9999
4	2022-05-05	Reliance RelianceIndustries reliancebazar rel...	2640	0.9946	0.032	0.868	0.9946

Here the tweets along with Positive, Negative and Neutral sentiments are shown. Compound tells us whether the overall sentiment is positive or negative.

```
% of positive tweets=  100.0
% of negative tweets=  0.0
[]
```



This is a pie chart of +ve and –ve tweet sentiments. During this period all the tweets were positive in nature.

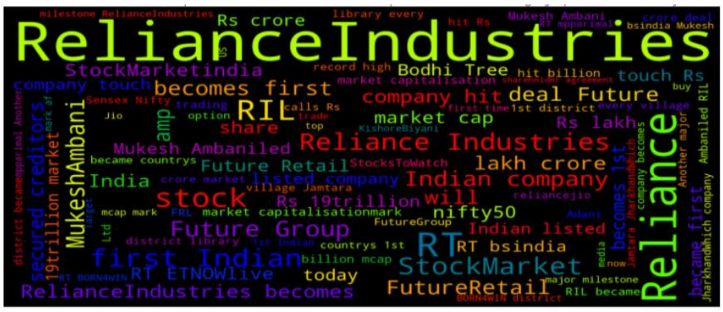


Random Forest of predicted price vs actual price. The first 4 days tweets in the week are taken as training data and the next 3 days are taken as testing data and the RMSE is calculated.

```
from sklearn.metrics import  mean_squared_error
rmse = mean_squared_error(y_test,prediction,squared=False)
print(rmse)
```

```
141.3561502022463
```

Here the RMSE is found to be 141.35615 for reliance stock during this time period.



Word Cloud is a visual representation of words. Most frequently used and popular words have been highlighted here.

11.3 Twitter Sentiment Analysis of Hindustan Unilever.

	Date	Tweets	Prices
0	2022-05-09	RT ETRetail Ecommerce sales gallop despite big...	2116
1	2022-05-08	RT ETRetail HUL expects more sequential infla...	
2	2022-05-07	Consumers are no longer interested in buying ...	
3	2022-05-04	RT ArvindChaturved HindustanUnilevers HULchi...	2171
4	2022-05-03	Adani Wilmar Ltd a packaged goods company tod...	
5	2022-05-02	RT businesstoday Packaged foods major AdaniWi...	2230
6	2022-05-01		
7	2022-04-30	Palm oil Distributors resort to panic buying ...	

Fig 11.2.a Tweets along with their Prices

	Date	Tweets	Prices
0	2022-05-09	RT ETRetail Ecommerce sales gallop despite big...	2116
1	2022-05-08	RT ETRetail HUL expects more sequential infla...	2172
2	2022-05-07	Consumers are no longer interested in buying ...	2172
3	2022-05-04	RT ArvindChaturved HindustanUnilevers HULchi...	2171
4	2022-05-03	Adani Wilmar Ltd a packaged goods company tod...	2172
5	2022-05-02	RT businesstoday Packaged foods major AdaniWi...	2230
6	2022-05-01		2172
7	2022-04-30	Palm oil Distributors resort to panic buying ...	2172

Fig 11.2.b Tweets along with their Prices as mean for days with holidays

	Date	Tweets	Prices	Comp	Negative	Neutral	Positive
0	2022-05-09	RT ETRetail Ecommerce sales gallop despite big...	2116	0.5083	0.077	0.827	0.5083
1	2022-05-08	RT ETRetail HUL expects more sequential infla...	2172	0.2177	0.0	0.896	0.2177
2	2022-05-07	Consumers are no longer interested in buying ...	2172	-0.25	0.101	0.804	-0.25
3	2022-05-04	RT ArvindChaturved HindustanUnilevers HULchi...	2171	0.0516	0.022	0.959	0.0516
4	2022-05-03	Adani Wilmar Ltd a packaged goods company tod...	2172	-0.2441	0.053	0.909	-0.2441

Fig 11.2.c Tweets, Prices along with their sentiments

```
% of positive tweets=  50.0
% of negative tweets=  37.5
[]
```

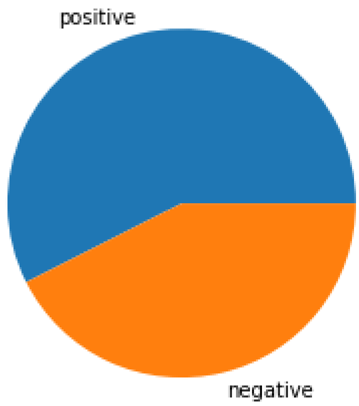


Fig 11.2.d Positive and Negative sentiments of tweets

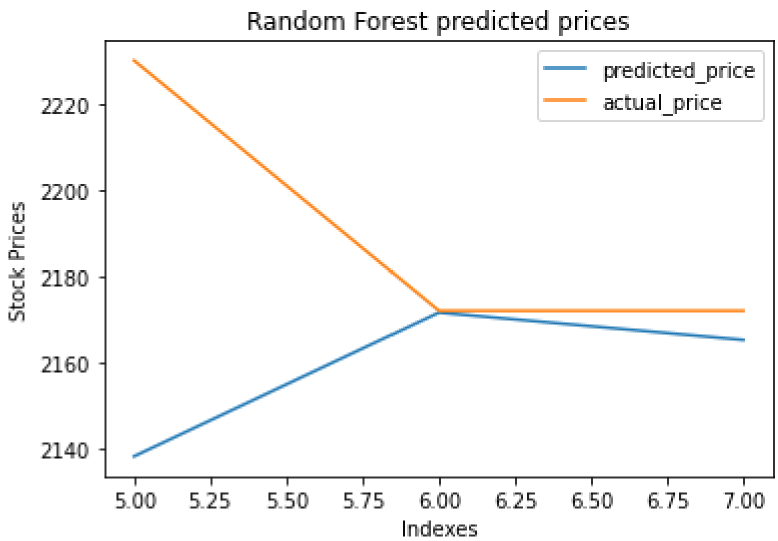


Fig 11.2.a Random Forest (Predicted Price VS Actual Price)

```
from sklearn.metrics import  mean_squared_error
rmse = mean_squared_error(y_test,prediction,squared=False)
print(rmse)
```

```
53.144814736090154
```

Fig 11.2.e RMSE of the dataset

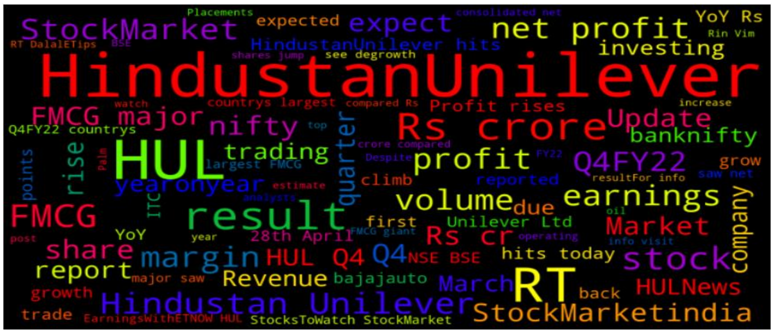


Fig 11.2.f WordCloud

11.4 RMSE values for various stocks using Random Forest.

	RMSE-(Random Forest)
RELIANCE	141.356150
CIPLA	20.471220
MARUTI	59.585130
HUL	53.144810
ICICI BANK	5.823822
NESTLE IND	125.947900
TCS	25.287660
RAMCO CEMENT	9.595165
NIFTYBEES	1.560416
BANKBEES	15.617812

Above are the RMSE values for the various stocks found in the portfolio.

12. Hierarchical Risk Parity

Marcos Lopez de Prado created HRP, an unique portfolio optimization technique (2016). This model is comprised of three steps:

- **Hierarchical Tree Clustering**: We use the correlation between financial assets to generate a hierarchical structure that may be shown as a dendrogram.
- **Matrix Seriation**: we sort the assets in the dendrogram by minimising the distance between leaves; Lopez de Prado referred to this as quasi-diagonalization.
- **Recursive Bisection**: From the top of the tree to the leaves, we split the weights along the dendrogram using naive risk parity (weights based on the inverse of the asset's risk).

Lopez de Prado (2016) proposed hierarchical risk parity (HRP) as a risk parity allocation technique to solve the shortcomings of MVO portfolios. HRP provides a hierarchical implementation of an inverse-variance allocation with weights determined across clusters of correlated asset returns at its core. The strategy takes use of risk parity portfolios' out-of-sample robustness and may be applied to singular covariance matrices, hence overcoming the stability problem. HRP achieves lower out-of-sample volatility and higher risk-adjusted return than inverse-variance allocations, according to Lopez de Prado (2016). Aside from attractive quantitative qualities, the technique offers the intuitive attraction of allowing groupings of linked assets (rather than individual assets) to compete for capital in the portfolio, hence enhancing diversification across risk sources.
Using historical daily returns data, the HRP portfolio is benchmarked against traditional risk-based allocation strategies in a systematic manner. The portfolio backtest investigates the performance of HRP in portfolios with varied degrees of N-diversification, with a focus on Indian equity markets. HRP achieves lower out-of-sample volatility and higher risk-adjusted return than inverse-variance and naive portfolios, and is a close competitor to the minimum-variance portfolio, but without the reliance on a positive-definite covariance matrix. Furthermore, the HRP portfolio has excellent diversification and a low turnover rate.

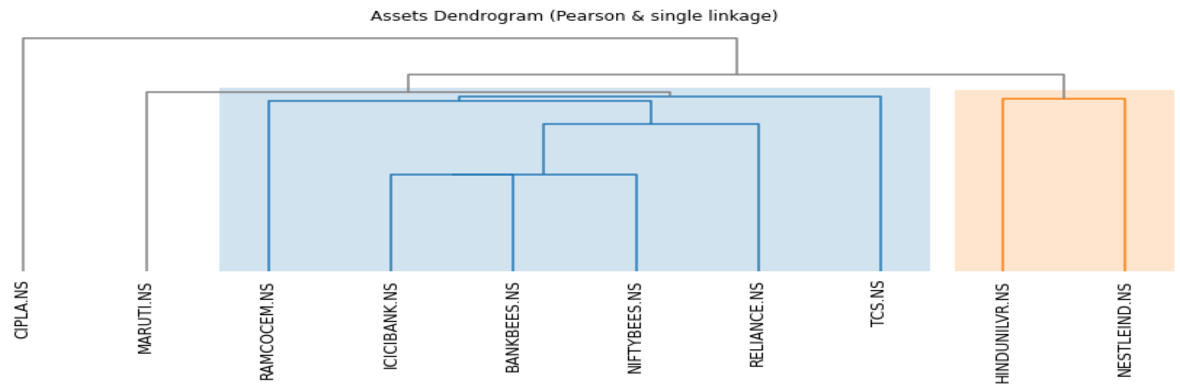
Below is the display of the datasets that is being used: There are 10 stocks in the portfolio and the data has been taken for a period of 1 year with daily returns as shown below.

Date	BANKBEES.NS	CIPLA.NS	HINDUNILVR.NS	ICICIBANK.NS	MARUTI.NS	NESTLEIND.NS	NIFTYBEES.NS	RAMCOCEM.NS	RELIANCE.NS	TCS.NS
2021-01-04	0.1498%	0.6835%	1.6314%	0.7962%	0.1430%	-0.3943%	0.7622%	0.4214%	0.1686%	3.7975%
2021-01-05	1.4676%	-0.6008%	0.9911%	1.0438%	-0.6083%	0.9811%	0.3915%	-0.5574%	-1.2432%	1.7618%
2021-01-06	0.2949%	-0.2962%	-1.3568%	1.7590%	-0.3507%	-0.2317%	-0.0529%	1.0770%	-2.6372%	-1.3417%
2021-01-07	0.4411%	0.2122%	-2.0043%	-1.0243%	-0.8199%	-2.0953%	-0.0132%	3.7074%	-0.1619%	-0.6128%
2021-01-08	0.2305%	1.4700%	0.9435%	0.1756%	5.9324%	0.9872%	1.1111%	2.3612%	1.1799%	2.9049%

```
display(Y.tail())
```

Date	BANKBEES.NS	CIPLA.NS	HINDUNILVR.NS	ICICIBANK.NS	MARUTI.NS	NESTLEIND.NS	NIFTYBEES.NS	RAMCOCEM.NS	RELIANCE.NS	TCS.NS
2022-04-25	0.0138%	-1.4323%	-1.2318%	0.6086%	0.0993%	-0.0718%	-1.1983%	-0.6848%	-2.3126%	-1.7813%
2022-04-26	0.7952%	2.6491%	2.2659%	0.2061%	-0.0708%	1.3263%	1.2238%	-1.3915%	2.9926%	-0.0535%
2022-04-27	-0.8709%	-0.0664%	-0.1513%	-2.2620%	-1.4968%	-1.0467%	-0.7146%	-1.8080%	0.0973%	0.4202%
2022-04-28	1.0052%	0.3529%	4.5103%	1.4185%	1.2808%	1.1233%	0.8821%	1.8092%	1.4937%	0.6487%
2022-04-29	-0.8016%	0.0000%	-0.3122%	-0.5153%	-2.1596%	-0.5285%	-0.5257%	0.1575%	-1.0497%	-1.0490%

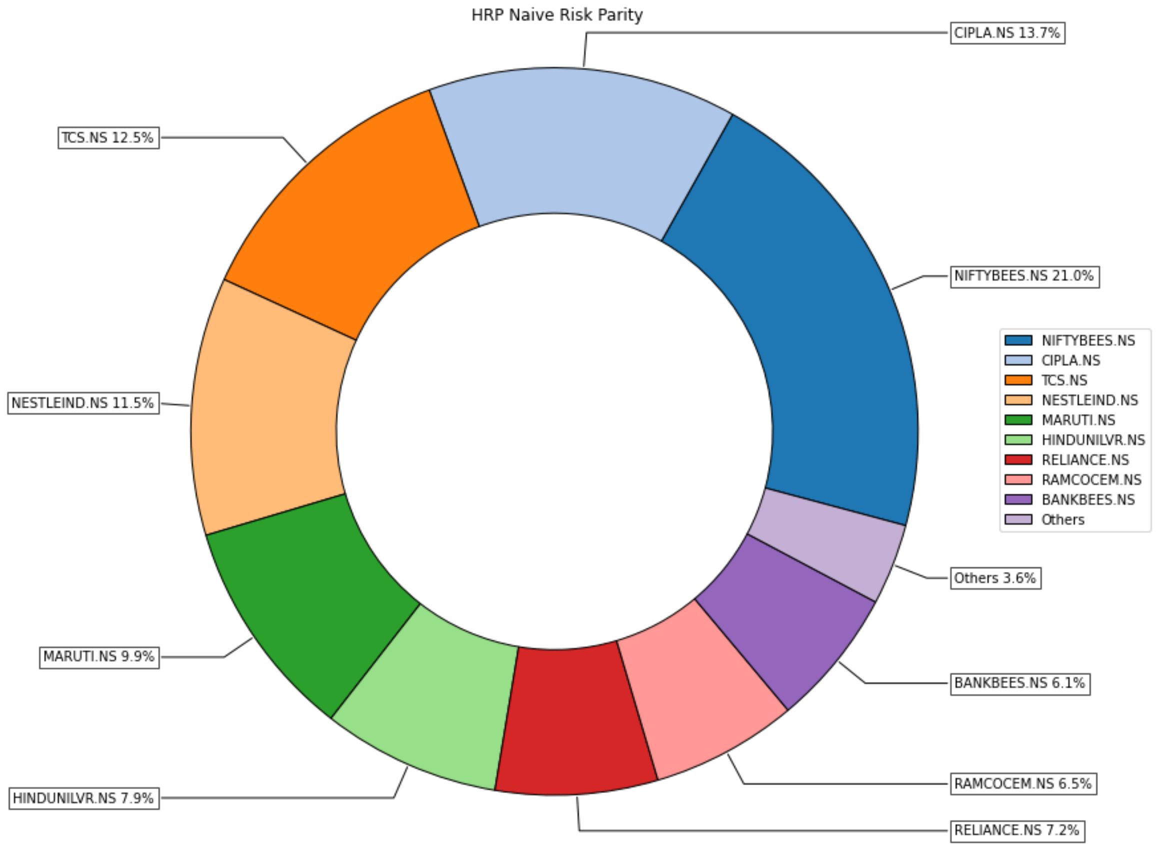
Lopez de Prado (2016) clusters similar assets based on their correlation using a single-linkage (AGNES) hierarchical clustering of the correlation matrix. [19]



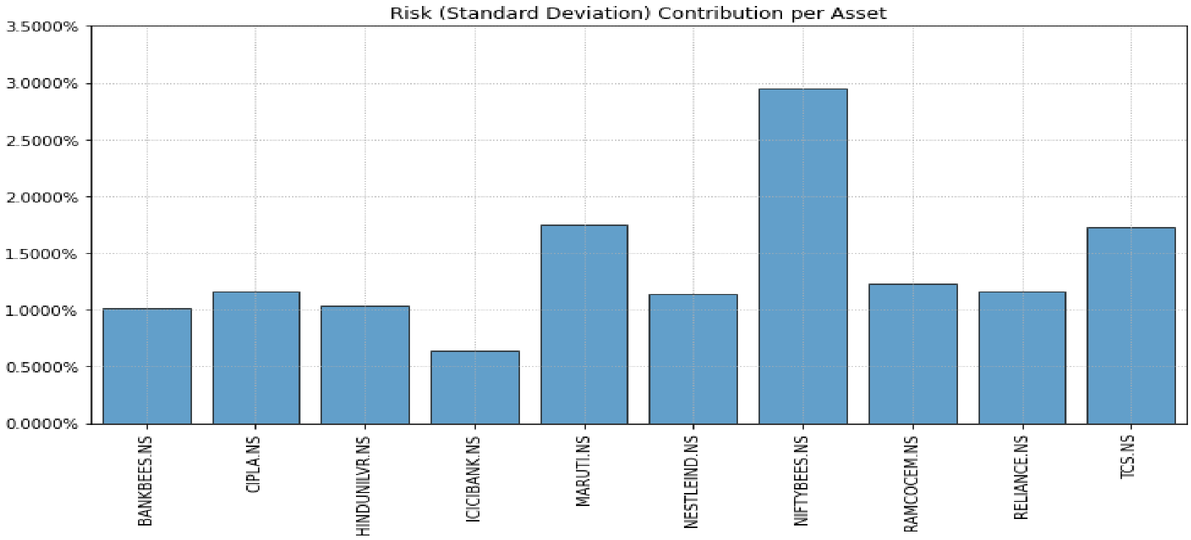
Then we must create the HCPortfolio object, which holds all portfolio models based on a hierarchical link between assets:

	BANKBEES.NS	CIPLA.NS	HINDUNILVR.NS	ICICIBANK.NS	MARUTI.NS	NESTLEIND.NS	NIFTYBEES.NS	RAMCOCEM.NS	RELIANCE.NS	TCS.NS
weights	6.1286%	13.7442%	7.9128%	3.6020%	9.8581%	11.4691%	21.0181%	6.5082%	7.2148%	12.5442%

As you can see, we acquire the best HRP portfolio in a few stages. A pie chart can be used to visualise the portfolio's structure:



This technique creates the most diverse portfolios when compared to traditional portfolio optimization models based on convex programming problems. HRP, on the other hand, creates portfolios that better spread risk across all assets. To see this property, we can use the following code to plot the risk contribution per asset:



Until now, we have worked with variance as a risk measure; however, Riskfolio-Lib has 22 risk measures available for portfolios based on the hierarchical relationship among assets; to compare asset allocation based on the 22 risk measures using HRP.

 # Risk Measures available:

'vol': Standard Deviation.
'MV': Variance.
'MAD': Mean Absolute Deviation.
'MSV': Semi Standard Deviation.
'FLPM': First Lower Partial Moment (Omega Ratio).
'SLPM': Second Lower Partial Moment (Sortino Ratio).
'VaR': Conditional Value at Risk.
'CVaR': Conditional Value at Risk.
'EVaR': Entropic Value at Risk.
'WR': Worst Realization (Minimax)
'MDD': Maximum Drawdown of uncompounded cumulative returns (Calmar Ratio).
'ADD': Average Drawdown of uncompounded cumulative returns.
'DaR': Drawdown at Risk of uncompounded cumulative returns.
'CDaR': Conditional Drawdown at Risk of uncompounded cumulative returns.
'EDaR': Entropic Drawdown at Risk of uncompounded cumulative returns.
'UCI': Ulcer Index of uncompounded cumulative returns.
'MDD_Rel': Maximum Drawdown of compounded cumulative returns (Calmar Ratio).
'ADD_Rel': Average Drawdown of compounded cumulative returns.
'DaR_Rel': Drawdown at Risk of compounded cumulative returns.
'CDaR_Rel': Conditional Drawdown at Risk of compounded cumulative returns.
'EDaR_Rel': Entropic Drawdown at Risk of compounded cumulative returns.
'UCI_Rel': Ulcer Index of compounded cumulative returns.

	vol	MV	FLPM	SLPM	CVaR	WR	MDD	ADD	DaR	CDaR	EDaR	UCI	MDD_Rel	ADD_Rel	DaR_Rel
BANKBEES.NS	6.29%	6.13%	6.91%	6.39%	5.44%	4.77%	5.66%	5.79%	6.34%	6.37%	6.02%	5.94%	5.63%	5.72%	6.17%
CIPLA.NS	13.34%	13.74%	13.20%	13.47%	14.36%	15.20%	21.20%	20.55%	21.11%	21.18%	21.15%	21.06%	20.16%	20.49%	20.59%
HINDUNILVR.NS	7.29%	7.91%	6.75%	7.31%	7.84%	10.14%	3.52%	3.28%	2.92%	2.93%	3.20%	3.20%	3.81%	3.51%	3.12%
ICICIBANK.NS	4.82%	3.60%	5.53%	5.35%	4.84%	4.98%	5.00%	5.90%	6.03%	5.84%	5.41%	5.74%	5.00%	5.81%	5.84%
MARUTI.NS	11.30%	9.86%	11.18%	10.82%	10.48%	10.23%	10.29%	8.49%	10.04%	10.33%	10.38%	9.38%	10.67%	8.67%	10.33%
NESTLEIND.NS	8.78%	11.47%	8.00%	9.04%	10.77%	12.40%	6.93%	5.44%	5.91%	6.05%	6.54%	5.88%	6.79%	5.38%	5.86%
NIFTYBEES.NS	16.34%	21.02%	16.64%	15.87%	15.02%	13.57%	15.49%	22.18%	18.56%	16.91%	16.18%	20.04%	15.59%	21.71%	18.34%
RAMCOCEM.NS	9.24%	6.51%	9.19%	9.09%	8.40%	8.28%	5.37%	6.53%	5.12%	5.44%	5.44%	6.14%	5.88%	6.82%	5.47%
RELIANCE.NS	9.58%	7.21%	9.68%	9.85%	10.04%	9.26%	11.52%	9.68%	10.40%	10.77%	10.91%	9.91%	11.46%	9.63%	10.66%
TCS.NS	13.02%	12.54%	12.93%	12.81%	12.80%	11.18%	15.02%	12.16%	13.56%	14.18%	14.79%	12.70%	15.01%	12.26%	13.61%

13.Conclusion and Future Work

This work provides an opportunity to explore many new developments in the field of Deep Learning. Because the size of data constrains the use of neural networks, continuing research in this subject focuses on finding effective strategies to speed up the training of big data sets. Many studies have been performed to make the deployment of enormous data easier because greater quantities of data are known to be computationally costly. Recently, research has focused on the use of sparsity to compress Deep Neural Networks and therefore minimise processing costs.

Predicting stock market returns is a challenging task due to consistently changing stock values which are dependent on multiple parameters which form complex patterns. The historical dataset available on company's website consists of only few features like high, low, open, close, adjacent close value of stock prices, volume of shares traded etc., which are not sufficient enough. Also I have been primarily able to access data through Yahoo Finance or through the NSE. Many a times, missing data results in a lower accuracy of the model and the predicted outcome. Also stock price movement in the market primarily is of three types i.e. Fundamental Factor, Technical Factor and Sentiment Factor. With the advancement in the technology social media impacts on the stock prices have been quite noticeable in recent times. Further study can be done by thorough analysis of external factors like geopolitical news, FED rate hikes etc to get a clear picture of the stock price movement.

References:

[1]R. Ji, K. Chang and Z. Jiang, "Risk-Aversion Adjusted Portfolio Optimization with Predictive Modelling," 2019 22th International Conference on Information Fusion (FUSION), 2019, pp. 1-8.

[2] Ethem Alpaydın, Introduction to Machine Learning, October 1, 2004

[3] Harry M Markowitz, G Peter Todd, and William F Sharpe Mean-variance analysis in portfolio choice and capital markets Volume 66.

[4]J. Sen, S. Mondal and S. Mehtab, "Portfolio Optimization on NIFTY Thematic Sector Stocks Using an LSTM Model," 2021 International Conference on Data Analytics for Business and Industry (ICDABI), 2021, pp. 364-369, doi: 10.1109/ICDABI53623.2021.9655886.

[5]Li, Lei, Yabin Wu, Yihang Ou, Qi Li, Yanquan Zhou, and Daoxin Chen. (2017) “Research on machine learning algorithms and feature extraction for time series.” IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC): 1-5.

[13] Eugene F. Fama, Efficient Capital Markets: II,The Journal Of Finance, Vol XLVI No 5, December 1991

[6] Mehar Vijh, Deeksha Chandola, Vinay Anand Tikkiwal, Arun Kumar,Stock Closing Price Prediction using Machine Learning Techniques,Procedia Computer Science,Volume 167,

[7] ‘A Machine Learning Approach to Stock Price Prediction and Portfolio Optimization’ Jorge Barreno, Anna Skarpalezou, Arjun Tisseverasinghe.

[8] V. Kedia, Z. Khalid, S. Goswami, N. Sharma and K. Suryawanshi, "Portfolio Generation for Indian Stock Markets Using Unsupervised Machine Learning," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018, pp. 1-5, doi: 10.1109/ICCUBEA.2018.8697771.

[9] J. Sen, S. Mondal and S. Mehtab, "Portfolio Optimization on NIFTY Thematic Sector Stocks Using an LSTM Model," 2021 International Conference on Data Analytics for Business and Industry (ICDABI), 2021, pp. 364-369, doi: 10.1109/ICDABI53623.2021.9655886.

[10] J. Sen, S. Mehtab, A. Dutta and S. Mondal, "Hierarchical Risk Parity and Minimum Variance Portfolio Design on NIFTY 50 Stocks," 2021 International Conference on Decision Aid Sciences and Application (DASA), 2021, pp. 668-675, doi: 10.1109/DASA53625.2021.9681925.

[11] L. Troiano, A. Bhandari, and E. M. Villa. “Hands-On Deep Learning for Finance,” Birmingham, UK: Packt, 2020.

[12] Gujarati, N.D. & Porter, D.C. (2008). Basic Econometrics. 5. ed. New York: McGraw-Hill/Irwin.

[13]. Efficient Capital Markets: A Review of Theory and Empirical Work.

[14]. L. Troiano, A. Bhandari, and E. M. Villa. “Hands-On Deep Learning for Finance,” Birmingham, UK: Packt, 2020.

[15]. J. Heaton and N. Polson, “Deep Learning for Finance: Deep Portfolios,” SSRN Electronic Journal, 2016.

[16]. https://www.investopedia.com/terms/s/sharperatio.asp.

[17] https://pyportfolioopt.readthedocs.io/en/latest/.

[18]. T. Mankar, T. Hotchandani, M. Madhwani, A. Chidrawar and C. S. Lifna, "Stock Market Prediction based on Social Sentiments using Machine Learning," 2018 International Conference on Smart City and Emerging Technology (ICSCET), 2018, pp. 1-3, doi: 10.1109/ICSCET.2018.8537242.

[19]. @orenji.eirl/hierarchical-risk-parity-with-python-and-riskfolio-lib-c0e60b94252e.