2020sc04239.pdf

by

Submission date: 20-Mar-2023 03:44PM (UTC+0530)
Submission ID: 2041577378
File name: 2020sc04239.pdf (1.65M)
Word count: 4868
Character count: 25175

A

Report On

Stock gain forecasting for day opening using ML and NLP Dissertation

DSECLZG628T: Dissertation

by

Sarvsav Sharma

2020SC04239

Dissertation work carried out at

Tata Consultancy Services Pvt. Ltd.

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE PILANI (RAJASTHAN)

A Report

On

Stock gain forecasting for day opening using ML and NLP Dissertation

Submitted in partial fulfillment of the requirements of the M.Tech Data Science and Engineering programme

By

Sarvsav Sharma
2020SC04239

Under the supervision of

Amit Sharma – Assistant Consultant
Tata Consultancy Services

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
Pilani (Rajasthan) INDIA

ACKNOWLEDGEMENTS

This project has been possible because of many individuals' consistent assistance and support. I would like to express my gratitude to them for all of their support.

Primary thanks to my supervisor, **Amit Sharma**, for his guidance throughout this project. Our meetings were energy filled and productive.

Vinaya Sathyanarayana, for giving me continuous feedback throughout the project stages, and helped me to understand the business value of the project.

I value my friends' helpfulness and upbeat attitudes; their motivation and support have been essential to my achievement.

Finally, I would like to offer my sincere gratitude to the Divine and my cherished family for their unfailing support and inspiration while I completed this endeavor.

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI FIRST SEMESTER 2022-23
DSECLZG628T **DISSERTATION**

Dissertation Title:	Stock gain forecasting for day opening using ML and NLP by suggesting to buy or sell the quantities
Name of Supervisor:	Amit Sharma
Name of Student:	Sarvsav Sharma
ID No. of Student:	2020SC04239

Key Words:
Stocks gain, loss recovery, machine learning for stocks, mean squared error, easy trading for individuals, stock grouping for gains

Abstract:
The prediction of stock market price trends is a constantly popular topic for academics in both the financial and technical professions since it is one of the most crucial areas of concentration for traders to generate rapid money.
With an emphasis on short-term price trend forecasting in the intraday market, the aim of this study is to construct a cutting-edge prediction model for price trend forecasting. Traders can decide how much stock they can trade for the least amount of loss based on the results.
The present methods for identifying the optimal movement produce poor stock predictions because they either model numerical data or statistical data. Consequently, it's important to take into account both elements in order to get a more accurate outcome.
A machine learning model that estimates the appropriate quantity of stocks to purchase or sell depending on trader risk level in order to achieve maximum gain or minimal loss is trained to handle this issue. In order to make predictions with a higher degree of accuracy, this algorithm scrapes real-time news and shares price information from the internet.
Due to the project's interactive flow, even a person with little computer experience can readily use it. The information for the news and stock prices is gathered using python requests and nsetools from reputable online sources like economonictimes. By putting the program on the cloud, the scraping may be done continuously.

A dataset with 1M entries would be an optimal size for a training model. The column names have been updated, filtered, cleared of outliers and null values, and scaled appropriately for a better result. The sentiment analyzer Vader is used to convert the textual data since it makes it possible to determine not just the positive and negative score but also the compound score and neutral score.

To gather data on model improvement, feature selection uses bar graphs, pair plots, line plots, and heat maps.

Several ML algorithms are applied to the dataset, and even the results are calculated using the auto h2o model that is capable of finding the best model. The model with highest accuracy is the SVR poly for the current dataset.

The best performing model is selected and saved as the news prediction model.pkl file for deployment based on the performance of multiple models. The stored model is also utilized in the Python project to forecast the optimal number of stocks that a trader should purchase or sell in order to realize the most profits.

The project stands out because it takes factors like platform fees and risk level and trains on both textual and numerical data in order to provide greater insights for traders regarding market gains.

(Signature of Student)
Name: Sarvsav Sharma
Date: 05/03/2023
Place: Gurgaon

(Signature of Supervisor)
Name: Amit Sharma
Date: 05/03/2023
Place: New Delhi

Problem statement:

Stock gain forecasting for day opening using ML and NLP by suggesting to buy or sell the quantities

Completeness criteria:

Project should be able to predict the quantity of stocks that a trader can buy or sell with maximum gain and less risk factors.

Best performance:

Model is performing better than the previous model in our organization and it performs better than the h2o model that compares various results and chooses the best model among them.

Success criteria:

Model is able to successfully predict the outcome for buying or selling the quantity of stocks for maximum gains.

Challenges Faced:

1. Collection of headlines data has some characters outside of ascii text and needs to remove them manually.
2. Handling of sentiment or news score for the weekend data, as there are no movements in stock prices because of the market closed.
3. Timeout issue with the python nsetools, and an active machine that runs every minute to collect the data. A cron job is suitable to run on the cloud with minimal resources and is best to collect the real time data for the stocks.
4. Data web scraping, or creating a universal web scraping code that works for all years for economictimes website to collect the latest headline, as they have different templates for the five years old pages.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

CERTIFICATE

This is to certify that the Dissertation entitled **Stock gain forecasting for day opening using ML and NLP** and submitted by Mr **Sarvsav Sharma** having IDNo. **2020SC04239** in partial fulfillment of the requirements of Dissertation, embodies the work done by him under my supervision.

Signature of the Supervisor
Amit Sharma
Assistant Consultant
Tata Consultancy Services Pvt. Ltd.

Place: Gurgaon
Date:05/03/2023

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

The value of stocks always fluctuates dramatically over time, making stock market prediction an intriguing study subject. Investors do two different forms of study before buying a stock. The fundamental analysis is one of the first approaches and very common. In order to determine whether to invest or not, investors consider factors such as the intrinsic worth of the stocks, the state of the market and economy, the political environment, etc. On the other side, technical analysis evaluates equities by looking at data produced by market activity, such as previous prices and volumes. Typically of attempting to determine a security's fundamental worth, technical analysts instead utilize stock charts to spot patterns and trends that could predict how a stock will act in the future.

Many methods for predicting stock movements have been developed throughout the years. Initially, stock trend predictions were made using traditional regression techniques. Non-linear machine learning methods have also been applied since stock data may be characterized as non-stationary time series data.

With a forget gate present, the LINEAR, POLY is similar to a long short-term memory (SVM), however it has fewer parameters than the SVM since it lacks an output gate. The vanishing gradient issue that arises when using a conventional scaler is addressed with LINEAR, POLY. The time sequence is erratic and disordered. Most of the forecasting model that uncovers the complex connection between financial information about an industry and its stock price is beneficial. The financial news in addition to the existing records concerning the firm is used to forecast future stock prices.

Semantic and linguistic traits may be extracted using a variety of ways. The following are a few of them: OpinionFinder, SentiWordNet, Linguistic Inquiry and Word Count (LIWC), Google Profile of Mood States (GPOMS), R sentiment analysis, and Python NLP package. In this approach, the sentimental score is also calculated based on news headlines, in addition to the statistical data for the model to produce more reliable results.

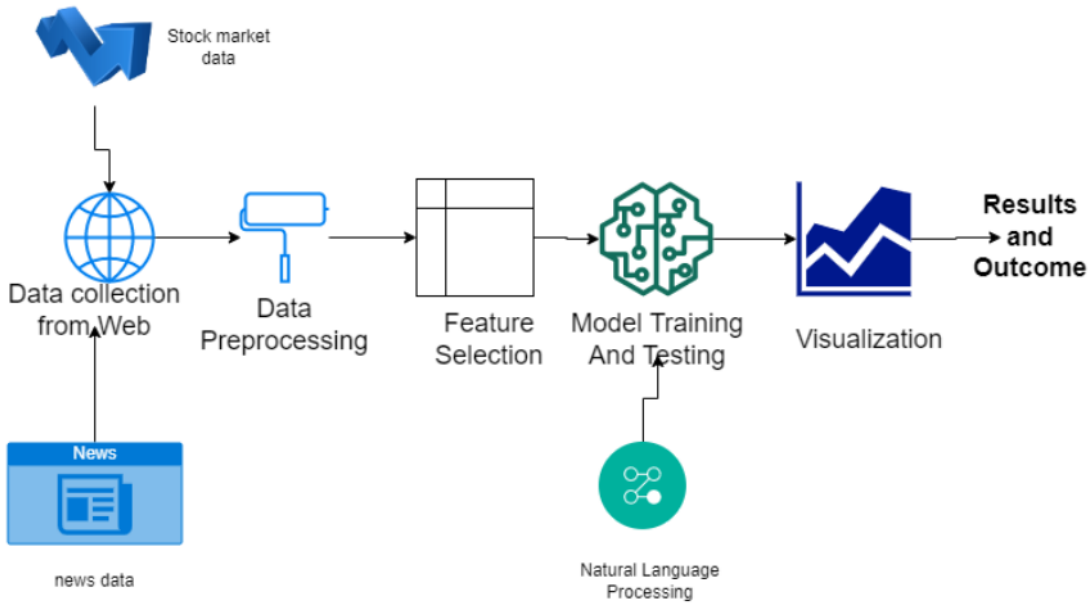


Figure 1: Approach for calculating gains in stock market using machine learning and natural language processing

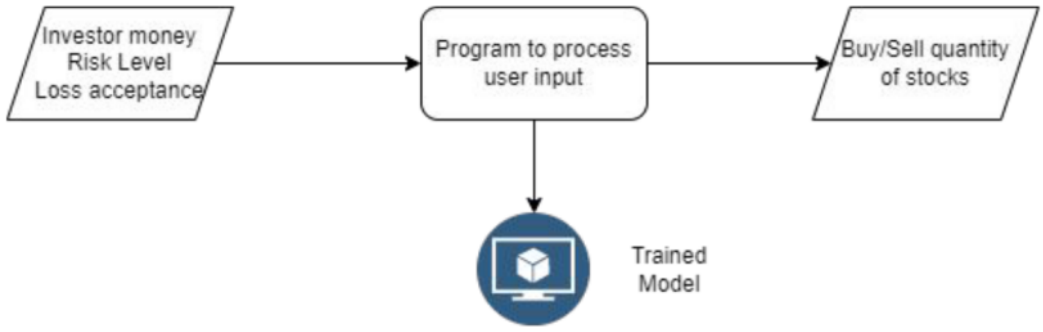


Figure 2: Data flow diagram for the project

Chapter 2: Scope of project

Goal of the project is to help the traders to buy or sell the correct quantity of shares to obtain maximum gains for a day and recover their losses. As a human, whenever we notice a small increase in price of stock from the buying rate, we sell it and do not capitalize the maximum gain that we can have from the share at the end of the day. And, the same is valid for selling also, as we keep believing that the stock will grow up and at the end of day it results in a loss.

The project can be used by individual traders who are investing money and small scale trading companies to help their customers in wise trading.

There are many models available on the internet that predict the value for the upcoming stock, however they are difficult to use and, on the other hand, don't guide you when not to invest. The existing models either predict whether the stock price will go up or down but don't advise you on the number of stocks that you can buy for minimum loss and maximum profit.

The project is open for anyone and recommended to use for the new traders who prefer to trade wisely with minimal risks.

The model has trained for a very limited number of stocks or popular companies like HDFC bank. To be used on a huge scale, the model needs to be trained for a bigger set of companies. It is limited in scope of predicting the results for the set of trained companies.

The inputs for the project are the name of the stock in which the trader would prefer to invest, the risk level on a scale from one to five considered higher, the number is higher the risk, the platform fees, and the acceptance loss. Once the inputs are provided, the program will decide whether the user needs to buy, sell, or quit for today based on the news and other statistical parameters. The output will be the quantity of stocks that a user can buy to have maximum gain at current time and price.

The output result is very understandable as it just prints the quantity of stocks to buy, sell, or not to invest based on the user risk level.

The small scale companies helping the traders can use it to track the status for multiple stocks from different companies on which the model has been already trained, and the individual traders can safely focus on one company or stock to trade for maximum profit.

Chapter 3: Literature Survey

Many investing decisions are influenced by financial markets worldwide. As a result of the geopolitical, social, and economic developments occurring throughout the world, stock markets experience significant shifts over time.

The majority of studies employed quantitative information, such as history or current prices, as predictor variables to forecast the current stock price. The utilization of the massive amounts of unstructured textual data created by the web in the form of news stories published on websites, user opinions on social media, and blogs written by professionals in the field of financial investments received less attention.

The investors and the financial institutions are the stakeholders and they in turn cause financial risks in investments. As a result, academics began examining the causal connection between different market conditions and the related changes in stock values.

Several machine learning models were compared while forecasting the price direction of companies and indices on the Indian stock market in a work by Patel et al. [7]. They were naive Bayes, random forest, support vector machine, and artificial neural network. Ten technical metrics based on open, high, low, and close prices were utilized as input data, expressed as continuous values between -1 and 1. The results showed that random forest performed better than the other three models overall, with an accuracy rate of 83.56%, while naive-Bayes performed the worst, with an accuracy rate of 73.3 %.

Now, there are different approaches to investing in stock like buy, sell, and hold. Nelson, Pereira, and Oliveira [8] investigated the effectiveness of LSTM networks for forecasting changes in stock price. By analyzing performance using the criteria accuracy, precision, recall, and F-measure, the outcomes were compared to other investment methods and machine learning models. Multi-layer perceptron, random for- est, and a pseudo-random model were the models selected for comparison. Additionally, they evaluated the success of buy-and-hold strategies and an optimistic strategy in which stocks were acquired if their prices had increased in the previous time step and sold in the next time step. The period of time selected was 15 minutes. The researchers' suggested LSTM model surpassed the baseline comparisons, averaging 55.9% accuracy on average.

Chapter 4: Data Collection

Information gathering is the project's most vital stage. Nsetools is only one of the numerous Python modules available for real-time data collection. To get headline data, you may scrape websites like Economic Times. The information about data gathering is provided in this section. While historical stock data may be retrieved from the internet, real-time news headline data is collected using a web scraper from the website economictimes.com.

In both the instance of the market data and the news data, the data is quite information-rich. There are several elements that might affect the stock value and may not be covered by the news, therefore it is not advisable to rely just on one news source. And the top six factors mentioned below that have the potential to influence stock prices

- Supply and demand
- business metrics
- details about the industry and advances
- broader market trends
- Geographical context
- The economy's interest rates
- Investor mindset

Table 1: Factors influencing stock price

The most popular Python web scraping packages or frameworks are Beautiful Soup, Scrappy, and requests. requests mainly utilized for websites with java script. As a consequence, data extraction is carried out using a Python script.

a. Data Extraction

The stock website's data is extracted using a Python script and nsetools, and the result is a CSV file. Additionally, economictimes.com is utilized to read archived information as a web page for news data, and the beautifulsoup library of Python is used to read the HTML information. The top news stories for the day were then obtained using the regex. The script has been adjusted to operate with the website's most recent ten years' worth of data.

The features and their specifications are covered in this document. It is now essential to choose the characteristics from the stock market data and important phrases like vaccination

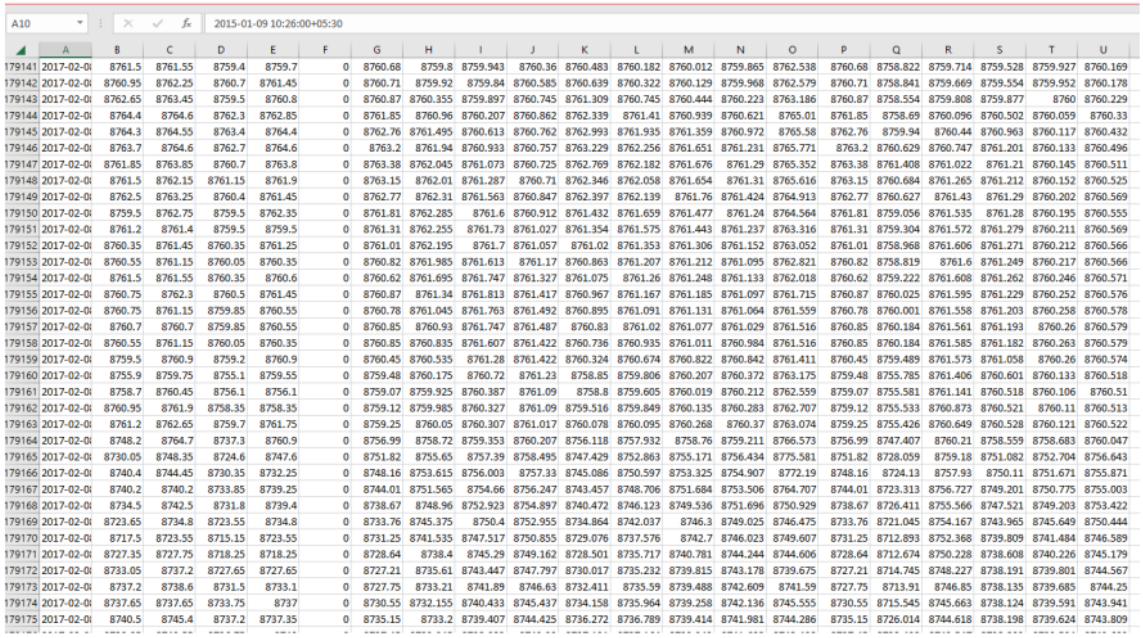
that have an influence on the stock market. Due to the output obtained from the website's collection of several needed variables and the possibility of trivial information, only the following feature is taken into account.

- Date - date of observation
- Open - opening value for the index on that date
- Close - closing value for the index on that date
- High - highest value achieved by the index on that date
- Low - lowest value by the index on that date

This information is gathered throughout the last 5 years and that can help to find the gains for nifty50 stock. Data with all the features manually collected for the news and script is ready to collect the data from nse (national stock exchange).

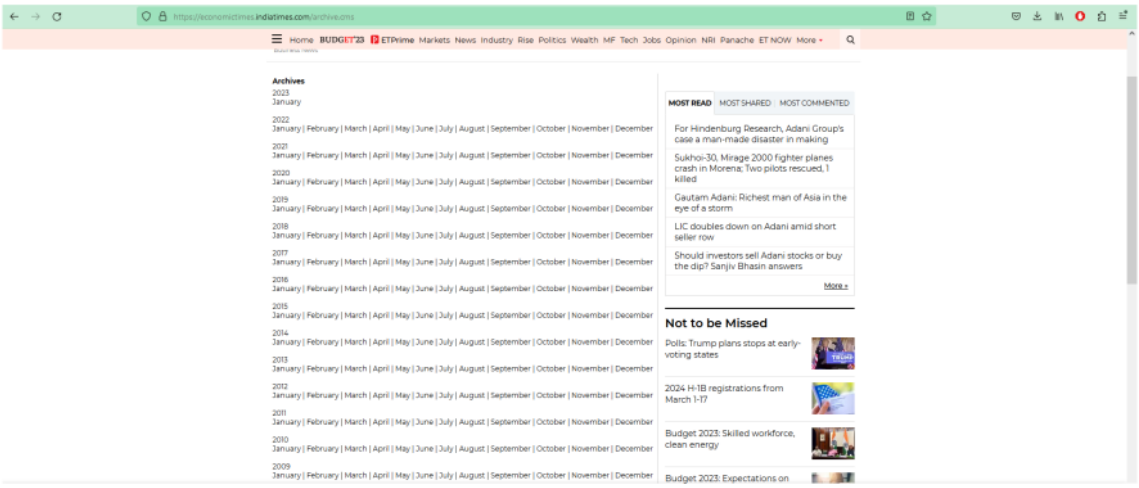
```
>>> q = nse.get_quote('nifty50') # it's ok to use both upper or lower case for codes.
>>> from pprint import pprint # just for neatness of display
>>> pprint(q)
{'adhocMargin': None,
 'applicableMargin': 12.5,
 'averagePrice': 1999.82,
 'bcEndDate': None,
 'bcStartDate': None,
 'buyPrice1': 1999.45,
 'buyPrice2': 1999.4,
 'buyPrice3': 1999.35,
 'buyPrice4': 1999.15,
 'buyPrice5': 1999.1,
 'buyQuantity1': 50.0,
 'buyQuantity2': 209.0,
 'buyQuantity3': 22.0,
 'buyQuantity4': 1.0,
 'buyQuantity5': 24.0,
 'change': 25.35,
 'closePrice': None,
 'cm_adj_high_dt': '01-DEC-14',
 'cm_adj_low_dt': '30-MAY-14',
 'cm_ffm': 190659.16,
 'companyName': 'Infosys Limited',
 'css_status_desc': 'Listed',


'dayHigh': 2010.0,
'dayLow': 1972.0,
'deliveryQuantity': 258080.0,
'deliveryToTradedQuantity': 51.54,
'exDate': '02-DEC-14',
'extremeLossMargin': 5.0,
'faceValue': 5.0,
'high52': 2201.1,
'indexVar': None,
'totalTradedValue': 22914.16,
'totalTradedVolume': 1145811.0,
'varMargin': 7.5}
>>>
```



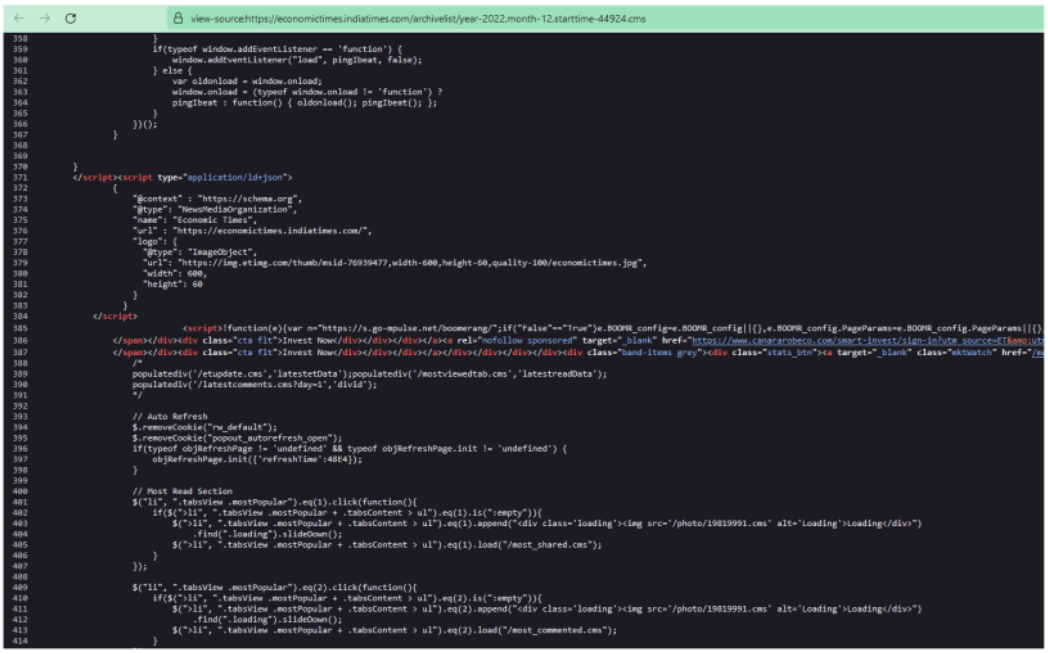
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
179141	2017-02-0	8761.5	8761.55	8759.4	8759.7	0	8760.68	8759.8	8759.943	8760.36	8760.483	8760.182	8760.012	8759.865	8762.538	8760.68	8758.822	8759.714	8759.528	8759.927	8760.169
179142	2017-02-0	8760.95	8762.25	8760.7	8761.45	0	8760.71	8759.92	8759.84	8760.585	8760.639	8760.322	8760.129	8759.968	8762.579	8760.71	8758.841	8759.669	8759.554	8759.952	8760.178
179143	2017-02-0	8762.65	8763.45	8759.5	8760.8	0	8760.87	8760.355	8759.897	8760.745	8761.309	8760.745	8760.444	8760.223	8763.186	8760.87	8758.554	8759.808	8759.877	8760	8760.229
179144	2017-02-0	8764.4	8764.6	8762.3	8762.85	0	8761.85	8760.96	8760.207	8760.862	8762.339	8761.41	8760.939	8760.621	8765.01	8761.85	8758.69	8760.096	8760.502	8760.059	8760.33
179145	2017-02-0	8764.3	8764.55	8763.4	8764.4	0	8762.76	8761.495	8760.613	8760.762	8762.993	8761.935	8761.359	8760.972	8765.58	8762.76	8759.94	8760.44	8760.963	8760.117	8760.432
179146	2017-02-0	8763.7	8764.6	8762.7	8764.6	0	8763.2	8761.94	8760.933	8760.757	8763.229	8762.256	8761.651	8761.231	8765.771	8763.2	8760.629	8760.747	8761.201	8760.133	8760.496
179147	2017-02-0	8761.85	8763.85	8760.7	8763.8	0	8763.38	8762.045	8761.073	8760.725	8762.769	8762.182	8761.676	8761.29	8765.352	8763.38	8761.408	8761.022	8761.21	8760.145	8760.511
179148	2017-02-0	8761.5	8762.15	8761.15	8761.9	0	8763.15	8762.01	8761.287	8760.71	8762.346	8762.058	8761.654	8761.31	8765.616	8763.15	8760.684	8761.265	8761.212	8760.152	8760.525
179149	2017-02-0	8762.5	8763.25	8760.4	8761.45	0	8762.77	8762.31	8761.563	8760.847	8762.397	8762.139	8761.76	8761.424	8764.913	8762.77	8760.627	8761.43	8761.29	8760.202	8760.569
179150	2017-02-0	8759.5	8762.75	8759.5	8762.35	0	8761.81	8762.285	8761.6	8760.912	8761.432	8761.659	8761.477	8761.24	8764.564	8761.81	8759.056	8761.535	8761.28	8760.195	8760.555
179151	2017-02-0	8761.2	8761.4	8759.5	8759.5	0	8761.31	8762.255	8761.73	8761.027	8761.354	8761.575	8761.443	8761.237	8763.316	8761.31	8759.304	8761.572	8761.279	8760.211	8760.569
179152	2017-02-0	8760.35	8761.45	8760.35	8761.25	0	8761.01	8762.195	8761.7	8761.057	8761.02	8761.353	8761.306	8761.152	8763.052	8761.01	8758.968	8761.606	8761.271	8760.212	8760.566
179153	2017-02-0	8760.55	8761.15	8760.05	8760.35	0	8760.82	8761.985	8761.613	8761.17	8760.863	8761.207	8761.212	8761.095	8762.821	8760.82	8758.819	8761.6	8761.249	8760.217	8760.566
179154	2017-02-0	8761.5	8761.55	8760.35	8760.6	0	8760.62	8761.695	8761.747	8761.327	8761.075	8761.26	8761.248	8761.133	8762.018	8760.62	8759.222	8761.608	8761.262	8760.246	8760.571
179155	2017-02-0	8760.75	8762.3	8760.5	8761.45	0	8760.87	8761.34	8761.813	8761.417	8760.967	8761.167	8761.185	8761.097	8761.715	8760.87	8760.025	8761.595	8761.229	8760.252	8760.576
179156	2017-02-0	8760.75	8761.15	8759.85	8760.55	0	8760.78	8761.045	8761.763	8761.492	8760.895	8761.091	8761.131	8761.064	8761.559	8760.78	8760.001	8761.558	8761.203	8760.258	8760.578
179157	2017-02-0	8760.7	8760.7	8759.85	8760.55	0	8760.85	8760.93	8761.747	8761.487	8760.83	8761.02	8761.077	8761.029	8761.516	8760.85	8760.184	8761.561	8761.193	8760.26	8760.579
179158	2017-02-0	8760.55	8761.15	8760.05	8760.35	0	8760.85	8760.835	8761.607	8761.422	8760.736	8760.935	8761.011	8760.984	8761.516	8760.85	8760.184	8761.585	8761.182	8760.263	8760.579
179159	2017-02-0	8759.5	8760.9	8759.2	8760.9	0	8760.45	8760.535	8761.28	8761.422	8760.324	8760.674	8760.822	8760.842	8761.411	8760.45	8759.489	8761.573	8761.058	8760.26	8760.574
179160	2017-02-0	8755.9	8759.75	8755.1	8759.55	0	8759.48	8760.175	8760.72	8761.23	8758.85	8759.806	8760.207	8760.372	8763.175	8759.48	8755.785	8761.406	8760.601	8760.133	8760.518
179161	2017-02-0	8758.7	8760.45	8756.1	8756.1	0	8759.07	8759.925	8760.387	8761.09	8758.8	8759.605	8760.019	8760.212	8762.559	8759.07	8755.581	8761.141	8760.518	8760.106	8760.51
179162	2017-02-0	8760.95	8761.9	8758.35	8758.35	0	8759.12	8759.985	8760.327	8761.09	8759.516	8759.849	8760.135	8760.283	8762.707	8759.12	8755.533	8760.873	8760.521	8760.11	8760.513
179163	2017-02-0	8761.2	8762.65	8759.7	8761.75	0	8759.25	8760.05	8760.307	8761.017	8760.078	8760.095	8760.268	8760.37	8763.074	8759.25	8755.426	8760.649	8760.528	8760.121	8760.522
179164	2017-02-0	8748.2	8764.7	8737.3	8760.9	0	8756.99	8758.72	8759.353	8760.207	8756.118	8757.932	8758.76	8759.211	8766.573	8756.99	8747.407	8760.21	8758.559	8758.683	8760.047
179165	2017-02-0	8730.05	8748.35	8724.6	8747.6	0	8751.82	8755.65	8757.39	8758.495	8747.429	8752.863	8755.171	8756.434	8775.581	8751.82	8728.059	8759.18	8751.082	8752.704	8756.643
179166	2017-02-0	8740.4	8744.45	8730.35	8732.25	0	8748.16	8753.615	8756.003	8757.33	8745.086	8750.597	8753.325	8754.907	8772.19	8748.16	8724.13	8757.93	8750.11	8751.671	8755.871
179167	2017-02-0	8740.2	8740.2	8733.85	8739.25	0	8744.01	8751.565	8754.66	8756.247	8743.457	8748.706	8751.684	8753.506	8764.707	8744.01	8723.313	8756.727	8749.201	8750.775	8755.003
179168	2017-02-0	8734.5	8742.5	8731.8	8739.4	0	8738.67	8748.96	8752.923	8754.897	8740.472	8746.123	8749.536	8751.696	8750.929	8738.67	8726.411	8755.566	8747.521	8749.203	8753.422
179169	2017-02-0	8723.65	8734.8	8723.55	8734.8	0	8733.76	8745.375	8750.4	8752.955	8734.864	8742.037	8746.3	8749.025	8746.475	8733.76	8721.045	8754.167	8743.965	8745.649	8750.444
179170	2017-02-0	8717.5	8723.55	8715.15	8723.55	0	8731.25	8741.535	8747.517	8750.855	8729.076	8737.576	8742.7	8746.023	8749.607	8731.25	8712.893	8752.368	8739.809	8741.484	8746.589
179171	2017-02-0	8727.35	8727.75	8718.25	8718.25	0	8728.64	8738.4	8745.29	8749.162	8728.501	8735.717	8740.781	8744.244	8744.606	8728.64	8712.674	8750.228	8738.608	8740.226	8745.179
179172	2017-02-0	8733.05	8737.2	8727.65	8727.65	0	8727.21	8735.61	8743.447	8747.797	8730.017	8735.232	8739.815	8743.178	8739.675	8727.21	8714.745	8748.227	8738.191	8739.801	8744.567
179173	2017-02-0	8737.2	8738.6	8731.5	8733.1	0	8727.75	8733.21	8741.89	8746.63	8732.411	8735.59	8739.488	8742.609	8741.59	8727.75	8713.91	8746.85	8738.135	8739.685	8744.25
179174	2017-02-0	8737.65	8737.65	8733.75	8737	0	8730.55	8732.155	8740.433	8745.437	8734.158	8735.964	8739.258	8742.136	8745.555	8730.55	8715.545	8745.663	8738.124	8739.591	8743.941
179175	2017-02-0	8740.5	8745.4	8737.2	8737.35	0	8735.15	8733.2	8739.407	8744.425	8736.272	8736.789	8739.414	8741.981	8744.286	8735.15	8726.014	8744.618	8738.198	8739.624	8743.809

Nifty50 data for every minute for the last 5 years and showing all the features



News archive from economic times

Top headlines for the day



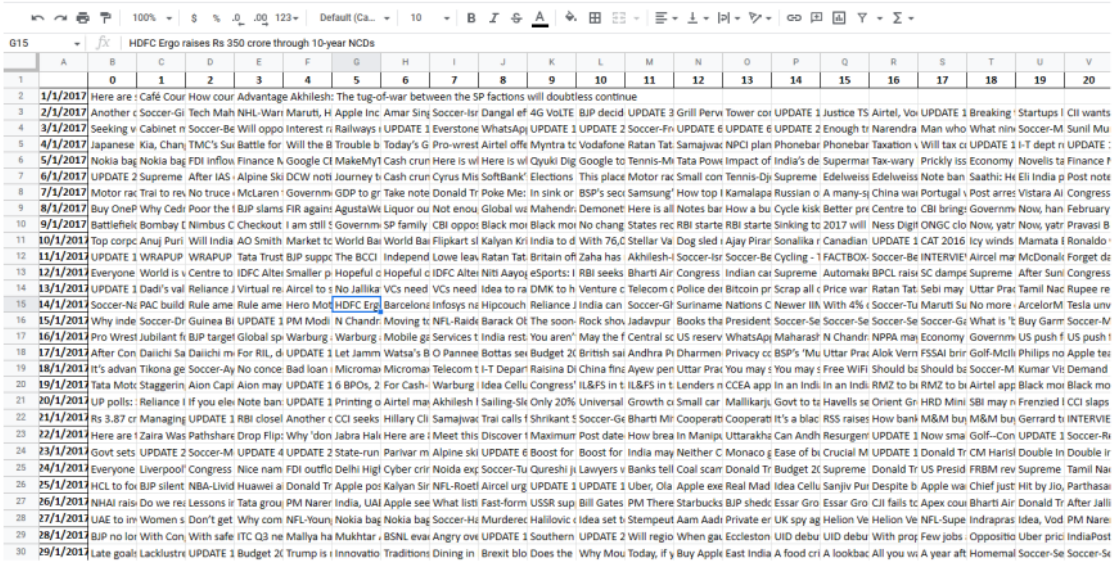
Extracted headlines from the website to parse in HTML format

b. Cleaning and preparing data

The gathered information is then cleansed and formatted in line with the model's requirements. All unnecessary data is eliminated along with duplicate and null values. Furthermore, superfluous columns are eliminated. The most important and labor-intensive stage of machine learning technology.

Python's built-in logic and a number of statistical techniques are used to clean and prepare the data. For instance, the price was a character type rather than a number. The most important phase in creating the derived columns from the present columns is finished at this point. As an example, the precision value has been improved to up to 2 decimal places. In the news headlines section, the special symbols have been removed.

Outliers are addressed separately during this step since they might provide unexpected consequences. The proper data type has been allocated to the data since type is the foundation of computer languages. For instance, date has been converted from a string to a datetime type and generic statistics have been determined. Data standardization has also been carried out.



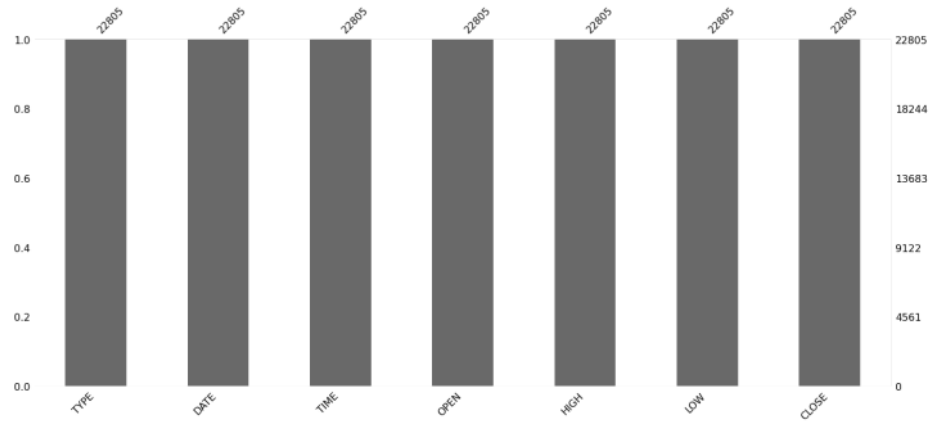
c. Converting data from textual to numerical form

The data is in the form of text and to make it ready for the model it needs to be converted into numerical form, as the model operates on numerical data. There are multiple options to convert the data into numerical form. News headlines can have two types of sentiments, either positive or negative. However, Vader[10] helps to calculate the compound and neutral score also. The resultant numerical field can be further used as a numerical input to predict the results. The reason for choosing VADER because with it there is no requirement for general processing the text as it handles everything and abstract a lot of work.

H	I	J	K	L
headlines	compound	negative	neutral	positive
ET Recom	0.9988	0.087	0.797	0.117
It is time v	0.9958	0.096	0.792	0.112
IT hardwa	-0.998	0.106	0.8	0.094
Saudi Arab	0.9918	0.09	0.803	0.107
Aamir Kha	0.998	0.074	0.828	0.098
Indira Gar	-0.9958	0.1	0.81	0.09
If China dc	0.9983	0.08	0.813	0.107
Zomato sh	0.9955	0.046	0.887	0.067

	prevclose	open	high	low	last	close	headlines	compound	negative	neutral	positive
1/1/2016	1082.15	1082.4	1090.25	1076.15	1088.7	1088.75	ET Recom	0.9988	0.087	0.797	0.117
1/4/2016	1088.75	1084	1084	1068.1	1068.5	1070.5	It is time v	0.9958	0.096	0.792	0.112
1/5/2016	1070.5	1070.2	1074.8	1061.35	1062	1062.4	IT hardwa	-0.998	0.106	0.8	0.094
1/6/2016	1062.4	1056.65	1076.75	1056.65	1067.55	1067.1	Saudi Arat	0.9918	0.09	0.803	0.107
1/7/2016	1067.1	1060.1	1064.9	1049.7	1052.55	1056.2	Aamir Kha	0.998	0.074	0.828	0.098
1/8/2016	1056.2	1061.95	1064.5	1057.25	1062	1062.35	Indira Gar	-0.9958	0.1	0.81	0.09

Table 2 : Clean and Prepared Dataset



Updating the data to the updated data type

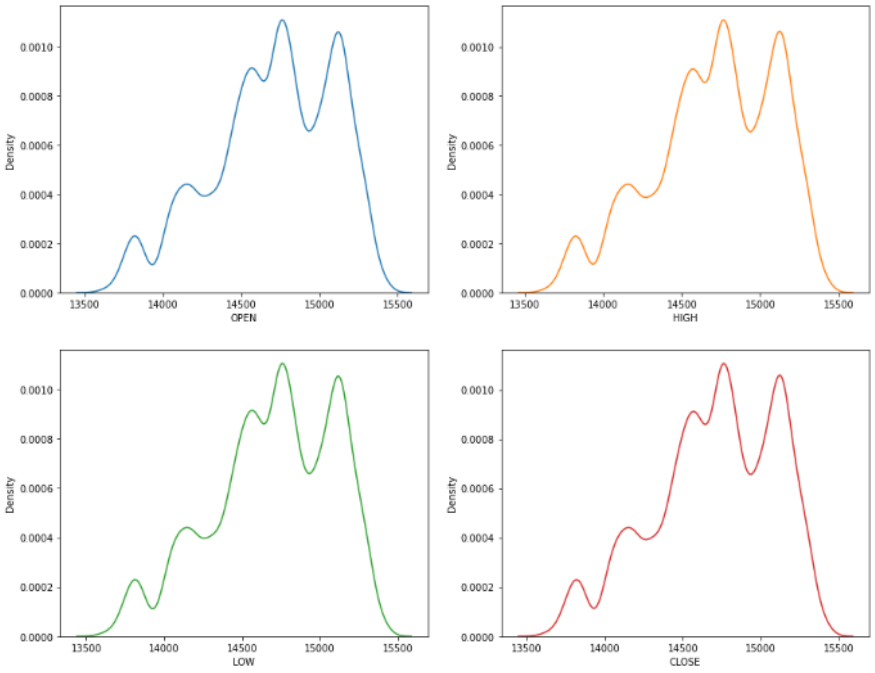
d. Analyzing data

Data preparation is followed by data analysis, the discovery of hidden trends, and eventually the use of various machine learning models. A few features can be derived from the current features utilizing statistical techniques. Exploratory data analysis consists of various approaches, that is helpful for below points:

- Improve understanding of a data set,
- find underlying structure,
- extraction of key variables,
- spot outliers and anomalies,
- test underlying hypotheses,
- build parsimonious models, and

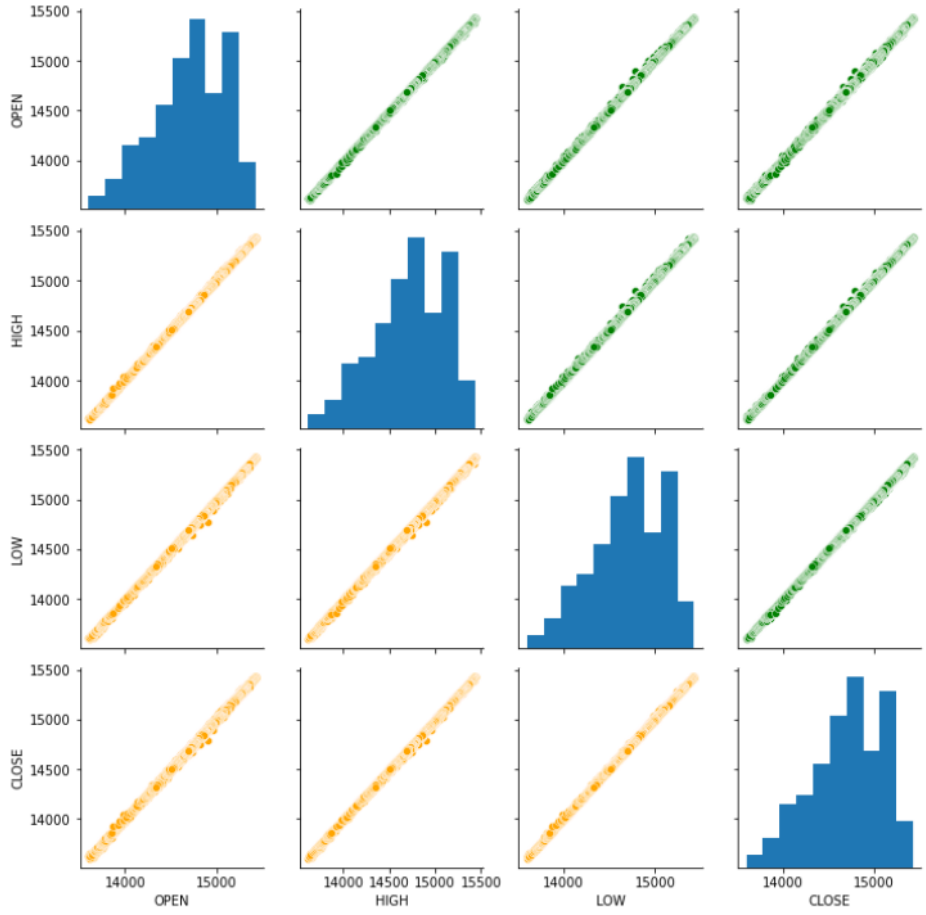
- choose the best factor settings.

Making a plot of the raw data, such as a block plot, probability plot, histogram, or bi-histogram. creating simple statistical plots from the raw data, such as mean plots, standard deviation plots, box plots, and major effects plots. putting numerous plots on a page to enhance our capacity for pattern identification when placing such plots.



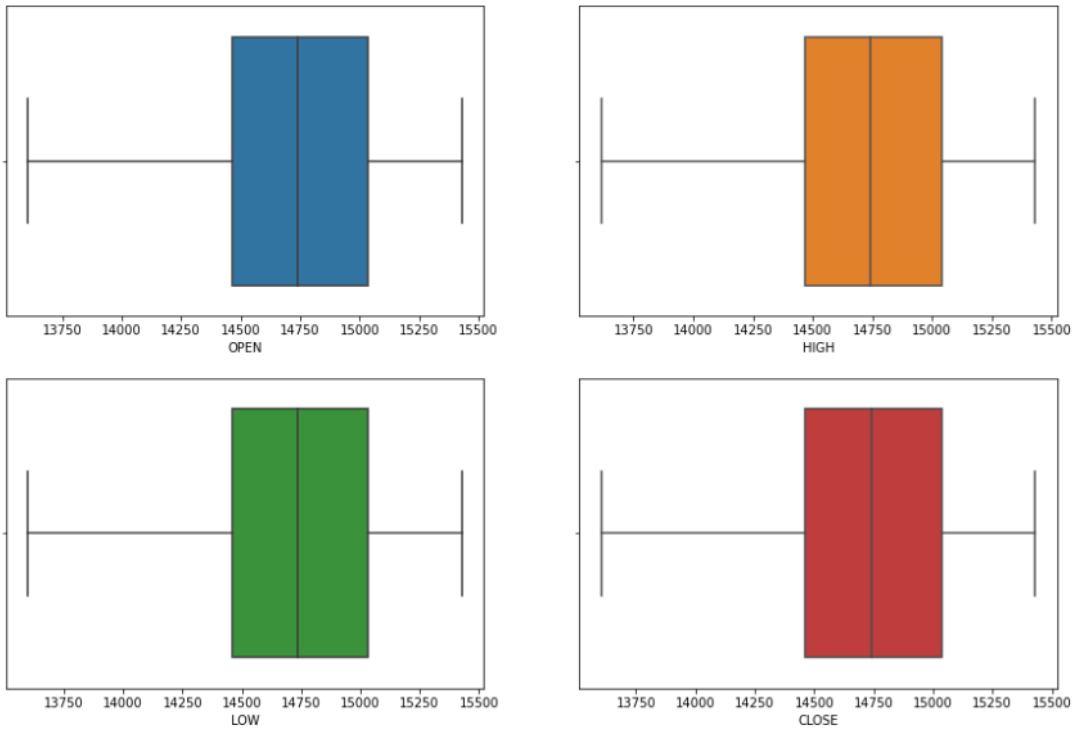
Distribution plots for the features

With the help of distribution plots, it has been identified that the data is skewed or normally distributed with different parameters.



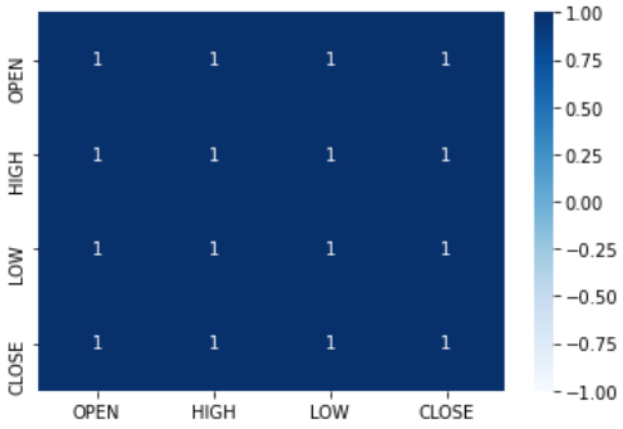
Scatter plots for the features

To identify the data distribution, scatter plots have been plotted.



Box plot to determine the outliers

Box plots are extensively used to determine the outliers in the dataset.



Heat map

The dataset's heat map/matrix examined the relationships between several characteristics.Now, the model has been trained using Linear, Poly SVM technique with 0.8 and 0.2 ratio with segregation of dependent and independent variables. The data has been transformed for feature scaling.

Chapter 5: Machine Learning Model Performance

To predict the price of stock, a variety of machine learning methods have been available, including Support Vector Machine (SVM) as Linear, Support Vector Machine (SVM) as Poly, ANN, LSTM, Naive Bayes and Random Forest Algorithm. These models may be applied using the Python module Scikit Learn. R2, MAE, and MSE are a few of the parameters that are used to evaluate how well these models work. The formulae for these three parameters are as follows:

$$R^2 = 1 - \frac{\sum_n^{t=1}(yi - \hat{yi})^2}{\sum_n^{t=1}(yi - \bar{yi})^2} \quad (1)$$

$$MAE = \frac{1}{n}\sum_{n}^{t=1}|yi - \hat{yi}| \quad (2)$$

$$MSE = \frac{1}{n}\sum_{n}^{t=1}(yi - \hat{yi})^2 \quad (3)$$

Below are the models compared and these parameters calculated to get the best model

5.1- Random Forest Regressor

In order to produce more precise forecasts, this algorithm combines the less predictive algorithms. By merging the basic model, it creates a massive model. The features are sampled and transferred to the trees without replacement to produce highly uncorrelated decision trees. To select the best split, there must be less link between the trees. The crucial concept is aggregate uncorrelated trees, which set the random forest apart from the decision tree.

```
[ ]  mse = mean_squared_error(rescaled_ytest,test_predict)
     print('MSE: '+str(mse))
     rmse = math.sqrt(mean_squared_error(rescaled_ytest,test_predict))
     print('RMSE: '+str(rmse))
     mape = np.mean(np.abs(test_predict - rescaled_ytest)/np.abs(rescaled_ytest))
     print('MAPE: '+str(mape))

     MSE: 565.2411820504213
     RMSE: 23.77480140927409
     MAPE: 0.015401272104545958
```

5.2 - SVM Regressor

Regression analysis and classification both employ the SVM supervised machine learning method. Since processing takes a long time, modest datasets are often used. It discovers the Hyperplane that divides the feature into many pieces. It provides an ideal hyperplane that categorizes many domains. Support vector points are the data points that are closest to the hyperplane, and margins are the separations between these points and the vector plane.

$$y = w_0 + \sum_{i=0}^{m} w_i x_i$$

The proposed work has exploited SVM for regression analysis. The performance depends on kernel function selection as a Non-parametric technique. Linear, Radial Basis Function and Polynomial are the kernels of support vector machine algorithms.

Support vector machines provide the following benefits:-

Effective in high-dimensional spaces.

SVMs are excellent when we don't know anything about the data. even with unstructured and semi-structured material, it performs effectively.

Even if there are more dimensions than samples, the method is still efficient.

It scales to high dimensional data rather well.

-It is memory efficient because the decision function uses a collection of training points known as support vectors.

Being adaptable allows for the specification of several kernel functions for the decision function. There are common kernels available, but you may also define your own kernels.

SVR Linear:

```
[ ]  mse = mean_squared_error(y_test_e,y_test_pred_e)
     print("Mean_squared_error is: ", mse)
     r2score = r2_score(y_test_e , y_test_pred_e)
     print("R2 score is: " , r2score)

     Mean_squared_error is:  7971.497158328377
     R2 score is:  0.9610811950203848
```

SVR Poly:

```
[ ]  from sklearn.metrics import mean_squared_error, r2_score
     mse = mean_squared_error(y_test_e,y_test_pred_f)
     print("Mean_squared_error is: ", mse)
     r2score = r2_score(y_test_e , y_test_pred_f)
     print("R2 score is: " , r2score)

     Mean_squared_error is:  15.522993756385382
     R2 score is:  0.9999242129358256
```

5.3 Xgb Regressor

This approach makes use of gradient boosting to discover the function that fits the input data the best. By learning from the mistakes of prior decision tree training, it trains several decision trees and gets better each time. The goal of each iteration is to reduce the discrepancy between the anticipated and input data. When it comes to performance and scalability in a distributed context, it outperforms gradient boosting. Although accurate, it once again falls into the overfitting issue area, which may be resolved by adjusting the hyperparameter.

```
[ ]  import xgboost
     import lightgbm

     gbm = lightgbm.LGBMRegressor()
     gbm.fit(X_train_c, y_train_c)
     prediction_c=gbm.predict(X_test_c)
     print(mean_squared_error(prediction_c, y_test_c))

     42.215051073569164

     xgb = xgboost.XGBRegressor()
     xgb.fit(X_train_c, y_train_c)
     prediction_c = xgb.predict(X_test_c)
     print(mean_squared_error(prediction_c, y_test_c))

     51.6639762495133
```

5.4 - Experimental results with compare to auto h2o library

There is an open source library for python, that runs different algorithms and tries the best value for the hyperparameter to obtain the best results and then produce the chart based on it. It is based on the JVM and works as a distributed system with in memory machine learning techniques.

```
MSE: 52.61985338060575
RMSE: 7.253954327165684
MAE: 4.4102307686098925
RMSLE: 0.005876066245025321
Mean Residual Deviance: 52.61985338060575
R^2: 0.9997441716107782
Null degrees of freedom: 986
Residual degrees of freedom: 981
Null deviance: 203702579.08870408
Residual deviance: 51935.79528665787
AIC: 6726.557938412846
```

Cross-Validation Metrics Summary:

	mean	sd	cv_1_valid	cv_2_valid	cv_3_valid	cv_4_valid	cv_5_valid
mae	4.4064765	0.3122633	4.831698	4.0752583	4.627423	4.2255373	4.272467
mean_residual_deviance	52.98148	28.916622	103.4491350	32.865864	49.900578	41.76592	36.925907
mse	52.98148	28.916622	103.4491350	32.865864	49.900578	41.76592	36.925907
null_deviance	40740516.0000000	5632848.0	40887808.0000000	32194294.0000000	39371752.0000000	44161680.0000000	47087044.0000000
r2	0.9997433	0.0001280	0.9995249	0.9998294	0.9997341	0.9998010	0.999827
residual_deviance	10387.159	5315.7344	19344.988	5488.5996	10429.221	8770.843	7902.1436
rmse	7.1014457	1.7856885	10.170995	5.732876	7.064034	6.4626555	6.0766687
rmsle	0.0056489	0.0018366	0.0085043	0.0038378	0.0059166	0.0057434	0.0042424

Chapter 6 : Integrating with Project

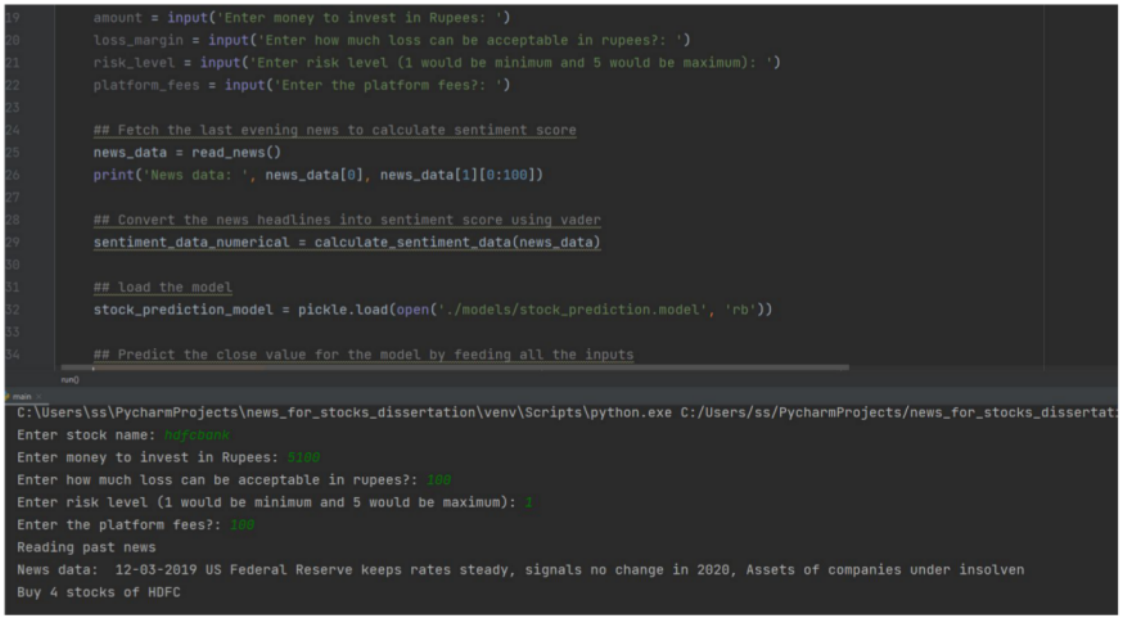
The final step for the project is to integrate the model with the python project to give an interactive interface to the trader, so that it can abstract the complexity of the model and operate efficiently. After comparing the different models, the poly models proved to be best among the others. To save the model in a file, there is an external library available in python named "pickle". With the help of pickle, the model can be saved and load from the binary file.

```
[ ]  filename = 'svr_linear.pkl'
     pickle.dump(svr, open(filename, 'wb'))

     filename = 'svr_poly.pkl'
     pickle.dump(model_poly, open(filename, 'wb'))
```

Fig: Saving the model

Once the model has been saved to the file, it will be loaded into the python application and few of the parameters are taken as input from the user, like the name of stock where the user wants to trade with a limited of companies available, as the model needs to be trained for that company share price from the past. The second essential parameter is the amount of the money to invest in form of integer value, platform fees also play a vital role, and risk level basically decides if the user is willing to take the risk or not. Once the mandatory inputs are provided by the user, the program processes it using the model and produces the output that will help the trader to trade wisely for the stock.


amount = input('Enter money to invest in Rupees: ')
loss_margin = input('Enter how much loss can be acceptable in rupees?: ')
risk_level = input('Enter risk level (1 would be minimum and 5 would be maximum): ')
platform_fees = input('Enter the platform fees?: ')
Fetch the last evening news to calculate sentiment score
news_data = read_news()
print('News data: ', news_data[0], news_data[1][0:100])
Convert the news headlines into sentiment score using vader
sentiment_data_numerical = calculate_sentiment_data(news_data)
load the model
stock_prediction_model = pickle.load(open('./models/stock_prediction.model', 'rb'))
Predict the close value for the model by feeding all the inputs
run()
main
C:\Users\ss\PycharmProjects\news_for_stocks_dissertation\venv\Scripts\python.exe C:/Users/ss/PycharmProjects/news_for_stocks_dissertat
Enter stock name: hdfcbank
Enter money to invest in Rupees: 5100
Enter how much loss can be acceptable in rupees?: 100
Enter risk level (1 would be minimum and 5 would be maximum): 1
Enter the platform fees?: 100
Reading past news
News data: 12-03-2019 US Federal Reserve keeps rates steady, signals no change in 2020, Assets of companies under insolven
Buy 4 stocks of HDFC

Chapter 7 : Conclusion and Inference

This dissertation gives an understanding of implementation of machine learning techniques to identify the factors affecting the price of the stock and providing a confidence to the trader to invest in the stock market in the form of either buying or selling. The proposed project gives equal weightage to the trader capabilities of bearing the loss. The main objective of the project is to save the trader from having a loss and gaining a maximum profit on the good day. The proposed model is trained using the numerical parameters like the open, low, high as well as the textual parameters like the news headlines that is indirectly transformed into the numerical values using the sentiment analyser. The study has been done with various dataset like one year dataset and two year dataset and the final model is trained using the five year dataset.In order to provide better results, the accuracy and performance of various models are compared. Many prediction models are examined to see which one predicts the closing price of the stock most accurately. The model is then further integrated with the python project that gives an interactive interface to the traders to conveniently use it and get the results.

Directions for future work:

Deep learning techniques can be implemented for improved results and it further helps to reduce one extra step for conversion of textual data into numerical data.

An enhanced user interface can be designed using web technologies that will help the end user to browse and select the options easily.

Proper way of handling the weekend data, as the stock market doesn't operate on weekends however the collective news is of three days, that may vary the predictions.

Grouping of potential stocks instead of calculating in isolated environments of the value of individual stocks.

Chapter 8 : References

[1] Thomas, J. D., and Sycara, K., 2000. Integrating Genetic algorithms and text learning for financial prediction. GECCO, July 8-12, Las Vegas, USA, pp. 72-75.

[2] https://economictimes.indiatimes.com/defaultinterstitial.cms

[3] https://nsetools.readthedocs.io/en/latest/

[4] https://www.sciencedirect.com/science/article/abs/pii/S0957417414004473

[5] NLP in Stock Market Prediction: A Review by Rodrigue Andrawos

[6] Sheikh Abdullah, Mohammad Rahaman, and Mohammad Rahman. Analysis of stock market using text mining and natural language processing, 05 2013.

[7] Jigar Patel et al. "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques". In: Expert systems with applications 42.1 (2015), pp. 259–268.

[8] David MQ Nelson, Adriano CM Pereira, and Renato A de Oliveira. "Stock market's price movement prediction with LSTM neural networks". In: 2017 International joint conference on neural networks (IJCNN).

[9] Stock Market Prediction Using LSTM Recurrent Neural Network
https://www.sciencedirect.com/science/article/pii/S1877050920304865

[10] Vader Sentiment Analyser C.J. HuttoEric Gilbert
https://ojs.aaai.org/index.php/ICWSM/article/view/14550

[11] https://www.researchgate.net/publication/360726066_NLP_in_Stock_Market_Prediction_A_Review
NLP in Stock Market Prediction: A Review by Rodrigue Andrawos

Appendix

The GitHub code utilizes the trained model created using collab notebook

GitHub:
https://github.com/sarvsav/dissertation

Collab:
https://colab.research.google.com/drive/1DXR8oXGeJ7yX96o72R2uoPWJGffi-hDj?usp=sharing

Dataset: Uploaded in github repository inside data folder
(https://www.kaggle.com/datasets/debashis74017/nifty-50-minute-data)
Models: Uploaded in github repository inside models folder

Checklist of items for the Final Dissertation Report

a)	Is the Cover page in proper format?	Y / ~~N~~
b)	Is the Title page in proper format	Y / ~~N~~
c)	Is the Certificate from the Supervisor in proper format? Has it been signed?	Y / ~~N~~ Y / ~~N~~
d)	Is Abstract included in the Report? Is it properly written?	Y / ~~N~~ Y / ~~N~~
e)	Does the Table of Contents page include chapter page numbers?	Y /~~N~~
f) i) ii) iii) iv) v)	Does the Report contain a summary of the literature survey? Are the Pages numbered properly? Are the Figures numbered properly? Are the Tables numbered properly? Are the Captions for the Figures and Tables proper? Are the Appendices numbered?	Y /~~N~~ Y /~~N~~ Y /~~N~~ Y /~~N~~ Y /~~N~~ Y /~~N~~
g)	Does the Report have Conclusion / Recommendations of the work?	Y /~~N~~
h)	Are References/Bibliography given in the Report?	Y /~~N~~
i)	Have the References been cited in the Report?	Y /~~N~~
j)	Is the citation of References / Bibliography in proper format?	Y /~~N~~

Declaration by Student:

I certify that I have properly verified all the items in this checklist and ensure that the reports are in proper format as specified in the course handout.

Place: Gurgaon

Signature of the student

Name: Sarvsav Sharma

ID No: 2020SC04239

2020sc04239.pdf

ORIGINALITY REPORT

14%	8%	2%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Birla Institute of Technology and Science Pilani Student Paper	8%
2	arxiv.org Internet Source	1%
3	kth.diva-portal.org Internet Source	1%
4	Submitted to Universität Liechtenstein Student Paper	1%
5	Submitted to Liverpool John Moores University Student Paper	1%
6	papasearch.net Internet Source	1%

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		