Quantitative Trading Analysis

A PROJECT REPORT

Submitted by

Muskan Patel	**171310132035**
Shivam Soni	**171310132055**
Jaimin Thakkar	**171310132057**
Jaydeep Thakur	**171310132058**

In fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

Information and Communication Technology

Adani Institute of Infrastructure Engineering, Ahmedabad

Gujarat Technological University, Ahmedabad

April, 2021

Adani Institute of Infrastructure Engineering

Information and Communication Technology
2021

CERTIFICATE

Date: 08/04/2021

This is to certify that the dissertation entitled "QUANTITATIVE TRADING ANALYSIS" has been carried out by MUSKAN YOGESHBHAI PATEL (171310132035) under my guidance in fulfilment of the degree of Bachelor of Engineering in INFORMATION AND COMMUNICATION TECHNOLOGY(ICT) (7^{th} Semester/8^{th} Semester) of Gujarat Technological University, Ahmedabad during the academic year 2020-21.

Guide:
Dr. Hitesh R. Chhikaniwala

Dr. Hitesh R. Chhikaniwala

Head of the Department

Adani Institute of Infrastructure Engineering

Information and Communication Technology
2021

CERTIFICATE

Date: 08/04/2021

This is to certify that the dissertation entitled "QUANTITATIVE TRADING ANALYSIS" has been carried out by SHIVAM DIPAK SONI (171310132055) under my guidance in fulfilment of the degree of Bachelor of Engineering in INFORMATION AND COMMUNICATION TECHNOLOGY(ICT) (7^{th} Semester/8^{th} Semester) of Gujarat Technological University, Ahmedabad during the academic year 2020-21.

Guide:
Dr. Hitesh R. Chhikaniwala

Dr. Hitesh R. Chhikaniwala

Head of the Department

Adani Institute of Infrastructure Engineering

Information and Communication Technology
2021

CERTIFICATE

Date: 08/04/2021

This is to certify that the dissertation entitled "QUANTITATIVE TRADING ANALYSIS" has been carried out by JAIMIN MANOJ THAKKAR(171310132057) under my guidance in fulfilment of the degree of Bachelor of Engineering in INFORMATION AND COMMUNICATION TECHNOLOGY(ICT) (7th Semester/8th Semester) of Gujarat Technological University, Ahmedabad during the academic year 2020-21.

Guide:
Dr. Hitesh R. Chhikaniwala

Dr. Hitesh R. Chhikaniwala

Head of the Department

Adani Institute of Infrastructure Engineering

Information and Communication Technology
2021

CERTIFICATE

Date: 08/04/2021

This is to certify that the dissertation entitled "QUANTITATIVE TRADING ANALYSIS" has been carried out by JAYDEEP UDAIRAJSINGH THAKUR (171310132058) under my guidance in fulfilment of the degree of Bachelor of Engineering in INFORMATION AND COMMUNICATION TECHNOLOGY(ICT) (7th Semester/8th Semester) of Gujarat Technological University, Ahmedabad during the academic year 2020-21.

Guide:
Dr. Hitesh R. Chhikaniwala

Dr. Hitesh R. Chhikaniwala

Head of the Department

ACKNOWLEDGEMENT

We take this opportunity with much pleasure to thank all the people who have helped us throughout the course of our journey towards producing this report. We sincerely thank our Project Guide **Dr. Hitesh Chhinkaniwala** for his guidance, encouragement, help and motivation. Apart from the subject of our project, we learnt a lot from her, which we are sure, will be useful to us in different stages of our life.

We would also like to express our gratitude to **Dr. Hitesh Chhinkaniwala** sir for his valuable help with the project design, methodology and for review and many helpful comments.

We are highly indebted to **Dr.Devang Pandya** sir and **Dr.NikunjTahilramani** sir of ICT Department at Adani Institute of Infrastructure Engineering, Ahmedabad for their assistance in finalizing the project problem and useful comments.

ABSTRACT

The following project is based on the machine learning prediction of the next possible direction of the price in the next interval of the stock. The following project would give the user the next possible direction of the user selected stock i.e. whether the stock price is going up or down using the indicator values. The indicators in the stock market represents how stock is moving and what way it may go or how it is going based on the closing price of the stock on the specific intervals which are 1m, 2m, 5m, 10m, 15m, 30m, 60m, 90m, 1d, 2d, 7d, 1mo, 1yr. The current project has interval set as 60m and is variable could be changed along with the interval the another this is period which is set as 730 days which makes 2 years. The project will give the user the option of stocks and user can select from them and the program returns the output in the form of How the stock is performing in past 2 years, the next interval price for the selected stock based on 12 indicator prices, the accuracy metrics of the current stock model, Comparison to NSE and how our model faired across it, Predict the price according to each indicator separately and return the possible difference, Indication based on the indicator price, whether to buy the stock or sell it, Confidence percentage of model how much confident is model for selling and buying the current stock, Give the success percentage of the model on the current stock how accurate model is giving direction whether to buy or sell stock of last 70 days, Sentiment analysis of the current stock, it fetches the general news and stock specific news and decodes whether news is positive, negative or neutral. The all the following details are useful for the user and will make the decision easy for the user to decide whether the user should invest in the current stock or not.

DECLARATION OF ORIGINALITY

We hereby certify that we are the sole authors of this IDP/UDP project report and that neither any part of this IDP/UDP project report nor the whole of the IDP/UDP Project report has been submitted for a degree by other student(s) to any other University or Institution.

We certify that, to the best of our knowledge, the current IDP/UDP Project report does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations or any other material from the work of other people included in our IDP/UDP Project report, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that we have included copyrighted material that surpasses the boundary of fair dealing within the meaning of the Indian Copyright (Amendment) Act 2012, we certify that we have obtained a written permission from the copyright owner(s) to include such material(s) in the current IDP/UDP Project report and have included copies of such copyright clearances to our appendix.

We have checked the write up of the present IDP/UDP Project report using anti-plagiarism database and it is in the allowable limit. In case of any complaints pertaining to plagiarism, we certify that we shall be solely responsible for the same and we understand that as per norms, University can even revoke BE degree conferred upon the student(s) submitting this IDP/UDP Project report, in case it is found to be plagiarized.

Date:

Place: Signature of Student

MUSKAN PATEL	171310132035
SHIVAM SONI	171310132055
JAIMIN THAKKAR	171310132057
JAYDEEP THAKUR	171310132058

Signature of Guide

Dr. Hitesh R. Chhikaniwala

List of Figures

Plagiarism Certificate

Quantitative Trading Analysis.docx

ORIGINALITY REPORT

9%	8%	2%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ohmisskate.com Internet Source	3%
2	www.slideshare.net Internet Source	2%
3	Submitted to Institute of Technology, Nirma University Student Paper	1%
4	Submitted to Gujarat Technological University Student Paper	<1%
5	Submitted to Queen Mary and Westfield College Student Paper	<1%
6	blog.quantinsti.com Internet Source	<1%
7	cschecker92.blogspot.com Internet Source	<1%
8	Darshna Patel, Hitesh Chhinkaniwala. "Fuzzy logic-based single document summarisation with improved sentence scoring technique", International Journal of Knowledge Engineering	<1%

TABLE OF CONTENTS

Chapters

Chapter 1: Introduction to Project

General:

The following chapter will consist of the introduction to the project, what we are thinking of creating the project, what is the objective and tools and methods we might use in the final year project.

1.1.1 Problem Summary: -

The biggest problem that a trader faces during trading hours is the huge amount of data being dumped onto the screen because of various new indicators and evolution of trading platforms and nowhere enough time to process it the way they used to do it in good old days when there were very few stocks and few trading strategies based on fewer traditional indicators.

Another big problem that a trader faces is the amount of time it has to spend to come up with a trading idea formulate it into a strategy, validate it and then spend hours observing the screen for any chance to use that or stop a loss-making idea after some time.

These are the initially the basic but widely prevalent problems we hope to tackle with our trading system.

1.1.2 Introduction to the project: -

As we saw the problems mentioned above that are faced by the traders on the daily basis our project tries to solve these problems at least partially if not fully with the use of neural network.

With the advantages of using a neural network we try to categorise various types of indicators such as momentum based, volume based etc. and then generate signals individually from them thus reducing the chances of generating a conflicting signal.

Our system generates trading ideas on both long and short sides to remain on even.

1.2.1 Aim: -

We aim to create a fully automated, high frequency trading system operating across stocks in cash and options both, indices, commodities and cryptocurrency at high precision and accuracy for desired time intervals.

1.2.2 Objective: -

To create a system consisting of various modern as well as traditional indicators operating across many stocks to generate trading signals with the time period of 1hr.

1.3 Problem Specifications: -

- Fetching and building a reliable data set which includes prices reflective of stock splits, bonuses and other corporate adjustments.
- Create a way to differentiate between two negatives values which are generated either by actual loss and other by shorting a stock which is actually a profit.
- No data set available which gives data of shorter time periods like 5mins, 30mins etc. of all stocks for a very long time horizon.

1.4 Prior Art Research:

There have been many trials of creating the perfect machine which can predict the movement of a stock but it's always impossible to get accuracy nearing to 100. The first approach of ours was to predict the movement of the stock which way it might go for this we started research on how to use the machine learning in predicting the time series data. The references for our research are added at the end of the report in the References

1.5 Plan of the work:

The plan of the work is divided in certain steps. Collect the data, learn about indicators, learn about the buy and sell strategies, create dataset from collected data and creating indicator values, merging all data, creating of the model, training of the model on different stocks, predicting of the stock price of near future.

1.6 Materials/ Tools Required:

Tools and dependencies required the project

- Integrated Development Environment (Microsoft Visual Studio Code is used)
- Python3 (Language used for code)
- yfinance (Yahoo API for python to get stock data)
- Tensorflow (Python Library used for Machine Learning)
- Pandas (Python library used to handle data collected and manipulation of data)
- Numpy (Python library to do mathematical operations)

- os & shutil (Library for doing OS operations e.g. creation of directory, etc)
- google_drive_downloader (Python library to download file from Google drive)
- random (Library for making random selections)
- ta (technical analysis: library for calculating one mathematical function of indicator)
- gnewsclient, GoogleNews, newscatcher, newspaper (libraries for collecting news)
- nltk (Natural language processing toolkit for getting the polarity score of news)

Summary:

The following chapter we showed the details regarding the project and what we would do in the project further, additionally to that we also showed which technologies we might use in the project completion.

Chapter 2: Brief History of Work

General:

The previous chapter showed what is the problem we are trying to work on, what is the plan of action and along with it what might be the technologies that would be used.

2.1. Prior Art Work:

There have been many trials of creating the perfect machine which can predict the movement of a stock but it's always impossible to get accuracy nearing to 100. The first approach of ours was to predict the movement of the stock which way it might go for this we started research on how to use the machine learning in predicting the time series data. People have tried to make models on based of the currency change, bonds transaction, social economic growth, change in growth of the subsidiary companies, etc and have achieved great accuracy. We tried to take a new approach which no one has ever tried till now, that is to use the indicator data of the stocks which indicates or shows how the stock is performing. In previous models the layers of model have been not hybrid it was monotonous type of layers in the model, we would make use of different layers of the Recurrent Neural Network to create a dynamic hybrid model for each type of stock.

Summary:

The following section was just the history of the previous approaches how the prediction of models takes place with different approaches.

Chapter 3: Design: Analysis, Design Methodology and Implementation.

3.1 AEIOU CANVAS

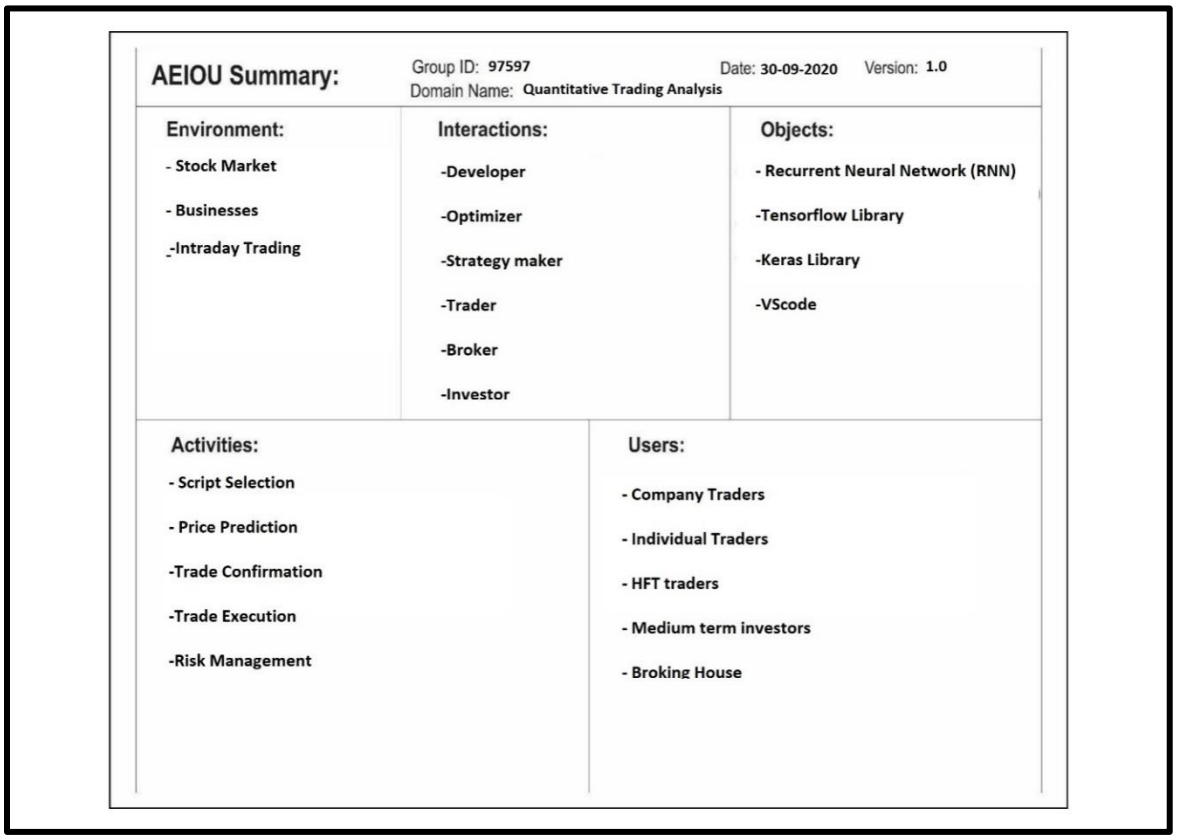

Fig. 3(a) AEIOU canvas

AEIOU stands for Activities, Environment, Interactions, Objects and Users. It helps us to interpret the activities that will be performed, Environment where our model would be useful, Interactions that would be taking place among the users, Objects required for our model and the final users of our project. It helps us to analyse all prospects for making our project.

The activity section includes all the activities that are usually done by the users of the stock market in real time and our project would be useful for such activities. These activities are observed and taken into consideration for making the project. The environment section includes the environment where our model would be useful for the activities taking place. The Object section includes all

the objects i.e. the models and the technologies and the libraries that are used for making our project. The user section includes all the users of our project.

AEIOU Canvas involves surveying and observing all the minute details required for the project. All details are looked into and considered well.

3.2. EMPATHY CANVAS

Fig. 3(b) Empathy Canvas

Empathy canvas is used as a tool to help understand a target persona. It helps one to understand what are the types of experiences a user can get. It can also help one understand the interpretations made by people and what are the problems they face. This helps us to find loopholes in the project. We have to put ourselves in a shoe of service so as to understand what the customers will experience or they feel while using our model. The user section includes all the users of our project. The stakeholder section includes the stakeholders of the project. The story boarding section includes the stories of the users, the good and the bad one. This section includes two happy stories and two bad stories. It plays a major role in analysing our users.

3.3 IDEATION CANVAS

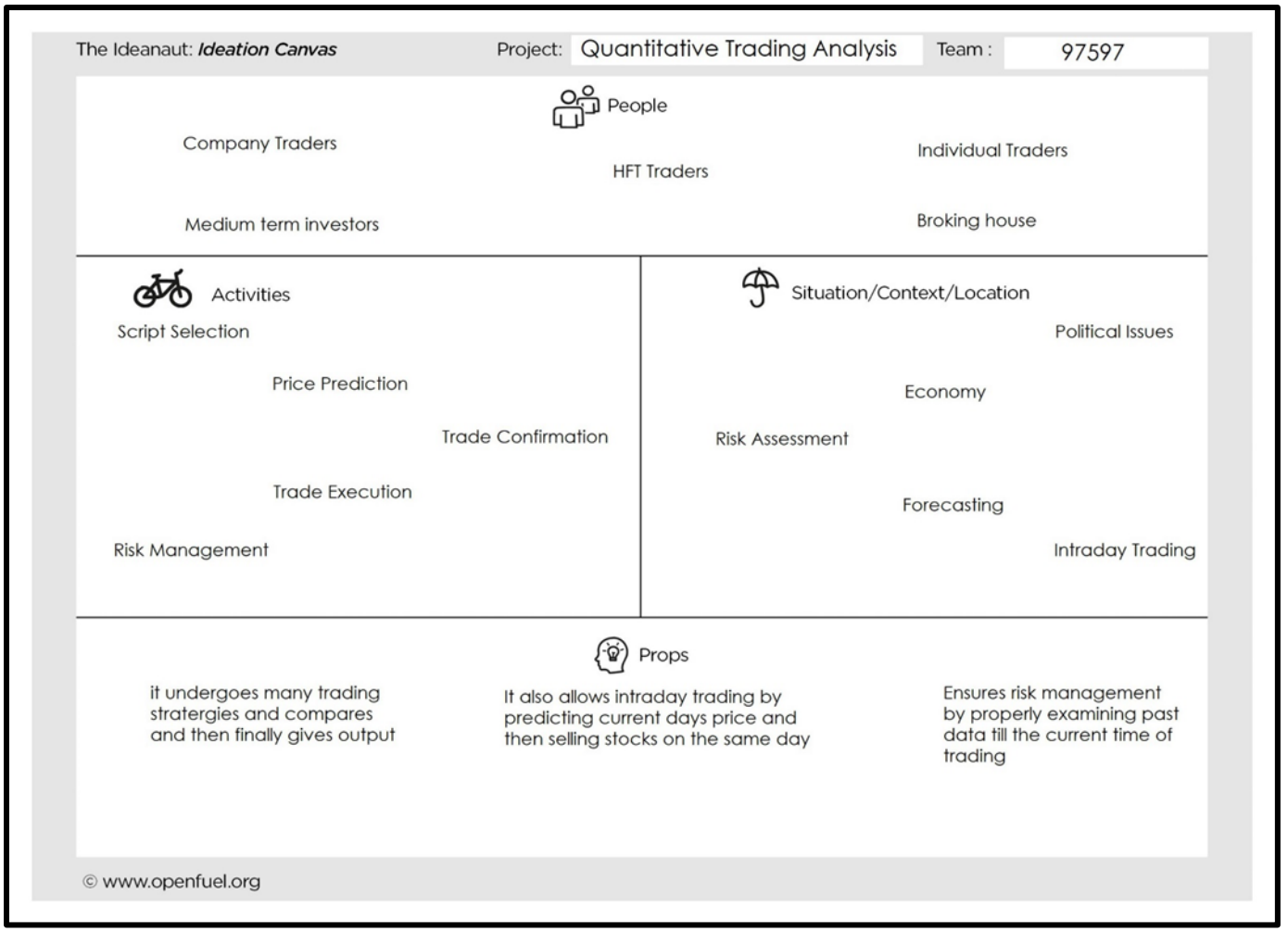

Fig. 3(c) Ideation Canvas

Ideation Canvas provides us the information of users involved in the project, their activities, Situation/Context /Location and props/possible solutions

Users indicates the peoples involved in the project either they are developers or end users using the product. A list of users is shown in Fig. 3(c).

Activities indicates the activities that are does by the users involved in this project. Fig. 3(c) indicates the activities done by the developers.

Situation/Context/Location specifies that the situations faced while designing those activities, the reference used in designing those activities and location of where these activities are done. Fig. 3(c) shows the list of Situation/Context/Location.

Props/Possible indicates the objects, technologies or solutions which may be possible outcomes to nay idea/challenge. Not necessary that it would be a solution to a defined problem. The props need not to be related as it always help in finding new ideas. A list of possible solutions is shown in Fig. 3(c).

3.4 PRODUCT DEVELOPMENT CANVAS

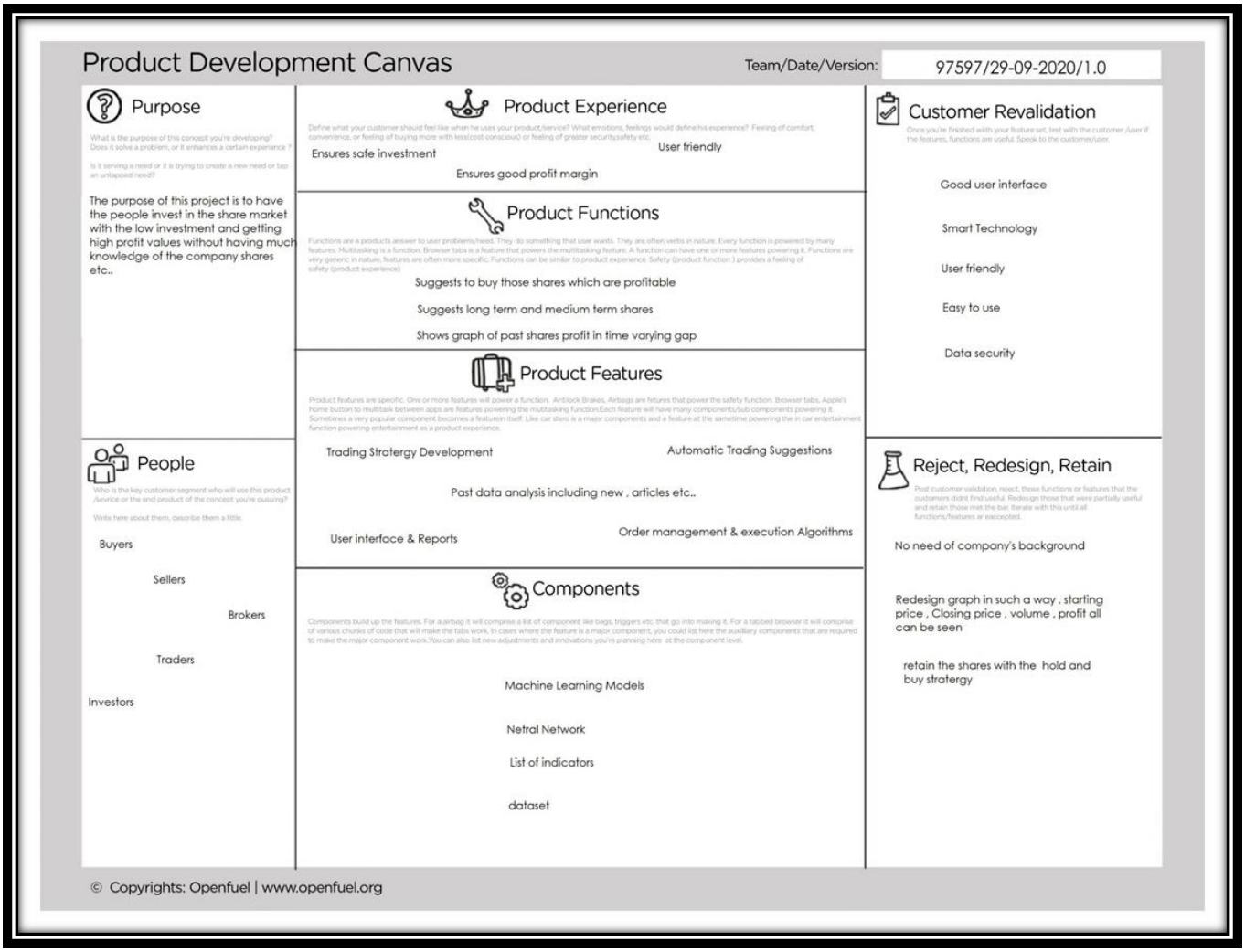

Fig. 3(d) Product Development Canvas

Purpose indicates the reason for developing this project.

Users indicates the peoples involved in the project either they are developers or end users using the product. A list of users is shown in Fig. 3(d).

Product Experience indicates how the customer feels using this product. The emotions, feelings would define his experience which is described in Fig. 3(d).

Product Functions indicates the functions of the project which is developing. Every function is powered by many features. Functions can have one or more feature and they are very generic in nature. A list of product functions is described in Fig. 3(d).

Product Features indicates the features used in designing those functions. Features are specific. Each feature will be designed by components powering it. A list of product features is described in Fig. 3(d).

Components indicates the software, its parts in designing those features. Components build up the features. A list of components is described in Fig. 3(d).

Customer validation indicates that once the product is finalized end user using it will review the advantages and disadvantages of that product.

Reject indicates reject those functions that end user didn't find useful. Redesign those that needs modification and Retain those who met the bar.

3.5 Business Model Canvas (BMC)

Fig. 3(e) Business Model Canvas

Key Partners:

- National Institute of Securities Market (NISM): Provides Certification in Financial Planning or asset or portfolio management or investment advisory. E.g.: - NISM-X-A/B, CFP, CWM.
- Security and Exchange Board of India (SEBI): Approves RIA application i.e., Registered Investment Advisor.
- Sub-Brokers: They are required for the expansion of customer base and setup physical point of contact for customers and their queries.
- National Stock Exchange (NSE): It is an exchange through which most of the trades will be executed.
- Bombay Stock Exchange (BSE): It is the oldest Asian exchange and still works as a reliable place to execute trades.
- Associated Banking Partners: All banks which have savings account of our customer are needed to be paired up with our demat account and also for Pay-In/Pay-Out requests.
- Bloomberg Finance: A far better but costlier alternative of Yahoo Finance for the requirement of our data needed to train models.
- Yahoo Finance: The primary source for data which is required for our model.

Key Activities

- Trade Confirmation: A confirmation that the requested trade has been completed successfully and if not the reason for failure.
- Trigger Alerts: An alert system placed so that upon reaching projected target or stoploss the user will be notified so he/she can take further action if needed.
- Trade Execution: It is imperative that this process is treated with highest priority and with as less as possible execution time because it can impact the entry price.
- Display Available Trading Opportunities: A simplified system that showcases all available trading opportunities of all stocks.
- Sub-Broker Relationships: They are required for the expansion of customer base and setup physical point of contact for customers and their queries.
- Pay-In/Pay-Out Requests: Transactional requests necessary for transferring funds from and to bank’s savings account.

Value Propositions

- Trading Opportunities: High number of trading opportunities spread out across Nifty companies in the initial rollout.
- No Restriction to Particular Stock: No restriction on the selection of just few stocks as entire Nifty components are available and will be only increasing with passage of time.
- Short Term Trading Ideas: All ideas in trading corpus are based for short term trading and thus with time interval of 1 hour.
- Quant Generated Ideas: All ideas are generated through a quant-based model which is built upon the relationship among stock price and indicator values.
- Sentiment Analysis: Every idea generated through Quant is verified via sentiment analysis to check if any non-quant component can affect the trade.

Customer Relationships

- High Net Worth Individuals: Dedicated Personal Assistance and Fully Automated service is provided to provide a more custom trading plan based on customer's risk appetite.
- High Net Worth Corporations/Groups: Dedicated Personal Assistance and Fully Automated service is provided to provide a more custom trading plan based on customer's risk appetite.
- High Frequency Traders: Fully Automated service is provided for HFTs so they just have to observe.
- Medium Frequency Traders: Semi Automated service is provided so that they can execute trades based on their capital constrictions and risk appetite.
- Trading Fund Participants: Fully Automated service and almost no executive decisions are available to them except entering and exiting the fund as everything is under the purview of fund managers.

- Small Retail Traders: Only general idea showcasing is provided and they create their own trading plans and select their trading ideas from our pool.

Customer Segments

- High Net Worth Individual/Groups: High Net worth people who prefer dedicated RMs and modulated plans.
- High/Medium Frequency Traders: Generally, individuals who like to make a trading plan and then an automated system to execute them and then just observe it.
- Trading Fund Participants: Just like mutual funds it is up to them to select which funds to invest in but no power as to what that funds does with their money.
- Small Retail Traders: Usually have low corpus and low number of trades.

Key Resources

- Bloomberg/Yahoo Finance Database: The source for gathering data continuously for training our model.
- Quant Algorithm: The real engine upon which the business is built which takes the data and provides us with 1 hour trading ideas.
- Trading Platform: The front end which makes the interaction between our model and customers possible.
- Sub-Broker System: They are source of the expansion of customer base and setup physical point of contact for customers and their queries.

Channels

- Relationship Managers: Responsible for creating and maintaining relationships with customers usually high end so that their requests are fulfilled in a more satisfactory way.
- Sub-Broker Franchises: They are required for the expansion of customer base and setup physical point of contact for customers and their queries.
- Customer Service Centre: Direct channel between a customer and the company whether for any feedback, query or for new registration.

- Mailing System: Required for updating with necessary information, regular updates and any documentation which might be required.

Cost Structure

- Registration Fee: A registration fee of 5 Lakhs is to be made upon successful application acceptance by SEBI.
- Tangible Asset: A tangible asset of 25 Lakhs has to shown as a sort of guarantee.
- Application Fee: An application fee of 25,000 to get RIA certification.
- Cloud Storage Fee: To be made for storing pre trained models.
- Bloomberg Terminal Fee: Upon scaling up every terminal will be Bloomberg and thus has to bought.
- Quant Engineer Salary: Salary for Quant Engineers and any others who observe and maintain the system running optimally.

Revenue Streams

- Registration Fee: A base registration fee to be charged upon every application.
- Broking Charges: A fixed charge which vary across every customer segment based on the size and number of trades executed.
- PMS Charges: Charged to customers who are participant of any fund and is generally charged as a percentage of Asset Under Management.
- Personal Advisory Commission: Charged to customers to whom customable dedicated Portfolio Management Service is provided.

Chapter 4: Implementation of the project

4.1 Programs:

The programs done are all in Python language. We have used neural networks in creation of the project and used supporting libraries and used historical data available to us through yahoo.

The first part of the project we started by finding the ways to collect the data from the different available API's in the python. The yahoo finance API available in python as "yfinance" is being used the project to collect data of the stocks. The data that we can get depends on what amount of the data we want i.e. at what amount of interval we want the data e.g. 1m, 5m, 15m, 60m, 90m, 1d, 1wk, 1mo.

The first part we did in the project was to find the returns from the data which was available. This program takes inputs from the user and gives the user the cumulative returns on the stock given by the user. The program first takes input from the user how many stocks returns you want to calculate, next input to be given are yahoo tickers(identifiers) for specific stock that is to be used by program to collect data from yahoo and lastly 2 dates in format "YYYY-MM-DD" i.e. start date and end date for which the data will be taken. The output of the following program would be returns on a specific stock if invested on the start date based on 2 strategies simple buy and hold and EMA indicator. The another output of the program is it gives signal when the stock will go up trend and when the stock may go down trend.

```
Enter the number of stocks you would like to enter:
1

Enter the yahoo ticker for stock:
reliance.ns

Enter the date in the format of YYYY-MM-DD, ex:- 2020-07-17

Enter starting date of stock:
2020-01-01

Enter ending date of stock:
2020-08-14
```

Fig. 4(a) Sample Input of First Program

```
Total portfolio return is: 40.02%
Average yearly return is: 71.08%
```

Fig. 4(b) Portfolio Returns Based on Simple Buy and Hold Strategy

```
Total portfolio return is 70.90%
Average yearly return is 135.10%
```

Fig. 4(c) Portfolio Returns Based on EMA indicator

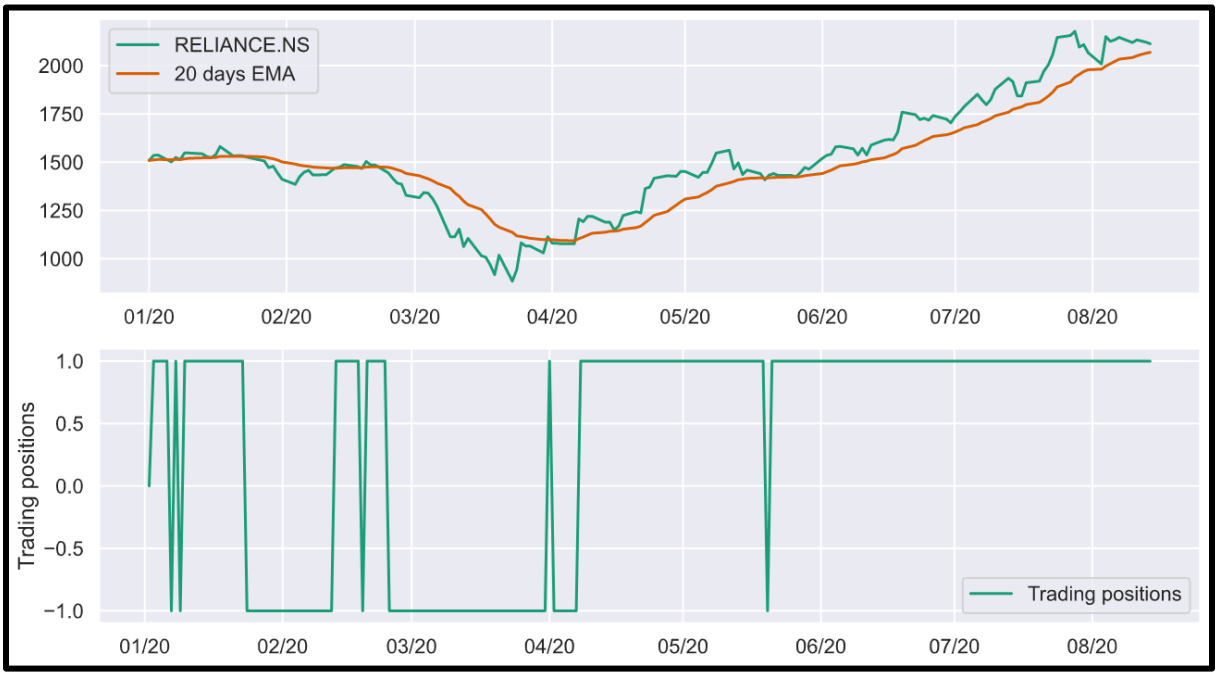

Fig. 4(d) Trading Positions or Signals Shown Over the Period Specified by User

The next part of the project, we tried making the indicators used by people for checking trend of the specific stock. There are plenty of indicators available and can be used easily and they are classified according to volatility, volume, momentum, etc. We started by making a momentum indicator known as “Relative Strength Index (RSI)”. This indicator shows whether the stock is being undersold or oversold. We made this program with the help of pandas library, yahoo for data. The output of the program is a graph which shows the stock in between ranges i.e. if the stock line is in between 30% to 70% than stock is performing fairly while if the line goes above 70% that signifies that stock is being overbought and the price for the stock may dip same applies for

stock value if it goes below 30% but inversely, if it goes below 30% it signifies it is being oversold and price may go up and gives a buy signal.

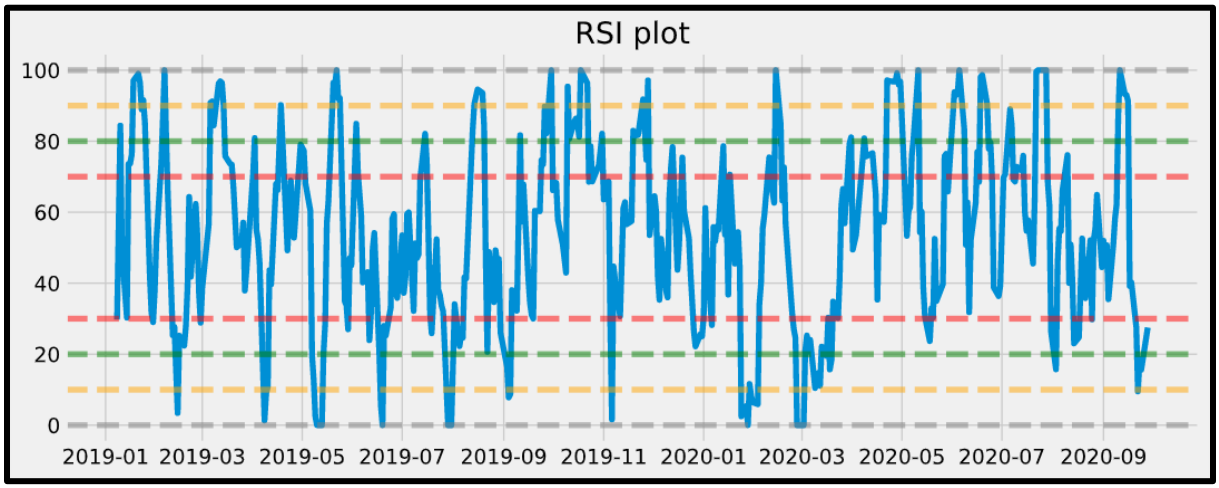

Fig. 4(e) RSI plot of ITC stock from date 01-01-2019 to 28-09-2020

Third step we did was creating a Machine learning model using fundamental approach. We created a program which was given the input of the closing price of the ITC stock from "01-01-2019" to "28-09-2020". The program was taking closing price of this period and then the dataset is converted in supervised learning dataset after that the dataset is divided in 2 parts train and test. The two regressors are further used in the program which predicts the future 60 prices using this 2 regressorLinearRegressor and DecisionTreeRegressor from the sklearn module of the python. The output given of the following program is 60 days future predictions.

```
#create the decision tree regression model
tree=DecisionTreeRegressor().fit(x_train,y_train)
#create the linear regression model
lr=LinearRegression().fit(x_train,y_train)
```

```
lr_score = lr.score(x_test, y_test)
lr_score
```

```
0.8051280943532011
```

```
tree_score = tree.score(x_test,y_test)
tree_score
```

```
0.8663640562745281
```

Fig. 4(f) Training of Models and Confidence Score of models

The Fourth part done was use of neural network, as the fundamental Machine learning was not effective we tried using a Recurrent Neural Network (RNN). In this neural network the data fed is that of the closing price of whichever stock we want to add and the interval of the data taken is that of 5 minutes, so the final predicted data is the data that might be after 5 minutes.

```
M↓
#Build LSTM model
model = Sequential()
model.add(LSTM(50, return_sequences=True, input_shape=(x_train.shape[1], 1)))
model.add(LSTM(50, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))

M↓
#compile the model
model.compile(optimizer='adam',loss='mean_squared_error')
```

Fig. 4(g) LSTM Recurrent Neural Network Structure

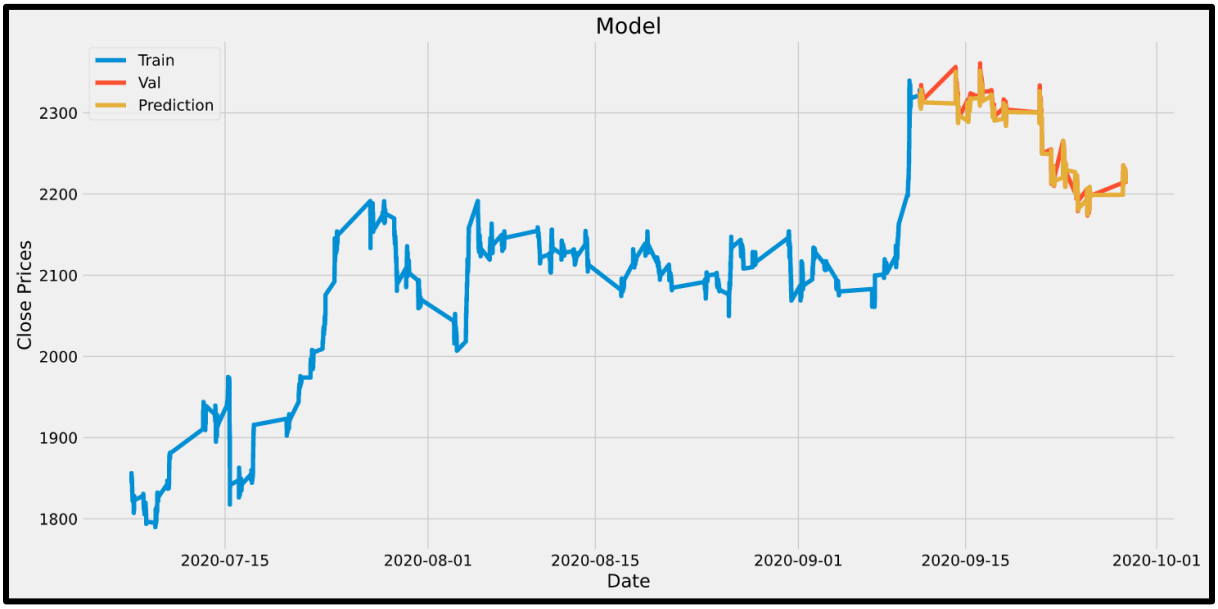

Fig. 4(h) Graph of Model Training along with Comparision of predicted data and original value.

```
    pred_price = scaler.inverse_transform(pred_price)
    print(pred_price)

[**********************100%**********************]  1 of 1 completed
[[2224.4163]]

 ▷  M↓

    #stock_price_check = web.DataReader(ticker, data_source='yahoo', start='2020-08-27', end='2020-08-27')
    stock_price_check = yf.download(tickers=ticker, period=period, interval=interval)
    print(stock_price_check['Close'][-1])

[**********************100%**********************]  1 of 1 completed
2229.300048828125
```

Fig. 4(i) The First price is predicted price for Reliance that might be after 5 minutes and second price was the price after 5 minutes

As the sequence was just closing price and time interval was as little as 5 minute the margin for the predicted was very less we started new approach. The next program we made was to get stock data for all NIFTY50 companies. We tried to make a cumulative dataset for all these companies but as the company HDFCLIFE is not older than 3 years the dataset would be incomplete hence 49 companies are selected. The output of the program was 4 directories 'companies_with_close_data' :- consists of 49 files with closing price of a specific stock

'companies_with_volume_data':- consists of 49 files with closing price and closing volume of a specific stock

'merged_data_of_all_company':- consists of 2 files each consists cumulative data of all closing prices of stock and volume respectively for all stocks

'all_datas_of_a_company':- consists of 49 files each consisting of closing price, opening price, high of day, low of day, volume of a specific stock.

```
tickers=["KOTAKBANK.NS", "RELIANCE.NS", "BAJAJ-AUTO.NS", "TITAN.NS",
         "COALINDIA.NS", "HINDALCO.NS", "TATASTEEL.NS", "ULTRACEMCO.NS", "NTPC.NS",
         "BRITANNIA.NS", "ITC.NS", "HEROMOTOCO.NS", "GAIL.NS", "TECHM.NS",
         "WIPRO.NS", "MARUTI.NS", "SHREECEM.NS", "NESTLEIND.NS", "ONGC.NS",
         "BAJFINANCE.NS", "ICICIBANK.NS", "ZEEL.NS", "TCS.NS", "INDUSINDBK.NS",
         "LT.NS", "M&M.NS", "BHARTIARTL.NS", "CIPLA.NS", "BAJAJFINSV.NS",
         "HCLTECH.NS", "GRASIM.NS", "ADANIPORTS.NS", "EICHERMOT.NS", "JSWSTEEL.NS",
         "INFY.NS", "SBIN.NS", "AXISBANK.NS", "IOC.NS", "DRREDDY.NS",
         "HINDUNILVR.NS", "SUNPHARMA.NS", "POWERGRID.NS", "ASIANPAINT.NS", "BPCL.NS",
         "UPL.NS", "TATAMOTORS.NS", "HDFCBANK.NS", "HDFC.NS", "INFRATEL.NS"]
```

Fig. 4(j) List of all 49 identifiers for extracting data

Fig. 4(k) Directories consisting of dataset.

Furthermore, we integrated indicators in the dataset. We created program which would calculate the values for indicators and formulate a dataset for a particular company which will have values of all indicators along with the closing price, high, low, open. The output of the program is similar as that of "Fig. 4(k)" but the files inside would consist of complete dataset that is to be fed to the model.

```
def create_data_for_indicator():
    os.chdir(data_dir)
    for file in glob.glob("*.csv"):
        data = pd.read_csv(file)

        add_sma_values(data,file)
        add_ema_values(data,file)
        add_stochastic_values(data,file)
        add_macd_values(data, file)
        add_bollinger_bands(data,file)
        add_rsi_values(data,file)
        add_average_true_range(data,file)
        add_cci_values(data,file)
        add_ichimoku_values(data,file)
        add_average_directional_index(data,file)
        add_fibonacci_values(data,file)
        add_parabolic_sar(data,file)

        remove_extra(data,file)
```

Fig. 4(l) Driver function to add all indicator values.

	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Close	20_SMA	50_SMA	100_SMA	200_SMA	20_EMA	50_EMA	100_EMA	200_EMA	Sto. Osc.(9	MACD	Bol. Up	Bol. M	Bol. Dn	RSI	ATR
2	219.9	221.2256	222.4792	223.5937	222.1603	221.2851	222.3087	222.6882	221.9246	14.28618	-0.60872	223.0445	221.2631	219.4817	38.52465	1.149114
3	220.1625	221.1338	222.3525	223.5806	222.177	221.1782	222.2246	222.6381	221.907	20.53573	-0.62472	222.9506	221.1525	219.3544	43.53447	1.127687
4	220.3875	221.0306	222.243	223.5806	222.1961	221.1029	222.1525	222.5936	221.8919	25.89336	-0.61219	222.8081	221.0638	219.3194	54.04055	1.063402
5	220.9125	221.0194	222.153	223.587	222.2143	221.0848	222.1039	222.5603	221.8822	30.99976	-0.55352	222.7657	221.0262	219.2868	54.72625	1.044649
6	221.1	220.9894	222.0765	223.5926	222.2338	221.0862	222.0645	222.5314	221.8744	35.99976	-0.48629	222.7179	220.9919	219.2658	53.33336	1.033932
7	220.725	220.9294	222.0113	223.5952	222.2537	221.0518	222.012	222.4956	221.863	25.99976	-0.45799	222.6638	220.9556	219.2475	46.27652	1.031254
8	220.3125	220.845	221.9468	223.5986	222.2649	220.9814	221.9454	222.4524	221.8475	15	-0.4635	222.4853	220.865	219.2447	29.55962	0.977683
9	220.5	220.8056	221.8815	223.6042	222.2762	220.9356	221.8887	222.4137	221.8341	20	-0.44758	222.409	220.8062	219.2035	32.91124	0.948217
10	220.35	220.8338	221.826	223.5975	222.2833	220.8798	221.8283	222.3728	221.8194	26.66626	-0.44197	222.408	220.8081	219.2083	35.37393	0.86786

Fig. 4(m) WIPRO company final dataset. (Dataset has 4851rows and 27 columns)

Final step we did for the project was to use the dataset we created and feed to a Recurrent Neural Network made of 6 layers. The model gives output what would be the stock price after 1 hour of the stock it has trained on. The final calculations also shows how much error is there in predicting the price.

```
M↓
model = keras.models.Sequential()
model.add(keras.layers.LSTM(24, input_shape=(None,24),return_sequences=True,activation='elu'))
model.add(keras.layers.LSTM(48, return_sequences=True,activation='elu'))
model.add(keras.layers.LSTM(96, return_sequences=True,activation='elu'))
model.add(keras.layers.LSTM(48, return_sequences=True,activation='elu'))
model.add(keras.layers.LSTM(24, return_sequences=False,activation='elu'))
model.add(keras.layers.Dense(1,activation='elu'))
M↓
model.compile(optimizer='adam', loss='huber')
#model.summary()
M↓
history = model.fit(X_train, y_train, epochs=50, batch_size=1, validation_data=(X_valid,y_valid))
```

Fig. 4(n) LSTM Recurrent Neural Network with 6 layers and loss function as Huber and trained for 50 epochs

```
Average error:  [1.56947]
Error percentage:  [0.35185307]
Projected profit percentage:  [0.623385]
Final profit percentage:  0.6008574618717737
Trade error %:  [0.2935396]
```

Fig. 4(o) Final metrics after training model on AXISBANK stock and predicting last 70 days price on 1 hour interval.

The next task that was performed was to create a model for each specific stock as there can't be a general model for every stock present. So we created a program that makes a model which predicts the next interval price of the stock with considering the all 12 indicator values and closing price of the stock and will train the data and save the model, it will train itself with different hyper parameters such as different number of batch sizes for training the model, different number of epochs till which the model has to train itself, it will train itself up until it reaches a defined threshold, i.e. allowed error for model which is set to 1.2% of average of last 7 interval's closing price of the stock.

Next part done was to create a program which will create a model for each indicator for each of the stock and save it. This program makes a dynamic model which makes a random model with

random layers of LSTM, GRU, SimpleRNN and all random numbers of epochs, layers, batch size and it will train itself until it reaches under the defined threshold limit, which is set 10 here.

The parallel models file which will use the dynamic model and train for each indicator in 1 file.

The final program will execute as given below steps.

Upon execution of the program it will give you option of stocks whichever are present in the Nifty 50

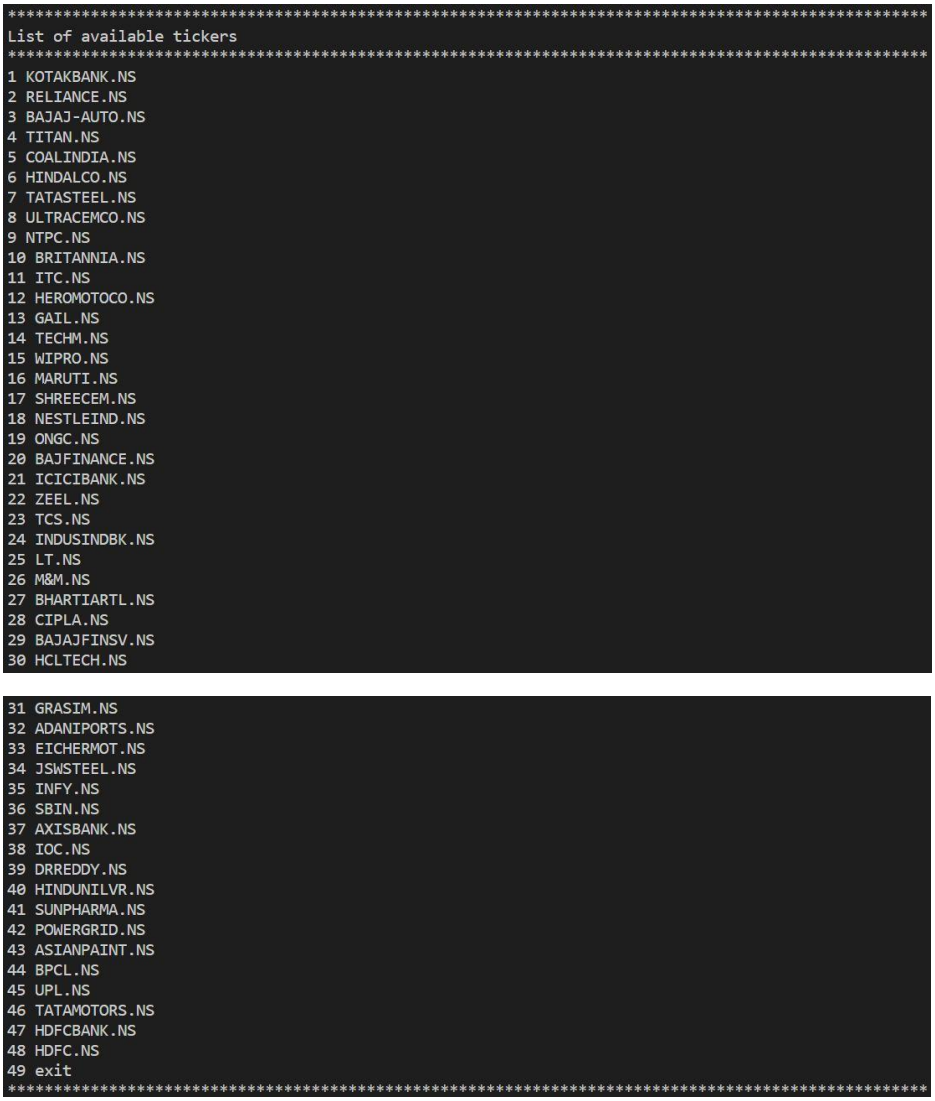

Fig. 4(p) List of all tickers shown.

After the stocks list are obtained it will ask for what ticker you would like to select, after entering the number of the ticker it will show you message which ticker you have selected.

```
**********************************************************************************************************
Select the stock: 36
**********************************************************************************************************

**********************************************************************************************************
Ticker selected is SBIN.NS
**********************************************************************************************************
```

Fig. 4(q) Ticker selected

After the stock is selected it will create a directory named "@temp" in the current working directory of program.This @temp directory is just created to download the saved model of the stock. This @temp would have a one more directory which would be named as the ticker's name. This ticker named folder will save the stock specific model, all 19 models 1 is main model and all 18 others are the indicator specific models

Fig. 4(r) creation of "@temp" folder and stock named folder.

After the ticker is selected the program will download the stock's historical data of specified period which is set as 730 days or last 2 years. The interval of this historical data is that of 1 hour. After the data is downloaded it will give the output of the performance of the stock. It will show what is the difference in the price of the stock price over the period of the data.

```
**********************************************************************************************************
The Past Performance of the stock is:
Percentage change in last 2 years is 44.736%
**********************************************************************************************************
```

Fig. 4(s) Performance of the stock.

After the performance of the stock is shown it will start creating the indicator values and create a file in form of csv which will consist of the closing price, high, low, open and columns which will have indicator values. This file would consist of 27 columns and using this all the model would predict the next possible price.

After the data is created the program will next download the pretrained model of the selected stock. This process first checks if the folder of the ticker exists in the user directory, if it exists than are there models already present in directory, if models are present than it won't download any models otherwise it would download the zip file of model from the drive which consist of 19 models.

```
**********************************************************************************************************
Downloading 1IzRfGt5XfjAe3y-JJSqlMmHb99Uzrnve into C:\Users\Shivam\@temp\SBIN.NS\SBIN.NS.zip... Done.
Unzipping...Done.
**********************************************************************************************************
```

Fig. 4(t) Downloading the stock specific model from drive.

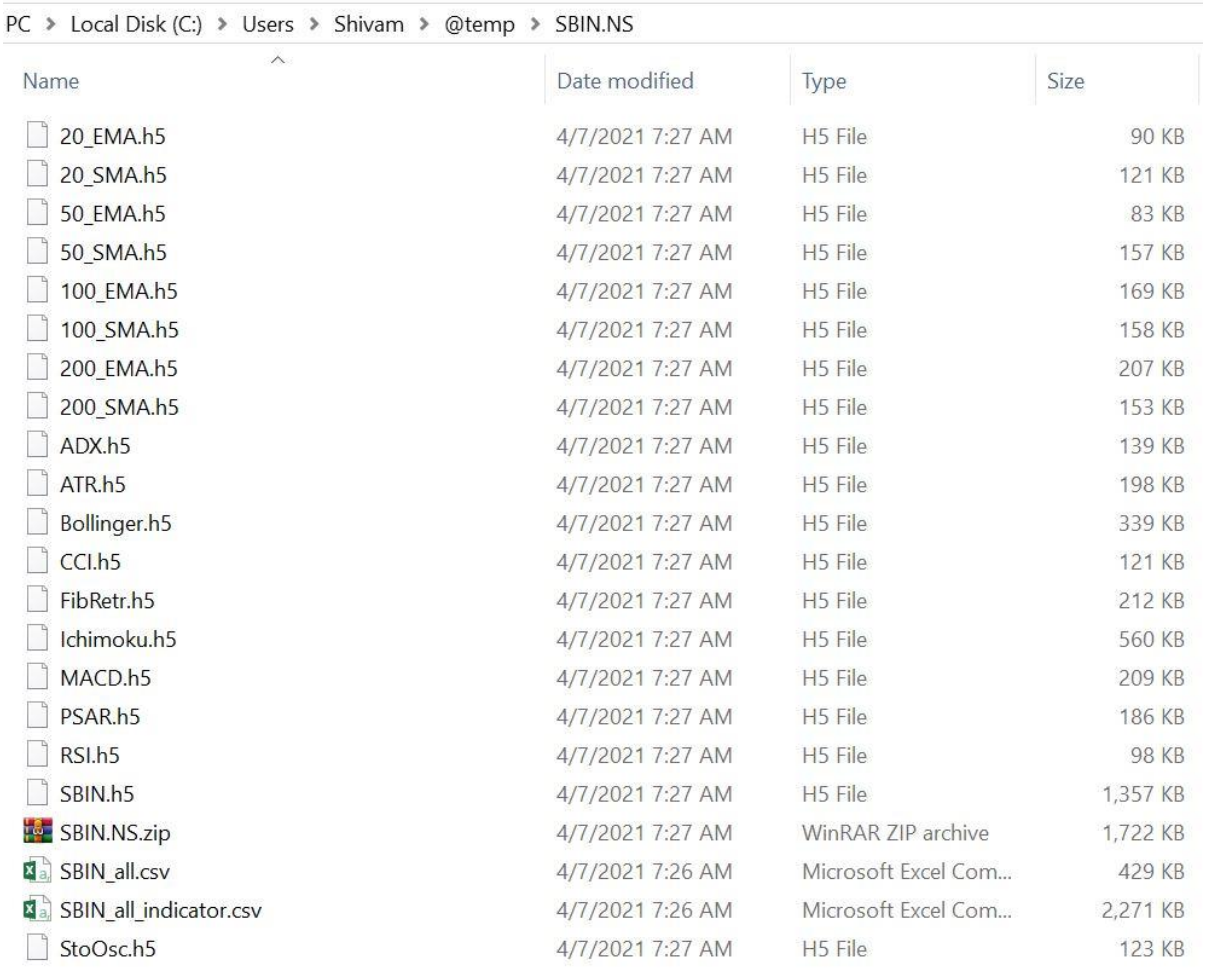

Fig. 4(u) Model downloaded and unzipped and saved in the working directory.

The next process it follows is, it will get the main model of the stock which predicts price using all 12 indicators combined. It will use the model and show us the metrics of the model i.e. average error of the model, error percentage, projected error percentage, final profit percentage and trade error of the model.

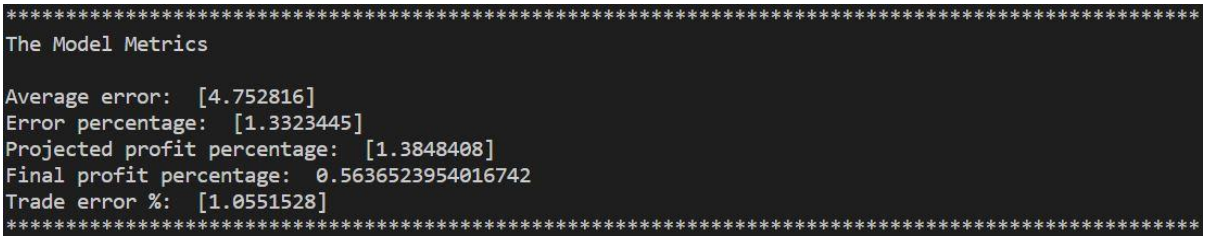

Fig. 4(v) The metrics of the model.

Furthermore, in the program next it would perform the benchmarking of the model which predicts the price with the Nifty index. It would compare the last 10% day intervals and would find how much returns nifty would have given you over that period and how much return our model would have given to us if our model would have been used to predict the prices.

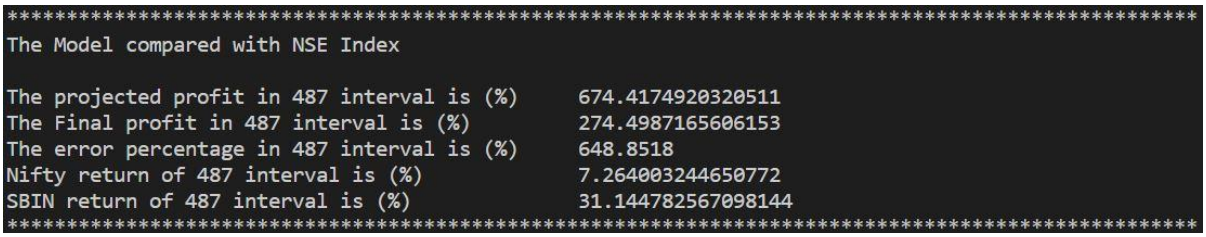

Fig. 4(w) Comparison to Nifty index.

Here if we would have invested in Nifty 487 intervals back, then after this duration we would have gained 7.26% profit, while if our model would strategy would have been used then in the same duration we would have the profit percentage increased by 4 times and reach to 31.14%.

Next the program will give the next interval predicted price of the stock as the output.

Fig. 4(x) Next interval price predicted.

The next strategy we thought was to strengthen our model's confidence by predicting the price of the next interval using all the indicators separately. So next output of the program is to predict the prices according to each indicator. The output would consist the next interval projected price, the difference between the last interval and next possible interval price and the 3rd column would consist of the indication whether to buy or sell the stock in the next interval.

```
**********************************************************************************************************
The prices predicted parallelly from the indicator models is:

                   Prediction  Difference Indication
-----                   -----       -----      -----
mainModelPrice      [[350.96]]   [[-0.44]]       Sell
sma20ModelPrice     [[348.51]]   [[-2.89]]       Sell
sma50ModelPrice     [[350.67]]   [[-0.73]]       Sell
sma100ModelPrice    [[359.42]]    [[8.02]]        Buy
sma200ModelPrice    [[334.57]]  [[-16.83]]       Sell
ema20ModelPrice     [[350.55]]   [[-0.85]]       Sell
ema50ModelPrice     [[359.74]]    [[8.34]]        Buy
ema100ModelPrice    [[360.15]]    [[8.75]]        Buy
ema200ModelPrice    [[349.42]]   [[-1.98]]       Sell
stoOscModelPrice    [[351.97]]    [[0.57]]        Buy
macdModelPrice      [[349.64]]   [[-1.76]]       Sell
bolBandsModelPrice  [[347.88]]   [[-3.52]]       Sell
rsiModelPrice       [[351.17]]   [[-0.23]]       Sell
atrModelPrice       [[346.81]]   [[-4.59]]       Sell
cciModelPrice        [[344.1]]    [[-7.3]]       Sell
ichimokuPrice       [[350.72]]   [[-0.68]]       Sell
adxModelPrice       [[351.44]]    [[0.04]]        Buy
fibRetrModelPrice   [[350.23]]   [[-1.17]]       Sell
psarModelPrice      [[355.56]]    [[4.16]]        Buy
**********************************************************************************************************
```

Fig. 4(y) Prices predicted using the models separately.

Based on the previous part of the program the program will perform next operation which is to how much confidence the model has on buying or selling the stock based on the prices predicted by the indicator models.

```
**********************************************************************************************************
The Buy and Sell Confidence on the stock using Parallel models is
Buy Confidence:  31.58%
Sell Confidence: 68.42%
**********************************************************************************************************
```

Fig. 4(z) Buy and Sell confidence of the model.

Next after completing the buy and sell metrics it would calculate the trade success percentage, it shows if you would have used and traded according to the model for last 10% of data, how much percentage of trades would have gone your way.

```
**********************************************************************************************************
Trade Success Percentage
Trade Success Percentage(for last 10% data): 64.07%
**********************************************************************************************************
```

Fig. 4(aa) Trade success percentage summary.

After completing all the above processes the next part is about the sentiment analysis of the stock. It would give how is the current situation of stocks in news, whether it is positive, negative or neutral.The program will obtain general news and stock specific news, it will return the latest

news from Google, GoogleNews and Moneycontrol website. After news headline is obtained the program would next obtain whole article and summary of the article and after all the data is obtained using the natural language processing the sentiment analysis of the latest news is obtained and program returns the average of the polarity score (factor on which we determine whether it's positive or negative or neutral) for general news and stock specific news.

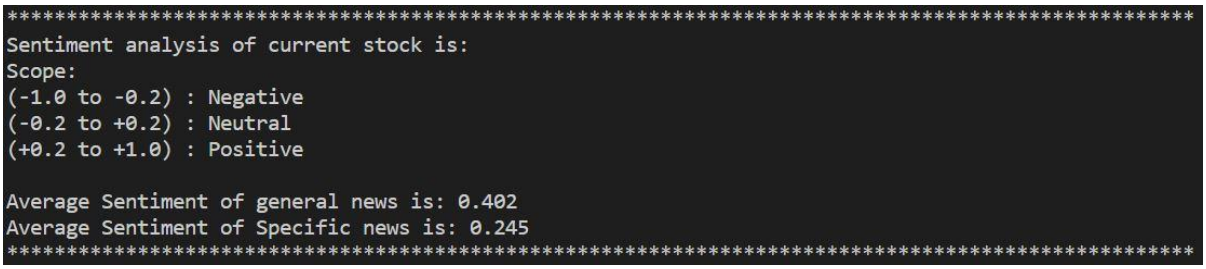

Fig. 4(ab) Sentiment analysis of news of the current stock(SBI).

Additional to this program we also created 2 more programs which we are planning to integrate in future, this programs are the more refined version of the indicator strategies. One program uses the moving average, it is more refined strategy which uses the "Golden cross" and "Death cross" strategy and would give more confident direction of the stock price which way it might go, it uses the simple moving average and exponential moving average. Another program uses the stochastic oscillator indicator and performs same type of the operation.

Chapter 5: Result Analysis

5.1.1 Summary of the results: -

The market functioning may seem very volatile to the ones who are just looking at it from the outside or just looking at the numbers in an abstract without any context. But when we look deeper into it with the help of price and volume movement, we start to get the bigger picture. We have indicators with us to help us quantify these price and volume movements into a signal or zone and that's what our project does to try and predict the price of instruments mainly equity scripts.

These indicators consist of various types such as momentum based, closing price based, volume based etc. and thus help us analyse the price and volume based of a stock in terms of different aspects and generate buy or sell signals.

5.1.2 Advantages of our work: -

- Our project is primarily based for intraday trading and thus have a greater accuracy because of shorter prediction period as compared to investment-based strategies.
- Also, the project currently uses more than 10 indicators thus reducing the scope of error.
- Because of the huge amount of data available the use of neural network exponentially increases the prediction accuracy.
- The platform is not limited to any particular stock and thus can be used by any trader to generate signals of the stock which they prefer to trade.

5.1.3 Usefulness of our work: -

- Most of today's platforms offer strategies which are not time based and rather target based only. We use time as constraint to provide an interval-based trading to help reduce losses due to volatility.
- Existing trading platforms provide select and very few trading ideas as compared to our system which generate signals on an hourly basis on every stock which is available in the database.

- Very few systems use multiple indicators to generate trading signals as compared to ours which has 12 indicators working currently.
- Except very big trading and broking institutions no one uses neural network to keep improving their trading idea's accuracy.

5.1.4 Scope of future work: -

- An efficient way to categorise the indicators according to their bias to generate a collective signal.
- Inclusion of volume and volume-based indicators to add another dimension of viewing the price movements as to whether they are speculative or reliable.
- Inclusion of morning impact of Singaporean Nifty (SGX Nifty) and afternoon impact of opening of European markets (FTSE, DAX, CAC) and full day impact of DOW Futures.
- Transition to semi-automated and then to fully automated system from currently manual trading system.
- Inclusion of impact of costs, slippage, brokering etc.
- Reduction in the error percentage found on the closing prices as shown during the benchmarking of the project.

5.2 Benchmarking of the project

For benchmarking we have taken the 'Nifty Index' and the stock on which our trading system is going to run which in this case is 'Axis Bank'.

```
The projected profit in 486 interval is (%)     302.96511590480804
The Final profit in 486 interval is (%)         292.016726469682
The error percentage in 486 interval is (%)     171.0006
Nifty return of 486 interval is (%)             12.786312952844755
Axisbank return of 486 interval is (%)          7.304547909677654
```

Fig. 5(a) Comparison of trading system with real time index and stock returns

We can clearly observe the huge outperformance of our trading system as compared to real time returns of that stock as well as the index in general. But it comes at the cost of the huge risk of high error percentage which we intend to solve as quickly as possible.

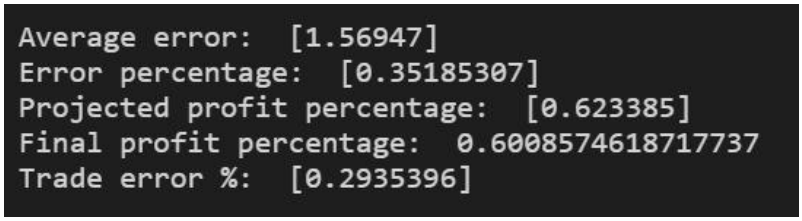

Fig. 5(b) Result of trading system on all quantifiable parameters

"Fig. 5(b)" crunches the huge numbers shown in "Fig. 5(a)" of the performance of our system by showing the parameters in terms of hourly periods with the help of percentages.

Average Error is the mean of error found in the closing prices i.e. actual and predicted. While Error Percentage converts the Average Error in terms of percentage so that it can become comparable to other stocks.

Projected Profit Percentage is the profit that our system predicts we can achieve on an hourly basis while the final profit percentage is the actual profit we earned if we follow the system and Trade error is the unrealized loss/profit while estimating the price that we left on the table.

Chapter 6: Conclusion

The following project is very helpful in creating the surety in predicting the direction of the stock, on which side the stock may go on further. The model is made unique with its feature of using the indicator values with parallelly running the models and predicting separately we get one more kind of surety in selecting the stock for investing in it. The project can be said lightweight cause for each stock the program is downloading the model from the cloud storage so we can use it easily without worrying about the storage. At last the model is also giving the sentiment analysis of the stock specific new analysis which is also great feature integrated inside cause traders need to continuously check for the news to analyze the condition of the stock in the market.

References

The following websites are references through which we have learned many things regarding our project

[1] https://medium.com/engineer-quant/market-invariants-can-there-really-be-predictive-power-in-market-data-bb4999dbe75e : This website shows market invariants.

[2]https://patents.google.com/patent/AU2018101512A4/en?q=Stock+prediction+neural+networks&oq=Stock+prediction+using+neural+networks : Describes a project related parent.

[3] https://www.thepythoncode.com/article/stock-price-prediction-in-python-using-tensorflow-2-and-keras : Describes use of tensorflow

[4] https://www.analyticsvidhya.com/blog/2018/10/predicting-stock-price-machine-learningnd-deep-learning-techniques-python/ : Shows different ML techniques for prediction.