**1. INTRODUCTION**

Predicting how the stock market will perform is one of the most difficult things to do. There are so many factors involved in the prediction – physical factors vs. psychological, rational and irrational behaviour, etc. All these aspects combine to make share prices volatile and very difficult to predict with a high degree of accuracy.

Can we use machine learning as a game-changer in this domain? Using features like the latest announcements about an organization, their quarterly revenue results, etc., machine learning techniques have the potential to unearth patterns and insights we didn’t see before, and these can be used to make unerringly accurate predictions.

The core idea behind this article is to showcase how these algorithms are implemented. I will briefly describe the technique and provide relevant links to brush up on the concepts as and when necessary.

Stock Price Prediction using machine learning helps you discover the future value of company stock and other financial assets traded on an exchange. The entire idea of predicting stock prices is to gain significant profits. Predicting how the stock market will perform is a hard task to do. There are other factors involved in the prediction, such as physical and psychological factors, rational and irrational behaviour, and so on. All these factors combine to make share prices dynamic and volatile. This makes it very difficult to predict stock prices with high accuracy.

**1.1 Definition of the Problem:**

Referring to the given problem we can say that stock is an unpredictable curve that had been in picture ever since. Its essence had been ever long living and indulging. It had grown its popularity with respect to time.

People are more fascinating and interested on the same then before times. Same for the case for the organization. Organization had created it as a better source of revenue generation rather than investing and taking a loan approval from the bank its way efficient and less hectic from the firm point of view.

Stock is unpredictable and it’s been the same from the start. Its way of escalating and deescalating had been phenomenon and experiencing the same is the best integral part of it. It has its upper hand and flexibility with the changes that has the chances of uprising as well as crashing the whole market. It’s easily defined in few words but making an essence and understanding the same is way more hectic and time consuming. Simpler its sound complex is its phenomenon and integrating the same. It has its whole different sets of dependencies and integration from different agents which fluctuate the same in the market. Finding an accurate and getting the exact values out of the same is still unaligned and no particular model of the same is seen in the market value. Finding the closest and getting an accurate proximate value out of such an unpredictability is a problem in itself. Merging of the data getting the best prediction to increase the efficiency alongside considering the different expects of the moderator is tough and we took the same in consideration and implemented with every aspect to generate the best out of the same and get a result that can be better interrupted and the efficiency remains the same with the value of different aspects of creating an impact of reducing the risk and influencing the same over the time period to gain the most out of it.

This is totally based on Machine Learning Algorithm to proceed and provide an effective result. Getting the data and processing it and generating a forecast for three days is the problem statement that we worked on.

Stock market price prediction for short time windows appears to be a random process. The stock price movement over a long period of time usually develops a linear curve. People tend to buy those stocks whose prices are expected to rise in the near future. The uncertainty in the stock market refrain people from investing in stocks. Thus, there is a need to accurately predict the stock market which can be used in a real-life scenario. The methods used to predict the stock market includes a time series forecasting along with technical analysis, machine learning modelling and predicting the variable stock market. The datasets of the stock market prediction model include details like the closing price opening price, the data and various other variables that are needed to predict the object variable which is the price in a given day. The previous model used traditional methods of prediction like multivariate analysis with a prediction time series model. Stock market prediction outperforms when it is treated as a regression problem but performs well when treated as a classification. The aim is to design a model that gains from the market information utilizing machine learning strategies and gauge the future patterns in stock value development. The Support Vector Machine (SVM) can be used for both classification and regression. It has been observed that SVMs are more used in classification based problem like ours. The SVM technique, we plot every single data component as a point in space (where n is the number of features of the dataset available) with the value of feature being the value of a particular coordinate and, hence classification is performed by finding the hyperplane that differentiates the two classes explicitly. Predictive methods like Random forest technique are used for the same. The random forest algorithm follows an ensemble learning strategy for classification and regression. The random forest takes the average of the various subsamples of the dataset, this increases the predictive accuracy and reduces the over-fitting of the dataset.

**1.2 LIABRARIES**

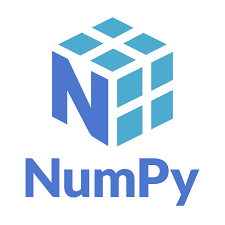
The model needs some libraries with specific functionality.

Following are the ones that are used in this project.

1) NumPy:

NumPy is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning. It is particularly useful for linear algebra, Fourier transform, and random number capabilities. High-end libraries like

Tensor Flow uses NumPy internally for manipulation of Tensors.



2) Scikit-learn:

Scikit-learn is one of the most popular ML libraries for classical ML algorithms. It is built on top of two basic Python libraries, viz., NumPy and SciPy. Scikit-learn supports most of the supervised and unsupervised learning algorithms. Scikit-learn can also be used for data-mining and data-analysis, which makes it a great tool who is starting out with ML.



3) Pandas:

Pandas is a popular Python library for data analysis. It is not directly related to Machine Learning. As we know that the dataset must be prepared before training. In this case, Pandas comes handy as it was developed specifically for data extraction and preparation. It provides high-level data structures and wide variety tools for data analysis. It provides many inbuilt methods for groping, combining and filtering data.



4) Matplotlib:

Matplotlib is a very popular Python library for data visualization. Like Pandas, it is not directly related to Machine Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots. A module named pyplot makes it easy for programmers for plotting as it provides features to control line styles, font properties, formatting axes, etc. It provides various kinds of graphs and plots for data visualization, viz., histogram, error charts, bar chats, etc.



**2. LITERATURE REVIEW**

**2.1 PROBLEM DEFINATION**

Stock market prediction is basically defined as trying to determine the stock value and offer a robust idea for the people to know and predict the market and the stock prices. It is generally presented using the quarterly financial ratio using the dataset. Thus, relying on a single dataset may not be sufficient for the prediction and can give a result which is inaccurate. Hence, we are contemplating towards the study of machine learning with various datasets integration to predict the market and the stock trends. The problem with estimating the stock price will remain a problem if a better stock market prediction algorithm is not proposed. Predicting how the stock market will perform is quite difficult. The movement in the stock market is usually determined by the sentiments of thousands of investors. Stock market prediction, calls for an ability to predict the effect of recent events on the investors. These events can be political events like a statement by a political leader, a piece of news on scam etc. It can also be an international event like sharp movements in currencies and commodity etc. All these events affect the corporate earnings, which in turn affects the sentiment of investors. It is beyond the scope of almost all investors to correctly and consistently predict these hyperparameters. All these factors make stock price prediction very difficult. Once the right data is collected, it then can be used to train a machine and to generate a predictive result.

Accurately predicting the stock market is a challenging task, but the modern web has proved to be a very useful tool in making this task easier. Due to the interconnected format of data, it is easy to extract certain sentiments thus making it easier to establish relationships between various variable and roughly scope out a pattern of investment. Investment pattern from various firms show sign of similarity, and the key to successfully predicting the stock market is to exploit these same consistencies between the data sets. The way stock market information can be predicted successfully is by using more than just technical historical data, and using other methods like the use of sentiment analyser to derive an important connection between people’s emotions and how they are influenced by investment in specific stocks. One more important segment of the prediction process was the extraction of important events from web news to see how it affected stock prices.

**2.2 SOFTWARE REQUIREMENT SPECIFICATION**

**2.2.1 FUNCTIONAL REQUIREMENTS:**

Functional requirements deal with the functionality of the software in the engineering view. The component flow and the structural flow of the same is enhanced and described by it.

The functional statement deals with the raw datasets that are categorized and learning from the same dataset. Later the datasets are categorized into clusters and the impairment of the same is checked for the efficiency purpose. After the dataset cleaning the data are cleansed and the machine learns and finds the pattern set for the same it undergoes various iteration and produce output.

**2.2.2 NON-FUNCTIONAL REQUIREMENTS:**

Non functional requirement deals with the external factors which are non functional in nature It is used for analysis purpose. Under the same the judgment of the operations is carried out for its performance. Stock is feasible and is ever changing so these extra effects and the requirements helps it to get the latest updates and integrate in a one goes where the technicians can work on and solve a bug or a draft if any.

The non-functional requirements followed are it's efficiency and hit gain ratio. The usability of the code for the further effectiveness and to implement and look for the security console. The System is reliable and the performance is maintained with the support of integration and portability of the same.

**3. DATASETS**

## 3.1 The dataset

## This section details the data that was extracted from the public data sources, and the final dataset that was prepared. Stock market-related data are diverse, so we first compared the related works from the survey of financial research works in stock market data analysis to specify the data collection directions. After collecting the data, we defined a data structure of the dataset. Given below, we describe the dataset in detail, including the data structure, and data tables in each category of data with the segment definitions.

### **3.2 Description of our dataset**

### In this section, we will describe the dataset in detail. This dataset consists of certain stocks from the Indian stock market. Besides the daily price data, daily fundamental data of each stock ID, we also collected the suspending and resuming history, top 10 shareholders, etc. We list two reasons that we choose 2 years as the time span of this dataset: (1) most of the investors perform stock market price trend analysis using the data within the latest 2 years, (2) using more recent data would benefit the analysis result.

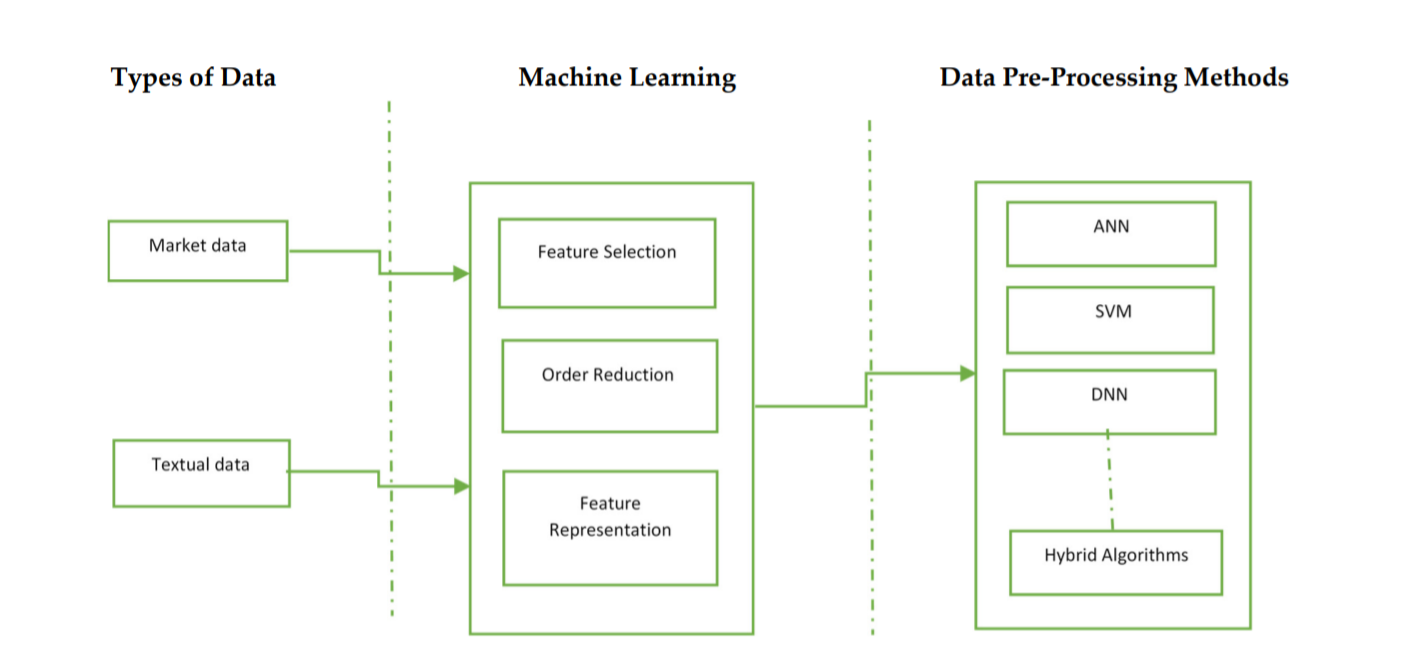
We used 20% of the dataset for training and 80% data for testing.

Once the data is available, it needs some pre-processing so that it can be fed to a machine learning model. The significance of the output depends on the pre-processing of the data. The textual data must be transformed into a structured format that can be used in a machine learning model. The previous studies revealed that there are three significant pre-processing steps, i.e., feature selection, order reduction and the representation of features.

**4. DATA PREPROCESSING**

Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this, we use data pre-processing task.

****

1. Block Diagram of Data pre-processing

**4.1 Why do we need Data Pre-processing?**

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

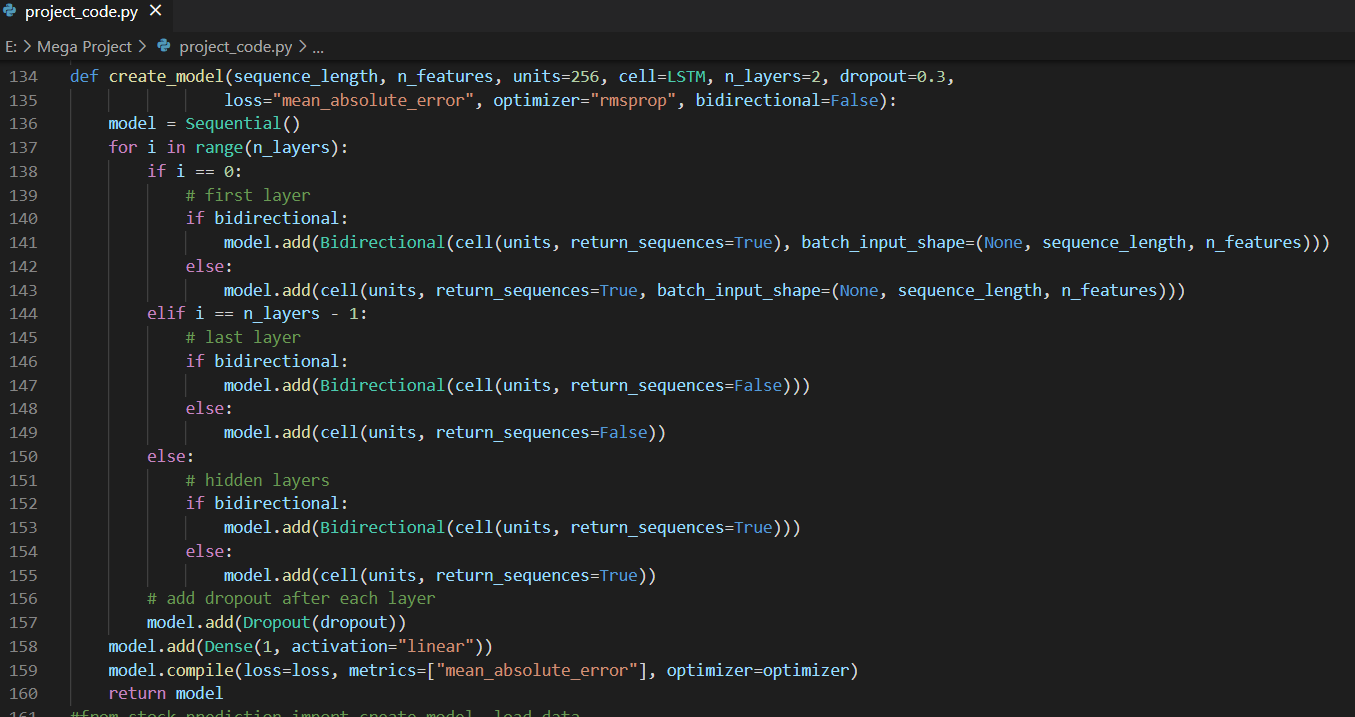
It involves below steps:

* **Getting the dataset**
* **Importing libraries**
* **Importing datasets**
* **Finding Missing Data**
* **Encoding Categorical Data**
* **Splitting dataset into training and test set**
* **Feature scaling**
  1. **Parameters used in code:**
* TEST\_SIZE : The testing set rate. For instance,0.2 means 20% of the total dataset.
* FEATURES\_COLUMNS : The features we going to use to predict the next price value

.

* N\_LAYERS : Number of RNN layers to use.
* CELL : RNN cell to use, default is LSTM.
* UNITS : Number of cell units.
* DROPOUT : The [dropout](https://www.thepythoncode.com/article/dropout-regularization-in-pytorch) rate is the probability of not training a given node in a layer, where 0.0 means no dropout at all. This regularization can help the model not overfit our training data. Check [this tutorial](https://www.thepythoncode.com/article/dropout-regularization-in-pytorch) for more information about dropout regularization.
* BIDIRECTIONAL : Whether to use [bidirectional recurrent neural networks](https://en.wikipedia.org/wiki/Bidirectional_recurrent_neural_networks).
* LOSS : Loss function to use for this regression problem, we're using [Huber loss](https://www.tensorflow.org/api_docs/python/tf/keras/losses/Huber), you can use mean absolute error or mean squared error as well.
* OPTIMIZER : Optimization algorithm to use, defaulting to [Adam](https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam).
* BATCH\_SIZE : The number of data samples to use on each training iteration.
* EPOCHS : The number of times the learning algorithm will pass through the entire training dataset, we used 500 here, but try to increase it further.

**4.3 MODEL CREATION:**

Now that we have a proper function to load and prepare the dataset, we need another core function to build our model:

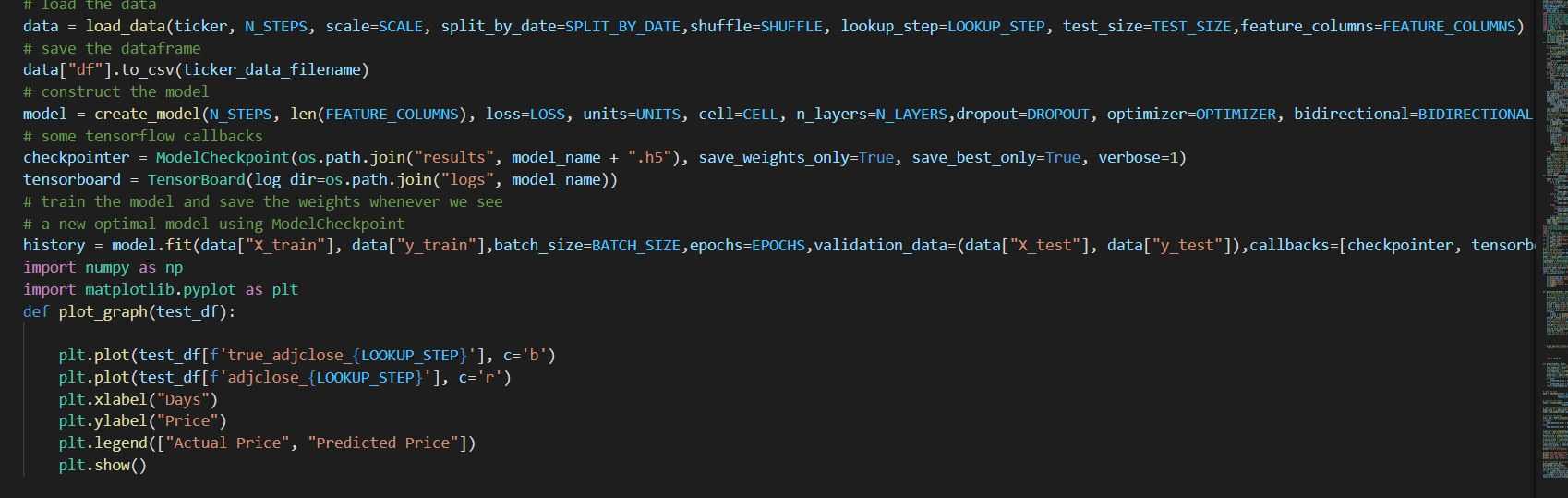
The above function constructs an RNN with a dense layer as an output layer with one neuron. This model requires a sequence of features of sequence\_length (in this case, we will pass 50 or 100) consecutive time steps (which are days in this dataset) and outputs a single value which indicates the price of the next time step.

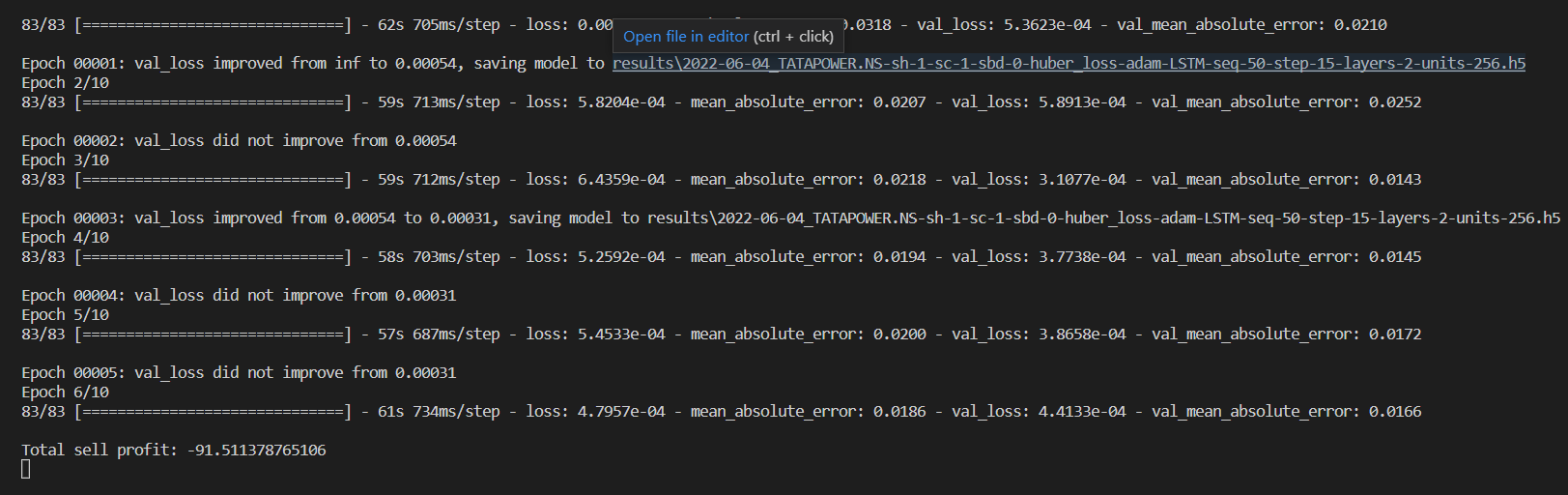
It also accepts n\_features as an argument, which is the number of features we will pass on each sequence, in our case, we'll pass adjclose, open, high, low and volume columns (i.e 5 features).

You can tweak the default parameters as you wish, n\_layers is the number of RNN layers you want to stack, dropout is the dropout rate after each RNN layer, units are the number of RNN cell units (whether it is LSTM, SimpleRNN, or GRU), bidirectional is a boolean that indicates whether to use bidirectional RNNs, experiment with those!

## 4.4Training the Model

Now that we have all the core functions ready, let's train our model.

We used [ModelCheckpoint](https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/ModelCheckpoint?version=stable), which saves our model in each epoch during the training. We also used [TensorBoard](https://www.tensorflow.org/tensorboard) to visualize the model performance in the training process.

After running the above block of code, it will train the model for 50 epochs (as we set previously), so it will take some time. Here are the first output lines:

**5. Algorithms**

**5.1 RNN(Recurrent neural network):**

A Deep Learning approach for modelling sequential data is **Recurrent Neural Networks (RNN)**. RNNs were the standard suggestion for working with sequential data before the advent of attention models. Specific parameters for each element of the sequence may be required by a deep feedforward model. It may also be unable to generalize to variable-length sequences.

Recurrent Neural Networks use the same weights for each element of the sequence, decreasing the number of parameters and allowing the model to generalize to sequences of varying lengths. RNNs generalize to structured data other than sequential data, such as geographical or graphical data, because of its design.

Neural networks imitate the function of the human brain in the fields of AI, machine learning, and deep learning, allowing computer programs to recognize patterns and solve common issues.

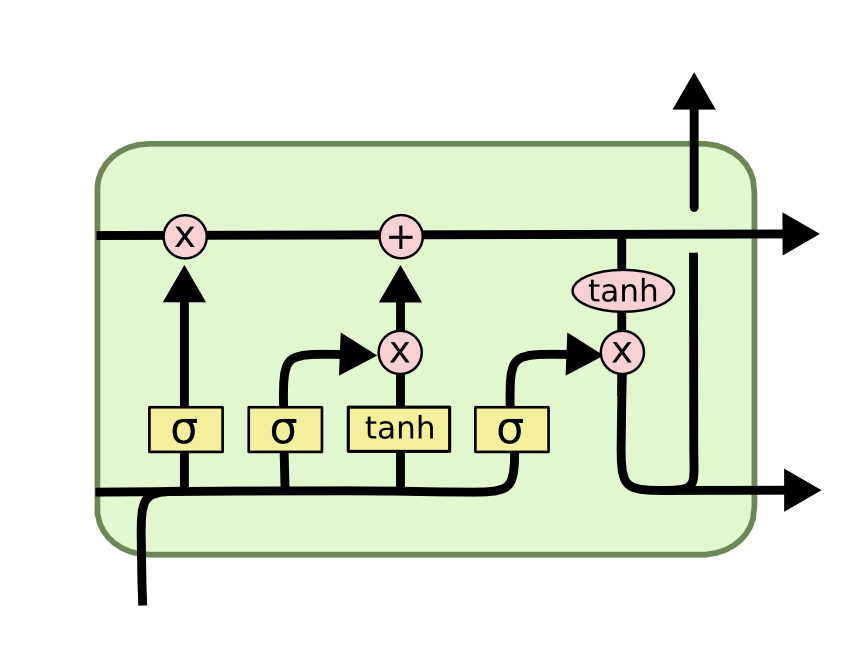
RNNs are a type of neural network that can be used to model sequence data. RNNs, which are formed from feedforward networks, are similar to human brains in their behaviour. Simply said, recurrent neural networks can anticipate sequential data in a way that other algorithms can’t.

**5.2 Long Short-Term Memory (LSTM):**

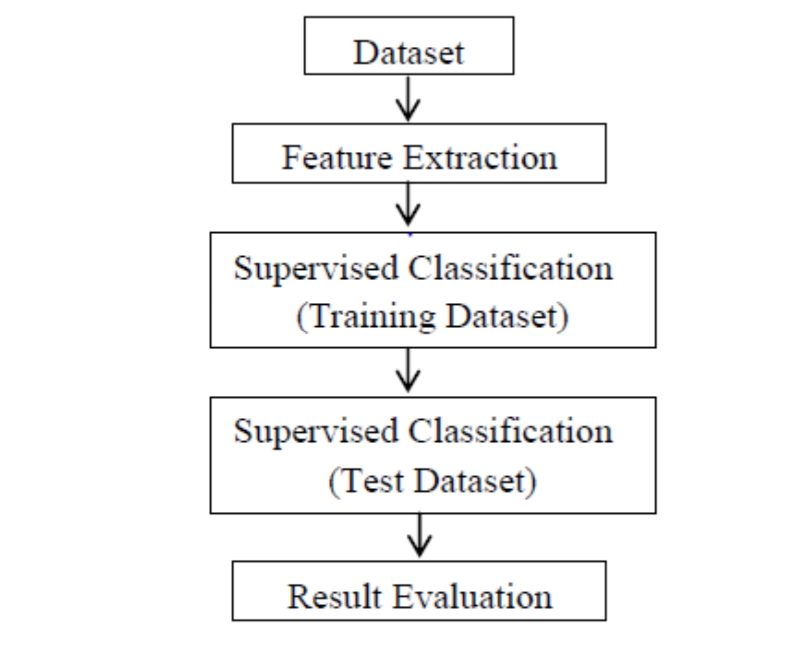
Long Short-Term Memory (LSTM) networks are a type of **recurrent neural network** capable of learning order dependence in sequence prediction problems. Sequence prediction problems have been around for a long time. They are considered as one of the hardest problems to solve in the data science industry. These include a wide range of problems; from predicting sales to finding patterns in stock markets’ data, from understanding movie plots to recognizing your way of speech, from language translations to predicting your next word on your iPhone’s keyboard. With the recent breakthroughs that have been happening in data science, it is found that for almost all of these sequence prediction problems, Long short Term Memory networks, LSTMs have been observed as the most effective solution. LSTMs have an edge over conventional feed-forward neural networks and RNN in many ways. This is because of their property of selectively remembering patterns for long durations of time. The purpose of this article is to explain LSTM and enable us to use it in real life problems.

LSTMs on the other hand, make small modifications to the information by multiplications and additions. With LSTMs, the information flows through a mechanism known as cell states. This way, LSTMs can selectively remember or forget things. The information at a particular cell state has three different dependencies. Industries use them to move products around for different processes. LSTMs use this mechanism to move information around.

We may have some addition, modification or removal of information as it flows through the different layers, just like a product may be moulded, painted or packed while it is on a conveyor belt.



**2. LSTM cell diagram**

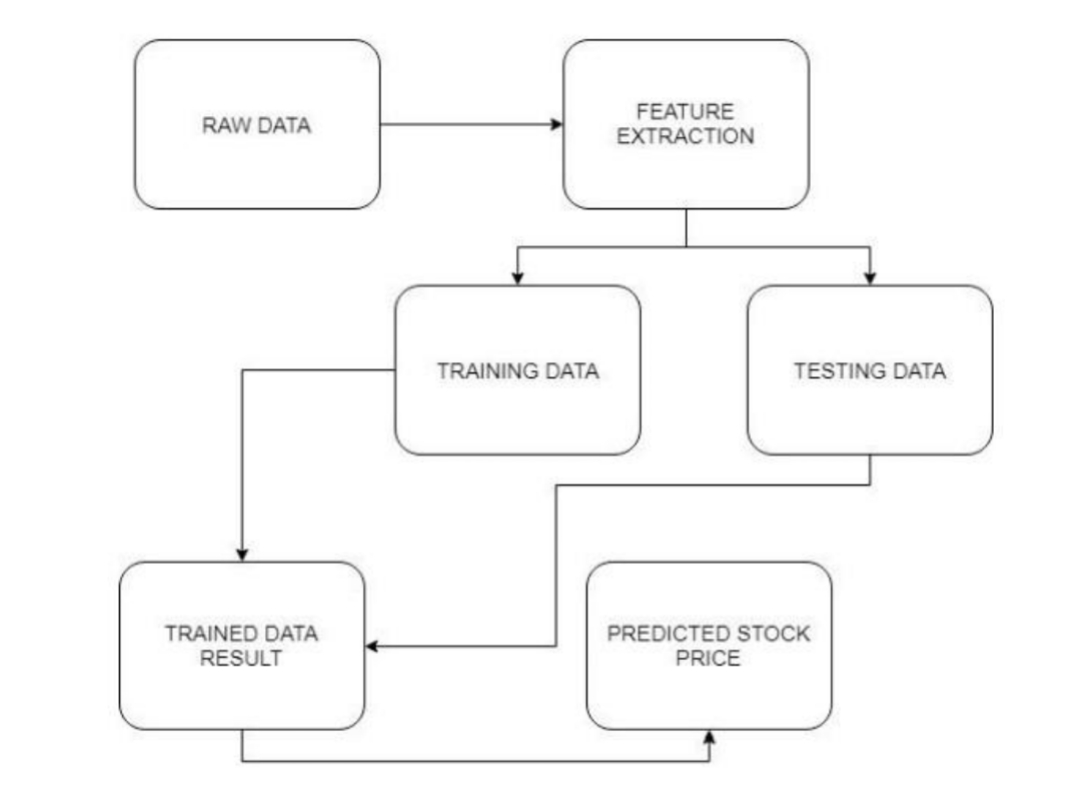
**5.3 The System Flow Chart:**

3. System Flow Chart

Here as described in the figure above, the proposed system will have an input from the dataset which will be extracted featured wise and Classified underneath. The classification technique used is supervised and the various techniques of machine level algorithms are implemented on the same.

Training Dataset are created for training the machine and the test cases are derived and implemented to carry out the visualization and the plotting’s. The result generated are passed and visualized in the graphical form.

**5.4 System Architecture:**



4. System Architecture

The above figure gives the demonstration on the dataset extraction and

refining the raw dataset by categorizing into two phases of training and testing data.

From the given dataset a well modified categorization is extracted and a graph

set is plotted to gain the required output which gives the stock prediction range.

**6. MODEL EVALUATION AND VALIDATION**

Once the model is built, the next step is to evaluate and validate the model. Model Evaluation is an essential part of the modeldevelopment process. It is used to test the final performance of the algorithm and is done on the test set. Also, it helps to find the best model that represents your data and how well the chosen model will work in the future.

Model validation is the set of processes and activities intended to confirm that models are performing as expected. Effective validation helps you to ensure that models are sound. Also, it identifies potential limitations and speculations, and assesses their possible impact.

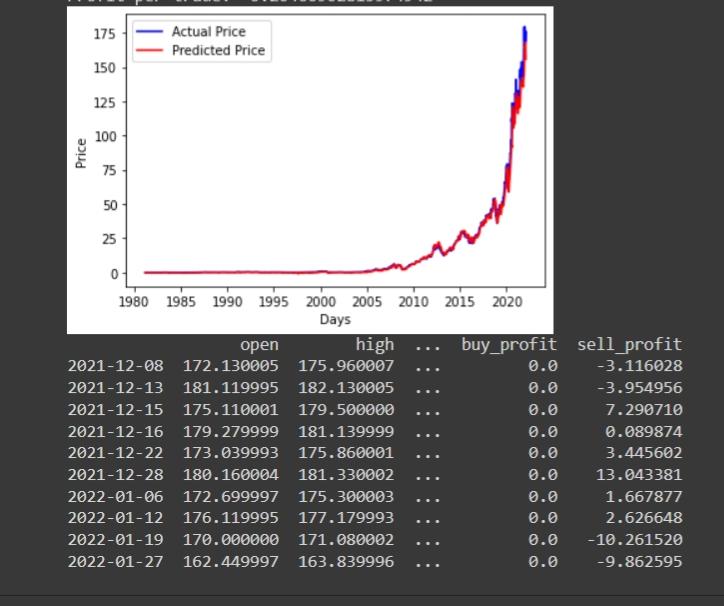
There are multiple measures that can be used to find out how good a regression model is predicting or how good a classifier is classifying the data.

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic and overfitted models. There are two methods of evaluating models in data science, Hold-Out and Cross-Validation. To avoid overfitting, both methods use a test set (not seen by the model) to evaluate model performance.

**Hold-Out:**In this method, the mostly large dataset is *randomly* divided to three subsets:

1. **Training set** is a subset of the dataset used to build predictive models.
2. **Validation set** is a subset of the dataset used to assess the performance of model built in the training phase. It provides a test platform for fine tuning model’s parameters and selecting the best-performing model. Not all modelling algorithms need a validation set.
3. **Test set** or unseen examples is a subset of the dataset to assess the likely future performance of a model. If a model fit to the training set much better than it fits the test set, overfitting is probably the cause.

**7. RESULTS AND DISCUSSIONS**



* In the above figure we can see the graph we got from the results show us the actual price and the predicted prices in their corresponding colours of blue and red respectively.
* By taking a particular stock we can also get its opening price and the high price of the stock(the moment at which the price hit its max. throughout the day) and the buying profit and the selling profit.
* **Buy/Sell profit:** This is the profit we get if we opened trades on all the testing samples, we calculated these on **get\_final\_df()** function.
* **Total profit:** This is simply the sum of buy and sell profits.
* **Profit per trade**: The total profit divided by the total number of testing samples.
* **Accuracy score**: This is the score of how accurate our predictions are. This calculation is based on the positive profits from all the trades from the testing samples.

**7.1 How many times is required to train the model?**

In code we have taken 50 epochs, 64 batch size and different parameters. That’s why It might take about 1-2 hours of time for  training the model. And also because of time series data it can takes a more than 2 hours. In case if we are facing longer time then we can upgrade our model. Longer training time is expected if used a greater number of hidden layers.

**7.2 Hardware Requirement:**

The Stock Price Prediction project is simple, reliable, and a user-friendly project which can result to profitable outcome for an individual who is interested to invest in stocks and for the ones who are expecting for higher gains from the stock market.

The functional/ hardware requirements needed for the user are as follows:

Processor : Intel i5 or above

RAM : Minimum 225MB or more.

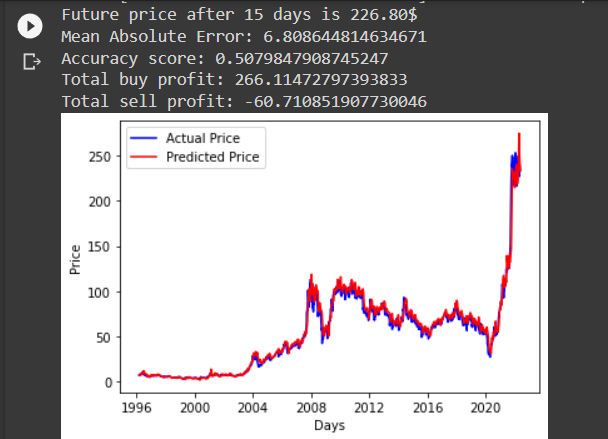
Hard Disk : Minimum 2 GB of space

Input Device : Keyboard

Output Device : Screens of Monitor or a Laptop

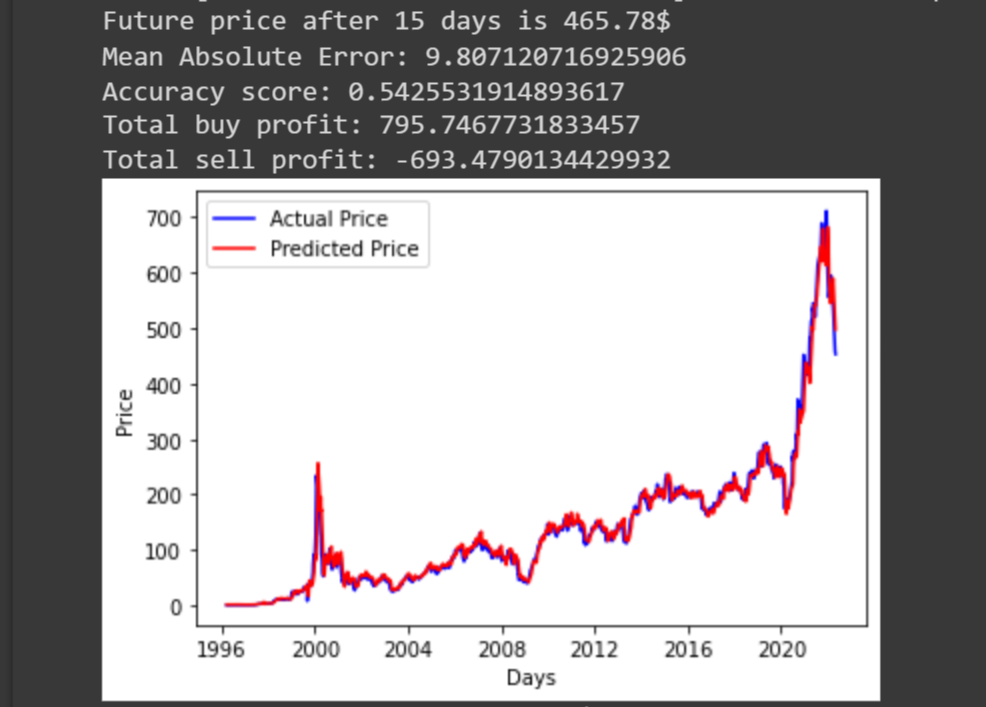
**8. CASE STUDIES**

**8.1 RESULT OF TATA POWER:**



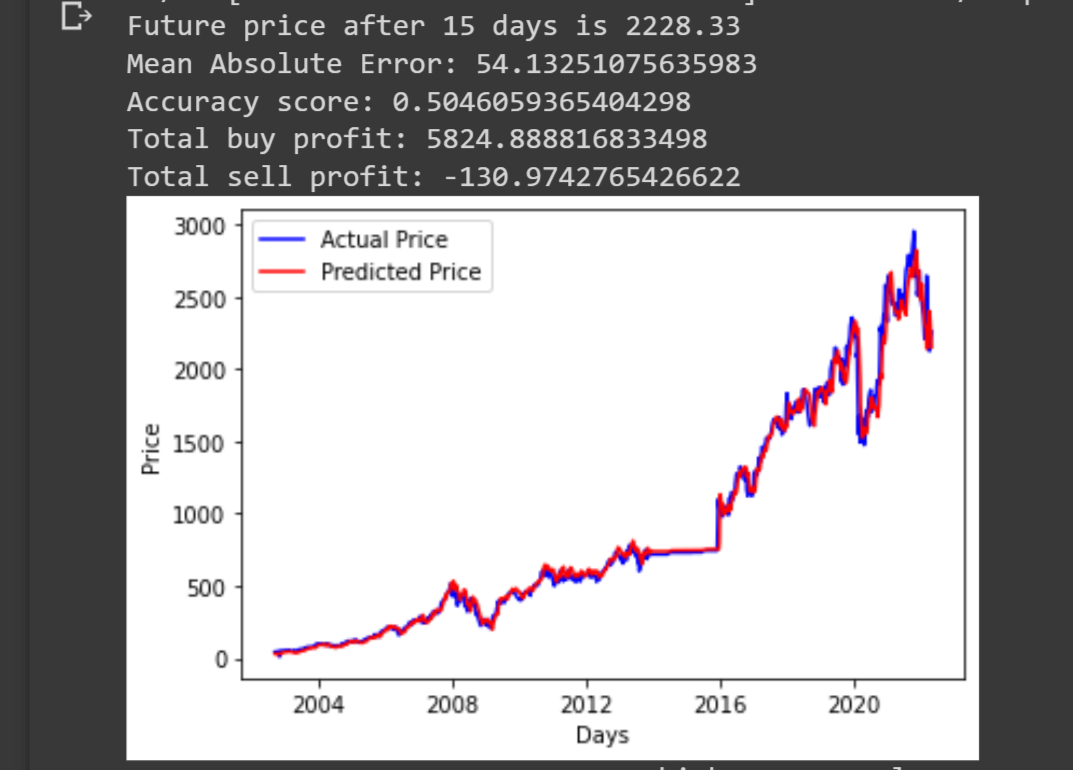


**8.2 RESULT OF WIPRO:**





**8.3 RESULT OF HDFC:**



Here we have showed the Tata power, Wipro and HDFC graph of actual price verses predicted price. We have predicted the future price after 15 days. And also, we have calculated the total buy profit and total sell profit using the past 50 days stock price values.

**9. CONCLUSION**

However, with the introduction of Machine Learning and its strong algorithms, the most recent market research and Stock Market Prediction advancements have begun to include such approaches in analysing stock market data. The Opening Value of the stock, the Highest and Lowest values of that stock on the same days, as well as the Closing Value at the end of the day, are all indicated for each date. With this information, it is up to the job of a Machine Learning Data Scientist to look at the data and develop different algorithms that may help in finding appropriate stocks values.

Predicting the stock market was a time-consuming and laborious procedure a few years or even a decade ago. However, with the application of machine learning for stock market forecasts, the procedure has become much simpler. Machine learning not only saves time and resources but also outperforms people in terms of performance. it will always prefer to use a trained computer algorithm since it will advise you based only on facts, numbers, and data and will not factor in emotions or prejudice.

**9. REFERENCES**

[1] Stock Price Prediction Using LSTM on Indian Share Market by Achyut Ghosh, Soumik Bose1, GiridharMaji, Narayan C. Debnath, Soumya Sen

[2] S. Selvin, R. Vinayakumar, E. A. Gopalkrishnan, V. K. Menon and K. P. Soman - Stock price predictionusing LSTM, RNN and CNN-sliding window model - 2017.

[3] Murtaza Roondiwala, Harshal Patel, Shraddha Varma, “Predicting Stock Prices Using LSTM” in Undergraduate Engineering Students, Department of Information Technology, Mumbai University, 2015.

[4] Xiongwen Pang, Yanqiang Zhou, Pan Wang, Weiwei Lin, “An innovative neural network approach for stock market prediction”, 2018

[5] Ishita Parmar, Navanshu Agarwal, Sheirsh Saxena, Ridam Arora, Shikhin Gupta, Himanshu Dhiman, Lokesh Chouhan Department of Computer Science and Engineering National Institute of Technology, Hamirpur – 177005, INDIA - Stock Market Prediction Using Machine Learning.

[6] Pranav Bhat Electronics and Telecommunication Department, Maharashtra Institute of Technology, Pune. Savitribai Phule Pune University - A Machine Learning Model for Stock Market Prediction.

[7] Anurag Sinha Department of computer science, Student, Amity University Jharkhand Ranchi, Jharkhand (India), 834001 - Stock Market Prediction Using Machine Learning.

[8] V Kranthi Sai Reddy Student, ECM, Sreenidhi Institute of Science and Technology, Hyderabad, India - Stock Market Prediction Using Machine Learning. epochs Accuracy MSE RMSE 10 96.25328 4839.5690 69.56701 20 97.63884 2653.1278 51.50852 30 98.19937 1650.3337 40.62430 40 98.13571 1616.9295 40.21106 50 98.37254 1361.8098 36.90270 Mukt Shabd Journal ISSN NO : 2347-3150 Volume X, Issue VI, JUNE/2021 Page No : 442 442

[9] Asset Durmagambetov currently works at the mathematics, CNTFI- 'The Riemann HypothesisMillennium Prize Problems' - stock market predictions.

[10] Mariam Moukalled Wassim El-Hajj Mohamad Jaber Computer Science Department American University of Beirut - Automated Stock Price Prediction Using Machine Learning.

[11] Nikfarjam, A. , Emadzadeh, E., and Muthaiyah , S. (2010). Text mining approaches for stock market prediction. In Computer and Automation Engineering (ICCAE), 2010

[12]Podding, T and Rehkugler, H, “A world of integrated financial markets using artificial neural networks”, neurocomputing, pages 251-273, 1996.