Jaypee Institute Of Information Technology

Minor Project 1

Team Members

Ashita Agrawal(19103249)

Anshika Patel(19103251)

Samarth Kumar Shah(19103260)

Stock Price Prediction

Using Machine Learning And Neural Network

ABSTRACT

The stock market is generally very unpredictable in nature. There are many factors that might be responsible for determining the price of a particular stock such as the market trend, supply and demand ratio, global economy, public sentiments, sensitive financial information, earning declaration, historical price and many more. These factors explain the challenge of accurate prediction. But, with the help of new technologies like data mining and machine learning, we can analyze big data and develop an accurate prediction model that avoids some human errors.

INTRODUCTION

The stock market refers to public markets that exist for issuing, buying, and selling stocks that trade on a stock exchange or over-the-counter.

A Stock Market allows us to buy and sell units of stock of a company.

If the company's profit goes up, then we own some of the profits and if they go down,then we lose profit with them.

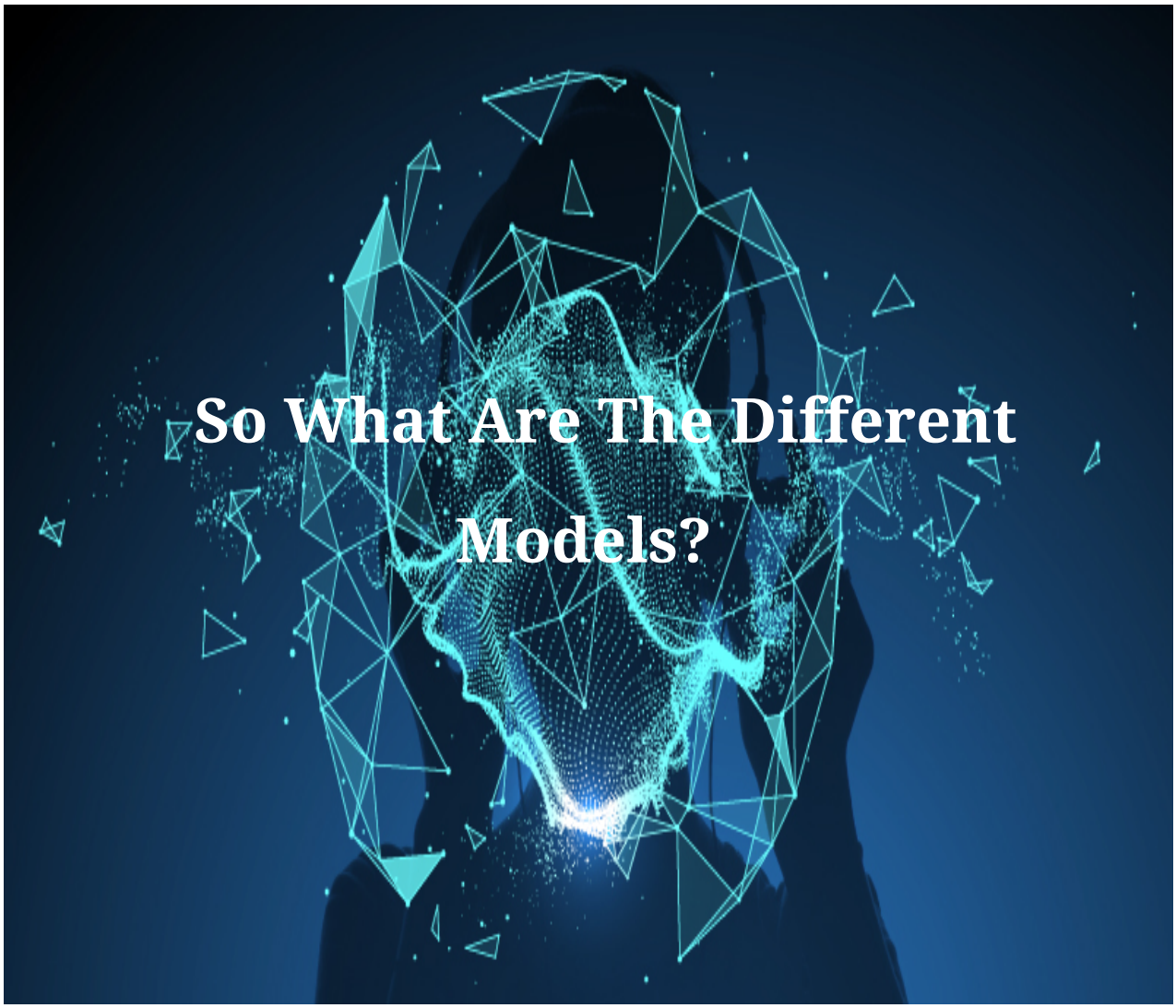
The use of algorithms to make trading decisions has become a prevalent practice in major stock exchanges of the world.

HOW TO READ A STOCK TABLE?


Company or Fund being traded
Interval
Ask
Net CHG
Previous Close
High
Volume
SPDR S&P 500 ETF SPY - 5 min B=281.12 A=281.13 L=281.13 2.09 0.75% PC=279.04 O=280.99 Hi=282.18 Lo=279.56 V=67,834,928
Ticker Symbol
Bid
Last Trade
Net% CHG
Open
Low
282.00
281.03
280.00
279.00
278.00
10:00
11:00
12:00
13:00
14:00
3/26
10:00
11:00
12:00
13:00
14:00
3/27

PROBLEM STATEMENT

1. To accurately predict the future closing value of a given stock across a given period of time in the future.
2. Use different machine learning and deep learning models available and compare them in terms of graphical analysis.



TYPES AND CLASSIFICATION OF MACHINE LEARNING

Machine learning algorithms can be categorized according to the types of data, whether they give feedback or a learning signal. Four types of machine learning are described below.

1. **Supervised machine learning** - Supervised Learning uses labeled data to predict future label data or events with some given features. If the label data is continuous then it is called a regression problem and if label data is categorical it is a classification problem. Sometimes it is called binary classification.

2. Unsupervised machine learning - Unsupervised learning uses information that is neither classified nor labeled. This is called non-structure data. We can group 3 similar types of things together from the data. We can call these either clustering or association problems.

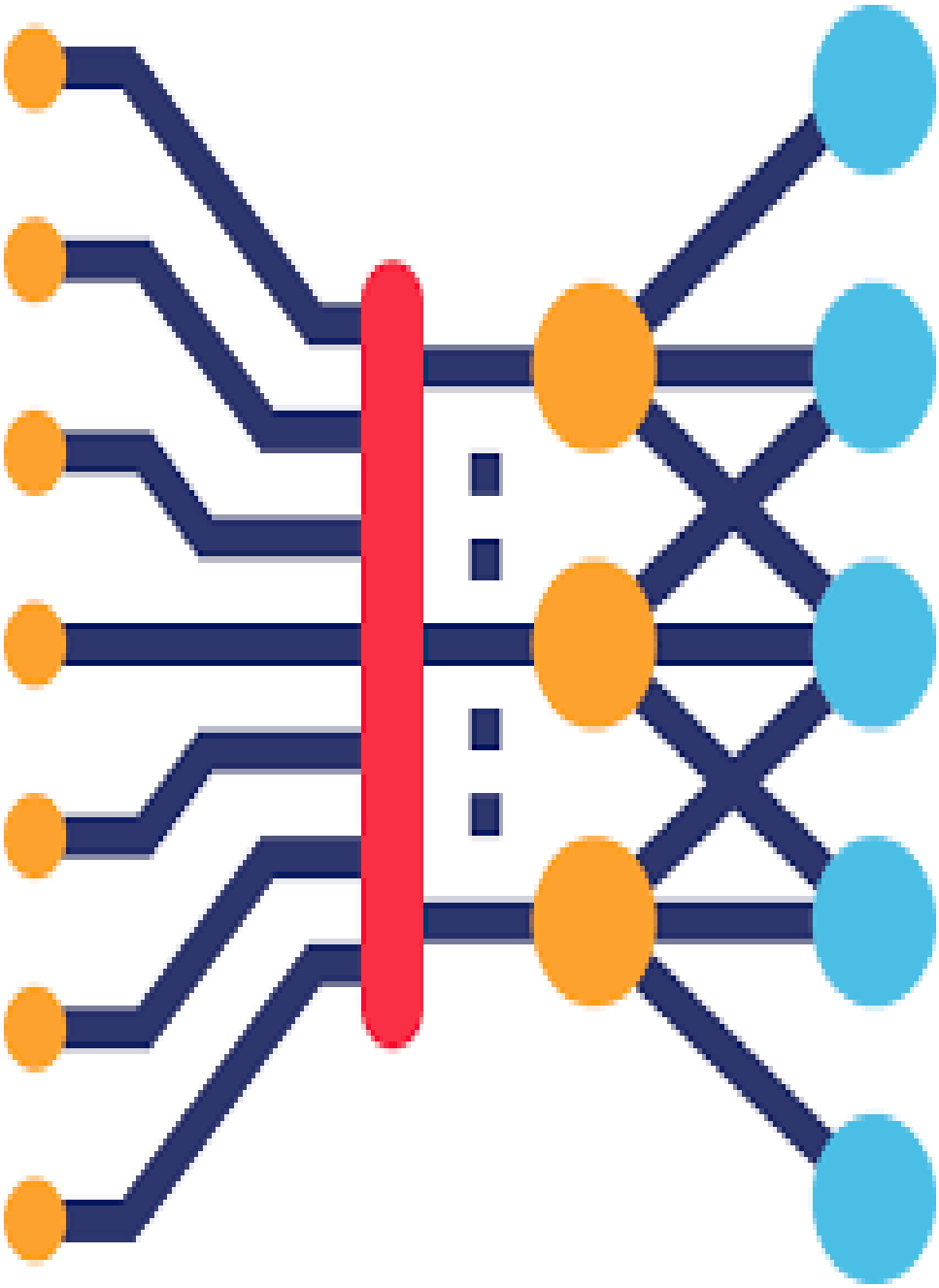
3. Semi-supervised machine learning - When the data are a combination of supervised and unsupervised learning, mostly a large amount of unlabeled data and a smaller amount of label data, then the problem falls under the Semi-supervised learning category. Under this method, the system is able to considerably improve learning accuracy.

4. Reinforcement machine learning - Reinforcement learning interacts with environments and works through trial and error to find which actions provide the greatest rewards.

Time-Series Analysis

A time series is a series of data that is collected over a period of time. Time series data are sequential data which follow some patterns. In order of time, data are points in an index or listed or graphed. Time series data are also called historical data or past data. Time series data are used for predicting a future value based on an historical value. This is called **time series analysis** . The daily closing price of stocks, heights of ocean tides, and counts of sunspots are some examples of time series data. Time series data are studied for several purposes, such as forecasting the future based on knowledge of the past, and understanding of the phenomenon. Underlying measures, or simply succinctly describing the salient features of the series. Forecasting or predicting future prices of an observed time series plays an important role in nearly all fields of science, engineering, finance, business intelligence, economics, meteorology, telecommunications etc. . To predict an outcome based on time series data, we can use **regression analysis, KNN algorithm , random forest algorithm,Recurrent Neural Networks (RNN) , LSTM** are widely used for analysis on time-series data.

APPROACHES TO SOLVE



LINEAR REGRESSION

KNN ALGORITHM

RANDOM FOREST

ARIMA

LSTM

Linear regression

1. Linear regression strives to show the relationship between two variables by applying a linear equation to observed data and is a useful measure for technical and quantitative analysis in financial markets.One variable is supposed to be an independent variable, and the other is to be a dependent variable.

2. Plotting stock prices along a normal distribution—bell curve—can allow traders to see when a stock is overbought or oversold.

3. Using linear regression, a trader can identify key price points—entry price, stop-loss price, and exit prices.

4. A stock's price and time period determine the system parameters for linear regression, making the method universally applicable.

How can we implement it ??

1.)The first step that was performed was to fetch the values or to download them as a CSV file.

2.)The obtained data-frame had two columns namely, Date and Close which were initially plotted onto the graph using the plot() functions.

Regression Formula:

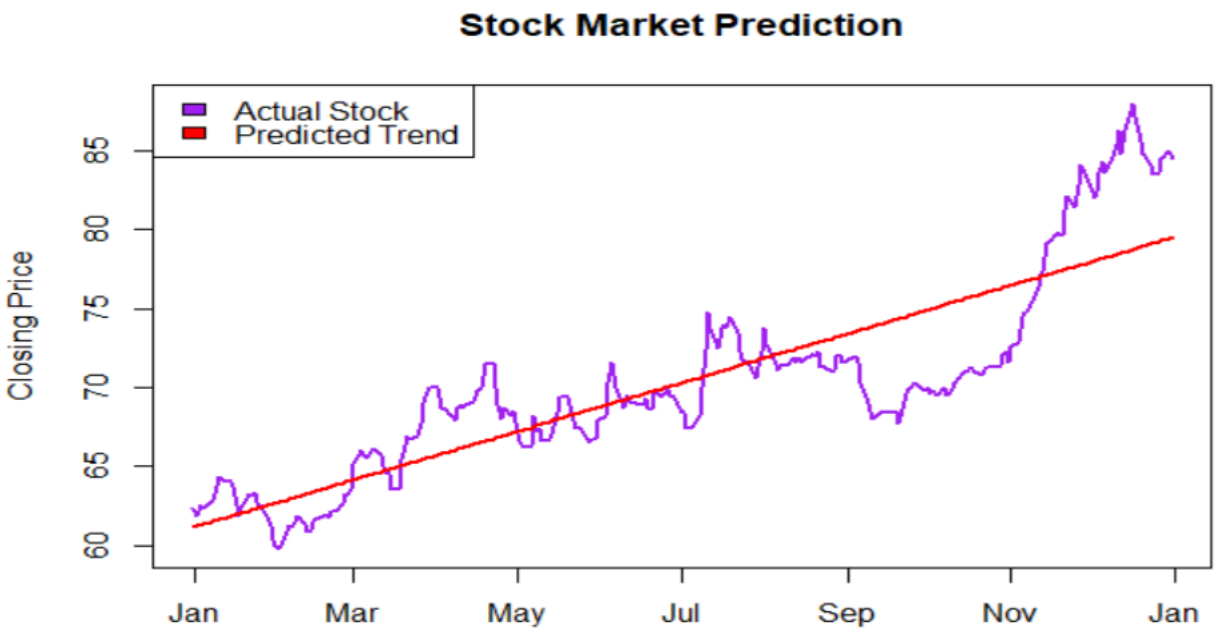
$Y = a + bX$

where slope of trend line is calculated as:

$$b_1 = \frac{\sum (x - \bar{x}) * (y - \bar{y})}{\sum (x - \bar{x})^2}$$

and the intercept is computed as:

$$b_0 = y - (b_1 * X)$$



K- nearest neighbour algorithm

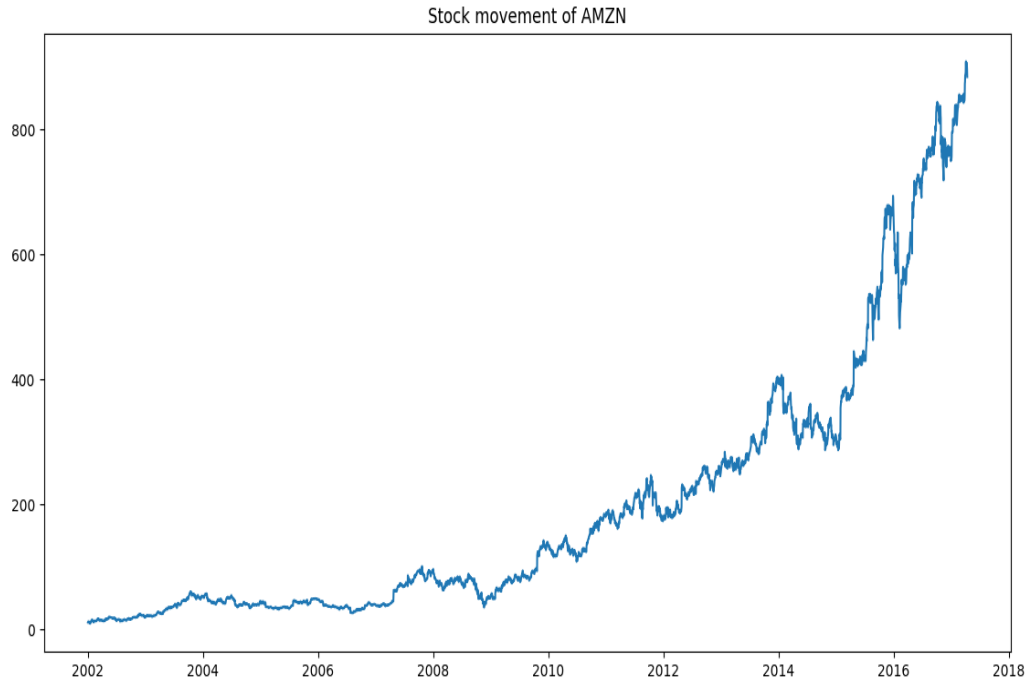
The model for kNN is the entire training dataset. When a prediction is required for an unseen data instance, the kNN algorithm will search through the training dataset for the k-most similar instances. The prediction attribute of the most similar instances is summarised and returned as the prediction for the unseen instance.

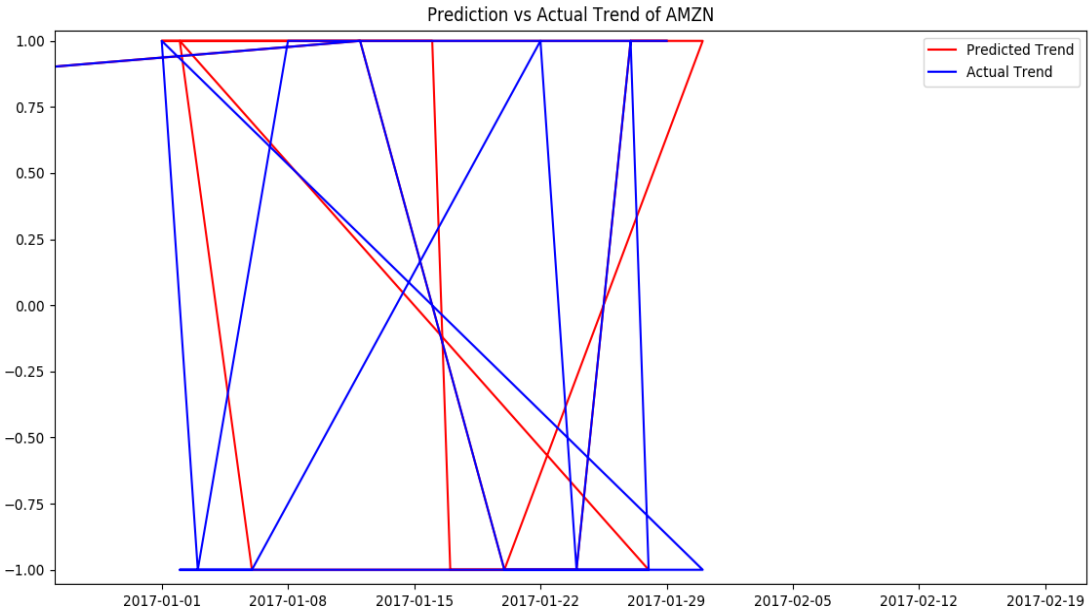
The similarity measure is dependent on the type of data. For real-valued data, **the Euclidean distance** can be used.

$$E(x, y) = \sqrt{\sum_{i=0}^{n} (x_i - y_i)^2}$$

How can we Implement kNN??

1. Handle data: Open the dataset from CSV and split into test/train datasets
2. Similarity: Calculate the distance between two data instances
3. Neighbors: Locate k most similar data instances
4. Response: Generate a response from a set of data instances
5. Accuracy: Summarize the accuracy of predictions
6. Main: Tie it all together


Stock movement of AMZN
800
600
400
200
0
2002
2004
2006
2008
2010
2012
2014
2016
2018


Prediction vs Actual Trend of AMZN
Predicted Trend
Actual Trend
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
2017-01-01
2017-01-08
2017-01-15
2017-01-22
2017-01-29
2017-02-05
2017-02-12
2017-02-19

RANDOM FOREST

Random Forests, which is a type of **ensemble learning method.** Random Forests are constructed by **multiple Decision Trees** during the training process and output the mode or mean of individual Decision Trees.

Random Forests algorithm to predict the returns for different stocks and pick corresponding stable stocks with high returns to form the portfolio.

Decision Trees are a useful machine learning algorithm.

HOW CAN WE IMPLEMENT ??

1. Begins with the dataset which should have multiple features.

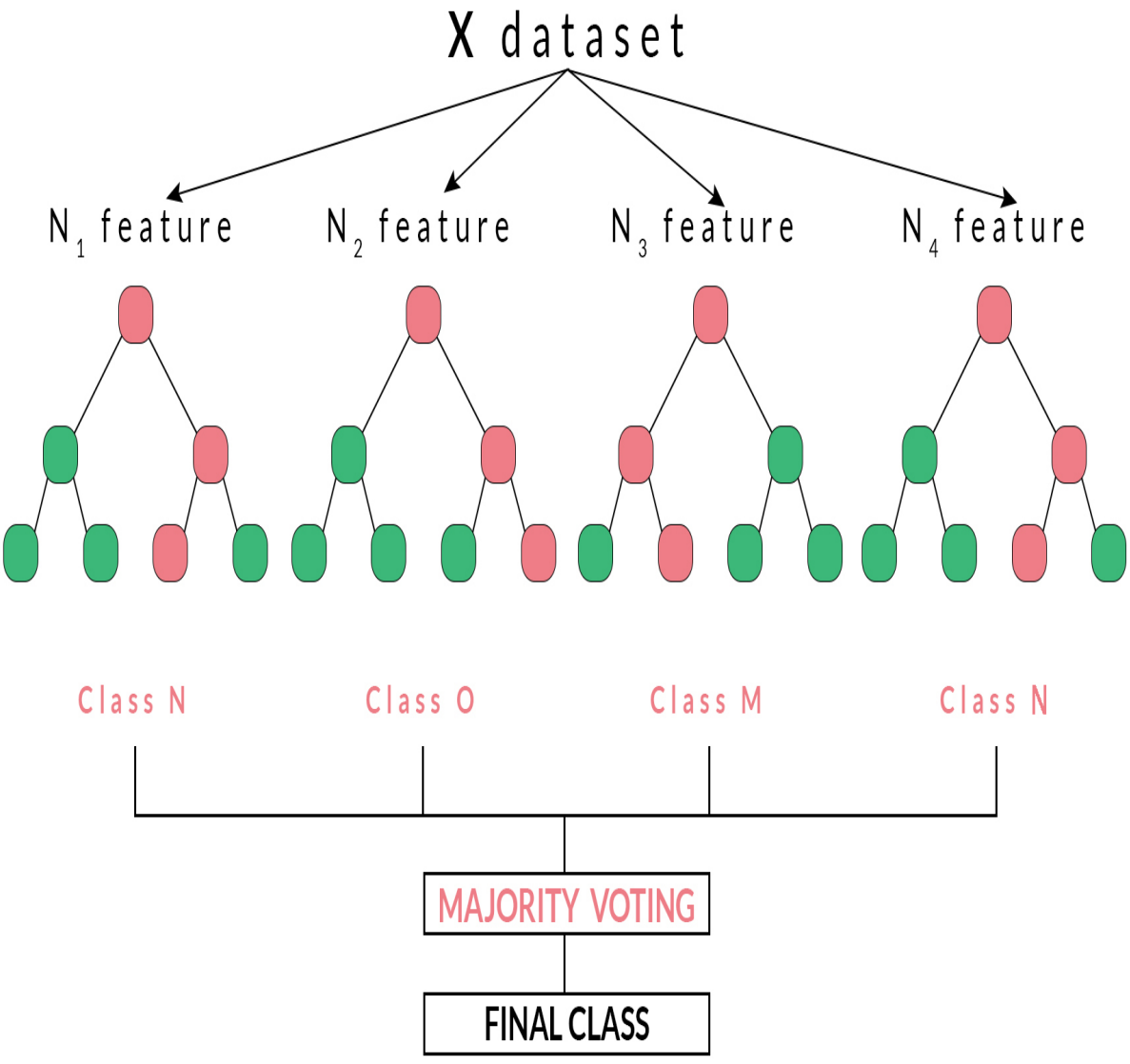
2. Find the best features in the dataset.

3. split the data into sub-data, which contains the possible values with the best features.

4.Recursively split the sub-data from step 3 by finding possible suitable features .

Decision Trees can be used for various machine learning applications, but trees that are grown really deep to learn highly irregular patterns, tend to overfit the training sets. A slight noise in the data may cause the tree to grow in a completely different manner . **Random Forests perform well on removing the over-fit problem caused by Decision Trees.**

GENERATING OF RANDOM FOREST



ARIMA

(AutoRegressive Integrated Moving Average)

AutoRegressive = the model takes advantage of the connection between a predefined number of lagged observations and the current one.

Integrated = differencing between raw observations (eg. subtracting observations at different time steps).

Moving Average = the model takes advantage of the relationship between the residual error and the observations.

The ARIMA model makes use of three main parameters (p,d,q). These are:

p = the order of the AR term

d = the degree of differencing.

q = q is the order of the MA term

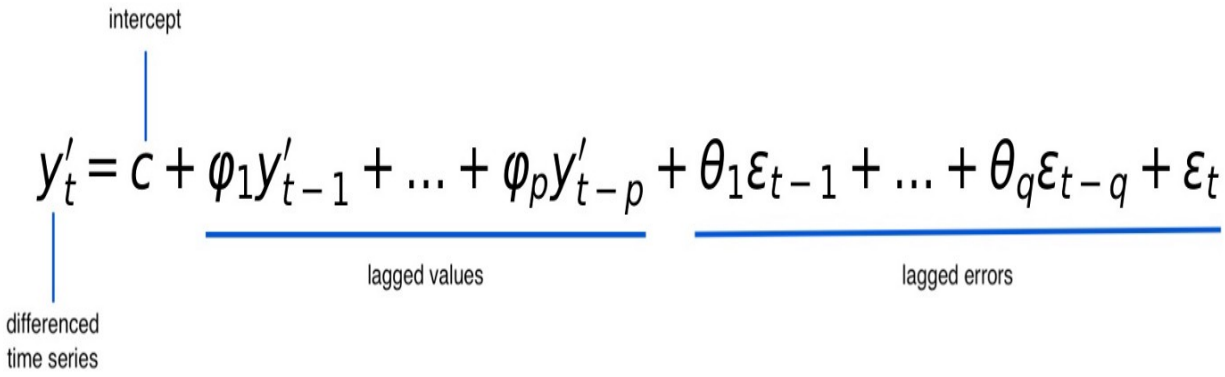
ARIMA can lead to particularly good results if applied to short time predictions

Now what is AR and MA ??

AR : forecast a series based solely on the past values in the series - called lags

MA : forecast a series based solely on the past errors in the series - called error lags

THE ARIMA MODEL EQUATION



Where:

- Y_t = the variable that will be explained in time t;
- c = constant or intercept;
- Ø = coefficient of each parameter p;
- θ = coefficient of each parameter q;
- e_t = Residuals or errors in time t.

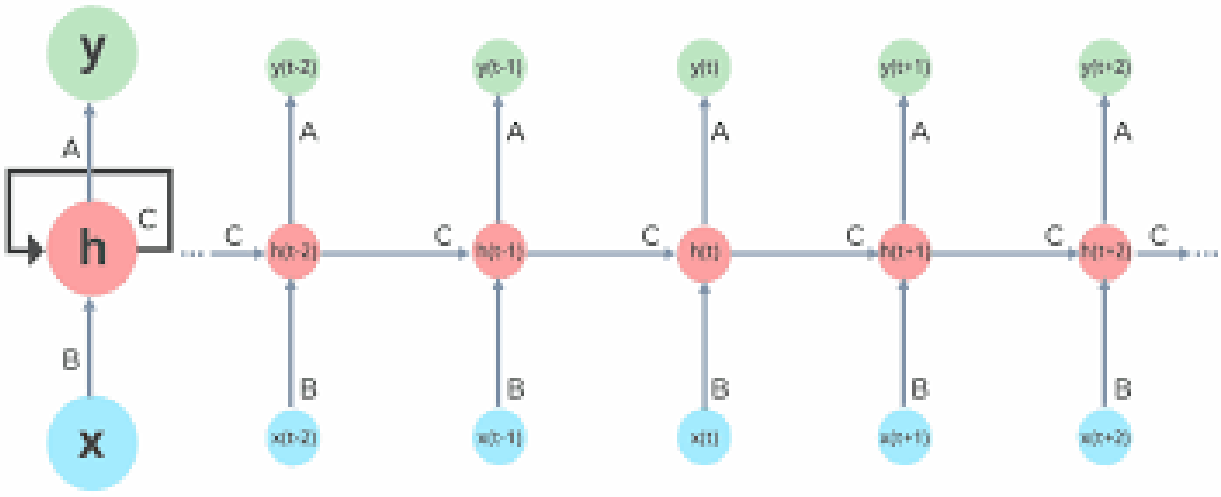
LSTM (LONG SHORT -TERM MODEL)

Long Short-term Memory (LSTM). **LSTM is a type of Recurrent Neural Networks (RNN) .**

What is RNN ?

Recurrent neural networks (RNNs) are able to process input sequences of arbitrary length via the recursive application of a transition function on a hidden state vector.

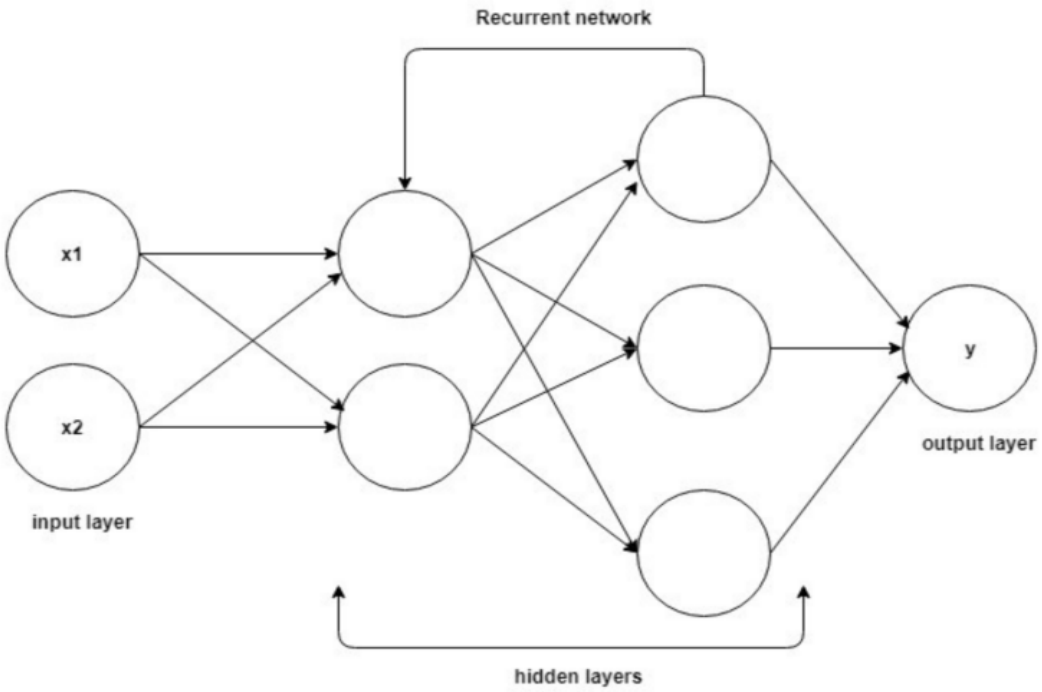
At **each time stamp t**, **the hidden state h** is resulted from a function of the **input vector x** at time t and its **previous hidden state h −1**.



For example, the input vector x could be a vector representation of the t-th word in the body of text .

As a result, recurrent networks can recirculate previous outputs back to the inputs, similar to the concept of using lagged variables in forecasting . Because RNN can learn the result from the previous round, this gives RNN a time series context.

RNN MODEL



Problem with rnn ??

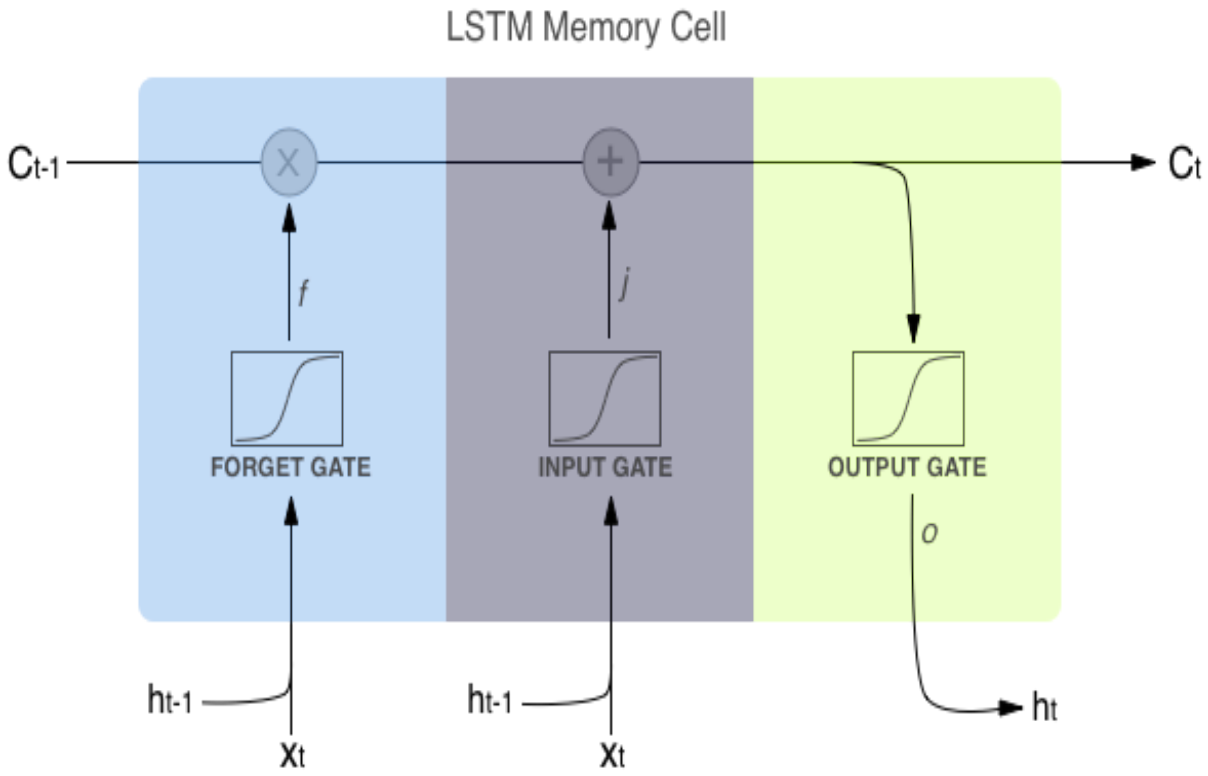
RNN can only read the results from the previous round. For stock price data, there are thousands of points of data. Therefore, only remembering the previous round result is not enough, because the stock price might also be influenced by the price weeks ago or months ago.

Solution ...

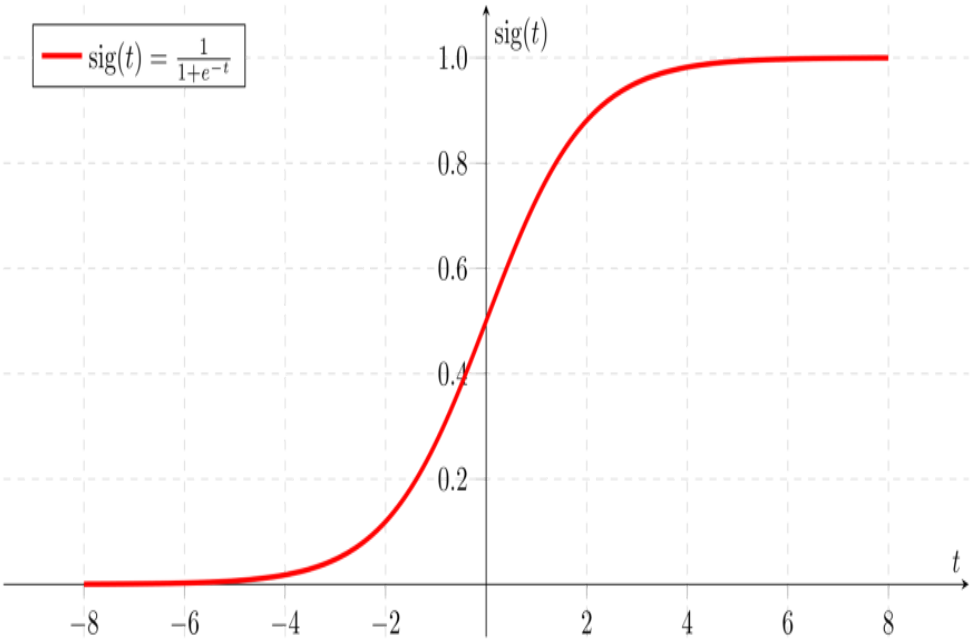
LSTM because LSTM can address this problem. The LSTM architecture fixes the lack of ability to learn long-term dependencies **by introducing a memory cell that is able to preserve states over long periods of time**.

In LSTM we will have 3 gates

1) Input Gate. **2) Forget Gate .** **3) Output Gate.**



Gates in LSTM are the sigmoid activation functions i.e they output a value between 0 or 1 and in most of the cases it is either 0 or 1.



Why do we use sigmoid function for gates??

Because we want a gate to give only positive values and should be able to give us a clear cut answer whether we need to keep a particular feature or we need to discard that feature.

“0” means the gates are blocking everything.

“1” means gates are allowing everything to pass through it.

The equations for the gates in LSTM are:

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$$

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f)$$

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o)$$

$i_t \rightarrow$ *represents input gate.*

$f_t \rightarrow$ *represents forget gate.*

$o_t \rightarrow$ *represents output gate.*

$\sigma \rightarrow$ *represents sigmoid function.*

$w_x \rightarrow$ *weight for the respective gate*(x) *neurons.*

$h_{t-1} \rightarrow$ *output of the previous lstm block*$(at\ timestamp\ t-1)$.

$x_t \rightarrow$ *input at current timestamp.*

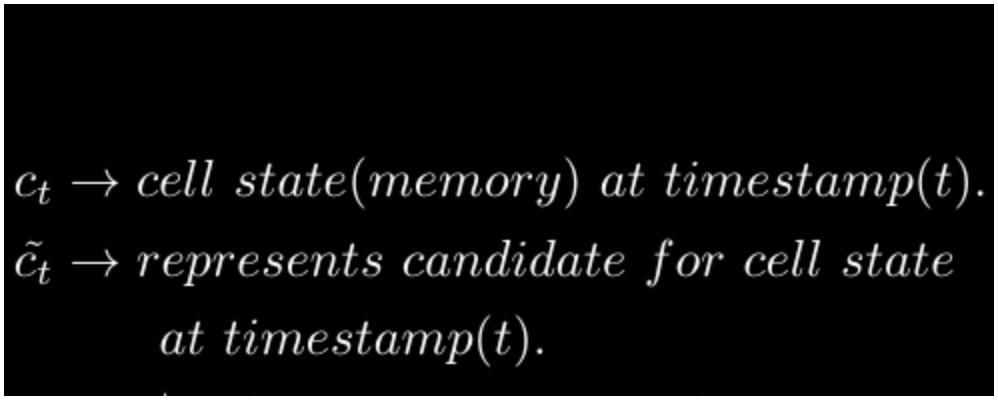
$b_x \rightarrow$ *biases for the respective gates*(x).

1

1. First equation is for the Input Gate which tells us what new information we're going to store in the cell state.
2. Second is for the forget gate which tells the information to throw away from the cell state.
3. Third one is for the output gate which is used to provide the activation to the final output of the lstm block at timestamp 't'.

The equations for the cell state, candidate cell state and the final output:

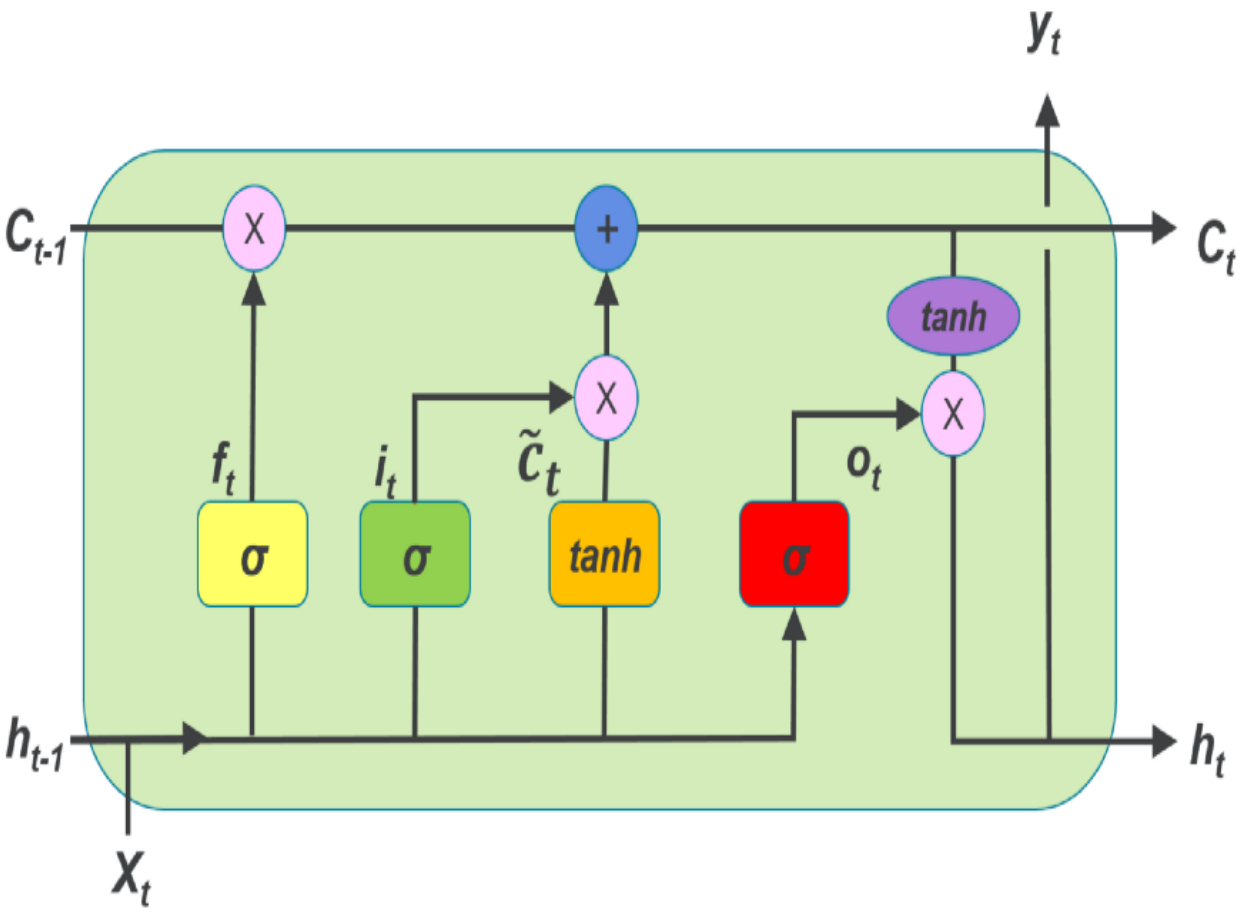
$$\tilde{c_t} = tanh(w_c[h_{t-1}, x_t] + b_c)$$
$$c_t = f_t * c_{t-1} + i_t * \tilde{c_t}$$
$$h_t = o_t * tanh(c^t)$$



To get the memory vector for the current timestamp (c_{t}) the candidate is calculated.Now, from the above equation we can see that at any timestamp, our cell state knows that what it needs to forget from the previous state(i.e f * c_{t-1}) and what it needs to consider from the current timestamp (i.e i_{t} * c`_{t}).

Lastly, we filter the cell state and then it is passed through the activation function which predicts what portion should appear as the output of the current lstm unit at timestamp t.We can pass this h_{t} the output from the current lstm block through the softmax layer to get the predicted output(y_{t}) from the current block.

Let's look at a block of lstm at any timestamp {t}.



Conclusion

We are going to use the lstm model in our project because it gives the maximum efficiency when the data set is large .