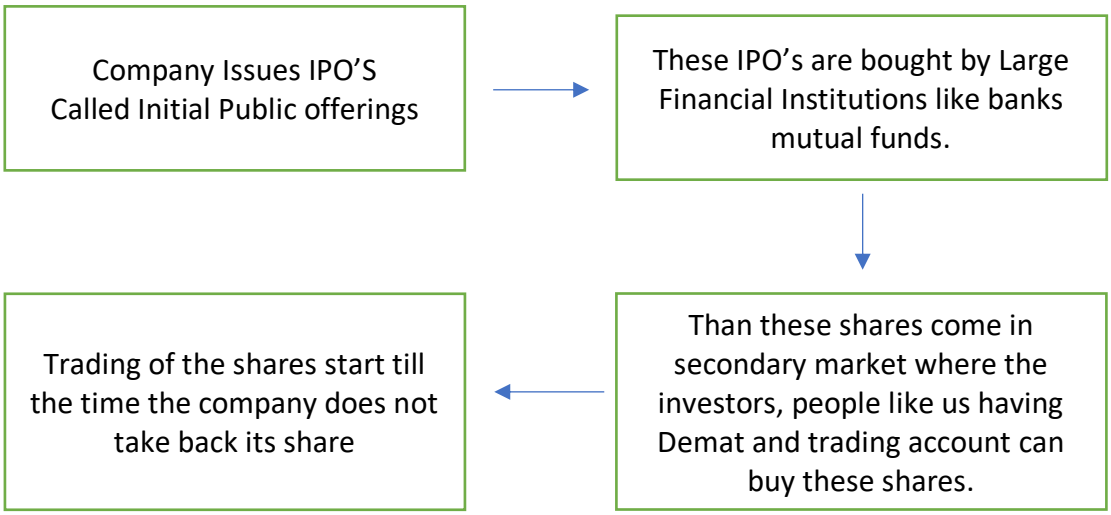
SYNOPSIS OF
"STOCK MARKET PREDICTION"
USING
ARTIFICIAL NEURAL NETWORK

Submitted By:
Khushboo Soni :- 11615575
Prateek Jaiswal :- 11614309
Shipra Aggarwal :-11604838
Vishesh Nautiyal :-11615627

INTRODUCTION

Stock Market:

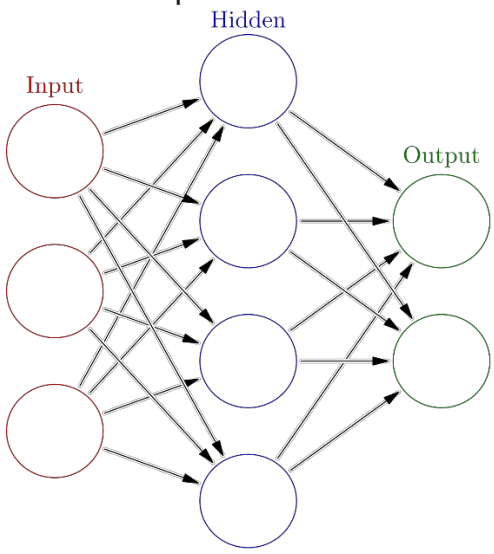
A stock market, equity market or share market is the collections of buyers and sellers it is not a physical market rather it is a platform for trading of stocks also called shares. These may include securities listed on a public Stock exchanges as well as some stocks which are traded privately.



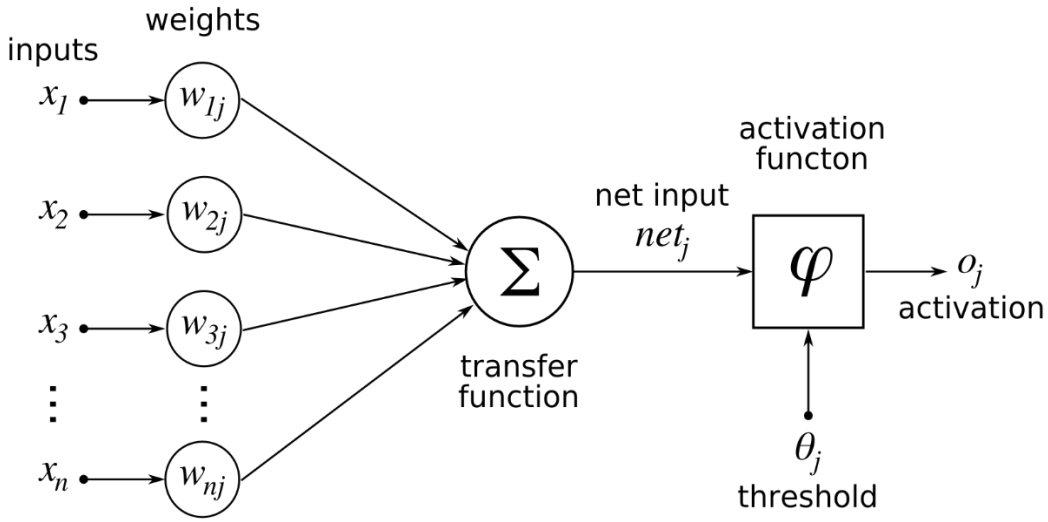
In India we are having 2 main Stock Exchanges where trading of these stock takes place National Stock Exchange (NSE) and Bombay Stock Exchange (BSE), and few regional stock exchanges. Through these stock exchange we can trade for securities. Generally, people don't know much about the trends in Stock Market how there price are changing so they Invest their money through **Brokers**.
These Brokers study the trend in market using the predictive Analysis and invest, trade in the market accordingly. But they charge high commission rate for the analysis they do. There comes the need for a model, a platform where people can run a code to predict the trend of the current share value and the probable ups and down according to which they can trade in market and avoid paying high commissions to the brokers or other intermediaries. For trading the investors can go for online trading platform which provides the platform to trade and Analysis can be done based on a model.
One Such possible network for Stock Market Prediction is **Artificial Neural Networks.**

***A*rtificial Neural Network:**

An artificial neural network is an interconnected group of nodes, like the vast network of neurons in a brain. Here, each circular node represents an artificial neuron and an arrow represents a connection from the output of one artificial neuron to the input of another. The each neuron has to perform the computations and then pass the output value to another node level.



In Artificial Neural Network we have different layers: 1 input layer, 1 output layer and Several hidden layer.



We must train our neural network, For the training purpose we update the weights and for that we go for **back propagation** for several iteration till we get least error, or we can say our predicted output values are like target values. For the practical implementation we divide our dataset into 2 parts in ration 9:1. The 9 parts are used to train our network to update the weights to get possible values and 1 part to check our result to see whether the training was successful or not.

Artificial Neural Network for Stock Market Prediction:

Now, we can use these artificial neural networks to predict the values of the stock Market.

- We use the past trends int stock Market as our input values i.e. the values of share of a company for a period.
- The Data includes various factors because of which the stock prices fluctuates or there comes the variation in price of Stock.
- Using the 90% of our data we train our networks update the weight for different layers using back propagation.
- After successful updating of the weights we use 10% of the data set to test our training whether it was successful or not.

In this way we use Artificial Neural Network for the Stock price , share market prediction.

TRADITIONAL STOCK MARKET PREDICTION

Many traditional methods have been applied to predict either stock market moving price or stock market closing price. There are two important theories used in conventional approach for stock market prediction namely efficient market hypothesis (EMH) and the random walk theory.

The efficient market hypothesis is introduced by Fama in 1964. According to EMH hypotheses, the future stock price is unpredictable based on the stock historical data. As new information enters the system the unbalanced stock is immediately discovered and quickly eliminated by the correct change in the price. The EMH exists in three forms namely weak EMH, semi-strong EMH, and strong EMH. In weak EMH, historical data are used to predict the stock price. In semi-strong EMH, besides historical data all the current public information are used to predict the stock price. In the strong EMH, all the data including historical, public and private information such as insider's information are used to predict the stock price. On the other hand, the random walk hypothesis states that stock prices do not depend

on past stocks. Thus, these are not patterns to be exploited since the historical data do not reflect the pattern of the current stock price.

Two conventional approaches used for stock market prediction are technical analysis and fundamental analysis. Technical analysis is a numerical time series approach to predict stock markets based on historical data using charts as the primary tool. This approach tries to mine information from the historical data to recognize the pattern, sometimes referred as mining the time series. Fundamental analysis is the study on the factors that affect supply and demand. The fundamental analysis states that information gathering and interpretation is the main process to predict the stock price. The trading opportunity of this analysis utilizes the gap between the occurrence of an event and the market response to the event. The important data that is used for fundamental analysis are economic data of companies (such as annual and quarterly reports), auditor's reports, balance sheets, and income statements. News also plays a role in fundamental analysis as news also reflects the current supply and demand chain in the market. These conventional approaches are now becoming inferior due to the increase in the computational power where computer can now analyse larger data set more accurately within a shorter time. However, these approaches still act as the base of new artificial intelligence approaches such as machine learning, computational intelligence and others. In this paper we will discuss some of the ANN approaches. In the next section, the concept of ANN is briefly described.

THE DATASET

The dataset that we would work on is of stock price of Apple. It is the standard dataset obtained from Yahoo Finance.

Date	Open	High	Low	Close	Adj Close	Volume
2018-01-03	172.529999	174.550003	171.960007	172.229996	170.274567	29517900
2018-01-04	172.539993	173.470001	172.080002	173.029999	171.065506	22434600
2018-01-05	173.440002	175.369995	173.050003	175	173.013123	23660000
2018-01-08	174.350006	175.610001	173.929993	174.350006	172.370514	20567800
2018-01-09	174.550003	175.059998	173.410004	174.330002	172.350739	21584000
2018-01-10	173.160004	174.300003	173	174.289993	172.311188	23959900
2018-01-11	174.589996	175.490005	174.490005	175.279999	173.289948	18667700
2018-01-12	176.179993	177.360001	175.649994	177.089996	175.079391	25226000
2018-01-16	177.899994	179.389999	176.139999	176.190002	174.189636	29565900
2018-01-17	176.149994	179.25	175.070007	179.100006	177.066589	34386800
2018-01-18	179.369995	180.100006	178.25	179.259995	177.224762	31193400
2018-01-19	178.610001	179.580002	177.410004	178.460007	176.433853	32425100

There are 7 columns in the data obtained: -

1) Date – The date of which stock value trend is obtained.
2) Open – The value of the stock at the time when the market opened.
3) High – The highest the stock value touched from when the market opened for trade and till the time it closed.
4) Low – The lowest the stock value touched from when the market opened for trade and till the time it closed.
5) Close – It is the value of stock with which the market closed.
6) Adj. Close – It is the adjusted value of stock which the market closed with
7) Volume – This is the quantity of stock that has been traded in the whole day span.

*The stock price obtained is in US dollars.

Software Requirement Analysis

- **Python** : Python is an interpreted high-level programming language for general-purpose programming. It is created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notaly using significant whitespace. It provides constructs that enable clear programming on both small and large scale.
- **Jupyter Notebook** : The Jupyter Notebook is an open-source web application that allows you to create and share documents that contains live code, equations, visualizations and narrative text.

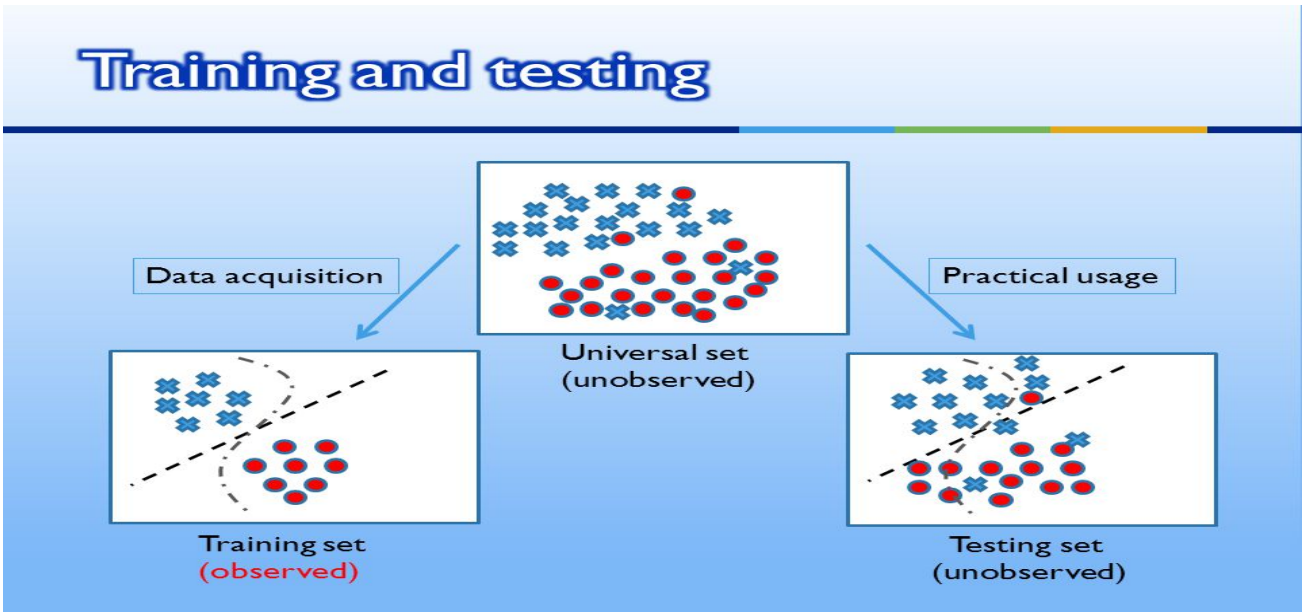
  Uses includes : data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.
- **Comma-separated values (CSV)** : In computing, a comma-separated values file is a delimited text file that uses a comma to separate values. A CSV file stores tabular data (numbers and text) in plain text. Each line of the file is a record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format.
- **matplotlib** : Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hard-copy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and the Jupyter notebook, Web application servers, and four graphical user interface toolkits.

- **Pandas** : Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.
- **Numpy** : NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays .It is the fundamental package for scientific computing with Python.

Training and Test Sets: Splitting Data

The idea of dividing your data set into two subsets:

- **training set**—a subset to train a model.
- **test set**—a subset to test the trained model.



Slicing a single data set into a training set and test set.

Make sure that your test set meets the following two conditions:

- Is large enough to yield statistically meaningful results.
- Is representative of the data set as a whole. In other words, don't pick a test set with different characteristics than the training set.

Assuming that your test set meets the preceding two conditions, your goal is to create a model that generalizes well to new data. Our test set serves as a proxy for new data. For example, consider the following figure. Notice that the model learned for the training data is very simple. This model doesn't do a perfect job—a few predictions are wrong. However, this model does about as well on the test

data as it does on the training data. In other words, this simple model does not overfits the training data.

Validating the trained model against test data.

Never train on test data. If you are seeing surprisingly good results on your evaluation metrics, it might be a sign that you are accidentally training on the test set. For example, high accuracy might indicate that test data has leaked into the training set.

For example, consider a model that predicts whether an email is spam, using the subject line, email body, and sender's email address as features. We apportion the data into training and test sets, with an 80-20 split. After training, the model achieves 99% precision on both the training set and the test set. We'd expect a lower precision on the test set, so we take another look at the data and discover that many of the examples in the test set are duplicates of examples in the training set (we neglected to scrub duplicate entries for the same spam email from our input database before splitting the data). We've inadvertently trained on some of our test data, and as a result, we're no longer accurately measuring how well our model generalizes to new data.