ARTIFICIAL INTELLIGENCE IN BUSINESS

(STOCK MARKET PREDICTION)

SUBMITTED BY:

MUSKAN KAPOOR, 12108224

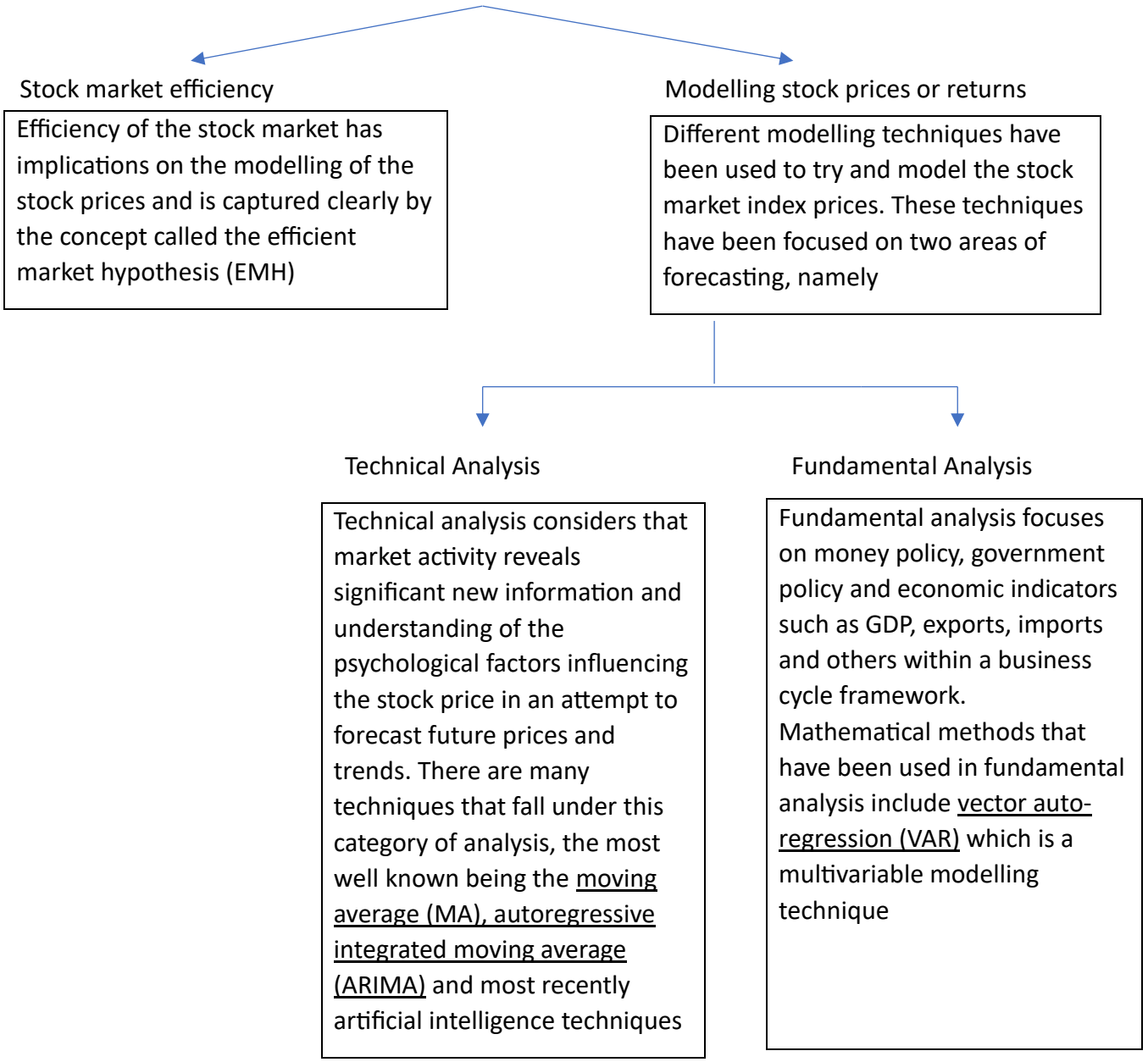
AVESH SINGH, 12107903

LOVEDEEP, 12107447

INTRODUCTION:

Predicting the Stock Market has been the goal of investors since its existence. Everyday billions of dollars are traded on the exchange, and behind each dollar is an investor hoping to profit in one way or another. Should an investor be able to accurately predict market movements, it offers tantalizing promises of wealth and influence. It is no wonder then that the Stock Market and its associated challenges find their way into the public imagination every time it misbehaves. The share market is a compilation of different people buying and selling the shares. Mostly known as stock (stake)which generally refers to claims of ownerships over a business by an individual or group of individuals. The way of finding the future valuation of the stock market prices is called the stock market estimate. Expected to be Strong, accurate and effective.

Studies of stock market changes focus on two very broad areas, namely



Efficient Market Hypothesis

EMH is related to the concept of “random walk” which asserts that future stock prices randomly depart from the past prices. The reason for the “random walk” is that new information is immediately reflected on the stock price and the future price will also reflect information which comes randomly.

There are three types of EMH:

- Weak-Form Efficiency - this form of efficiency states that the past price information is fully incorporated in the current price and does not have any predictive power. This means that predicting the future returns of an asset based on technical analysis is impossible.
- Semi-Strong Form Efficiency - this form of efficiency states that any public information is fully incorporated in the current price of an asset. Public information includes the past prices and 12 3.2. EFFICIENT MARKET HYPOTHESIS also the data reported in a company’s financial statements, earnings and dividends announcements, the financial situation of company’s competitor, expectations regarding macroeconomic factors, etc.
- Strong Form Efficiency - this form of efficiency states that the current price incorporates all information, both public and private. This means that no market actor can be able to consistently derive profits even if trading with information that is not already public knowledge.

EMH is a statement about:

- the assertion that stock prices reflect the true value of stocks;
- the absence of arbitrage opportunities in the economy dominated by rational, profit maximizing agents;
- the hypothesis that all available information comes to the market randomly and is fully reflected on the market prices

Studies evaluating EMH

Previous empirical studies of testing EMH have mainly used econometric models such as the run test, serial correlation test and the variance ratio test. The serial correlation test and the run tests are used to test the dependence of share prices. Another way of testing this kind of efficiency is to find statistical relationships between past prices and future prices. These statistical relationships allow forecasters to make predictions about future stock market returns. Various statistical methods have been used to test the EMH such as Auto Regressive Conditional Heteroskedasticity (ARCH), Generalised Auto Regressive Conditional Heteroskedasticity (GARCH) and Auto Regressive Moving Average (ARMA)

One can argue that the traditional econometric approach based on models with simple specifications and constant parameters, such as Box-Jenkins ARMA models, are unable to respond to the dynamics inherent in economic and financial series. However, the debate concerning the gains coming from the use of nonlinear models has not reached a consensus yet, stimulating further research in areas such as nonlinear model selection, estimation and evaluation approaches.

Semi-strong efficiency

Semi-strong efficiency is concerned with publicly available information. The test for this form of efficiency focuses on the evaluation of the speed and accuracy of the adjustment of the stock prices due to the availability of the information. The way it is usually done is to try and predict a future price using different asset pricing methodologies and then compare the price to the adjusted one. The difference of the two prices is then analyzed to determine the speed and the accuracy of the adjustments

Strong-form of efficiency

The strong-form of efficiency is tested by evaluating whether there are individuals who have access to private information that helps them to earn above average return in the market. If there is an investor who persistently outperforms the market then there is evidence for the lack of strong form efficiency

Modelling stock prices or return

Short term forecasting problems are concerned with prediction of variation under the assumption that the essential nature of the process will continue. It is appropriate to try to determine fixed rules to predict the near future from the recent past. The availability of extensive historical information makes it possible to derive a statistical forecaster based on this data, which can predict better than the ones chosen by judgment. Various forecasting models have been developed to predict the future based on past observations

Previous techniques

Forecasting is an attempt to predict how a future event will occur. The main objective of forecasting the occurrence of this event is for decision makers to make better decisions. This section presents the different models that have been used previously and the ones that are being used currently and the ones that are still under research.

SMOOTHING

Smoothing methods are used to determine the average value around which the data is fluctuating. Two examples of this type of approach are the moving average and exponential smoothing.

Moving averages are constructed by summing up a series of data and dividing by the total number of observations. The total number of observations is determined arbitrarily to compromise between stability and the responsiveness of the forecaster.

Exponential smoothing is constructed in such a way that the forecast value is a weighted average of the preceding observations, where the weights decrease with the age of the past observations.

CURVE LIFTING

A graph of the history of different time series processes sometimes exhibits characteristic patterns which repeat themselves over time. The tendency to extrapolate such is often hard to resist. A number of forecasting methods have been based on the premise that such extrapolation is a reasonable thing to do. Curve fitting, or data mining, is the 'art' of drawing conclusions based on past information. When applied to an investment scheme or trading strategy, history shows that often such conclusions do not hold true once they are implemented.

TECHNIQUES BASED ON MATHEMATICS

LINEAR MODELLING

Linear regression forecasting models have demonstrated their usefulness in predicting returns in both developed markets and developing markets. Linear regression models that have been tested can correctly predict direction in the market over 55-65 percent of the time. It was established that random walk hypothesis, which assumes that the best prediction for the future price is the current price, could predict the direction of market prices 50 percent of the time. It may be reasonable to state at this point that nonlinearities in the behaviour of the stock market prices could be the cause of this inability of linear methods to exhibit significant superiority over random walk hypothesis. For this reason, linear regression methods have been unable to give satisfactory results for investors.

Autoregressive-integrated moving average (ARIMA) which is a univariable model, is one the linear techniques that has been extensively used to try to predict the direction of market price.

Linear models are simple and as a result tend to want to simplify a system as complicated as financial market behaviour. However, the advantage of linear models lies in the simplicity. A model is only useful as long as its predictions do not deviate too far from the outcome of the underlying process. With linear models there is a trade off between simplicity and accuracy of the predictions.

NON-LINEAR MODELLING

The discovery of non-linear movements in the financial markets has been greatly emphasised by various researchers and financial analysts. Non-linear models are much more complicated than linear models and therefore much more difficult to construct. Part of the reason for this difficulty is the number of diverse models which is higher for non-linear models making it difficult for one to choose a suitable model. Research has established some methods of identifying non-linear models such as non-linear regression, parametric models such GARCH, and non-linear volatility models and nonparametric models. Artificial intelligence techniques such as neural networks and support vector machines are also under investigation to further the research of non-linear models. Even though there are a number of non-linear statistical techniques that have been used to produce better predictions of future stock returns or prices, most techniques are model-driven approaches which require that the non-linear model be specified before the estimation of parameters can be determined. In contrast, artificial

intelligence techniques are data-driven approaches which do not require a pre-specification during the modelling process because they independently learn the relationship inherent in the variables. Thus, neural networks are capable of performing non-linear modelling without a priori knowledge about the relationship between input and output variables. As a result, there has been a growing interest in applying artificial intelligence techniques to capture future stock behaviours. Among the different nonlinear methods artificial neural networks are being used by forecasters as a non-parametric regression method. The advantage with using neural networks (NN) is that as a non-linear approximator it exhibits superiority when compared to other non-linear models. The reason is because NN is able to establish relationships in areas where mathematical knowledge of the stochastic process underlying the analyzed time series is unknown and difficult to rationalise. In another case, a back propagation neural network developed by Tsibouris and Zeidenburg , that only used past share prices as input had some predictive ability, which refuted the weak form of the EMH.

In contrast, a study on IBM stock movement by, did not find evidence against the EMH. Mukherjee et al., showed the applicability of support vector machines (SVM) to time-series forecasting. Recently, the predictability of financial time-series including five time series data with SVMs is examined which showed that SVMs outperformed the Back Propagation networks on the criteria of normalised mean square error, mean absolute error, directional symmetry and weighted directional symmetry. Some applications of SVM to financial forecasting have been reported recently.

Practical applications of fuzzy systems include systems for stock selection, foreign exchange trading, etc. Chen et al used fuzzy time-series to forecast the Taiwan stock market. Fuzzy logic is also used to improve the effectiveness of neural networks by “incorporating structured knowledge about financial markets, including rules provided by traders, and explaining how the output or trading recommendations were derived” .

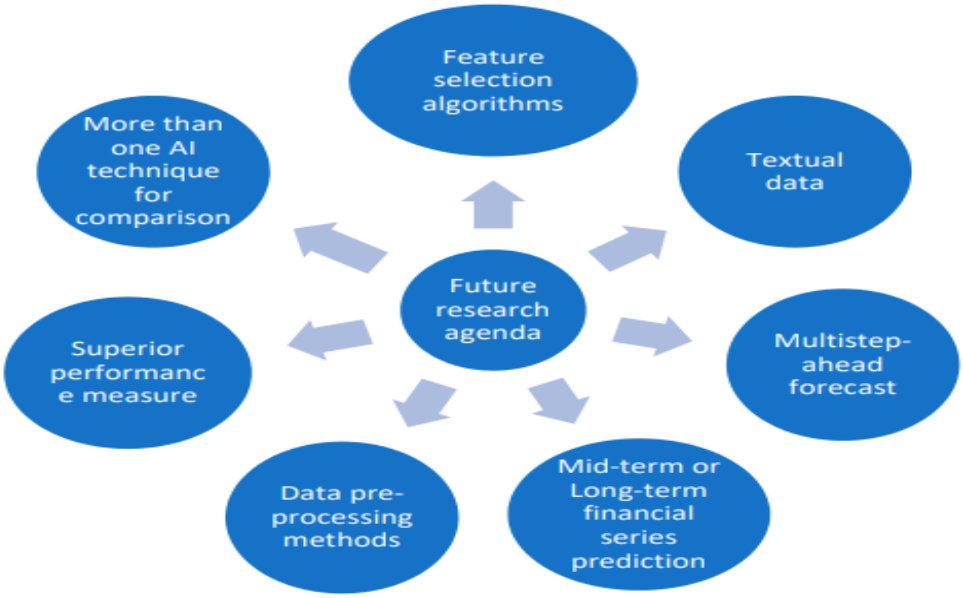
EXISTING SYSTEM

In the present system the various algorithms used for forecasting can be divided into queues (AR, MA, ARIMA, ARMA) and incompatible models (ARCH, GARCH, Neural Network) and AI framework such as Naive Bayes, the closest neighbours k (k-NN), Support Vector Machine (SVM), Linear Regression, Artificial Neural Network (ANN) and Random Forest were used to advance the gauge model.

Current models predict that the stock market uses only one algorithm to predict different conditions and variables and also does not combine multiple algorithm results or consider multiple algorithms to accurately predict. The current system does perform optimally if there is a change in the operating surrounding . It uses only one data source, thus being extremely biased. The existing system requires some kind of input translation, so it needs to be measured. The existing system uses only historical data or media analysis simultaneously, not used together.

The current system does not take into account certain important data such as trade volume and transaction value in the trading volume and the percentage of the amount that are be delivered and the percentage of delivery that predicts an investment or investment that occurs in a particular stock by a major fund manager or large investors. The combination of these factors and volume data can be an important predictor parameter that is not considered much in existing systems.

Most of the existing systems uses only one algorithm and one data at a time. The existing system also does try to predict share prices in all conditions and on all days but in real world share market cannot be predictable every time so certain conditions need to be checked before predicting share market.



FUTURE RESEARCH AGENDA

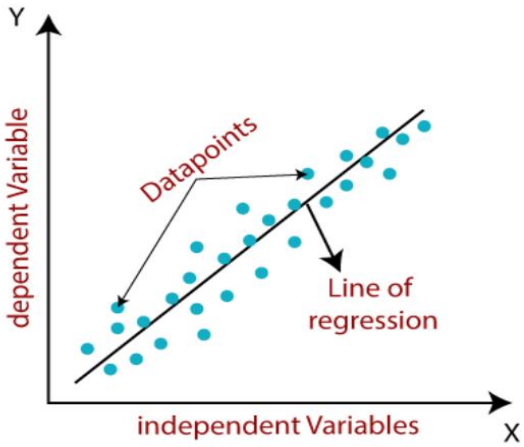
AI Techniques

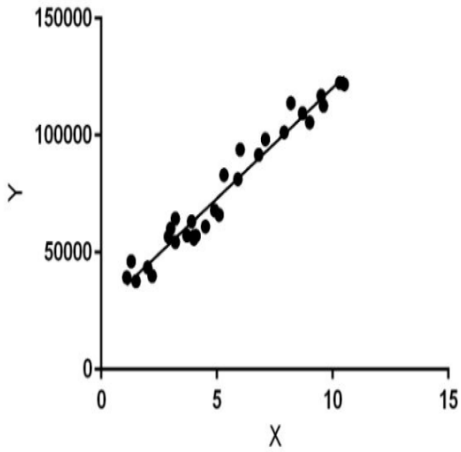
- LINEAR REGRESSION

It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

The linear regression model provides a sloped straight line representing the relationship between the variables. Consider the below image:





Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x)). Hence, the name is Linear Regression. In the second figure above, X (input) is the work experience and Y (output) is the salary of a person.

Mathematical Representation of Linear Regression ->

$y = a_0 + a_1x + \dot{\varepsilon}$, where,

y = dependent variable , x = independent variable

a_0 = intercept of line(gives additional degree of freedom)

a_1 = Linear regression coefficient (scale factor to each input value.)

$\dot{\varepsilon}$ = random error

Types of Linear Regression

Simple Linear Regression

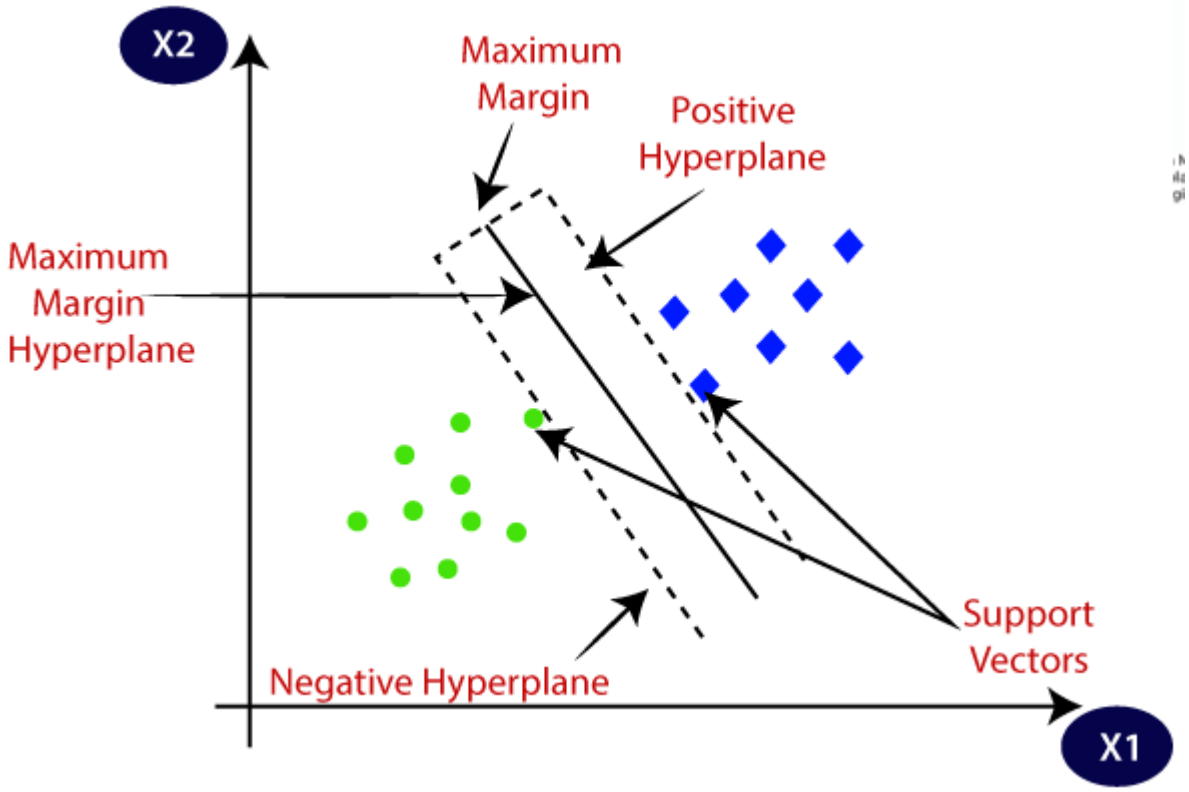
Multiple Linear Regression

- SUPPORT VECTOR MACHINE

  Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems

  The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

  SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



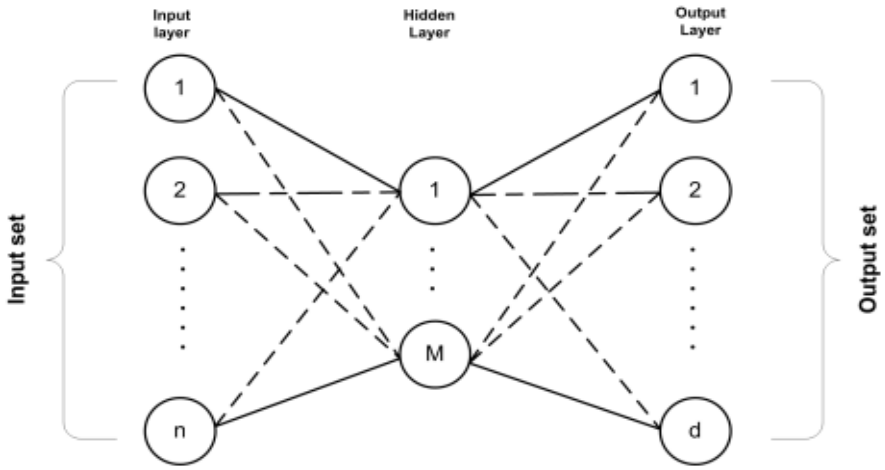
The advantages of support vector machines are:

- Effective in high dimensional spaces.
- Still effective in cases where number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different Kernal functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

- NEURAL NETWORKS

The theory of neural network (NN) computation provides interesting techniques that mimic the human brain and nervous system. A neural network is characterized by the pattern of connections among the various network layers, the numbers of neurons in each layer, the learning algorithm, and the neuron activation functions. In general, a neural network is a set of connected input and output units where each connection has a weight associated with it. Neural networks can be used for classification or regression. For this study neural networks were used as a regression tool for predicting the future price of a stock market index.

A neural network evaluates price data and unearths opportunities for making trade decisions based on the data analysis. The networks can distinguish subtle nonlinear interdependencies and patterns other methods of technical analysis cannot. According to research, the accuracy of neural networks in making price predictions for stocks differs. Some models predict the correct stock prices 50 to 60% of the time, while others are accurate in 70% of all instances. Some have posited that a 10% improvement in efficiency is all an investor can ask for from a neural network.



- FUZZY LOGIC AND FUZZY RULE

  Fuzzy logic typically processes non-linear datasets by mapping input data (feature) vectors into scalar output: i.e. it maps numbers into numbers. FL handles non-linearity well because of the fuzzy rules it uses to map the non-linear relationship between inputs and outputs. Fuzzy rule generation is usually based on past knowledge and experience. Compared to the ANN, FL offers a clear insight into the model. FL is especially popular for dealing with non-linear systems and hence it is suitable for forecasting the stock market. Nevertheless, the performance of the method depends on the fuzzification of the time series data. The generation of appropriate fuzzy rules for a financial data is often challenging. For instance, a dataset with n dimensions will produce a maximum of n^n rules.

- ARIMA METHODLOGY

  The ARIMA methodology is a statistical method for analyzing and building a forecasting model which best represents a time series by modeling the correlations in the data. Owing to purely statistical approaches, ARIMA models only need the historical data of a time series to generalize the forecast and manage to increase prediction accuracy while keeping the model parsimonious.

  If we combine differencing with autoregression and a moving average model, we obtain a non-seasonal ARIMA model. ARIMA is an acronym for AutoRegressive Integrated Moving Average. ARIMA models are also capable of modelling a wide range of seasonal data.

DISADVANTAGES OF AI IN STOCK MARKET PREDICTION:

While AI can be a powerful tool for stock market prediction, there are also several potential disadvantages to consider. Here are some of them:

- Overfitting: AI models can sometimes be too complex and end up fitting the data too closely, which can lead to overfitting. This means the model may perform well on the training data but poorly on new, unseen data, leading to inaccurate predictions.

- Data quality: The accuracy of AI models depends on the quality and quantity of the data used for training. If the data is incomplete, inaccurate, or biased, the model may produce inaccurate predictions.

- Market volatility: The stock market can be highly unpredictable, and sudden changes in market conditions can cause AI models to produce inaccurate predictions.

- Limited interpretation: AI models may produce accurate predictions, but they may not be able to provide insight into why a particular stock price is rising or falling. This can limit the ability of investors to make informed decisions based on the predictions.

- Cybersecurity risks: The use of AI in the stock market also introduces new cybersecurity risks, as hackers may try to manipulate the data used to train the models or interfere with the algorithms used for trading.

Overall, while AI has the potential to improve stock market prediction, it is important to be aware of these potential disadvantages and to use AI in conjunction with other sources of information and human expertise to make informed decisions.

Applying Linear Regression Modelling on Tesla Stock for stock prediction

- Firstly we need to import some libraries
  - For calculation :-
    1. Pandas
    2. Numpy
    3. Matplotlib
  - For Plotting the graph :-
    1. Plotly

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

import chart_studio.plotly as py
import plotly.graph_objs as go
from plotly.offline import plot

#for offline plotting
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)
```

- Importing tesla stocks on and showing first 5 rows of the dataset.

```
tesla =
pd.read_csv('C:\Users\avesh\Downloads\datasetsandcodefilesstockmarketprediction\Tesla stock price prediction.ipynb')
tesla.head()
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2010-06-29	19.000000	25.00	17.540001	23.889999	23.889999	18766300
1	2010-06-30	25.790001	30.42	23.299999	23.830000	23.830000	17187100
2	2010-07-01	25.000000	25.92	20.270000	21.959999	21.959999	8218800
3	2010-07-02	23.000000	23.10	18.709999	19.200001	19.200001	5139800
4	2010-07-06	20.000000	20.00	15.830000	16.110001	16.110001	6866900

- Displaying number of rows and columns in our data set.

```
tesla.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2193 entries, 0 to 2192
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Date       2193 non-null   object
 1   Open       2193 non-null   float64
 2   High       2193 non-null   float64
 3   Low        2193 non-null   float64
 4   Close      2193 non-null   float64
 5   Adj Close  2193 non-null   float64
 6   Volume     2193 non-null   int64
dtypes: float64(5), int64(1), object(1)
memory usage: 120.1+ KB
```

- Describing the time frame of dataset.

```
tesla['Date'] = pd.to_datetime(tesla['Date'])


print(f'Dataframe contains stock prices between {tesla.Date.min()} {tesla.Date.max()}')
print(f'Total days = {(tesla.Date.max()  - tesla.Date.min()).days} days')
```

```
Dataframe contains stock prices between 2010-06-29 00:00:00 2019-03-15 00:00:00
Total days = 3181 days
```

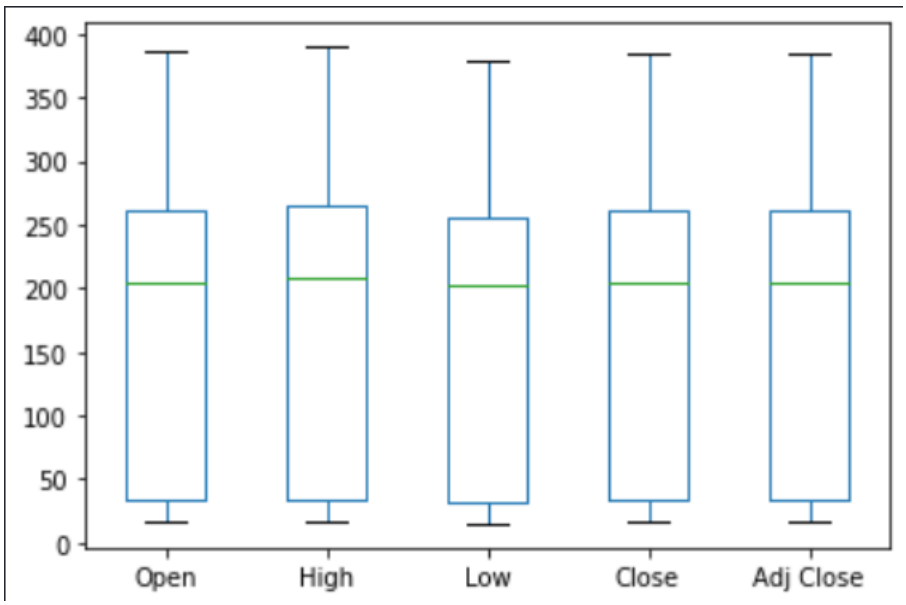
- Using the .describe() method of plotlib library for describing all the attributes of the dataset like changes in standard deviation in the open and close object cells etc.

```
tesla.describe()
```

	Open	High	Low	Close	Adj Close	Volume
count	2193.000000	2193.000000	2193.000000	2193.000000	2193.000000	2.193000e+03
mean	175.652882	178.710262	172.412075	175.648555	175.648555	5.077449e+06
std	115.580903	117.370092	113.654794	115.580771	115.580771	4.545398e+06
min	16.139999	16.629999	14.980000	15.800000	15.800000	1.185000e+05
25%	33.110001	33.910000	32.459999	33.160000	33.160000	1.577800e+06
50%	204.990005	208.160004	201.669998	204.990005	204.990005	4.171700e+06
75%	262.000000	265.329987	256.209991	261.739990	261.739990	6.885600e+06
max	386.690002	389.609985	379.350006	385.000000	385.000000	3.716390e+07

- Creating box plot for open, high, low, close, adj close.

```
tesla[['Open','High','Low','Close','Adj Close']].plot(kind='box')
```



- Setting the layout for our graph.

```
layout = go.Layout(
    title='Stock Prices of Tesla',
    xaxis=dict(
        title='Date',
        titlefont=dict(
            family='Courier New, monospace',
            size=18,
            color='#7f7f7f'
        )
    ),
    yaxis=dict(
        title='Price',
        titlefont=dict(
            family='Courier New, monospace',
            size=18,
            color='#7f7f7f'
        )
    )
)

tesla_data = [{'x':tesla['Date'], 'y':tesla['Close']}]
plot = go.Figure(data=tesla_data, layout=layout)
```

- Importing libraries for Building regression model, pre-processing of dataset and model evaluation.

```
# Building the regression model
from sklearn.model_selection import train_test_split

#For preprocessing
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler

#For model evaluation
from sklearn.metrics import mean_squared_error as mse
from sklearn.metrics import r2_score
```

- Splitting the dataset into test and train sets.

```
#Split the data into train and test sets
X = np.array(tesla.index).reshape(-1,1)
Y = tesla['Close']
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=101)

# Feature scaling
scaler = StandardScaler().fit(X_train)
```

- Importing libraries for Linear Regression modelling.

```
from sklearn.linear_model import LinearRegression
```

- Creating linear regression model.

```
#Creating a linear model
lm = LinearRegression()
lm.fit(X_train, Y_train)
```

- Plotting actual and predicted values for train dataset.

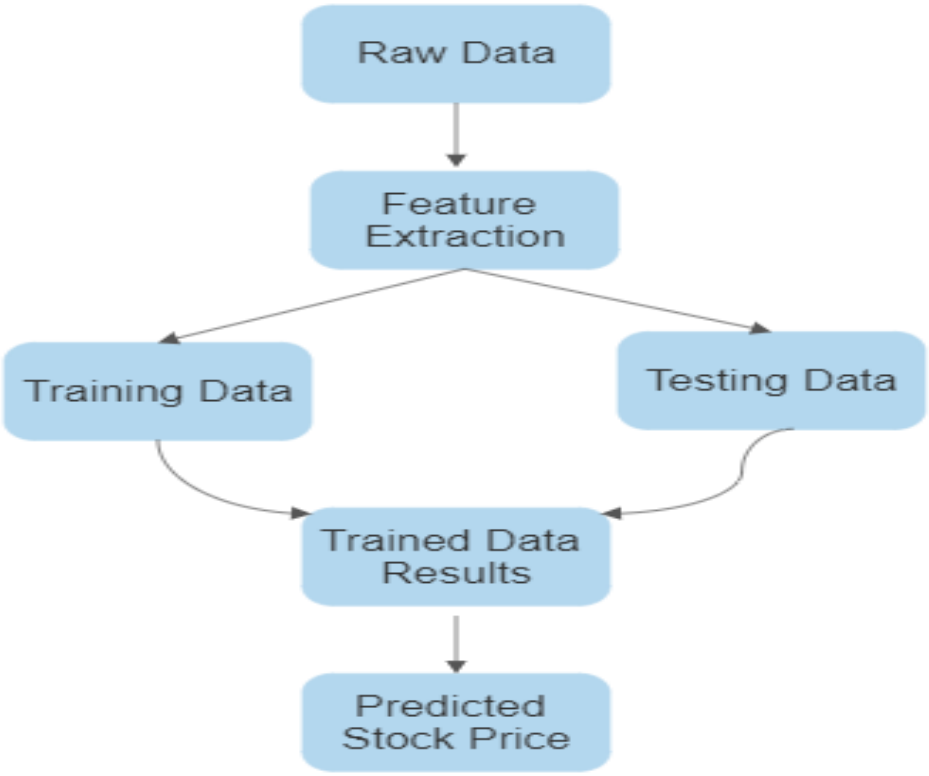
```
trace0 = go.Scatter(
    x = X_train.T[0],
    y = Y_train,
    mode = 'markers',
    name = 'Actual'
)
trace1 = go.Scatter(
    x = X_train.T[0],
    y = lm.predict(X_train).T,
    mode = 'lines',
    name = 'Predicted'
)
tesla_data = [trace0,trace1]
layout.xaxis.title.text = 'Day'
plot2 = go.Figure(data=tesla_data, layout=layout)

iplot(plot2)
```

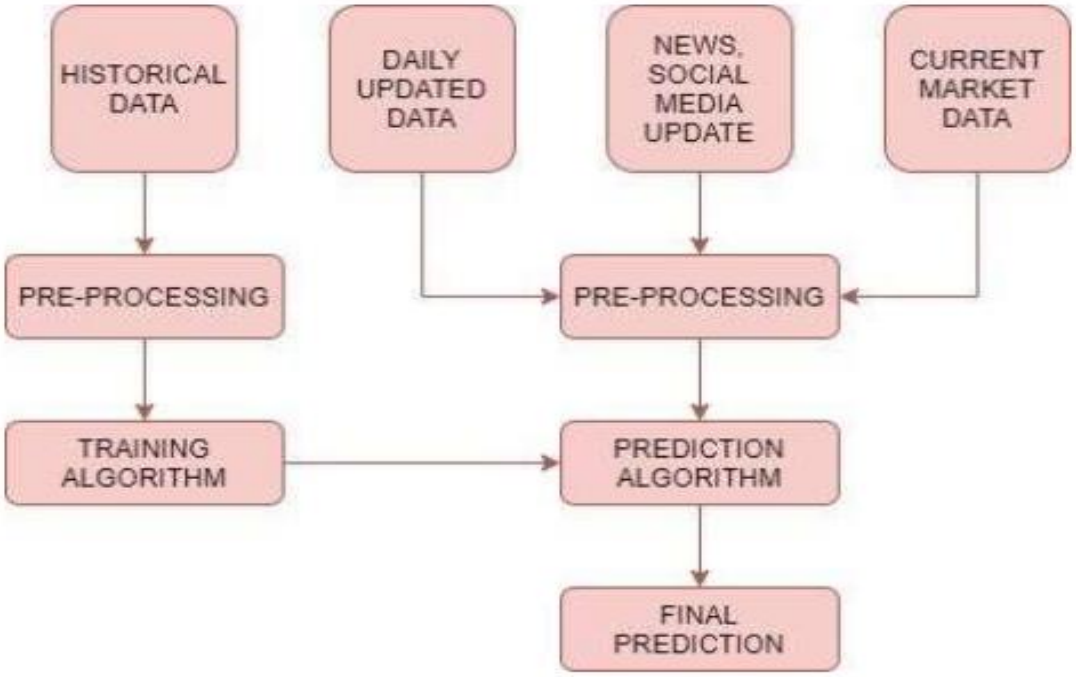
- Calculating root mean square scores for both test and train sets.

```
#Calculate scores for model evaluation
scores = f'''
{'Metric'.ljust(10)}{'Train'.center(20)}{'Test'.center(20)}
{'r2_score'.ljust(10)}{r2_score(Y_train, lm.predict(X_train))}\t{r2_score(Y_test,
lm.predict(X_test))}
{'MSE'.ljust(10)}{mse(Y_train, lm.predict(X_train))}\t{mse(Y_test, lm.predict(X_test))}
'''
print(scores)
```

```
Metric          Train                Test
r2_score  0.8658871776828707    0.8610649253244574
MSE       1821.3833862936174    1780.987539418845
```



System Architecture



References ->

- https://core.ac.uk/download/pdf/39667613.pdf
- https://www.ibm.com/in-en/topics/neural-networks
- https://www.javatpoint.com/linear-regression-in-machine-learning
- https://www.geeksforgeeks.org/ml-linear-regression/
- https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm
- https://scikit-learn.org/stable/modules/svm.html
- https://www.sciencedirect.com/science/article/abs/pii/S0925231209001805
- https://otexts.com/fpp2/index.html