****

**An Internship Project Report on**

“Stock Prediction”

Submitted

In partial fulfilment of the requirement for the

Internship in Machine Learning & Artificial Intelligence during the academic year 2022-2023

**Submitted By**

**NAME:Loknath Seth University Serial Number:10901617091**

**(MAKAUT)**

**Under the guidance of**

Mr. Shashank G

Technical Lead

QuantMasters Pvt. Ltd.

NETAJI SUBHASH ENGINEERING COLLEGE

Mauza Ranabhutia, Techno City, Garia, West Bengal 700152

**CONTENTS**

**Chapters Page no.**

**ABSTRACT 3**

**Chapter-1 Introduction**

**1.1 Stock Market 4**

**1.2 ML Techniques 5-9**

**Chapter-2 Related Work 10-12**

**Chapter-3 System Architecture 13-18**

**Chapter-4 Results 19-26**

**Chapter-5 Conclusion and Future Work 27**

**Bibilography/References 28**

**CHAPTER 3:ABSTRACT**

In the finance world stock trading is one of the most important activities. Stock market prediction is an act of trying to determine the future value of a stock other financial instrument traded on a financial exchange. This paper explains the prediction of a stock using Machine Learning. The recent trend in stock market prediction technologies is the use of machine learning which makes predictions based on the values of current stock market indices by training on their previous values. Machine learning itself employs different models to make prediction easier and authentic.The various models used are KNN,Moving Average,Auto Arima,Linear Regression and LSTM,out of which the LSTM model seemed the most promising.

**CHAPTER 1:INTRODUCTION**

**1.1 Stock Market**

**Stock market prediction** is the act of trying to determine the future value of a company [stock](https://en.wikipedia.org/wiki/Stock) or other [financial instrument](https://en.wikipedia.org/wiki/Financial_instrument) traded on an [exchange](https://en.wikipedia.org/wiki/Exchange_(organized_market)). The successful prediction of a stock's future price could yield significant profit. The [efficient-market hypothesis](https://en.wikipedia.org/wiki/Efficient-market_hypothesis) suggests that stock prices reflect all currently available information and any price changes that are not based on newly revealed information thus are inherently unpredictable. Others disagree and those with this viewpoint possess myriad methods and technologies which purportedly allow them to gain future price information.

The research work done by V Kranthi Sai Reddy Student, ECM, Sreenidhi Institute of Science and Technology, Hyderabad, India. In the finance world stock trading is one of the most important activities. Stock market prediction is an act of trying to determine the future value of a stock other financial instrument traded on a financial exchange. This paper explains the prediction of a stock using Machine Learning. The technical and fundamental or the time series analysis is used by the most of the stockbrokers while making the stock predictions. The programming language is used to predict the stock market using machine learning is Python. In this paper we propose a Machine Learning (ML) approach that will be trained from the available stocks data and gain intelligence and then uses the acquired knowledge for an accurate prediction. In this context this study uses a machine learning technique called Support Vector Machine (SVM) to predict stock prices for the large and small capitalizations and in the three different markets, employing prices with both daily and up-to-the-minute frequencies.

The research work done by Lufuno Ronald Marwala A dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science in Engineering. The weak form of Efficient Market hypothesis (EMH) states that it is impossible to forecast the future price of an asset based on the information contained in the historical prices of an asset. This means that the market behaves as a random walk and as a result makes forecasting impossible. Furthermore, financial forecasting is a difficult task due to the intrinsic complexity of the financial system. The objective of this work was to use artificial intelligence (AI) techniques to model and predict the future price of a stock market index. Three artificial intelligence techniques, namely, neural networks (NN), support vector machines and neuro-fuzzy systems are implemented in forecasting the future price of a stock market index based on its historical price information. Artificial intelligence techniques have the ability to take into consideration financial system complexities and they are used as financial time series forecasting tools.

A stock market is a platform for trading of a company’s stocks and derivatives at an agreed price. Supply and demand of shares drive the stock market. In any country stock market is one of the most emerging sectors. Nowadays, many people are indirectly or directly related to this sector. Therefore, it becomes essential to know about market trends. Thus, with the development of the stock market, people are interested in forecasting stock price. But, due to dynamic nature and liable to quick changes in stock price, prediction of the stock price becomes a challenging task. Stock m Prior work has proposed effective methods to learn event representations that can capture syntactic and semantic information over text corpus, demonstrating their effectiveness for downstream tasks such as script event prediction. On the other hand, events extracted from raw texts lacks of common-sense knowledge, such as the intents and emotions of the event participants, which are useful for distinguishing event pairs when there are only subtle differences in their surface realizations. To address this issue, this paper proposes to leverage external common-sense knowledge about the intent and sentiment of the event.

Experiments on three event-related tasks, i.e., event similarity, script event prediction and stock market prediction, show that our model obtains much better event embeddings for the tasks, achieving 78% improvements on hard similarity task, yielding more precise inferences on subsequent events under given contexts, and better accuracies in predicting the volatilities of the stock market1. Markets are mostly a nonparametric, non-linear, noisy and deterministic chaotic system (Ahangar et al. 2010). As the technology is increasing, stock traders are moving towards to use Intelligent Trading Systems rather than fundamental analysis for predicting prices of stocks, which helps them to take immediate investment decisions. One of the main aims of a trader is to predict the stock price such that he can sell it before its value decline, or buy the stock before the price rises. The efficient market hypothesis states that it is not possible to predict stock prices and that stock behaves in the random walk. It seems to be very difficult to replace the professionalism of an experienced trader for predicting the stock price. But because of the availability of a remarkable amount of data and technological advancements we can now formulate an appropriate algorithm for prediction whose results can increase the profits for traders or investment firms. Thus, the accuracy of an algorithm is directly proportional to gains made by using the algorithm.

The term ‘machine learning’ is often, incorrectly, interchanged with Artificial Intelligence[JB1] , but machine learning is actually a sub

field/type of AI. Machine learning is also often referred to as predictive analytics, or predictive modelling.

Coined by American computer scientist Arthur Samuel in 1959, the term ‘machine learning’ is defined as a “computer’s ability to learn without being explicitly programmed”.

At its most basic, machine learning uses programmed algorithms that receive and analyse input data to predict output values within an acceptable range. As new data is fed to these algorithms, they learn and optimise their operations to improve performance, developing ‘intelligence’ over time.

There are four types of machine learning algorithms: supervised, semi-supervised, unsupervised and reinforcement.

### Supervised learning

In supervised learning, the machine is taught by example. The operator provides the machine learning algorithm with a known dataset that includes desired inputs and outputs, and the algorithm must find a method to determine how to arrive at those inputs and outputs. While the operator knows the correct answers to the problem, the algorithm identifies patterns in data, learns from observations and makes predictions. The algorithm makes predictions and is corrected by the operator – and this process continues until the algorithm achieves a high level of accuracy/performance.

Under the umbrella of supervised learning fall: Classification, Regression and Forecasting.

1. **Classification**: In classification tasks, the machine learning program must draw a conclusion from observed values and determine to  
   what category new observations belong. For example, when filtering emails as ‘spam’ or ‘not spam’, the program must look at existing observational data and filter the emails accordingly.
2. **Regression**: In regression tasks, the machine learning program must estimate – and understand – the relationships among variables. Regression analysis focuses on one dependent variable and a series of other changing variables – making it particularly useful for prediction and forecasting.
3. **Forecasting**: Forecasting is the process of making predictions about the future based on the past and present data, and is commonly used to analyse trends.

### 

### Semi-supervised learning

Semi-supervised learning is similar to supervised learning, but instead uses both labelled and unlabelled data. Labelled data is essentially information that has meaningful tags so that the algorithm can understand the data, whilst unlabelled data lacks that information. By using this

combination, machine learning algorithms can learn to label unlabelled data.

### Unsupervised learning

Here, the machine learning algorithm studies data to identify patterns. There is no answer key or human operator to provide instruction. Instead, the machine determines the correlations and relationships by analysing available data. In an unsupervised learning process, the machine learning algorithm is left to interpret large data sets and address that data accordingly. The algorithm tries to organise that data in some way to describe its structure. This might mean grouping the data into clusters or arranging it in a way that looks more organised.

As it assesses more data, its ability to make decisions on that data gradually improves and becomes more refined.

Under the umbrella of unsupervised learning, fall:

1. **Clustering**: Clustering involves grouping sets of similar data (based on defined criteria). It’s useful for segmenting data into several groups and performing analysis on each data set to find patterns.
2. **Dimension reduction**: Dimension reduction reduces the number of variables being considered to find the exact information required.

### 

### Reinforcement learning

Reinforcement learning focuses on regimented learning processes, where a machine learning algorithm is provided with a set of actions, parameters and end values. By defining the rules, the machine learning algorithm then tries to explore different options and possibilities, monitoring and evaluating each result to determine which one is optimal. Reinforcement learning teaches the machine trial and error. It learns from past experiences and begins to adapt its approach in response to the situation to achieve the best possible result.

Some common machine learning algorithms are:

* **Naïve Bayes Classifier Algorithm (Supervised Learning - Classification)**The Naïve Bayes classifier is based on Bayes’ theorem and classifies every value as independent of any other value. It allows us to predict a class/category, based on a given set of features, using probability.  
    
  Despite its simplicity, the classifier does surprisingly well and is often used due to the fact it outperforms more sophisticated classification methods.
* **K Means Clustering Algorithm (Unsupervised Learning - Clustering)**The K Means Clustering algorithm is a type of unsupervised learning, which is used to categorise unlabelled data, i.e. data without defined categories or groups. The algorithm works by finding groups within the data, with the number of groups represented by the variable K. It then works iteratively to assign each data point to one of K groups based on the features provided.
* **Support Vector Machine Algorithm (Supervised Learning - Classification)**Support Vector Machine algorithms are supervised learning models that analyse data used for classification and regression analysis. They essentially filter data into categories, which is achieved by providing a set of training examples, each set marked as belonging to one or the other of the two categories. The algorithm then works to build a model that assigns new values to one category or the other.
* **Linear Regression (Supervised Learning/Regression)**Linear regression is the most basic type of regression. Simple linear regression allows us to understand the relationships between two continuous variables.
* **Logistic Regression (Supervised learning – Classification)**Logistic regression focuses on estimating the probability of an event occurring based on the previous data provided. It is used to cover a binary dependent variable, that is where only two values, 0 and 1, represent outcomes.
* **Artificial Neural Networks (Reinforcement Learning)**An artificial neural network (ANN) comprises ‘units’ arranged in a series of layers, each of which connects to layers on either side. ANNs are inspired by biological systems, such as the brain, and how they process information. ANNs are essentially a large number of interconnected processing elements, working in unison to solve specific problems.  
    
  ANNs also learn by example and through experience, and they are extremely useful for modelling non-linear relationships in high-dimensional data or where the relationship amongst the input variables is difficult to understand.
* **Decision Trees (Supervised Learning – Classification/Regression)**A decision tree is a flow-chart-like tree structure that uses a branching method to illustrate every possible outcome of a decision. Each node within the tree represents a test on a specific variable – and each branch is the outcome of that test.
* **Random Forests (Supervised Learning – Classification/Regression)**Random forests or ‘random decision forests’ is an ensemble learning method, combining multiple algorithms to generate better results for classification, regression and other tasks. Each individual classifier is weak, but when combined with others, can produce excellent results. The algorithm starts with a ‘decision tree’ (a tree-like graph or model of decisions) and an input is entered at the top. It then travels down the tree, with data being segmented into smaller and smaller sets, based on specific variables.
* **Nearest Neighbours (Supervised Learning)**The K-Nearest-Neighbour algorithm estimates how likely a data point is to be a member of one group or another. It essentially looks at the data points around a single data point to determine what group it is actually in. For example, if one point is on a grid and the algorithm is trying to determine what group that data point is in (Group A or Group B, for example) it would look at the data points near it to see what group the majority of the points are in.  
    
  Clearly, there are a lot of things to consider when it comes to choosing the right machine learning algorithms for your business’ analytics. However, you don’t need to be a data scientist or expert statistician to use these models for your business. At SAS, our products and solutions utilise a comprehensive selection of machine learning algorithms, helping you to develop a process that can continuously deliver value from your data.
* **Moving Average Method**
* The moving average is a statistical method used for forecasting long-term trends. The technique represents taking an average of a set of numbers in a given range while moving the range. For example, let’s say the sales figure of 6 years from 2000 to 2005 is given and it is required to calculate the moving average taking three years at a time. In order to calculate the moving average, one would take an average of 2000-2002, 2001-2003, 2002-2004, 2003-2005, and 2004-2006.
* **Auto-Arima Method**
* ARIMA models are generally denoted as ARIMA (p,d,q) where p is the order of autoregressive model, d is the degree of differencing, and q is the order of moving-average model. ARIMA models use differencing to convert a non-stationary time series into a stationary one, and then predict future values from historical data. These models use “auto” correlations and moving averages over residual errors in the data to forecast future values.
* **LSTM Method**
* Long short-term memory (LSTM) is an artificial [recurrent neural network](https://en.wikipedia.org/wiki/Recurrent_neural_network) (RNN) architecture[[1]](https://en.wikipedia.org/wiki/Long_short-term_memory#cite_note-lstm1997-1) used in the field of [deep learning (DL)](https://en.wikipedia.org/wiki/Deep_learning). Unlike standard [feedforward neural networks](https://en.wikipedia.org/wiki/Feedforward_neural_network), LSTM has feedback connections. It can process not only single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected [handwriting recognition](https://en.wikipedia.org/wiki/Handwriting_recognition),[[2]](https://en.wikipedia.org/wiki/Long_short-term_memory#cite_note-2) [speech recognition](https://en.wikipedia.org/wiki/Speech_recognition)[[3]](https://en.wikipedia.org/wiki/Long_short-term_memory#cite_note-sak2014-3)[[4]](https://en.wikipedia.org/wiki/Long_short-term_memory#cite_note-liwu2015-4) and anomaly detection in network traffic or IDSs (intrusion detection systems).LSTM networks are well-suited to [classifying](https://en.wikipedia.org/wiki/Classification_in_machine_learning), [processing](https://en.wikipedia.org/wiki/Computer_data_processing) and [making predictions](https://en.wikipedia.org/wiki/Predict) based on [time series](https://en.wikipedia.org/wiki/Time_series) data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the [vanishing gradient problem](https://en.wikipedia.org/wiki/Vanishing_gradient_problem) that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs, [hidden Markov models](https://en.wikipedia.org/wiki/Hidden_Markov_models) and other sequence learning methods in numerous applications.
* **CHAPTER 2:RELATED WORK**
* Stock determines the share of the ownership of a company. It represents the assets and earnings and overall contribution of the company in any country's economy. The stock of a company is partitioned into shares. Decision making in a stock market is not easy as it involves price trends, market nature, company's stability, different rumors, brand image, venture capitalist funds etc. It becomes very imperative to necessarily extract information that is vital for the people to understand and analyze the risk factors necessarily involved to forecast the stock market from the investor's viewpoint. Thus methods like technical analysis, time series analysis and statistical analysis are an attempt to predict the price but unfortunately none of these methods are a consistently acceptable tool. Hence artificial neural network i.e. a field of Artificial Intelligence is a desired way to discover unknown and hidden patterns of the data. There are two different phases i.e. training and other is predicting. Here Back propagation algorithm is used to training session and Multilayer feed forward network is a network model for predicting price accordingly. This prediction would be done on various parameters that would be considered as input to the multilayer perceptron model. These parameters are depends on data i.e. gained by the company. Major prediction analysis methods are fundamental analysis, technical analysis and machine learning method. 2.1 Fundamental Analysis Fundamental analysis considers economic factors as fundamentals. Fundamental analysis is the physical study of a company with respect to products sales, workers, infrastructure, and quality [2]. This analysis is mostly suitable for long terms prediction as it depends on statistical data of the company [3]. 2.2 Technical Analysis Technical analysis mainly considers indicators on stock charts that will decide the future movement [10]. It normally uses technical data like price, volume, highest and lowest prices to forecast price moments. This kind of analysis is normally suitable for short time span [2]. 2.3 Machine Learning Machine learning method uses artificial Intelligence (AI), for training the system and then use that trained forecasting future movements in stock.
* In machine learning, tasks are generally classified into broad categories. These categories are based on how learning is received or how feedback on the learning is given to the system developed.
* Two of the most widely adopted machine learning methods are supervised learning which trains algorithms based on example input and output data that is labeled by humans, and unsupervised learning which provides the algorithm with no labeled data in order to allow it to find structure within its input data.

## Machine Learning Methods

* In machine learning, tasks are generally classified into broad categories. These categories are based on how learning is received or how feedback on the learning is given to the system developed.
* Two of the most widely adopted machine learning methods are supervised learning which trains algorithms based on example input and output data that is labeled by humans, and unsupervised learning which provides the algorithm with no labeled data in order to allow it to find structure within its input data. Let’s explore these methods in more detail.

### Supervised Learning

* In supervised learning, the computer is provided with example inputs that are labeled with their desired outputs. The purpose of this method is for the algorithm to be able to “learn” by comparing its actual output with the “taught” outputs to find errors, and modify the model accordingly. Supervised learning therefore uses patterns to predict label values on additional unlabeled data.
* For example, with supervised learning, an algorithm may be fed data with images of sharks labeled as fish and images of oceans labeled as water. By being trained on this data, the supervised learning algorithm should be able to later identify unlabeled shark images as fish and unlabeled ocean images as water.
* A common use case of supervised learning is to use historical data to predict statistically likely future events. It may use historical stock market information to anticipate upcoming fluctuations, or be employed to filter out spam emails. In supervised learning, tagged photos of dogs can be used as input data to classify untagged photos of dogs.

### Unsupervised Learning

* In unsupervised learning, data is unlabeled, so the learning algorithm is left to find commonalities among its input data. As unlabeled data are more abundant than labeled data, machine learning methods that facilitate unsupervised learning are particularly valuable.
* The goal of unsupervised learning may be as straightforward as discovering hidden patterns within a dataset, but it may also have a goal of feature learning, which allows the computational machine to automatically discover the representations that are needed to classify raw data.
* Unsupervised learning is commonly used for transactional data. You may have a large dataset of customers and their purchases, but as a human you will likely not be able to make sense of what similar attributes can be drawn from customer profiles and their types of purchases. With this data fed into an unsupervised learning algorithm, it may be determined that women of a certain age range who buy unscented soaps are likely to be pregnant, and therefore a marketing campaign related to pregnancy and baby products can be targeted to this audience in order to increase their number of purchases.
* Without being told a “correct” answer, unsupervised learning methods can look at complex data that is more expansive and seemingly unrelated in order to organize it in potentially meaningful ways. Unsupervised learning is often used for anomaly detection including for fraudulent credit card purchases, and recommender systems that recommend what products to buy next. In unsupervised learning, untagged photos of dogs can be used as input data for the algorithm to find likenesses and classify dog photos together.

**CHAPTER 3:SYSTEM ARCHITECTURE**

* The Dataset used in this project is taken from the Indian Multinational Conglomerate Company **TATA**. The captured Stock Shares data - spans across 5 consecutive years, before Pandemic (**08/10/2013** to **08/10/2018**)

Table

Description automatically generated

Features of Dataset:

* The Data (in CSV File format) is captured chronologically in descending order. [Latest Data are stacked Infront i.e. – at the top (check the screenshot attached above)]
* There are multiple attributes in the dataset – ‘Date’, ‘Open’, ‘High’, ‘Low’, ‘Last’, ‘Close’, ‘Total Trade Quantity’, and ‘Turnover’:

1. The columns ‘**Open’** and ‘**Close’** represent the starting and final price at which the stock is traded on a particular business day.
2. **‘High’**, ‘**Low’** and ‘**Last’** represent the maximum, minimum, and last price of the share for the particular business day.
3. **‘Total Trade Quantity’** is the number of shares bought or sold in a particular business day and ‘**Turnover** **(Lacs)’** is the turnover of the particular company on a given date.

* Date Format = YYYY-MM-DD
* Weekdays [“Saturday”, ”Sunday”] and Holidays are skipped in between months
* No. of Columns present = 08
* No. of Rows (Data Tuples) present = 1235
* Presence of Corrupt/Null/Unidentified objects = NIL
* Presence of Data duplications = NIL
* Significant columns (as per usage in this project) = 03 [“Date”, ”Open”, ”Close” - columns]
* “**Date**” column -> Max = “2018-10-03”, Min = “2013-10-03”, Datatype = Timestamp
* “**Open**” column -> Max = 327.70, Min = 103.00, Avg. = 168.95, Precision = 2 decimal places, Datatype = Numeric [Float]
* “**Close**” column -> Max = 325.75, Min = 102.65, Avg. = 168.73, Precision = 2 decimal places, Datatype = Numeric [Float]
* Software(s) utilized: Jupyter Notebook [version: 6.4.8], Microsoft Excel, Google Collaboratory [Python Interpreter]
* Package(s) utilized: [Python] -> “pandas”, “numpy”, “matplotlib”, “sklearn”, “pmdarima”, “tensorflow”, “keras”
* The Dataset [Data.csv file] is first parsed through Pandas module [“pandas.read\_csv” function]. Then the dataset is loaded inside the data-frame and is checked thoroughly for any discrepancy. E.g.: To check Null values inside the Data-frame (df), this is followed:

Table

Description automatically generated

* Now, as per the questions / evaluations of the project, we must predict closing stock values for a day or for a period. This clearly indicates that we have to consider the “Close” attribute and as well as take other deciding attributes for this. After a close scrutiny, it became obvious to include these attributes : “Date” and “Open”, (since stocks greatly depend on the invested amount at the beginning of a business day and also about in-numerous factors related to the day it is invested [might be an important/crucial day for share markets or might be something related with opening and closing of a week / month])
* Thus, to effectively utilise Machine learning algorithms for this particular use-case, we have to split and train datasets to the model we are considering. So, during this ongoing project, we have experimented with 5 different methods with the same training dataset. [Note: The datasets are split into {1000:235} fashion. 1000 (data) is used for training the models considered, and the rest 235 (data) are used to test and validate the predicted results - with respect to the actual ones]. For E.g.: Here, we are sorting, creating a duplicate data-frame (to avoid unwanted data manipulations of the original dataset) and then finally splitting the data-frame into ‘train’ and ‘valid’ datasets considering only the “Date” and “Close” columns (as shown below):

Text

Description automatically generated

* Machine Learning methods used:

1. **Moving Average**: This is one of the basic naive methods of utilising datasets for machine learning use cases. Here, the predicted “closing stock price” for each business day will be calculated - based on the average of a set of previously captured / observed values. Thus, instead of using the simple average, we will be using this method / technique using the latest set of values, for each prediction. In other words, for each subsequent step, the predicted values are taken into consideration while removing the oldest captured value from the dataset. The logic applied: Graphical user interface, text, application

   Description automatically generated
2. **Linear Regression**: This is also one of the most popular machine learning method used, till date. The linear regression model basically returns and holds an equation that determines the relationship between independent and dependent variables.

The equation for ‘linear regression’ can be written as:

Diagram

Description automatically generated

Here, we are referring “Close” attribute to be the dependant variable, and utilising “Date” and “Open” attributes as the independent ones. Likewise, we have trained and fitted the model using both the attributes as input. Here, in case of “Date” attribute, a slight modification has been made to mark as a feature – The starting and ending of the weekdays [excluding holidays] are marked as ‘1’ and rest are kept at ‘0’. This way we can normalize the values and proceed further.

**For “Date” and “Close” attribute**

Graphical user interface, text, application

Description automatically generated

**For “Open” and “Close” attribute**

Text

Description automatically generated

1. **k-Nearest Neighbours**: This is another well known and widely used machine learning method. Based on the independent variables, kNN finds the similarity between test data points and old (trained) data points. As the name suggests, the model basically determines a zonal mapping of points based on clustering of data(s). Thus whenever the input data is provided, the model maps the data as per the “best-fit” equation and hence finds the corresponding output. Also, this has to be kept in mind that we have a “k-factor” associated to the regression of this model which will fine tune the graph’s equation and give more precise and accurate results. Thus, to aid in this process we have used “Grid Search” module to auto-determine the best fit “k- value” [this determines the amount of data-points considered for model training] corresponding to the input dataset.

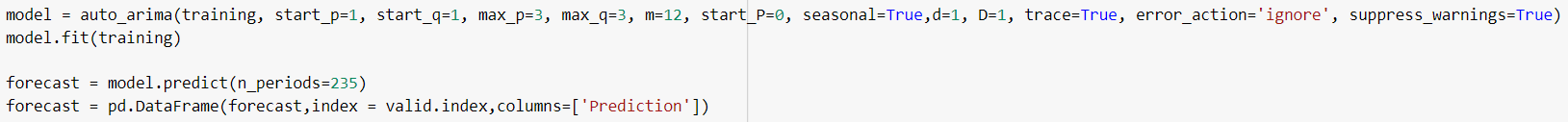
Text, letter

Description automatically generated

1. **Auto ARIMA**: This method is a well known statistical approach to determine / predict values as per the given historical data (for the training set). Mainly used in “Time-series forcasting” use-cases, this approach comprises:

* p (past values used for - forecasting the next value)
* q (past forecast errors used to - predict the future values)
* d (order of differencing)

To tune the model created, we can change the hyper-parameters of “p”, “q”, “d” with respect to start, max and period attrributes. For this dataset the following tuning is done for best results:



1. **Long Short Term Memory (LSTM)**: In Tensor-Flow and Deep learning methods, this method is one of the widely and most effectively used to train predicting models. This method basically follows a gate logic - wherein it stores useful and important information and forgets insignificant data points. LSTM has namely 3 gates:
2. **Input** gate: The input gate adds information to the cell state
3. **Forget** gate: It removes the information which is no longer required by the model
4. **Output** gate: Output Gate at LSTM selects the information to be shown as output

Now, after fitting the model with the required train data, we tune the hyper-parameters (epoch value, batch-size, verbose, loss attribute etc) and plot the appropriate output graph (Shares vs Time). The initial creation and fitting of the model is done as follows:

Text

Description automatically generated

* The following DFD (Data Flow Diagram) will chalk out the basic overall flow and architecture (as aforesaid) of this machine learning project:

A picture containing diagram

Description automatically generated

**CHAPTER 4:RESULTS**

After implementation of machine learning models, the success rate of prediction and efficiency of those models are calculated - based on some predefined metrics. Alongside, plotting of graphs gives a visual output to the predicted data. Here, in this section we will go through each model one by one and justify the best model fitted with this current “Stock-Market” use-case.

As discussed above, let’s talk about some important model metrics:

* **Confusion Matrix**: This is basically a table (in matrix format – 2\*2) which is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the ‘true values’ [actual] are known.

Chart

Description automatically generated with medium confidence**Schematic Diagram of a Confusion** **Matrix**

Matrix properties:

1. The target variable has mainly 2 values: **Positive** or **Negative**
2. The **columns** represent the actual values of the target variable
3. The **rows** represent the predicted values of the target variable
4. Each cell represents a value corresponding to TP,FP,FN,TN i.e. ->

**True Positive (TP)**

* The predicted value matches the actual value
* The actual value was positive, and the model also predicted a positive value

**True Negative (TN)**

* The predicted value matches the actual value
* The actual value was negative, and the model also predicted a negative value

**False Positive (FP)** – [Type 1 error]

* The predicted value was falsely predicted
* The actual value was negative, but the model predicted a positive value

**False Negative (FN)** – [Type 2 error]

* The predicted value was falsely predicted
* The actual value was positive, but the model predicted a negative value

Now, using these properties we will calculate the rest of the metrices and proceed further.

* **Precision**: Precision tells the user how many of the correctly predicted cases actually turned out to be positive. The formula to calculate this metric is:

Text, letter

Description automatically generated

* **Recall**: Recall tells the user how many of the actual positive cases we were able to predict correctly with the chosen model. The formula to calculate this metric is:

Text

Description automatically generated

* **F1 Score**: F1-score is mathematically a harmonic mean of ‘Precision’ and ‘Recall’, and so it gives a combined idea about these two metrics. It is maximum when Precision is equal to Recall. The formula to calculate this metric is:

Diagram

Description automatically generated

* **AUC Score**: The Area Under the Curve (AUC) score is the measure of the ability of a model classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.
* **Mean Absolute Error (MAE)**: In context of machine learning, absolute error refers to the magnitude of difference between the prediction of an observation and the true value of that observation. MAE takes the average of absolute errors for a group of predictions and observations as a measurement of the magnitude of errors for the entire group. MAE can also be referred as L1 loss function. Thus, this is calculated by taking the summation of the absolute difference between the actual and calculated values of each observation over the entire array and then dividing the sum obtained by the number of observations in the array. The formula to calculate this metric is:

A picture containing text, clock, watch

Description automatically generated

* **Mean Squared Error (MSE)**: MSE is the average of the squared error that is used as the loss function for least squares regression. It is basically the sum over all the data points, of the square of the difference between the predicted and actual target variables, divided by the number of data points. The formula to calculate this metric is:

Text

Description automatically generated with medium confidence

* **Max Error (ME)**: ME or Maximum Error is the absolute value of the most significant difference between a predicted variable and its real value. The formula to calculate this metric is:

**Text, logo

Description automatically generated**

Now, we will go step by step analysing all the results that we have, after implementing the models.

[F.Y.I – In the upcoming plots, due to space management - the legends might not be shown in the screenshots taken. So, please consider the following important keys for reference:

* Blue Line – Trained Data Set
* Green Line – Test Data Set
* Red Line – Predicted Data Set]
* **Initial Step (Data Preparation):** Upon parsing and creating a data-frame from the input dataset (CSV File), we plot the “Close” attribute with respect to the “Date” attribute. The python module used here is “Matplotlib”. The graph plotted is as follows:

Graphical user interface, chart

Description automatically generated

**Observation**: Data looks good and has a noticeably huge surge at the end of the dataset range i.e. after mid of the year – ‘2017’

* **First Model Implementation -> “KNN Method”**

The graph is plotted after the best fit value of “k” determined by ‘GridSearchCV’ module:

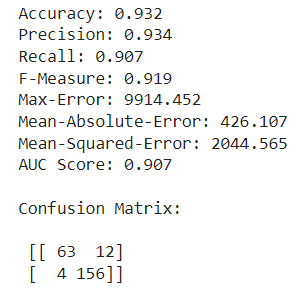
Chart, line chart

Description automatically generated

**Observation:** The graph / plot is well plotted with crests and troughs, but the data values predicted are pretty less than expected.

**Metrics Observation**:

**Evaluation**: Low - Moderate range of model efficiency



* **Second Model Implementation -> “Moving Average Method”**

A very basic and naïve statistical implementation, plots the following graph:

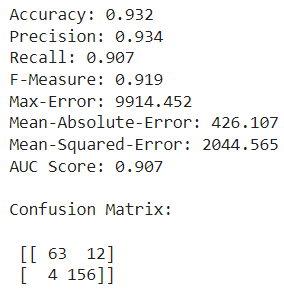
Chart, line chart

Description automatically generated

**Observation:** The graph / plot shows very less efficiency of predictions just with a small surge at the end.

**Metrics Observation**:

**Evaluation**: Very Low model efficiency [Worst Model]



* **Third Model Implementation -> “Auto Arima Method”**

Known for Statistical Time series forecasting, the model is best fitted with the tuning hyper-parameters (p, q, d etc) and is plotted as follows:

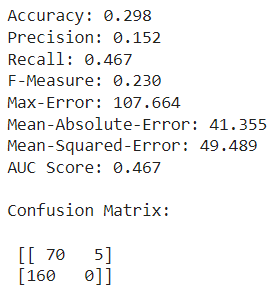
Chart, line chart

Description automatically generated

**Observation:** The graph / plot looks progressively linear, where it intersects at the end.

**Metrics Observation**:

**Evaluation**: Low - Moderate model efficiency



* **Fourth Model Implementation -> “Linear Regression Method”**

After fitting the model with attributes, the plot looks great:

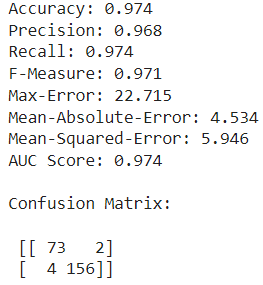
Chart, line chart, histogram

Description automatically generated

**Observation:** The graph / plot looks very promising with much better accuracy and precision of the predicted values.

**Metrics Observation**:

**Evaluation**: Moderately high - High model efficiency



* **Fifth Model Implementation -> “LSTM Method”**

One of the most basic, efficient and complex method is taken from Tensor Flow and the model is fitted with data-sets after fine tuning the appropriate hyper-parameters (epoch values, verbose, batch size, loss optimizer etc). The graph is then plotted as below:

Chart

Description automatically generated with medium confidence

**Total Data-set History [Blue Line – Trained Data, Red Line – Actual Test Data, Black – Predicted Test Data]**

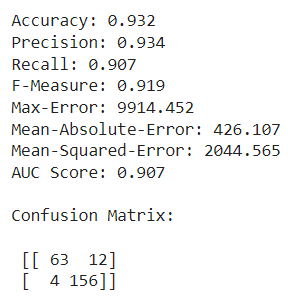
**Chart, line chart, histogram

Description automatically generated**

**Zoomed Plot [Only Predicted and Actual Test Data is shown]**

**Metrics Observation**:

**Evaluation**: Very High model efficiency [Best Model]



Thus LSTM Model is being used for the rest of the use cases related to this dataset. The following results are captured as per the Problem Statements Provided:

* Q. To create a model for predicting the next day’s closing price based on last n-days of data.

**Solution:**

Here, **N**  is taken as 60 (2 months roughly) and predicted the next day’s closing share value.

Code snippet:

Text, letter

Description automatically generated

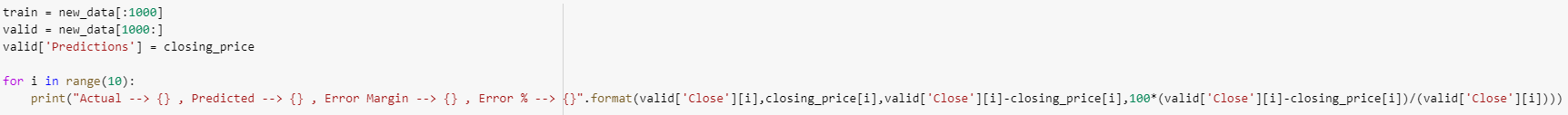
**Output:** ₹ 211.427571 /-

Graphical user interface, text, application

Description automatically generated

* Q. Once you predict the next day’s closing value, do the same for 10 days data (How close are you for 10 predictions)?

**Solution:**



**Code Snippet**

**Output:**

**Table

Description automatically generated with low confidence**

* Q. Using this model, if you invest 10000 for each of the 10 days wherever it shows profit. What is your return value?

**Solution / Output**:

Revenue % Achieved **=** 0.83%

**Text

Description automatically generated**

**Project’s Google Collab Link:**

https://colab.research.google.com/drive/1gdK-Q5KwMtjqClhJzJC5nSjefvL4RQG9?usp=sharing

**CHAPTER 5:CONCLUSION AND FUTURE WORK**

This project revolves around - how we can predict the statistical stock market data, using the historical data that we have in hand. The things/metrics to focus when we commit predictions are:

* Accuracy
* Precision
* Reliability
* Consistency

So firstly, the workflow that needs to be followed is :- analysing the input data. When we are done with cleaning, scrubbing and augmenting the dataset into data frames, we split and train the datasets to specific models and test their accuracy by tuning the related hyper-parameters.

The model that gives the best results are chosen and moved forward with specific use cases / requirements. This is the general procedure, that is followed for all the machine learning problems.

Now, the most crucial part of this entire workflow is understanding the data. If the user working on the datasets doesn’t understand the properties of the data, it becomes really difficult to proceed through. The main reason being :- Every dataset is unique from each other. So, not only working with Stock market prediction dataset, but there are also several use cases where we have to predict a type of fruit, or maybe analyse transactions for fraud management, or may be track human or animal emotions. For all these problems, we have to understand and analyse the data, else otherwise there is no chance of fitting the models accurately.

Now, upon working on this particular project topic, we are capable to learn and understand how datasets are being used to predict closing prices of the stocks for upcoming days and scenarios.

Though we have to keep in mind that stocks largely depend on various circumstantial factors around the globe. So, ideally if everything is ignored, we can approximately predict the upcoming extrapolations of the plot, using the model that we chose best for this problem statement. This is how various organizations and companies can get a brief overview about how their invested stocks may vary over time. This not only gives them a better understanding, but also aids them a lot in finance and management sectors of their companies which in turn carries them way forward – a step ahead of it’s other business competitors.

Now theoretically,

the term – “Accurate Prediction” is defined when we get plot metrics result equal to 100%. But everyone knows that - this is practically impossible to achieve. So what is the solution? The only thing that we need to follow and keep in mind that - we have to go by hit and trial methods of considering models which supports our problem statement. Maybe we are not successful at our initial attempts, but with deeper insights and investigation we are sure to find out a more advanced one which will align and give us better results. Here in this project, we have seen that, when we proceeded forward from basic naïve model implementations to Tensor Flow modules, the metrics became more precise and closer to the desired ones. So, likewise for any user who will follow up and go through this project, might look up for more models relating to deep machine learning procedures for better estimations and enhancements. Fun fact: Nobody knows what future may behold!

**BIBILOGRAPHY/REFERENCES:**

While our implementations and brainstorming this project’s solution, there are a couple of articles, links and whitepapers that have been gone through for more information on the related work. So, for more work related references, some of the most useful resources are included below:

* [https://www.digitalocean.com/community/tutorials/an-introduction-to-machine-learning](about:blank)
* <https://www.geeksforgeeks.org/learning-model-building-scikit-learn-python-machine-learning-library/?ref=lbp>
* <https://www.tensorflow.org/guide/keras/rnn>
* <https://neptune.ai/blog/predicting-stock-prices-using-machine-learning>
* <https://iopscience.iop.org/article/10.1088/1742-6596/2161/1/012065/pdf><https://www.mit.edu/~6.s085/notes/lecture3.pdf>
* <https://towardsdatascience.com/time-series-forecasting-using-auto-arima-in-python-bb83e49210cd>
* <https://realpython.com/knn-python/>
* <https://www.linkedin.com/pulse/machine-learning-stock-price-forecasting-13-ali-el-shayeb>
* <https://www.sciencedirect.com/science/article/pii/S1877050920307924>
* Murkute, Amod, and Tanuja Sarode. (2015) “Forecasting market price of stock using artificial neural network.” International Journal of Computer Applications 124 (12) : 11-15
* Seber, George AF and Lee, Alan J. (2012) “Linear regression analysis.” John Wiley & Sons 329