**CHAPTER 1**

**INTRODUCTION**

* 1. **EXISTING SYSTEM**

Existing Systems use only one data source hence the data is extremely biased

When Traditional Classifiers are used, Stock Prices seem to be unpredictable

Existing systems do not focus on external events in the environment like news or social media. Other existing systems are manual or analog in nature and are highly speculative, there’s no proper way of measuring its accuracy or precision.

Existing systems do not perform well when there is a change in operating environment.

* 1. **PROPOSED SYSTEM**

We propose to build a standalone application to predict a stock price using more than one method of prediction from various data sources.

In our proposed system we have two main components for supporting buyer decision.

1. Price Prediction using Regression and Classification.
2. Sentiment Analysis to determine market sentiment of a particular stock.

In Price Prediction we use various algorithms to determine the price, out of these various algorithms we perform model evaluation to select the best model that’s fitting our data.

The various algorithms are:

1. Multivariate Linear Regression
2. Artificial Neural Networks using Backpropagation
3. Random Forest
4. Sentiment Analysis
   1. **MOTIVATION**

Businesses in most cases run over purchaser’s pleasure, purchaser opinions approximately their products. Shifts in sentiment on social media were proven to correlate with shifts in inventory markets. Identifying purchaser grievances thereby resolving them results in purchaser pleasure in addition to trustworthiness of an organization. Hence there's a need of an unbiased computerized machine to categorize purchaser opinions regarding any problem. In today’s surroundings in which we’re justifiably laid low with statistics overload (even though this doesn't imply higher or deeper insights), businesses may have mountains of purchaser remarks collected; however, for mere humans, it’s nevertheless not possible to examine it manually with none type of mistakes or bias. Oftentimes, businesses with the quality intentions locate themselves in an insights vacuum. You recognize you want insights to tell your decision making and you understand that you’re missing them, however, don’t recognize how quality to get them. Sentiment evaluation offers a few solutions into what the maximum vital problems are, from the attitude of customers, at least. Because sentiment evaluation may be computerized, selections may be made primarily based totally on a tremendous quantity of statistics as opposed to simple intuition.

* 1. **OBJECTIVES OF THE WORK**

Our main objective is to help support buyer decisions in Intraday trading of stocks in the Indian Stock Market. We aim to achieve our main objective by trying to predict stock prices and determining user sentiment.

We use two methods to predict stock prices, they are:

1. Using Regression and Neural Networks to predict the closing price of a stock for the next day
2. Using Classification to determine whether the stock price will rise, fall or stay approximately the same for the next day

We also use Sentiment Analysis as a supporting factor as user sentiment plays a huge role in determining stock prices.

**CHAPTER 2**

**LITERATURE SURVEY**

Literature survey is an assignment of previous task done by some authors and collection of information or data from research papers published in journals to progress our task. It is a way through which we can find new ideas, concept. There is lot of literatures published before on the same task; some papers are taken into considerations from which gives the ideas of the project is taken.

2.1: TITLE: Survey of Stock Market Prediction Using Machine Learning Approach

Author: ASHISH SHARMA, DINESH BHURIYA, UPENDRA SINGH

Published in 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)

Publisher: IEEE

In this project they have used efficient regression approach to predict the stock market price from stock market data based. Prediction plays a very important role in stock market business which is very complicated and challenging process. To achieve this, they used regression analysis to model and analyse multiple variables.

**Merits and Demerits**

This study provided a theoretical approach for predicting stock prices using a basic regression model and was developed by Ashish Sharma and his colleagues. This study described regression models and how their applications can help with price forecasting. This study gave us an idea of how to provide some forecasts by choosing the appropriate factors that affect stock prices as variables.

2.2: TITLE: Analysis and Forecast of Nepalese Stock Prices Using Various Machine Learning Algorithms

Authors: Pushkar Khanal, Shree Raj Shakya

Published: Minutes of the 2016 IOE Graduate Meeting

Publisher: IEEE

In this project, we used machine learning algorithms to predict patterns and analyse stock price trends. Increased accuracy, reliability, and changeability have also increased our reliance on smart trading systems to help us predict and analyse stock prices in a variety of situations. Forecasting methods based on the type of data and the type of tools each method uses to forecast the market are divided into technical analysis methods, fundamental analysis methods, traditional time-series forecasting methods, and machine learning methods.

An algorithm called a support vector machine in combination with boosting provided the most accurate results compared to most other machine learning algorithms and traditional engineering techniques. On the other hand, the mean square error is lower even in linear regression.

2.3: TITLE: Multi-kernel learning to predict the direction of stock prices

Authors: Amit Kumar Shirohi, Pradeep Kumar Mahat, Ph.D. Vahida Atter

Published in: International Conference on Advances in Engineering & Technology Research

Publisher: IEEE

This project used multiple kernel learnings for stock price forecasting. It aims to build a kernel model that linearly combines fixed-based kernels to create a kernel. The success of an SVM depends on the selection of the appropriate pre-crafted and prepared kernel. They used MKL to integrate different types of kernels.

The pre-processing component first collects and processes raw data from the market, then extracts some technical features or indicators based on historical stock prices and trading volumes, and finally normalizes the entire function set. did. The forecasting component first builds different base kernels (RBF and polykernel) on the normalized dataset, then combines these base kernels in MKL, and then rises on the next trading day after the previous trading day. Or the day you set the criteria for predicting the movement of daily stock price trends such as downtrends. The Performance component calculates the prediction accuracy to evaluate the performance of the proposed and baseline methods.

This task required the appropriate set of functions to predict the most accurate values. This study by Amit Kumar Sirohi described the two-stage model in detail. The first stage provided information on selecting features such as open and close prices, and the second stage built a different kernel for the extracted features. This gave me a good idea of how to apply the appropriate function to the model.

2.4: TITLE: Neural network by predicting stock market data

Authors: Rohit Verma, P Kumar Chore, Upendra Singh

Published in: International Conferences on Electronics, Communications and Aerospace Technologies

Publisher: IEEE

This article described an artificial neural network application for the task of stock index forecasting. They explained the theory behind ANN and the neural network model and its salient features.

The results obtained in both cases were accurate. The predictions are accurate unless there is a large sudden difference in the actual data, such as the right edge where it becomes impossible to predict the changes accurately. On the other hand, this also proves the hypothesis that the stock market is unpredictable. The minimum error in test and training data was as low as 3.5% for a single hidden layer.

Therefore, we can see that neural networks are an effective tool for forecasting stock markets and can be used with real-world datasets such as the Nifty dataset. From this project, it was recognized that deep learning models can significantly improve accuracy. Rohit Verma and his colleagues have proposed a theoretical approach for predicting stock prices using artificial neural networks. Here, the results were taken from the Nifty Stock Index dataset based on the values from the last few days. This study gave 96% accuracy.

**CHAPTER 3**

**SYSTEM REQUIREMENTS AND SPECIFICATION**

**3.1 SYSTEM ANALYSIS**

System analysis is a method of enquiring and understanding an existing problems system, defining requirements and analyzing the best possible solution is the best way to increase the efficiency of the existing system. It is also considered to be a great problem-solving methodology that will break the existing system into different component modules and evaluate each module to determine the efficiency of the system individually, by doing module wise analysis we will be able to individually increase the speed of each module.

**3.2 FUNCTIONAL REQUIREMENTS**

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

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

**3.3 NON-FUNCTIONAL REQUIREMENTS**

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

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

**3.4 TOOLS AND TECHNOLOGIES REQUIRED**

**Hardware Requirements**

Processor: Intel i5 or above.

RAM: Minimum 225MB or more.

Hard Disk: Minimum 2 GB of space

Input Device: Keyboard and Mouse

Output Device: Screens of Monitor or a Laptop

**Software Requirements**

1. IDE
   1. PyCharm for Application Development
   2. Jupyter Notebook for Testing and Prototyping
2. Python Packages
   1. Keras
   2. Scikit-Learn
   3. NLPK

**CHAPTER 4**

**SYSTEM DESIGN**

System design is mechanism or the flow of outlining the structural design, sections, interfaces, and data for a system to fulfil its definite requirements. The system design is critical to the creation of the outline and the development of the product. The primary goal of this study is to offer the system's needed functionality to the end user. User must obtain the predicted output in accurate manner and time-efficient manner.

# **4.1 SYSTEM ARCHITECTURE**

Diagram

Description automatically generated

Overall System Architecture for Price Prediction and Sentiment Analysis

# **4.2 INPUT/OUTPUT DESIGN**

We’ve developed a GUI for our application, we used the Python Tkinter to build the GUI. Tkinter is available on Windows and Linux/Unix.

There are 2 main components in our GUI.

1. Data Input
   1. Stock Name
   2. Dataset Input (CSV File)
2. Output

The output section is divided into two categories

* 1. Prediction Output
     1. Price Prediction
     2. Price Classification
  2. Sentiment Analysis Output

The Image below shows a rough version of the GUI and its structure

Graphical user interface, website

Description automatically generated

# **4.3 OBJECT ORIENTED DESIGN**

**4.3.1 SEQUENCE DIAGRAM**

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

Sequence diagrams can be useful references for businesses and other organizations. Try drawing a sequence diagram to:

• Represent the details of a UML use case.

• Model the logic of a sophisticated procedure, function, or operation.

• See how objects and components interact with each other to complete a process.

• Plan and understand the detailed functionality of an existing or future scenario.

**Timeline

Description automatically generated**

* + 1. **ACTIVITY DIAGRAM**

An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system.

An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

**Diagram

Description automatically generated**

* + 1. **DATAFLOW DIAGRAM**

A data flow diagram is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing

**Diagram

Description automatically generated**

# **4.4 ALGORITHM**

**4.4.1 MULTIVARIATE LINEAR REGRESSION**

Multivariate linear regression is the generalization of the univariate linear regression. As the name suggests, there are more than one independent variables, x1,x2⋯,xn and a dependent variable y.

**4.4.2 RANDOM FOREST**

The Random forest procedure functions by instructing numerous unstable classification trees with the help of a secure amount of unintentionally chosen qualities. Later captivating the approach of separate class to construct a robust classifier. Once the training set for the present tree is sketched by sampling with a substitute, almost one-third of the instances are ignored from the sample. The missing information is swapped by proximities, locating outliers, and producing illuminating low-dimensional interpretations of the data. Nevertheless, as this mode picks a controlled number of qualities in each iteration, the implementation of random trees is quicker. The random-trees technique initiates a component of unpredictability into the prototype. Instead of viewing for the elite quality though splitting a node, it examines for the finest items amongst an unplanned subset of qualities. This development commonly produces in an enhanced model.

Diagram

Description automatically generated

Figure 4.2.2 Random Forest