

FORECASTING STOCK PRICE

PROJECT THESIS

SUBMITTED

TO

AWH ENGINEERING COLLEGE

KUTTIKKATTOOR, CALICUT-8

IN PARTIAL FULFILMENT

OF THE REQUIREMENTS FOR THE AWARD OF THE

DEGREE OF

Master of Computer Application

BY

KISHAN LAL P T



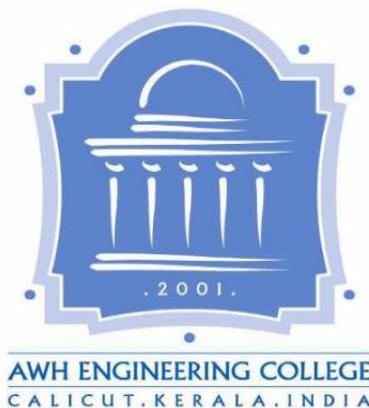
DEPARTMENT OF COMPUTER APPLICATIONS

AWH ENGINEERING COLLEGE, KUTTIKKATTOOR.

CALICUT

JULY 2020

DEPARTMENT OF COMPUTER APPLICATIONS



AWH ENGINEERING COLLEGE

CALICUT

CERTIFICATE

This is to certify that this thesis entitled "FORECASTING STOCK PRICE" submitted herewith is an authentic record of the thesis work done by KISHAN LAL P T (LAWH17MCA013), under our guidance in partial fulfilment of the requirements for the award of Master of Computer Applications from APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY during the academic year 2020.

Mrs. Shareeja T
Assistant Professor
Head of the Department

Mrs.Sruti Sudevan
Assistant Professor
Project Guide

External Examiner

Internal Examiner

ACKNOWLEDGEMENT

I express my sincere gratitude to our beloved principal **Dr.Sabeena M V**, for providing me an opportunity with the required facilities for doing this project. I express my hearty thanks to **Mrs. Shareeja T**, Head of department of MCA, for her guidance. I also wish to thank **Mrs. Sruti Sudevan** for timely guidance and advice. I am thankful to all other staff of MCA department for their encouragement, timely guidance, valuable suggestions and inspiring ideas given throughout this main project. I am grateful to my friends for the way they have co-operated, expected me to achieve success and have always stirred my ambition to do the best. Above all, I am grateful to The Almighty, who has showered His blessings on me all throughout my life and throughout the main project.

KISHAN LAL P T

ABSTRACT

Forecasting stock market prices has always been a challenging task for many business analysts and researchers. In fact, stock market price prediction is an interesting area of research for investors. For successful investment, many investors are interested in knowing about the future situation of the market. Effective prediction systems indirectly help traders by providing supportive information such as the future market direction. Data mining techniques are effective for forecasting the future by applying various algorithms over data.

The aim of this project is to predict the daily adjusted closing prices of a company's Stock Market price, using data from the previous N days. And with the help of that data, check how many days it can predict accurately. This project is using at least thirty years of historical prices, which is downloaded from authorized websites. In this project, deep learning, Time series analytics, LSTM (Long short-term memory) are used.

Broadly, stock market analytics is divided into two parts – Fundamental Analysis and Technical Analysis. Fundamental Analysis involves analysing the company's future profitability based on its current business environment and financial performance. Technical Analysis, on the other hand, includes reading the charts and using statistical figures to identify the trends in the stock market. In this project, focus will be on the technical analysis part.

CONTENTS

	PageNo.
1. INTRODUCTION	1
2. SYSTEM ANALYSIS	4
2.1 Existing System	5
2.2 Proposed System	6
2.3 Module Description	6
2.4 Sprint	7
2.5 User Stories	8
3. FEASIBILITY STUDY	10
3.1 Economic Feasibility	12
3.2 Technical Feasibility	13
3.3 Operational Feasibility	14
3.4 Behavioral Feasibility	16
3.5 Software Feasibility	16
3.6 Hardware Feasibility	16
4. SOFTWARE ENGINEERING PARADIGM	17
4.1 Agile Model	18
4.2 Scrum	21
5. SYSTEM REQUIREMENT SPECIFICATION	23
5.1 Software Requirements	24
5.2 Hardware Requirements	25
6. SYSTEM DESIGN	26
6.1 Input Design	27
6.2 Output Design	28
6.3 Database Design	29
6.4 UML Design	31

7. SYSTEM DEVELOPMENT	40
7.1 Coding	41
7.2 Algorithm	49
8. SYSTEM TESTING AND IMPLEMENTATION	54
8.1 Types of Testing	56
8.2 Implementation	57
9. SYSTEM MAINTENANCE	59
10. FUTURE ENHANCEMENT	61
11. CONCLUSION	63
12. APPENDIX	65
13. BIBLIOGRAPHY	75

INTRODUCTION

1. INTRODUCTION

Stock price forecasting is a popular and important topic in financial and academic studies. Share Market is an untidy place for predicting since there are no significant rules to estimate or predict the price in the share market. Many methods like technical analysis, fundamental analysis, time series analysis and statistical analysis, etc. are all used to attempt to predict the price in the share market.

Stock Market prediction and analysis is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange. Stock market is the important part of economy of the country and plays a vital role in the growth of the industry and commerce of the country that eventually affects the economy of the country. Both investors and industry are involved in stock market and wants to know whether some stock will rise or fall over certain period of time.

The stock market is the primary source for any company to raise funds for business expansions. It is based on the concept of demand and supply. If the demand for a company's stock is higher, then the company share price increases and if the demand for company's stock is low then the company share price decrease.

Forecasting stock market prices has always been challenging task for many business analyst and researchers. In fact, stock market price prediction is an interesting area of research for investors. For successful investment lot many investors are interested in knowing about future situation of market. Effective prediction systems indirectly help traders by providing supportive information such as the future market direction. Data mining techniques are effective for forecasting future by applying various algorithms over data.

There are a lot of complicated financial indicators and also the fluctuation of the stock market is highly violent. However, as the technology is getting advanced, the opportunity to gain a steady fortune from the stock market is increased and it also helps experts to find out the most informative indicators to make a better prediction. The prediction of the market value is of great importance to help in maximizing the profit of stock option purchase while keeping the risk low.

The aim of this project is to predict the daily adjusted closing prices of a company's Stock Market price, using data from the previous N days. And with the help of that data check how many days it can predict accurately. This project is using at least thirty years of historical prices, which is downloaded from authorized websites. In this project deep learning, Time series analytics, LSTM (Long short-term memory) are used.

Long Short-Term memory is one of the most successful RNNs architectures. LSTM introduces the memory cell, a unit of computation that replaces traditional artificial neurons in the hidden layer of the network. With these memory cells, networks are able to effectively associate memories and input remote in time, hence suit to grasp the structure of data dynamically over time with high prediction capacity.

Broadly, stock market analytics is divided into two parts – Fundamental Analysis and Technical Analysis. Fundamental Analysis involves analysing the company's future profitability based on its current business environment and financial performance. This includes characteristics such as financial results, company's assets, liabilities, and stock and growth forecasts. It's very important to understand that this type of analysis is not static; newly released financial information, corporate announcements and other news can influence the fundamental outlook of a company. Fundamental analysis requires expertise in a particular sector and is often conducted by professional analysts. Their recommended investments are regularly published and updated.

Technical Analysis, on the other hand, includes reading the charts and using statistical figures to identify the trends in the stock market. Technical analysis does not try to gain deep insight into a company's business. It assumes the available public information does not offer a competitive trading advantage. Instead, it focuses on studying a company's historical share price and on identifying patterns in the chart. The intention is to recognize trends in advance and to capitalize on them. In this project focus will be on the technical analysis part.

SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

System analysis is the process of gathering facts, diagnosing problems and using this information to recommend improvements of the system. Thus, an analysis is inevitable part of any system for their better working and satisfied performance.

2.1 Existing System

There are many existing system nowadays for forecasting stock price. Data are humongous, nowadays there are a rapid-explosion of numerical stock-quotes and textual data. They are provided from all different sources. Demand forecasts are important since the basic operations management process, going from the vendor raw materials to finished goods in the customer's hands, takes some time. Most firms cannot wait for demand to elevate and then give a reaction. Instead, they make-up their mind and plan according to future demand so that they can react spontaneously to customer's order as they arrive.

Generally, demand forecasts-lead to good ops and great levels of customer satisfaction, while bad forecast will definitely lead to costly ops and worst levels of customer satisfaction. A confusion for the forecast is the horizon, which is, how distant in the future will the forecast project. As a simple rule, the away into the future can see, the more blurry of vision will become distant forecasts will be inaccurate that short range forecasts.

Support vector regression (SVR), a variant of the SVM, is typically used to solve nonlinear regression problems by constructing the input-output mapping function. The least squares support vector regression (LSSVR) algorithm is a further development of SVR and its use considerably reduces computational complexity and increases efficiency compared to standard SVR. The Firefly Algorithm (FA), which is a nature-inspired metaheuristic method, has recently performed extremely well in solving various optimization problems.

Disadvantages of Existing system

- The system does not allow the import of raw data directly.
- Uses ordinary Recurrent Neural Network (RNN).
- The system does not have a user-interface which can be distributed as a web app.

2.2 Proposed System

Basically the main objective of this project is to collect the stock information for some previous years and then accordingly predict the results for the predicting what would happen next. So forth are going to use of two well-known techniques neural network and data mining for stock market prediction. Extract useful information from a huge amount of data set and data mining is also able to predict future trends and behaviors through neural network. Therefore, combining both these techniques could make the prediction more suitable and much more reliable. Aim of this project is to predict the daily adjusted closing prices of Companies. Stock Market price, using data from the previous N days and with the help of that data are checking how many days can predict accurately. This system is using deep learning, Time series analytics, LSTM (Long short-term memory) for this project.

Advantages of proposed system

- Easy prediction and focuses on optimization.
- Increase stock share of company.
- The system can be extended to analyse multivariate time series data and import raw dataset directly.
- Technical analysis includes reading the charts and using statistical figures to identify the trends in the stock market.
- Uses LSTM(Long Short-Term Memory).
- The development of a web-based application has been considered to improve the user-friendliness and usability of the expert system.

2.3 Module Description

Admin

- Registration
- Model Creation and Training
- Predict Stock Price
- View Graphical Representations

User

- Registration
- Predict Stock Price
- View Graphical Representations

2.4 Sprint**Sprint 1**

MODULES	TASK	PENDING TASK IF ANY	HOUR FOR COMPLETION	EXPECTED DATE OF COMPLETION	ACTUAL DATE OF COMPLETION	REASON FOR DEVIATION
Admin	Selecting and downloading suitable Dataset	-	4hrs	20-02-2020	20-02-2020	-
	Model Creation	-	30hrs	10-03-2020	14-03-2020	Kaggle/ Google Colab delay
	Form Designing	-	10hrs	24-03-2020	24-03-2020	-
	Prediction	-	10hrs	20-04-2020	19-04-2020	-

Sprint 2

MODULES	TASK	PENDING TASK IF ANY	HOUR FOR COMPLETION	EXPECTED DATE OF COMPLETION	ACTUAL DATE OF COMPLETION	REASON FOR DEVIATION
User	Form Designing	-	4hrs	20-02-2020	20-02-2020	-
	Front-end Creation	-	30hrs	04-03-2020	07-03-2020	Error
	Prediction	-	20hrs	28-04-2020	28-04-2020	-

2.5 User Stories

The end user of the system will be Data Analyst or Business Analyst of a company who will be admin. Other employees which the company decide must be the users. This system can also be used by individuals who can support with high performance hardware capabilities. In the present scenarios if a company want to increase its boundaries they need funds, so in order to raise funds they trade their shares in share market. So whether they want to invest it in or not is based on the trend that goes in this market. There are no accurate prediction for the stock market price in reality, because this market is very dynamic in nature, many internal and external factors can fluctuate the graphs of share of the company. This system propose a prediction on closing price of a company that is more accurate, but as said earlier stock market is very untidy area and dynamic in nature that will affect the accuracy of the system. This system provide much accuracy in prediction for about to one to three days.

In this system data is initially collected from online sources such as yahoo finance. The data is then used to train the system by admin, trained model is saved. Using this model, closing prices are predicted. User uploads a company's stock data. He requests for prediction to be made. The Stock Market Prediction system with the model already trained using the data predicts closing price. Result is displayed along with graphical representations. Various tests are done in system for the finding the accuracy of the system, it predicts accurately most of the time. Dynamic behaviour in stock market, internal and external factors may have effect in the system proposed.

FEASIBILITY STUDY

3. FEASIBILITY STUDY

A feasibility study is a preliminary study undertaken to determine and document a project's viability. The results of this study are used to make a decision whether to proceed with the project. If it indeed leads to a project being approved, it will - before the real work of the proposed project starts - be used to ascertain the likelihood of the project's success. It is an analysis of possible alternative solutions to a problem and a recommendation on the best alternative. It, for example, can decide whether an order processing be carried out by a new system more efficiently than the previous one. The feasibility study proposes one or more conceptual solutions to the problem set for the project. The conceptual solution gives an idea of what the new system will look like. They define what will be done on the computer and what will remain manual. It also indicates what input will be needed by the system and what outputs will be produced. These solutions should be proven feasible and a preferred solution is accepted.

The feasibility study environment enables all alternatives to be discussed and evaluated. This phase starts with an identification of the main characteristics of the required system. During this stage it is important to collect information as much as possible about the software package that might meet the specification from as many sources as possible. Normally, the central endeavour of a feasibility study is a cost benefit analysis of various alternatives. It can be defined as a systematic comparison between the cost of carrying out a service or activity and the value of that service or activity. The main benefits are qualitative than quantitative.

A feasibility study could be used to test a new working system, which could be used because:

- The current system may no longer suit its purpose,
- Technological advancement may have rendered the current system obsolete,
- The business is expanding, allowing it to cope with extra work load,
- Customers are complaining about the speed and quality of work the business provides.
- Competitors are now winning a big enough market share due to an effective integration of a computerized system.

An analysis of the ability to complete a project successfully, taking into account legal, economic, technological, scheduling, and other factors is considered as feasibility study.

Rather than just diving into a project and hoping for the best, feasibility study allows project managers to investigate the possible negative and positive outcomes of a project before investigate the possible outcomes of a project before investing too much time and money.

Stock market cannot be accurately predicted. The future, like any complex problem, has far too many variables to be predicted. The stock market is a place where buyers and sellers converge. When there are more buyers than sellers, the price increases. When there are more sellers than buyers, the price decreases. So, there is a factor which causes people to buy and sell. It has more to do with emotion than logic. Because emotion is unpredictable, stock market movements will be unpredictable. It's futile to try to predict where markets are going. They are designed to be unpredictable.

The proposed system will not always produce accurate results since it does not account for the human behaviour. Factors like change in company's leadership, internal matters, strikes, protests, natural disasters, and change in the authority cannot be taken into account for relating it to the change in Stock market by the machine. The objective of the system is to give an approximate idea of where the stock market might be headed. It does not give a long term forecasting of a stock value. There are way too many reasons to acknowledge for the long term output of a current stock. Many things and parameters may affect it on the way due to which long term forecasting is just not feasible.

When a new project is proposed, it normally goes through feasibility assessment. Feasibility study is carried out to determine whether the proposed system is possible to develop with available resources and what should be the cost consideration.

Facts considered in the feasibility analysis are:

3.1 Economic Feasibility

Economic analysis could also be referred to as cost/benefit analysis. It is the most frequently used method for evaluating the effectiveness of a new system. In economic analysis the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and

implement the system. An entrepreneur must accurately weigh the cost versus benefits before taking an action.

The concerned business must be able to see the value of the investment it is pondering before committing to an entire system study. If short-term costs are not overshadowed by long-term gains or produce no immediate reduction in operating costs, then the system is not economically feasible, and the project should not proceed any further. If the expected benefits equal or exceed costs, the system can be judged to be economically feasible. Economic analysis is used for evaluating the effectiveness of the proposed system. The economic feasibility will review the expected costs to see if they are in-line with the projected budget or if the project has an acceptable return on investment. At this point, the projected costs will only be a rough estimate.

The exact costs are not required to determine economic feasibility. It is only required to determine if it is feasible that the project costs will fall within the target budget or return on investment. A rough estimate of the project schedule is required to determine if it would be feasible to complete the systems project within a required timeframe. The required timeframe would need to be set by the organization.

The economic analysis is done to determine the benefits and savings that are expected from candidate system and compare them with costs. Thus coming to a conclusion that whether the system is economically feasible or not. This system is cost effective as well as time effective, thereby making it economically feasible. This study presents tangible and intangible benefits from the project by comparing the developments and operational costs. The technique of cost benefit analysis is often used as a basis for assessing economic feasibility. This system needs some more initial investment than the existing system, but it can be justifiable that it will improve the quality of service.

3.2 Technical Feasibility

A large part of determining resources has to do with assessing technical feasibility. It considers the technical requirements of the proposed project. The technical requirements are then compared to the technical capability of the organization. The systems project is considered technically feasible if the internal technical capability is sufficient to support the project requirements. The analyst must find out whether current technical resources can be upgraded or

added to in a manner that fulfils the request under consideration. This is where the expertise of system analysts is beneficial, since using their own experience and their contact with vendors they will be able to answer the question of technical feasibility.

Stock prediction system deals with the modern technology system that needs the well efficient technical system to run this project. All the resource constrains must be in the favour of the better influence of the system. Keeping all this facts in mind had selected the favourable hardware and software utilities to make it more feasible. The technical requirements for the system are economic and it does not use additional software. That is whether the system can be implemented using the existing technologies or not. This application is develop using python, whose development kit are easily available and free of cost, thus making the system technically feasible.

3.3 Operational Feasibility

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

Operational feasibility reviews the willingness of the organization to support the proposed system. This is probably the most difficult of the feasibilities to gauge. In order to determine this feasibility, it is important to understand the management commitment to the proposed project. If the request was initiated by management, it is likely that there is management support and the system will be accepted and used. However, it is also important that the employee base will be accepting of the change.

The operational feasibility is the one that will be used effectively after it has been developed. If users have difficulty with a new system, it will not produce the expected benefits. It measures the viability of a system in terms of the PIECES framework. The PIECES frame work can help in identifying operational problems to be solved, and their urgency:

1. Performance: Does current mode of operation provide adequate throughput and response time? As compared to traditional methods of manually retrieving the stock data from the web and forecasting the stock prices with large number of manual calculations, this system plays a very

important role in designing an application that automates the stock price prediction with the help of a user-friendly dashboard, thus making the process easier and faster.

2. Information: Does current mode provide end users and managers with timely, pertinent, accurate and usefully formatted information? System provides end users with timely, pertinent, accurate and usefully formatted information. Since all the stock related information is being pulled from Yahoo Finance against a unique NSE Stock Symbol, it will provide for meaningful and accurate data to the investor. The investing decisions are made by the traditional investors manually. This results in loss of validity of data due to human error. The information handling and the investing decision in the proposed system will be driven by computerized prediction and validation of stock data. The human errors will be minimal.

3. Economy: Does current mode of operation provide cost-effective information services to the business? Could there be a reduction in costs and/or an increase in benefits? Determines whether the system offers adequate service level and capacity to reduce the cost of the business or increase the profit of the business. The deployment of the proposed system, manual work will be reduced and will be replaced by an IT savvy approach. Moreover, it has also been shown in the economic feasibility report that the recommended solution is definitely going to benefit economically in the long run. The system is built on Excel, Python-Django and MySQL. Excel do not need any additional installation; they are in-built in every system in a company. Python needs installation but it is free software. So, overall the application is very economically feasible.

4. Control: Does current mode of operation offer effective controls to protect against fraud and to guarantee accuracy and security of data and information? As all the data is pulled from Yahoo Finance, which is a public stock data provider, it does not contain any confidential information which can be misused, so on that contrast there should be no use of any security corner for this system.

5. Efficiency: Does current mode of operation makes maximum use of available resources, including people, time, and flow of forms? Efficiency work is to ensure a proper workflow structure; can ensure the proper utilization of all the resources. It determines whether the system makes maximum use of available resources including time, people, flow of forms, minimum processing delay. In the current system a lot of time is wasted as the investing decisions are made

by the traditional investors manually. The proposed system will be a lot efficient as it will be driven by computerized prediction of stock data.

6. Services: Does current mode of operation provide reliable service? Is it flexible and expandable? The system is desirable and reliable services to those who need it and also whether the system is flexible and expandable or not. The proposed system is very much flexible for better efficiency and performance of the organization. The scalability of the proposed system will be inexhaustible as the storage capacity of the system can be increased as per requirement. This will provide a strong base for expansion. The new system will provide a high level of flexibility.

3.4 Behavioural Feasibility

This analysis involves how it will work when it is installed and the assessment of political and managerial environment in which it is implemented. People are inherently resistant to change and computers have been known to facilitate change. The new proposed system is very much useful to the users and there for it will accept broad audience.

3.5 Software Feasibility

Even though this application is developed in very high software environment, it is also supported by many other environments with minimum changes. The system is fully feasible to be executed on any kind of modern operating systems and browsers.

3.6 Hardware Feasibility

Software can be developed with the existing resources. But the existing resources may or may not be used to produce hardware. If no hardware is newly bought for project, then software is said to achieve hardware feasibility. When this system is used by an individual he might need new hardware resources, but not in any company or organization for which they already have. The system is hardware wise feasible because it needed absolutely no new hardware in case of a company or organization.

SOFTWARE ENGINEERING PARADIGM

4. SOFTWARE ENGINEERING PARADIGM

The software engineering paradigm which is also referred to as a software process model or Software Development Life Cycle (SDLC) model is the development strategy that encompasses the process, methods and tools. SDLC describes the period of time that starts with the software system being conceptualized.

4.1 Agile model

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like:

- Planning
- Requirement analysis
- Design
- Coding
- Unit testing
- Acceptance testing

At the end of the iteration, a working product is displayed to the customer and important stakeholders. Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer. The Agile thought process had started early in the software development and started becoming popular with time due to its flexibility and adaptability. The following figure describes the graphical illustration of the agile model.

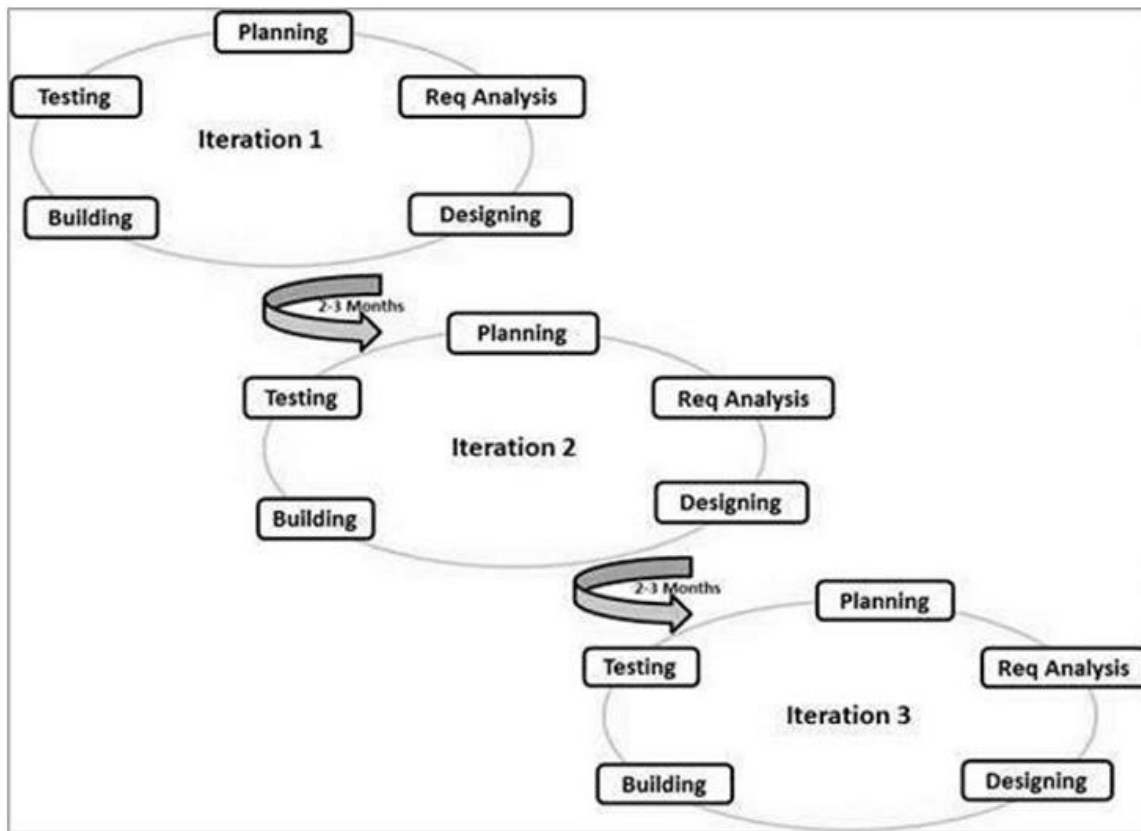


Figure:4.1 Graphical Illustration of the Agile Model

The most popular agile methods include Rational Unified Process (1994), Scrum (1995), Crystal Clear, Extreme Programming (1996), Adaptive Software Development, Feature Driven Development, and Dynamic Systems Development Method (DSDM) (1995). These are now collectively referred to as Agile Methodologies, after the Agile Manifesto was published in 2001.

Following are the Agile Manifesto principles

- **Individuals and interactions** – In Agile development, self-organization and motivation are important, as are interactions like co-location and pair programming
- **Working software** – Demo working software is considered the best means of communication with the customers to understand their requirements, instead of just depending on documentation.

- **Customer collaboration** – As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.
- **Responding to change** – Agile Development is focused on quick responses to change and continuous development

Agile Vs Traditional SDLC Models

Agile is based on the adaptive software development methods, whereas the traditional SDLC models like the waterfall model is based on a predictive approach. Predictive teams in the traditional SDLC models usually work with detailed planning and have a complete forecast of the exact tasks and features to be delivered in the next few months or during the product life cycle.

Predictive methods entirely depend on the requirement analysis and planning done in the beginning of cycle. Any changes to be incorporated go through a strict change control management and prioritization.

Agile uses an adaptive approach where there is no detailed planning and there is clarity on future tasks only in respect of what features need to be developed. There is feature driven development and the team adapts to the changing product requirements dynamically. The product is tested very frequently, through the release iterations, minimizing the risk of any major failures in future.

Customer Interaction is the backbone of this agile methodology, and open communication with minimum documentation are the typical features of agile development environment. The agile teams work in close collaboration with each other and are most often located in the same geographical location.

The advantages of the Agile Model are as follows:

- It is a very realistic approach to software development.
- Promotes teamwork and cross training.
- Functionality can be developed rapidly and demonstrated.
- Resource requirements are minimum.

- Suitable for fixed or changing requirements
- Delivers early partial working solutions.
- Good model for environments that change steadily.
- Minimal rules, documentation easily employed.
- Enables concurrent development and delivery within an overall planned context.
- Little or no planning required.
- Easy to manage.
- Gives flexibility to developers

4.2 Scrum

Scrum is an agile framework for managing knowledge work, with an emphasis on software development. It is designed for teams of three to nine members, who break their work into actions that can be completed within time boxed iterations, called "sprints", no longer than one month and most commonly two weeks, then track progress and re-plan in 15-minute stand-up meetings, called daily scrums. The roles in Scrum are quite different from the traditional software methods. Clearly defined roles and expectations help individuals perform their tasks efficiently. In Scrum, there are three roles: Product Owner, Development Team, and Scrum Master. Together these are known as the Scrum Team. Each add their respective contributions to the project.

The Scrum Master helps the product group learn and apply Scrum to achieve business value. The Scrum Master does whatever is in their power to help the Team, Product Owner and organization be successful. The Scrum Master is not the manager of the Team members, nor are they a project manager, team lead, or team representative. Instead, the Scrum Master serves the Team; he or she helps to remove impediments, protects the Team from outside interference, and helps the Team to adopt agile development practices. He or she educates, coaches and guides the Product Owner, Team and the rest of the organization in the skillful use of Scrum.

The Product Owner is responsible for maximizing return on investment (ROI) by identifying product features, translating these into a prioritized list, deciding which should be at the top of the list for the next Sprint, and continually re-prioritizing and refining the list. The Product Owner has profit and loss responsibility for the product, assuming it is a commercial

product. Product Owner in Agile is like a spokesperson for customer and needs to represent them.

A Development Team is a collection of individuals working together to develop and deliver the requested and committed product increments. It comprises of cross-functional members who are capable of achieving the sprint goals. This could include software engineers, architects, programmers, analysts, system admins, QA experts, testers, UI designers, etc.

Scrum is an iterative and incremental framework for managing product development. It defines "a flexible, holistic product development strategy where a development team works as a unit to reach a common goal", challenges assumptions of the "traditional, sequential approach to product development, and enables teams to self-organize by encouraging physical co-location or close online collaboration of all team members, as well as daily face-to-face communication among all team members and disciplines involved.

A key principle of Scrum is the dual recognition that customers will change their minds about what they want or need and that there will be unpredictable challenges-for which a predictive or planned approach is not suited. As such, Scrum adopts an evidence-based empirical approach accepting that the problem cannot be fully understood or defined up front, and instead focusing on how to maximize the team's ability to deliver quickly, to respond to emerging requirements, and to adapt to evolving technologies and changes in market conditions.

Many of the terms used in Scrum (e.g., scrum master) are typically written with leading capitals (e.g. Scrum Master) or as conjoint words written in camel case (e.g., Scrum Master). To maintain an encyclopaedic tone, however, this article uses normal sentence case for these terms-unless they are recognized marks. This is occasionally seen written in all -capitals, as SCRUM in its title SCRUM. The word is not an acronym, so this is not correct; however, it likely arose due to an early paper by Ken Schwaber which capitalized.

While the trademark on the term Scrum itself has been allowed to lapse, so that it is deemed as owned by the wider community rather than an individual, the leading capital is retained-except when used with other words.

SYSTEM REQUIREMENT SPECIFICATION

5. SYSTEM REQUIREMENTS SPECIFICATION

System Specification

Hardware and software requirements for the installation and smooth functioning of this product could be configured based on the requirements needed by the component of the operating environment that works as front-end system here, suggest minimum configuration for the both hardware and software components. Working off with this software is requirements concrete on system environments. It includes two phases.

- Software Requirements
- Hardware Requirements

5.1 Software Requirements

This system is built on PyCharm IDE using Python language and Django as framework. This system actually does not need a database, even though a MySQL database is used just to add login credentials and user registration. The dataset required for this system is downloaded manually from Yahoo Finance website. And the data model after training is saved in system itself as HDF5 (Hierarchical Data Format) file commonly known as H5 file.

- **Software requirements for Developer:**
 - Operating System : Windows 8 or above
 - Technology used : Python, MS Excel
 - IDE : PyCharm, Nteract
 - Cloud Computational Environment : Kaggle notebook, Google Colab notebook
 - Dataset and Model : CSV file, H5 file
 - Framework : Django
 - Database : MySQL
 - Browser : Google Chrome

- **Software requirements for Clients:**

- Operating System : Windows 8 or above
- Technology needed : Python, MS Excel
- Database : MySQL
- Browser : Google Chrome or Equivalent

5.2. Hardware Requirements

The hardware requirements explain the hardware components the need to develop the project. In this system hardware requirement for the developer are minimum of about 8GB RAM, i5 processor, and 15GB Graphical Processing Unit(GPU) which was provided by the Kaggle kernel and Google Colab because it does not take much time in training model. Using a GPU with adequate memory makes training a deep learning network many times faster than using a CPU alone. Because it's much nicer to get feedback in minutes or hours instead of days and weeks. It is selected such a way that it reduces the work and easy to implement the project. Depending on the client requirement hardware components can be increased so as to get timely predictions.

- **Hardware requirements for Developers:**

- Processor : i5 or above
- Ram : 8 GB or above
- GPU : 15 GB Nvidia Tesla K80 Cuda core – 4992(Kaggle kernal)
- Hard Disk : 1 TB

- **Hardware requirements for Clients:**

- Processor : i5 or above
- Ram : 8 GB or above
- GPU : 8 GB Nvidia GeForce GTX 1050 Ti Cuda core -768 or above
- Hard Disk : 1 TB

SYSTEM DESIGN

6. SYSTEM DESIGN

System design is the first in the development phase for many engineered product or system. It may define the process of applying various techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

This phase is the first step in moving from the problem domain to the solution domain. It is an iterative process through which requirements are transmitted into “blue print” for constructing the software initially. Blue print depicts holistic new software. That is, design is represented at high level of abstraction, functional and behavioural requirements. System design is creative art of inverting and developing inputs, database and offline files, communication between the software clients and server, methods and procedure for processing data to get meaningful output that satisfy the organization objectives. From the project management view, software design is conducted in two steps. Preliminary design is concerned with the transformation of the requirements into data and architecture. Detailed design focuses on refinement to the architectural representation for software. Some properties for the system design are:

- Verifiability
- Completeness
- Efficiency
- Traceability

6.1 Input Design

The input design is the process of converting the user oriented into the computer based formats. So the design of the user interface is very important for any application. a good interface design lead to an interface software application. Input design consist of the development of specification, the necessary step to convert the given data into useable form for processing, data entry and activity of putting the data into computer for processing. All input processes have been designed with utmost care to avoid the entry of the invalid data into the system.

The input screens have been validated effectively in order to give the most, accurate input details. Some aspects have to be followed while designing the input screen.

- The data to input
- The details of how data should be arranged or coded
- Allow the information only the user needs
- Same styles among screens have to be kept
- Confirmation of critical details to be asked
- Validate data as soon as possible on inputs

The input design requirements such as user friendliness, consistent format and interactive dialogue for giving the right message and helps for the users at the right time are also necessary for the development of good software.

6.2 Output design

Output design generally refers to the results and information that are generated by the system. The results are of in interactive mode. A common user can also use the application. Each request is having an acknowledgment for ensuring correct operation get performed.

Computer output is most and direct source of information to the user .In output design the emphasis is given to the design of the hard copy and a soft copy of the information needed to the user. Efficient, intelligent and friendly screens improve the relationship between clients and system for design making. System analysis decides the following input design details:

- Who will receive the output?
- What is planned use?
- How much details are needed?
- When and how often is the output?
- By what method?

Output from the system as information and results are referred in this phase. Here the determination is taken about the type and format of the output. It is needed to remember that the user evaluate the usefulness of the system only through the output it generates. The distribution specifications of the output to indented receipts are determined here. The user friendliness of the screen with necessity controls and layouts fits for these purposes.

Output Types:

- External Outputs, whose destination is outside the organization and is the main image of the organization.
- Internal Outputs, whose destination is within the organization and which require careful design because it is user's main interface with the computer.
- Operational Outputs, whose use is purely within the computer departments.
- Interactive outputs, which involve the user in communicating directly with the computer.

6.3 Database Design

Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data.

In the relational model these are the tables and views. In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database management system. The process of doing database design generally consists of a number of steps which will be carried out by the database designer. Usually, the designer must: Determine the relationships between the different data elements and superimpose a logical structure upon the data on the basis of these relationships.

Normalization

It is a process of converting a relation to a standard form. The process is used to handle the problems that can arise due to data redundancy i.e. repetition of data in the database, maintain data integrity as well as handling problems that can arise due to insertion, updation, deletion anomalies.

Decomposing is the process of splitting relations into multiple relations to eliminate anomalies and maintain data integrity. To do this use normal forms or rules for structuring relation.

- Insertion anomaly: Inability to add data to the database due to absence of other data.
- Deletion anomaly: Unintended loss of data due to deletion of other data.
- Update anomaly: Data inconsistency resulting from data redundancy and partial update
- Normal Forms: These are the rules for structuring relations that eliminate anomalies.

1. First Normal Form (1NF):

A relation is said to be in first normal form if the values in the relation are atomic for every attribute in the relation. By this it mean simply that no attribute value can be a set of values or, as it is sometimes expressed, a repeating group.

2. Second Normal Form (2NF):

A relation is said to be in second Normal form is it is in first normal form and it should satisfy any one of the following rules.

- Primary key is a not a composite primary key
- No non key attributes are present
- Every non key attribute is fully functionally dependent on full set of primary key.

Table

Table name: login

COLUMN NAME	DATA TYPE	DESCRIPTION
id	int	Primary key
emp_id	int	Foreign key
username	varchar(20)	Username
password	varchar(20)	Password
role	varchar(20)	User role

Table name: employee

COLUMN NAME	DATA TYPE	DESCRIPTION
id	int	Primary key
name	varchar(20)	Name of employee
desi	varchar(30)	Designation of employee
dob	varchar(30)	Date of Birth of employee
email	varchar(30)	Email of employee

6.4 UML Designs

The Unified Modelling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the artefacts of the software systems, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing objects-oriented software and software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

Goals of UML

The primary goals in the design of the UML are:

- Provide users with a ready-to-use, expressive visual modelling language so they can develop and exchange meaningful models.
- Provide extensibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development processes.
- Provide a formal basis for understanding the modelling language.
- Encourage the growth of the OO tools market.
- Support higher-level development concepts such as collaborations frameworks, patterns and components.
- Integrate best practice.

As the strategic value of the software increase for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time to market. These techniques include component technology. Visual programming, patterns and frameworks. Businesses also seek techniques to manage the complexity of systems as they increase in the scale. In particular, they recognize the need to solve recurring problems such as physical distribution, concurrency, replication, security, load balancing and fault tolerance. Additionally, the development for the World Wide Web, while making something simpler, has extracted these architectural problems

Use Case diagram, a use case describes how a user uses a system to accomplish a particular goal. A use case diagram consists of the system, the related use cases and actors and relates these to each other to visualize. Use cases help ensure that the correct system is developed by capturing the requirements from the user's point of view.

Class Diagram, models class structure and contents using design elements such as classes, packages and objects. It also displays relationships such as containment, inheritance, associations and others.

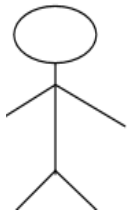
Interaction Diagrams

- **Sequence Diagram** displays the time sequence of the objects participating in the interaction. This consists of the vertical dimension (time) and horizontal dimension (different objects).
- **Collaboration Diagram** displays an interaction organized around the objects and their links to one another. Numbers are used to show the sequence of messages.
- **State Diagram** displays the sequences of states that an object of an interaction goes through during its life in response to received stimuli, together with its responses and actions.
- **Activity Diagram** displays a special state diagram where most of the states are action states and most of the transitions are triggered by completion of the actions in the source states. This diagram focuses on flows driven by internal processing.

Physical Diagrams

- **Component Diagram** displays the high-level packaged structure of the code itself. Dependencies among components are shown, including source code components, binary code components, and executable components. Some components exist at compile time, at link time, at run times well as at more than one time.
- **Deployment Diagram** displays the configuration of run-time processing elements and the software components, processes, and objects that live on them. Software component. Instances represent run-time manifestations of code units.

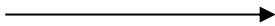
Sequence Diagram is an interaction diagram that shows how the processes operate with one another and what order. It is a construct of a message sequence chart. Sequence diagram shows message interactions arranged in same sequence. It depicts the object and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realization in logical view of the system under development. Sequence diagram re sometimes called event diagrams or event scenarios.



Actors

An object: A class

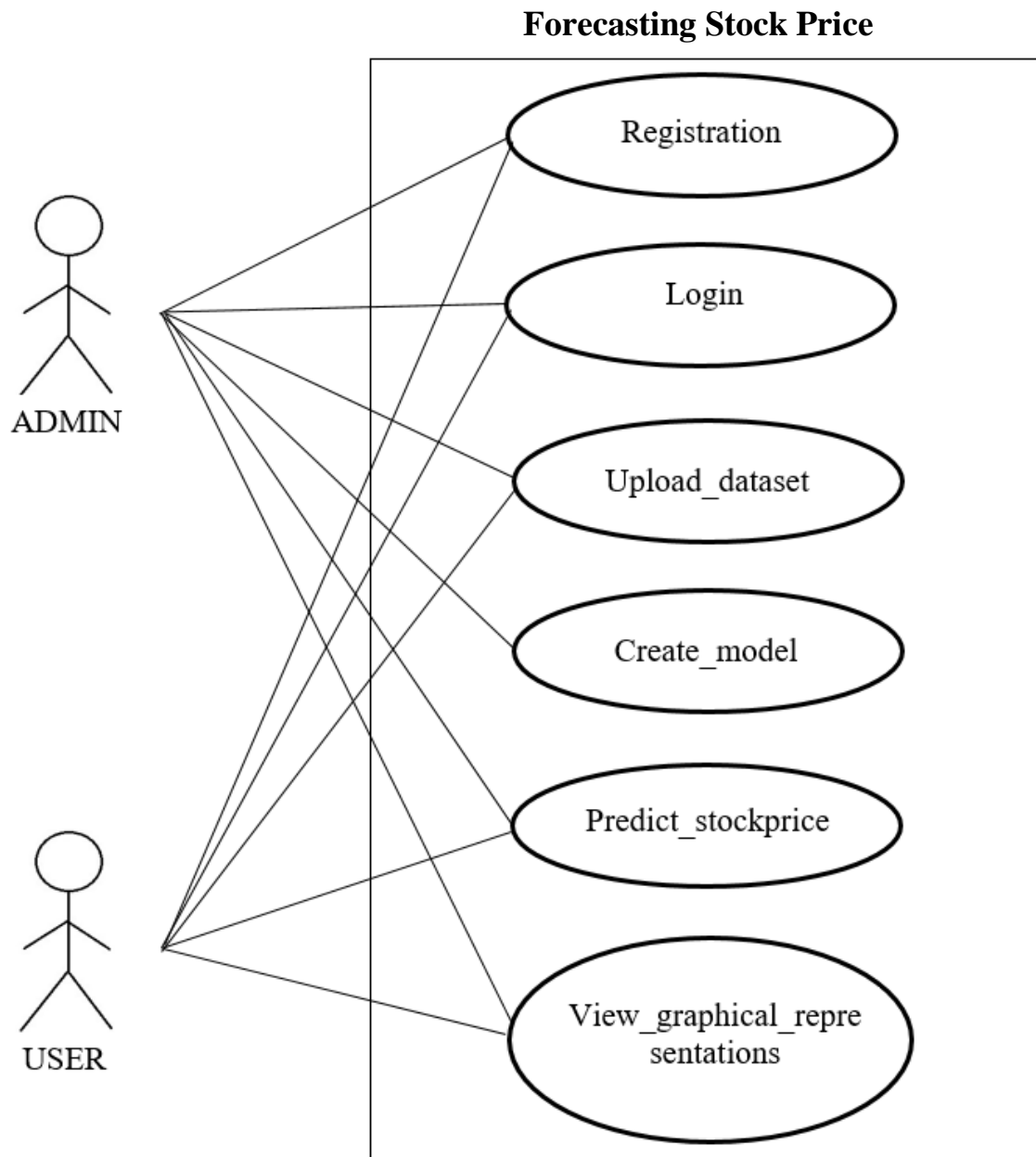
Have different processes or objects that live simultaneously



Horizontal arrows that show the message exchanged between them, in order in which they occur



Focus of control is an execution occurrence shown as rectangle on a lifeline that represents the period during which an element is performing an operation. The top and the bottom of the rectangle is aligned with the initiation and the completion time respectively

Use case Diagram

Scenario

Admin:

In admin module data training is done. For data training dataset are taken from company server or from Yahoo finance website, which are downloaded in CSV format. The dataset are then cleaned and pre-processed by admin manually using MS Excel. Admin need to split the dataset into training set and testing set. After which admin login to the system using username and password provided and from the system, browse for the dataset and upload both the training and testing dataset to the system. By clicking on the training button the training process will run on the background and a model is created saved in the system. Now admin can predict the closing stock price of adjacent day using trained model. A graphical representation and predicted value is displayed on the system which can be save for future reference. Admin roles are:

- Analysing and Cleaning dataset
- Train this dataset
- Create model
- Predict stock price
- View graphical representations

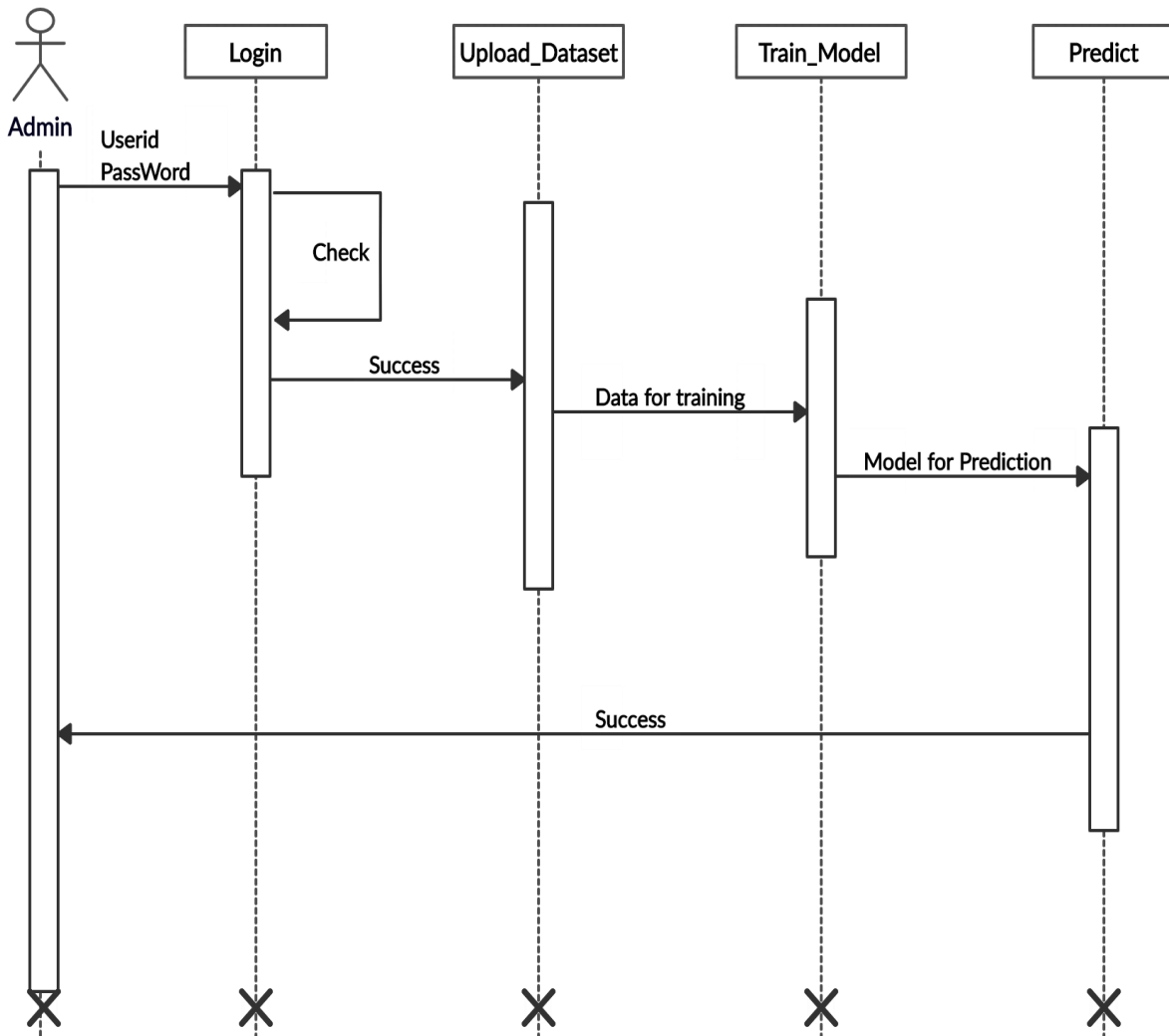
User:

In this users can login into the system by username and password. Can upload a dataset which is formerly downloaded from Yahoo Finance. Can predict closing stock price of adjacent day of the company by the model already trained by the admin. Can view graphical representation and predicted value of stock price displayed on the system. User roles are:

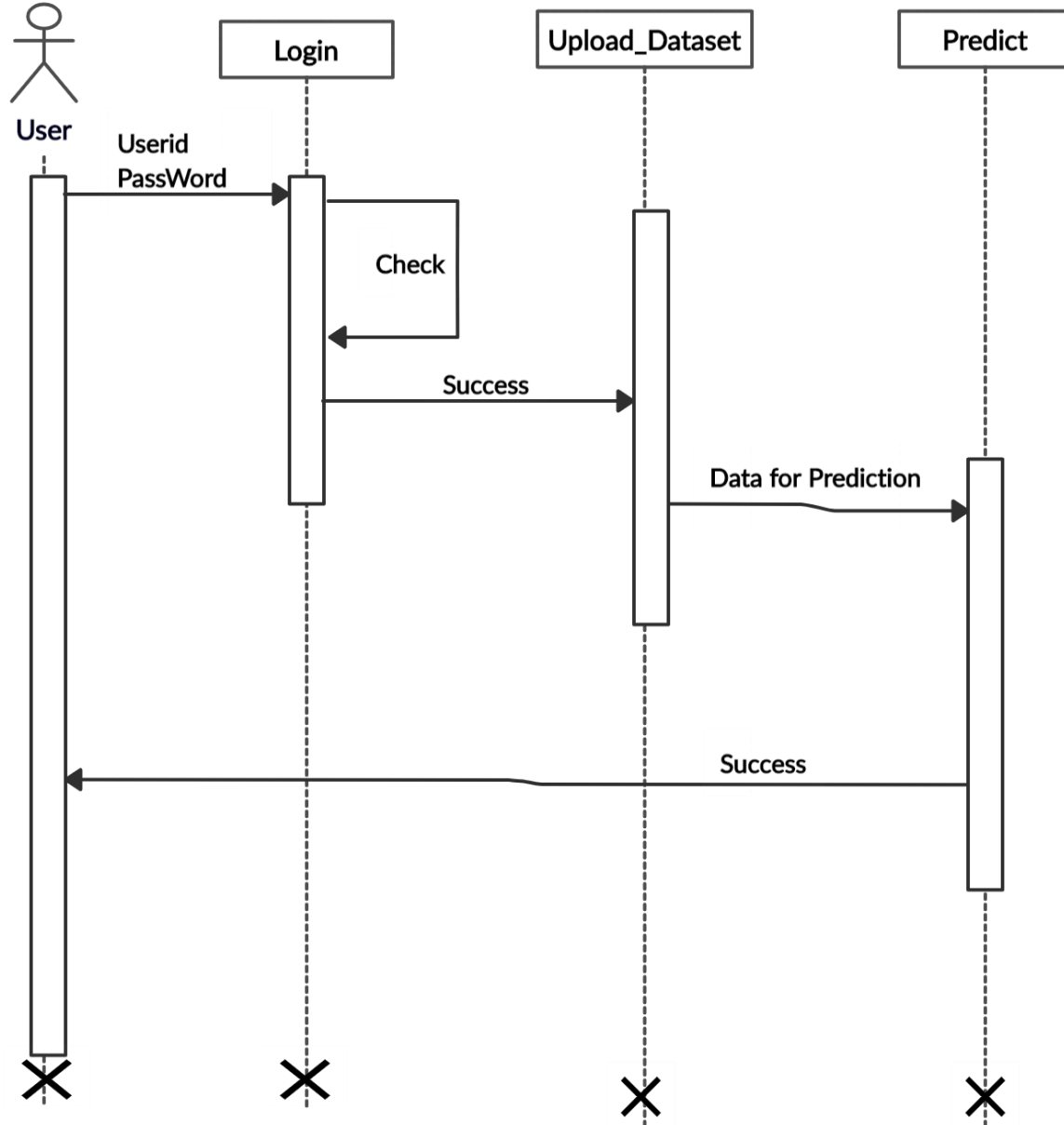
- Predict stock price
- View graphical representations

Sequence Diagram

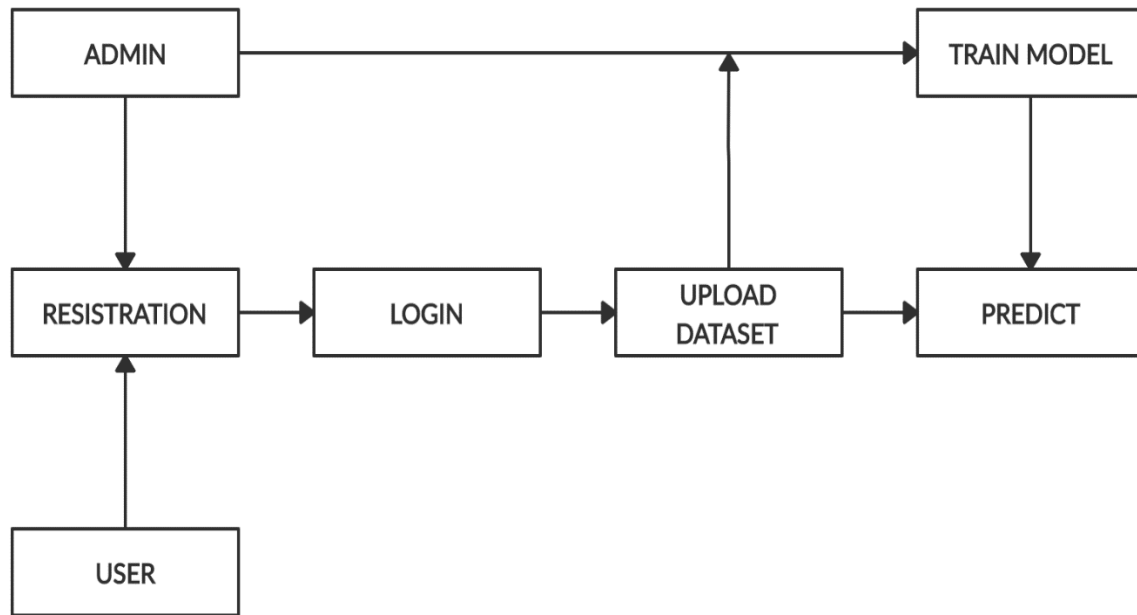
Admin



User



Class Diagram



SYSTEM DEVELOPMENT

7. SYSTEM DEVELOPMENT

System development is series of operations to manipulate data to produce output from computer system. The principle activities performed during the development phase can be divided into two major related sequences.

- External system development
- Internal system development

The major external system activities are:

- Implementation
- Planning
- Equipment acquisition
- Installation

7.1 Coding

The purpose of code is to facilitate the identification and retrieval of items of information. A code is an ordered collection of symbols designed to provide unique identification of entity or an attribute. Code also show interrelationship among different items. Codes are used to identify, access, sort, matching records. The code ensures that only one value of code with a single meaning is applied to give entity or attribute as described in various ways.

Python

Python is a widely used high-level programming language for general purpose programming, created by Guido van Rossum and first released in 1991. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java.

The language provides constructs intended to enable writing clear programs on both a small and large scale. Python features a dynamic type system and automatic memory management and supports multiple programming paradigms, including object- oriented,

imperative, functional programming, and procedural styles. It has a large and comprehensive standard library. Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems. Python is open source software and has a community-based development model, as do nearly all of its variant implementations.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Python is a multi-paradigm programming language. Object- oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by meta programming and meta objects (magic methods). Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing, and a combination of reference counting and a cycle detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Python's design offers some support for functional programming in the Lisp tradition. It has `filter()`, `map()`, and `reduce()` functions; list comprehensions, dictionaries, and sets; and generator expressions. The standard library has two modules (`itertools` and `functools`) that implement functional tools borrowed from Haskell and Standard ML.

The language's core philosophy is summarized in the document *The Zen of Python* (PEP 20), which includes aphorisms such as:

- Beautiful is better than ugly
- Explicit is better than implicit
- Simple is better than complex
- Complex is better than complicated
- Readability counts

Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.

While offering choice in coding methodology, the Python philosophy rejects exuberant syntax (such as that of Perl) in favour of a simpler, less-cluttered grammar. As Alex Martelli put it: "To describe something as 'clever' is not considered a compliment in the Python culture Python's philosophy rejects the Perl "there is more than one way to do it" approach to language design in favour of "there should be one and preferably only one obvious way to do it.

Python's developers strive to avoid premature optimization, and reject patches to non-critical parts of CPython that would offer marginal increases in speed at the cost of clarity. When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use PyPy, a just-in-time compiler. Cython is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter.

An important goal of Python's developers is keeping it fun to use. This is reflected in the language's name—a tribute to the British comedy group Monty Python—and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (from a famous Monty Python sketch) instead of the standard foo and bar.

A common neologism in the Python community is *pythonic*, which can have a wide range of meanings related to program style. To say that code is *pythonic* is to say that it uses Python idioms well, that it is natural or shows fluency in the language, that it conforms with Python's

minimalist philosophy and emphasis on readability. In contrast, code that is difficult to understand or reads like a rough transcription from another programming language is called unpythonic. Users and admirers of Python, especially those considered knowledgeable or experienced, are often referred to as Pythonists, Pythonistas, and Pythoneers.

Syntax and Semantics

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are optional. It has fewer syntactic exceptions and special cases than C or Pascal.

Indentation

Python uses whitespace indentation, rather than curly brackets or keywords, to delimit blocks. An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block. This feature is also sometimes termed the off-side rule.

Libraries

Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited too many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary precision decimals, manipulating regular expressions, and unit testing.

Some parts of the standard library are covered by specifications (for example, the Web Server Gateway Interface (WSGI) implementation `wsgiref` follows PEP 333), but most modules are not. They are specified by their code, internal documentation, and test suites (if supplied). However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

As of March 2020, the Python Package Index, the official repository for third party Python software, contains over 235,000 packages with a wide range of functionality, including

- Graphical user interfaces

- Web frameworks
- Multimedia
- Databases
- Networking
- Test frameworks
- Automation
- Web scraping
- Documentation
- System administration
- Scientific computing
- Text processing
- Image processing

In this system many python libraries and packages are used. The libraries used are:

Pandas :

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

Numpy :

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Sklearn :

Scikit-learn (also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Preprocessing package in sklearn is used in this system.

Keras :

Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. Packages like models and layers are used and also module in Sequential, Dense, LSTM, Dropout are used.

Matplotlib :

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. Pyplot package is used in this system.

Tensorflow :

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks.

Jupyter Notebook

Jupyter Notebook (formerly IPython Notebooks) is a web based interactive computational environment for creating Jupyter notebook documents. The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context. A Jupyter Notebook document is a JSON document, following a versioned schema, and containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media, usually ending with the ".ipynb" extension. Nteract is used to manage jupyter notebook in developing this system.

A Jupyter Notebook can be converted to a number of open standard output formats (HTML, presentation slides, LaTeX, PDF, Restructured text, Markdown, Python) through "Download As" in the web interface, via the nbconvert library or "jupyter nbconvert" command line interface in a shell. To simplify visualisation of Jupyter notebook documents on the web, the nbconvert library is provided as a service through Nb Viewer which can take a URL to any publicly available notebook document, convert it to HTML on the fly and display it to the user.

Kaggle Notebook and Google Colab

Google Colab or “the Colaboratory” is a free cloud service hosted by Google to encourage Machine Learning and Artificial Intelligence research, where often the barrier to learning and success is the requirement of tremendous computational power. Besides being easy to use, the Colab is fairly flexible in its configuration and does much of the heavy lifting for you. Kaggle Kernels is also a free platform to run Jupyter notebooks in the browser. Both Kaggle and Colab supports python, has free GPU acceleration, and has Pre-installed libraries: All major python libraries like TensorFlow, Scikit-learn, Matplotlib among many others are pre-installed and ready to be imported, built on top of Jupyter notebook. Google Colab allows developers to use and share Jupyter notebook among each other without having to download, install, or run anything other than a browser. Supports bash commands. Google Colab notebooks are stored on the drive. All major process of this project work is done on Kaggle.

Django

Django is a Python-based free and open-source web framework that follows the model-template-view (MVT) architectural pattern. It is maintained by the Django Software Foundation (DSF), an American independent organization established as a non-profit.

Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

Git

Git is a type of version control system (VCS) that makes it easier to track changes to files. For example, when you edit a file, git can help you determine exactly what changed, who changed it, and why. It's useful for coordinating work among multiple people on a project, and for tracking progress over time by saving “checkpoints”. You could use it while writing an essay, or to track changes to artwork and design files. Git isn't the only version control system out there,

but it's by far the most popular. Many software developers use git daily, and understanding how to use it can give a major boost to your resume. In a classical source code repository, folder with files and subfolders are handled as content. The repository holds all versions of the content, while the working directory is the place where you modify the code. You checkout code from the repository to the working directory and commit changes you have made in this working directory back into a new version of the content in the repository.

A git repository contains, among other things, the following

- In complex projects, where multiple people might be making changes to the same files simultaneously, it's easy to get into weird state. The purpose of Git is to manage a project, or a set of files, as they change over time. Git stores this information in a data structure called a repository. Call *git commit* to create the commit object. The new commit object will have the current HEAD as its parent (and then, after the commit is complete, HEAD will point to the new commit objects).

As a shortcut, *git commit -a* will automatically add all modified files (but not new ones). Note that if you modify a file but do not add it, then Git will include the previous version (before modifications) to the commit. The modified files will remain in place.

MYSQL Database

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications-which may run either on the same computer or on another computer across a network (including the Internet).

Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users. Structured Query Language is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS).

Originally based upon relational algebra and tuple relational calculus, SQL consist of a data definition language, data manipulation language, and data control language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control. Although SQL is often described as, and to a great extent is, a declarative language (4GL), it also includes procedural elements.

SQL was one of the first commercial languages for Edgar F. Codd's relational model, as described in his influential 1970 paper, "A Relational Model of Data for Large Shared Data Banks". Despite not entirely adhering to the relational model as described by Codd, it became the most widely used database language. SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987. Since then, the standard has been revised to include a larger set of features. Despite the existence of such standards, most SQL code is not completely portable among different database systems without adjustments. Even though there is no use of a database in this system, added a database only for login credentials and registration.

7.2 Algorithm

LSTM :

Long Short-Term Memory (LSTM) Long Short-Term Memory (LSTM) is one of many types of Recurrent Neural Network RNN, it's also capable of catching data from past stages and use it for future predictions. In general, an Artificial Neural Network (ANN) consists of three layers:

- 1) Input layer.
- 2) Hidden layers.
- 3) Output layer.

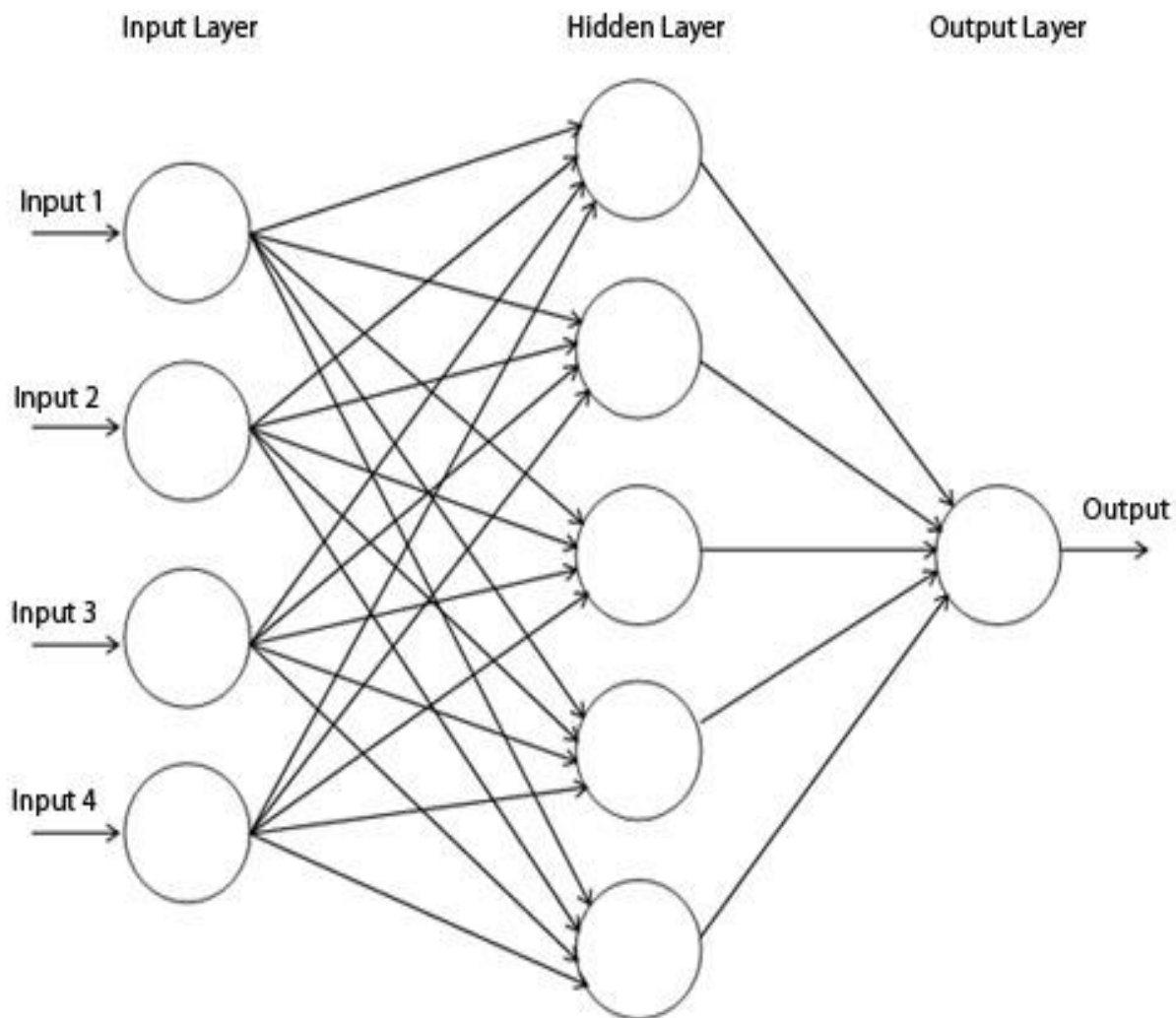


Figure 7.1 Layers of Artificial Neural Network.

Input Layer : Made up of units; the qualities measured for each drill tuple matches to the input to the network. Inputs are served to this layer instantaneously. The input passes through input layer and weighted & instantaneously served to the next layer i.e. hidden layer.

Hidden Layer: The productions of the input layer are input to this hidden layer. The number of hidden layer is random; in rehearsal only one hidden layer is used. The weighted output of the

hidden layer are input to the next or output layer, which actually releases the network forecast for given tuples.

Output Layer: This layer actually discharges the network forecast for given tuples. Multilayer feed-forward network are able to model the class forecast as a nonlinear grouping of the input. For given hidden units and enough preparation samples can carefully estimate to any function.

In a NN that only contains one hidden layer the number of nodes in the input layer always depend on the dimension of the data, the nodes of the input layer connect to the hidden layer via links called ‘synapses’. The relation between every two nodes from (input to the hidden layer), has a coefficient called weight, which is the decision maker for signals. The process of learning is naturally continues adjustment of weights, after completing the process of learning, the Artificial NN will have optimal weights for each synapses.

The hidden layer nodes apply a sigmoid or tangent hyperbolic (tanh) function on the sum of weights coming from the input layer which is called the activation function, this transformation will generate values, with a minimized error rate between the train and test data using the SoftMax function.

The values obtained after this transformation constitute the output layer of the NN, these value may not be the best output, in this case a back propagation process will be applied to target the optimal value of error, the back propagation process connect the output layer to the hidden layer, sending a signal conforming the best weight with the optimal error for the number of epochs decided. This process will be repeated trying to improve predictions and minimize the prediction error.

After completing this process, the model will be trained. The classes of NN that predict future value base on passed sequence of observations is called Recurrent Neural Network(RNN) this type of NN make use of earlier stages to learn of data and forecast futures trends.

The earlier stages of data should be remembered to predict and guess future values, in this case the hidden layer act like a stock for the past information from the sequential data. The

term recurrent is used to describe the process of using elements of earlier sequences to forecast future data.

RNN can't store long time memory, so the use of the Long Short-Term Memory (LSTM) based on “memory line” proved to be very useful in forecasting cases with long time data. In a LSTM the memorization of earlier stages can be performed through gates with along memory line incorporated. The following figure 7.2 describe the composition of LSTM nodes.

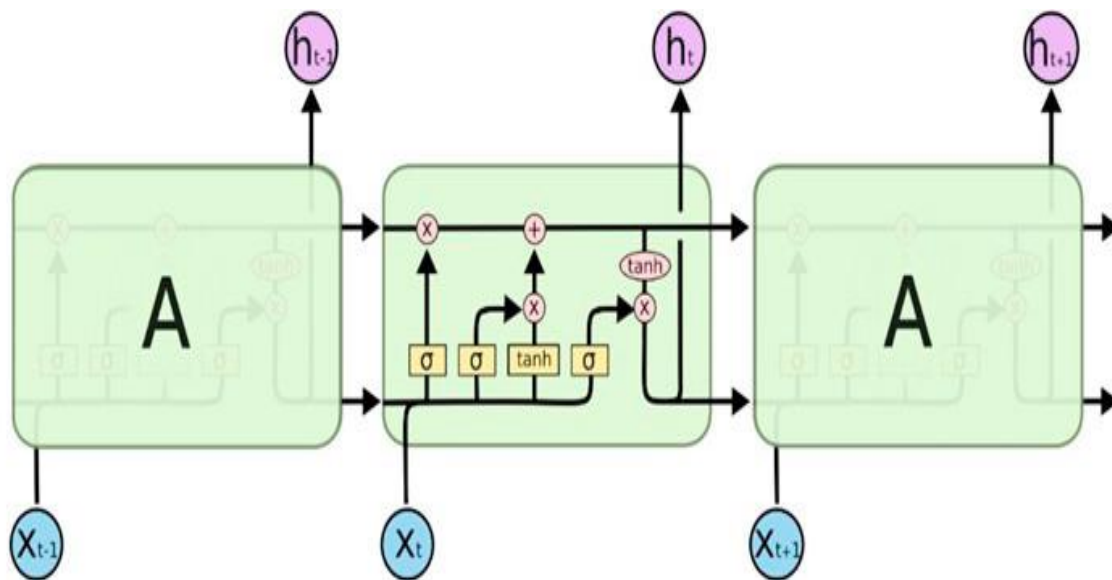


Figure 7.2 The internal structure of an LSTM.

The ability of memorizing sequence of data makes the LSTM a special kind of RNNs. Every LSTM node must be consisting of a set of cells responsible of storing passed data streams, the upper line in each cell links the models as transport line handing over data from the past to the present ones, the independency of cells helps the model dispose filter of add values of a cell to another. In the end the sigmoidal neural network layer composing the gates drive the cell to an optimal value by disposing or letting data pass through. Each sigmoid layer has a binary value (0 or 1) with 0 “let nothing pass through”; and 1 “let everything pass through.” The goal here is to control the state of each cell, the gates are controlled as follow:

Forget Gate outputs a number between 0 and 1, where 1 illustrates “completely keep this”; whereas, 0 indicates “completely ignore this.”

Memory Gate chooses which new data will be stored in the cell. First, a sigmoid layer “input door layer” chooses which values will be changed. Next, a tanh layer makes a vector of new candidate values that could be added to the state.

Output Gate decides what will be the output of each cell. The output value will be based on the cell state along with the filtered and freshest added data

Investment in stock market carries a higher risk due to its uncertainty and volatility and hence forecasting the stock price behaviour is very difficult. The difficulty arises due to the nonlinear and complex behaviour of stock prices. As the primary application of artificial neural networks is in areas where problems are ill-defined, data is incomplete or noisy in nature and the environment itself is dynamic. As artificial neural networks are able to adapt to noisy data and establish input-output relationship of nonlinear data, the behaviour of stock price prediction is possible. In the last two decades extensive researches have been attempted through neural networks to forecast stock prices.

SYSTEM TESTING AND IMPLEMENTATION

8. SYSTEM TESTING AND IMPLEMENTATION

Testing is the vital to the success of the system. It makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved in this project. It is the stage of implementation, which ensures that system works accurately and effectively before the live operation commences. It is a confirmation that all are correct and opportunity to show users that the system must be tested and show that the system will operate successfully and produce expected results under expected conditions.

Software testing is a crucial element of software quality assurance and represents the unlimited review of specification, design and coding. Testing represents an interesting anomaly for the software. During the earlier definition and development phase, it was attempted to build the software from an abstract concept to implement.

Testing is a set of activity that can be planned in advance and conducted. Systematically, this is aimed at ensuring that the system works accurately and efficiently before live operations commences.

Software testing determines the correctness, completeness, and quality of software being developed. Validation refers to the process of checking that the developed software meets the requirements specified by the user. The activities involved in the testing phase basically evaluate the capability of that system meets its requirements.

The main objective of software testing is to detect errors in the software. Errors occur if some part of the developed system is found to be incorrect, incomplete or inconsistent. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs (errors or other defects). It involves the execution of a software component or system to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

- Meets the requirements that guided its design and development,
- Responds correctly to all kinds of inputs,
- Performs its functions within an acceptable time,

- Is sufficiently usable,
- Can be installed and run in its intended environments, and
- Achieves the general result its stakeholder's desire.

8.1 Types of Testing:

Different types of testing are,

- Unit testing
- Integration testing
- System testing
- User acceptance testing.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases. All modules were tested and individually as soon as they were completed and were checked for their correct functionality.

Unit testing deals with testing a unit as a whole. This would test the interaction of many functions but confine the test within one unit. This testing is carried out during programming stage itself. In this testing step each Module is found to be working satisfactorily as regard to the expected output from the module.

Integration Testing

Integration testing is a systematic technique for testing to overcome the errors associated within the interface. In this system all the modules such as login, registration, admin in web, user in web, and the final hardware and software are combined and then the entire program is tested as a whole. Thus in the integration testing step all the errors in the implementation of the system are corrected. Data can be lost across an interface one module can have an adverse effect on others; sub-functions when combined may not produce the desired major functions integration

testing is a systematic testing for constructing the program structure. The objective is to take unit tested modules and to combine them and test it as a whole.

- The system contains various components and they have to be combined and tested.
- The software is embedded into hardware and tested.
- The modules are combined and tested.
- The hardware part tested separately.

System Testing:

After performing the validation testing, the next step is output testing of the proposed system since no system could be useful if it doesn't produce the required data in the specific format. The output displayed or generated by the system under consideration is tested by, asking the user about the format displayed. The output format on the screen is found to be correct as the format was designed in the system phase according to the user needs. Hence the output testing doesn't result in any correction in the system.

User Acceptance Testing:

User Acceptance of the system is the key factor for the success of the system. The system under consideration is tested for user Acceptance by constantly keeping in touch with prospective system at the time of developing and making change wherever required. This is done with regard to the following points:

- Output Screen design.
- Input Screen design.
- Menu driven system.

8.2 Implementation

Implementation is the stage of project, when theoretical design is turned in to a working system. The most crucial stage is achieving a successful system and confidence that the new system will be work effectively. It involves careful planning, investigation of the manual system and to new system.

Implementation means converting a new or revised system design into an operational one. The implementation includes all those activities that take place to convert from the old system to the new one

There are several activities involved while implementing a project:

- Careful planning.
- Investigating the current system and its constraints on implementation.
- Design of methods to achieve the changeover.
- Training of the staff in the changeover procedure and evaluation of change over method.

Implementation is the final stage and it is an importance phase. The first task in implementation was the implementation planning, that is deciding on methods to be adopted. After the system was implemented successfully, training of the user was one of the most important subtasks of the developer. For this purpose the user or system manual were prepared and handled over to the user to operate the developed system. So change over plays a vital role, which checks the developed tool for the following requirements, and then only the user accepted the developed tool. The changeover took place only when the system had been proved to the satisfaction of the system analysis and other implementation activities have been completed.

SYSTEM MAINTENANCE

9. SYSTEM MAINTENANCE

Maintenance is making adaptation of the software for external changes (requirements changes or enhancements) and internal changes (fixing bugs). When changes are made during the maintenance phase all preceding steps of the model must be revisited.

There are three types of maintenance:

- Corrective (Fixing bugs/errors)
- Adaptive (Updates due to environment changes)
- Perfective (Enhancements, requirements changes).

Maintenance is an enigma of the system development. The definition of the software maintenance can be given describing four activities that are undertaken after the program is released for use.

The maintenance activity occurs since it is unreasonable to assume that software testing will uncover all in a large system. The second activity that contributes the definition of maintenance occurs since rapid changes are encountered in every aspects of computing. The third activity involves recommendation for new capabilities, modification to the existing functions and general enhancements when the software is used. The fourth maintenance activity occurs when software is changed to improve future maintainability or reliability.

FUTURE ENHANCEMENT

10. FUTURE ENHANCEMENT

The popularity of stock market trading is growing rapidly, which is encouraging researchers to find out new methods for the prediction using new techniques. The forecasting technique is not only helping the researchers but it also helps investors and any person dealing with the stock market. In order to achieve more accuracy external factors can be added to the system. In this proposed system users need to download the data from Yahoo finance, need pre-process and cleaning for data, in order to be trained and create model. But that can be automated in future. This system can only predict to maximum of one to three days. But can increase the days of prediction future. With the rise in technology system can be more accurate in future. The limitation of the proposed system is its computational speed, especially with respect to LSTM as the computational cost increases with the number of forward day predictions. The proposed model does not predict well for sudden changes in the trend of stock data. This occurs due to external factors and real-world changes affecting the stock market. Can overcome this in the future. Can modify the same system to an online-learning system that adapts in real-time.

CONCLUSION

11. CONCLUSION

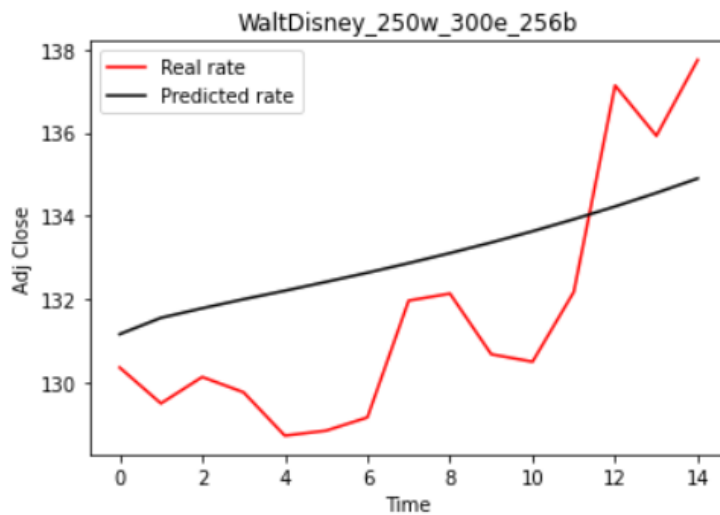
Evaluating the Stock market prediction has at all times been tough work for analysts. Thus, the attempt to make use of vast written data to forecast the stock market indices. Joining both techniques of textual mining and numeric time series analysis the accuracy in predictions can be achieved. Artificial neural network is qualified to forecast BSE market upcoming trends. Financial analysts, investors can use this prediction model to take trading decision by observing market behavior. Thus, as above in the proposed method, train the data using existing stock dataset that is available. Using this data to predict and forecast the stock price of n-days into the future. The average performance of the model decreases with increase in number of days, due to unpredictable changes in trend. The current system can update its training set as each day passes so as to detect newer trends and behave like an online-learning system that predicts stock in real-time.

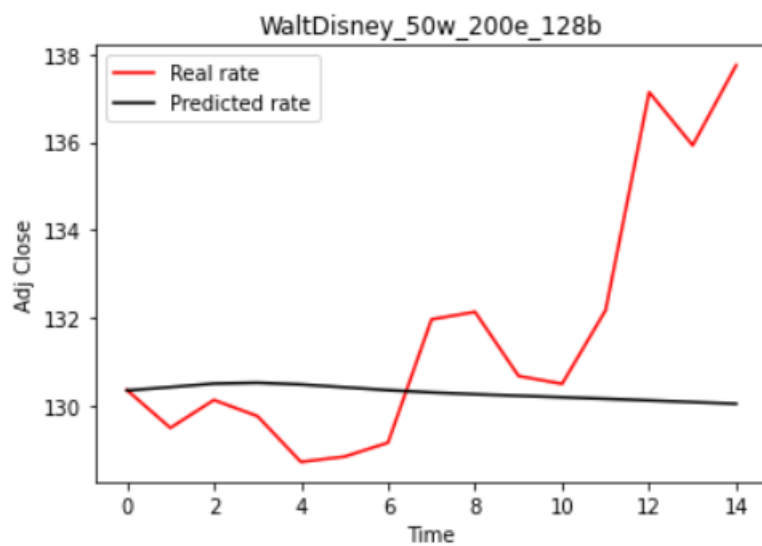
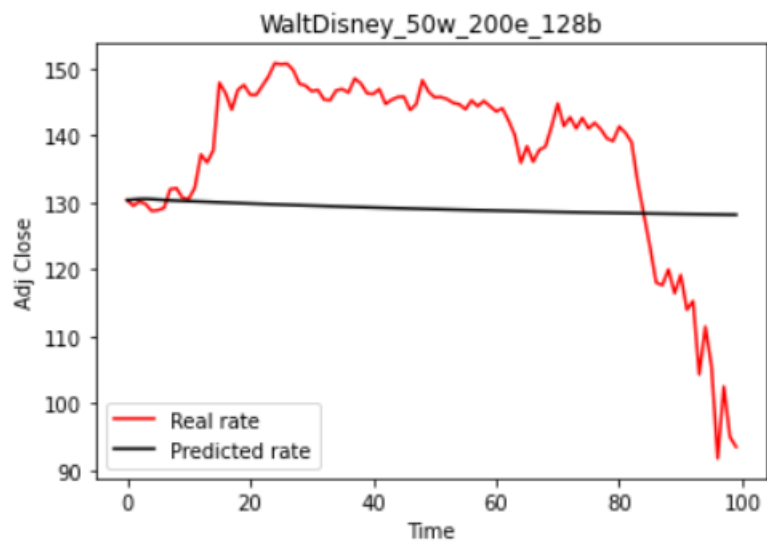
APPENDIX

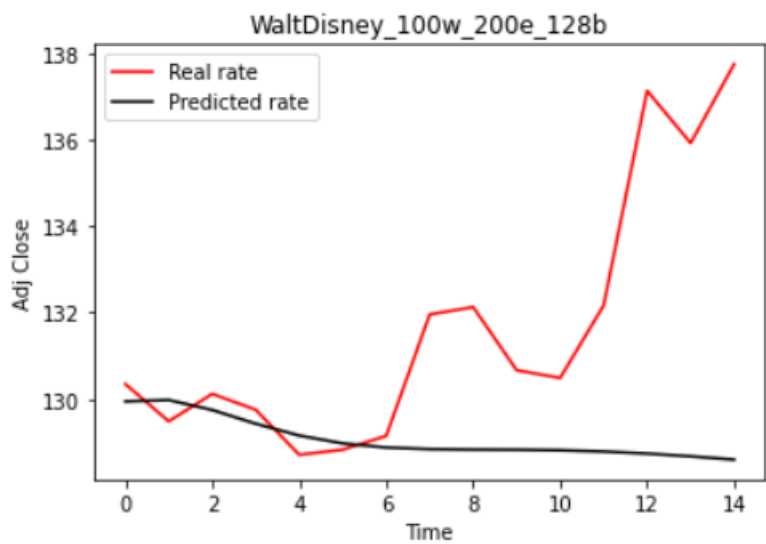
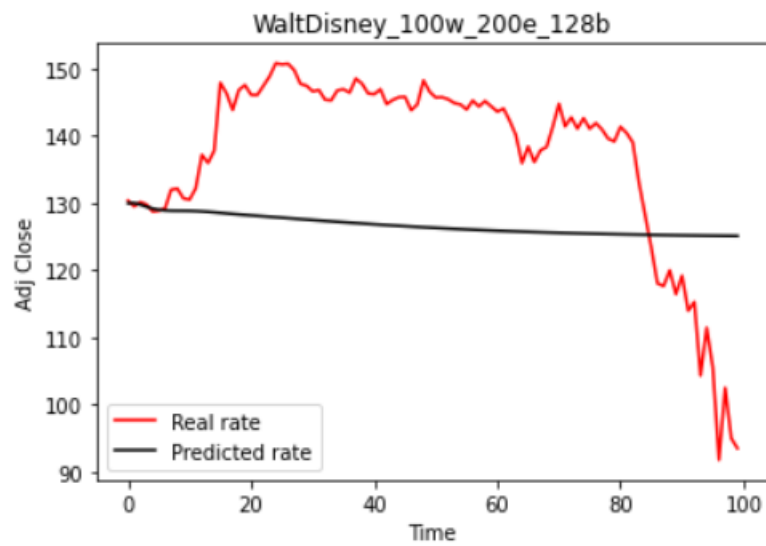
12. APPENDIX

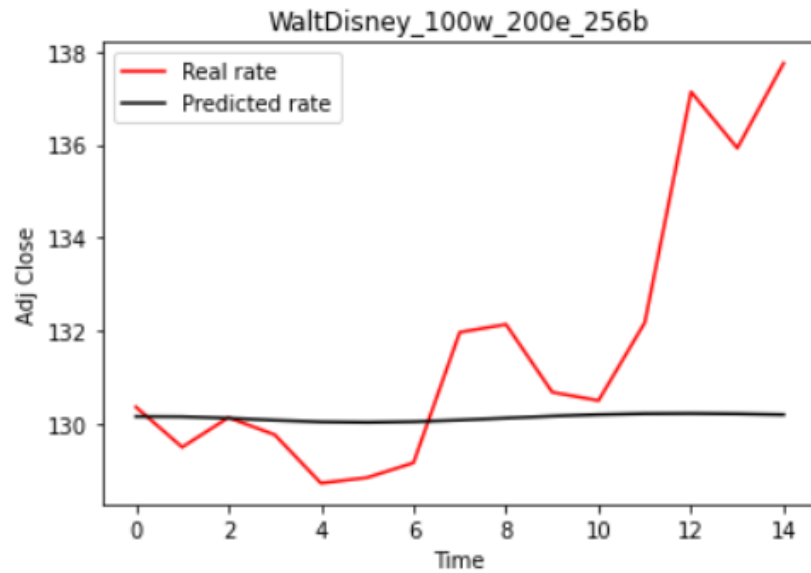
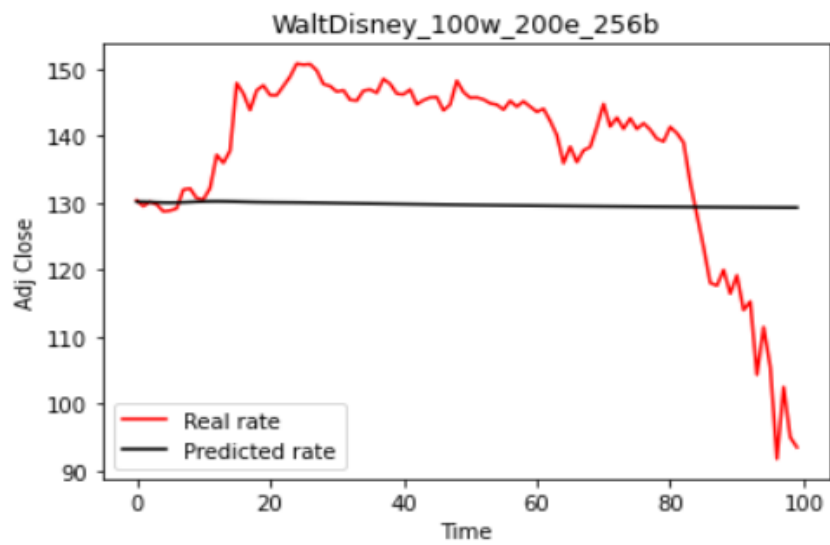
Screenshots

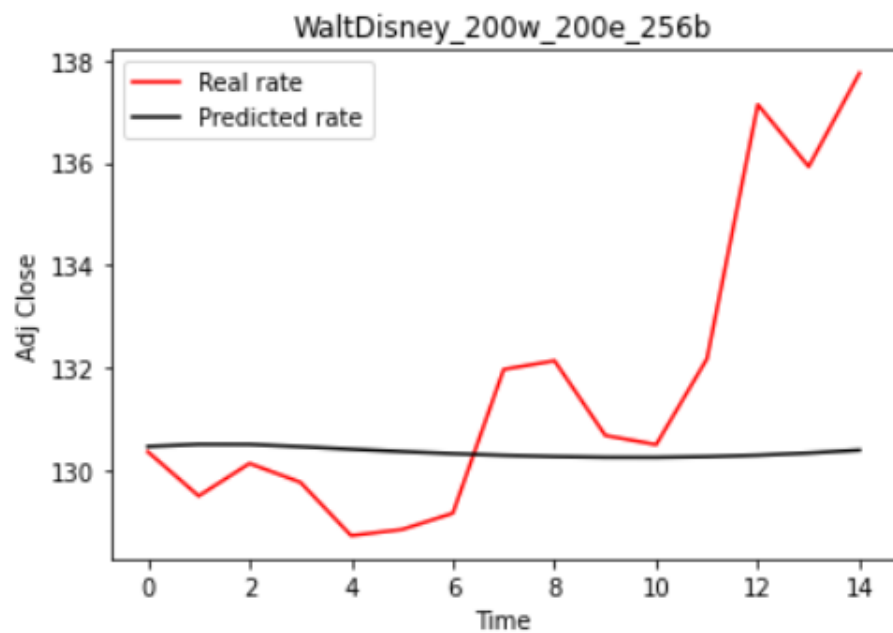
Predictions

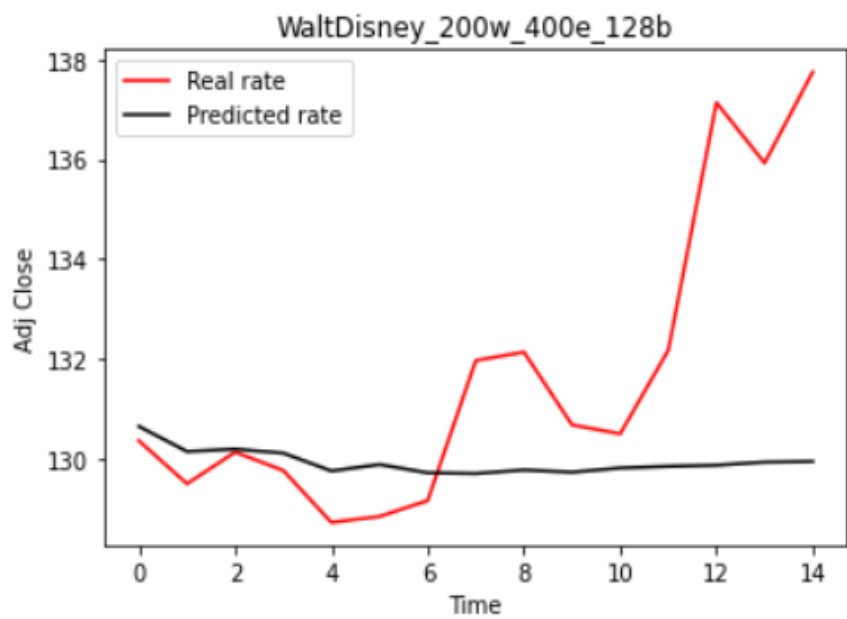
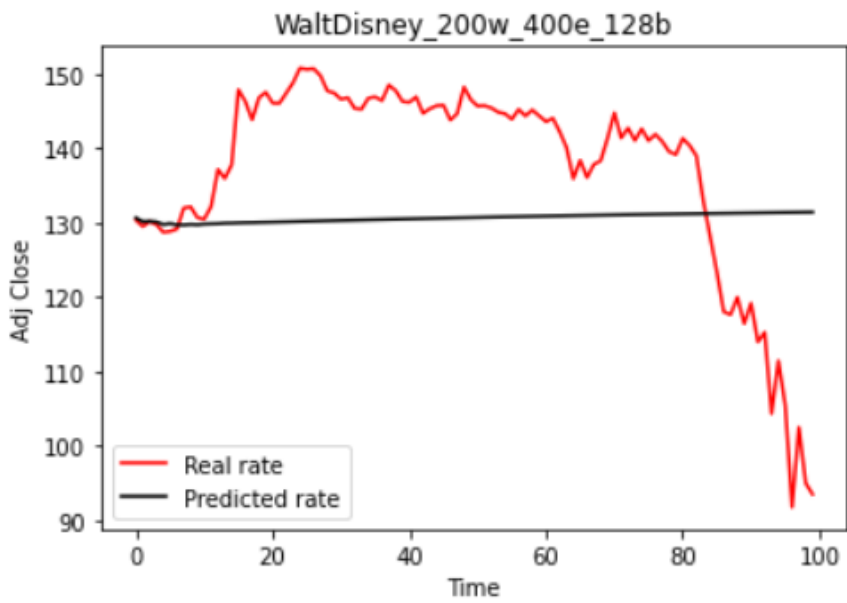


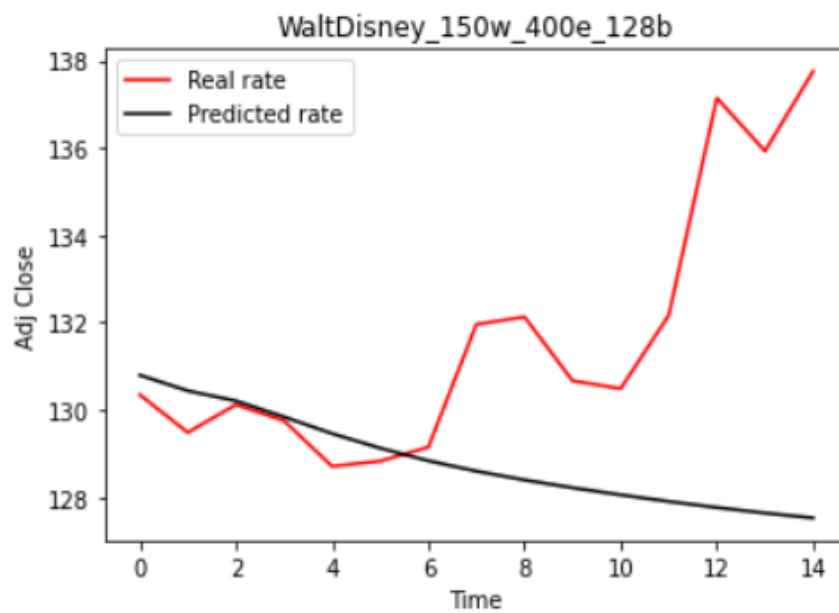
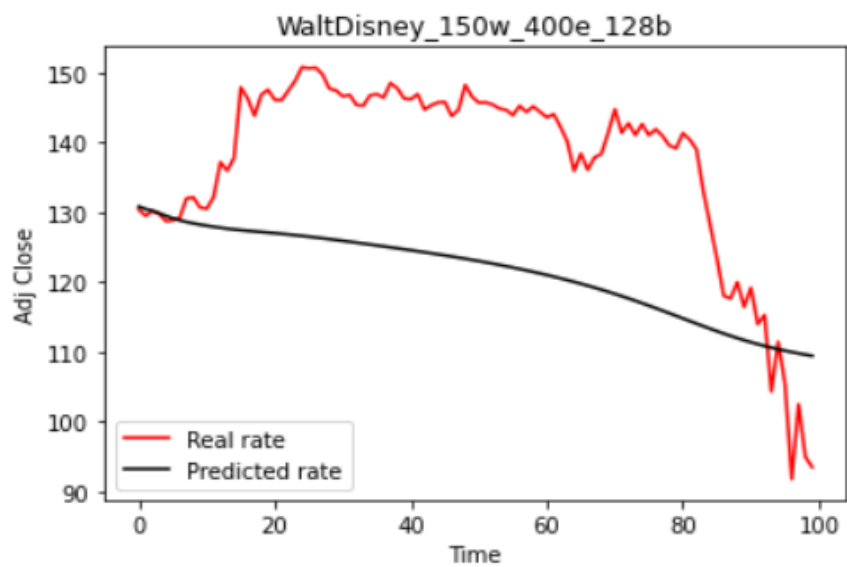


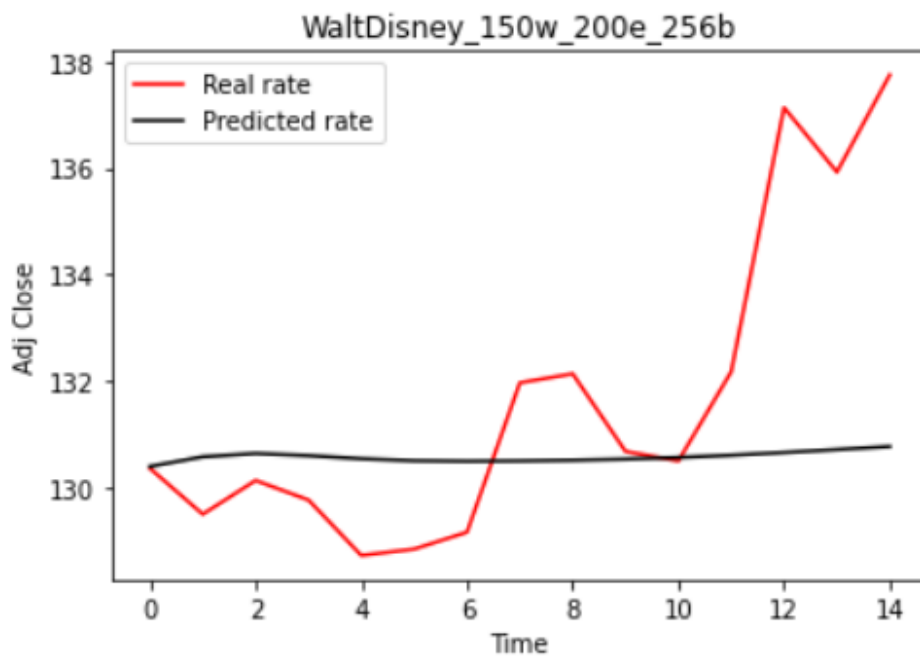












Yahoo finance historical data on stock price

Time period: 16-Sep-1994 - 04-Mar-2020 ▾

Show: Historical prices ▾

Frequency: Daily ▾

Apply

Currency in USD

[Download data](#)

Date	Open	High	Low	Close*	Adj. close**	Volume
03-Mar-2020	105.77	109.44	104.77	105.38	105.38	6,18,200
02-Mar-2020	102.83	105.90	101.38	105.77	105.77	11,59,800
28-Feb-2020	100.80	103.21	99.39	102.29	102.29	9,40,500
27-Feb-2020	106.26	107.97	104.28	104.49	104.49	6,53,600
26-Feb-2020	108.23	109.85	107.84	108.62	108.62	7,63,500
25-Feb-2020	109.06	110.31	106.69	107.46	107.46	5,74,400
24-Feb-2020	107.11	109.64	106.78	108.75	108.75	4,85,600
21-Feb-2020	108.47	111.97	107.79	110.95	110.95	8,87,600
20-Feb-2020	117.52	118.86	106.01	109.40	109.40	14,23,800
19-Feb-2020	116.51	117.78	116.32	117.20	117.20	6,81,700
18-Feb-2020	115.77	116.91	115.27	116.02	116.02	3,65,500
14-Feb-2020	116.35	116.65	115.33	116.05	116.05	4,05,500
13-Feb-2020	116.55	116.75	116.14	116.38	116.38	1,95,300

BIBLIOGRAPHY

13. BIBLIOGRAPHY

Websites

- [1] <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- [2] <https://www.kaggle.com/notebooks>
- [3] <https://colab.research.google.com/notebooks/intro.ipynb>
- [4] <https://docs.djangoproject.com/en/3.0/intro/tutorial01/>
- [5] <https://in.finance.yahoo.com/>

Journals

- [1] A LSTM-based method for stock returns prediction: A case study of China stock market, by Kai Chen, Yi Zhou and Fangyan Dai, *IEEE International Conference on Big Data*, 2015.
- [2] Prediction of Stock Market Using Artificial Neural Network, by Neelima Budhani, Dr. C. K. Jha, Sandeep K. Budhani, *International Conference on Soft Computing Techniques for Engineering and Technology (ICSCTET)*- 2014.
- [3] Price Prediction of Share Market using Artificial Neural Network (ANN), by Z.H. Khan, T.S. Alin and M.A. Hussain, *International Journal of Computer Applications*, 22, No 2, pp. 42-47, 2011.
- [4] Performance Comparison of Time Series Data Using Predictive Data Mining Techniques, *Advances in Information Mining*, by S.Saigal and D. Mehrotra, No.1, pp. 57-66, 2012.

Books

- [1] Deep learning for multivariate financial time series, by Gilberto Batres-Estrada.
- [2] Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, by Aurélien Géron. 2nd Edition.
- [3] Neural Networks and Deep Learning: A Textbook Book, by Charu C. Aggarwal.