**STOCK MARKET PREDICTION USING PYTHON**

Submitted in partial fulfillment of the requirements for the award of Bachelor of science in computer science

**BY**

**MS.B.JOTHILAKSHMI(39290044)**

**MS.M.SHALINI (39290093)**


**DEPARTMENT OF COMPUTER SCIENCE**

**SCHOOL OF COMPUTER SCIENCE**

# SATHYABAMA

2022 INSTITUTE OF SCIENCE AND TECHNOLOGY

**(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by JEPPIAR NAGAR, RAJIV GANDHI SALAI, CHENNAI-600 119**

**May-2022**

# SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY**

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

JEPPIAAR NAGAR , RAJIV GANDHI SALAI CHENNAI -600119

www.sathyabama.ac.in

**DEPARTMENT OF COMPUTER SCIENCE**

## BONAFIDE CERTIFICATE

This is to certify that the Project Report is the bonafide work of B.JOTHILAKSHMI (39290093) and M.SHALINI(39290093) who carried out the project entitled "STOCK MARKET PREDICTION USING PYTHON " under our supervision from November 2021 to May 2022.

**Internal Guide**

(**Mrs. M.SELVI, ME.,PH.D.,**)

**Head of the department**

**(**Dr. L. LAKSHNMANAN M.E., Ph.D.,**)**

---

**Submitted for viva voce examination held on**

**Internal Examiner** **External Examiner**

## DECLARATION

I , **M.SHALINI** here by declare that the Project Report entitled "**STOCK MARKET PREDICTION USING PYTHON**" done by me under the guidance of **Dr. M.SELVI** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering / Technology degree in Computer science.

**DATE**:

**PLACE: CHENNAI** **SIGNATURE OF THE CANDIDATE**

## ACKNOWLEDGEMENT

**I** am pleased to acknowledge my sincere thanks to Board of Management Of **SATHYABAMA for their kind encouragement in doing this project and for Completing it successfully. I am grateful to them.**

I convey my thanks to **Dr. T. SASIKALA, M.E., Ph.D., Dean,** School of Computer and **Dr. S . VIGNESHWARI M.E., Ph.D., and Dr. L. LAKSHNMANAN M.E., Ph.D., Head** of the Department for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide Mrs. **M.SELVI, ME., PH.D.,** for his valuable guidance, suggestions andconstant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of computer science** who were helpful in many ways for the Completion of the project.

## ABSTRACT

The prediction of a stock market direction y serve as an early recommendation system for short -term investors and early financial distress warning system for a long term shareholder. Forecasting accuracy is the most important factor in selecting any forecasting methods. Research efforts in improving the accuracy of forecasting models are increasing since the last decade. The appropriate stock selections those are suitable for investment is a very difficult task. The key factor for each investor is to earn maximum profits on their investments. In this paper support vector machine algorithm (SVM) is used. SVM algorithm is very specific type of learning algorithms characterized by the capacity control of the decision function, the user of the kernel functions and the scarcity of the solution. In this paper, we investigate the predictability of the financial movement with SVM. To evaluate the forecasting ability of SVM, we compare its performed with decision trees. These methods are applied on 2 years of data retrieved from Yahoo finance. The result will be used to analyze the stock prices and their prediction in depth in future research efforts.

## TABLE OF CONTENT

## LIST OF FIGURES

## LIST OF ABBREVIATION

| S.NO | ABBERVIATION | EXPANSION |
|---|---|---|
| 1 | LSTM | Long short-term memory |
| 2 | GRU | Gated recurrent unit |

# CHAPTER 1

# INTRODUCTION

Forecasting of stock market is a way to predict future prices of stocks. It is a long time attractive topic for researcher and investors from its existence. The Stock prices are dynamic day by day, so it is hard to decide what is the best time to buy and sell stocks. Machine Learning provides a wide range of algorithms, which has been reported to be quite effective in predicting the future stock prices. In this project, we explored different data mining algorithms to forecast stock market prices for NSE stock market. Our goal is to compare various algorithms and evaluate models by comparing prediction accuracy. We examined a few models including Linear regression, Arima, LSTM, Random Forest and Support Vector Regression. Based on the accuracy calculated using RMSE of all the models, we predicted prices of different industries. For forecasting, we used historical data of NSE stock market and applied a few preprocessing methods to make prediction more accurate and relevant. The ultimate goal of our application is to serve retail investors as a third party investment tool that uses machine learning to help them navigate in the fast-changing stock market. The project aims to introduce and democratize the latest machine learning technologies for retail investors. No prediction is 100% accurate. Therefore, the upper bound and lower bound of the stock prices will be displayed to illustrate the trading range the investors should be looking this application serves. Stock market prediction ad analysis are some of the most difficult jobs to complete. There are numerous causes for this, including market volatility and variety of other dependent and independent variables that influence the value of a certain stock in the market. These variable make it extremely difficult for any stock market expert to anticipate the rise and fall on the market with great precision. Machine learning is a strong algorithm the most recent market research and stock market prediction advancements have begun to include such approaches in analyzing stock market data. Implementation of analyzing and forecasting the stock price in python using various machine learning algorithms. Stock price prediction using machine learning helps to discover the future value of a company stock and other financial assets traded on an exchange. The entire idea of stock prices is an gain significant profits. Stock

market helps companies to raise capital . It helps to generate personal wealth. Stock market server as an indicator of the state of the economy. It is a widely used source for the people to invest money in companies with high growth.

# CHAPTER 2
# 2.1 LITERATURE SURVEY

### 2.1.1 STOCK PRICE PREDICITION

From the research paper “Machine Learning in Stock Price Trend Forecasting” written by Y. Daiand Y. Zhang In Stanford University, they used features like PE ratio, PX volume, PXEBITDA, 10-day volatility, 50-day moving average, etc. To predict the next-day stock price and long-term stock price. The machine learning algorithms used in the research are Logistic Regression, Gaussian Discriminant Analysis ,Quadratic Discriminant Analysis ,and SVM. The accuracy ratio is defined as the number of days that the model correctly classified the testing data over the total number of testing days. With the short-term model predicting the next day stock price, it has very low accuracy, the Quadratic Discriminant Analysis is the best among all models, it scored a 58.2% accuracy. With the long -term model predicting the next n days stock prices, the longer the time frame, the better in the accuracy for SVM. With a time window of 44 days, the VM model ’saccuracyreached79.3%. Apart from that, It was found that by increasing the number of features, the accuracy increased. When all of the 16 features were used, the accuracy of the model reached 79%, while it fell to 64% when only 8 features were used, and 55% if only 1 feature was used. Our project will also investigate how the time frame would affect the accuracy of price predictions of different models. As models have to reach a certain significance for the users to work as a reference, it is essential for us to optimize our model to figure out what the optimal parameters and model structure are for our stock price prediction purpose. B.Wanjawa and L. Muchemi demonstrated the potential in predicting stock prices using ANN, as shown in the research paper “ANN Model to Predict Stock Prices at Stock Exchange Markets”[4]. They used 70% of the training data to predict the stock prices for the next 60 days. Through optimizations, they were able to predict the actual closing prices within 0.71% mean absolute percentage error (MAPE), with the highest variance-3.2% among all of the 62 days. This demonstrated a high potential for using machine learning to accurately predict stock prices. This is one of the key components in our application where algorithms have to be designed to have high accuracy, such that the platform could be useful for retail investors.

### 2.1.2 NEURAL NETWORK

A neural network attempts to learn a function that maps the input features to the output predictions, serving as a universal function approximator. It consists of a network of neurons, each of which represents a weighted sum of inputs. Outputs from neurons are fit into activation functions which introduce non-linear to the system, and the n passed to some other neurons. In a typical dense feedforward neural network, the network consists of layers of neurons stacked together, with neurons between individual layers fully connected.

### 2.1.3 LONG SHORT-TERM MEMORY[LSTM]:

Long short-term memory [8] was first introduced by Hochreiter and Schmid huber in 1997 to address the a for mentioned problems. Long-short term memory tackles the problem of learning to remember information over a time interval, by introducing memory cells and gate units in the neural network architecture. A typical formulation involves the use of memory cells, each of which has a cell state that store previously encountered information. Every time an input is passed into the memory cell, and the output is determined by a combination of the cell state (which is a representation of the previous information), and the cell state is updated. When another input is passed into the memory cell, the updated cell state and the new input can be used to compute the new output.

### 2.1.4 EVOLUTION ALGORITHM:

Researches have shown that large-scale evolution can auto-generate neural network model architectures and hyperparameters with performance comparable with state-of-the-art human-designed models. In a research in 2017 , a large-scale evolution for discovering image classification neural networks was run. It started with a huge population of randomized simple1-layermodels, then slowly evolved the population by removing a poor model and generating a new model by mutating some parameters of a good model in each iteration. After hundreds of hours of running the algorithm with huge computing power, most models in the population achieved state-of-the-art results on CIFAR datasets. In each iteration, only a simple mutation that changed 1 parameter was

applied, which allowed searching in a large search space. The paper showed the possibility of finding good models by using lots of computational power to replace human-machine learning experts and has set the foundation of learning with AutoML.

## 2.2 SYSTEM ANALYSIS

### 2.2.1 EXISTING ANALYSIS :

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

### 2.2.2 PROPOSED SYSTEM :

In this project, we have made a time-series analysis and it doesn't need n-fold cross validation. Methodology since it's sequential data. We split our dataset in train and test data. Top 80 percent of data will be Train data and the remaining will be test. The prediction model, which is based on SVM and independent analysis, combined called SVM-ICA, is proposed for stock market prediction. various time series analysis models are based on machine learning. The SVM designed to solve regression problems in non-linear classification and time series. In this proposed system, we focus on predicting the stock values using machine learning algorithms like Random Forest and Support Vector Machines.

## CHAPTER 3

### 3.1 AIM OF THE PROJECT:

Stock market prediction means determining the future scope of the market . The prediction of a stock market direction y serve as an early recommendation system for short -term investors and early financial distress warning system for a long term shareholder. Forecasting accuracy is the most important factor in selecting any forecasting methods. Research efforts in improving the accuracy of forecasting models are increasing since the last decade. The appropriate stock selections those are suitable for investment is a very difficult task. The key factor for each investor is to earn maximum profits on their investments .In this paper support vector machine algorithm (SVM) is used. SVM algorithm is very specific type of learning algorithms characterized by the capacity control of the decision function , the user of the kernel functions and the scarcity of the solution .

### 3.2 SCOPE AND OBJECTIVES OF THE PROJECT:

The project aims to introduce and democratize the latest machine learning technologies for retail investors. No prediction is 100% accurate. Therefore, the upper bound and lower bound of the stock prices will be displayed to illustrate the trading range the investors should be looking at. This application serves as a supplementary quantitative tool for investors to see the market at a different perspective with the help of technology. This project is divided into 2 parts, namely are search component and an application component, aiming to provide retail investors with stock price predictions using different machine learning models in a good user experience way for reference.

#### 3.2.1RESEARCH:

This project will investigate how different machine learning techniques can be used and will affect the accuracy of stock price predictions. Different models, from linear regression to dense and recurrent neural networks are tested. Different hyperparameters are also tuned for better performance These arch space for all neural network architectures and hyperparameter combinations is huge, and with limited time in

conducting this project, apart from manually trying combinations, the team optimizes the models with evolution algorithm, replicating AutoML techniques from other researches with promising results in the financial context.

### 3.2.2 APPLICATION:

This project aims to provide stock price predictions based on the latest machine learning technologies to all retail investors. A mobile web application is developed to provide predictions in an intuitive way. Different models' performance and accuracy can also be compared. The application also serves as another user interface (UI) in visualizing results from the research apart from Jupyter notebooks with lots of tables and graphs

## 3.3 OVERVIEW OF THE PROJECT:

There are over 2.2 million Hong Kong stock investors, who contributed about 15% of the cash market trading value in 2016. The total cash market trading turn over is around HK$1.6trillion. In particular, retail investors have made buy or sell investment decisions worth a total turn over of $240 billion for the year of 2016 [1].In Hong Kong, there are a lot of investment decisions that involve a large sum amount of money being made. Retail investors spend a lot of time finding investment opportunities. Wealthier investors could seek professional financial advisory services, but for typical retail investors, the costs are prohibitive. Thus, retail investors have to figure out the market themselves and make informed decisions on their own. This makes investment very stressful in modern societies.

Unfortunately, humans are irrational in nature. Without quantitative, data-driven models, decisions get swayed by cognitive biases or personal emotions, resulting in unnecessary losses. Even if investors are cautious enough, most do not have sufficient skills to process a huge volume of data required to make good judgments. Institutional investors rely on sophisticated models supported by technologies to avoid traps, but retail investors do not have access to such technologies and often find themselves falling behind the market. Without access to quantitative and data-driven models, one obvious approach retail investors could use to evaluate the market is through simple indicators, for example, linear regression and exponential moving average (EMA) (Figure 1.1).Two important

indicators are 20-day EMA and 50-day EMA. When the 20-day EMA rises above the 50-day EMA, the stock is likely to trend upward, and vice versa. Another obvious approach retail investors might use to predict the stock market is to draw a linear regression line that connects the maximum or minimum of candle sticks.


**Figure 3.3** Linear regression method to evaluate and predict the market trend

Inspired by the increasing popularity of deep learning algorithms for forecasting application, these algorithms might serve as potential tools to find hidden patterns in the trend of stock prices, this information could be useful to provide extra insights for retail investors when making investment decisions. Therefore, this final year project aims to investigate the usefulness of deep learning algorithms in predicting stock prices and democratize such technologies through an easy to use interface for the general public.

# CHAPTER 4

## TECHNOLOGY USED:

- MACHINE LEARNING MODELS USED: FB Prophet
- Python for backend
- Flask framework for integration of frontend and backend
- JavaScript frame work with CSS and HTML for front end
- Dygraphs for plotting

## 4.1 HARDWARE REQUIRED:

- Hard disk : 500 GB and above.
- Processor : i3 and above.
- Ram : 4GB and above.
-

## 4.2 SOFTWARE REQUIRED:

- Operating System : Windows 10 (64 bit)
- Programming Language : Python
- Tool : Jupyter

# CHAPTER 5
# METHODLOGY DESIGN

## 5.1 SYSTEM ARCHITECTURE:

The architecture of the system follows a client-server model ,where the server and the client are loosely coupled.

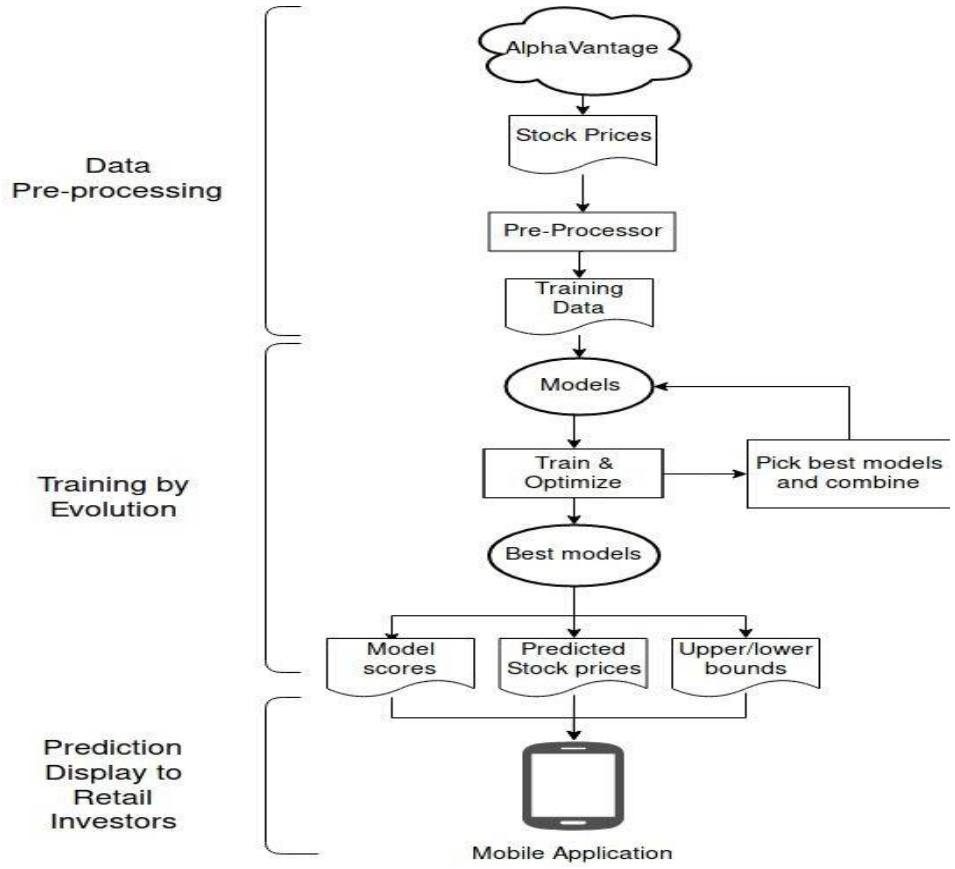
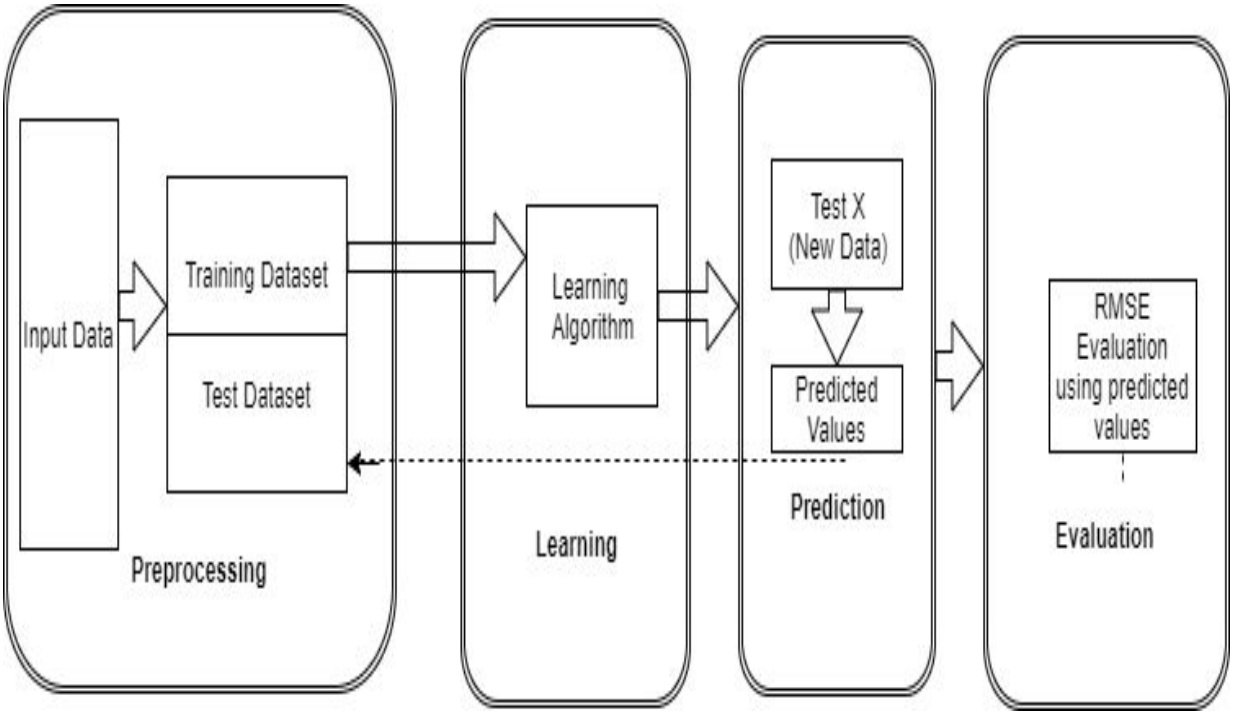



**Figure 5.1** System Architecture Diagram




After relevant stock data are retrieved from the third-party data provider through the cloud, the backend pre-processes the data and builds the models. After that, predictions are made and the prediction results will be stored on another cloud, which can be retrieved from the mobile application. The advantages of the loosely coupled architecture include improved scalability and ease of collaboration. The workload for the cloud which serves the models and the one which serves the mobile application will be very different. One cloud serves the model prediction results, which are simple text files; another cloud serves the mobile application with a lot of rich user content such as images and large UI libraries. Having two clouds to adapt to two different demand patterns is more efficient, especially since cloud providers these days usually serve content on demand. Also, these parathion allows different team members in the team to focus on different parts after agreeing on a common interface. It speeds up development as team members responsible for different parts of the system do not need to take care of the under lying implementation details. Also, it is easier to swap out different components, e.g. to replace the models the team could simply make changes to the backend, while the frontend remains unaffected.

## 5.2 PROBLEM FRAMING:

The problem of the project is set to predict the stock price for the next 10 business days. “10days” is chosen as the time frame as short term price movements tend to depend more on trend momentum and price pattern, while long term price movements depend on the fundament also a stock (e.g. company management capabilities, revenue model, market demand, macroeconomic factors, etc.). The loss function of the training algorithm is the mean squared error of the 10 predicted stock prices. The training algorithm or optimizer is set to minimize its value, and it serves as the basic performance metric for comparing different models. Other scores are defined to provide more in-depth in sights on a model predictability performance and finance-domain-based comparisons between models for investors. Two different prediction approaches are mainly tested, predicting the stock prices for the next10 days directly and predicting the stock price of the next day 1 at a time. It is suspected that the two different problem framing approaches will result in different abstractions learn performance for different use-cases.

## 5.3 ROBUST DESGIN:

For the research side, the system is designed to be as robust as possible to facilitate model testing. Each model can be defined by a pair of model options and input options, specifying the model configurations and the input sit takes. This accelerates the process of testing out different model and/or input configuration combinations.

## 5.4 PERFORMANCE EVALUATION:

Model configurations and the inputs it takes. This accelerates the process of testing out different model and/or input configuration combinations.

### 5.4.1 MOTIVATION:

Apart from the mean squared error that a model tries to minimize, different finance-specifics cores are introduced to evaluate and compare performance of different models, namely model accuracy score, model trend score and stock buy/sell score. The scores are also designed to convey useful and meaningful messages to help investors

understand a stock and make investment decisions.

### 5.4.2 DEFINITION:

In this project, the test set is defined as the last 100 days stock price.

To clearly explain the performance evaluation rationale, the following symbols are defined.

$$P_i : \text{actual price} \quad \hat{P}_i : \text{predicted price} \quad \sigma = stdev(\frac{P_{t+1}}{P_t} - 1) \quad S : \text{set of } Snakes$$

"Snakes" is defined as 10-day disjoint prediction segments in the test set, which will be a set of 10"snake". It includes the actual prices and the predicted prices for the last 100days.

Specifically, Snakes are defined below:

$$Snakes = \{ Snake_i \mid i \in [0, 9] \}$$
$$Snake_i = \{ (\hat{P}_{10i+j}, P_{10i+j}) \mid j \in [1, 10] \}$$

It is named as "Snakes" because intuitively the 10-day disjoint segments look like snakes when being plotted on a graph of historical prices.

### 5.5 MODEL:

Different common neural network models are tested, including dense neural network, simple recurrent neural networks (RNNs), Long short-term memory networks(LSTMs)and gated recurrent unit networks (GRUs).Different model architectures are tested by changing the number of hidden layers, the number of hidden units per hidden layer, and the activation function or recurrent activation function used in each hidden layer. All recurrent neural networks, RNNs, LSTMs, and GRUs, are set to have the recurrent neural networks same high-level architecture (Figure2.2), a stack of recurrent layers by passing

the full output sequence to the next layer, followed by a stack of dense layers.

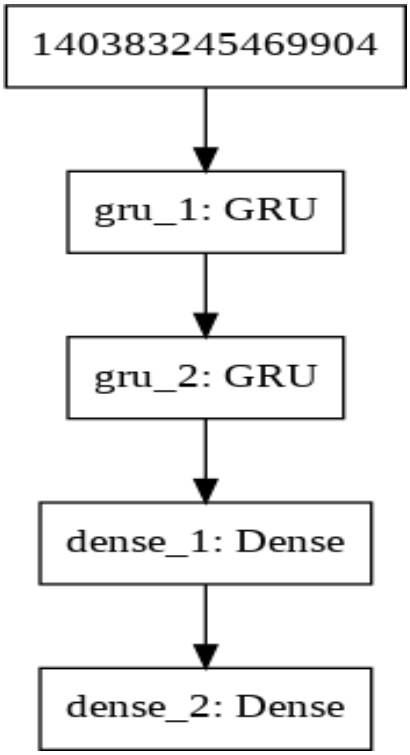
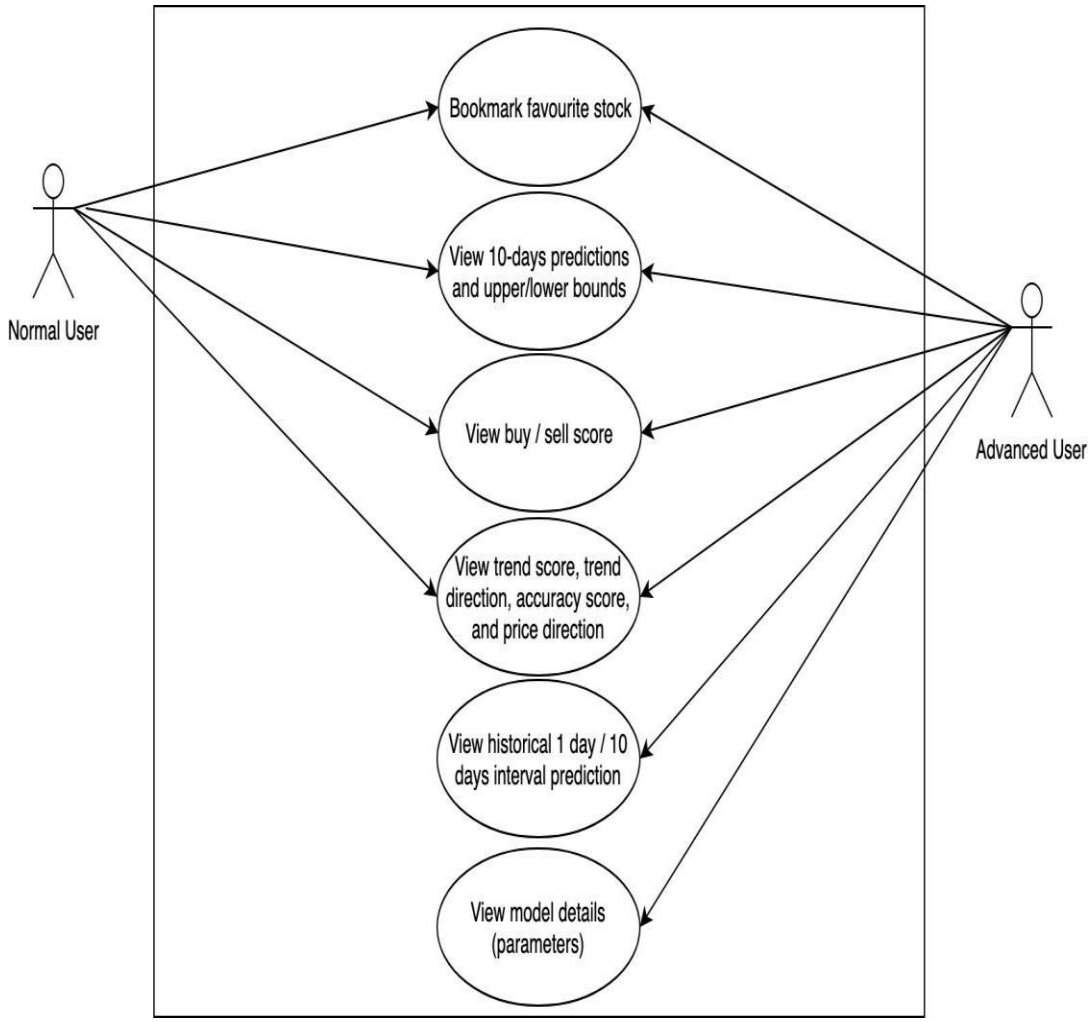



**Figure 5.5** Example of the common high-level architecture

## 5.6 APLICATION DESIGN

### 5.6.1 USER GROUP:

Users are separated into two groups, normal users and advanced users. For users that would like to know about the historical (test set) performance of a model and more information behind the machine learning models like the architecture and inputs, they can enable advanced user mode in the settings page to view those details in each individual stock page.




**5.6** Functionality accessible by normal users and advanced user

### 5.6.2 USER JOURNEY:

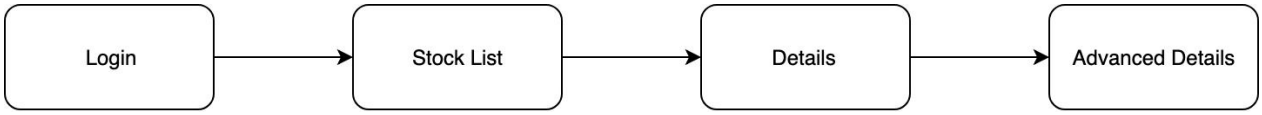



**Figure 5.6** User Journey

First of all, users need to login to use the system, as there will be customization options for different users. Since users might not want to create a separate account just for our application, it will more convenient if users can log in with their existing social media accounts. In particular, Facebook login is a good option, since there are over 2 billion users worldwide. Thus, it might be possible to reach a larger market by adopting Facebook login. Only the very basic user information like the user name will be collected by the system. For normal users (advanced user mode disabled), after logging into the system, they can view the stock list and search from it by stock name.

## 5.7 RESARCH IMPLEMENTATION:

All machine learning-related code are written in Python. Neural networks are implemented with Keras while linear regression model is implemented with scikit-learn .

### 5.7.1 STOCK PRICE PREDICITION:

Data is collected from Alpha Vantage Stock Price API [18].It offers up to 20 years of daily stock price information on S&P500 stocks. A Python script is written to retrieve stock prices of different stocks automatically. The retrieved stock prices are stored as .csv files in a local folder during development and testing. In deployment, the downloaded stock price data will be transformed into a 2D JavaScript array and uploaded to Firebase Cloud Storage immediately. A cron job that launches the data-fetching and data-uploading script is scheduled to run every 8p.m. (EDT) after NYSE and NASDAQ are closed.

### 5.7.2 DATA PRE-PROCESSING:

3 Python scripts are written to transform the raw stock prices (.csv files) into feature vectors, for training, predicting and testing respectively. The scripts take the input options and the raw stock prices as inputs and produce the correct features by building the lookback arrays and the moving averages. It concatenates the features into the final feature vectors, which will be passed to the model for training or testing. The 3 scripts share common operations in building a dataset except the output size and the range of dates to build from, so common functions are written to centralize the logic instead of repeating the same index-calculation-intensive work across functions . Is frequently used for machine learning tasks because it is much for performant than Python lists ,as NumPy arrays are implemented as densely packed lists, instead of a dynamic array where the elements are not stored contiguously. Pandas is a popular framework for pre-processing time series data. It has various utilities for reading raw input files such as .csv and transforming time series data to the correct format. Pandas uses NumPy as the underlying data structure, so it is very convenient to inter operate between the two.

### 5.7.3 MODEL:

A model base class is used as a common interface for all machine learning models. All models then have their own model class, specifying model-specific details like methods to build the model, train the model, use the model and save the model.

To decouple model configurations from software code to provide flexibility and robustness and save engineering effort as mentioned , each model is defined by a JSON object, which specifies the model's architecture and hyperparameters with model options and the model inputs with input options. A corresponding model can then be created by passing the object to the model class constructor. The model options specify which machine learning model to use, and the hyperparameters for the model like the number of hidden layers, the number of hidden units, activation functions used, as well as optimization algorithms and loss functions. Some example model options are in Appendix A.

### 5.7.4 TRAINING:

In training, a randomized initial model is first generated from the model options definition. A training set is generated by the build training dataset script, which generates the training set features from the input options and the raw stock price data. Then, the data is fed into the model for training.

### 5.7.5 SAVING TRAINED MODEL:

All trained models are saved for predicting stock prices in the future. Keras models are saved inh5 format, and scikit-learn models are saved with a Python library named pickle. A dedicated saving format is designed (Appendix C), such that same models (same hash for same model options and input options) for different stocks are saved in the same directory with no collision.

### 5.7.6 PREDICTING STOCK PRICE:

When predicting stock price ,the saved model will first be loaded. Then, a feature vector specified by the input options is built with the build predict dataset script, which is the same as the build training dataset except it returns a flatten 1D feature vector. The feature vector is inputted into the model to predict stock price. For 10-day predict, the predictions are directly out putted. For 1-day predict, the predicted stock price is appended to the raw dataset as if it happened before, then a new feature vector is generated for predicting the stock price for the day after, the process is repeated to predict the stock prices for all next 10 days.

### 5.7.7 SAVE PREDICITIONS:

For each stock, a prediction file can be generated from the save predictions script. It includes all the data and results that the application needs to display, including all 10-day predictions from all models, both 1-day predict test set and snakes test, and the model options and input options for each model. The saved predictions file is then saved to Firebase Cloud Storage and served to the application. During development, the saved predictions file is saved in a local directon.

## 5.8APPLICATION IMPLEMENTATION

### 5.8.1 STOCK INFORMATION COLLECTION:

Company information is collected from the IEX Stock API [25]. A Firebase Cloud Function [26] is written to get the data from the API and store it in Firebase Cloud Fire store [27], which the application will access through another Firebase Callable Cloud Function.

### 5.8.2 GOOGLE CHARTS:

Google Charts [34] is used to plot the stock prices and the predictions. It provides a simple and separate set of APIs that does not depend on other libraries while maintaining customizability. Also, it integrates nicely with the user interface which follows the Material Design, which improves the overall user experience.

### 5.8.3 FB LOGIN VIA FIREBASE AUTHENTICATION TECHNOLOGY:

To facilitate user logins ,the popular Facebook login service powered by Firebase Authentication is used, which provides a set of rich set of APIs to interact with a range of authentication providers, including Facebook Login, and integrates well with other Fire base services that the system relies on. It is also convenient to add other authentication service providers in the future.

### 5.8.4 COMPONENT:

The whole application is broken down into the following hierarchy of components. At the top level, the App component brings everything together. If the user is not logged in he/she will be directed to the login page (Login Page).The router component (Router) controls where the user will end up at, including the home page(Home Page), which is the default starting point for users with a list of stocks and a search bar, all broken down into separate components Other components include the details page (Detail Page), which is where details about a stock price, including the stock price chart, a list of models with a model score attached to each model, along with a buy/sell score that indicates the overall predictions for whether the stock should be purchased or sold, are included in.

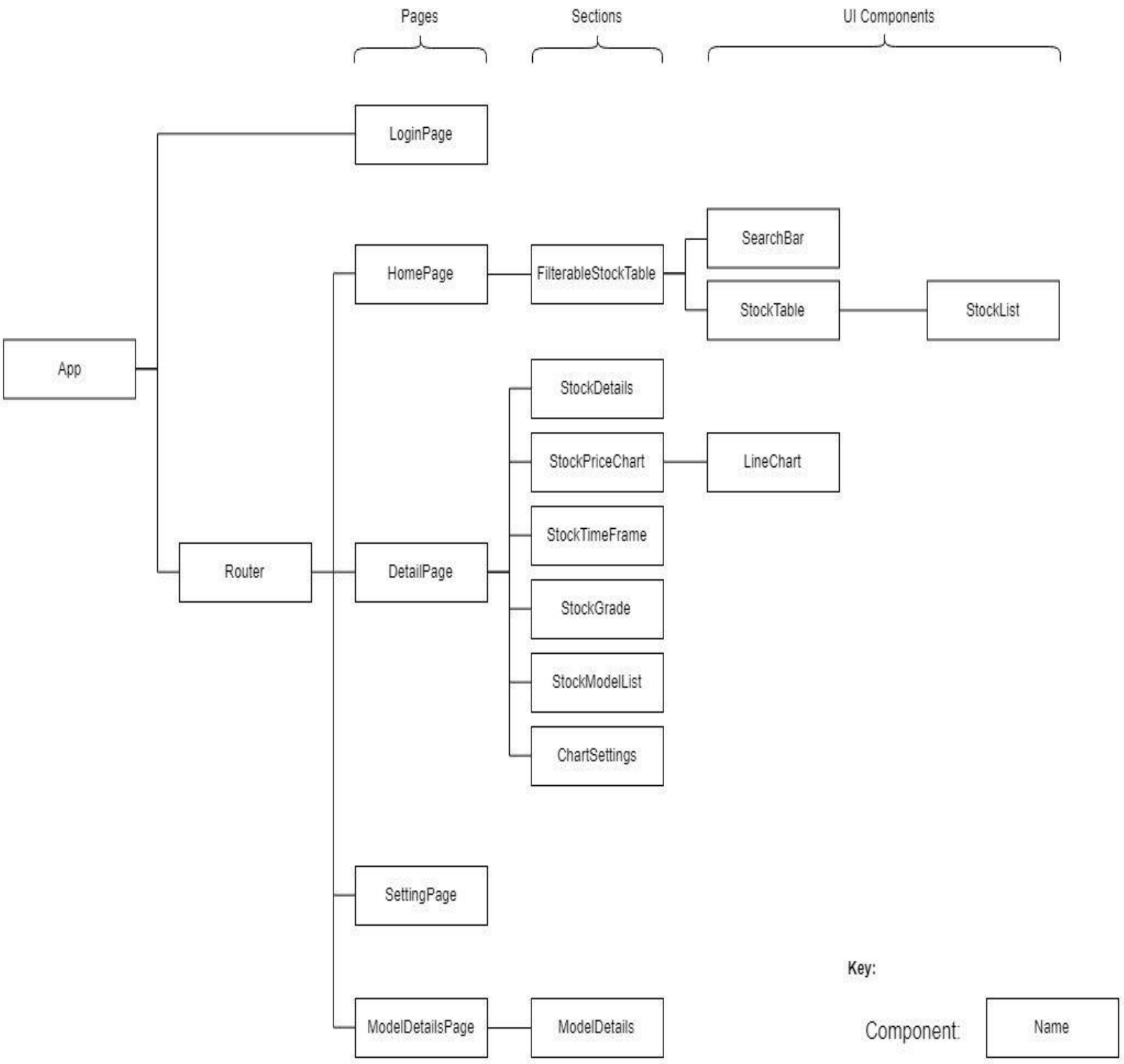



**Figure 5.8** Component Diagram

## 5.9 FINDINGS

### 5.9.1 GENERAL FINDINGS:

The following are some general findings from testing out different machine learning models.

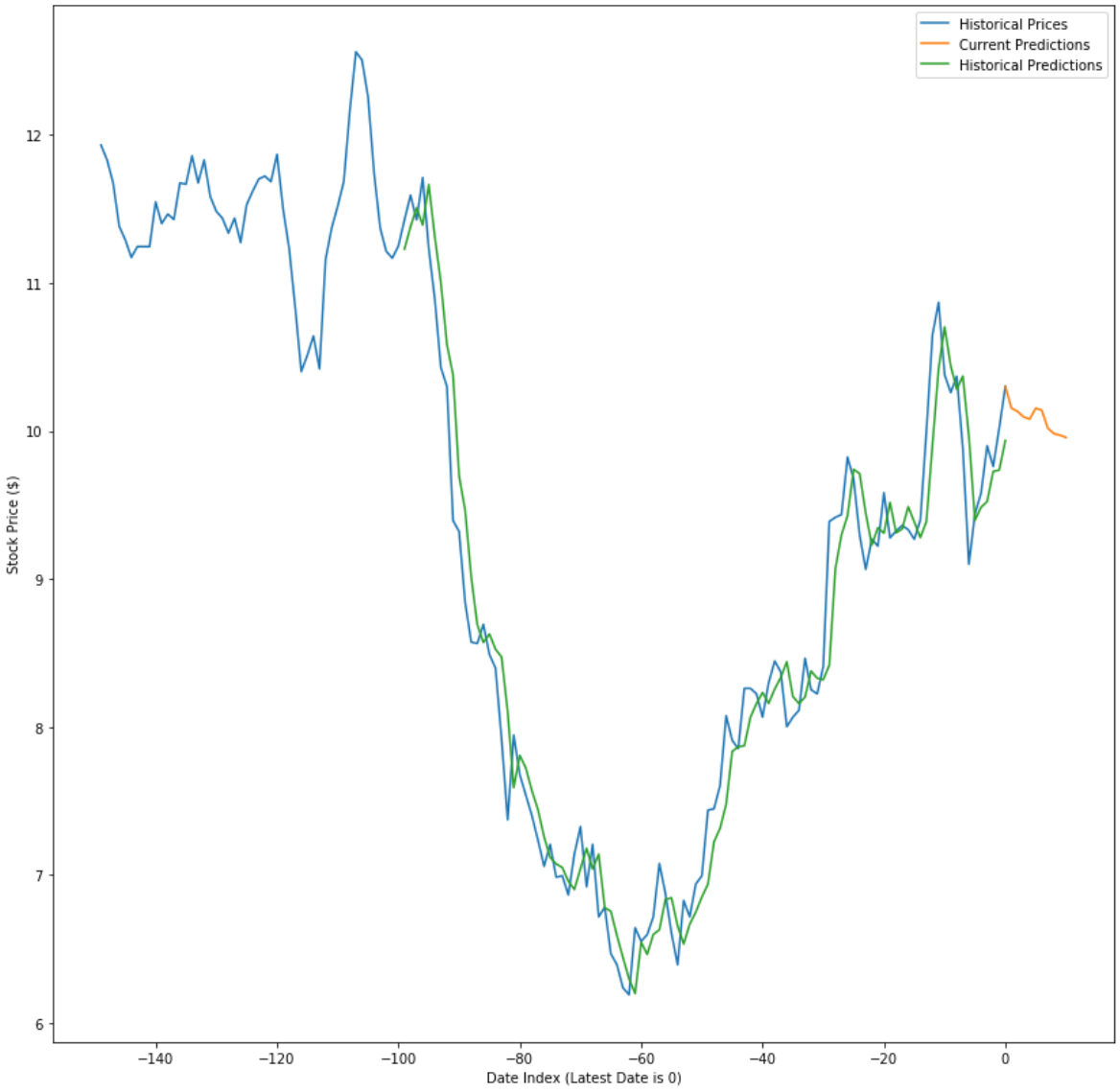



**Figure 5.9** 1-day interval historical predictions (GE, Dense Neural Network)

From Figure 5.9, it shows that the 1-day interval historical predictions line follows closely with the historical prices. The graph looks like the prediction line is just 1 day shifting from the historical prices, similar to a shifted and smoothed out historical prices line. Therefore, the shape of the historical predictions line is similar to the shape of the exponential moving averages (EMA), where the price changes from ttot+1 heavily depends on the direction and Magnitude of changes fromt-1tot, followed by decreasing importance from earlier historical prices. Other models in predicting stock prices of other stocks also show similar results. Although price reflects all available information, the magnitude of price changes in the future might need other data for forecasting purpose, such as market sentiment company announcement, retail and institutional investors' attention on the company, etc. This is one of the possible explanation of why the 10-day interval prediction might have a large difference to actual values as there are potential shifts in market momentum. Therefore, the price might be too compact and other information is required to make a more accurate prediction.

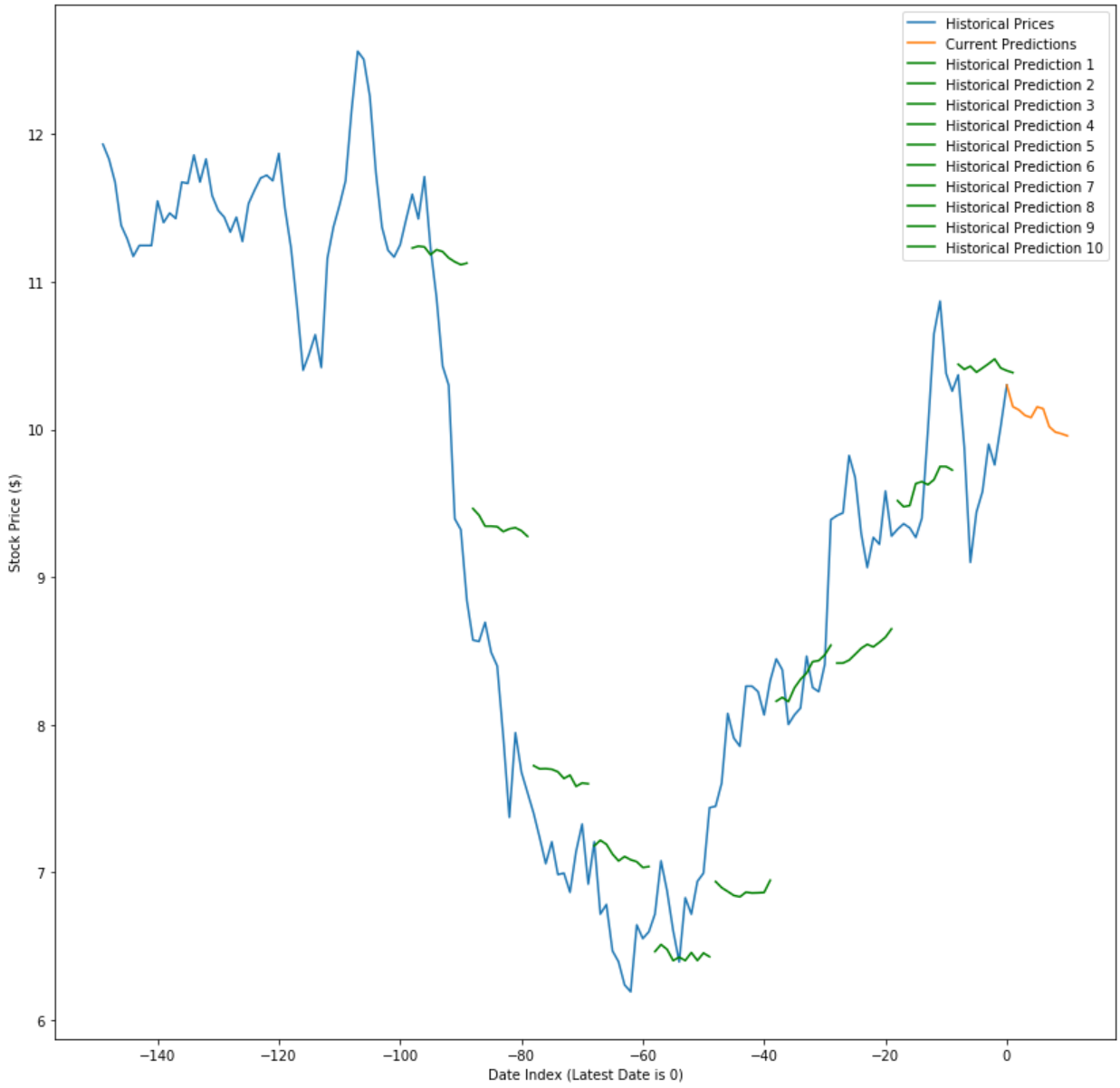



**Figure 5.9** 10-day interval historical predictions (GE, Dense Neural Network)

From Figure 5.9 it shows that the 10-day interval historical predictions line do not follow closely with the historical prices but could demonstrate the trend. For example, historical predictions 1, 2, 3, 4, 7, 8, 9, 10 provided insights on the correct market direction, yet the magnitude did not match the actual price movements. A possible reason for this error can be the 10-day interval prediction has to predict more values while having fewer data compared to the case of 1-day interval prediction, which for 1-day interval prediction, data of close prices until previous day are available. Therefore, a longer period of interval prediction could subject to greater changes in market fundamentals, including market news, macroeconomic factors, earning reports, etc. Other models in predicting stock prices of other stocks also show similar results.

### 5.9.2 PREDICTION FINDINGS

As mentioned in approaches are tested in predicting the stock prices for the next 10days, predicting all 10-day stock prices directly and predicting each stock price one at a time. The 2 different approaches frame the problem totally differently, which introduces a significant language bias. According to the results (e.g. Figure 6.2a and 6.2b), for most stocks, most models that predict10-day stock prices directly have a higher error than predicting individual stock price. However, the errors in predicting different days in the future are relatively constant for models thatpredict10-day stock prices directly, while the error increases with the time from now for models that predict stock prices one day at a time.

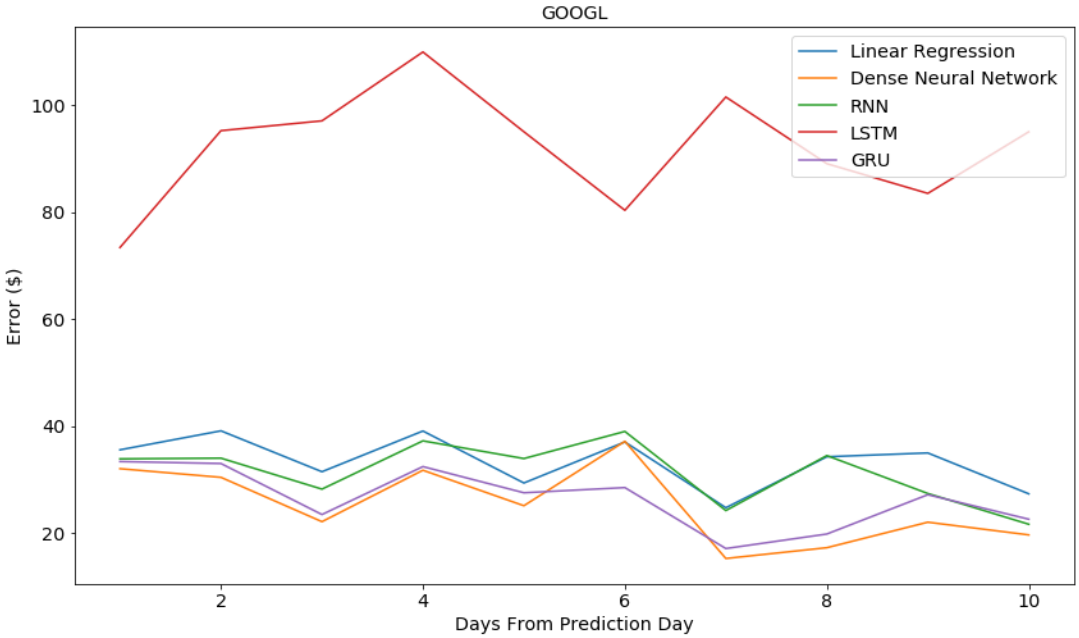



**Figure 5.9** Prediction error in predicting stock price at different future dates(GOOGL, 10-day predict)

# CHAPTER 6
# TESTING

## 6.1 UNIT TESTING:

The unit test module from Python is used to implement all unit tests, as it is available by default in Python and integrates well with existing Python codes. Unit tests are done for the build dataset script, which transforms the raw input data into feature vectors usable for training and testing, as well as models core calculations. Unit tests are conducted because the components are error-pron, calculation intensive. Also, they exhibit garbage-in-garbage-out properties, that the model will be completely wrong if it receives the wrong input, and if the model scores are wrong, the final buy-sell recommendation will be totally incorrect.. Combinations of input options are tested, including n-day stock price look back as well as n-day moving average. Correctness is ensured by asserting the feature vectors' shapes, as well as starting and ending elements.

## 6.2TOOL USED FOR TESTING:

Various tools have been used to assist in the development of the mobile application. In particular , Chrome Mobile Emulator is used to simulate the mobile view while developing the mobile application on desktop/laptop computers. After the application is deployed to the cloud, mobile phones with different operating systems and browsers, including Google Pixel running Android 9 (Google Chrome) and iPhone 7 running iOS 12.1 (Safari), are used to verify the user experience is consistent across different devices with different resolutions .

# CHAPTER 7

## RESULT AND DISCUSSION

**RESULT:**

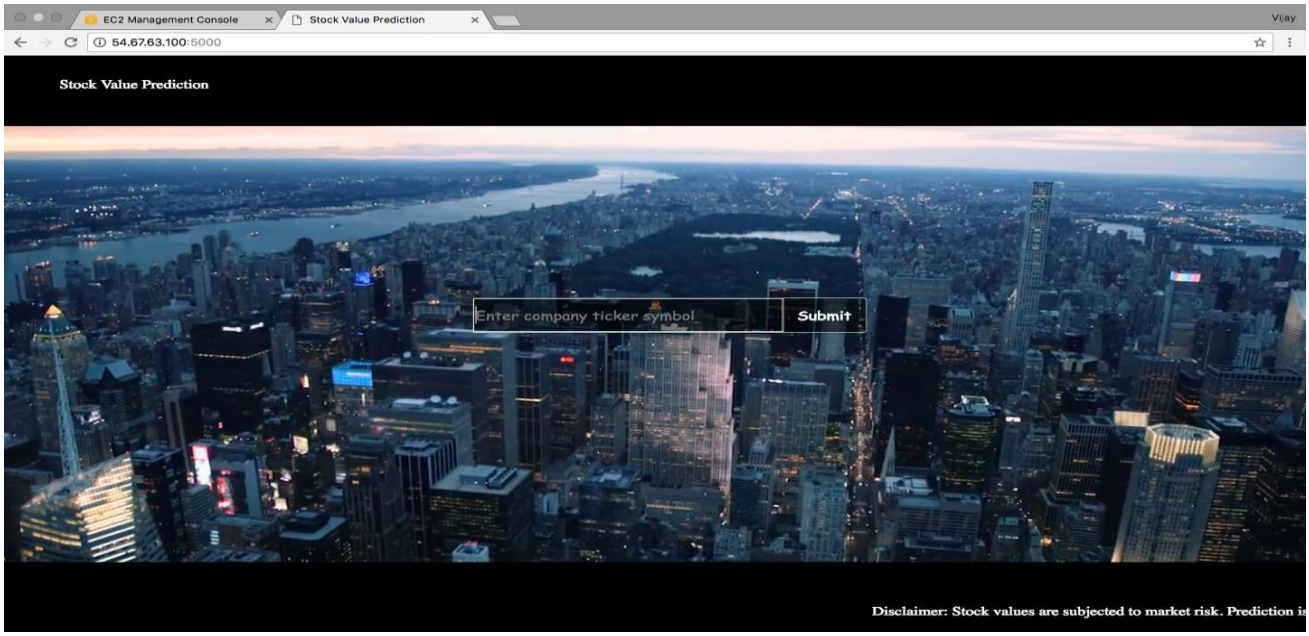



**FIG 7.1 HOME PAGE-LOGIN PAGE**

This is the login page of the project . Enter the http link in the google chrome ,then the login page will be appear .Enter the company tricker symbol and submit by clicking submit button


**FIG 7.2 GRAPH**

After submitting the company tricker symbol, it display the prediction graph of the stock market.

## DISCUSSION:

### 7.1 ACCURACY OF STOCK PRICE PREDICTION:

while the 1-day stock price prediction follow closely with actual stock prices, the predictions for stock prices after 10 days deviate considerably from the actual stock prices. This shows that machine learning models fail to provide accurate stock price predictions to retail investors.

### 7.2 DEMOCRATIZATION OF MACHINE LEARNING:

An other factor when evaluating the project's success is when there investors can use and understand the predictive information provided by the machine learning technologies using our mobile application . In spite of the confusions found in some parts of the user interface ,especially in the advanced user mode, users found useful insights provided by the machine learning models, such as identifying stocks with upside potential .The result is significant, in the sense that users with little background on machine learning technology and stock trading could find potential use cases for the application .The result simply that machine learning technologies could be democratized to serve the interest of the general public. Stock price prediction is a particularly exciting area, because the level of expertise required to succeed in making profitable short-term investments is considered to be prohibitive for small , retail investors , and trading with help of machine learning is a feat only institutional investors could perform. The application demonstrates one possible way retail investors could use machine learning technologies on their own.

## 8 .CONCLUSION AND FUTUER WORK:

The project analysis the foundation for democratizing machine learning technologies for retail investors, connecting predictions made by machine learning model store tail investors through a mobile application. It helps investors navigate through the stock markets with additional analysis and help them make more informed decisions. The findings demonstrated that the application provides significance in trend prediction. When compared to the baseline, the prediction shows useful trend tendency with the real stock trend. Through the application interface, the user can easily compare the predictions and model scores from different machine learning models, then choosing the one that fits their preference. The models used in the application will continue to improve itself by searching for a better model topology, structure and hyper parameters through evolution algorithm. The findings concluded the usefulness of evolution algorithm in lowering the mean squared error when predicting stock prices, which is helpful for improving the trend prediction for retail investors. Therefore, with the application and research findings, to large extent the project team achieved the aim of creating an user-friendly system for retail investors whom does not have previous technical knowledge to navigate the machine model predictions result with useful benchmarks.

There are 4 possible further improvements building upon the findings of this project. First, multiple approach test of raming the problems could be explained ,such as Predicting Whether the stock price goes upon down (binary classification

based on the previous stock prices. Other features could being corporated, such as market news and sentiment .Combined with the development of more advanced machine learning techniques, the accuracy of the information provided to retail investors might be improved significantly. Second, a larger scale of evolution with larger population size and more iterations could also be tested for achieving better results. Model inputs can also be included into the evolution algorithm as a variable to optimize. Regularized evolution [38] can be tested to eliminate old models regardless of their accuracy, which could allow the algorithm to search for more distant models in the search space. Third, it is also possible to use more finance-specific scores, like those introduced, as the objective function instead of simple mean squared errors to achieve better results. Fourth, mobile applications with better presentation of stock price predictions could be developed to help investors understand the implications of the stock price predictions, e.g. when to buy or sell. This would allow investors to make more informed decisions based on the machine learning models and truly democratize machine learning technologies, which were believed to be only in the hands of very few people.

## REFERNCE


[1] “Survey Finds Hong Kong Securities Market Attracts Wide Range of Investors,” *HKEX*, 13 Jul2017;http://www.hkex.com.hk/news/news-release/2017/170713news?sc_lang=en. Y. Dai and Y. Zhang, “Machine Learning in Stock Price Trend Forecasting,” *Stan ford University*; http://cs229.stanford.edu/proj2013/DaiZhang-MachineLearningInStockPriceTrendForecasting.pdf.

[2] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, “Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques, ”*Expert Systems with Applications: An International Journal,* Vol. 42, Jan. 2015, pp. 259-268

[3]B. Wanjawa and L. Muchemi, “ANN Model to Predict Stock Prices at Stock Exchange Markets,” arXiv:1502.06434 [q-fin.ST], 2014

[4] D. Mandic and J. Chambers, *Recurrent Neural Networks for Prediction*, Wiley, 2001

[5] R. Williams and D. Zipser, “Gradient-based learning algorithms for recurrent networksandtheircomputationalcomplexity”,in*Back-propagation:Theory,ArchitecturesandAppli-Cations*, Hillsdale, NJ: Erlbaum, 1992, pp. 433 - 486

[6] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, “Gradient flow in recurrent nets: the difficulty of learning long-term dependencies”, in *A Field Guide to Dynamical Recurrent Neural Networks*, S. C. Kremer and J. F. Kolen, eds., IEEE press, 2001

[7] S. Hochreiter and J. Schmidhuber, “Long Short-term Memory”,*Neural Computation*, vol. 9,no. 8, pp. 1735 - 1780, 1997

[8] K. Cho et al., “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation” , arXiv:1406. 1078 [cs.CL], 2014.

[9] W. Gail, G. Yoav, and Y. Eran, "On the Practical Computational Power of Finite PrecisionRNNs for Language Recognition", arXiv:1805.04908 [cs.NE], 2018

[10] J. Chung, C. Gulcehre, K. Cho and Y. Bengio, "Empirical Evaluation of Gated RecurrentNeural Networks on Sequence Modeling". arXiv:1412.3555 [cs.NE]. 2014

[11] E. Real, et al. “Large-Scale Evolution of Image Classifiers,”arXiv:1703.01041

[cs.NE]. Jun2017.

[12] D. Alajbeg, Z. Bubas and D. Vasic, “Price Distance To Moving Averages And SubsequentReturns”, *International Journal of Economics, Commerce and Management*, Vol. V, Dec 2017,pp. 33 - 47

[13] Progressive Web Apps, Google. Available:https://developers.google.com/web/progressive-web-apps/

[14] Neoteric, “Single-page application vs. multiple-page application”, Medium. 2016. Available:https://medium.com/@NeotericEU/single-page-application-vs-multiple-page-application-2591588efe58

[15] Keras: The Python Deep Learning library, Keras. Available:https://keras.io/.

[16] Documentation of scikit-learn 0.19.2. Available:http://scikit-learn.org/stable/documentation.html.

[17] Alpha Vantage API Documentation, Alpha Vantage. Availablehttps://www.alphavantage.co/documenT/.

NumPy v1.14 Manual, The SciPy community. Available: https://docs.scipy.org/doc/numpy-1.14.5/

[18]pandas: powerful Python data analysis toolkit, Pandas. Available:http://pandas.pydata.org/pandas-docs/version/0.23/.

[20]Hello, Colaboratory - Colaboratory - Google, Google. Available :https://colab.research.google.com.

[21]Tensor Board: Visualizing Learning, Google. Available:https://www.tensorflow.org/guide/summaries_and_tensorboard.

[22] Welcome to Flask — Flask 1.0.2 documentation, Pallets Team. Available :http://flask.pocoo.org/docs/1.0/.

[23] Cloud Storage, Google. Available: https://firebase.google.com/docs/storage/.

[24] Getting Started, The Investors Exchange. Available :https://iextrading.com/developer/docs/.

[25] Cloud Functions for Firebase, Google. Available:https://firebase.google.com/docs/function/.

[26] Cloud Fire store, Google. Available: https://firebase.google.com/docs/firestore/.

[27] React – A JavaScript library for building user interfaces, Facebook Inc;https://reactjs.org/.

[28] React Router: Declarative Routing for React.js, React Training. Available:https://reacttraining.com/react-router/web/guides/philosophy.

[29] Read Me - Redux, Redux. Available: https://redux.js.org/.

[30]Immutable collections for JavaScript, Facebook Inc. Available:https://github.com/ face book/immutable-j s/.

[31] Introduction - Material Design, Google. Available :https://material.io/design/introduction/.

[32] Material UI, Material UI Team. Available :https://material-ui.com/

## APPENDIX

## A. SOURCE CODE:

**Predict stock:**

```
import os
import sys
import requests
import numpy as np
from keras.models import Sequential
from keras.layers import Dense

FILE_NAME = 'historical.csv'

def get_historical(quote):
    # Download our file from google finance
    url =
'http://www.google.com/finance/historical?q=NASDAQ
%3A'+quote+'&output=csv'
    r = requests.get(url, stream=True)

    if r.status_code != 400:
        with open(FILE_NAME, 'wb') as f:
            for chunk in r:
                f.write(chunk)

        return True

def stock_prediction():

    # Collect data points from csv
    dataset = []
```

```
    with open(FILE_NAME) as f:
        for n, line in enumerate(f):
            if n != 0:
                str = line.split(',')[1]
                if str != "-":
                    dataset.append(float(line.split(',')[1]))

    dataset = np.array(dataset)

    # Create dataset matrix (X=t and Y=t+1)
    def create_dataset(dataset):
        dataX = [dataset[n+1] for n in range(len(dataset)2)]
        return np.array(dataX), dataset[2:]

    trainX, trainY = create_dataset(dataset)

    # Create and fit Multilinear Perceptron model
    model = Sequential()
    model.add(Dense(8, input_dim=1, activation='relu'))
    model.add(Dense(1))

model.compile(loss='mean_squared_error',optimizer=
adam')
 model.fit(trainX, trainY, nb_epoch=200, batch_size=2,
verbose=2)

    # Our prediction for tomorrow
    prediction = model.predict(np.array([dataset[0]]))
    result = 'The price will move from %s to %s' %
(dataset[0], prediction[0][0])
```

```
    return result

# Ask user for a stock quote
stock_ quote = input('Enter a stock quote from NASDAQ (e.j: AAPL, FB, GOOGL): ').upper()

# Check if we got the historical data
if not get_ historical(stock_ quote):
    print ('Google returned a 404, please re-run the script and')
    print ('enter a valid stock quote from NASDAQ')
    sys.exit()

# We have our file so we create the neural net and get the prediction
print (stock _prediction())

# We are done so we delete the csv file os.remove (FILE_NAME)
```

**Prophet:**

```
import pandas as pd
import numpy as np
import pandas_datareader.data as web
from fbprophet import Prophet
import datetime
from flask import Flask, render_template
from flask import request, redirect
from pathlib import Path
import os
import os.path
import csv
from itertools import zip_longest

app = Flask(__name__)

@app.after_request
def add_header(response):
    """
    Add headers to both force latest IE rendering engine or Chrome Frame,
    and also to cache the rendered page for 10 minutes.
    """
    response.headers['X-UA-Compatible'] = 'IE=Edge,chrome=1'
    response.headers['Cache-Control'] = 'public, max-age=0'
    return response

@app.route("/")
def first_page():
    """
    original_end = 175
    forecast_start = 200
```

```
    stock = "IBM"
    return render_template("plot.html", original = original_end, forecast = forecast_start,
stock_tinker = stock)
    """
    tmp = Path("static/prophet.png")
    tmp_csv = Path("static/numbers.csv")
    if tmp.is_file():
        os.remove(tmp)
    if tmp_csv.is_file():
        os.remove(tmp_csv)
    return render_template("index.html")

#function to get stock data
def yahoo_stocks(symbol, start, end):
    return web.DataReader(symbol, 'yahoo', start, end)

def get_historical_stock_price(stock):
    print ("Getting historical stock prices for stock ", stock)

    #get 7 year stock data for Apple
    startDate = datetime.datetime(2010, 1, 4)
    #date = datetime.datetime.now().date()
    #endDate = pd.to_datetime(date)
    endDate = datetime.datetime(2017, 11, 28)
    stockData = yahoo_stocks(stock, startDate, endDate)
    return stockData

@app.route("/plot" , methods = ['POST', 'GET'] )
def main():
    if request.method == 'POST':
        stock = request.form['companyname']
```

```
df_whole = get_historical_stock_price(stock)

df = df_whole.filter(['Close'])

df['ds'] = df.index
#log transform the ‘Close’ variable to convert non-stationary data to stationary.
df['y'] = np.log(df['Close'])
original_end = df['Close'][-1]

model = Prophet()
model.fit(df)

#num_days = int(input("Enter no of days to predict stock price for: "))

num_days = 10
future = model.make_future_dataframe(periods=num_days)
forecast = model.predict(future)

print (forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail())

#Prophet plots the observed values of our time series (the black dots), the forecasted values (blue line) and
#the uncertainty intervalsof our forecasts (the blue shaded regions).

#forecast_plot = model.plot(forecast)
#forecast_plot.show()

#make the vizualization a little better to understand
df.set_index('ds', inplace=True)
forecast.set_index('ds', inplace=True)
#date = df['ds'].tail(plot_num)
```

```
        viz_df = df.join(forecast[['yhat', 'yhat_lower','yhat_upper']], how = 'outer')
        viz_df['yhat_scaled'] = np.exp(viz_df['yhat'])

        #close_data = viz_df.Close.tail(plot_num)
        #forecasted_data = viz_df.yhat_scaled.tail(plot_num)
        #date = future['ds'].tail(num_days+plot_num)

        close_data = viz_df.Close
        forecasted_data = viz_df.yhat_scaled
        date = future['ds']
        #date = viz_df.index[-plot_num:-1]
        forecast_start = forecasted_data[-num_days]

        d = [date, close_data, forecasted_data]
        export_data = zip_longest(*d, fillvalue = '')
        with open('static/numbers.csv', 'w', encoding="ISO-8859-1", newline='') as myfile:
            wr = csv.writer(myfile)
            wr.writerow(("Date", "Actual", "Forecasted"))
            wr.writerows(export_data)
        myfile.close()

        return render_template("plot.html", original = round(original_end,2), forecast =
round(forecast_start,2), stock_tinker = stock.upper())
'''
if __name__ == "__main__":
    main()
'''

if __name__ == "__main__":
    app.run(debug=True, threaded=True
```

**prophet without flask:**

```
import pandas as pd
import numpy as np
import pandas_datareader.data as web
from fbprophet import Prophet
import datetime
import matplotlib.pyplot as plt

plt.rcParams['figure.figsize']=(20,10)
plt.style.use('ggplot')

#function to get stock data
def yahoo_stocks(symbol, start, end):
    return web.DataReader(symbol, 'yahoo', start, end)

def get_historical_stock_price(stock):
    print ("Getting historical stock prices for stock ", stock)

    #get 7 year stock data for Apple
    startDate = datetime.datetime(2010, 1, 4)
   # date = datetime.datetime.now().date()
   # endDate = pd.to_datetime(date)
    endDate = datetime.datetime(2017, 11, 27)
    stockData = yahoo_stocks(stock, startDate, endDate)
    return stockData

def main():
```

```
stock = input("Enter stock name(ex:GOOGL, AAPL): ")
df_whole = get_historical_stock_price(stock)

df = df_whole.filter(['Close'])

df['ds'] = df.index
#log transform the ‘Close’ variable to convert non-stationary data to stationary.
df['y'] = np.log(df['Close'])

model = Prophet()
model.fit(df)

num_days = int(input("Enter no of days to predict stock price for: "))

future = model.make_future_dataframe(periods=num_days)
forecast = model.predict(future)

print (forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail())

#Prophet plots the observed values of our time series (the black dots), the forecasted values (blue line) and
#the uncertainty intervalsof our forecasts (the blue shaded regions).
forecast_plot = model.plot(forecast)
forecast_plot.show()

#make the vizualization a little better to understand
df.set_index('ds', inplace=True)
forecast.set_index('ds', inplace=True)

viz_df = df.join(forecast[['yhat', 'yhat_lower','yhat_upper']], how = 'outer')
viz_df['yhat_scaled'] = np.exp(viz_df['yhat'])
```

```
    fig = plt.figure()
    ax1 = fig.add_subplot(111)
#    ax1.xaxis_date()
    ax1.plot(viz_df.index, viz_df.Close)
    ax1.plot(viz_df.index, viz_df.yhat_scaled, linestyle=':')
    ax1.set_title('Actual Close (Orange) vs Close Forecast (Black)')
    ax1.set_ylabel('Closing Price in Dollars')
    ax1.set_xlabel('Date')

    L = ax1.legend() #get the legend
    L.get_texts()[0].set_text('Actual Close') #change the legend text for 1st plot
    L.get_texts()[1].set_text('Forecasted Close') #change the legend text for 2nd plot

    plt.savefig('graph/prophet.png', bbox_inches='tight')
    plt.show()

    #plot using dataframe's plot function
    viz_df['Actual Close'] = viz_df['Close']
    viz_df['Forecasted Close'] = viz_df['yhat_scaled']

    viz_df[['Actual Close', 'Forecasted Close']].plot()

if __name__ == "__main__":
    main()
```

**INDEX HTML:**

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title> Stock Value Prediction </title>
      <link rel="stylesheet" href="../static/style.css">
   </head>

   <video plays in line auto play muted loop id="bgvid">
      <source  src="{{ url _for('static', filename='video/video.mp4') }}" type="video/mp4">
   </video>
   <body>
      <div >
         <form action=http://localhost:5000/plot method = "POST">
            <input type="text" name="company name" placeholder="Enter company ticker
symbol" class = "search box">
            <input type="submit" name="Search"  class="submit">
         </form>
       <a href="/" class = "homepage">Stock Value Prediction</a>
            <marquee class = "bottom bar">Stock values are subjected to market risk.
Prediction is based on machine learning models. Please invest at your own
risk.</marquee>
      </div>
   </body>

</html>
```

**HTML PLOT:**

```
<html>
<head>
   <meta charset="utf-8">
   <title> Prediction Results </title>
   <script src="/static/js/dygraph.js"></script>
   <link rel="stylesheet" href="../static/style_result.css">
   <link rel="stylesheet" href="/static/css/dygraph.css"/>
</head>

<body class = "bodyclass" >

   <div class="topnav" >
      <a href="/" class = "homepage" style="color:White" >Stock Value Prediction</a>
   </div>
```

```
    <h6>
            <object class = "summary" data="../static/summary/{{stock_tinker}}.txt"
style="width:85%; height:40%;" ></object>
    </h6>


    <table class="print_details" align="center" >
        <tr>

            <td>
                <font color="grey">{{ stock_tinker }}</font>
            </td>

            <td>
                <font color="grey">Last Closing Stock Value</font>
            </td>

            <td>
                <font color="grey">Predicted Stock Value</font>
            </td>

        </tr>

        <tr>
            <td></td>

            <td>
                <font color="grey">{{original}}</font>
            </td>

            <td>
                <script type="text/javascript">

                    if ({{forecast}} > {{original}})
                                    document.write("<center><b><font   color=\"green\"><font
size=\"5\">  {{forecast}}  </font><\/font></b></center>");
                    else
                                        document.write("<center><b><font   color=\"red\"><font
size=\"5\">  {{forecast}}  </font><\/font></b></center>");
                    //<font color="grey">{{forecast}}</font>

                </script>
            </td>

        </tr>
```

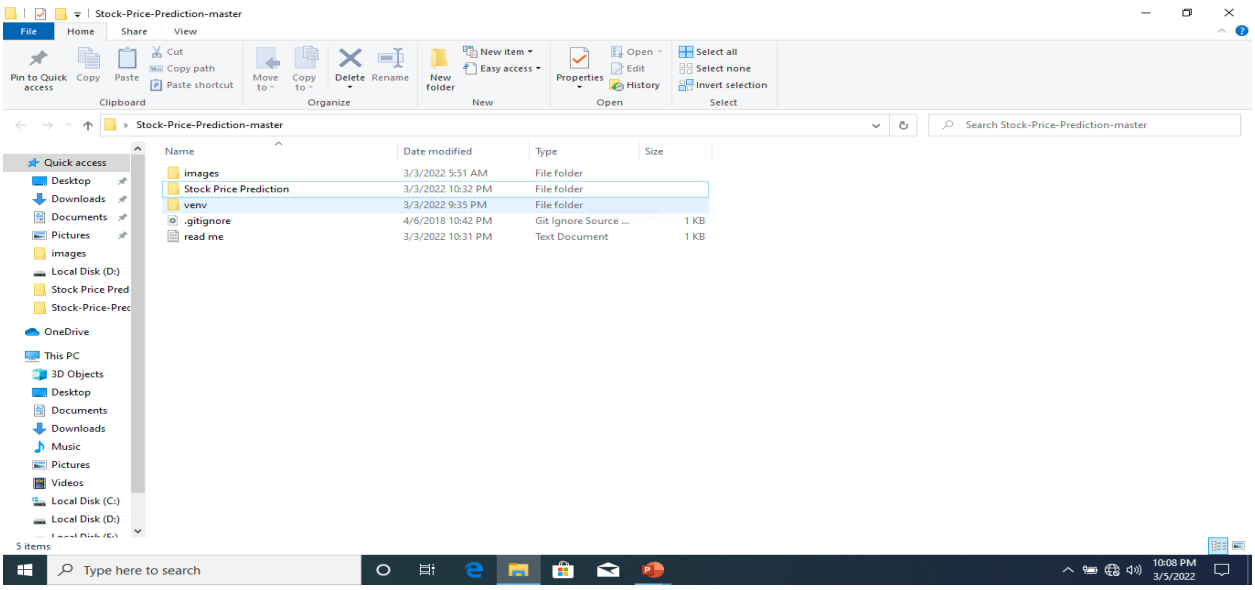
```
    </table>

    <script type="text/javascript">
        if( {{original}} - {{forecast}} < 0)
        {
                                    document.write("<center><b><font      color=\"green\"><font
size=\"5\">BUY</font><\/font></b></center>");

        }
        else
        {
            document.write("<center><b><font color=\"red\"><font size=\"5\">SELL - if stock
is owned </font><\/font></b></center>");
                    //document.write("<center><b><font  color=\"red\">Sell  -  if  stock  is
owned<\/font></b></center>");
                //document.write("<center><b><font color=\"green\">Short - if stock is not
owned<\/font></b></center>")
             document.write("<center><b><font color=\"green\"><font size=\"5\">SHORT - if
stock is not owned</font><\/font></b></center>");
        }
    </script>

    <table class="container">
        <tr>
            <td>
                <div id="graph" style="width:1000px; height:500px;" class="graph"></div>
            </td>
            <td>
              <div id="graphLegend" style="width:20px; height:3px;" class="graphbox"></div>
            </td>
        </tr>
        <tr>
            <td>
                <div class = "labels">
                    Show Data:
                    <input type=checkbox id="0" checked onClick="change(this)">
                        <label for="0"> <font color="008000">Actual</font></label>
                    <input type=checkbox id="1" checked onClick="change(this)">
                        <label for="1"> <font color="#000080">Forecasted</font></label>
                </div>
            </td>
        </tr>
    </table>

<script type="text/javascript">
    g2 = new Dygraph(
```

```
                document.getElementById("graph"),
                // CSV or path to a CSV file.
                "{{ url_for('static', filename='numbers.csv') }}", // path to CSV file
                {
                visibility: [true, true],
                labelsDiv: document.getElementById("graphLegend")
                }         // options
                );
    setStatus();

    function setStatus() {
                    document.getElementById("visibility").innerHTML =
                    g2.visibility().toString();
                }
   function change(el) {
                   g2.setVisibility(parseInt(el.id), el.checked);
                   setStatus();
                }
</script>

<div><marquee>
   <table class="newsbox" scrollamount="1" >
      <tr>
         <td>
            <div>
                  <a style="display:block;color:black; width: 200px; margin-left: 20px" font-family: "Comic Sans MS" href="https://www.cnbc.com/2017/12/03/bitcoin-hits-all-time-high-above-11700-as-recovery-accelerates.html" target="_blank" ><font color="808080"><b>Bitcoin hits all-time high above $11,700 as recovery accelerates</font color></b></a>
            </div>
         </td>
         <td>
            <div>
                     <a style="display:block;color:black; width: 200px; margin-left: 20px" href="http://beta.latimes.com/business/la-fi-markets-20171129-story.html" target="_blank" ><b><font color="808080">Tech stocks drop steeply, offsetting bank and healthcare shares' rise</font color></b></a>
            </div>
         </td>
         <td>
            <div>
                     <a style="display:block;color:black; width: 200px; margin-left: 20px" href="https://www.cnbc.com/2017/11/30/amazon-gets-its-most-bullish-call-yet-1500-price-target.html" target="_blank" ><b><font color="808080">Wall Street is giddy over Amazon shares, gets its most bullish call yet: $1,500 price target</font color></b></a>
```

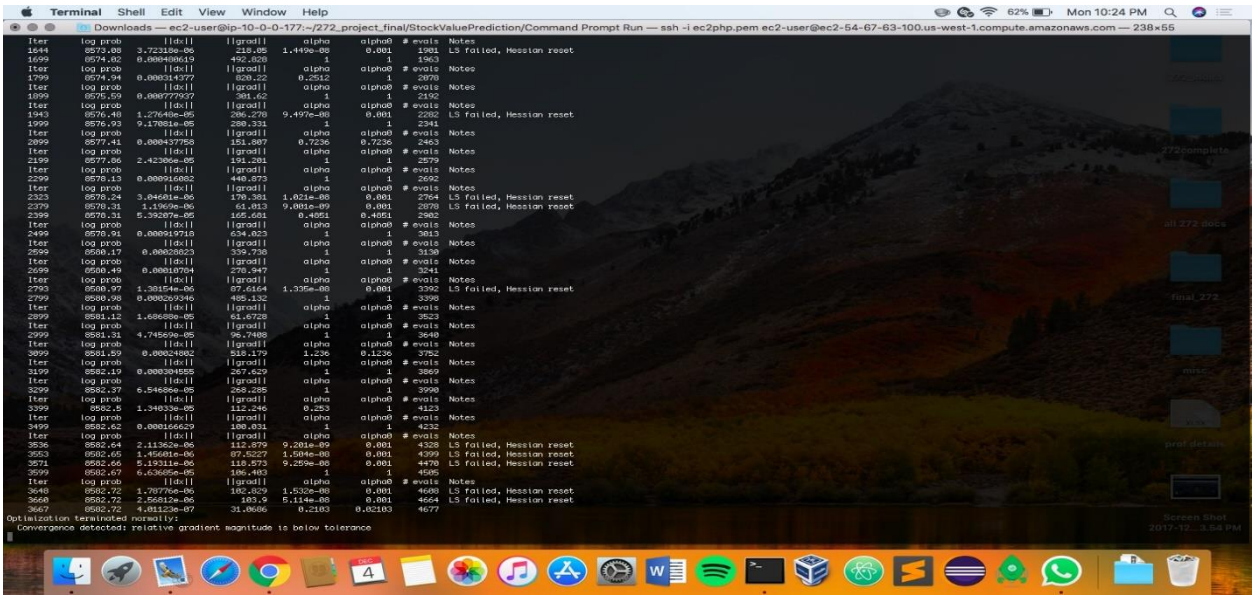
```
            </div>
         </td>
         <td>
            <div>
                  <a style="display:block;color:black; width: 200px; margin-left: 20px" href="http://fortune.com/2017/03/31/amazon-stock-trillion-dollar-company-apple-tesla-google/" target="_blank" ><b><font color="808080">Amazon and the Race to Be the First $1 Trillion Company</font color></b></a>
            </div>

         </td>
         <td>
            <div>
                  <a style="display:block;color:black; width: 200px; margin-left: 20px" href="https://www.forbes.com/sites/timworstall/2014/04/13/fun-number-apple-has-twice-as-much-cash-as-the-us-government/#790e34d75570" target="_blank" ><b><font color="808080">Apple Has Twice As Much Cash As The US Government</font color></b></a>
            </div>

         </td>
      </tr>
   </table>
   </marquee>
   <marquee direction = "right">
   <table class="newsbox" scrollamount="1">
      <tr>
         <td>
            <div>
                  <a style="display:block;color:black; width: 200px; margin-left: 20px" href="http://www.latimes.com/business/hiltzik/la-fi-hiltzik-cvs-aetna-20171204-story.html" target="_blank" ><b><font color="808080">CVS and Aetna say their huge merger will be great for consumers. Here's why you should be skeptical</font color></b></a>
            </div>
         </td>
         <td>
            <div>
                  <a style="display:block;color:black; width: 200px; margin-left: 20px" href="https://www.cnbc.com/2017/12/01/amid-tax-bill-euphoria-one-big-part-of-the-stock-market-is-getting-left-behind-tech.html" target="_blank" ><b><font color="808080">Amid tax-bill euphoria, one big part of the stock market is getting left behind: Tech</font color></b></a>
            </div>
         </td>
         <td>
```

```
            <div>
                    <a style="display:block;color:black; width: 200px; margin-left: 20px" href="https://www.nytimes.com/2017/12/04/business/dealbook/broadcom-qualcomm.html?rref=collection%2Fsectioncollection%2Fbusiness&action=click&contentCollection=business&region=rank&module=package&version=highlights&contentPlacement=5&pgtype=sectionfront" target="_blank" ><b><font color="808080">Broadcom Puts Qualcomm on the Clock</font color></b></a>
            </div>
          </td>
          <td>
            <div>
                    <a style="display:block;color:black; width: 200px; margin-left: 20px" href="https://www.reuters.com/article/us-usa-stocks/dow-hits-record-as-investors-bet-on-tax-cuts-idUSKBN1DY1EP" target="_blank" ><b><font color="808080">Dow hits record as investors bet on tax cuts</font color></b></a>
            </div>
          </td>
          <td>
            <div>
                    <a style="display:block;color:black; width: 200px; margin-left: 20px" href="http://www.businessinsider.com/meet-the-paypal-mafia-the-richest-group-of-men-in-silicon-valley-2014-9" target="_blank" ><b><font color="808080">Meet The PayPal Mafia, The Richest Group Of Men In Silicon Valley</font color></b></a>
            </div>
          </td>
        </tr>
    </table>
    </marquee>
</div>

</body>
</html>
```

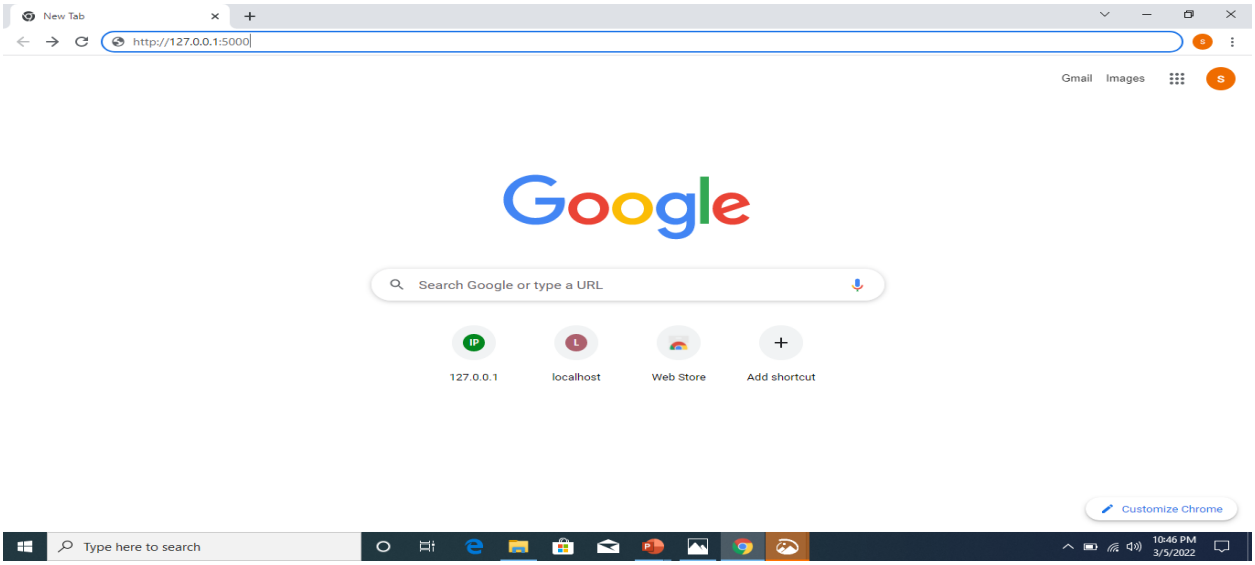
## B .OUTPUT:



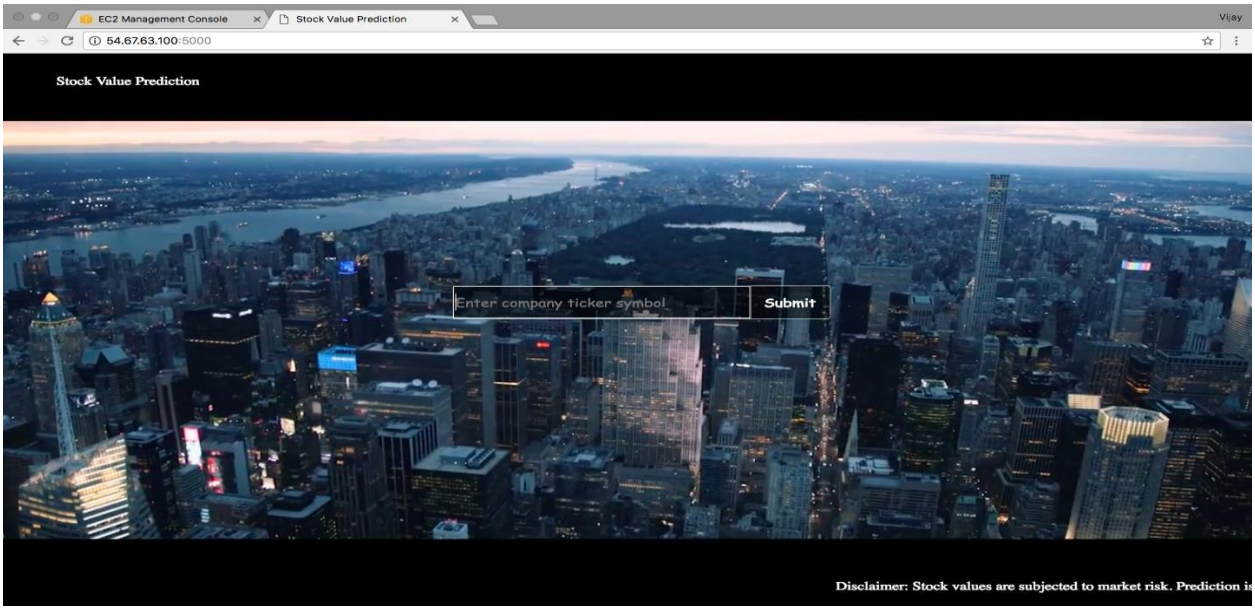

### B.1 SERVING FOR STOCK MARKET PREDICTION




### B.2 RUNNING ON HTTP




## B.3 OPEN THE LINK IN THE BROWSER




## B.4 HOME PAGE OR LOGIN PAGE


## B.5 OUTPUT OF THE GRAPH