WEB APPLICATION FOR PREDICTING FEATURE STOCK PRICE

Nidrabingi Krishna Veni

Student
Andhra University

Abstract- **The stock market is a well-known investment choice that has an impact on the current economy. It will be an investor's dream to correctly forecast the rise and fall due to the large returns and losses. However, future prices are incredibly uncertain. Although other analyses, such as fundamental analysis and technical analysis, have been around for years. Different algorithms are now employed to predict the price in the future. To calculate the longer-term share prices, the forecast of stock value is a challenging process that requires a reliable algorithm to run in the background. Using machine learning, Due to the structure of the market, stock prices are connected, making it challenging to estimate costs. The suggested methods employ machine learning to forecast share prices using market data The suggested algorithms use market data to forecast share prices using machine learning techniques such recurrent neural networks called Long Short-Term Memory. Stochastic Gradient Descent is used in this process to correct weights for each data point. In contrast to the stock price predictor algorithms that are now accessible, our system will produce accurate results. To drive the graphical results, the network is trained and assessed with a range of input data sizes. The project's main goal is to anticipate feature stock values using machine learning techniques based on linear regression and LSTM. Open, close, low, high, and volume are all factors.**

INTRODUCTION

Numerous factors influence stock price changes. It is challenging to determine the variables influencing supply and demand. It is feasible to observe the present supply and demand, but it is incredibly challenging to pinpoint the causes of changes in supply and demand. The majority of the elements are socioeconomic, including market movement, trends, news, etc. Every investor wants to make money, thus their goal is to precisely predict the stock price. Trading on the stock market involves millions of rupees every day. Investors must examine the stock chart, stock market indices, and corporate news in order to make a profit. The manual analysis of every one of the aforementioned factors is very challenging. the initial public offering of stocks IPO stands for initial public offering. Government policy, social media, and news all have a big impact on the stock market. Therefore, for more precise price prediction, news and social media must be taken into account in addition to technical analysis and fundamental analysis. Future price prediction is now more precise and efficient because to several techniques like Artificial Neural Networks (ANN) and deep learning (DL). In this project, our primary goal is to effectively anticipate these erratic stock values using DL technique and to comprehend the impact of the number of model training epochs. Using Python, LSTM, and Linear Regression models, predictive analysis is utilized to forecast the future trading price of five equities traded on the National Stock Exchange (NSE).

Problem definition

The basic definition of stock market prediction is the process of attempting to calculate a stock's value and provide a solid framework for understanding and forecasting the market and stock prices. Typically, it is displayed using the dataset's quarterly financial ratio. In light of this, relying solely on one dataset for the prediction may not be adequate and may produce erroneous results. As a result, we are considering using machine learning research to integrate many datasets and forecast market and stock movements. If a better stock market prediction algorithm is not put forth, the issue with estimating the stock price will continue to exist. . It might be challenging to forecast how the stock market will fare. Thousands of investors' opinions typically influence the direction of the stock market. An aptitude for predicting how recent events will affect investors is necessary for stock market forecasting. These occurrences may be political, such as a political leader's speech or news of a swindle. It may also be a global event, such as a sudden change in the value of a currency or a commodity, etc. All of these things have an impact on business earnings, which in turn have an impact on investor mood. Almost no investor has the capacity to regularly and correctly predict these hyper-parameters. It is quite difficult to estimate stock price because of all these aspects. Once the proper information is gathered, it then utilized to get a forecast outcome by training a machine.

Literature review

[2]Shruti Goswami, A deep learning (DL) model is utilized in this instance for Long-Short-Term Memory. The price of the needed stock was found to be accurately predicted by the DL model. Using mean absolute percentage error, the model's performance is quantified.

[3]**Ishita Parmar et** al The goal of stock market prediction is to forecast the value of a company's financial stocks in the future. The use of machine learning, which produces forecasts based on the values of current stock market indices by training on their prior values, is a recent trend in stock market prediction technologies. Multiple models are used by machine learning itself to facilitate and authenticate prediction. The paper focuses on LSTM-based machine learning and regression for stock value prediction. Open, close, low, high, and volume are all factors.

[5]Deeksha Chandola et al The long short-term memory (LSTM) algorithm and the Word2Vec technique are combined to create a hybrid deep learning model in this work. The main goal is to create a smart tool to predict the direction of price movement on the stock market using financial time series and news headlines as inputs. Investors could make better selections with the help of the binary projected output produced by the suggested model. Accuracy in predicting the direction of movement of the stock prices of five businesses from various industries is used to measure the effectiveness of the suggested model.

Existing system

Traditional methods for analyzing the stock market and predicting stock prices include fundamental analysis, which considers a stock's performance in the past and the overall credibility of the company, and statistical analysis, which is only interested in crunching numbers and finding patterns in stock price variation. The latter is frequently accomplished using Artificial Neural Networks (ANNs) or Genetic Algorithms (GA), however these algorithms are unable to capture correlation between stock prices in the form of long-term temporal dependence. Another significant problem with utilizing straightforward ANNs for stock prediction is the phenomena of exploding / vanishing gradient, in which the weights of a big network either grow too large or too little (respectively), greatly slowing their convergence to the ideal value. Usually, two things contribute to this: Weights are initialized at random, and they also change significantly more toward the end of the network than they do at the beginning. Feature selection algorithms can be used to narrow down a core set of features (such as GDP, oil price, inflation rate, etc.) that have the biggest effects on stock prices or currency exchange rates across markets. This is an alternative method for stock market analysis. However, because it neglects to take into account the whole history of trends and because there is no provision for outlier detection, this method does not take into account long-term trading strategies.

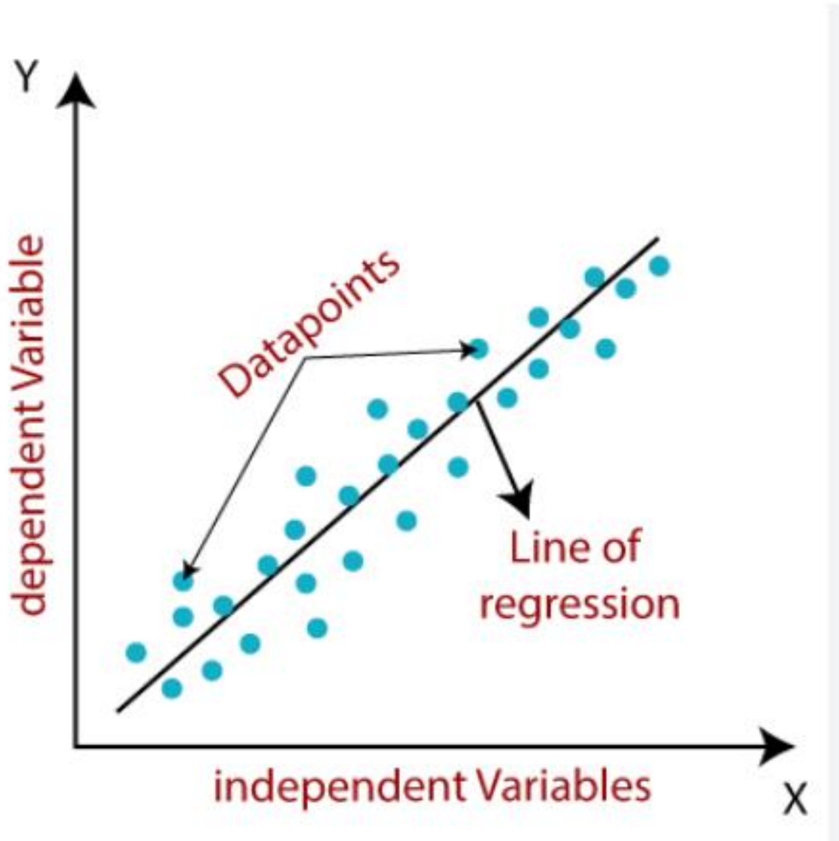
Proposed system

The user can enter the stock symbol and choose the desired model, and the web app will display the anticipated stock prices and trends for the upcoming period. Stock Price/Trend Prediction Web Application This is a web application that provides stock price and trend prediction using the FB Prophet tool, LSTM, and Linear Regression. Using machine learning algorithms like Linear Regression and Long Short-Term Memory (LSTM), the suggested system primarily focuses on predicting stock values. I created the "Stock Price Prediction Using Web Application" system, and I used the LSTM algorithm to anticipate stock market prices. We were able to train the computer in this project's suggested approach using multiple historical data points to anticipate the future. To train the algorithm, I used information from stocks from the preceding year. To overcome the challenge, two machine-learning libraries were mostly employed. The first one was NumPy, which was used to prepare the data for analysis by cleaning and manipulating it using statistical techniques. Scikit, on the other hand, was employed for accurate analysis and forecasting. The data set we used was taken from the public online database of stock market data from past years, with 80% of the data used to train the machine and the remaining 20% used to test the data. The supervised learning model's fundamental methodology is discovering patterns and correlations in the data from the training set and then reproducing them in the test set. We processed the data using the Python pandas module, which aggregated many datasets into a data frame. We were able to get the data ready for feature extraction thanks to the optimized data-frame. Date and a specific day's closing price were the data-frame features. With the help of all these features, we trained the computer to predict the object variable—the price for a certain day—using linear regression and LSTM. Using the test set predictions and the actual data, we also calculated the accuracy. The suggested system makes use of various research fields, including data pre-processing LSTM.

Methodology

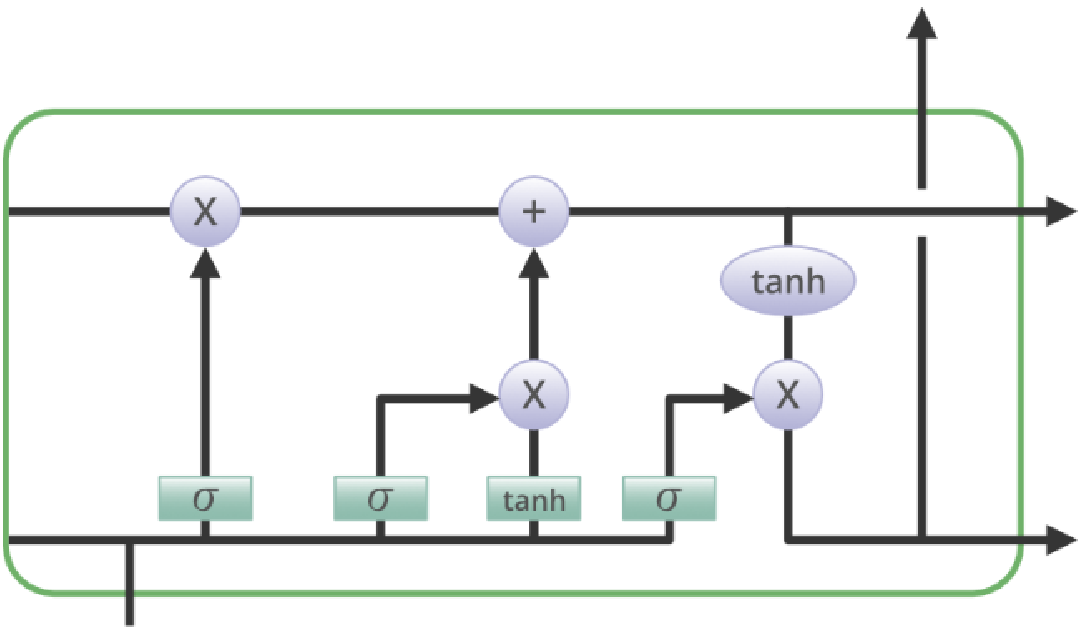
Linear Regression

To find a correlation between historical stock prices and predictor variables like volume, market trends, or other important variables, you can use linear regression. This will enable you to forecast future stock prices using these factors. In order to forecast the result of upcoming events, a linear relationship between an independent variable and a dependent variable is provided by the linear regression algorithm. It is a statistical technique used in predictive analysis in data science and machine learning.



LSTM(Long Short-Term Memory)

we can utilize LSTM (Long Short-Term Memory) models for patterns and time dependencies that are more complicated. Recurrent neural networks with the ability to capture long-term dependencies in sequential data, such as stock price prediction, are known as LSTM models. Recurrent neural networks (RNNs) with Long Short-Term Memory (LSTM) are especially made to handle sequential data, including time series, speech, and text. LSTM networks are particularly suited for applications like language translation, speech recognition, and time series forecasting because they can learn long-term dependencies in sequential data. The input gate controls what information is added to the memory cell. The forget gate controls what information is removed from the memory cell. And the output gate controls what information is output from the memory cell. This allows LSTM networks to selectively retain or discard information as it flows through the network, which allows them to learn long-term dependencies.



Prophet

The user can input the stock symbol and select the desired model, and the web app will display the predicted stock prices and trends for the next period. Prophet is a method for predicting time series data that uses an additive model to suit non-linear trends with seasonality that occurs annually, monthly, daily, and on weekends as well as during holidays. Strongly seasonal time series and multiple seasons of historical data are ideal for it. Prophet typically manages outliers well and is robust to missing data and changes in the trend.

Installation To run this application, you need to have Python 3 installed on your machine. You can clone this

Steps

How to build an LSTM-based Recurrent Neural Network (RNN)linear regression and to predict Google stock price step by step. It is split into 7 parts as below.

1. Problem statement
2. Data processing
3. Model building
4. Model compiling
5. Model fitting
6. Model prediction
7. Result visualization

1 Problem statement

The stock market is frequently covered in the news. Every time it reaches a new high or low, news of it is reported. If a reliable algorithm could be created to forecast the short-term price of a single stock, the rate of investment and business opportunities in the stock market may rise. artificial neural networks, which have an average error loss of 20%, have been used in previous stock prediction techniques.

2 Data processing

2.1 Import data

Dataset may be of different formats So each dataset is different from another dataset. To use the dataset in our code, we usually put it into a CSV **file**. However, sometimes, we may also need to use an HTML or xlsx file. The train/test data is saved in *.csv* files, respectively. We will use ***Open*** price for prediction. Figure 1 shows a snippet of the training set and its scatter plot.

	Date	Open	High	Low	Close	Adj Close	Volume
0	2022-01-13 00:00:00	83.5	91.35	82.7	91.35	90.7353	1,378,325
1	2022-01-14 00:00:00	91.55	99.65	88.8	96.8	96.1486	2,453,246
2	2022-01-17 00:00:00	97.7	105.5	96	102.55	101.8599	1,746,435
3	2022-01-18 00:00:00	103.3	108.25	94.2	95.3	94.6587	1,393,149
4	2022-01-19 00:00:00	93	102.7	93	100.5	99.8237	910,806
5	2022-01-20 00:00:00	100.95	103.9	94.7	99.5	98.8304	949,272
6	2022-01-21 00:00:00	99	101.8	91.25	95.35	94.7083	673,266
7	2022-01-24 00:00:00	97	100.2	88.2	90.05	89.444	632,528
8	2022-01-25 00:00:00	89.95	97	85.15	94.7	94.0627	520,109
9	2022-01-27 00:00:00	92.9	93.7	89	91.9	91.2816	336,785
10	2022-01-28 00:00:00	92.8	101	92.75	97.35	96.6949	1,058,107
11	2022-01-31 00:00:00	98.7	99.75	89.65	91.5	90.8843	594,190
12	2022-02-01 00:00:00	92.3	94.5	88.55	91.95	91.3312	303,710

2.2 Handling missing data

In any real-world dataset, there are always few null values. It doesn't really matter whether it is a regression, classification or any other kind of problem, no model can handle these NULL or NaN values on its own so we need to intervene. First of all, we need to check whether we have null values in our dataset or not. We can do that using the isnull() method. To handle missing values, we will use **Scikit-learn** library in our code, which contains various libraries for building machine learning models. Here we will use **Imputer** class of **sklearn.preprocessing** library

2.3 Handling Categorical Variables

Handling categorical variables is another integral aspect of Machine Learning. Categorical variables are basically the variables that are discrete and not continuous. Ex — color of an item is a discrete variable whereas its price is a continuous variable.
Categorical variables are further divided into 2 types:

- **Ordinal categorical variables** — These variables can be ordered. Ex — Size of a T-shirt. We can say that M<L<XL.
- **Nominal categorical variables** — These variables can't be ordered. Ex — Color of a T-shirt. We can't say that Blue<Green as it doesn't make any sense to compare the colors as they don't have any relationship.

2.4 Feature scaling

The next step is to scale the stock prices between (0, 1) to avoid intensive computation. Common methods include **Standardization** and **Normalization** as shown in Figure It is recommended to take Normalization, particularly when working on RNN with a *Sigmoid* function in the output layer.

```
From sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler(feature_range = (0, 1))
```

- Rescaling (min-max normalization)
- Mean normalization
- Standardization (Z-score Normalization)
- Scaling to unit length

2.5 Data reshaping

- As stated above, we use ***Open*** price for prediction. Namely, we only have one indicator or feature. But we can add more indicators following the same data processing methods. To do that, we need to add a new dimension for **number of indicators**. Specifically, *newshape* is in (batch size, number of timestamps, number of indicators). *(batch size, number of timestamps) is the shape of X_train*. Here we only have 1 indicator.

2.6 Splitting Data into Training, Validation and Evaluation Sets

Finally, we need to split our data into three different sets, training set to train the model, validation set to validate the accuracy of our model and finally test set to test the performance of our model on generic data. Before splitting the Dataset, it is important to shuffle the Dataset to avoid any biases. An ideal proportion to divide the Dataset is 60:20:20 i.e. 60% as the training set, 20% as test and validation set. To split the Dataset use train_test_split of sklearn.model_selection twice. Once to split the dataset into train and validation set and then to split the remaining train dataset into train and test set.

3 MODEL BUILDING

Fundamentally, we are building a NN regressor for continuous value prediction using LSTM**. First, initialize the model.**

```
regressor = Sequential()
```

Then, add the 1st LSTM layer with the **Dropout** layer followed.

```
regressor.add(LSTM(units = 50, return_sequences = True, input_shape = (X_train.shape[1], 1)))regressor.add(Dropout(rate = 0.2))
```

Note for the LSTM layer, *units* is the number of LSTM neurons in the layer. 50 neurons will give the model high dimensionality, enough to capture the upwards and downward trends. *return_sequences* is True as we need to add another LSTM layer after the current one. *input_shape* corresponds to the number of time stamps and the number of indicators. For dropout, 20% of 50 neurons will be ignored randomly during each iteration of training. Following the above same method, add 2nd, 3rd, and 4th LSTM layer.

4 Model compiling

Now, let's compile the RNN by choosing an SGD algorithm and a loss function. For optimizer, we use Adam, a safe choice to start with. The loss function is the mean of squared errors between actual values and predictions.

```
regressor.compile(optimizer = 'adam', loss = 'mean_squared_error')
```

5 Model fitting

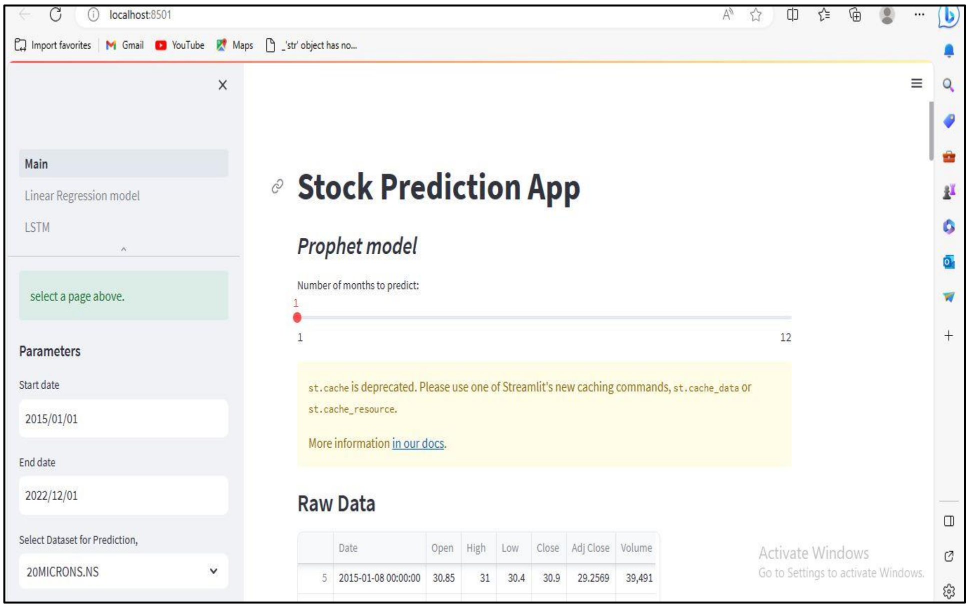
Now, let's fit our RNN.

```
regressor.fit(x = X_train, y = y_train, batch_size = 32, epochs = 100)
```

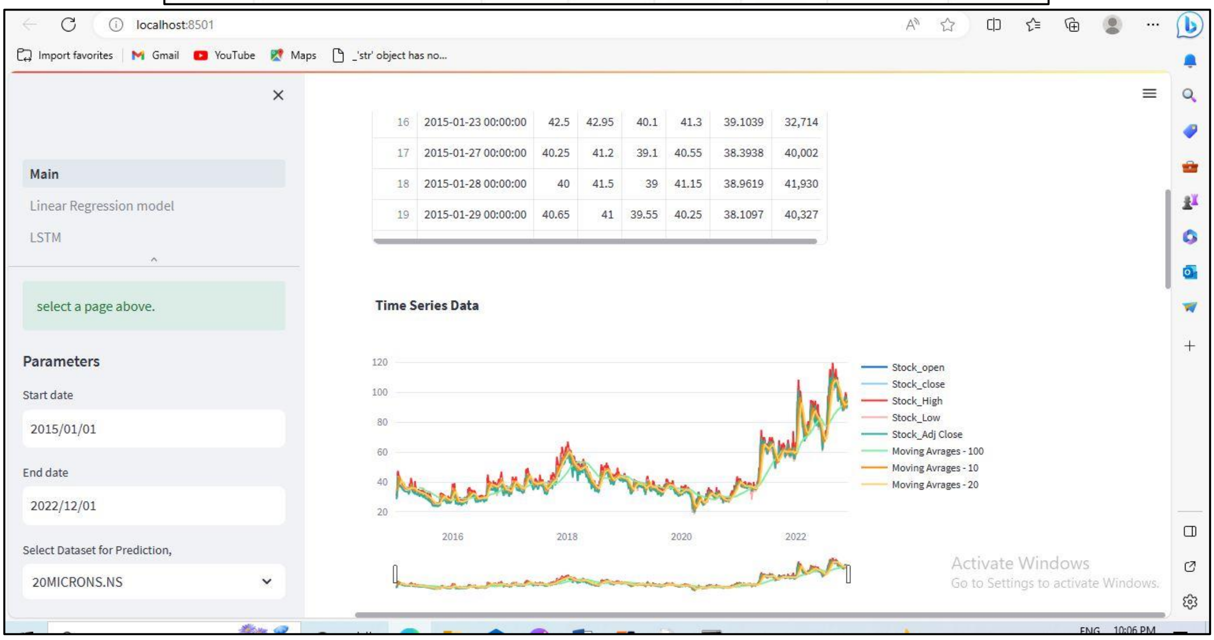
RNN weights are updated every 32 stock prices with a batch size of Feel free to try more batches and epochs if the loss of the model is not converging. Great, now let's execute the training. Model prediction by combining all the above steps we can predict the model that whether the model is correctly predicted the value or not

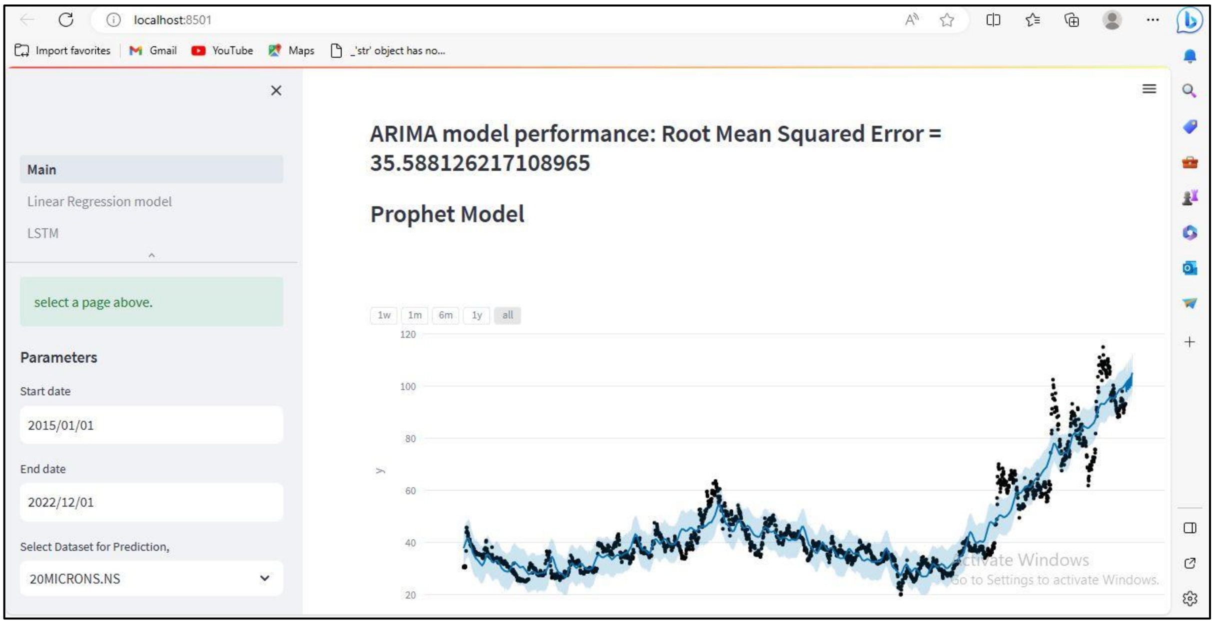
6 Result visualization

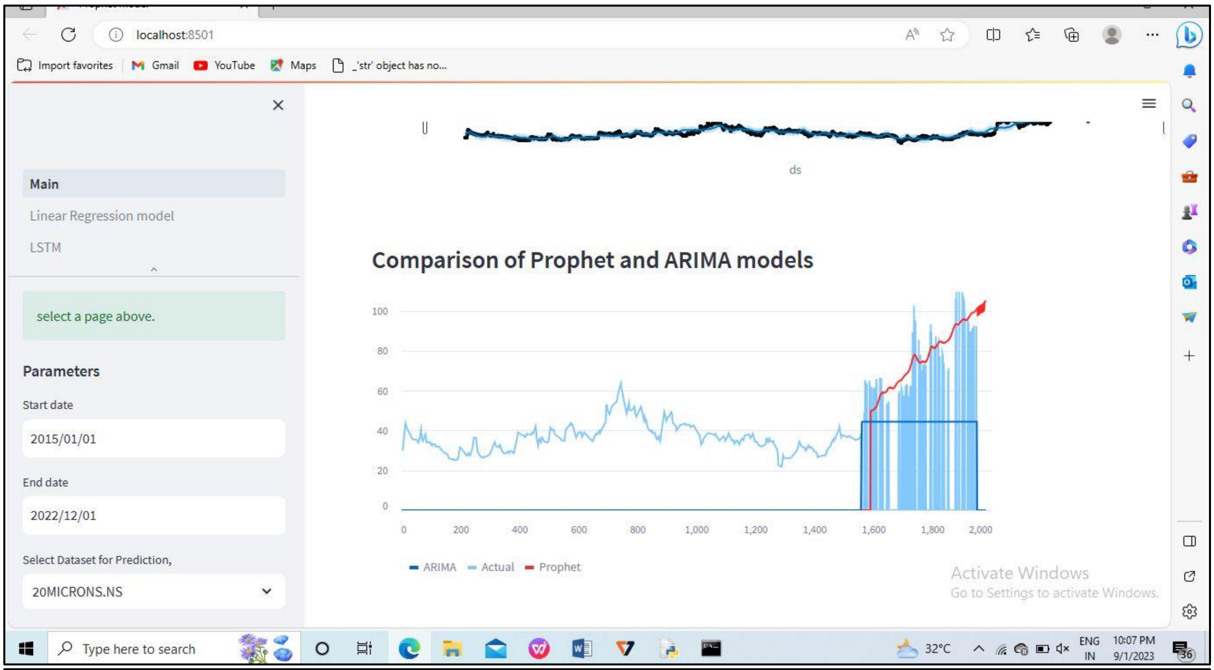
By using visualization techinquies we can visualize the data. In the graphical form

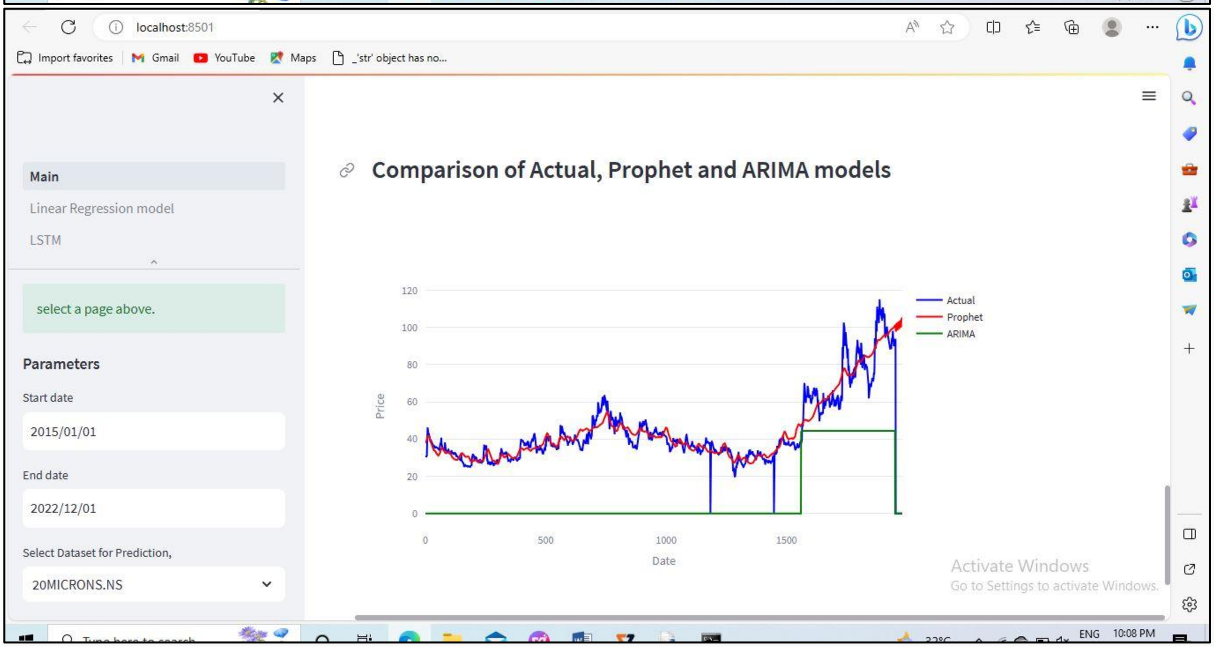


	Date	Open	High	Low	Close	Adj Close	Volume
0	2022-01-13 00:00:00	83.5	91.35	82.7	91.35	90.7353	1,378,325
1	2022-01-14 00:00:00	91.55	99.65	88.8	96.8	96.1486	2,453,246
2	2022-01-17 00:00:00	97.7	105.5	96	102.55	101.8599	1,746,435
3	2022-01-18 00:00:00	103.3	108.25	94.2	95.3	94.6587	1,393,149
4	2022-01-19 00:00:00	93	102.7	93	100.5	99.8237	910,806
5	2022-01-20 00:00:00	100.95	103.9	94.7	99.5	98.8304	949,272
6	2022-01-21 00:00:00	99	101.8	91.25	95.35	94.7083	673,266
7	2022-01-24 00:00:00	97	100.2	88.2	90.05	89.444	632,528
8	2022-01-25 00:00:00	89.95	97	85.15	94.7	94.0627	520,109
9	2022-01-27 00:00:00	92.9	93.7	89	91.9	91.2816	336,785
10	2022-01-28 00:00:00	92.8	101	92.75	97.35	96.6949	1,058,107
11	2022-01-31 00:00:00	98.7	99.75	89.65	91.5	90.8843	594,190
12	2022-02-01 00:00:00	92.3	94.5	88.55	91.95	91.3312	303,710




localhost:8501
Main
Linear Regression model
LSTM
select a page above.
Parameters
Start date
2015/01/01
End date
2022/12/01
Select Dataset for Prediction,
20MICRONS.NS
ARIMA model performance: Root Mean Squared Error = 35.588126217108965
Prophet Model
1w 1m 6m 1y all


localhost:8501
Main
Linear Regression model
LSTM
select a page above.
Parameters
Start date
2015/01/01
End date
2022/12/01
Select Dataset for Prediction,
20MICRONS.NS
ds
Comparison of Prophet and ARIMA models
ARIMA
Actual
Prophet


localhost:8501
Main
Linear Regression model
LSTM
select a page above.
Parameters
Start date
2015/01/01
End date
2022/12/01
Select Dataset for Prediction,
20MICRONS.NS
Comparison of Actual, Prophet and ARIMA models
Actual
Prophet
ARIMA
Price
Date

REFERENCES:

[1] Sonali Antad, Saloni Khandelwal, Anushka Khandelwal, Rohan Khandare, Prathamesh Khandave, Dhawal Khangar, Raj Khanke.” Stock Price Prediction Website Using Linear Regression - A Machine Learning Algorithm” ITM Web of Conferences 56, 05016 (2023) https://doi.org/10.1051/itmconf/20235605016 ICDSAC 2023

[2] Shruti Goswami; Anil Kumar Sagar; Parma Nand; Osamah Ibrahim Khalaf “Time Series Analysis Using Stacked LSTM Model for Indian Stock Market” INSPEC Accession Number: 22028729 Published in: 2022 IEEE IAS Global Conference on Emerging Technologies (GlobConET)

[3] Ishita Parmar; Navanshu Agarwal; Sheirsh Saxena; Ridam Arora; Shikhin Gupta “Stock Market Prediction Using Machine Learning” DOI: 10.1109/ICSCCC.2018.8703332 Publisher: IEEE

[4] Parag P. Kadu; G. R. Bamnote “Comparative Study of Stock Price Prediction using Machine Learning” INSPEC Accession Number: 21012993 DOI: 10.1109/ICCES51350.2021.9489170 Publisher: IEEE

[5] Deeksha Chandola, Akshit Mehta, Shikha Singh, Vinay Anand Tikkiwal, Himanshu Agrawal “Forecasting Directional Movement of Stock Prices using Deep Learning” https://link.springer.com/article/10.1007/s40745-022-00432-6