INSTITUTE FOR ADVANCED COMPUTING AND SOFTWARE DEVELOPMENT AKRUDI, PUNE

DOCUMENTATION ON

"Stock Market Price Predection"

PG-DBDA Feb-2020

Submited by:
Group No: 12
Shreyas Shivanikar(1343)
Balaji Dhumale(1316)

Mr.Prashant Karhale
Center Coordinator

Mr.Shantanu Pathak
Project Guide

Index
Table of Contents

List of figures

Chapter 1
Introduction

A huge volume of stock market price data generates in which is very dynamic in nature, which changes in every day based on various factors. We all are aware of the highly volatile financial market conditions considering the complex and challenging stock market system where gain or loss happens based on right predictions and market analysis. A correct prediction can be extremely profitable in the many cases. Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on a financial exchange. The successful prediction of a stock's future price will maximize investor's gains. Price prediction has been at focus for years since it can yield significant profits. Predicting the stock market is not a simple task, mainly due to the great number of parameters to consider like news, economy and many external factors.

1.1 **Purpose**

The goal of this project is to develop a system that can predict the upcoming price of the targeted stock that is considered as the input to the system. The project was accomplished by collecting the data from yahoo fiance TCS data from national Stock exchange and processing the data from LSTM and ARIMA model which predicts the value with the results.

1.2 Scope of the project

1.2.1 Initial functional requirement :

- Selecting appropriate algorithms.
- Determining the appropriate input format to algorithm.
- Train the model.
- Test the model.
- Display the results of both algorithm.

1.2.2 Initial nonfunctional requirement :

- Getting the data-sets which can provide model enough records to train the model.

1.3 Software Development Life Cycle Model:

In order to maintain the development of this project we are going to use SDLC.. SDLC stands for Software Development Lifecycle. SDLC is a systematic process for building software that ensures the quality and correctness of the software built. SDLC process aims to produce high-quality software that meets customer expectations. The system development should be complete in the pre-defined time frame and cost. SDLC consists of a detailed plan which explains how to plan, build, and maintain specific software. Every phase of the SDLC life cycle has its own process and deliverables that feed into the next phase.

1.3.1 SDLC Model:

The waterfall model is sequential software development process, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of conception initiation, Analysis, Design (validation), construction. Testing and maintained.

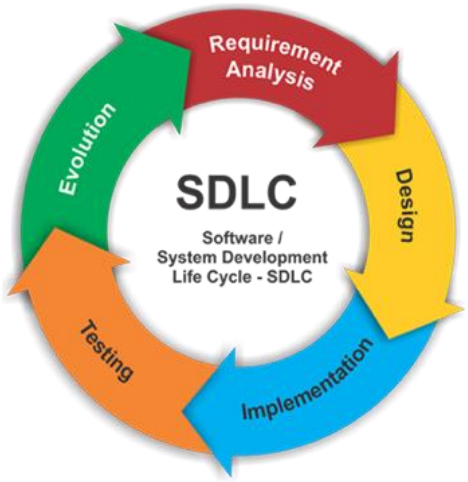


Fig 1.3.1. Software life cycle model

System Engineering and Analysis:

Because software is always a part of larger system work. Begins by establishing requirement for all system elements and Then allocating some subset of these requirement to the software system Engineering and analysis encompasses the requirement gathering at the system level with a small amount of top level design and analysis.

Software requirement Analysis:

The requirement gathering process is intensified and focused specifically on the software Requirement for the both system and software are discussed and reviewed with the customer. The customer specifies the entire requirement or the software and the function to be performed by the software.

Design:

Software design is actually a multi-step process that focuses on distinct attributes of the program data structure, software architecture, procedural detail and interface of the software that can be assessed or quality before coding begins.

Coding:

The design must be translated into a machine readable form. The coding step performs this task. If design is programmed in a detailed manner, coding can be accomplished mechanically.

Testing:

Once code has been generated programmed testing begins. The testing process focuses on the internals of the software ensuring that all statement have been tested and on the functional externals hat is conducting tests to uncover the errors and ensure that defined input will produce the results that agree with the required results.

Maintenance:

Software will undoubtedly undergo change after it is Delivered to the customer .Change will occur because errors have been encountered because the software must be able adopted to accommodate changes in its external environment because the customer requires functional or performance enhancement enhancements. The classic life cycle is the oldest and most widely used paradigm or software engineering

1.4 Overview of document

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the project. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, System analysis section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

This section describe the data and use of the models.

The forth chapter consists of requirements specification which describes the overall factors to be considered for the project and some functional and software requirements.

The fifth chapter is the main section where there is details of the complete implementation with the output and procedure.

Chapter 2
Overall Description

The stock rate prediction uses the data from yahoo fiance from data frame by selecting the appropriate stock name and range of dates.The data is transformed according to the requirements of the system. The following data is sent to the model to train it and then for prediction the test data is used. After it is done their respective RSME and R2 Score are analyzed.

2.1 Data:

In this project we selected TCS data set which is listed on National Stock Exchange in India. The data set has more than 10 years of historic data ranging from jan-2010 to current date when the algorithm is executed. The data set has 7 columns in it namely

Date, High, Low, Open, Close, Volume and Adj Close.

2.2 Data Description:

The data has over 2400 records with indexed 0 to n.

Column name	Column Description
Date	Trading day date in format dd-mm-yyyy.
High	The value with the highest stock rate of that particular day.
Low	The value with the lowest stock rate of that particular day.
Open	The opening value of the stock at 10.00am.
Close	The closing value of the stock at 3.30pm.
Volume	The total transactions of that stock. (1volume = 1 transaction)
Adj Close	The official final closing value.

Table 1 : Data Description

2.3 Imports:

Keras: Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow.

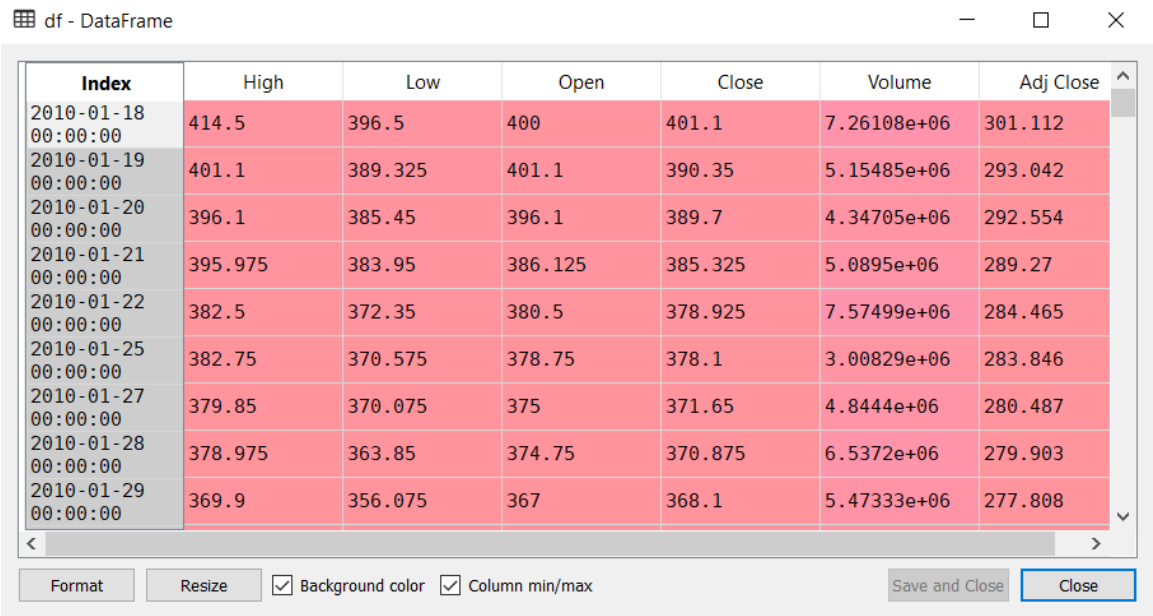
Sklearn: Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms.

Pandas DataReader: The Pandas datareader is a sub package that allows one to create a dataframe from various internet datasources, currently including Yahoo! Finance,Google Financ,St.Louis FED (FRED),Kenneth French's data library,World Bank,Google Analytics.

Pandas: Pandas is an open-source, BSD-licensed Python library providing high performance, easy-to-use data structures and data analysis tools for the Python programming language. We have used pandas for perform operation on data.

Numpy: Numpy is used to for mathematical operations. This package provides easy use of mathematical function.

2.4 Preparing the data:



Index	High	Low	Open	Close	Volume	Adj Close
2010-01-18 00:00:00	414.5	396.5	400	401.1	7.26108e+06	301.112
2010-01-19 00:00:00	401.1	389.325	401.1	390.35	5.15485e+06	293.042
2010-01-20 00:00:00	396.1	385.45	396.1	389.7	4.34705e+06	292.554
2010-01-21 00:00:00	395.975	383.95	386.125	385.325	5.0895e+06	289.27
2010-01-22 00:00:00	382.5	372.35	380.5	378.925	7.57499e+06	284.465
2010-01-25 00:00:00	382.75	370.575	378.75	378.1	3.00829e+06	283.846
2010-01-27 00:00:00	379.85	370.075	375	371.65	4.8444e+06	280.487
2010-01-28 00:00:00	378.975	363.85	374.75	370.875	6.5372e+06	279.903
2010-01-29 00:00:00	369.9	356.075	367	368.1	5.47333e+06	277.808

Fig 2.1: Unprocessed Data

The data set which is requested on demand hand some flaws in it which needed to be handled. The index value from the source was coming with the date. The records had some nulls in it as well. After processing it the results look like

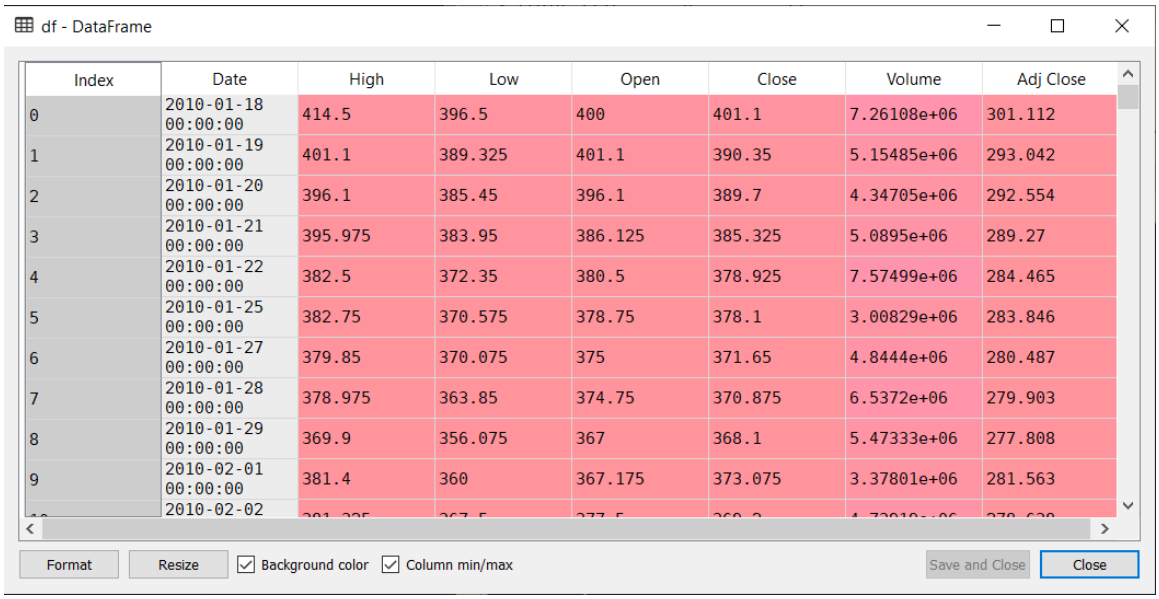


Fig 2.2: Processed Data

Chapter 3
System Analysis

3.1 Analysis:

A time series is a series of data points indexed (or listed or graphed) in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Thus it is a sequence of discrete data. Examples of time series are stock market data, data captured from sensors, weather data etc.Time series analysis comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data. Time series forecasting is the use of a model to predict future values based on previously observed values.Time series data have a natural temporal ordering. This makes time series analysis distinct from cross-sectional studies, in which there is no natural ordering of the observations.

3.2 Data Analysis:

The following fig shows the yearly performance of the stock. Analysing such kind of data makes it easy to visualize that the stock is kept increasing every since 2010 till 2020.

For the period of time 2014 to 2017 there was not any exciting growth in the TCS stock rate. From 2017 to 2020 there is good amount of growth.

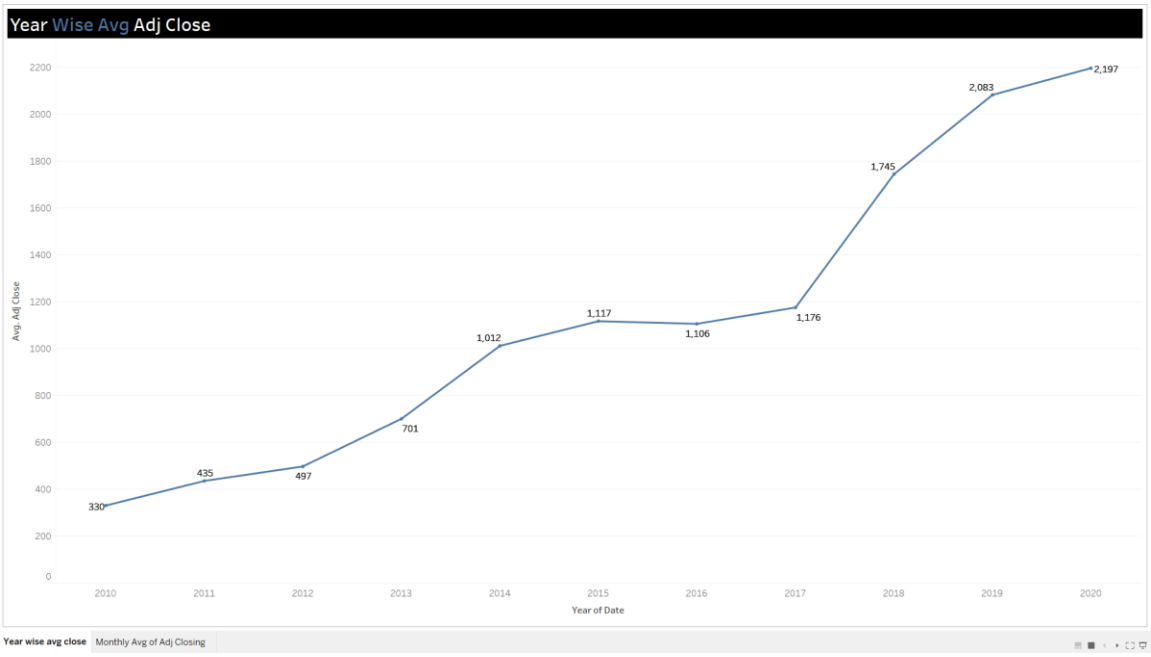


Fig 3.1: Overall yearly performance

Fig 3.2: Monthly avg of adj closing

From the above figure which shows previous one year closing average which gives a clear insights about the rise and fall of the stock data. The figure also shows the maximum value the stock has touched and the average value across one year.

Similarly there are other terms in data sets that is high and low of the particular day of trade. This below fig: shows of the distribution of price over a period of one year with monthly minimum and monthly maximum price. It was clearly seen that some months had considerable amount of difference during the particular month. Where as some months showed there is no considerable difference between the low and high. This insights can be particularly useful for the investors who prefers mid term trading where in they invest for few moths of days. It is different from daily trading and long term investing. Insights gives an clear idea of what is the performance of stock at the first look.

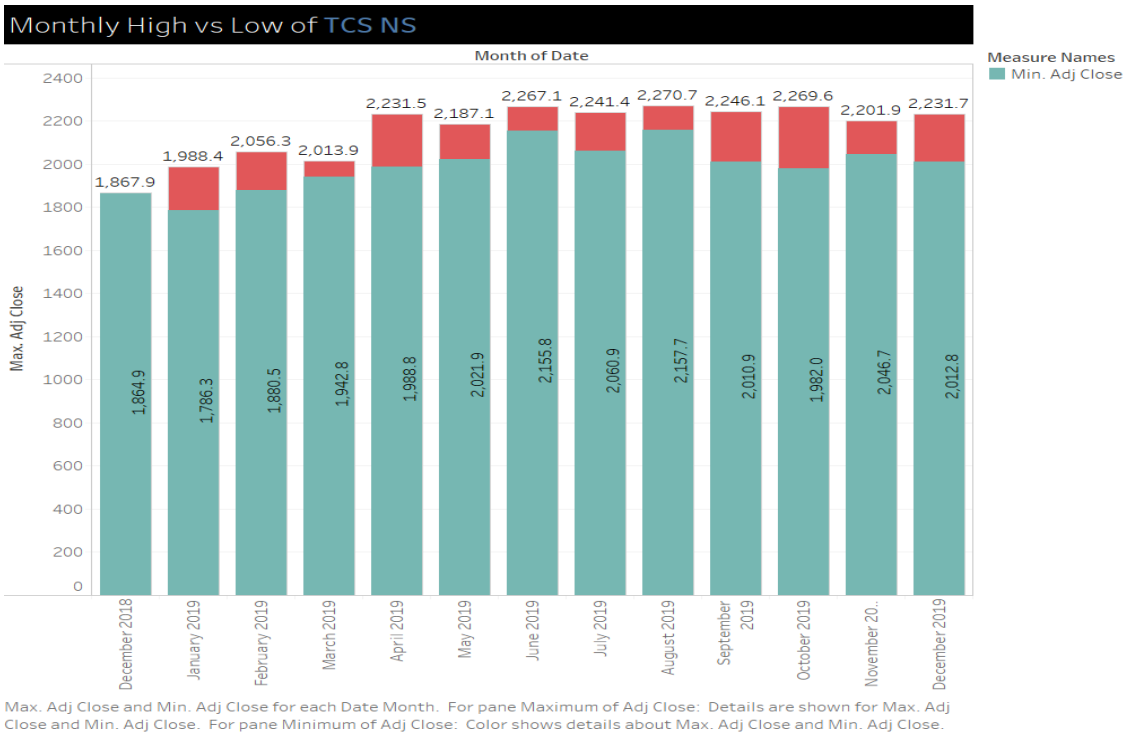


Fig 3.3 : Monthly high vs low

3.3 Training Process of LSTM model:

LSTM stands for Long Short Term Memory which is a part of RNN. It excepts the input to model in the form of vector. So training model required an input of vector with steps 5 as the ACF with 5 was very high.

```
In [6]: model.fit(X, y, epochs=50, verbose=1)
Epoch 1/50
1995/1995 [==============================] - 9s 5ms/step - loss: 70340.4393
Epoch 2/50
1995/1995 [==============================] - 1s 672us/step - loss: 384.8710
Epoch 3/50
1995/1995 [==============================] - 1s 641us/step - loss: 375.1643
Epoch 4/50
1995/1995 [==============================] - 1s 645us/step - loss: 322.0855
Epoch 5/50
1995/1995 [==============================] - 1s 640us/step - loss: 316.9989
Epoch 6/50
1995/1995 [==============================] - 1s 679us/step - loss: 309.7125
Epoch 7/50
1995/1995 [==============================] - 1s 648us/step - loss: 290.3627
Epoch 8/50
1995/1995 [==============================] - 1s 652us/step - loss: 289.6534
Epoch 9/50
1995/1995 [==============================] - 1s 669us/step - loss: 277.3618
Epoch 10/50
1995/1995 [==============================] - 1s 661us/step - loss: 328.9221
Epoch 11/50
1995/1995 [==============================] - 1s 661us/step - loss: 300.6045
Epoch 12/50
1995/1995 [==============================] - 1s 642us/step - loss: 274.2768
Epoch 13/50
 640/1995 [========>.....................] - ETA: 0s - loss: 396.1463
```

Fig 3.4: Training Model of LSTM

3.4 Training Process of ARIMA model:

The ARIMA model has the input of the type series. Y train is the series of training predicate which is passed as input to ARIMA for prediction.Auto ARIMA finds the optimal value of p,d and q.

```
Fit ARIMA: order=(2, 1, 2) seasonal_order=(0, 0, 0, 1); AIC=20986.140, BIC=21020.988, Fit time=3.516 seconds
Fit ARIMA: order=(0, 1, 0) seasonal_order=(0, 0, 0, 1); AIC=20996.779, BIC=21008.395, Fit time=0.075 seconds
Fit ARIMA: order=(1, 1, 0) seasonal_order=(0, 0, 0, 1); AIC=20995.143, BIC=21012.567, Fit time=0.182 seconds
Fit ARIMA: order=(0, 1, 1) seasonal_order=(0, 0, 0, 1); AIC=20994.906, BIC=21012.329, Fit time=0.424 seconds
Fit ARIMA: order=(0, 1, 0) seasonal_order=(0, 0, 0, 1); AIC=20999.709, BIC=21005.517, Fit time=0.052 seconds
Fit ARIMA: order=(1, 1, 2) seasonal_order=(0, 0, 0, 1); AIC=20995.452, BIC=21024.491, Fit time=1.703 seconds
Fit ARIMA: order=(2, 1, 1) seasonal_order=(0, 0, 0, 1); AIC=20985.395, BIC=21014.434, Fit time=1.956 seconds
Fit ARIMA: order=(1, 1, 1) seasonal_order=(0, 0, 0, 1); AIC=20993.456, BIC=21016.688, Fit time=1.047 seconds
Fit ARIMA: order=(2, 1, 0) seasonal_order=(0, 0, 0, 1); AIC=20994.615, BIC=21017.847, Fit time=0.239 seconds
Fit ARIMA: order=(3, 1, 1) seasonal_order=(0, 0, 0, 1); AIC=20997.255, BIC=21032.102, Fit time=1.474 seconds
Fit ARIMA: order=(3, 1, 0) seasonal_order=(0, 0, 0, 1); AIC=20996.104, BIC=21025.143, Fit time=0.232 seconds
Fit ARIMA: order=(3, 1, 2) seasonal_order=(0, 0, 0, 1); AIC=20987.272, BIC=21027.928, Fit time=3.361 seconds
Total fit time: 14.272 seconds
```

Fig 3.5 : Training Model of ARIMA

CHAPTER 4
Requirement Specification

4.1 External requirement specification:

Stock market is one of the most volatile market to predict.There can be numerous factors that effect an stock value in real world.Few of them can be like news that is against a company or in favour of a company.

4.2 Detailed Non-Functional Requirements:

4.2.1 Functional Requirement:

First of all model should be successfully trained by developer.

Installing Anaconda on Windows:

1. Download and install Anaconda (windows version).

2. Select the default options when prompted during the installation of Anaconda.

3. After you finished installing, open **Anaconda Prompt**. Type the command below to see that you can use a Jupyter (IPython) Notebook.

4. If you didn't check the add Anaconda to path argument during the installation process, you will have to add python and conda to your environment variables. You know you need to do so if you open a **command prompt** (not anaconda prompt) and get the following messages.

5. This step gives two options for adding python and conda to your path .If you don't know where your conda and/or python is, you type the following commands into your **anaconda prompt.**

4.2.2 Hardware Requirement:

- **Processor:** Intel Core i3
- **RAM:** Minimum 8 GB
- **OS:** Windows,Linux,MacOS

4.2.3 Software Requirement:

Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch applications and easily manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, macOS and Linux. To get Navigator, get the Navigator cheat sheet and install Anaconda.

The Navigator Getting started with Navigator section shows how to start Navigator from the shortcuts or from a terminal window.

Following libraries of python should be install:

1. **Scikit-learn -** pip install scikit-learn
2. **TensorFlow –** pip install tensorflow (keras uses TensorFlow as backend).
3. **Keras –** pip install keras (to build our classification model).
4. **Pmdarima -** pip install pmdarima==1.2.1

CHAPTER 5
System Evaluation Process

5.1 Flowchart of the System:

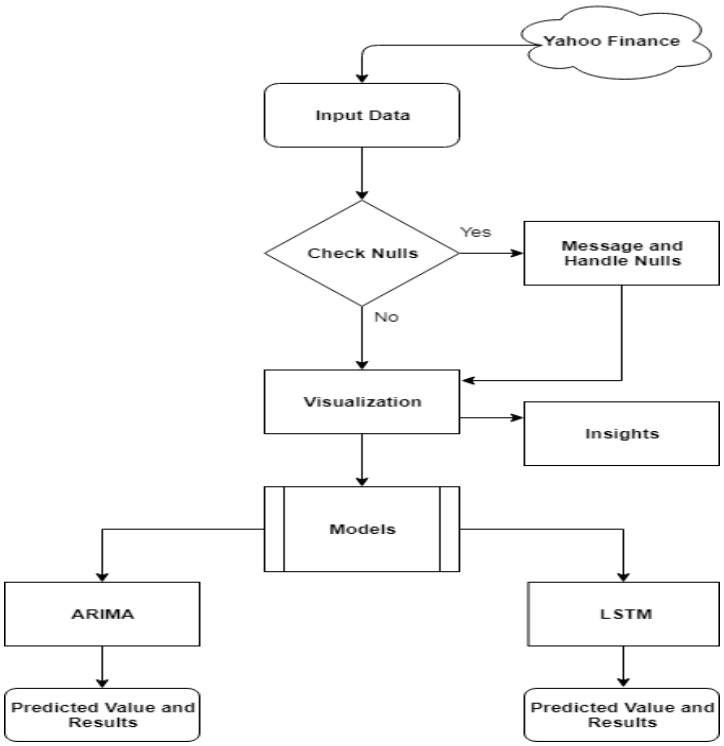


Fig 5.1: Flow Chart of the system

The flow chart diagram in Fig 5.1 refers to the overall execution flow in the system. The main source of data was yahoo fiance. Yahoo finance is an stock exchange site where all the companies listed on National Stock Exchange and Bombay Stock Exchange can be retrieved on demand with an extension of pandas called data reader. It is capable to store the input in the format of Data Frame.

The input data step in flow chart refers to the data which is processed according to the requirements. The train test data set is also declared and splitted in this stage. Further the dataset is passed onto the check nulls stage where in any nulls in the data set is first checked and the boolean result is passed onto the following stage. If result is yes the an alert message is displayed and NA is handled accordingly.

Further the dataset is visualized for better insights. The insights gives an perfect idea about the overall summary of data w.r.t trends and seasonality.

Next stage has Model building stage where respective models namely LSTM and ARIMA is processed. After their respective training is completed the models evaluate the prediction function onto the test data which then is validated by RSME and R2 Score.
The next day predicted value along with their validations are displayed.

5.2 Data Flow:

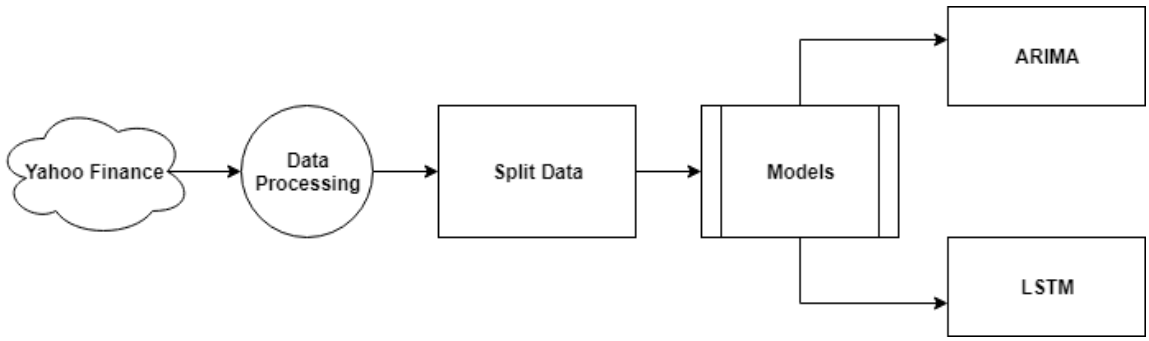


Fig 5.2: Data Flow Diagram

A data-flow diagram is a way of representing a flow of a data of a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops.
Yahoo finance is the source of data which is requested on the go by the stock name and start and end days. Further the data is moved to the Data processing model wherein the data is processed with all the required validations. This validated data is passed onto the split data process where the data is pushed into LSTM and ARIMA.

5.3 LSTM model:

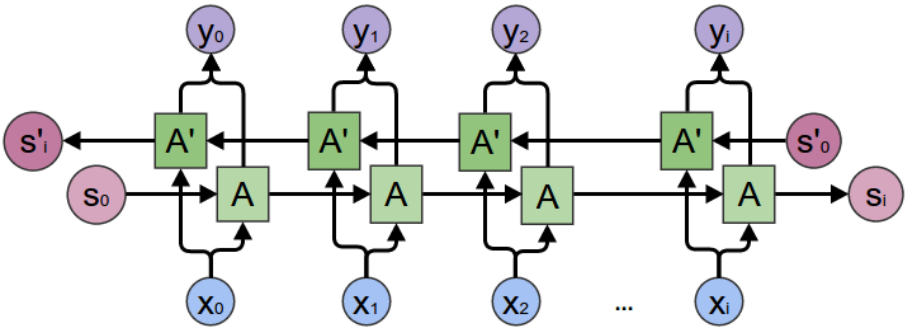


Fig 5.3 : LSTM model

Long short-term memory is an artificial recurrent neural network architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points, but also entire sequences of data.

Fig 5.3 refers to an bidirectional LSTM which is an type of RNN. The bidirectional LSTM can be referred to an example of real life by human where is a person reads a book onces can remember for a shorter period of time but if that person reads for more number of times say 8 to 10 times then can remember for longer.Same is the overall concept of RNN. Long Short Term Memory networks are a special kind of RNN, capable of learning long-term dependencies RNN are networks with loops in them, allowing information to persist. LSTMs are explicitly designed to avoid the long-term dependency problem.

5.4 LSTM implementation

```
model = Sequential()
model.add(Bidirectional(LSTM(162, activation='relu'), input_shape=(n_steps, n_features)))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')

model.fit(X, y, epochs=50, verbose=1)
```

Fig 5.4 LSTM model using keras

Sequential:

There are two ways to build Keras models: sequential and functional. The sequential API allows you to create models layer-by-layer for most problems. It is limited in that it does not allow you to create models that share layers or have multiple inputs or outputs.

Bidirectional :

Bidirectional LSTMs are an extension of traditional LSTMs that can improve model performance on sequence classification problems. In problems where all timesteps of the input sequence are available, Bidirectional LSTMs train two instead of one LSTMs on the input sequence. The first on the input sequence as-is and the second on a reversed copy of the input sequence. This can provide additional context to the network and result in faster and even fuller learning on the problem.

LSTM :

The Long Short-Term Memory network, or LSTM network, is a recurrent neural network that is trained using Backpropagation Through Time and overcomes the vanishing gradient problem. As such, it can be used to create large recurrent networks that in turn can be used to address difficult sequence problems in machine learning and achieve state-of-the-art results. Instead of neurons, LSTM networks have memory blocks that are connected through layers.A block has components that make it smarter than a classical neuron and a memory for recent sequences. A block contains gates that manage the block's state and output. A block operates upon an input sequence and each gate within a block uses the sigmoid activation units to control whether they are triggered or not, making the change of state and addition of information flowing through the block conditional.

Activation :

Activation function to use is default hyperbolic tangent(tanh). If you pass None, no activation is applied.

RELU :

The rectified linear activation function is a piece wise linear function that will output the input directly if is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance.

Dense :

A fully connected layer that often follows LSTM layers and is used for outputting a prediction is called Dense()

Compile :

Configures the model for training, with proper optimizer passed as argument.

5.5 LSTM results:

```
In [8]: np.sqrt(mean_squared_error(y1,predicted))
Out[8]: 35.30782191780164

In [9]: print("R2 Score: ",r2_score(y1,predicted))
R2 Score:  0.9715021477672268

In [10]: PredictValue()
2470   2020-01-29
Name: Date, dtype: datetime64[ns]   [[2200.6755]]
```

Fig 5.5 : LSTM result

Refer to Fig 5.4 that concludes the model with predicted value on dated 2020-01-29 to be 2200.6755 which is expected to be nearby closing rate of the stock. The input dataset was upto 2020-01-28. Here in Fig 5.4 we can also find and evidently strong R2 Score with 0.97 value and RSME with 35.30.

RSME:

Root Mean Square Error is the measure of how well a regression line fits the data points. RMSE can also be construed as Standard Deviation in the residuals.

R-Square :

R-squared in Regression Analysis. R-squared is a statistical measure that represents the goodness of fit of a regression model. The more the value of r-square near to 1, the better is the model.

5.6 ARIMA model :

ARIMA stands for Autoregressive Integrated Moving Average models. Univariate (single vector) ARIMA is a forecasting technique that projects the future values of a series based entirely on its own inertia. Its main application is in the area of short term forecasting requiring at least 40 historical data points. It works best when your data exhibits a stable or consistent pattern over time with a minimum amount of outliers**.** Sometimes called Box-Jenkins (after the original authors), ARIMA is usually superior to exponential smoothing techniques when the data is reasonably long and the correlation between past observations is stable. If the data is short or

highly volatile, then some smoothing method may perform better. If you do not have at least 38 data points, you should consider some other method than ARIMA.

Basic Concepts: The first step in applying ARIMA methodology is to check for stationary. "Stationary" implies that the series remains at a fairly constant level over time. If a trend exists, as in most economic or business applications, then your data is NOT stationary. The data should also show a constant variance in its fluctuations over time. This is easily seen with a series that is heavily seasonal and growing at a faster rate. In such a case, the ups and downs in the seasonality will become more dramatic over time.

Differencing: If a graphical plot of the data indicates non stationary, then you should difference the series. Differencing is an excellent way of transforming a no stationary series to a stationary one.

Autocorrelations: Autocorrelations are numerical values that indicate how a data series is related to itself over time. More precisely, it measures how strongly data values at a specified number of periods apart are correlated to each other over time. The number of periods apart is usually called the "lag". There are Two types of Autocorrelation functions as given below:

1)Auto-correlation function (ACF)

2) Partial Auto-correlation Function (PACF)

Auto-correlation Function: Auto-correlation function correlates a time series with its own past and future values. Simply, ACF measures and explains the internal association between observations within the time series. If correlation exists within a time series, current and past values can be exploited in order to predict the future values.

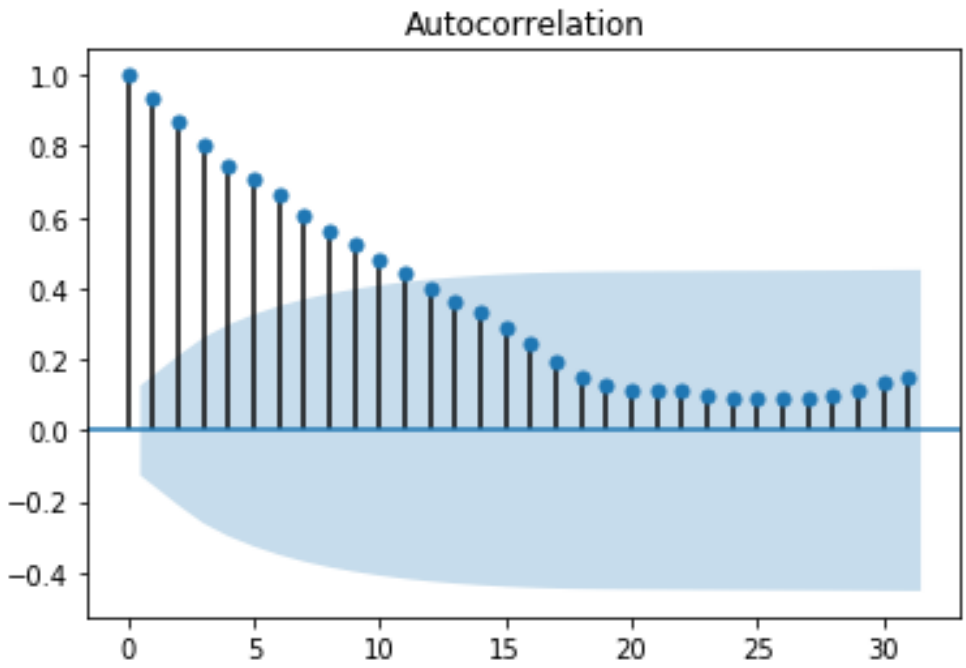


Fig 5.6 : Auto-correlation Graph

Partial Auto-correlation Function: Partial auto-correlation function determines the partial correlation of a time series with its own lagged values. In contrast with the autocorrelation function, PACF controls for the values at all shorter lags of the series. In time series analysis, PACF plays a vital role by identifying the extent of the lag in an autoregressive model. This function determine the appropriate lags p in an AR (p) model, in a mixed ARMA (p, q) model or in an extended ARIMA (p, d, q) model. For an AR(p) model,the partial auto-correlation is 0 at lag p+1 and greater. If the auto-correlation plot determines that an AR model may be appropriate to apply, then the partial auto-correlation plot can be examined for identifying the order. The partial auto-correlations for all higher lags are essentially 0. An indication of the sampling uncertainty of the PACF can be placed on the plot which helps for this purpose and it is constructed on the basis that the true value of the PACF is 0 at any given positive lag.

The below Fig 5.6 refers to the partial auto-correlation between data which results that it shows strong correlation between 2 days.

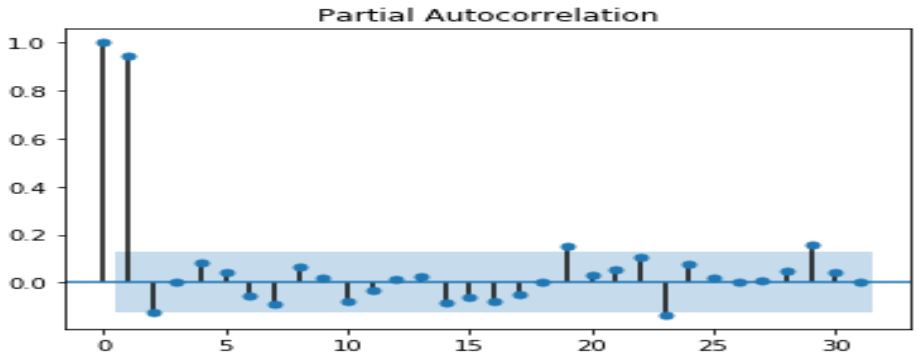


Fig 5.7 : Partial Autocorrelation

5.7 ARIMA model results :

```
Fit ARIMA: order=(2, 1, 2) seasonal_order=(0, 0, 0, 1); AIC=20986.140, BIC=21020.988, Fit time=3.516 seconds
Fit ARIMA: order=(0, 1, 0) seasonal_order=(0, 0, 0, 1); AIC=20996.779, BIC=21008.395, Fit time=0.075 seconds
Fit ARIMA: order=(1, 1, 0) seasonal_order=(0, 0, 0, 1); AIC=20995.143, BIC=21012.567, Fit time=0.182 seconds
Fit ARIMA: order=(0, 1, 1) seasonal_order=(0, 0, 0, 1); AIC=20994.906, BIC=21012.329, Fit time=0.424 seconds
Fit ARIMA: order=(0, 1, 0) seasonal_order=(0, 0, 0, 1); AIC=20999.709, BIC=21005.517, Fit time=0.052 seconds
Fit ARIMA: order=(1, 1, 2) seasonal_order=(0, 0, 0, 1); AIC=20995.452, BIC=21024.491, Fit time=1.703 seconds
Fit ARIMA: order=(2, 1, 1) seasonal_order=(0, 0, 0, 1); AIC=20985.395, BIC=21014.434, Fit time=1.956 seconds
Fit ARIMA: order=(1, 1, 1) seasonal_order=(0, 0, 0, 1); AIC=20993.456, BIC=21016.688, Fit time=1.047 seconds
Fit ARIMA: order=(2, 1, 0) seasonal_order=(0, 0, 0, 1); AIC=20994.615, BIC=21017.847, Fit time=0.239 seconds
Fit ARIMA: order=(3, 1, 1) seasonal_order=(0, 0, 0, 1); AIC=20997.255, BIC=21032.102, Fit time=1.474 seconds
Fit ARIMA: order=(3, 1, 0) seasonal_order=(0, 0, 0, 1); AIC=20996.104, BIC=21025.143, Fit time=0.232 seconds
Fit ARIMA: order=(3, 1, 2) seasonal_order=(0, 0, 0, 1); AIC=20987.272, BIC=21027.928, Fit time=3.361 seconds
Total fit time: 14.272 seconds
```

```
ARIMA Model :-
          Predicted      Adj Close
 2461    2205.769859   2221.854492
2462    2202.491181   2233.727783
2463    2207.337313   2214.072510
2464    2203.412483   2165.432861
2465    2205.899886   2166.131348
2466    2206.381067   2201.899902
2467    2206.087296   2190.949951
2468    2211.121684   2183.399902
2469    2209.548030   2169.250000
2470    2210.066806   2183.750000
dtype: float64
Test RMSE: 24.976
```

Fig 5.8 : Arima model results

The above Fig 5.7 refers to Arima model execution where in the test results are shows with predicted price and expected price. Based on the model training the ARIMA model forecast the value to be 2232.48 of its next day closing day. The model predicted the value with an RMSE 24.97.

CHAPTER 6
Conclusion

In this project the main source of data was yahoo finance. Yahoo finance is an financial based site where the real time data, historic data and custom range of data can be accessed typically by inputting the company name listed on any stock exchange.The data for this project is loaded from the data reader. The data has records from 2010 upto the time project is executed. The program has an time function which takes an input of current date from the system itself. The problem encountered with the live data coming from the source was the index was populated by the date column. Also there where some null values. The index was reset to the data numbered from 0 to n.Out of all the columns the suitable column was Adj close was selected.

LSTM is the first model selected where in the data needed to passed as an vector format and the output is of vector as well. Here LSTM model used was Bidirectional RNN where in the model is processed in loop by feedback mechanism. The model used a hidden layer of 162 layers and 50 epochs. R2 Score for the data was 0.97 and RSME 35.

ARIMA is the second model used for processing the data. During arima the auto correlation function showed an link of 5 days to be more closely related. Here ARIMA take an input of test data after the model is trained. The test data displayed an result of an RMSE value 24.9

CHAPTER 7
Future Scope

It is observed that LSTM model gives better accuracy with more number of hidden layers and epochs. But due to the limitation of hardware resources the process was time consuming.Also in this project can be developed for more advance consumption of stock data like the real time streaming data where in the live stock rate is streaming into the algorithm. This project can be made equipped with automation process where there can be a bot designed who is capable to take real time decision for buy and sell of stock. This can be extremely useful for intra-day traders.

Currently the system was handling a single stock for analysis and prediction it can be scaled in future to process and predict the data in real time for multiple stocks. More number of stocks gives more chances to be more in positive side than in negative in overall.

References

- https://www.analyticsvidhya.com/
- https://machinelearningmastery.com/
- https://towardsdatascience.com/
- https://finance.yahoo.com/