**A Report On**

# "STOCK MARKET TREND PREDICTION"

**Submitted to the**

## Department of Computer Applications

**In Partial Fulfilment of the course**

## Master of Computer Applications

**Under the guidance of**

## MS.REMYA ANAND

**BY**


# ATHUL K KUMAR

**(REG NO: SGI18MCA-I020)**


## DEPARTMENT OF COMPUTER APPLICATION


**SNGIST GROUP OF INSTITUTIONS**

**North Paravur - 683520**


***2018-2023***

# SNGIST GROUP OF INSTITUTIONS

**North Paravur-683520**

## BONAFIDE CERTIFICATE

Certified that the Project Work entitled

## ”Stock Market Trend Prediction”

**is a bonafide work done by**

**ATHUL K KUMAR**

*In partial fulfillment of the requirement for the Award of*

## MASTER OF COMPUTER APPLICATIONS

**Degree from**

APJ Abdul Kalam Technological University, Thiruvananthapuram

(2018-2023)

| | |
|---|---|
| **Prof.Dr. KAVITHA C R** | **Ms.REMYA ANAND** |
| **HEAD OF DEPARTMENT** | **INTERNAL GUIDE** |

Submitted for viva-voce examination held on .....

| | |
|---|---|
| **External Examiner 1** | **External Examiner 2** |

# SNGIST GROUP OF INSTITUTIONS

**North Paravur-683520**


# CERTIFICATE

This is to certify that the project entitled **"Stock Market Trend Prediction"** has been successfully carried out by **ATHUL K KUMAR** (Reg no: SGI18MCA-I020) in partial fulfillment of the Course **Master of Computer Applications**.

**Place:Paravur** **Prof.Dr. KAVITHA C R**

**Date:.....** **HEAD OF DEPARTMENT**

# SNGIST GROUP OF INSTITUTIONS

**North Paravur-683520**


# CERTIFICATE

This is to certify that the project entitled “Stock Market Trend Prediction” has been successfully carried out by ATHUL K KUMAR (Reg no: SGI18MCA-I020) in partial fulfillment of the course Master of Computer Applications under my guidance.

**Place:Paravur**

**Ms.Remya Anand**

**Date:.....** **(Internal Guide)**

# SNGIST GROUP OF INSTITUTIONS

**North Paravur-683520**


# DECLARATION

I, ATHUL K KUMAR, hereby declare that the project work entitled "Stock Market Trend Prediction" is an authenticated work carried out by me under the guidance of Ms.REMYA ANAND for the partial fulfillment of the course **MASTER OF COMPUTER APPLICATIONS**. This work has not been submitted for similar purpose anywhere else except to **SNGIST GROUP OF INSTITUTIONS, North Paravur**, affiliated to **APJ ABDUL KALAM UNIVERSITY, THIRUVANANTHAPURAM**. I understand that detection of any such copying is liable to be punished in any way the college deems fit.

**Place:Paravoor** **Athul K Kumar**

**Date:**

# ACKNOWLEDGEMENT

In the name of almighty **GOD**, I express my sincere thanks to him keeping me fit for successful completion of the project.

I am thankful to **Prof. Dr. SAGINI THOMAS MATHAI, Principal, SNGIST GROUP OF INSTITUTIONS** for her kind support in all respect during our study.

I sincerely thank **Prof. Dr. KAVITHA C.R, HOD, Department of Computer Applications, SNGIST GROUP OF INSTITUTIONS**, for her encouragement to carry out this project.

I especially thank my Guide **Ms.REMYA ANAND, Department of Computer Applications, SNGIST GROUP OF INSTITUTIONS**, in examining the draft of this project and for valuable suggestions.

I want to thank the Department of Computer Applications for giving me the permission to prepare the project on the topic **"Stock Market Trend Prediction"**.

**ATHUL K KUMAR**

# Contents

## List of Figures

# 1 EXECUTIVE SUMMARY

In the era of big data, deep learning for predicting stock market prices and trends has become even more popular than before. We collected years of data from yahoo finance and proposed a comprehensive customization of feature engineering and deep learning-based model for predicting price trend of stock markets. The proposed solution is comprehensive as it includes pre-processing of the stock market dataset, utilization of multiple feature engineering techniques, combined with a customized deep learning based system for stock market price trend prediction. We conducted comprehensive evaluations on frequently used machine learning models and conclude that our proposed solution outperforms due to the comprehensive feature engineering that we built. The system achieves overall high accuracy for stock market trend prediction.

# 2 INTRODUCTION

The financial market is a dynamic and composite system where people can buy and sell currencies, stocks, equities and derivatives over virtual platforms supported by brokers. The stock market allows investors to own shares of public companies through trading either by exchange or over the counter markets. This market has given investors the chance of gaining money and having a prosperous life through investing small initial amounts of money, low risk compared to the risk of opening new business or the need of high salary career. Stock markets are affected by many factors causing the uncertainty and high volatility in the market. Although humans can take orders and submit them to the market, automated trading systems that are operated by the implementation of computer programs can perform better and with higher momentum in submitting orders than any human. However, to evaluate and control the performance of ATSs, the implementation of risk strategies and safety measures applied based on human judgements are required. Many factors are incorporated and considered when developing an ATS, for instance, trading strategy to be adopted, complex mathematical functions that reflect the state of a specific stock, machine learning algorithms that enable the prediction of the future stock value, and specific news related to the stock being analysed. Time-series prediction is a common technique widely used in many real-world applications such as weather forecasting and financial market prediction. It uses the continuous data in a period

of time to predict the result in the next time unit. Many timeseries prediction algorithms have shown their effectiveness in practice. The most common algorithms now are based on Recurrent Neural Networks (RNN), as well as its special type - Long-short Term Memory (LSTM) . Stock market is a typical area that presents time-series data and many researchers study on it and proposed various models. In this project, LSTM model is used to predict the stock price.

## 2.1 EXISTING SYSTEM

### 2.1.1 Forecasting the Stock Market Index Using Artificial Intelligence Techniques

The research work done by Lufuno Ronald Marwala A dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science in Engineering. The weak form of Efficient Market hypothesis (EMH) states that it is impossible to forecast the future price of an asset based on the information contained in the historical prices of an asset. This means that the market behaves as a random walk and as a result makes forecasting impossible. Furthermore, financial forecasting is a difficult task due to the intrinsic complexity of the financial system. The objective of this work was to use artificial intelligence (AI) techniques to model and predict the future price of a stock market index. Three artificial intelligence techniques, namely, neural networks (NN), support vector machines and neuro-fuzzy systems are implemented in forecasting the future price of a stock market index

based on its historical price information. Artificial intelligence techniques have the ability to take into consideration financial system complexities and they are used as financial time series forecasting tools.

### 2.1.2 Automated Stock Price Prediction Using Machine Learning

The research work done by Mariam Moukalled Wassim El-Hajj Mohamad Jaber Computer Science Department American University of Beirut. Traditionally and in order to predict market movement, investors used to analyse the stock prices and stock indicators in addition to the news related to these stocks. Hence, the importance of news on the stock price movement. Most of the previous work in this industry focused on either classifying the released market news as (positive, negative, neutral) and demonstrating their effect on the stock price or focused on the historical price movement and predicted their future movement. In this work, we propose an automated trading system that integrates mathematical functions, machine learning, and other external factors such as news' sentiments for the purpose of achieving better stock prediction accuracy and issuing profitable trades. Particularly, we aim to determine the price or the trend of a certain stock for the coming end-of-day considering the first several trading hours of the day. To achieve this goal, we trained traditional machine learning algorithms and created/trained multiple deep learning models taking into consideration the importance of the relevant news. Various experiments were conducted, the highest accuracy (82.91SVM for Apple Inc. (AAPL)

stock.

## 2.2 PROBLEM DEFINITION

Time Series forecasting  modelling plays an important role in data analysis. Time series analysis is a specialized branch of statistics used extensively in fields such as Econometrics  Operation Research. Time Series is being widely used in analytics data science. Stock prices are volatile in nature and price depends on various factors. The main aim of this project is to predict stock prices using Long short term memory (LSTM).

## 2.3 PROPOSED SYSTEM

There are ways to classify different categories of stocks. Some investors prefer long-term investments, while others show more interest in short-term investments. It is common to see the stock-related reports showing an average performance, while the stock price is increasing drastically, this is one of the phenomena that indicate the stock price prediction has no fixed rules, thus finding effective features before training a model on data is necessary.In this research, we focus on the short-term price trend prediction. Currently, we only have the raw data . So, the very first step is to label the data. We mark the price trend by comparing the current closing price with the closing price of n trading days. The prediction methods can be roughly divided into two categories, statistical methods and Deep learning methods. Statistical

methods include leniar regression model, Descion tree model, etc. Deep learning methods include recurrent neural network, etc. They used Long short-term memory network (LSTM).

## 2.4 OBJECTIVE OF THE PROJECT

In the past decades, there is an increasing interest in predicting markets among economists, policymakers, academics and market makers. The objective of the proposed work is to study and improve the supervised learning algorithms to predict the stock price.Three versions of prediction system will be implemented; one using Leniar regression and other using Descion tree and LSTM. The experimental objective will be to compare the forecasting ability of these and We will test and evaluate both the systems with same test data to find their prediction accuracy

## 2.5 SCOPE OF THE PROJECT

In traditional model stock market trends were predicted using statistical methods and using financial statements.Here machine learning models are used to do the same in a more efficient way. Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange. The successful prediction of a stock's future price could yield significant profit. Through this project we predict the trend of the stock using different model and predict and display the result through graphical representation.

## 2.6 HARDWARE REQUIREMENTS

### 2.6.1 Operating system Windows 7 and above

Microsoft Windows, also called Windows and Windows OS, computer operating system (OS) developed by Microsoft Corporation to run personal computers (PCs). Featuring the first graphical user interface (GUI) for IBM-compatible PCs, the Windows OS soon dominated the PC market. Approximately 90 percent of PCs run some version of Windows.

## 2.7 SOFTWARE REQUIREMENTS

### 2.7.1 Python

Python is a general-purpose language which means it is versatile and can be used to program many different types of functions. Because it is an interpreted language, it precludes the need for compiling code before execution and because it is a high-level programming language, Python is able to abstract details from code. In fact, Python focuses so much attention on abstraction that its code can be understood by most novice programmers.

### 2.7.2 Jupyter Notebook

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter.

Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

### 2.7.3 Streamlit

Streamlit is a free and open-source framework to rapidly build and share beautiful machine learning and data science web apps. It is a Python-based library specifically designed for machine learning engineers. Data scientists or machine learning engineers are not web developers and they're not interested in spending weeks learning to use these frameworks to build web apps. Instead, they want a tool that is easier to learn and to use, as long as it can display data and collect needed parameters for modeling. Streamlit allows you to create a stunning-looking application with only a few lines of code.

# 3 METHODOLOGY

## 3.1 SCRUM

Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value. Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems.

## 3.2 SCRUM ROLES

### 3.2.1 Product Owner

Mr.Shameer K S ,Associate professor,senior faculty,was the product owner for this project ,and acted as spokesman for the customer and defines features of the product based on each backlog items or each specific request of the customer.He would prioritize these features according to the market values,decide on a release date for the product, and is responsible for the profitability of the product . The product owner should also adjust the contents of the features and their priority after every sprint and decide if what has been produced is acceptable.

### 3.2.2 Scrum Master

Prof. Dr. C.R.Kavitha, HOD MCA was the Scrum master for this project. The Scrum master is responsible for making sure a Scrum team lives by the values and practices of Scrum, and for removing any impediments to the progress of the team.

As such, she should shield the team from external interference's, and ensure that the Scrum process is followed, including issuing invitations to the daily Scrum meetings.

#### 3.2.3 Scrum Team

The Scrum team consists of a group of people developing the software product. In this project, the scrum team consists of Mr . Shameer, the product owner, Prof. Dr. C.R.Kavitha, who acted as the project supervisor Ms.Remya Anand and Athul K Kumar, Developer. There is no personal responsibility in Scrum, the whole team fails or succeeds as a single entity.

## 3.3 SPRINT PLANNING MEETING

Most of the time our sprint planning meetings went as planned,though sometimes the product owner was unavailable . In these cases,the meeting simply needed to be scheduled one or two days later . These extra days would come in handy for cleaning up what we had produced the earlier sprint.

## 3.4 DAILY SCRUM MEETING

Our daily Scrums took place at 10am. People could arrive as early as 9.00am and work until then, but as long as they did arrive before the meeting started it did not matter(formally).

## 3.5 SPRINT REVIEW MEETING

Our review meetings were always held on fridays . The product owner would visit the team project room along with any other interested parties,and the team would demonstrate new features on a live system, and answer any questions that might arise during the demo . Usually, we would spend one or two days before the demo checking id everything was working, and run test demonstrations internally.

## 3.6 PRODUCT BACKLOG

### 3.6.1 USER STORIES

| User | Epic | User story |
|---|---|---|
| User | Browsing Dataset | As an user ,I can view the dataset that is used to prepare the machine learning model |
| | Viewing output | As an user ,I can view the final evaluation. |
| | Selecting stock | As an user ,I can select the stock. |

# 4 MILESTONES

## 4.1 Sprint 1

Conducted the first meeting with the College Authority on 8th December 2022 and gathered their requirements for the development of the System.

## 4.2 Sprint 2

Second sprint started on 6nd February 2023 created an model using linear regerssion and evalute it.

## 4.3 Sprint 3

Improve the model by using Decision tree regressor and evaulvate it.

## 4.4 Sprint 4

Developed final model using the LSTM and improve the accuracy based on the past models.

## 4.5 Sprint 5

Developed an user friendly website which user can interact an see the predicted result of the data through the graph.

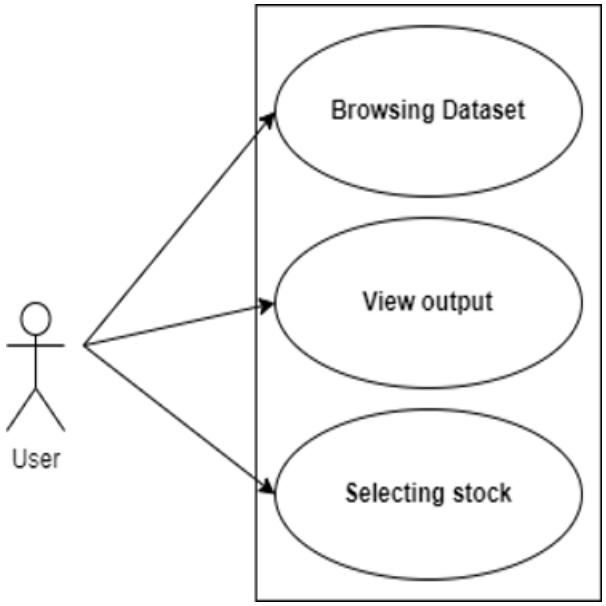
# 5 MODULE DESCRIPTION

## 5.1 Prediction Module

The prediction module is used to predict the drive of the asset. This module is develop by comparing many machine learning algorithm such as linear regression , Decision tree regressor etc. The algorithm for developing the model is selected after evaluating the performance of all the options that are mentioned above and selecting the one which provide the most accurate result.

# 6 SYSTEM DESIGN

## 6.1 USE CASE DIAGRAM



# 7 TESTING

## 7.1 TEST CASE

1. Using the Linear regression in the model it did not show an accurate prediction. There is an large number of bias in shown in the predicted model.

2. Using the Decision Tree regressor comparing with the past model it shown an little bit accuracy but there is high bias at some point.

3. After using LSTM model which is based on deep learning it shows an accurate predicted result with low bias comparing with the other models.

# 8 SYSTEM IMPLEMENTATION

## 8.1 SCREENSHOTS

In [1]:
```
# Install dependeincies
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeRegressor
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
plt.style.use('bmh')

from pandas_datareader import data as pdr
import yfinance as yfin
yfin.pdr_override()
```

In [2]:
```
# Load data
start = '2021-01-01'
end = '2022-12-31'
# start = dt.datetime(2013, 1, 1)
# end = dt.datetime(2016, 1, 27)
df = pdr.get_data_yahoo("AAPL", start, end)
# print(df)
# df = web.DataReader('AAPL', 'yahoo', start='2019-09-10', end='2019-10-09')
# df = web.DataReader('AAPL', 'yahoo' ,start ,end)
df.head()
```

```
[**********100%**********]  1 of 1 completed
```

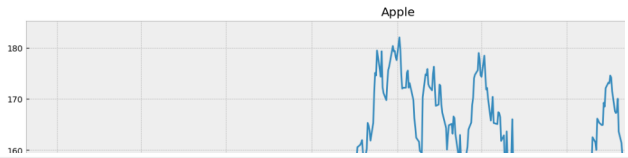
Out[2]:

| Date | Open | High | Low | Close | Adj Close | Volume |
|---|---|---|---|---|---|---|
| 2021-01-04 | 133.520004 | 133.610001 | 126.760002 | 129.410004 | 127.679985 | 143301900 |
| 2021-01-05 | 128.889999 | 131.740005 | 128.429993 | 131.009995 | 129.258606 | 97664900 |
| 2021-01-06 | 127.720001 | 131.050003 | 126.379997 | 126.599998 | 124.907555 | 155088000 |
| 2021-01-07 | 128.360001 | 131.630005 | 127.860001 | 130.919998 | 129.169785 | 109578200 |
| 2021-01-08 | 132.429993 | 132.630005 | 130.229996 | 132.050003 | 130.284714 | 105158200 |

In [3]:
```
df.shape
```

Out[3]:
```
(503, 6)
```

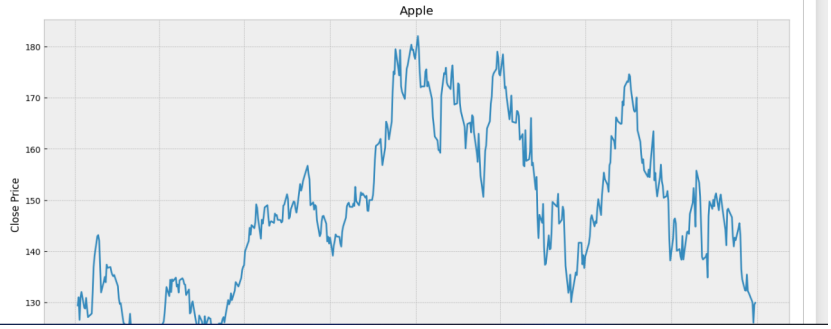
In [4]:
```
# Visualize the close price data
plt.figure(figsize=(16,8))
plt.title('Apple')
plt.xlabel('Year')
plt.ylabel('Close Price')
plt.plot(df['Close'])
plt.show()
```




In [3]:
```
df.shape
```

Out[3]:
```
(503, 6)
```

In [4]:
```
# Visualize the close price data
plt.figure(figsize=(16,8))
plt.title('Apple')
plt.xlabel('Year')
plt.ylabel('Close Price')
plt.plot(df['Close'])
plt.show()
```



In [5]:
```
df = df.reset_index()
df.head()
```

Out[5]:

| | Date | Open | High | Low | Close | Adj Close | Volume |
|---|---|---|---|---|---|---|---|
| 0 | 2021-01-04 | 133.520004 | 133.610001 | 126.760002 | 129.410004 | 127.679985 | 143301900 |
| 1 | 2021-01-05 | 128.889999 | 131.740005 | 128.429993 | 131.009995 | 129.258606 | 97664900 |
| 2 | 2021-01-06 | 127.720001 | 131.050003 | 126.379997 | 126.599998 | 124.907555 | 155088000 |
| 3 | 2021-01-07 | 128.360001 | 131.630005 | 127.860001 | 130.919998 | 129.169785 | 109578200 |
| 4 | 2021-01-08 | 132.429993 | 132.630005 | 130.229996 | 132.050003 | 130.284714 | 105158200 |

In [6]:
```
df.tail()
```

Out[6]:

| | Date | Open | High | Low | Close | Adj Close | Volume |
|---|---|---|---|---|---|---|---|
| 498 | 2022-12-23 | 130.919998 | 132.419998 | 129.639999 | 131.860001 | 131.658981 | 63814900 |
| 499 | 2022-12-27 | 131.380005 | 131.410004 | 128.720001 | 130.029999 | 129.831772 | 69007800 |
| 500 | 2022-12-28 | 129.669998 | 131.029999 | 125.870003 | 126.040001 | 125.847855 | 85438400 |
| 501 | 2022-12-29 | 127.989998 | 130.479996 | 127.730003 | 129.610001 | 129.412415 | 75703700 |
| 502 | 2022-12-30 | 128.410004 | 129.949997 | 127.430000 | 129.929993 | 129.731918 | 76960600 |

In [7]:
```
# Get the close price
df = df[['Close']]
# df = df.drop(['Date'],axis)
df.head(4)
```

Out[7]:

| | Close |
|---|---|
| 0 | 129.410004 |
| 1 | 131.009995 |
| 2 | 126.599998 |
| 3 | 130.919998 |

In [8]:
```
# Create a varible to predict 'x' days out into the future
future_days = 25

# Create a new column (target) shifted 'x' units/days up
df['Prediction'] = df[['Close']].shift(-future_days)
df.head(4)
```

Out[8]:

| | Close | Prediction |
|---|---|---|
| 0 | 129.410004 | 136.009995 |
| 1 | 131.009995 | 135.389999 |
| 2 | 126.599998 | 135.130005 |
| 3 | 130.919998 | 135.369995 |

In [9]:
```
df.tail(4)
```

Out[9]:

| | Close | Prediction |
|---|---|---|
| 499 | 130.029999 | NaN |
| 500 | 126.040001 | NaN |
| 501 | 129.610001 | NaN |
| 502 | 129.929993 | NaN |

In [10]:
```
# Create the feature data set (x) and convert it into numparray and remove the last 'x' days
X = np.array(df.drop(['Prediction'], 1))[:-future_days]
print(X)
```

```
[[129.41000366]
 [131.00999451]
 [126.59999847]
 [130.91999817]
 [132.05000305]
```

In [12]:
```
# Spiting the data into 75% traning and 25% testing
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25)
```

In [13]:
```
# Create the model
# Create the decision tree regressor model
tree = DecisionTreeRegressor().fit(X_train, Y_train)

# Create the linear regression model
lr = LinearRegression().fit(X_train, Y_train)
```

In [14]:
```
# Get the last 'x' rows of the feature data set
X_future = df.drop(['Prediction'], 1)[:-future_days]
# X_future
X_future = X_future.tail(future_days)
X_future = np.array(X_future)
X_future
```

```
C:\Users\AKK\AppData\Local\Temp\ipykernel_4160\605164691.py:2: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'labels' will be keyword-only.
  X_future = df.drop(['Prediction'], 1)[:-future_days]
```

Out[14]:

```
array([[143.38999939],
       [147.27000427],
       [149.44999695],
       [152.33999634],
       [149.3500061 ],
       [144.80000305],
       [155.74000549],
       [153.33999634],
       [150.6499939 ],
       [145.02999878],
       [138.88000488],
       [138.38000488],
       [138.91999817],
       [139.5       ],
       [134.86999512],
```

In [15]:
```
# Show model tree prediction
tree_prediction = tree.predict(X_future)
print(tree_prediction)
print()

# Show the model linear regression prediction
lr_prediction = lr.predict(X_future)
print(lr_prediction)
```

```
[149.32000732 155.11000061 141.16999817 148.02999878 148.30999756
 147.80999756 146.63000488 176.27999878 140.94000244 142.6499939
 142.16000366 147.88999939 164.86999512 143.21000671 136.5
 147.53999329 132.36999512 132.30000305 135.44999695 132.22999573
 131.86000061 130.02999878 126.04000092 129.61000061 143.42999268]

[145.46237746 147.7968943  149.10854976 150.84740125 149.04838739
 146.31074683 152.89311481 151.44908009 149.83056254 146.44913039
 142.74880919 142.44796977 142.77287231 143.12184714 140.33607117
 147.55621725 149.25896947 148.40458662 149.46353808 148.71143953
 149.87268446 150.21563663 148.24213076 149.54777275 150.08327573]
```

In [16]:
```
# Visualize the data
Predictions = tree_prediction

valid = df[X.shape[0]:]
valid['Predictions'] = Predictions
plt.figure(figsize=(16,8))
plt.title('Model')
plt.xlabel('Days')
plt.ylabel('Close Price')
plt.plot(df['Close'])
plt.plot(valid[['Close', 'Predictions']])
plt.legend(['Orig', 'Val', 'Pred'])
plt.show()
```

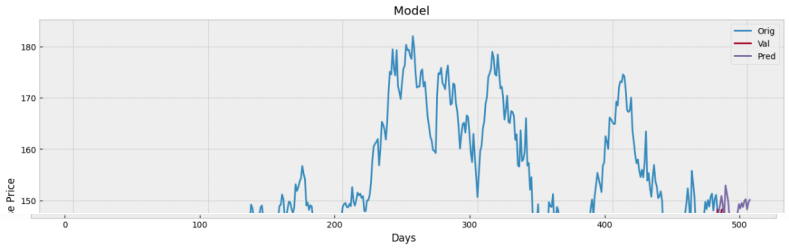
```
C:\Users\AKK\AppData\Local\Temp\ipykernel_4160\1521810972.py:5: SettingWithCopyWarning:
```

In [17]:
```
# Visualize the data
Predictions = lr_prediction

valid = df[X.shape[0]:]
valid['Predictions'] = Predictions
plt.figure(figsize=(16,8))
plt.title('Model')
plt.xlabel('Days')
plt.ylabel('Close Price')
plt.plot(df['Close'])
plt.plot(valid[['Close', 'Predictions']])
plt.legend(['Orig', 'Val', 'Pred'])
plt.show()
```

```
C:\Users\AKK\AppData\Local\Temp\ipykernel_4160\3959655313.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  valid['Predictions'] = Predictions
```




In [18]:
```
start = '2010-01-01'
end = '2022-12-31'
# start = dt.datetime(2013, 1, 1)
# end = dt.datetime(2016, 1, 27)
df = pdr.get_data_yahoo("AAPL", start, end)
# print(df)
# df = web.DataReader('AAPL', 'yahoo', start='2019-09-10', end='2019-10-09')
# df = web.DataReader('AAPL', 'yahoo' ,start ,end)
df.head()
```

```
[**********100%**********]  1 of 1 completed
```

Out[18]:

| Date | Open | High | Low | Close | Adj Close | Volume |
|---|---|---|---|---|---|---|
| **2010-01-04** | 7.622500 | 7.660714 | 7.585000 | 7.643214 | 6.505281 | 493729600 |
| **2010-01-05** | 7.664286 | 7.699643 | 7.616071 | 7.656429 | 6.516527 | 601904800 |
| **2010-01-06** | 7.656429 | 7.686786 | 7.526786 | 7.534643 | 6.412872 | 552160000 |
| **2010-01-07** | 7.562500 | 7.571429 | 7.466071 | 7.520714 | 6.401018 | 477131200 |
| **2010-01-08** | 7.510714 | 7.571429 | 7.466429 | 7.570714 | 6.443575 | 447610800 |

In [19]:
```
df.tail()
```

Out[19]:

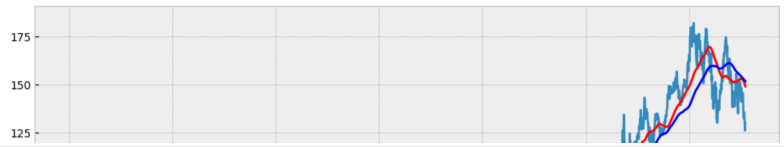
| Date | Open | High | Low | Close | Adj Close | Volume |
|---|---|---|---|---|---|---|
| **2022-12-23** | 130.919998 | 132.419998 | 129.639999 | 131.860001 | 131.658981 | 63814900 |
| **2022-12-27** | 131.380005 | 131.410004 | 128.720001 | 130.029999 | 129.831772 | 69007800 |
| **2022-12-28** | 129.669998 | 131.029999 | 125.870003 | 126.040001 | 125.847855 | 85438400 |

In [25]:
```
# Finding 200 days moving average
ma200 = df.Close.rolling(200).mean()
ma200
```

Out[25]:
```
0            NaN
1            NaN
2            NaN
3            NaN
4            NaN
           ...
3267    152.1331
3268    152.0096
3269    151.8867
3270    151.7593
3271    151.6110
Name: Close, Length: 3272, dtype: float64
```

In [26]:
```
# Visualize 200 days moving average
plt.figure(figsize=(12,6))
plt.plot(df.Close)
plt.plot(ma100, 'r')
plt.plot(ma200, 'b')
```

Out[26]:
```
[<matplotlib.lines.Line2D at 0x270664836a0>]
```




In [34]:
```
x_train = []
y_train = []

for i in range(100, data_traning_array.shape[0]):
    x_train.append(data_traning_array[i-100: i])
    y_train.append(data_traning_array[i,0])

x_train, y_train = np.array(x_train), np.array(y_train)
```

In [35]:
```
from keras.layers import Dense, Dropout, LSTM
from keras.models import Sequential
```

In [36]:
```
model = Sequential()
model.add(LSTM(units = 50, activation='relu', return_sequences=True,
               input_shape = (x_train.shape[1], 1)))
model.add(Dropout(0.2))


model.add(LSTM(units = 60, activation='relu', return_sequences=True))
model.add(Dropout(0.3))


model.add(LSTM(units = 80, activation='relu', return_sequences=True))
model.add(Dropout(0.4))


model.add(LSTM(units = 120, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(units = 1))
```

In [37]:
```
model.summary()
```

```
Model: "sequential"
```

# 9 CONCLUSION AND FUTURE ENHANCEMENT

In this paper, we analyze the growth of the companies from different sector and try to find out which is the best time span for predicting the future price of the share. So, this draws an important conclusion that companies from a certain sector have the same dependencies as well as the same growth rate. The prediction can be more accurate if the model will train with a greater number of data set. Moreover, in the case of prediction of various shares, there may be some scope of specific business analysis. We can study the different pattern of the share price of different sectors and can analyze a graph with more different time span to fine tune the accuracy. This framework broadly helps in market analysis and prediction of growth of different companies in different time spans. Incorporating other parameters (e.g. investor sentiment, election outcome, geopolitical stability) that are not directly correlated with the closing price may improve the prediction accuracy.

# 10 APPENDIX A

## 10.1 SAMPLE SOURCE CODE

```
    import numpy as np
import pandas as pd
# import pandas_datareader.data as web
import matplotlib.pyplot as plt
import datetime as dt
from keras.models import load_model
import streamlit as st


from pandas_datareader import data as pdr
import yfinance as yfin
yfin.pdr_override()


start = '2010-01-01'
end = '2022-12-31'
# start = dt.datetime(2013, 1, 1)
# end = dt.datetime(2016, 1, 27)


st.title('Stock Trend Prediction')
```

```
user_input = st.text_input('Enter Stock Ticker', 'AAPL')


df = pdr.get_data_yahoo(user_input, start, end)

# print(df)

# df = web.DataReader('AAPL', 'yahoo', start='2019-09-10', end='2019-10-09

# df = web.DataReader('AAPL', 'yahoo' ,start ,end)


st.subheader('Data Description from 2010 - 2019')

st.write(df.describe())


# Visulization

st.subheader('Closing Price vs Time Chart')

fig = plt.figure(figsize = (12,6))

plt.plot(df.Close)

st.pyplot(fig)



st.subheader('Closing Price vs Time Chart with 100 MA')

ma100 = df.Close.rolling(100).mean()

fig = plt.figure(figsize = (12,6))

plt.plot(ma100)
```

```
plt.plot(df.Close)

st.pyplot(fig)



st.subheader('Closing Price vs Time Chart with 100 MA & 200 MA')

ma100 = df.Close.rolling(100).mean()

ma200 = df.Close.rolling(200).mean()

fig = plt.figure(figsize = (12,6))

plt.plot(ma100)

plt.plot(ma200)

plt.plot(df.Close)

st.pyplot(fig)



data_traning = pd.DataFrame(df['Close'][0:int(len(df)*0.70)])

data_testing = pd.DataFrame(df['Close'][int(len(df)*0.70):int(len(df))])


from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler(feature_range=(0,1))


data_traning_array = scaler.fit_transform(data_traning)
```

```
# Splitting data into xtrain and ytrain

# x_train = []

# y_train = []


# for i in range(100, data_traning_array.shape[0]):

#     x_train.append(data_traning_array[i-100: i])

#     y_train.append(data_traning_array[i,0])


# x_train, y_train = np.array(x_train), np.array(y_train)


# Load my model


model = load_model('kera_model.h5')


# Testing part


past_100_days = data_traning.tail(100)

final_df = past_100_days.append(data_testing, ignore_index=True)
```

```
input_data = scaler.fit_transform(final_df)



x_test = []

y_test = []


for i in range(100, input_data.shape[0]):

    x_test.append(input_data[i-100: i])

    y_test.append(input_data[i, 0])



x_test, y_test = np.array(x_test), np.array(y_test)

y_predicted = model.predict(x_test)

scaler = scaler.scale_

scale_factor = 1/scaler[0]

y_predicted = y_predicted*scale_factor

y_test = y_test*scale_factor




# Final Graph
```

```
st.subheader('Prediction vs Orginal')

fig2 = plt.figure(figsize=(12,6))

plt.plot(y_test, 'b', label = 'Original Price')

plt.plot(y_predicted, 'r', label = 'Predicted Price')

plt.xlabel('Time')

plt.ylabel('Price')

plt.legend()

st.pyplot(fig2)
```

# 11 APPENDIX B

## 11.1 WEBLIOGRAPHY


1. **https://www.simplilearn.com/tutorials/machine-learning-tutorial/stock-price-prediction-using-machine-learning**
2. **https://intellipaat.com/blog/what-is-lstm/**
3. **https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/**
4. **https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/**
5. **https://www.geeksforgeeks.org/long-short-term-memory-networks-explanation/**


## 11.2 BIBLIOGRAPHY


1. **Heaven, Will Douglass. "AI Is Wrestling with a Replication Crisis." MIT Technology Review, 2020.**
2. **Munro, Robert. Human-in-the-Loop Machine Learning. Shelter Island, New York: Manning, 2020.**
3. **"Milestones:DIALOG Online Search System, 1966 - Engineering and Technology History Wiki," 2019.**

4. **Rudin, Cynthia. "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead." Nature Machine Intelligence 1, no. 5 (2019): 206–15.**