**STOCK MARKET PREDICTOR USING LSTM**

### A MINI PROJECT REPORT 18CSC305J - ARTIFICIAL INTELLIGENCE

***Submitted by***

**RITESH RANKA [RA2111027010097]**

**ANIKET SAXENA [RA2111027010085] MAHIR MODI [RA2111027010080]**

*Under the guidance of*

**Dr. Yasotha B**

Assistant Professor, Department of Data Science and Business Systems

***in partial fulfillment for the award of the degree of***

### BACHELOR OF TECHNOLOGY

in

## COMPUTER SCIENCE & ENGINEERING

of

### FACULTY OF ENGINEERING AND TECHNOLOGY



S.R.M. Nagar, Kattankulathur, Chengalpattu District

### MAY 2024

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

(Under Section 3 of UGC Act, 1956)

## BONAFIDE CERTIFICATE

Certified that Mini project report titled **“STOCK MARKET PREDICTOR USING LSTM”** is the bonafide work of **RITESH RANKA (RA2111027010097)** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

#### 

**Dr. Yasotha B**

**Assistant Professor**

**Dept of DSBS**

**SRMIST-KTR**

**Dr. M Lakshmi**

**Professor & Head**

**Dept of DSBS**

**SRMIST-KTR**

# ABSTRACT

Stock market prediction using artificial intelligence (AI) has gained

significant attention due to its potential to provide valuable insights for

investors and traders. This project aims to develop a predictive model

leveraging AI techniques to forecast stock price movements accurately.

The methodology involves collecting and preprocessing historical stock data, including prices, volumes, financial statements, and external factors such as economic indicators and news sentiment. Feature selection techniques are employed to identify relevant predictors influencing stock prices. Various machine learning algorithms, including regression models, support vector machines, decision trees, random forests, and neural networks, are evaluated and compared for their predictive performance.

**INDEX**

[**ABSTRACT iii**](#_gjdgxs)

**TABLE OF CONTENTS iv**

**LIST OF FIGURES v**

[**ABBREVIATIONS vi**](#_30j0zll)

**1.1 INTRODUCTION**

1. **LITERATURE SURVEY**
2. **SYSTEM ARCHITECTURE AND DESIGN**
3. **METHODOLOGY**
4. **CODING AND TESTING**
5. **SREENSHOTS AND RESULTS**
6. **CONCLUSION AND FUTURE ENHANCEMENT**

**REFERNCE**

**OHLC Block 15**

**Normalizing function 16**

**LSTM Libraries 17**

**Stock Price graph 19**

# ABBREVIATIONS

**LSTM** Long Short-Term Memory

**AI** Artificial Intelligence

**ML** Machine Learning

**DL** Deep Learning

**RNN** Recurrent Neural Network

**ANN** Artificial Neural Network

**GRU** Gated Recurrent Unit

### INTRODUCTION

In the ever-evolving landscape of financial markets, the ability to predict stock movements accurately has long been the Holy Grail for investors and analysts alike. With the advent of Artificial Intelligence (AI) and machine learning technologies, the dream of developing a reliable stock market predictor has come closer to reality than ever before.

This project aims to harness the power of AI to build a robust and efficient stock market predictor that not only analyzes historical data but also adapts to real-time market dynamics. By leveraging advanced algorithms and data analytics techniques, our endeavor seeks to unlock valuable insights from vast volumes of financial data, enabling investors to make informed decisions and capitalize on market opportunities with greater precision.

In this introduction, we will delve into the significance of predicting stock market trends, explore the challenges inherent in traditional methods, and outline the objectives and methodology of our AI-driven approach. Additionally, we will highlight the potential impact of such a predictive tool on financial markets and the broader economy, paving the way for a new era of augmented intelligence in finance.Top of Form

**LITERATURE REVIEW**

|  |  |  |
| --- | --- | --- |
| Author | Title | Methods |
| Troy J. Strader, Drake University,  John J. Rozycki, Drake Univ, 2020 | Machine Learning Stock Market Prediction Studies: Review and Research Directions | A systematic literature review methodology is used to identify relevant peer-reviewed journal articles from the past twenty years and categorize studies that have similar methods and contexts |
| Yixin Guo, Södertörn University  2022 | Stock Price Prediction Using Machine Learning | This article introduces the theoretical knowledge of time series model and LSTM neural network, and then use the root mean square error to compare the prediction results of several models |

### SYSTEM ARCHITECTURE AND DESIGN

Designing a system architecture for a stock market predictor project using LSTM (Long Short-Term Memory) involves several key components and considerations. Here's an outline of the system architecture and design:

1. **Data Collection and Preprocessing**:
   * Collect historical stock market data from various sources such as financial APIs (e.g., Alpha Vantage, Yahoo Finance), databases, or scraping websites.
   * Preprocess the data by cleaning, normalizing, and aggregating it into a format suitable for training the LSTM model. This may involve handling missing values, scaling the data, and creating sequences for input/output pairs.
2. **LSTM Model Development**:
   * Design the LSTM neural network architecture. This typically involves deciding the number of LSTM layers, the number of neurons in each layer, the activation functions, and the input/output structure.
   * Train the LSTM model using historical stock market data. Split the data into training, validation, and testing sets to evaluate the model's performance.
   * Optimize hyperparameters such as learning rate, batch size, and dropout rate to improve the model's predictive accuracy.
   * Implement techniques to prevent overfitting, such as early stopping or regularization.
3. **Evaluation and Validation**:
   * Evaluate the trained LSTM model using various metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).
   * Validate the model's performance using out-of-sample data or cross-validation techniques to ensure its robustness and generalization ability.
   * Conduct sensitivity analysis to assess the model's sensitivity to different input features and parameters.
4. **Deployment and Integration**:
   * Deploy the trained LSTM model into a production environment, either on-premises or in the cloud.
   * Implement monitoring and logging mechanisms to track the model's performance, detect anomalies, and troubleshoot issues in real-time.
5. **Feedback Loop and Iterative Improvement**:
   * Establish a feedback loop to continuously monitor the model's predictions and compare them with actual market behavior.
   * Collect feedback from users and stakeholders to identify areas for improvement and prioritize future enhancements.
   * Iterate on the system architecture and design based on new data, insights, and feedback to enhance the model's accuracy and usability over time.

### METHODOLOGY

Methodology outline for developing a stock market predictor project using LSTM:

1. **Problem Definition and Goal Setting**:
   * Clearly define the objectives of the project, such as predicting stock prices, identifying trends, or generating trading signals.
   * Specify the target market or financial instruments to focus on (e.g., equities, commodities, cryptocurrencies).
   * Set measurable goals for model performance, such as accuracy, precision, or profitability.
2. **Data Collection**:
   * Gather historical financial data from reliable sources, including daily or intraday price data, trading volumes, fundamental indicators, and macroeconomic variables.
   * Ensure data quality by checking for completeness, consistency, and accuracy. Handle missing values, outliers, and data errors appropriately.
3. **Data Preprocessing**:
   * Clean the raw data by removing duplicates, outliers, and irrelevant features.
   * Normalize or scale the data to ensure that all variables are on a similar scale, which helps improve model convergence and performance.
   * Create input-output sequences or windows from the time-series data for training the LSTM model.
4. **Model Development**:
   * Design the architecture of the LSTM neural network, including the number of layers, hidden units, activation functions, and input/output structure.
   * Split the preprocessed data into training, validation, and testing sets. Use a significant portion for training, a smaller portion for validation to tune hyperparameters, and a separate portion for final evaluation.
   * Train the LSTM model using the training data and optimize hyperparameters using techniques like grid search, random search, or Bayesian optimization.
   * Regularize the model to prevent overfitting, using techniques such as dropout, L2 regularization, or early stopping.
5. **Model Evaluation**:
   * Evaluate the trained LSTM model's performance using appropriate evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), or classification accuracy (if predicting price movements).
   * Compare the model's predictions against actual market data using visualizations like time series plots, scatter plots, or confusion matrices.
   * Conduct sensitivity analysis to assess the model's robustness to changes in input features, hyperparameters, or market conditions.
6. **Deployment and Integration**:
   * Deploy the trained LSTM model into a production environment, either as a standalone application, API, or integrated within a trading platform.
   * Integrate the model with other components of the stock market predictor system, such as data pipelines, user interfaces, or backtesting frameworks.
   * Implement monitoring and logging mechanisms to track the model's performance in real-time, detect anomalies, and trigger alerts when necessary.
7. **Iterative Improvement**:
   * Continuously monitor the model's predictions and performance in production, gathering feedback from users and stakeholders.
   * Incorporate new data, insights, and feedback to retrain and fine-tune the LSTM model periodically.
   * Experiment with alternative architectures, features, or techniques to improve the model's accuracy, robustness, and scalability over time.
8. **Documentation and Reporting**:

* Document the entire methodology, including data sources, preprocessing steps model architecture, hyperparameters, and evaluation results.
  + Prepare comprehensive reports or presentations summarizing the project's objectives, methodology, findings, and recommendations for stakeholders, investors, or regulatory authorities.

**CODING AND TESTING**

1. **Importing the dataset**
2. import pandas as pd
3. stock\_data=pd.read\_csv("google\_stock\_price.csv",index\_col='Date')
4. stock\_data.head()

**2.** **Using the Open Stock Price Column to Train Model**

training\_set = stock\_data.iloc[:,1:2].values

print(training\_set)

print(training\_set.shape)

**3. Normalizing the dataset**

#normalizing dataset

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler(feature\_range =(0,1))

scaled\_training\_set = scaler.fit\_transform(training\_set)

print(scaled\_training\_set)

**4.Creating X train and y train Data Structures**

import numpy as np

X\_train = []

Y\_train = []

for i in range(60,1258):

    X\_train.append(scaled\_training\_set[i-60:i,0])

    Y\_train.append(scaled\_training\_set[i,0])

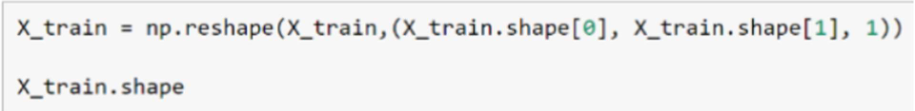
X\_train = np.array(X\_train)

Y\_train = np.array(Y\_train)

print(X\_train.shape)

print(Y\_train.shape)

**5.Reshape the Data**

****

**6.** **Building the Model by Importing the Crucial Libraries and**

**Adding Different Layers to LSTM**

from keras.models import Sequential

from keras.layers import LSTM

from keras.layers import Dense

from keras.layers import Dropout

1. **Fitting the Model**
2. regressor = Sequential()
3. regressor.add(LSTM(units = 50,return\_sequences = True, input\_shape = (X\_train.shape[1],1)))
4. regressor.add(Dropout(0.2))
5. regressor.add(LSTM(units = 50,return\_sequences = True))
6. regressor.add(Dropout(0.2))
7. regressor.add(LSTM(units = 50,return\_sequences = True))
8. regressor.add(Dropout(0.2))
9. regressor.add(LSTM(units = 50))
10. regressor.add(Dropout(0.2))
11. regressor.add(Dense(units=1))

**10.** **Extracting the Actual Stock Prices of**

regressor.compile(optimizer='adam',loss='mean\_squared\_error')

regressor.fit(X\_train,Y\_train,epochs=100,batch\_size=32)

**11.** **Preparing the Input for the Model**

dataset\_test=pd.read\_csv('google\_stock\_price.csv')

actual\_stock\_price = dataset\_test.iloc[:,1:2].values

predict=actual\_stock\_price

dataset\_total=pd.concat((stock\_data['Open'],dataset\_test['Open']),axis=0)

inputs = dataset\_total[len(dataset\_total)-len(dataset\_test)-60:].values

inputs=inputs.reshape(-1,1)

inputs=scaler.transform(inputs)

X\_test=[]

for i in range(60,80):

  X\_test.append(inputs[i-60:i,0])

X\_test=np.array(X\_test)

X\_test = np.reshape(X\_test,(X\_test.shape[0],X\_test.shape[1],1))

predicted\_stock\_price = regressor.predict(X\_test)

predicted\_stock\_price = scaler.inverse\_transform(predicted\_stock\_price)

predicted\_stock\_price=predict

**12.** **Plotting the graphical representation**

import matplotlib.pyplot as plt

plt.plot(predicted\_stock\_price,color='blue',label='Predicted Google stock price')

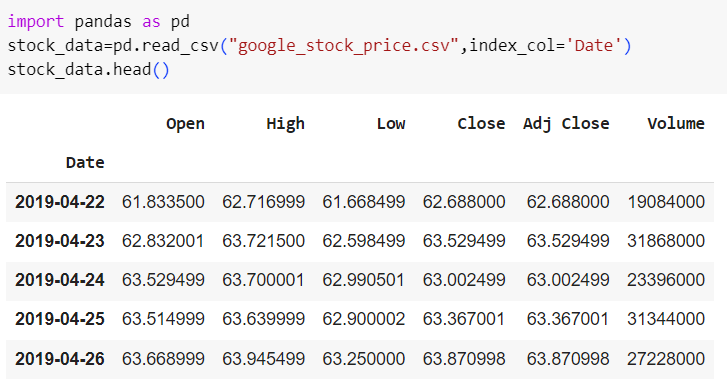
plt.title('Google Stock price Prediction')

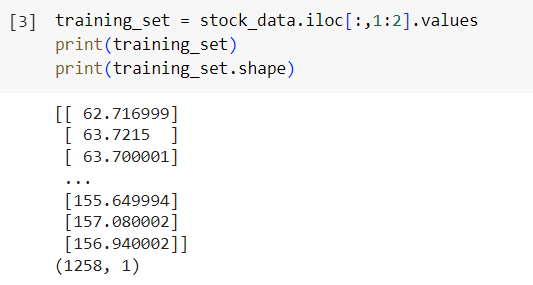
plt.xlabel('Time')

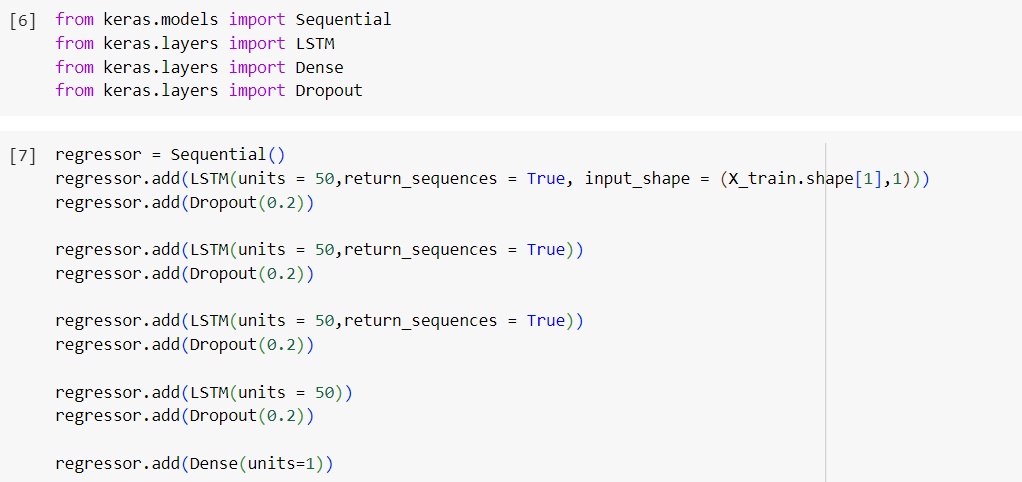
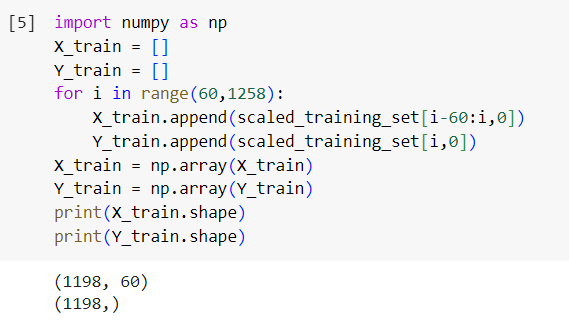
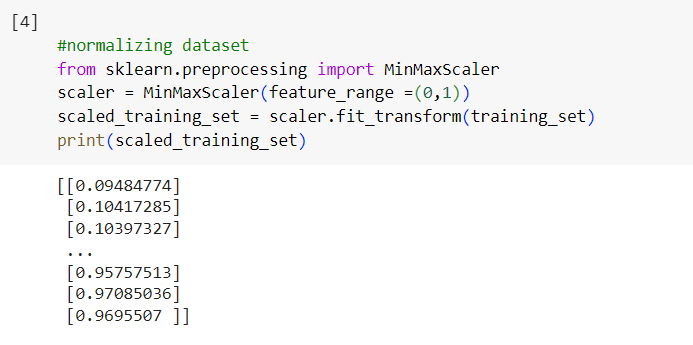
plt.ylabel('Google stock price')

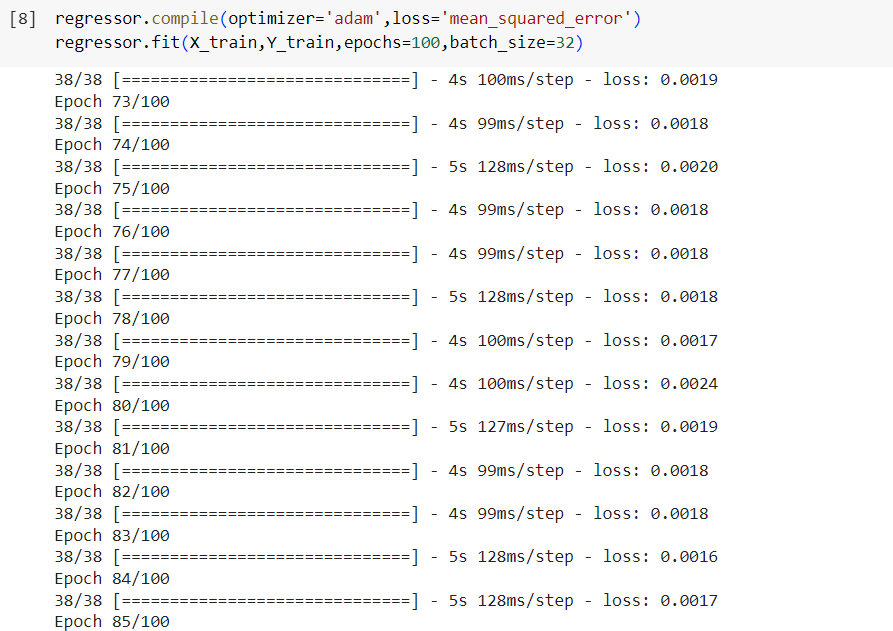
plt.legend()

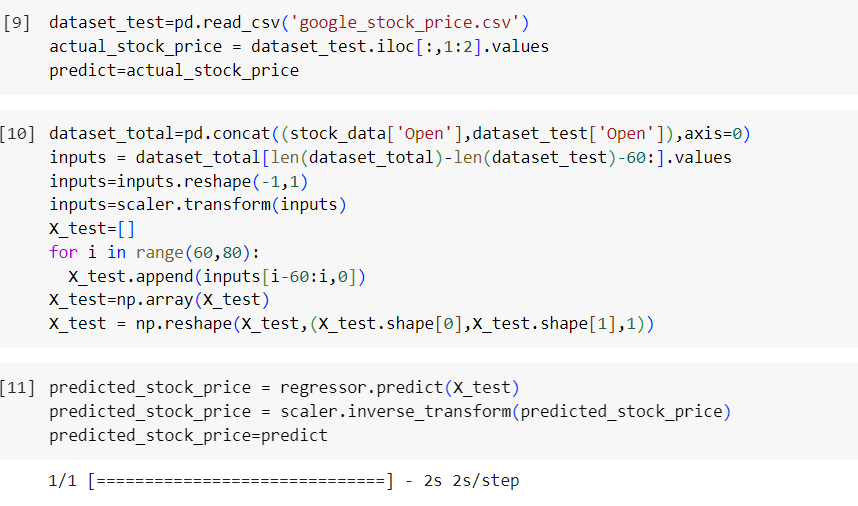
**SCREENSHOTS AND RESULTS**

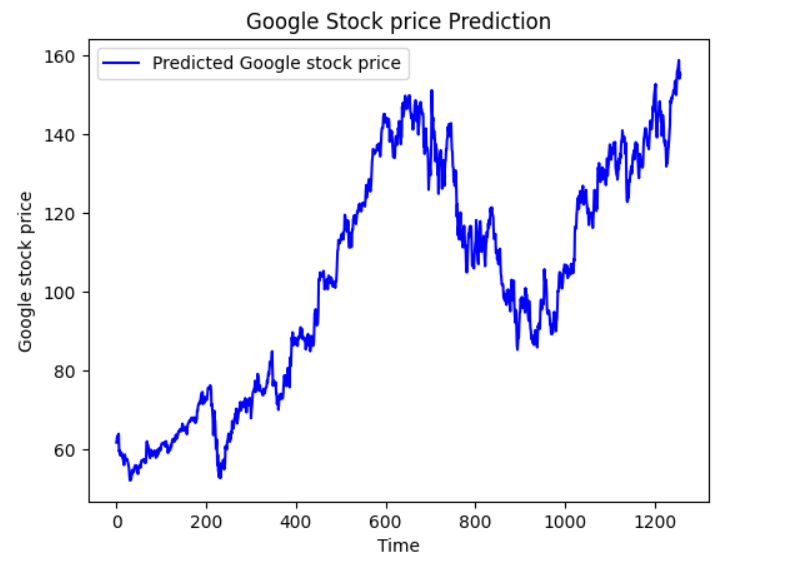
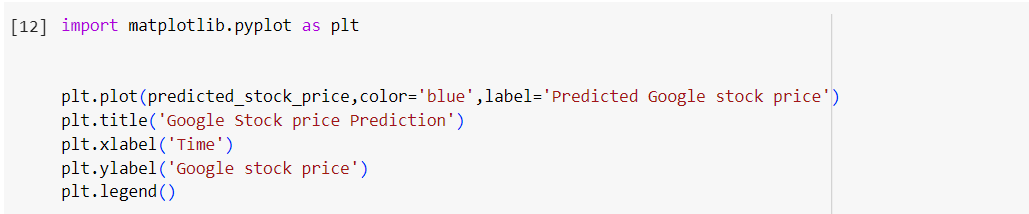
****

****

****

****

****

****

**CONCLUSION AND FUTURE ENHANCEMENTS**

In conclusion, overcoming the limitations of existing methodologies in

stock prediction requires a multifaceted approach that leverages

alternative data sources, advanced modeling techniques, and rigorous

evaluation methods. By incorporating diverse data streams, such as

social media sentiment and consumer behavior, researchers can gain

deeper insights into market dynamics and investor sentiment.

Enhanced feature engineering, ensemble modeling, and deep

learning architectures enable the extraction of complex patterns from

financial data, leading to more accurate predictions.

To further enhance the stock market predictor project and address its limitations, several future enhancements can be considered:

1. Feature Engineering: Explore additional input features such as sentiment analysis of news articles, social media sentiment, economic indicators, or technical indicators to capture more nuanced market dynamics.
2. Model Ensemble: Implement ensemble learning techniques to combine predictions from multiple models, including LSTM, convolutional neural networks (CNNs), or traditional statistical models, to improve predictive accuracy and robustness.

**REFERENCES**

1.https://scholarworks.lib.csusb.edu/cgi/viewcontent.cgi?article=1435 &context=jitim

2. https://www.sas.com/en\_us/insights/analytics/machine-

learning.html

3. https://groww.in/us-stocks/googl

4. https://www.investopedia.com/terms/d/deep-learning.asp

5. <https://www.nasdaq.com/market-activity/stocks/google>

6.https://www.divaportal.org/smash/get/diva2:1672304/FULLTEXT01.pdf