**Stock Market Predictor**

****

**Submitted By:**

**Name- Sparshi Yadav Name- Vidhi Dabral**

**Roll No – 2016843 Roll No - 2016866**

PROJECT REPORT

BANASTHALI VIDYAPITH

**2022-2023**

# ACKNOWLЕDGЕMЕNT

DEPARTMENT OF COMPUTER SCIENCE

BANASTHALI VIDYAPITH

**ACKNOWLEDGEMENT**

I would like to express my deep gratitude and convey my heartfelt thanks to my professor Prof. KF Rehman for his ingenious ideas, tremendous help and cooperation. His valuable suggestions and guidance helped me to successfully complete my project.

I would also like to thank my friends as the cooperation and healthy criticism came handy and useful with them.

I would also like to thank my parents for their support and encouragement throughout my study.

Finally, I would like to thank all the above-mentioned people once again.

# TABLЕ OF CONTЕNTS

|  |  |
| --- | --- |
| **Title** | **Page Number** |
| Acknowledgement | 2 |
| Table of Contents | 3 |
| Introduction | 4 |
| Libraries used and steps involved | 5 |
| Concepts, Algorithm and Working | 6-8 |
| Features for User | 9 |
| Code | 10-11 |
| Result and Discussion | 12 |
| References | 13 |

# INTRODUCTION

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. If you are able to successfully predict the price of a stock, you could gain an immense amount of profit.

In this Project, we’ll build a Python deep learning model that will predict the future behavior of stock prices. We assume that the reader is familiar with the concepts of deep learning in Python, especially Long Short-Term Memory.

While predicting the actual price of a stock is an uphill climb, we can build a model that will predict whether the price will go up or down.

It’s important to note that there are always other factors that affect the prices of stocks, such as the political atmosphere and the market. However, we won’t focus on those factors for this project.

LSTMs are very powerful in sequence prediction problems because they’re able to store past information. This is important in our case because the previous price of a stock is crucial in predicting its future price.

We’ll kick off by importing NumPy for scientific computation, Matplotlib for plotting graphs, and Pandas to aide in loading and manipulating our datasets.

# LIBRARIES USED AND STEPS INVOLVED

The programming language used is Python.

The IDE used is Jupyter Notebook or Google Colab.

Various tools and libraries used include Pandas, Numpy, Keras, Sklearn and Dense and LSTM layers.

To perform mathematical operations Matplotlib and Math libraries are used.

The core steps involved in this Stock Marker Prediction are:

1) Getting and analyzing the dataset.

2) Training and Testing of the Model.

3) Using the concept of time-step.

4) Building the LSTM model.

5) Decreasing RMSE (root mean square error) using optimizer in order to predict the prices close to their actual values as precise as possible.

# CONCEPTS, ALGORITHM AND WORKING

To solve the problem of Vanishing and Exploding Gradients in a deep Recurrent Neural Network, many variations were developed. One of the most famous of them is the Long Short-Term Memory Network (LSTM). In concept, an LSTM recurrent unit tries to “remember” all the past knowledge that the network is seen so far and to “forget” irrelevant data. This is done by introducing different activation function layers called “gates” for different purposes. Each LSTM recurrent unit also maintains a vector called the Internal Cell State which conceptually describes the information that was chosen to be retained by the previous LSTM recurrent unit.

A Long Short-Term Memory Network consists of four different gates for different purposes as described below: -

1) Forget Gate(f): It determines to what extent to forget the previous data.

2) Input Gate(i): It determines the extent of information to be written onto the Internal Cell State.

3) Input Modulation Gate(g): It is often considered as a sub-part of the input gate and many literatures on LSTM’s do not even mention it and assume it inside the Input gate. It is used to modulate the information that the Input gate will write onto the Internal State Cell by adding non-linearity to the information and making the information Zero-mean. This is done to reduce the learning time as Zero-mean input has faster convergence. Although this gate’s actions are less important than the others and is often treated as a finesse-providing concept, it is good practice to include this gate into the structure of the LSTM unit.

4) Output Gate(o): It determines what output (next Hidden State) to

generate from the current Internal Cell State.

The basic work-flow of a Long Short-Term Memory Network is similar to the work-flow of a Recurrent Neural Network with only difference being that the Internal Cell State is also passed forward along with the Hidden State.

**Working of an LSTM Recurrent Unit:**

1) Take input the current input, the previous hidden state and the previous internal cell state.

2) Calculate the values of the four different gates by following the below steps:

* For each gate, calculate the parameterized vectors for the current input and the previous hidden state by element-wise multiplication with the concerned vector with the respective weights for each gate.
* Apply the respective activation function for each gate element-wise on the parameterized vectors. Below given is the list of the gates with the activation function to be applied for the gate.

Input Gate : Sigmoid Function

Forget Gate : Sigmoid Function

Output Gate : Sigmoid Function

Input Modulation Gate : Hyperbolic Tangent Function

3) Calculate the current internal cell state by first calculating the element-wise multiplication vector of the input gate and the input modulation gate, then calculate the element-wise multiplication vector of the forget gate and the previous internal cell state and then adding the two vectors.

4) Calculate the current hidden state by first taking the element-wise hyperbolic tangent of the current internal cell state vector and then performing element wise multiplication with the output gate.

The dataset which we have used has been taken from Yahoo and considers the stock prices of AAPL from the 01-01-2015 to 01-01-2020.

Further, our model uses a time step of 60, that means, it uses the first 60 values to generate the 61st value and then repeats it to use the values from 1-61 to generate the 62nd value and so on until all the data has been processes and the model is completely trained up. Finally, the RMSE value is calculated and optimizations performed to further enhance the model and make it ready for predictions.

**The working of the project is as follows:**

1) At first, we access the many previous years stock data of a particular company and analyze it.

2) After analyzing is done, the dataset is split into training and testing data sets in the ratio of nearly 80:20 in order to ensure that our model gets sufficient data to be trained upon so that it gives minimal error.

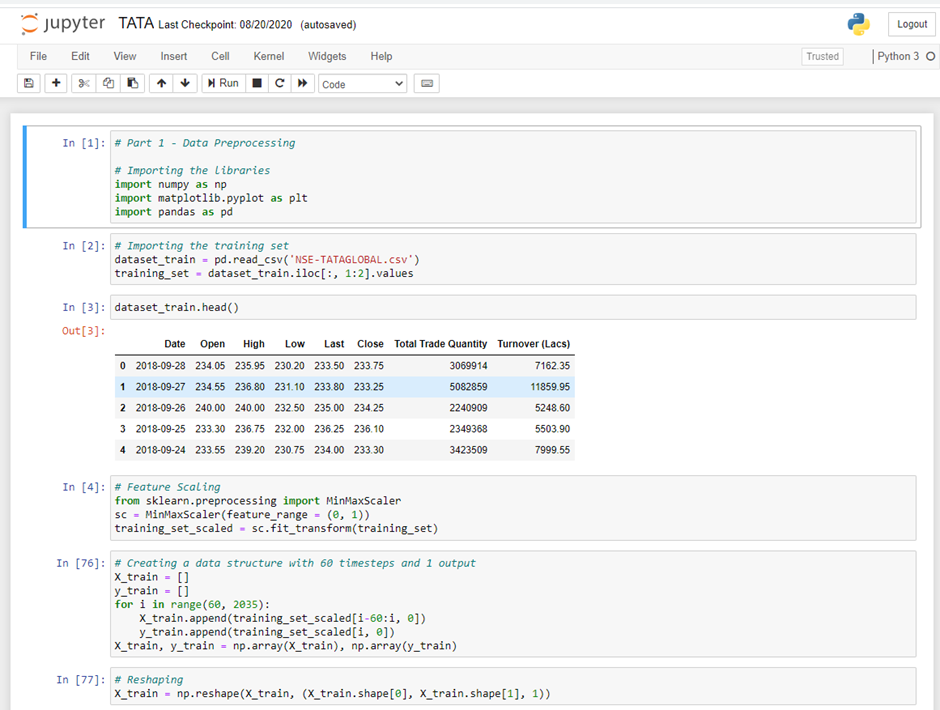
3) Scaling is applied to the training dataset and then the model is trained on the resulting scaled dataset. (concept of timestep is used)

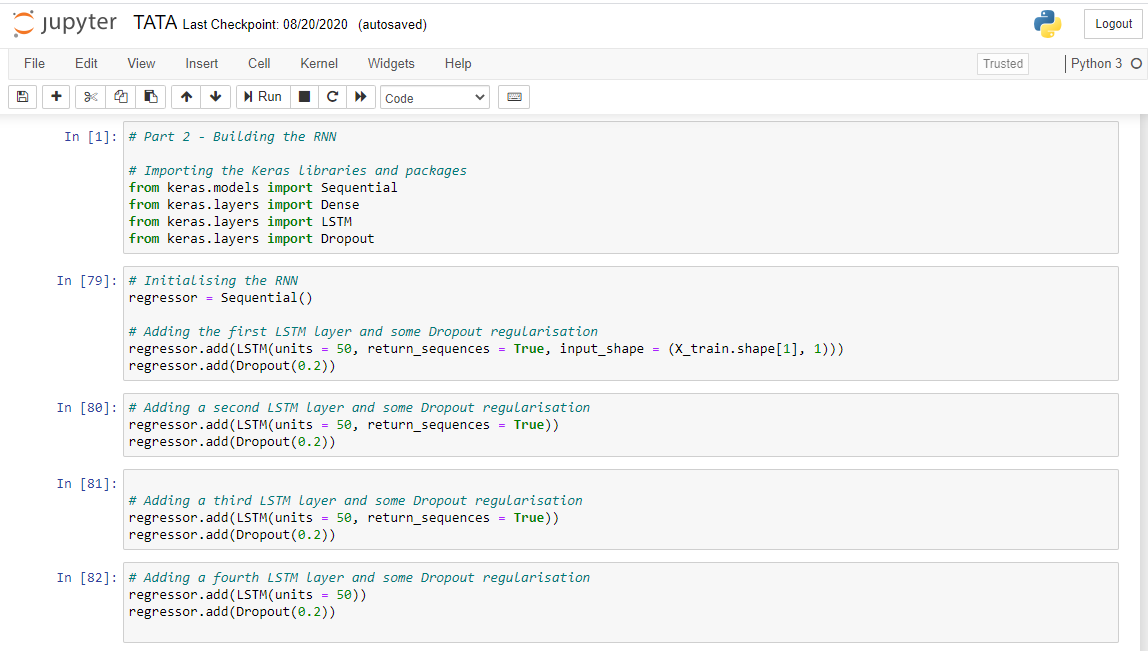
4) The model is now trained and is used for testing. Based on the testing we calculate the RMSE (error) and ensure it lies in the acceptable range.

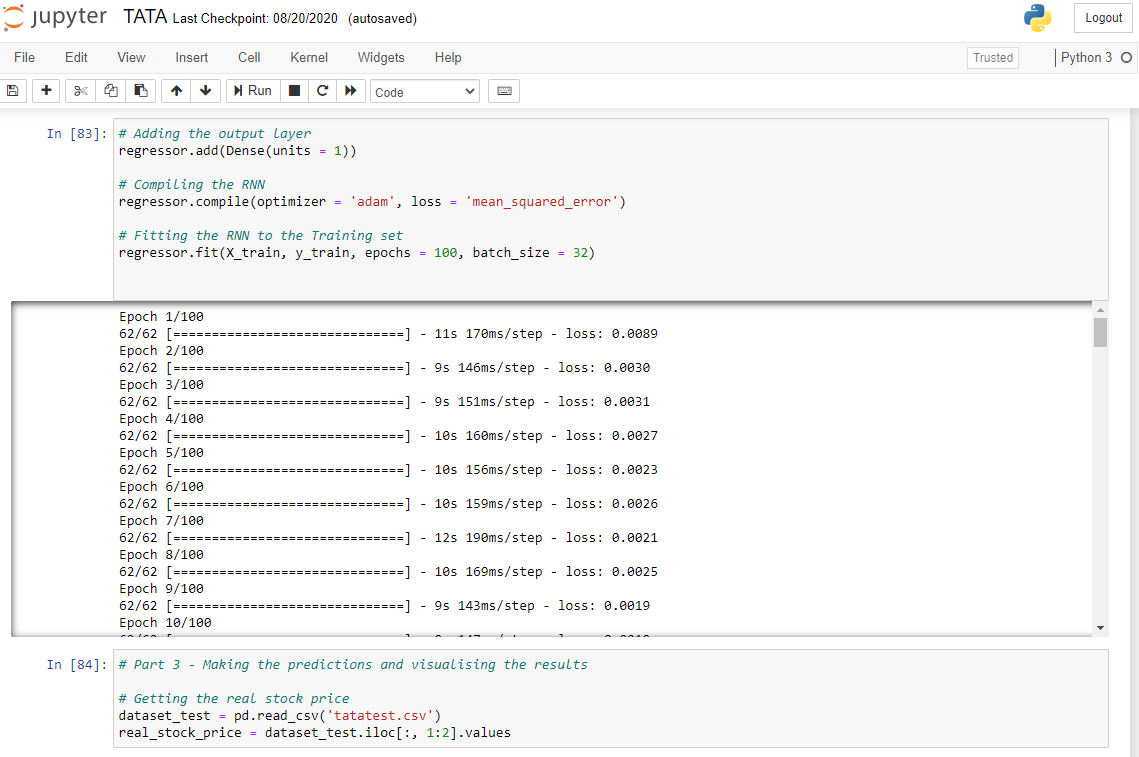
5) The model is now ready and is used to predict the future stock prices.

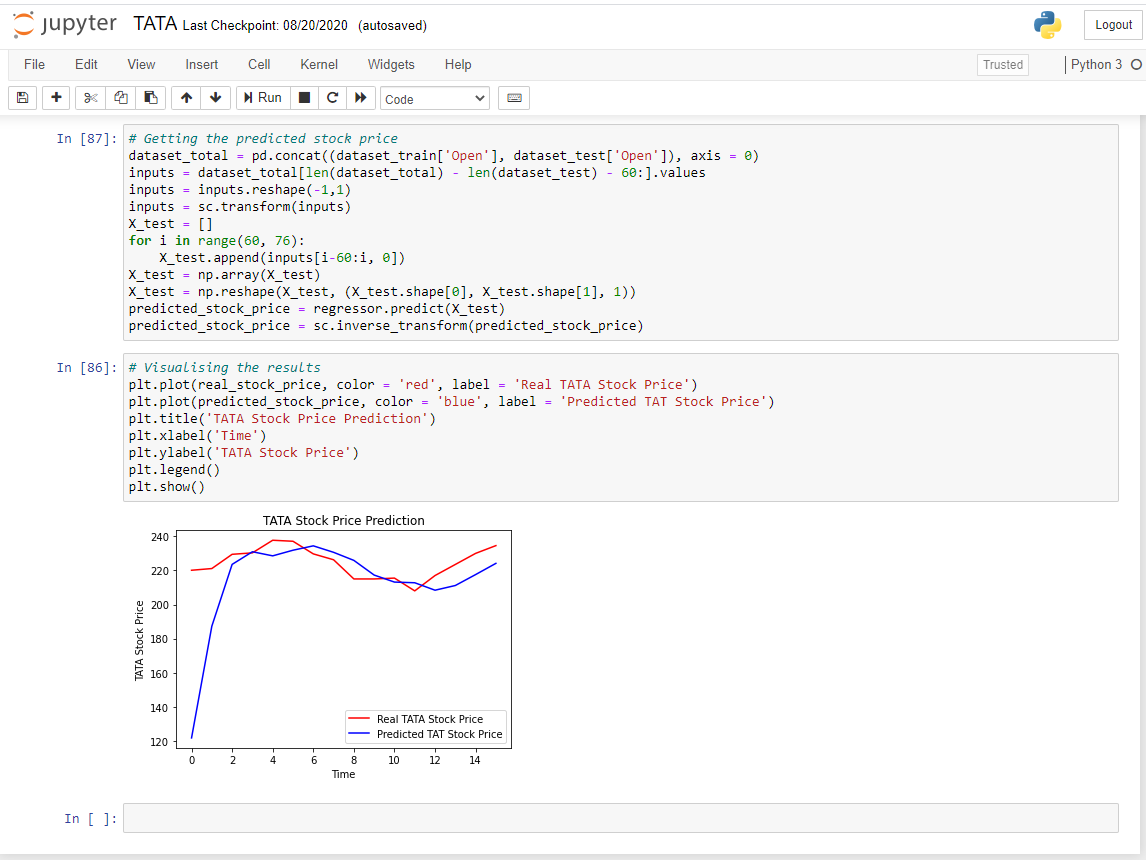
6) The program further compares the predicted price and the current price to guide the user on whether he/she should be investing in the stock or not.

# CODE









**RESULT AND DISCUSSION**

From the plot we can see that the real stock price went up while our model also predicted that the price of the stock will go up. This clearly shows how powerful LSTMs are for analysing time series and sequential data. There are a couple of other techniques of predicting stock prices such as moving averages, linear regression, K-Nearest Neighbours, ARIMA and Prophet. These are techniques that one can test on their own and compare their performance with the Keras LSTM.

# BIBLIOGRAPHY:

Text Sources:

1. https://colah.github.io/posts/2015-08-Understanding-LSTMs/

2. https://www.geeksforgeeks.org/long-short-term-memory-networks-explanation/

Video Sources:

1. https://www.youtube.com/watch?v=DooxDIRAkPA&t=4215s

2. https://www.edx.org/course/python-basics-for-data-science

3. https://www.udemy.com/course/complete-python-bootcamp/