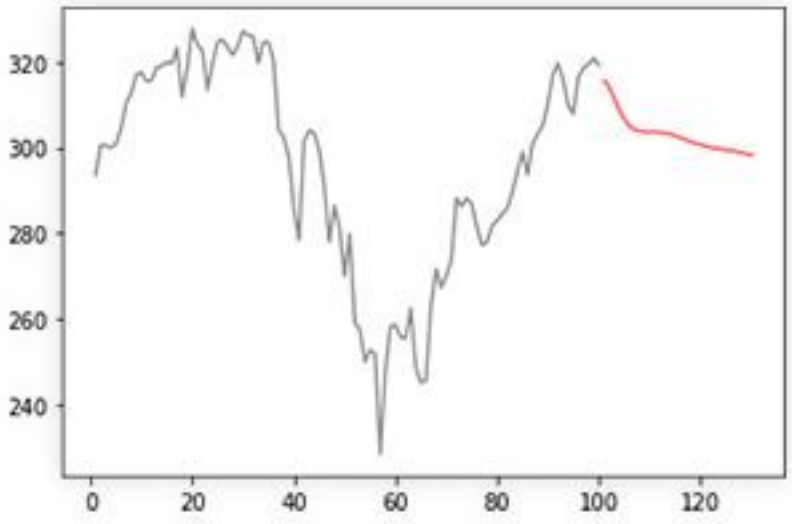
Project Report:

Stock Value Prediction

Using LSTM



Created by:
Jack Ilango
RA1811026010004

Project Title:

Stock Prices Prediction Using Stacked LSTM

Overview:

Stock values is very valuable but extremely hard to predict correctly for any human being on their own. This project seeks to solve the problem of **Stock Prices Prediction** by utilizes *Deep Learning models, Long-Short Term Memory (LSTM) Neural Network algorithm*, to predict future stock values. Historical data about the stock values that have been publicly listed by Quandl has been used in this project and I have used the stock value data of 'Tata Global Beverages'. This can be considered as a Time series analysis is a specialized branch of statistics used extensively in fields such as Econometrics & Operation Research. This is specifically designed time series problem for you and the challenge is to forecast traffic.

Aim/Objective:

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.

Technical Objective:

The technical objectives will be implemented in Python. The system must be able to access a list of historical prices. It must calculate **the estimated price of stock** based on the historical data for **the next 30 days**. It must also provide an *instantaneous visualization* of the market index in a neatly formatted Python-Based Web App.

```
In [1]: import pandas as pd

In [2]: df=pd.read_csv('AAPL.csv')

In [3]: df.head()
```

Out[3]:

	Unnamed: 0	symbol	date	close	high	low	open	volume	adjClose	adjHigh	adjLow	adjOpen	adjVolume	divCash	splitF
0	0	AAPL	2015-05-27 00:00:00+00:00	132.045	132.260	130.05	130.34	45833246	121.682558	121.880685	119.844118	120.111360	45833246	0.0	
1	1	AAPL	2015-05-28 00:00:00+00:00	131.780	131.950	131.10	131.86	30733309	121.438354	121.595013	120.811718	121.512076	30733309	0.0	
2	2	AAPL	2015-05-29 00:00:00+00:00	130.280	131.450	129.90	131.23	50884452	120.056069	121.134251	119.705890	120.931516	50884452	0.0	
3	3	AAPL	2015-06-01 00:00:00+00:00	130.535	131.390	130.05	131.20	32112797	120.291057	121.078960	119.844118	120.903870	32112797	0.0	
4	4	AAPL	2015-06-02 00:00:00+00:00	129.960	130.655	129.32	129.86	33667627	119.761181	120.401640	119.171406	119.669029	33667627	0.0	

Long Short-Term Memory: (LSTM)

Long Short-Term Memory (or) LSTMs are widely used for sequence prediction problems and have proven to be extremely effective. The reason they work so well is because LSTM is able to store past information that is important and forget the information that is not. LSTM has three types of gates:

1. The input gate: The input gate adds information to the cell state.
2. The forget gate: It removes the information that is no longer required by the model.
3. The output gate: Output Gate at LSTM selects the information to be shown as output.

Training was done under the following metrics and functions:

- Number of layers = 4
- Loss Function= Mean Square Error
- Optimizer = Adam
- Epochs = 100
- Batch size = 64

```
In [21]: ### Create the Stacked LSTM model
         from tensorflow.keras.models import Sequential
         from tensorflow.keras.layers import Dense
         from tensorflow.keras.layers import LSTM

In [22]: model=Sequential()
         model.add(LSTM(50,return_sequences=True,input_shape=(100,1)))
         model.add(LSTM(50,return_sequences=True))
         model.add(LSTM(50))
         model.add(Dense(1))
         model.compile(loss='mean_squared_error',optimizer='adam')

In [23]: model.summary()
         Model: "sequential"
         _________________________________________________________________
         Layer (type)                 Output Shape              Param #
         =================================================================
         lstm (LSTM)                  (None, 100, 50)           10400
         _________________________________________________________________
         lstm_1 (LSTM)                (None, 100, 50)           20200
         _________________________________________________________________
         lstm_2 (LSTM)                (None, 50)                20200
         _________________________________________________________________
         dense (Dense)                (None, 1)                 51
         =================================================================
         Total params: 50,851
         Trainable params: 50,851
         Non-trainable params: 0
         _________________________________________________________________
```

Output:

```
In [41]: plt.plot(day_new,scaler.inverse_transform(df1[1158:]))
         plt.plot(day_pred,scaler.inverse_transform(lst_output))
         plt.savefig('30daypredict.png')

Out[41]: [<matplotlib.lines.Line2D at 0x25cd7e8e670>]
```

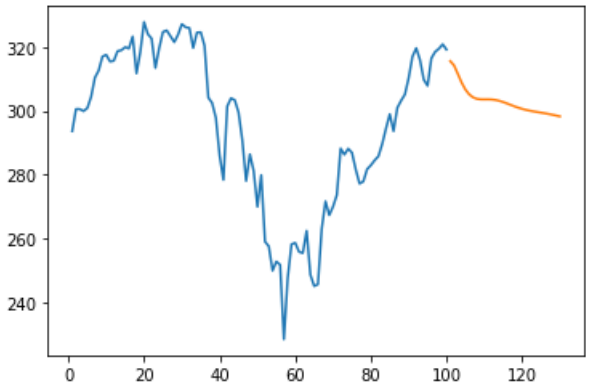


Figure 1: Predicted Graph For Next 30 Days

```
In [47]: plt.plot(df3)

Out[47]: [<matplotlib.lines.Line2D at 0x25cd7f281f0>]
```

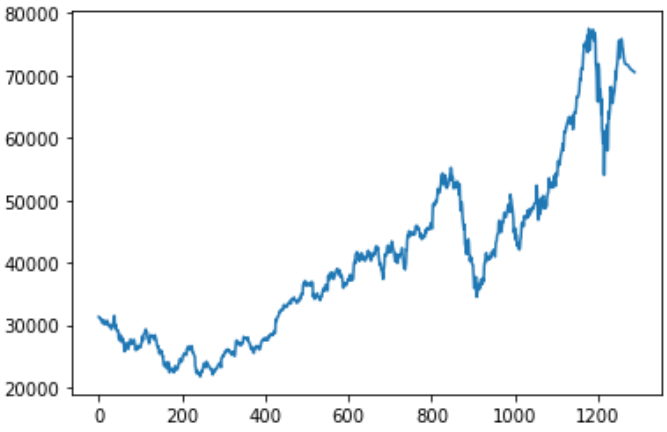


Figure 2: Full Dataset + Predicted Values

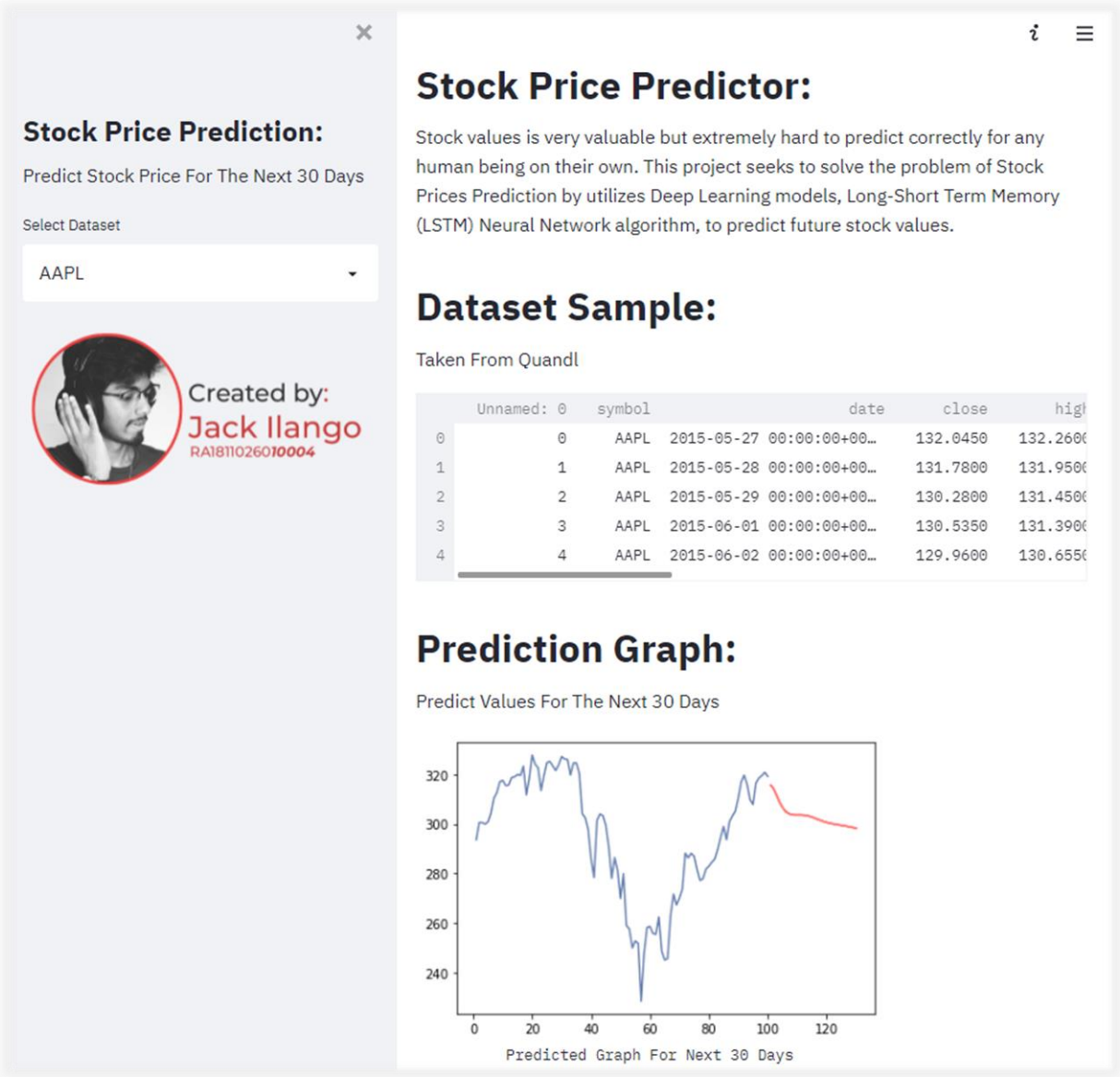


Figure 3: Stock Prediction WebApp

Conclusion:

The proposed Stock Price Predictor has been successfully trained by using LSTM learning model on the sample datasets and the Stock value prediction process has been successfully performed by the trained LSTM model being tested on the test data set.