Abstract

Lets say you are going to an exposition filled with paints and you are trying to find the special paint that has the more value. You will be looking at the shapes, colors, how ancient it is, how rare it is and how many people are willing to get that rare paint. Trying to invest to invest into the stock market is similar to look for that special paint. The question will be which company or shares that will make you get more profit. Investing in stocks or shares has always been a controversial topic. Because if there are people knowing exactly which company to invest, when to invest or which shares to buy now or in the future, those people will be the richest in the world. Fortunately, nowadays there is a hope that technology(AI) will help fullfil that goal in a way that will be described throughout this analysis.

Introduction:

AI can be found everywhere on apps for example when you are using Uber, google maps. Those use a technology called self organizing map which is just an algorithm that organize things in a way it will help you find the shortest route to go from point A to B. However, our interest will be in a different type of machine learning algorithm called neural network but some insights must be given. Predicting stock market price fall into the time series forecasting problem. In that problem the time is considered as a dependent continous variable for some reasons. For example, the price of a stock can be updated at any second and also the price of a stock or shares on a day may depend of the price of that stock the day before. Because of the dependencies between the prices and time, a sequential model which will be LSTM(Long short term memory) will be used with keras(Library that contains predefined models and neural network). LSTM network is a type of recurrent neural network that remembers long term dependencies between the data.

The steps will be defined as follow:

-Defining the variables use as input

-Statings the assumptions or common beliefs about the model

- Data preprocessing using sklearn and numpy which are common libraries used for this task
- Data manipulation

```
# Numpy is used for data manipulation
import numpy as np


# Matplot lib is used to visualize the data
import matplotlib.pyplot as plt

# Pandas to read to data
import pandas as pd

# sklearn for data preprocessing
from sklearn.preprocessing import Imputer
from sklearn.preprocessing import MinMaxScaler

from sklearn.metrics import mean_squared_error

#keras to create LSTM network and compile it later on
from keras.models import Sequential
from keras.layers import Dense, Dropout
from keras.callbacks import Callback
from keras.models import Sequential
from keras.layers import LSTM, Dense, Activation
```

```
import time
import tensorflow as tf

import types
```

```
Using TensorFlow backend.
```

There are 2 datasets. 1 for training and the other to test how well the model perform to predict future prices. All datasets comes from **finance.yahoo.com**

```
from google.colab import files
uploaded = files.upload()
```

Choose Files Google_Stoc..._Train.csv

- **Google_Stock_Price_Train.csv**(application/vnd.ms-excel) - 63488 bytes, last modified: 7/5/2018 - 100%

```
Saving Google_Stock_Price_Train.csv to Google_Stock_Price_Train.csv
```

```
df_data = pd.read_csv('Google_Stock_Price_Train.csv')
df_data
```

4	1/9/2012	322.04	322.29	309.46	620.76	11,688,800
5	1/10/2012	313.70	315.72	307.30	621.43	8,824,000
6	1/11/2012	310.59	313.52	309.40	624.25	4,817,800
7	1/12/2012	314.43	315.26	312.08	627.92	3,764,400
8	1/13/2012	311.96	312.30	309.37	623.28	4,631,800
9	1/17/2012	314.81	314.81	311.67	626.86	3,832,800
10	1/18/2012	312.14	315.82	309.90	631.18	5,544,000
11	1/19/2012	319.30	319.30	314.55	637.82	12,657,800
12	1/20/2012	294.16	294.40	289.76	584.39	21,231,800
13	1/23/2012	291.91	293.23	290.49	583.92	6,851,300
14	1/24/2012	292.07	292.74	287.92	579.34	6,134,400
15	1/25/2012	287.68	288.27	282.13	567.93	10,012,700
16	1/26/2012	284.92	286.17	281.22	566.54	6,476,500
17	1/27/2012	284.32	289.08	283.60	578.39	7,262,000
18	1/30/2012	287.95	288.92	285.63	576.11	4,678,400
19	1/31/2012	290.41	290.91	286.50	578.52	4,300,700
20	2/1/2012	291.38	291.66	288.49	579.24	4,658,700
21	2/2/2012	291.34	292.11	289.95	583.51	4,847,400
22	2/3/2012	294.23	297.42	292.93	594.7	6,360,700
23	2/6/2012	296.39	304.27	295.90	607.42	7,386,700
24	2/7/2012	302.44	303.56	300.75	605.11	4,199,700
25	2/8/2012	303.18	304.53	301.24	608.18	3,686,400
26	2/9/2012	304.87	306.10	303.36	609.79	4,546,300
27	2/10/2012	302.81	302.93	300.87	604.25	4,667,700
28	2/13/2012	304.11	305.77	303.87	610.52	3,646,100
29	2/14/2012	304.63	304.86	301.25	608.09	3,620,900
...	...	...	...	...	...	...
1228	11/17/2016	766.92	772.70	764.23	771.23	1,304,000
1229	11/18/2016	771.37	775.00	760.00	760.54	1,547,100
1230	11/21/2016	762.61	769.70	760.60	769.2	1,330,600
1231	11/22/2016	772.63	776.96	767.00	768.27	1,593,100
1232	11/23/2016	767.73	768.28	755.25	760.99	1,478,400

1233	11/25/2016	764.26	765.00	760.52	761.68	587,400
1234	11/28/2016	760.00	779.53	759.80	768.24	2,188,200
1235	11/29/2016	771.53	778.50	768.24	770.84	1,616,600
1236	11/30/2016	770.07	772.99	754.83	758.04	2,392,900
1237	12/1/2016	757.44	759.85	737.03	747.92	3,017,900
1238	12/2/2016	744.59	754.00	743.10	750.5	1,452,500
1239	12/5/2016	757.71	763.90	752.90	762.52	1,394,200
1240	12/6/2016	764.73	768.83	757.34	759.11	1,690,700
1241	12/7/2016	761.00	771.36	755.80	771.19	1,761,000
1242	12/8/2016	772.48	778.18	767.23	776.42	1,488,100
1243	12/9/2016	780.00	789.43	779.02	789.29	1,821,900
1244	12/12/2016	785.04	791.25	784.35	789.27	2,104,100
1245	12/13/2016	793.90	804.38	793.34	796.1	2,145,200
1246	12/14/2016	797.40	804.00	794.01	797.07	1,704,200
1247	12/15/2016	797.34	803.00	792.92	797.85	1,626,500
1248	12/16/2016	800.40	800.86	790.29	790.8	2,443,800
1249	12/19/2016	790.22	797.66	786.27	794.2	1,232,100
1250	12/20/2016	796.76	798.65	793.27	796.42	951,000
1251	12/21/2016	795.84	796.68	787.10	794.56	1,211,300
1252	12/22/2016	792.36	793.32	788.58	791.26	972,200
1253	12/23/2016	790.90	792.74	787.28	789.91	623,400
1254	12/27/2016	790.68	797.86	787.66	791.55	789,100
1255	12/28/2016	793.70	794.23	783.20	785.05	1,153,800
1256	12/29/2016	783.33	785.93	778.92	782.79	744,300
1257	12/30/2016	782.75	782.78	770.41	771.82	1,770,000

1258 rows × 6 columns

```
df_data.iloc[:,2].values
```

```
array([332.83, 333.87, 330.75, ..., 794.23, 785.93, 782.78])
```

Let **X** represents the highest daily price from 05/24/2019 to 06/24/2019. X is also the input vector that will be feed into the neural network.

```
X = df_data.iloc[:,[2]].values
```

```
training_set = []

for i in range(1237):
    training_set.append(X[i])

training_set
```

```
[array([332.83]),
 array([333.87]),
 array([330.75]),
 array([328.77]),
 array([322.29]),
 array([315.72]),
 array([313.52]),
 array([315.26]),
 array([312.3]),
 array([314.81]),
 array([315.82]),
 array([319.3]),
 array([294.4]),
 array([293.23]),
 array([292.74]),
 array([288.27]),
 array([286.17]),
 array([289.08]),
 array([288.92]),
 array([290.91]),
 array([291.66]),
 array([292.11]),
 array([297.42]),
 array([304.27]),
 array([303.56]),
 array([304.53]),
 array([306.1]),
 array([302.93]),
 array([305.77]),
 array([304.86]),
 array([305.32]),
 array([303.27]),
 array([302.68]),
 array([307.79]),
 array([307.24]),
 array([302.83]),
 array([304.68]),
 array([305.04]),
 array([308.73]),
 array([311.63]),
 array([311.68]),
 array([310.83]),
 array([310.08]),
 array([303.27]),
 array([304.45]),
 array([304.61]),
 array([304.81]),
 array([302.37]),
 array([307.77]),
 array([310.23]),
 array([310.59]),
 array([311.79]),
 array([317.45]),
 array([316.84]),
 array([322.49]),
 array([323.19]),
 array([323.04]),
```

```
        array([323.53]),
        array([325.53]),
        array([328.07]),
        array([327.07]),
        array([325.52]),
        array([322.54]),
        array([322.77]),
        array([318.31]),
        array([317.03]),
        array([316.48]),
        array([316.07]),
        array([316.81]),
        array([325.35]),
        array([323.28]),
        array([310.74]),
        array([307.69]),
        array([305.26]),
        array([306.98]),
        array([303.29]),
        array([298.11]),
        array([302.18]),
        array([304.53]),
        array([307.85]),
        array([307.22]),
        array([306.89]),
        array([304.66]),
        array([302.92]),
        array([306.27]),
        array([302.81]),
        array([304.14]),
        array([307.3]),
        array([307.04]),
        array([306.94]),
        array([306.13]),
        array([303.11]),
        array([306.35]),
        array([313.87]),
        array([317.73]),
        array([315.03]),
        array([306.7]),
        array([305.76]),
        array([303.66]),
        array([304.82]),
        array([299.74]),
        array([298.45]),
        array([294.84]),
        array([293.9]),
        array([285.26]),
        array([289.16]),
        array([287.99]),
        array([289.9]),
        array([292.85]),
        array([289.42]),
        array([291.57]),
        array([284.09]),
        array([282.44]),
        array([281.48]),
        array([281.21]),
```

```
array([286.03]),
array([291.05]),
array([288.92]),
array([288.84]),
array([284.67]),
array([282.98]),
array([282.24]),
array([285.92]),
array([282.06]),
array([288.98]),
array([290.41]),
array([293.11]),
array([298.91]),
array([295.65]),
array([293.2]),
array([295.11]),
array([287.85]),
array([284.9]),
array([288.49]),
array([288.51]),
array([289.25]),
array([290.76]),
array([298.12]),
array([305.33]),
array([308.02]),
array([307.81]),
array([305.54]),
array([307.28]),
array([316.31]),
array([320.1]),
array([317.06]),
array([318.56]),
array([317.82]),
array([320.66]),
array([323.48]),
array([320.93]),
array([321.73]),
array([321.98]),
array([319.92]),
array([328.84]),
array([335.17]),
array([335.87]),
array([336.06]),
array([337.36]),
array([338.17]),
array([337.73]),
array([339.03]),
array([338.97]),
array([338.95]),
array([334.75]),
array([337.54]),
array([343.21]),
array([342.41]),
array([343.]),
array([341.22]),
array([341.97]),
array([348.64]),
array([354.8]),
```

```
        array([354.8]),
        array([355.07]),
        array([349.02]),
        array([346.16]),
        array([353.18]),
        array([355.17]),
        array([355.11]),
        array([357.99]),
        array([362.92]),
        array([364.32]),
        array([366.09]),
        array([373.62]),
        array([381.02]),
        array([379.2]),
        array([380.]),
        array([378.23]),
        array([381.07]),
        array([381.56]),
        array([380.53]),
        array([383.51]),
        array([385.74]),
        array([380.36]),
        array([379.24]),
        array([372.37]),
        array([377.83]),
        array([376.03]),
        array([370.53]),
        array([372.1]),
        array([376.76]),
        array([378.29]),
        array([352.03]),
        array([341.04]),
        array([342.38]),
        array([342.22]),
        array([339.73]),
        array([340.24]),
        array([339.23]),
        array([344.16]),
        array([346.48]),
        array([342.15]),
        array([341.97]),
        array([337.85]),
        array([334.49]),
        array([332.92]),
        array([333.65]),
        array([332.55]),
        array([329.85]),
        array([328.77]),
        array([325.29]),
        array([333.21]),
        array([337.73]),
        array([333.65]),
        array([333.75]),
        array([332.25]),
        array([336.24]),
        array([341.18]),
        array([345.65]),
        array([348.3]),
```

```
array([351.63]),
array([346.46]),
array([345.95]),
array([346.51]),
array([347.14]),
array([344.53]),
array([349.65]),
array([350.44]),
array([356.9]),
array([352.59]),
array([359.61]),
array([363.19]),
array([360.15]),
array([360.97]),
array([358.07]),
array([356.25]),
array([355.11]),
array([353.1]),
array([352.14]),
array([353.96]),
array([362.14]),
array([364.6]),
array([369.35]),
array([368.31]),
array([366.78]),
array([367.8]),
array([371.11]),
array([369.83]),
array([369.71]),
array([366.13]),
array([360.82]),
array([358.48]),
array([355.05]),
array([351.35]),
array([373.1]),
array([377.]),
array([377.82]),
array([376.39]),
array([377.06]),
array([379.05]),
array([377.4]),
array([386.85]),
array([383.8]),
array([384.12]),
array([385.04]),
array([387.95]),
array([391.87]),
array([390.04]),
array([392.48]),
array([391.21]),
array([392.9]),
array([395.15]),
array([401.99]),
array([402.97]),
array([401.22]),
array([399.13]),
array([402.7]),
array([396.49]),
```

```
        array([400.87]),
        array([401.99]),
        array([402.06]),
        array([409.88]),
        array([418.51]),
        array([420.42]),
        array([416.75]),
        array([415.9]),
        array([418.28]),
        array([414.39]),
        array([413.79]),
        array([411.95]),
        array([408.62]),
        array([404.86]),
        array([408.1]),
        array([407.23]),
        array([406.93]),
        array([406.1]),
        array([408.09]),
        array([405.48]),
        array([401.99]),
        array([401.18]),
        array([399.63]),
        array([405.89]),
        array([405.58]),
        array([401.37]),
        array([392.03]),
        array([388.32]),
        array([390.41]),
        array([394.7]),
        array([395.07]),
        array([394.57]),
        array([397.01]),
        array([396.51]),
        array([393.94]),
        array([391.43]),
        array([400.22]),
        array([400.48]),
        array([406.23]),
        array([407.47]),
        array([406.71]),
        array([402.36]),
        array([409.81]),
        array([412.27]),
        array([410.82]),
        array([415.72]),
        array([421.82]),
        array([429.32]),
        array([430.32]),
        array([435.31]),
        array([438.19]),
        array([438.63]),
        array([439.59]),
        array([442.69]),
        array([456.48]),
        array([458.27]),
        array([455.04]),
```

```
 array([458.58]),
 array([454.13]),
 array([452.96]),
 array([443.32]),
 array([437.77]),
 array([444.4]),
 array([437.36]),
 array([437.81]),
 array([436.86]),
 array([435.35]),
 array([433.67]),
 array([433.1]),
 array([431.55]),
 array([438.36]),
 array([443.84]),
 array([441.84]),
 array([441.35]),
 array([438.19]),
 array([440.7]),
 array([443.05]),
 array([448.82]),
 array([453.72]),
 array([448.82]),
 array([443.28]),
 array([436.52]),
 array([438.2]),
 array([437.36]),
 array([440.69]),
 array([439.27]),
 array([444.37]),
 array([443.84]),
 array([442.92]),
 array([446.03]),
 array([451.44]),
 array([454.77]),
 array([453.87]),
 array([458.69]),
 array([459.78]),
 array([462.27]),
 array([462.25]),
 array([461.7]),
 array([458.28]),
 array([449.79]),
 array([454.63]),
 array([454.8]),
 array([453.45]),
 array([446.75]),
 array([443.33]),
 array([445.74]),
 array([446.13]),
 array([446.58]),
 array([450.59]),
 array([451.81]),
 array([451.07]),
 array([453.16]),
 array([447.55]),
 array([446.25]),
 array([446.09]),
```