Stock Market Analysis and Prediction

Smriti Gupta – A2345918059
Vivisha Singh- A2345918014

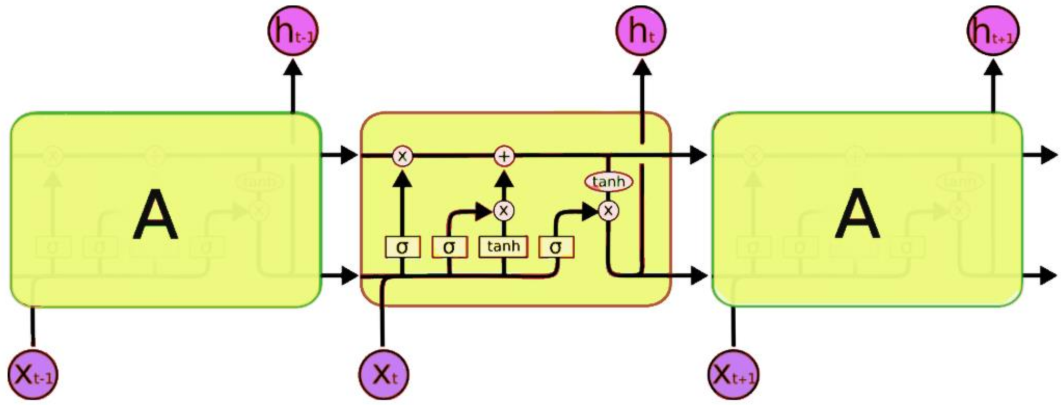
OVERVIEW

- A correct prediction of stocks can lead to huge profits for the seller and the broker. Frequently, it is brought out that prediction is chaotic rather than random, which means it can be predicted by carefully analyzing the history of respective stock market.
- Machine learning is an efficient way to represent such processes. It predicts a market value close to the tangible value, thereby increasing the accuracy.
- Introduction of machine learning to the area of stock prediction has appealed to many researches because of its efficient and accurate measurements . The vital part of machine learning is the dataset used.
- The dataset should be as concrete as possible because a little change in the data can perpetuate massive changes in the outcome.

ABOUT THE PROJECT

- In this project, supervised machine learning is employed on a dataset obtained from TATA NSE Global Beverages Limited.
- This dataset comprises of the following variables: Date, Open, High, Low, Last, Close, Total Trade Quantity, Turnover(Lacs).
- Open, close, low and high are different bid prices for the stock at separate times with nearly direct names.
- The different model is then tested on the test data.
- As the RMSE *(Root Mean Square Error)* value of LSTM *(Long Short-Term Memory)*model is very low comparison to other models. We use LSTM model to make stock prediction.
- LSTM model is able to store past information that is important, and forget the information that is not.

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2018-10-08	208.00	222.25	206.85	216.00	215.15	4642146.0	10062.83
1	2018-10-05	217.00	218.60	205.90	210.25	209.20	3519515.0	7407.06
2	2018-10-04	223.50	227.80	216.15	217.25	218.20	1728786.0	3815.79
3	2018-10-03	230.00	237.50	225.75	226.45	227.60	1708590.0	3960.27
4	2018-10-01	234.55	234.60	221.05	230.30	230.90	1534749.0	3486.05
5	2018-09-28	234.05	235.95	230.20	233.50	233.75	3069914.0	7162.35
6	2018-09-27	234.55	236.80	231.10	233.80	233.25	5082859.0	11859.95
7	2018-09-26	240.00	240.00	232.50	235.00	234.25	2240909.0	5248.60
8	2018-09-25	233.30	236.75	232.00	236.25	236.10	2349368.0	5503.90
9	2018-09-24	233.55	239.20	230.75	234.00	233.30	3423509.0	7999.55



SOURCE CODE

1. Imports:

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

import matplotlib.pyplot as plt
%matplotlib inline
from matplotlib.pylab import rcParams
rcParams['figure.figsize']=20,10

from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler(feature_range=(0,1))

import seaborn as sns
sns.set_style('darkgrid')
```

2. Read the dataset:

In [2]:
```
df=pd.read_csv("NSE-TATA.csv")
df.head()
```

Out[2]:

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2018-10-08	208.00	222.25	206.85	216.00	215.15	4642146.0	10062.83
1	2018-10-05	217.00	218.60	205.90	210.25	209.20	3519515.0	7407.06
2	2018-10-04	223.50	227.80	216.15	217.25	218.20	1728786.0	3815.79
3	2018-10-03	230.00	237.50	225.75	226.45	227.60	1708590.0	3960.27
4	2018-10-01	234.55	234.60	221.05	230.30	230.90	1534749.0	3486.05

SOURCE CODE

3. Analyze the closing prices from data frame:

In [13]:
```
df["Date"]=pd.to_datetime(df.Date,format="%Y-%m-%d")
df.index=df['Date']

plt.figure(figsize=(16,8))
plt.plot(df["Close"],label='Close Price history')
```

Out[13]: [<matplotlib.lines.Line2D at 0x22837bb7a90>]

SOURCE CODE

4. Sort the dataset on date time and filter “Date” and “Close” columns:

In [6]:
```
from keras.models import Sequential
from keras.layers import LSTM,Dropout,Dense
```

In [9]:
```
data=df.sort_index(ascending=True,axis=0)
new_dataset=pd.DataFrame(index=range(0,len(df)),columns=['Date','Close'])
for i in range(0,len(data)):
    new_dataset["Date"][i]=data['Date'][i]
    new_dataset["Close"][i]=data["Close"][i]
data.head()
```

Out[9]:

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
Date								
2013-10-08	2013-10-08	157.00	157.80	155.20	155.8	155.80	1720413.0	2688.94
2013-10-09	2013-10-09	155.70	158.20	154.15	155.3	155.55	2049580.0	3204.49
2013-10-10	2013-10-10	156.00	160.80	155.85	160.3	160.15	3124853.0	4978.80
2013-10-11	2013-10-11	161.15	163.45	159.00	159.8	160.05	1880046.0	3030.76
2013-10-14	2013-10-14	160.85	161.45	157.70	159.3	159.45	1281419.0	2039.09

In [10]:
```
new_dataset.index=new_dataset.Date
new_dataset.drop("Date",axis=1,inplace=True)
new_dataset.head()
```

Out[10]:

	Close
Date	
2013-10-08	155.8
2013-10-09	155.55
2013-10-10	160.15
2013-10-11	160.05
2013-10-14	159.45

SOURCE CODE

5. Normalize the new filtered dataset:

In [11]:

```
final_dataset=new_dataset.values
train_data=final_dataset[0:987,:]
valid_data=final_dataset[987:,:]

scaler=MinMaxScaler(feature_range=(0,1))
scaled_data=scaler.fit_transform(final_dataset)

x_train_data,y_train_data=[],[]

for i in range(60,len(train_data)):
    x_train_data.append(scaled_data[i-60:i,0])
    y_train_data.append(scaled_data[i,0])

x_train_data,y_train_data=np.array(x_train_data),np.array(y_train_data)

x_train_data=np.reshape(x_train_data,(x_train_data.shape[0],x_train_data.shape[1],1))
```

SOURCE CODE

6. Build and train the LSTM model:

In [12]:

```
lstm_model=Sequential()
lstm_model.add(LSTM(units=50,return_sequences=True,input_shape=(x_train_data.shape[1],1)))
lstm_model.add(LSTM(units=50))
lstm_model.add(Dense(1))

lstm_model.compile(loss='mean_squared_error',optimizer='adam')
lstm_model.fit(x_train_data,y_train_data,epochs=1,batch_size=1,verbose=2)

inputs_data=new_dataset[len(new_dataset)-len(valid_data)-60:].values
inputs_data=inputs_data.reshape(-1,1)
inputs_data=scaler.transform(inputs_data)
```

```
927/927 - 27s - loss: 0.0011
```

SOURCE CODE

7. Take a sample of a dataset to make stock predictions using the LSTM model:

In [14]:
```
X_test=[]
for i in range(60,inputs_data.shape[0]):
    X_test.append(inputs_data[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_closing_price=lstm_model.predict(X_test)
predicted_closing_price=scaler.inverse_transform(predicted_closing_price)
```

8. Save the LSTM model:

In [15]:
```
lstm_model.save("saved_lstm_model.h5")
```

SOURCE CODE

9. Visualize the predicted stock costs with actual stock costs:

In [16]:

```
train_data=new_dataset[:987]
valid_data=new_dataset[987:]
valid_data['Predictions'] = predicted_closing_price
plt.plot(train_data["Close"])
plt.plot(valid_data[['Close',"Predictions"]])
```

Out[16]:

```
[<matplotlib.lines.Line2D at 0x21af27b3ee0>,
 <matplotlib.lines.Line2D at 0x21af27b3fa0>]
```

THE PROS AND CONS:

PROS	CONS
▪ Minimize emotional trading. ▪ Allows for back testing ▪ Preserves the trader's discipline ▪ Allows multiple accounts	▪ Mechanical failures can happen ▪ Requires the monitoring of functionality ▪ Can perform poorly

FUTURE SCOPE:

Future scope of this project will involve adding more parameters and factors like the financial ratios, multiple instances, etc. The more the parameters are taken into account more will be the accuracy. The algorithms can also be applied for analyzing the contents of public comments and thus determine patterns/relationships between the customer and the corporate employee. The use of traditional algorithms and data mining techniques can also help predict the corporation’s performance structure as a whole.

Thank You