Stock Market Prediction Using Machine Learning and Python

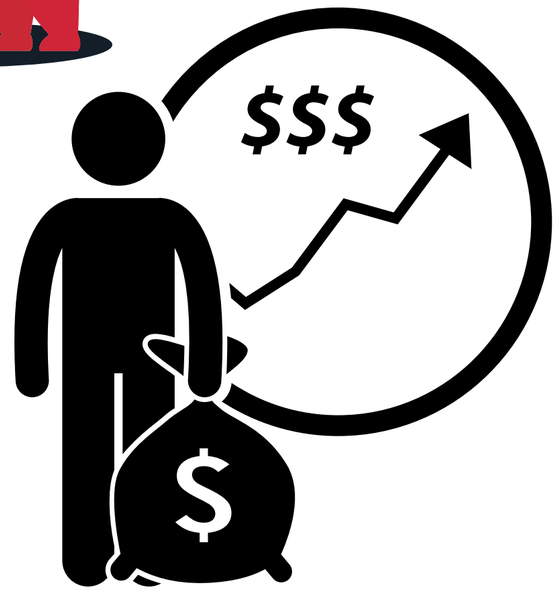
-- TEAM OUTLAW,
VVCE Mysore

OUTLAW

- MANOJ M
- MUYEEN BAIG
- LAKSHIN D K
- MANOHAR S N
- NAGSHARAN A
- AKHILESH S
- MOHAN PRAKASH V

SUMMARY

- 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
- We use a Machine Learning (ML) approach that will be trained from the available stocks data and gain intelligence and then uses the acquired knowledge for an accurate prediction.
- Recurrent Neural Networks(RNN) and long short term memory to predict stock prices for the large and small capitalizations and in the three different markets, employing prices with both daily and up-to-the-minute frequencies.
- We Use the data Visualization technique to compare the Actual and the predicted stock prices



PURPOSE

- The main purpose of the prediction is to reduce uncertainty associated to investment decision making. Stock Market follows the random walk, which implies that the best prediction you can have about tomorrow’s value is today’s value.
- Accuracy plays an important role in stock market prediction.
- The successful prediction will maximize the benefit of the customer.

DATA SOURCES

Date	Open	High	Low	Close	Volume
01-03-2012	325.25	332.83	324.97	663.59	73,80,500
01-04-2012	331.27	333.87	329.08	666.45	57,49,400
01-05-2012	329.83	330.75	326.89	657.21	65,90,300
01-06-2012	328.34	328.77	323.68	648.24	54,05,900
01-09-2012	322.04	322.29	309.46	620.76	1,16,88,800
01-10-2012	313.7	315.72	307.3	621.43	88,24,000
01-11-2012	310.59	313.52	309.4	624.25	48,17,800
01-12-2012	314.43	315.26	312.08	627.92	37,64,400
1/13/2012	311.96	312.3	309.37	623.28	46,31,800
1/17/2012	314.81	314.81	311.67	626.86	38,32,800
1/18/2012	312.14	315.82	309.9	631.18	55,44,000
1/19/2012	319.3	319.3	314.55	637.82	1,26,57,800
1/20/2012	294.16	294.4	289.76	584.39	2,12,31,800
1/23/2012	291.91	293.23	290.49	583.92	68,51,300
1/24/2012	292.07	292.74	287.92	579.34	61,34,400
1/25/2012	287.68	288.27	282.13	567.93	1,00,12,700
1/26/2012	284.92	286.17	281.22	566.54	64,76,500
1/27/2012	284.32	289.08	283.6	578.39	72,62,000
1/30/2012	287.95	288.92	285.63	576.11	46,78,400
1/31/2012	290.41	290.91	286.5	578.52	43,00,700

The historical stock data is collected from Google stock price this historical data is used to prediction of future stock prices

Analytics

#raw data

```
path='Google_Stock_Price_Train.csv'
stock_data = pd.read_csv(path,index_col="Date",parse_dates=True)
stock_data.head(10)
```

#data pre-processing

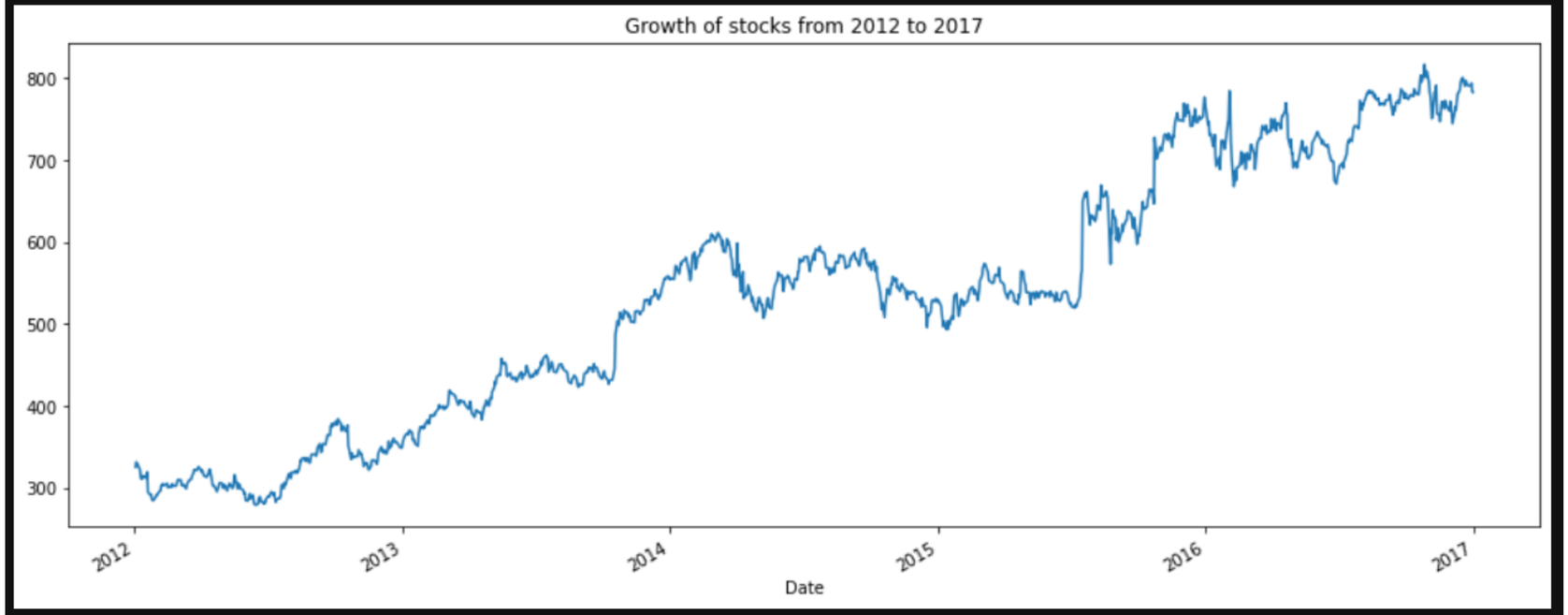
```
stock_data.isnull().sum()
stock_data.isna().any()
#no nan or missing data hence data is cleaned
```

```
Open      False
High      False
Low       False
Close     False
Volume    False
dtype: bool
```

Analytics

#growth of stock from 2012-2017

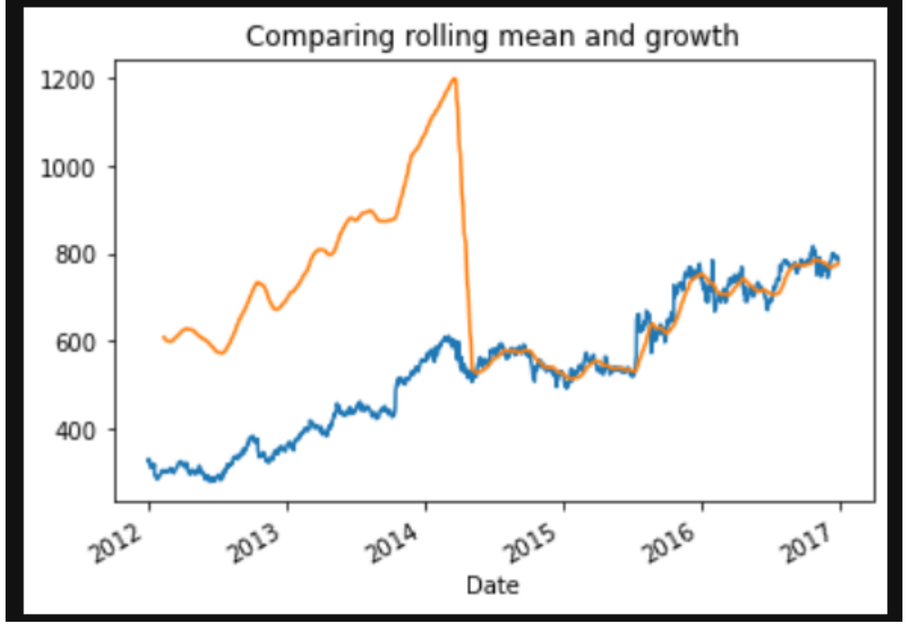
```
stock_data['Open'].plot(figsize=(16,6))
plt.title('Growth of stocks from 2012 to 2017')
plt.show()
```



Analytics

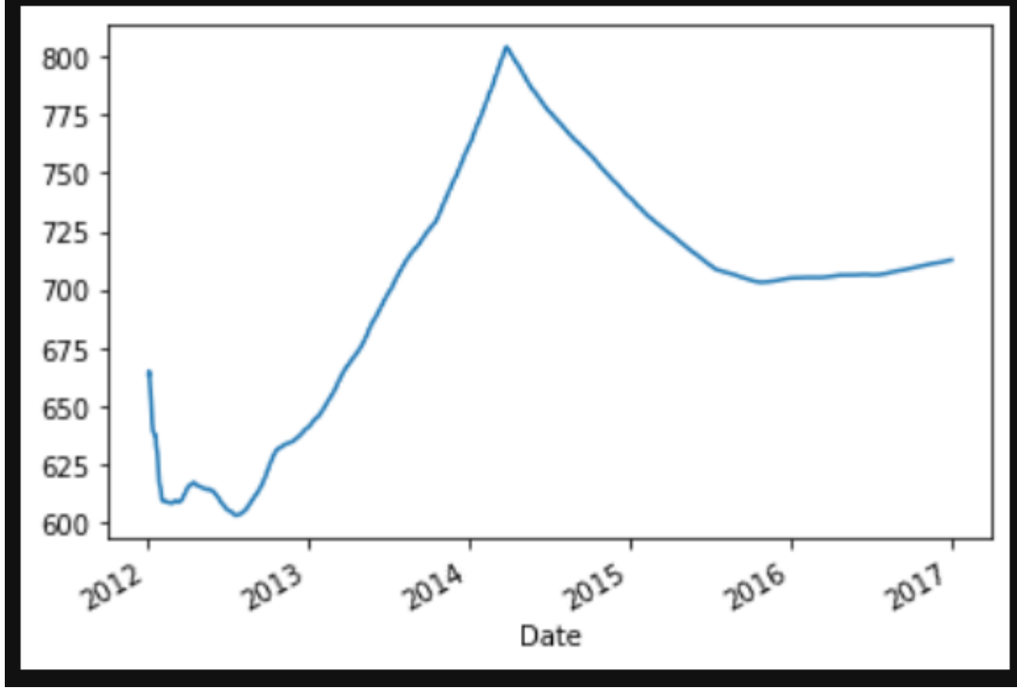
#rolling mean and growth

```
stock_data['Open'].plot()
stock_data.rolling(window=30).mean()['Close'].plot()
plt.title("Comparing rolling mean and growth")
plt.show()
```



#min_period_1

```
stock_data['Close'].expanding(min_periods=1).mean().plot()
plt.show()
```



Analytics

#building RNN

```
regression = Sequential()
```

```
#Adding the first layer and some Dropout regularisation
regression.add(LSTM(units=50,return_sequences=True ,input_shape=(X_train.shape[1],1)))
regression.add(Dropout(0.2))

#Adding the second layer and some Dropout regularisation
regression.add(LSTM(units=50,return_sequences=True ))
regression.add(Dropout(0.2))

#Adding the third layer and some Dropout regularisation
regression.add(LSTM(units=50,return_sequences=True ))
regression.add(Dropout(0.2))

#Adding the third layer and some Dropout regularisation
regression.add(LSTM(units=50))
regression.add(Dropout(0.2))

#Adding output layer
regression.add(Dense(units=1))
```

Analytics

#training RNN

```
#compiling the rnn
regression.compile(optimizer='adam',loss='mean_squared_error')

#fitting the rnn into Training set
regression.fit(X_train,Y_train,epochs=100,batch_size=32)
```

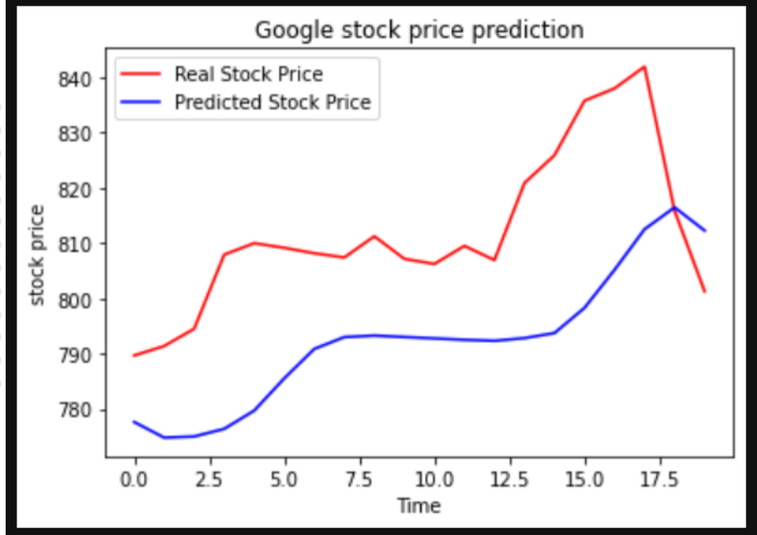
#building RNN

```
regression = Sequential()

#Adding the first layer and some Dropout regularisation
regression.add(LSTM(units=50,return_sequences=True ,input_shape=(X_train.shape[1],1)))
regression.add(Dropout(0.2))

#Adding the second layer and some Dropout regularisation
regression.add(LSTM(units=50,return_sequences=True ))
regression.add(Dropout(0.2))

#Adding the third layer and some Dropout regularisation
regression.add(LSTM(units=50,return_sequences=True ))
regression.add(Dropout(0.2))

#Adding the third layer and some Dropout regularisation
regression.add(LSTM(units=50))
regression.add(Dropout(0.2))

#Adding output layer
regression.add(Dense(units=1))
```

Analytics

#output

```
plt.plot(real_stock_price,color='red',label="Real Stock Price")
plt.plot(predicted_stock_price,color='blue',label="Predicted Stock Price")
plt.title("Google stock price prediction")
plt.xlabel("Time")
plt.ylabel("stock price")
plt.legend()
plt.show()
```



THANK YOU