STOCK PRICE PREDICTION

NIVT

Mentor: Tania Chakraborty

SUMMER TRAINING KOLKATA

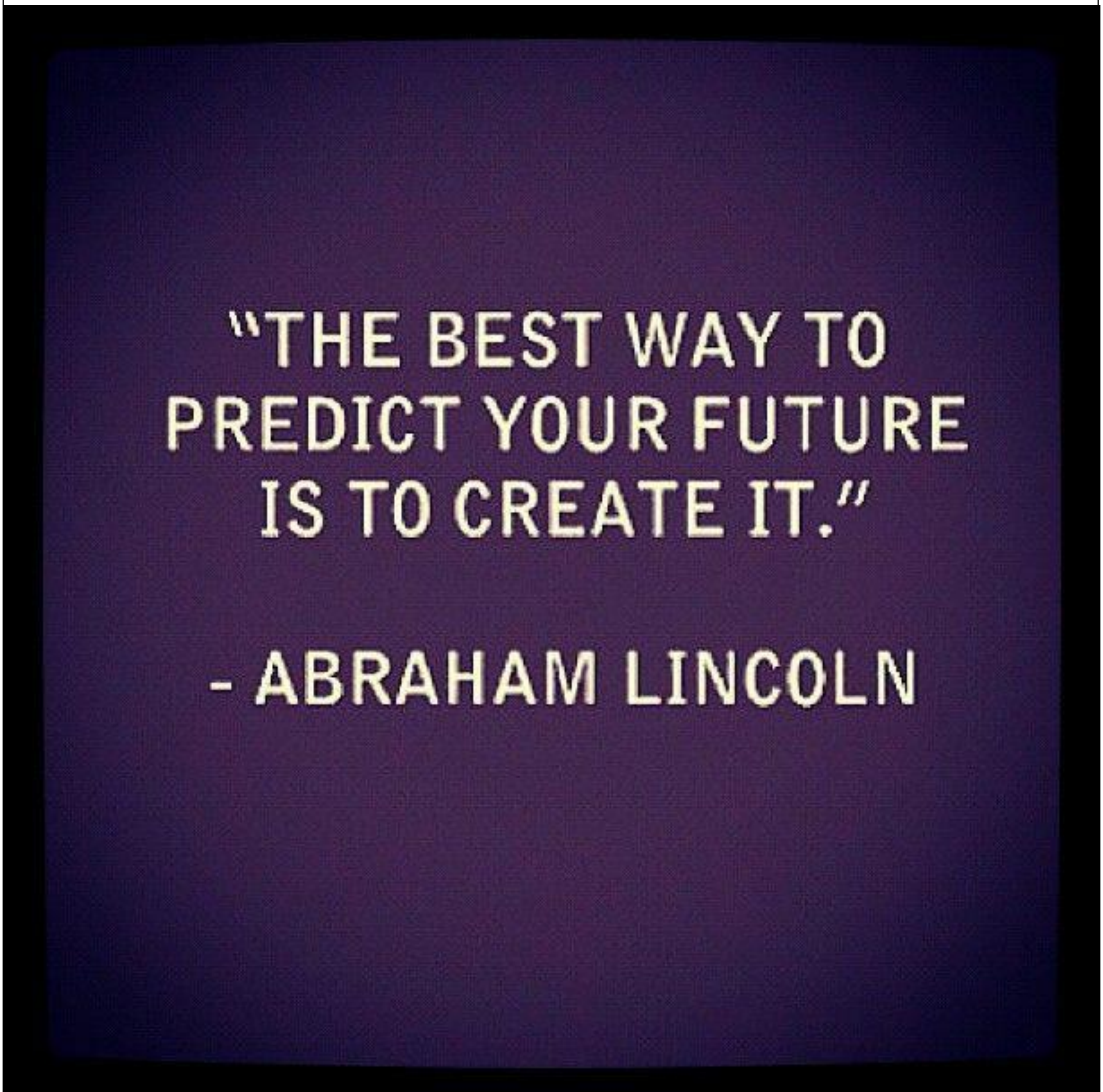
Group

- Prasenjit Karmakar
- Sagar Roy
- Yasar Reza Rahaman

ACKNOWLEDGEMENT

We would like to express my special thanks of gratitude to my teacher Tania Chakraborty as well as our institute NIVT who gave us the golden opportunity to do this wonderful project on the topic STOCK PRICE PREDICTIONS which also helped us to learn so many new things. I am really thankful to them. Secondly, I would also like to thank our friends & group members who helped us a lot in finalizing this project within the limited time frame.

CONTENTS

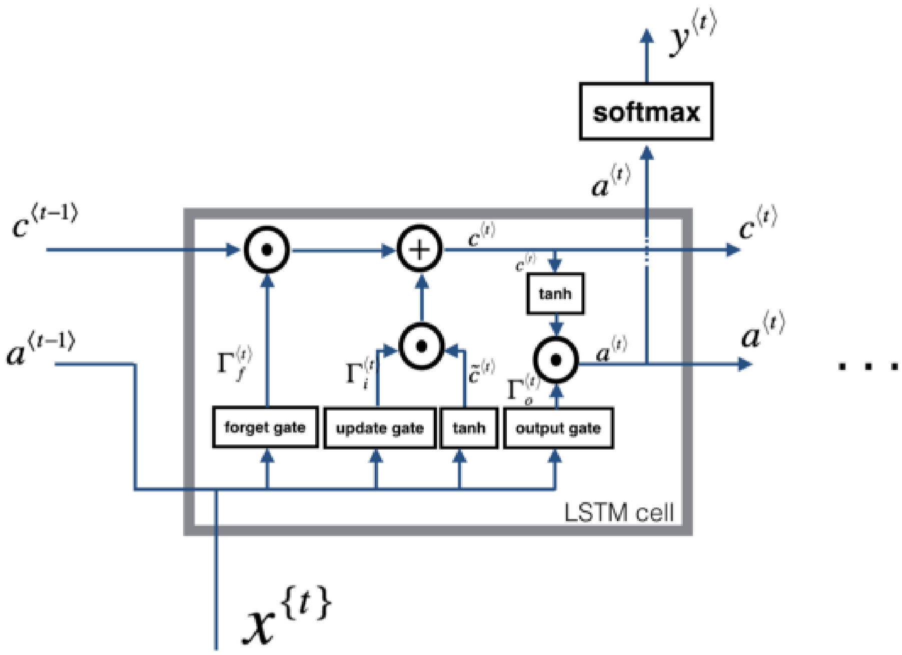

"THE BEST WAY TO
PREDICT YOUR FUTURE
IS TO CREATE IT."
- ABRAHAM LINCOLN

INTRODUCTION

There are a lot of complicated financial indicators and also the fluctuation of the stock market is highly violent. However, as the technology is getting advanced, the opportunity to gain a steady fortune from the stock market is increased and it also helps experts to find out the most informative indicators to make a better prediction. The prediction of the market value is of great importance to help in maximizing the profit of stock option purchase while keeping the risk low. Recurrent neural networks (RNN) have proved one of the most powerful models for processing sequential data. Long Short-Term memory is one of the most successful RNNs architectures. LSTM introduces the memory cell, a unit of computation that replaces traditional artificial neurons in the hidden layer of the network. With these memory cells, networks are able to effectively associate memories and input remote in time, hence suit to grasp the structure of data dynamically over time with high prediction capacity.

HISTORY OF STOCK PREDICTIONS

The art of forecasting stock prices has been a difficult task for many of the researchers and analysts. In fact, investors are highly interested in the research area of stock price prediction. For a good and successful investment, many investors are keen on knowing the future situation of the stock market. Good and effective prediction systems for stock market help traders, investors, and analyst by providing supportive information like the future direction of the stock market. In this work, we present a recurrent neural network (RNN) and Long Short-Term Memory (LSTM) approach to predict stock market indices.



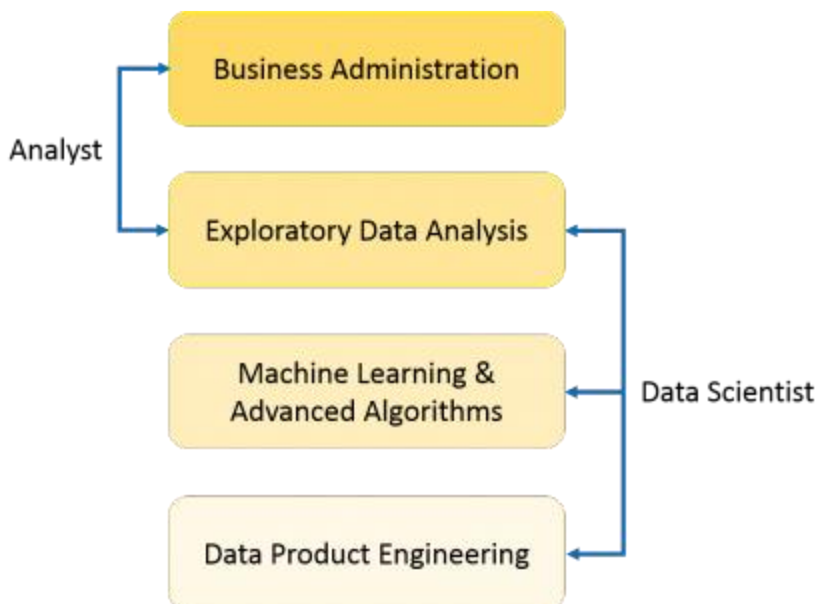
$$\Gamma_f^{\langle t \rangle} = \sigma(W_f[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_f)$$
$$\Gamma_u^{\langle t \rangle} = \sigma(W_u[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_u)$$
$$\tilde{c}^{\langle t \rangle} = \tanh(W_C[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_C)$$
$$c^{\langle t \rangle} = \Gamma_f^{\langle t \rangle} \circ c^{\langle t-1 \rangle} + \Gamma_u^{\langle t \rangle} \circ \tilde{c}^{\langle t \rangle}$$
$$\Gamma_o^{\langle t \rangle} = \sigma(W_o[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_o)$$
$$a^{\langle t \rangle} = \Gamma_o^{\langle t \rangle} \circ \tanh(c^{\langle t \rangle})$$

WHAT IS DATA SCIENCE?

To manipulate data and extract the important part of data is called Data Science.

Data science is a multidisciplinary blend of **data inference, algorithm development, and technology** in order to solve analytically complex problems.

At the core is data. Troves of raw information, streaming in and stored in enterprise data warehouses. Much to learn by mining it. Advanced capabilities we can build with it.



MEANING OF DATA SCIENCE

Data science is the field of study that combines domain expertise, programming skills, and knowledge of math and statistics to extract meaningful insights from data. Data science practitioners apply machine learning algorithms to numbers, text, images, video, audio, and more to produce artificial intelligence (AI) systems that perform tasks which ordinarily require human intelligence. In turn, these systems generate insights that analysts and business users translate into tangible business value.

DEFINE DATA SCIENCE COMPONENTS

Basically, here three component of data science-

- Data Management
- Data Analytics
- Machine Learning

Data Management:

Data Management is a comprehensive collection of practices, concepts, procedures, processes, and a wide range of accompanying systems that allow for an organization to gain control of its **data** resources.

- **Data** Storage and Big **Data**.
- Business Intelligence and Analytics.
- Metadata **Management**.

Data Analytics:

Data analytics is the science of analyzing raw data in order to make conclusions about that information. Many of the techniques and processes of data analytics have been automated into mechanical processes and algorithms that work over raw data for human consumption.

Machine Learning:

Machine Learning (ML) is basically that field of computer science with the help of which computer systems can provide sense to data in much the same way as human beings do. In simple words, ML is a type of artificial intelligence that extract patterns out of raw data by using an algorithm or method. The key focus of ML is to allow computer systems to learn from experience without being explicitly programmed or human intervention.

DATA

Data is a set of values of qualitive or quantitative variables. It is information in raw or unorganized form. It may be fact, figure, character symbols etc.

- Qualitative -Categorical or Nominal:
  - Discrete Data
  - Continuous Data
- Quantitative -Measurable or Countable:
  - Attribute
  - Nominal
  - Ordinal

Information:

Meaningful or organized data is information.

Big Data:

Extremely large data sets that may be analyzed computationally to reveal patterns, trends and associations, especially relating to human behavior and interactions.

Design distributed systems that manage big data using HADOOP and related technologies.

Big data cannot manage by RDBMS

TYPES OF DATA

There are three types of dataset-

1. STRUCTURED DATA
2. SEMI-STRUCTURED DATA
3. UNSTRUCTURED DATA

Structured Data:

Data which can be stored in database SQL in table with rows and columns are called structured data.

Only 5 to 10 % of all informatics data.

	A	B	C	D	E	F
1	Country	Salesperson	Order Date	OrderID	Units	Order Amoun
2	USA	Fuller	1/01/2011	10392	13	1,440.00
3	UK	Gloucester	2/01/2011	10397	17	716.72
4	UK	Bromley	2/01/2011	10771	18	344.00
5	USA	Finchley	3/01/2011	10393	16	2,556.95
6	USA	Finchley	3/01/2011	10394	10	442.00
7	UK	Gillingham	3/01/2011	10395	9	2,122.92
8	USA	Finchley	6/01/2011	10396	7	1,903.80
9	USA	Callahan	8/01/2011	10399	17	1,765.60
10	USA	Fuller	8/01/2011	10404	7	1,591.25
11	USA	Fuller	9/01/2011	10398	11	2,505.60
12	USA	Coghill	9/01/2011	10403	18	855.01
13	USA	Finchley	10/01/2011	10401	7	3,868.60
14	USA	Callahan	10/01/2011	10402	11	2,713.50
15	UK	Rayleigh	13/01/2011	10406	15	1,830.78
16	USA	Callahan	14/01/2011	10408	10	1,622.40
17	USA	Farnham	14/01/2011	10409	19	319.20
18	USA	Farnham	15/01/2011	10410	16	802.00

Semi-Structured Data:

Doesn't reside in a relational database but that does have some organizational properties that make it easier to analyze.

CSV, XML AND JSON documents are semi-structured documents, NoSQL databases are considered as semi-structured.

A few parts of data(5 to 10 %).

CSV

	A	B	C	D
1	ID	Gender	City	Monthly_I
2	ID000002C	Female	Delhi	20000
3	ID000004E	Male	Mumbai	35000
4	ID000007H	Male	Panchkula	22500
5	ID000008I	Male	Saharsa	35000
6	ID000009J	Male	Bengaluru	100000
7	ID000010K	Male	Bengaluru	45000
8	ID000011L	Female	Sindhudui	70000
9	ID000012N	Male	Bengaluru	20000
10	ID000013N	Male	Kochi	75000
11	ID000014C	Female	Mumbai	30000
12	ID000016C	Male	Mumbai	25000
13	ID000018S	Female	Surat	25000
14	ID000019T	Female	Pune	24000
15	ID000021V	Male	Bhubanes	27000
16	ID000022V	Female	Howrah	28000

JSON


```
{
  "Employee": [

    {

      "id":"1",

      "Name": "Ankit",

      "Sal": "1000",

    },
    {

      "id":"2",

      "Name": "Faizy",
```



```
<?xml version="1.0"?>

<contact-info>

<name>Ankit</name>

<company>Anlytics Vidhya</company>

<phone>+9187654321</phone>

</contact-info>
```


Unstructured Data:

Unstructured data represent around 80% of data. It often includes text and multimedia content.

Example: e-mail messages, word processing documents, videos, photos, audio files, presentation, webpages and many other kinds of business documents.

Some example of machine-generated unstructured data :

- ✓ Satellite images
- ✓ Scientific data
- ✓ Photographs and video
- ✓ Rader or Sonar data

Some example of human-generated unstructured data:

- Text internal
- Social media data
- Mobile data
- Website content

```
weblogic.application.utils.StateMachineDriver.nextState(StateMachineDriver.java:26)
>
####<Dec 29, 2006 2:14:24 PM IST> <Notice> <Log Management> <svaidyan02> <xbusServer>
<[ACTIVE] ExecuteThread: '0' for queue: 'weblogic.kernel.Default (self-tuning)'> <<WLS
Kernel>> <> <> <1167381864275> <BEA-170027> <The server initialized the domain log
broadcaster successfully. Log messages will now be broadcasted to the domain log.>
####<Dec 29, 2006 2:14:24 PM IST> <Notice> <WebLogicServer> <svaidyan02> <xbusServer> <Main
Thread> <<WLS Kernel>> <> <> <1167381864976> <BEA-000365> <Server state changed to ADMIN>
####<Dec 29, 2006 2:14:24 PM IST> <Notice> <WebLogicServer> <svaidyan02> <xbusServer> <Main
Thread> <<WLS Kernel>> <> <> <1167381864996> <BEA-000365> <Server state changed to RESUMING>
####<Dec 29, 2006 2:14:28 PM IST> <Notice> <Security> <svaidyan02> <xbusServer> <[STANDBY]
ExecuteThread: '5' for queue: 'weblogic.kernel.Default (self-tuning)'> <<WLS Kernel>> <> <>
<1167381868541> <BEA-090171> <Loading the identity certificate and private key stored under
the alias DemoIdentity from the jks keystore file
C:\bea2613a\WEBLOG~1\server\lib\DemoIdentity.jks.>
####<Dec 29, 2006 2:14:29 PM IST> <Notice> <Security> <svaidyan02> <xbusServer> <[STANDBY]
ExecuteThread: '5' for queue: 'weblogic.kernel.Default (self-tuning)'> <<WLS Kernel>> <> <>
<1167381869643> <BEA-090169> <Loading trusted certificates from the jks keystore file
C:\bea2613a\WEBLOG~1\server\lib\DemoTrust.jks.>
####<Dec 29, 2006 2:14:29 PM IST> <Notice> <Security> <svaidyan02> <xbusServer> <[STANDBY]
ExecuteThread: '5' for queue: 'weblogic.kernel.Default (self-tuning)'> <<WLS Kernel>> <> <>
<1167381869713> <BEA-090169> <Loading trusted certificates from the jks keystore file
C:\bea2613a\JROCKI~1\jre\lib\security\cacerts.>
####<Dec 29, 2006 2:15:32 PM IST> <Warning> <Server> <svaidyan02> <xbusServer>
<DynamicSSLListenThread[DefaultSecure[1]]> <<WLS Kernel>> <> <> <1167381932743> <BEA-002611>
<Hostname "svaidyan02.apac.bea.com", maps to multiple IP addresses: 192.168.1.5,
172.22.56.120>
####<Dec 29, 2006 2:15:32 PM IST> <Notice> <Server> <svaidyan02> <xbusServer> <[STANDBY]
ExecuteThread: '5' for queue: 'weblogic.kernel.Default (self-tuning)'> <<WLS Kernel>> <> <>
<1167381932753> <BEA-002613> <Channel "Default[2]" is now listening on 127.0.0.1:7021 for
```

DATA ANALYTICS

<u>Data Analytics is the process of examining data sets in order to draw conclusion about the information it contains.</u>

increasingly with the aid of specialized systems and software. Data analytics technologies and techniques are widely used in commercial industries to enable organizations to make more-informed business decisions and by scientists and researchers to verify or disprove scientific models, theories and hypotheses.

<u>Analytics is not a tool of technology, rather it is the way of thinking and acting on data.</u>

TYPES OF DATA ANALYTICS

Data analytics are of three types those are: -

a) Descriptive Analytics
b) Predictive Analytics
c) Prescriptive Analytics

Descriptive Analytics:

90% of organizations today use descriptive analytics which is the most basic form of analytics. The simplest way to define descriptive analytics is that, it answers the question "What has happened?". This type of analytics, analyses the data coming in real-time and historical data for insights on how to approach the future. The main objective of descriptive analytics is to find out the reasons behind precious success or failure in the past. The 'Past' here, refers to any particular time in which an event had occurred and this could be a month ago or even just a minute ago. The vast majority of big data analytics used by organizations falls into the category of descriptive analytics.

Predictive Analytics:

The subsequent step in data reduction is predictive analytics. Analyzing past data patterns and trends can accurately inform a business about what could happen in the future. This helps in setting realistic goals for the business, effective planning and restraining expectations. Predictive analytics is used by businesses to study the data and ogle into the crystal ball to find answers to the question "What could happen in the future based on previous trends and patterns?"

Prescriptive Analytics:

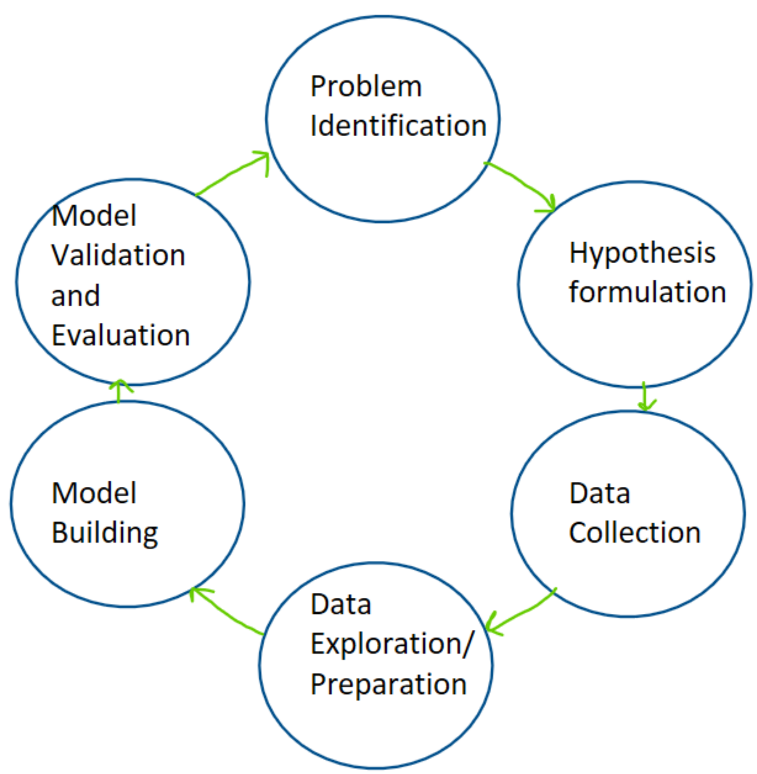
Big data might not be a reliable crystal ball for predicting the exact winning lottery numbers but it definitely can highlight the problems and help a business understand why those problems occurred. Businesses can use the data-backed and data-found factors to create prescriptions for the business problems, that lead to realizations and observations.

PROCESS OF ANALYZING THE DATA IN DATA SCIENCE

Data goes through various process during data analytics in data Science. Those are given below

DATA SCIENCE PROCESS OR LIFE CYCLE

1. Problem Identification
2. Hypothesis formulation
3. Data Collection
4. Data Exploration/Preparation
5. Model Building
6. Model Validation and Evaluation



DATA MANAGEMENT TOOLS

NumPy

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

scikit learn

TOP 10 PYTHON LIBRARIES

One of the reason PYTHON Is mostly popular among developers is its wide range of libraries. We can't even count how many libraries it has but We will be considering the following 10 libraries:

- NUMPY
- PANDAS
- MATPLOTLIB
- SEABORN
- IPYTHON
- TENSORFLOW
- SCIKIT-LEARN
- KERAS
- PYTORCH
- SCIPY

JUPETER NOTEBOOK

Jupiter Notebook is an interface we use to write python programs. It is Widely popular because most of the popular python libraries are pre-installed in it.

And we can write and run part of a code written in a code cell. So, it is very helpful for debugging.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits import mplot3d
import seaborn as sns
plt.style.use('seaborn-darkgrid')
%matplotlib inline
```

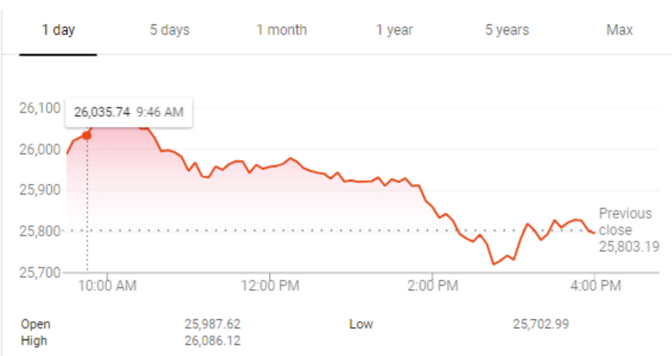
```
df=pd.read_csv("HR_comma_sep.csv")
```

```
df.head(10)
```

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years	sales	salary
0	0.38	0.53	2	157	3	0	1	0	sales	low
1	0.80	0.86	5	262	6	0	1	[illegible]	sales	medium
2	0.11	0.88	7	272	4	[illegible]	[illegible]	0	sales	medium
3	0.72	0.87	5	223	5	[illegible]	[illegible]	0	sales	low
4	0.37	0.52	2	159	3	0	1	0	sales	low
5	0.41	0.50	2	153	3	0	1	0	sales	low
6	0.10	0.77	6	247	4	0	1	0	sales	low
7	0.92	0.85	5	259	5	0	1	0	sales	low
8	0.89	1.00	5	224	5	0	1	0	sales	low
9	0.42	0.53	2	142	3	0	1	0	sales	low

DATA SCIENCE IN STOCK ANALYSIS

Trying to predict the stock market is an enticing prospect to data scientists motivated not so much as a desire for material gain, but for the challenge. We see the daily up and downs of the market and imagine there must be patterns we, or our models, can learn in order to beat all those day traders with business degrees. Naturally, when I started using additive models for time series prediction, I had to test the method in the proving ground of the stock market with simulated funds. Inevitably, I joined the many others who have tried to beat the market on a day-to-day basis and failed. However, in the process, I learned a ton of Python including object-oriented programming, data manipulation, modeling, and visualization. I also found out why we should avoid playing the daily stock market without losing a single dollar (all I can say is play the long game)!



One day vs 30 years: which would you rather put your money on?

When we don't experience immediate success — in any task, not just data science — we have three options:

1. Tweak the results to make it look like we were successful
2. Hide the results so no one ever notices
3. Show all our results and methods so that others (and ourselves) can learn how to do things better

USE OF DATA SCIENCE IN STOCK MARKET

Machine Learning and Data Analytics are making trading much more efficient. Together, they complement each other and act as catalysts towards improved ability to identify opportunities and reduce trading costs.

During the research phase, the ability to analyze and learn effectively from the vast amount of data gives the strategies an edge.

Technology is scaling at an exponential rate and today we are processing vast loads of data if numbers are to be believed. A recent report reveals that the total data existing in the world will grow at a CAGR of 61% to 175 zettabytes by 2025 from 33 zettabytes in 2018. These numbers are huge!

Today, Machine Learning and Data Analytics are making trading much more efficient. Together, they complement each other and act as catalysts towards improved ability to identify opportunities and reduce trading costs. Leveraging data in trading at present comes mostly in two flavours. During the research phase, the ability to analyse and learn effectively from the vast amount of data gives the strategies an edge. Here the analysis is mostly exploratory and speed is important but not critical. During the execution phase, these same tools can be used to gain the ability to quickly react to changing market conditions. Here speed is critical to success.

DTYPES OF GIVEN RAW DATA

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn  as sns
from IPython.display import clear_output
plt.style.use("ggplot")
%matplotlib inline
```

```
data=pd.read_csv("dataset.csv")

data['Date']=pd.to_datetime(data.Date)

#creating necessory data
data['DAY']=data.Date.apply(lambda x:x.day_name())
data['MONTH']=data.Date.apply(lambda x:x.month_name())
data['YEAR']=data.Date.apply(lambda x:x.year)
```

```
data.head()
```

	Date	Close	DAY	MONTH	YEAR
0	2010-01-04	133.899994	Monday	January	2010
1	2010-01-05	134.690002	Tuesday	January	2010
2	2010-01-06	132.250000	Wednesday	January	2010
3	2010-01-07	130.000000	Thursday	January	2010
4	2010-01-08	133.520004	Friday	January	2010

This dataset originally belongs to AMAZON.

It has 5 columns and 1762 rows for their employees.

```
data.dtypes
```

```
Date      datetime64[ns]
Close            float64
DAY               object
MONTH             object
YEAR               int64
dtype: object
```

The columns represent date, closing stock price, day of weak month, year of a given working day in amazon's stock market data.

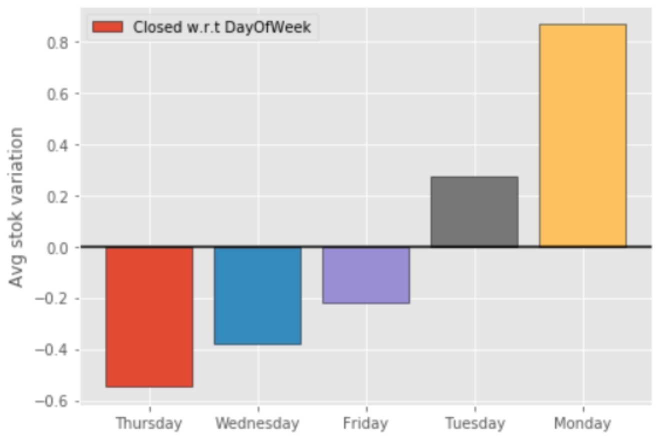
DATA DESCRIPTION

Q1. IN which day of the week stock prices are max and min ?

Min: Thursday
Max: Monday

```
d=data.groupby('DAY')['Close'].mean().sort_values()
```

```
fig=plt.figure(figsize=(7,5))
ax1=fig.add_subplot(111)
ax1.bar(d.index,d-d.mean(),color=sns.color_palette(),ec='k',label='Closed w.r.t DayOfWeek')
ax1.axhline(y=0,c='k')
ax1.legend()
ax1.set_ylabel("Avg stok variation")
plt.show()
```



In this graph we can see that in Monday the sock prices are maximum and in Thursday it is minimum. So, it will be optimal for a person to buy stocks in Thursday and sell them in Monday.

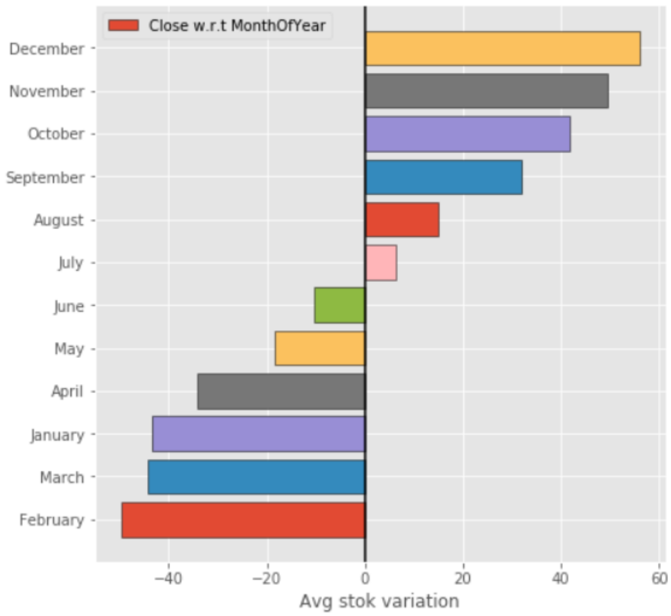
The bar chart shows the corresponding chance of increasing or decreasing chance of every working day in Amazon.

Q2. IN which month of the year stock prices are max and min ?

Min:February
Max:December

```
d=data.groupby('MONTH')['Close'].mean().sort_values()
```

```
fig=plt.figure(figsize=(7,7))
ax1=fig.add_subplot(111)
ax1.barh(d.index,d-d.mean(),color=sns.color_palette(),ec='k',label="Close w.r.t MonthOfYear")
ax1.axvline(x=0,c='k')
ax1.legend()
ax1.set_xlabel("Avg stok variation")
plt.show()
```



In this graph we can see that in December the sock prices are maximum and in February it is minimum. So, it will be optimal for a person to buy stocks in February and sell them in December.

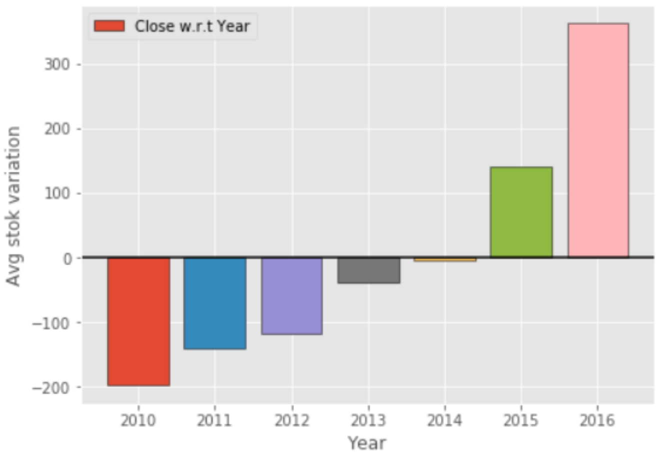
The bar chart shows the corresponding chance of increasing or decreasing chance of every month in Amazon.

Q3. IN which year stock prices are max and min ?

Min:2010
Max:2016

```
d=data.groupby('YEAR')['Close'].mean().sort_values()
```

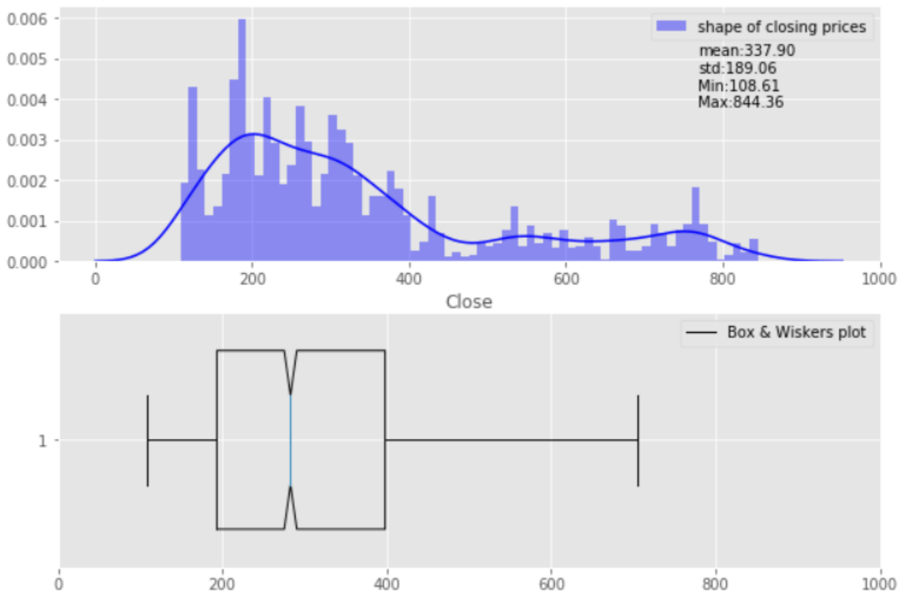
```
fig=plt.figure(figsize=(7,5))
ax1=fig.add_subplot(111)
ax1.bar(d.index,d-d.mean(),color=sns.color_palette(),ec='k',label="Close w.r.t Year")
ax1.axhline(y=0,c='k')
ax1.legend()
ax1.set_ylabel("Avg stok variation")
ax1.set_xlabel("Year")
plt.show()
```



this graph shows the gradual increase of stock prices over the year range 2010 to 2016 for Amazon. So, we can conclude that the company is improving its economic conditions.

Q4. Show the shape and Quantiles of the Series Close.

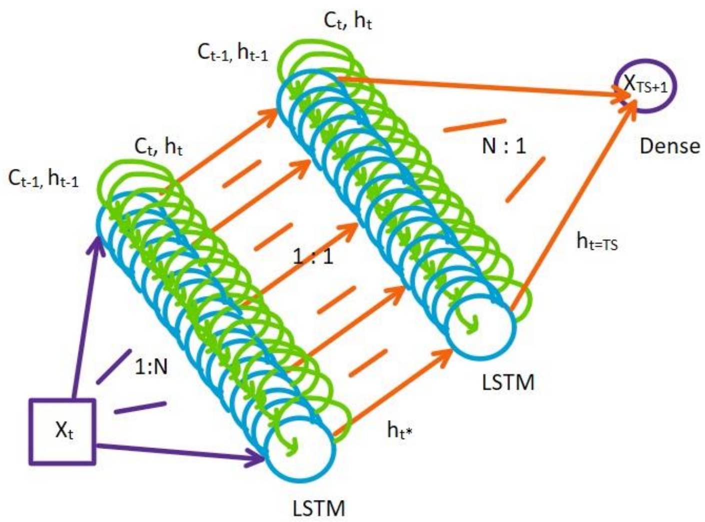
```
fig=plt.figure(figsize=(10,7))
ax1=fig.add_subplot(211)
sns.distplot(data.Close,bins=70,label='shape of closing prices',ax=ax1,color='b',norm_hist=True)
ax1.text(769,0.0038,"mean:{:.2f}\nstd:{:.2f}\nMin:{:.2f}\nMax:{:.2f}"
         .format(np.mean(data.Close),np.std(data.Close),data.Close.min(),data.Close.max()))
ax1.legend()
ax2=fig.add_subplot(212)
ax2.boxplot(data.Close,notch=True,showfliers=False,vert=False,widths=0.7)
ax2.legend(["Box & Wiskers plot"])
ax2.set_xlim(0,1000)
plt.show()
```



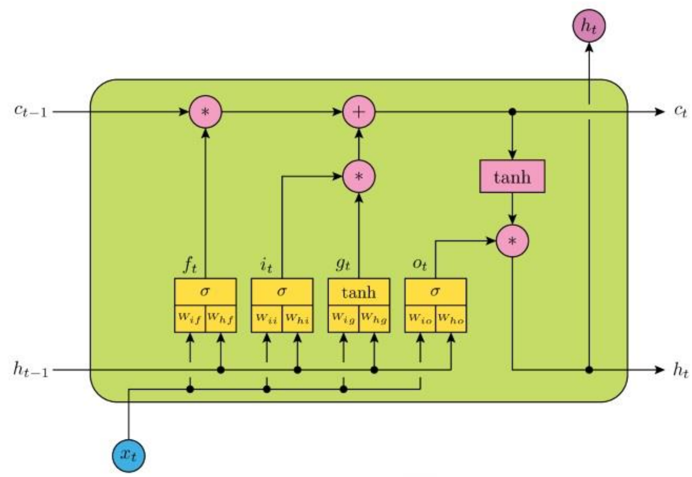
This graph represents how the stock data is distributed and give a general understanding of the expectation and variance of the stocks. We can see that its mean is 337.90 and standard deviation is 189.06.

OUR APPROACH TO THE PROBLEM

LSTM cell and RNN:



Recurrent Neural Network(N=50)



LSTM Cell

Goal: Our goal is to predict the next stock price from previous 10 days stock prices.

TRAINING AND TESTING DATA SET

whole data:

```
import tensorflow as tf
from sklearn.preprocessing import MinMaxScaler
```

```
data=data[['Close']]
data.head()
```

	Close
0	133.899994
1	134.690002
2	132.250000
3	130.000000
4	133.520004

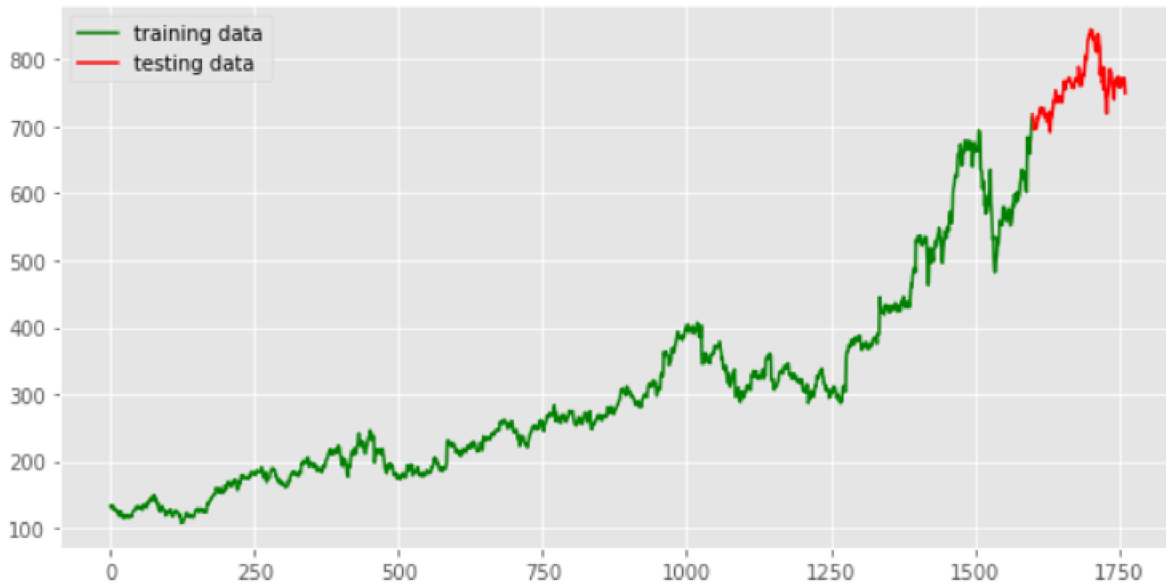
```
data.plot(figsize=(10,5),color='brown')
plt.show()
```

SPLITING THE DATA:

Partitioning the Data

```
train_data=data.iloc[0:1600,:] #taking 1600 points for training
test_data=data.iloc[1600:,:] #taking remaining 162 points for testing
```

```
fig=plt.figure(figsize=(10,5))
ax=fig.add_subplot(111)
ax.plot(train_data.index,train_data.Close,label='training data',color='green')
ax.plot(test_data.index,test_data.Close,label='testing data',color='red')
ax.legend()
plt.show()
```



Scaling the Data

```
s=MinMaxScaler()
X_train=s.fit_transform(train_data).flatten()
X_test=s.transform(test_data).flatten()
```

Setting Hyperparameters

```
time_stamp=10
batch_size=32
```

Formating the dataset for RNN

```
#training dataset
l=[]
for seq in range(X_train.shape[0]-(time_stamp+1)): #10 = time step
    l.append(X_train[seq:(time_stamp+1)+seq])

ALLSeries=np.array(l) # all of size 10 + 1 elements
X_series,y_val=ALLSeries[:,0:-1].reshape(-1,time_stamp,1),ALLSeries[:,-1]
```

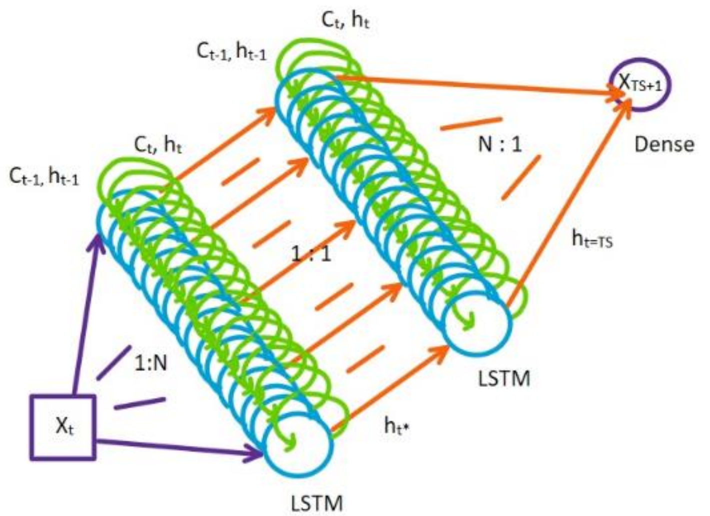
```
#testing dataset
l=[]
for seq in range(X_test.shape[0]-(time_stamp+1)): #10 = time step
    l.append(X_test[seq:(time_stamp+1)+seq])

ALLSeries=np.array(l) # all of size 10 + 1 elements
X_test_series,y_test_val=ALLSeries[:,0:-1].reshape(-1,time_stamp,1),ALLSeries[:,-1]
```

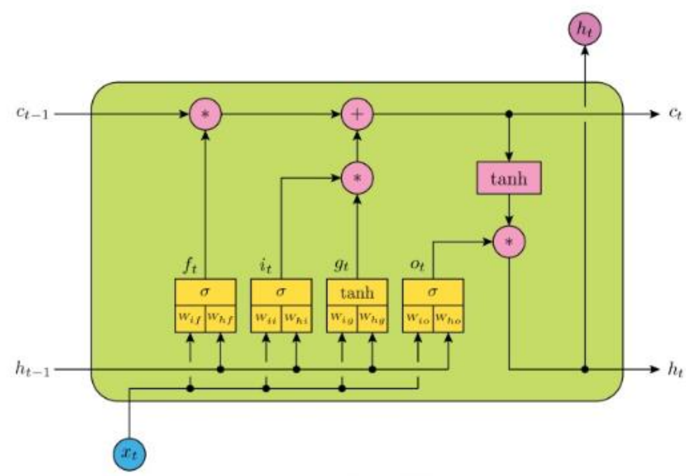
OUR RNN MODEL

RNN Model

```
model=tf.keras.Sequential()
model.add(tf.keras.layers.LSTM(units=50,return_sequences=True,input_shape=(time_stamp,1)))
model.add(tf.keras.layers.LSTM(units=50,return_sequences=False))
model.add(tf.keras.layers.Dense(units=1))
model.compile(loss='mse',optimizer='adam')
clear_output()
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
lstm (LSTM)                  (None, 10, 50)            10400     
_________________________________________________________________
lstm_1 (LSTM)                (None, 50)                20200     
_________________________________________________________________
dense (Dense)                (None, 1)                 51        
=================================================================
Total params: 30,651
Trainable params: 30,651
Non-trainable params: 0
_________________________________________________________________
```





TRAINING OR LOADING OUR MODEL

Training the Model

```
history=model.fit(x=X_series,y=y_val,batch_size=batch_size,epochs=100)
clear_output()

#fig=plt.figure(figsize=(10,5))
#ax=fig.add_subplot(111)
#ax.plot(history.history['loss'],label='MSE cost function')
#ax.legend()
#plt.show()
```

Save & Load the Model

```
#model.save("./keras_Model/my_rnn_model.h5")
#del model
model=tf.keras.models.load_model("./keras_Model/my_rnn_model.h5")
clear_output()
print("Model Loaded")
```

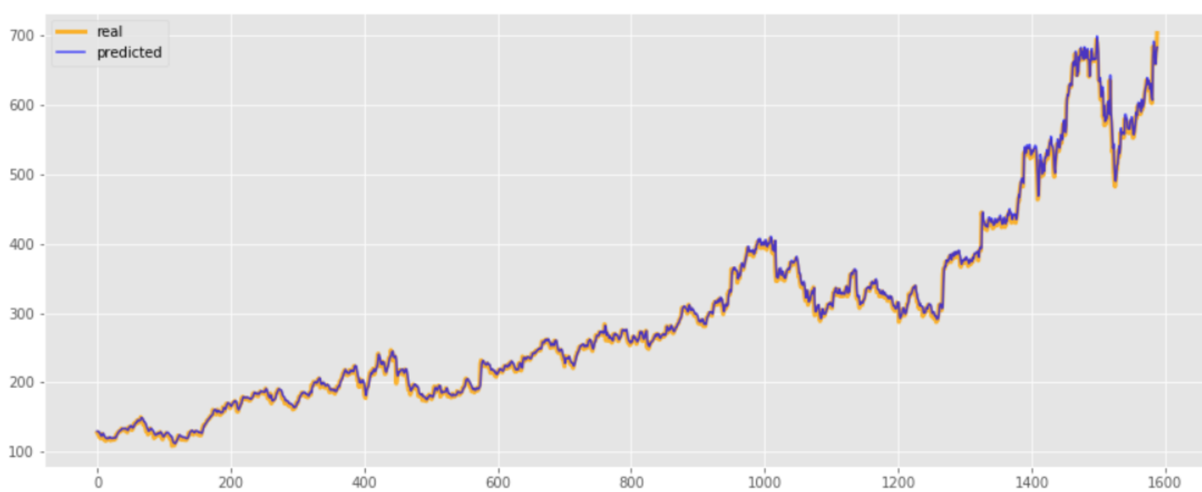
```
Model Loaded
```

TESTING OUR MODEL

ON TRAINING DATA:

RNN prediction on Training data

```
pred=model.predict(X_series)

fig=plt.figure(figsize=(15,6))
ax=fig.add_subplot(111)
ax.plot(s.inverse_transform(y_val.reshape(-1,1)),label='real',color='orange',alpha=0.8,linewidth=3)
ax.plot(s.inverse_transform(pred),label='predicted',color='b',alpha=0.7)
ax.legend()
plt.show()
```



ON TESTING DATA:

RNN prediction on Testing data

```
test_pred=model.predict(X_test_series)

fig=plt.figure(figsize=(15,6))
ax=fig.add_subplot(111)
ax.plot(s.inverse_transform(test_pred),label='predicted',linewidth=3,color='g')
ax.plot(s.inverse_transform(y_test_val.reshape(-1,1)),label='real',color='r',linestyle='dashed')
ax.legend()
plt.show()
```

WHAT IS BETTER APPROACH?

RNNs in general and LSTMs in particular have received the most success when working with sequences of words and paragraphs, generally called natural language processing.

This includes both sequences of text and sequences of spoken language represented as a time series. They are also used as generative models that require a sequence output, not only with text, but on applications such as generating handwriting.

Use RNNs for:
Text data
Speech data
Classification prediction problems
Regression prediction problems
Generative models
Recurrent neural networks are not appropriate for tabular datasets as you would see in a CSV file or spreadsheet. They are also not appropriate for image data input.

Don't Use RNNs for:
Tabular data
Image data
RNNs and LSTMs have been tested on time series forecasting problems, but the results have been poor, to say the least. Autoregression methods, even linear methods often perform much better. LSTMs are often outperformed by simple MLPs applied on the same data.

DECISION MAKING

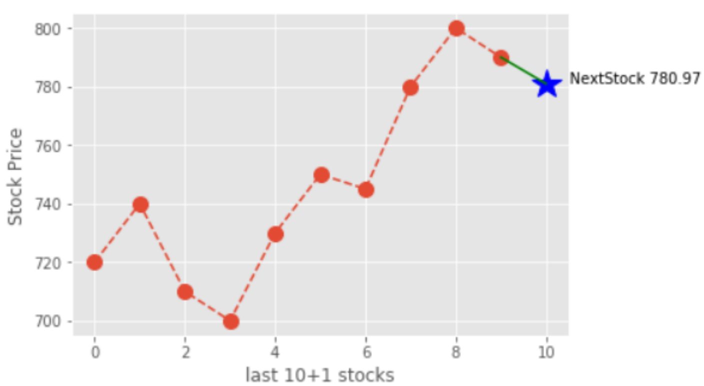
Now from the RNN we can predict next day stock prices from the previous 10 days stock price value. This have a great accuracy we see on the testing data. So, we are confident that it will work on current real data also. So, by using this advanced statistical approach we can successfully predict stocks and increase our economic assets.

```
def PredictNextStockPrice():
    def nextStock(inp):
        return s.inverse_transform(model.predict(s.transform(np.array(inp).reshape(-1,1)).reshape(1,10,1)))[0,0]

    inp=[float(e) for e in input("Enter last 10 days\nstock prices:").split(" ")]
    assert len(inp)==10,"Give Previous 10 days values"
    inp.append(nextStock(inp))

    plt.plot(inp[:-1],marker='o',linestyle='dashed',markersize=10)
    plt.plot(10,inp[-1],markersize=20,marker='*',color='blue')
    plt.xlabel("last 10+1 stocks")
    plt.ylabel("Stock Price")
    plt.annotate("     NextStock {:.2f}".format(inp[-1]),(10,inp[-1]))
    plt.plot([9,10],[inp[-2],inp[-1]],color='g')
    plt.show()
    print("Next stock Price will be: {:.2f}".format(inp[-1]))
```

example stock prices: 720 740 710 700 730 750 745 780 800 790

```
PredictNextStockPrice() #give values seperated by space
```

```
Enter last 10 days
stock prices:720 740 710 700 730 750 745 780 800 790
```




```
Next stock Price will be: 780.97
```

BENEFITS

- Using This we can successfully predict stocks of a given company.
- We can invest wisely and make money.
- The company can also take economic decisions based on the prediction.
- With this we can successfully simulate a company's economic behavior.
- We can create jobs under this.
- Help to make educated gushes of the future.

CONCLUSION

The popularity of stock market trading is growing rapidly, which is encouraging researchers to find out new methods for the prediction using new techniques. The forecasting technique is not only helping the researchers but it also helps investors and any person dealing with the stock market. In order to help predict the stock indices, a forecasting model with good accuracy is required. In this work, we have used one of the most precise forecasting technologies using Recurrent Neural Network and Long Short-Term Memory unit which helps investors, analysts or any person interested in investing in the stock market by providing them a good knowledge of the future situation of the stock market.

END