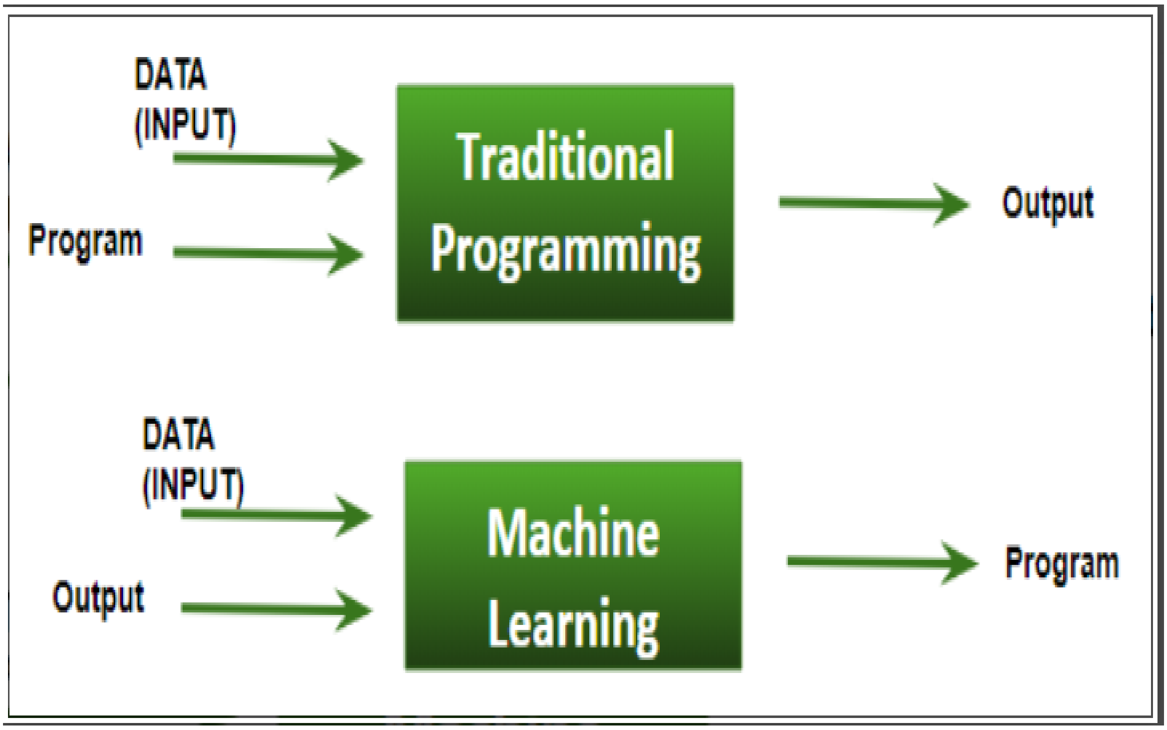
WINTER INTERNSHIP REPORT

Area of Online Internship	AI/ML/DL
Intern Name	GUDDU KUMAR
Name of Institution	INDIAN INSTITUTE OF TECHNOLOGY, INDORE
Faculty Mentor Name	Prof. Vimal Bhatia
Duration	2 MONTHS (01/01/2022 TO 28/02/2022)
Date of Submission	14/03/2022

Table Of Contents

Machine Learning:- It is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that makes it more similar to humans: ***The ability to learn.*** Machine Learning(ML) can be explained as automating and improving the learning process of computers based on their experiences without being actually programmed i.e. without any human assistance. The process starts with feeding good quality data and then training our machines(computers) by building machine learning models using the data and different algorithms. The choice of algorithms depends on what type of data do we have and what kind of task we are trying to automate.



Basic Difference in ML and Traditional Programming

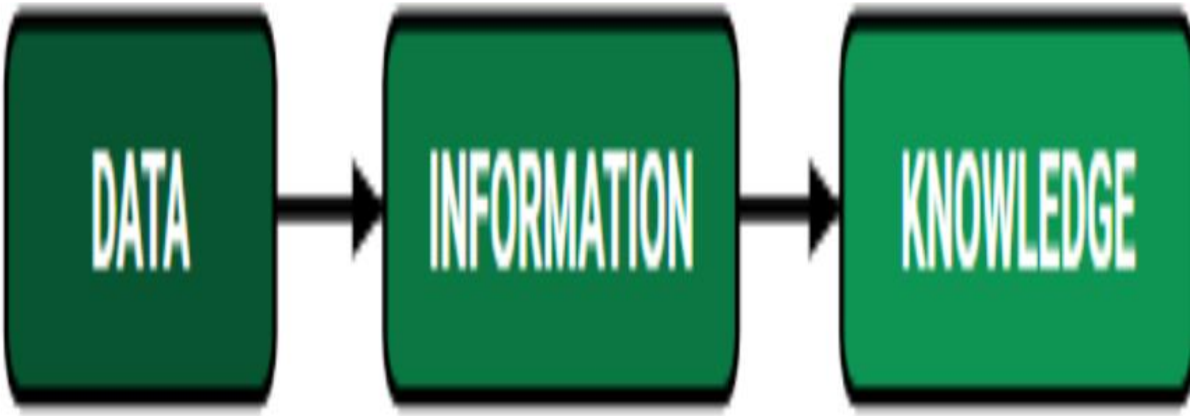
- **Traditional Programming :** We feed in DATA (Input) + PROGRAM (logic), run it on machine and get output.

- **Machine Learning :** We feed in DATA(Input) + Output, run it on machine during training and the machine creates its own program(logic), which can be evaluated while testing.

DATA: It can be any unprocessed fact, value, text, sound, or picture that is not being interpreted and analyzed. Data is the most important part of all Data Analytics, Machine Learning, Artificial Intelligence. Without data, we can't train any model and all modern research and automation will go in vain. Big Enterprises are spending lots of money just to gather as much certain data as possible.

INFORMATION: Data that has been interpreted and manipulated and has now some meaningful inference for the users.

KNOWLEDGE: Combination of inferred information, experiences, learning, and insights. Results in awareness or concept building for an individual or organization.

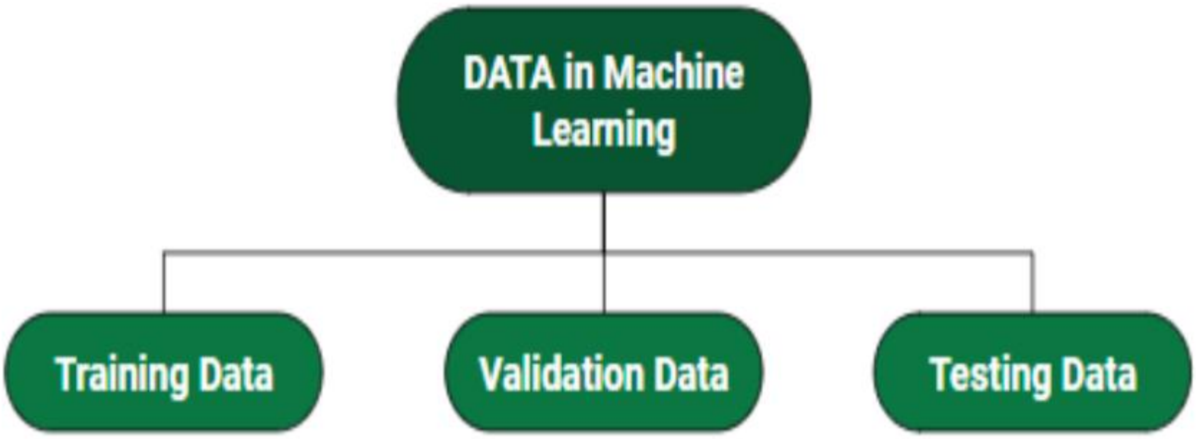


How we split data in Machine Learning:-

- **Training Data:** The part of data we use to train our model. This is the data that your model actually sees(both input and output) and learns from.

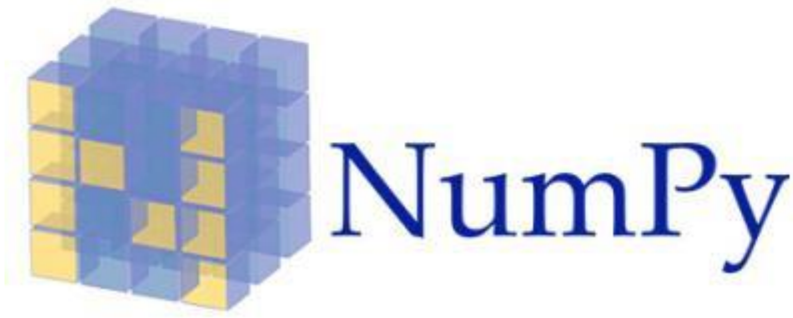
- **Validation Data:** The part of data that is used to do a frequent evaluation of the model, fit on the training dataset along with improving involved hyperparameters (initially set parameters before the model begins learning). This data plays its part when the model is actually training.

Testing Data: Once our model is completely trained, testing data provides an unbiased evaluation. When we feed in the inputs of Testing data, our model will predict some values(without seeing actual output). After prediction, we evaluate our model by comparing it with the actual output present in the testing data. This is how we evaluate and see how much our model has learned from the experiences feed in as training data, set at the time of training.



Best Python libraries for Machine Learning:-

- Numpy
- Scikit-learn
- TensorFlow
- Pandas
- Matplotlib



NumPy is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions. It is very useful for fundamental scientific computations in Machine

Learning. It is particularly useful for linear algebra, Fourier transform, and random number capabilities. High-end libraries like TensorFlow uses NumPy internally for manipulation of Tensors.

```python
# Python program using NumPy
# for some basic mathematical
# operations

import numpy as np

# Creating two arrays of rank 2
x = np.array([[1, 2], [3, 4]])
y = np.array([[5, 6], [7, 8]])

# Creating two arrays of rank 1
v = np.array([9, 10])
w = np.array([11, 12])

# Inner product of vectors
print(np.dot(v, w), "\n")

# Matrix and Vector product
print(np.dot(x, v), "\n")

# Matrix and matrix product
print(np.dot(x, y))
```

Output:

```
219


[29 67]


[[19 22]
 [43 50]]
```

scikit
learn

Scikit-learn is one of the most popular ML libraries for classical ML algorithms. It is built on top of two basic Python libraries, viz., NumPy and SciPy. Scikit-learn supports most of the supervised and unsupervised learning algorithms. Scikit-learn can also be used for data-mining and data-analysis, which makes it a great tool who is starting out with ML.

```
# Python script using Scikit-learn
# for Decision Tree Classifier

# Sample Decision Tree Classifier
from sklearn import datasets
from sklearn import metrics
from sklearn.tree import DecisionTreeClassifier

# load the iris datasets
dataset = datasets.load_iris()

# fit a CART model to the data
model = DecisionTreeClassifier()
model.fit(dataset.data, dataset.target)
print(model)

# make predictions
expected = dataset.target
predicted = model.predict(dataset.data)

# summarize the fit of the model
print(metrics.classification_report(expected, predicted))
print(metrics.confusion_matrix(expected, predicted))
```

Output:

```
DecisionTreeClassifier(class_weight=None, criterion='gini',
max_depth=None,

            max_features=None, max_leaf_nodes=None,

            min_impurity_decrease=0.0, min_impurity_split=None,

            min_samples_leaf=1, min_samples_split=2,

            min_weight_fraction_leaf=0.0, presort=False,
random_state=None,

            splitter='best')

              precision    recall  f1-score   support
```

```
           0       1.00      1.00      1.00        50
           1       1.00      1.00      1.00        50
           2       1.00      1.00      1.00        50

   micro avg       1.00      1.00      1.00       150
   macro avg       1.00      1.00      1.00       150
weighted avg       1.00      1.00      1.00       150

[[50  0  0]
 [ 0 50  0]
 [ 0  0 50]]
```

TensorFlow is a very popular open-source library for high performance numerical computation developed by the Google Brain team in Google. As the name suggests, Tensorflow is a framework that involves defining and running computations involving tensors. It can train and run deep neural networks that can be used to develop several AI applications. TensorFlow is widely used in the field of deep learning research and application.

```
#  Python program using TensorFlow
#  for multiplying two arrays

# import `tensorflow`
import tensorflow as tf
```

```
# Initialize two constants
x1 = tf.constant([1, 2, 3, 4])
x2 = tf.constant([5, 6, 7, 8])

# Multiply
result = tf.multiply(x1, x2)

# Initialize the Session
sess = tf.Session()

# Print the result
print(sess.run(result))

# Close the session
sess.close()
```

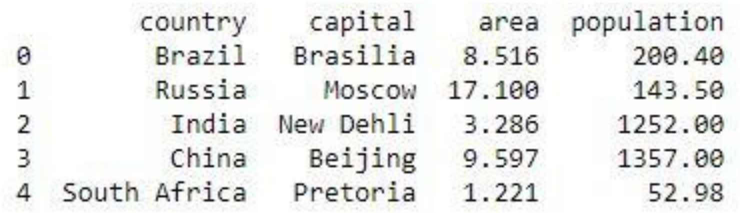
Output:

```
[ 5 12 21 32]
```

Pandas is a popular Python library for data analysis. It is not directly related to Machine Learning. As we know that the dataset must be prepared before training. In this case, Pandas comes handy as it was developed specifically for data extraction and preparation. It provides high-level data structures and wide variety tools for data analysis. It provides many inbuilt methods for groping, combining and filtering data.

```
# Python program using Pandas for
# arranging a given set of data
# into a  table
# importing pandas as pd
import pandas as pd

data = {"country": ["Brazil", "Russia", "India", "China", "South Africa"],
       "capital": ["Brasilia", "Moscow", "New Delhi", "Beijing",
"Pretoria"],
       "area": [8.516, 17.10, 3.286, 9.597, 1.221],
       "population": [200.4, 143.5, 1252, 1357, 52.98] }

data_table = pd.DataFrame(data)
print(data_table)
```

Output:-



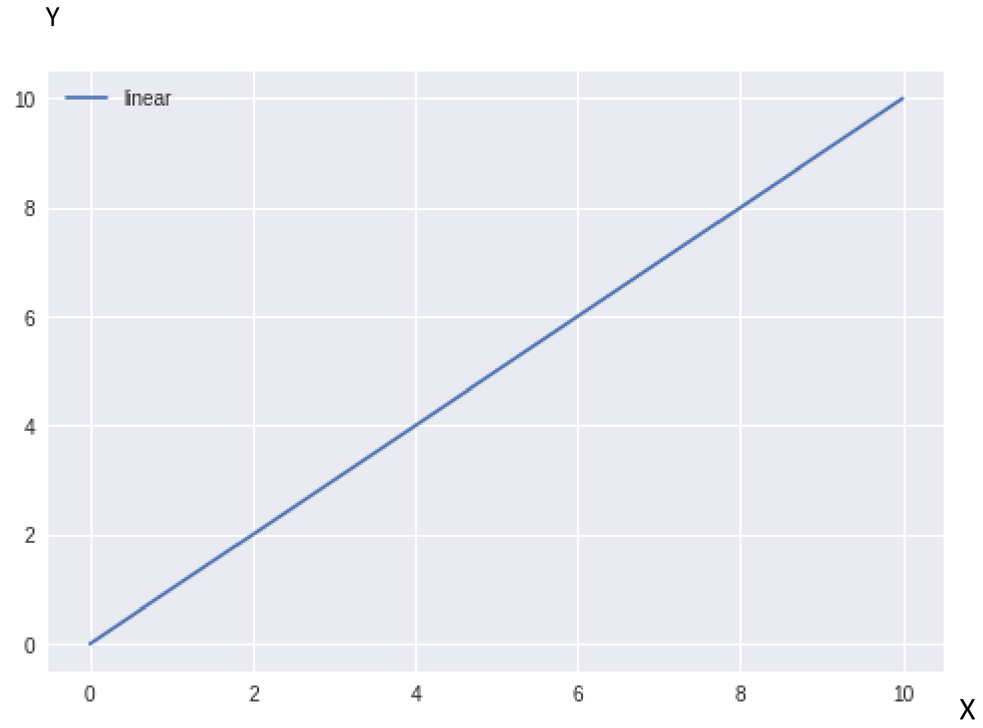
```
        country   capital    area  population
0        Brazil  Brasilia   8.516      200.40
1        Russia    Moscow  17.100      143.50
2         India New Dehli   3.286     1252.00
3         China   Beijing   9.597     1357.00
4  South Africa  Pretoria   1.221       52.98
```

Matplotlib is a very popular Python library for data visualization. Like Pandas, it is not directly related to Machine Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots. A module named pyplot makes it easy for programmers for plotting as it provides features to control line styles, font properties, formatting axes, etc. It provides various kinds of graphs and plots for data visualization, viz., histogram, error charts, bar chats, etc,

```
#  Python program using Matplotlib
# for forming a linear plot

# importing the necessary packages and modules
```

```
import matplotlib.pyplot as plt
import numpy as np

# Prepare the data
x = np.linspace(0, 10, 100)

# Plot the data
plt.plot(x, x, label ='linear')

# Add a legend
plt.legend()

# Show the plot
plt.show()
```

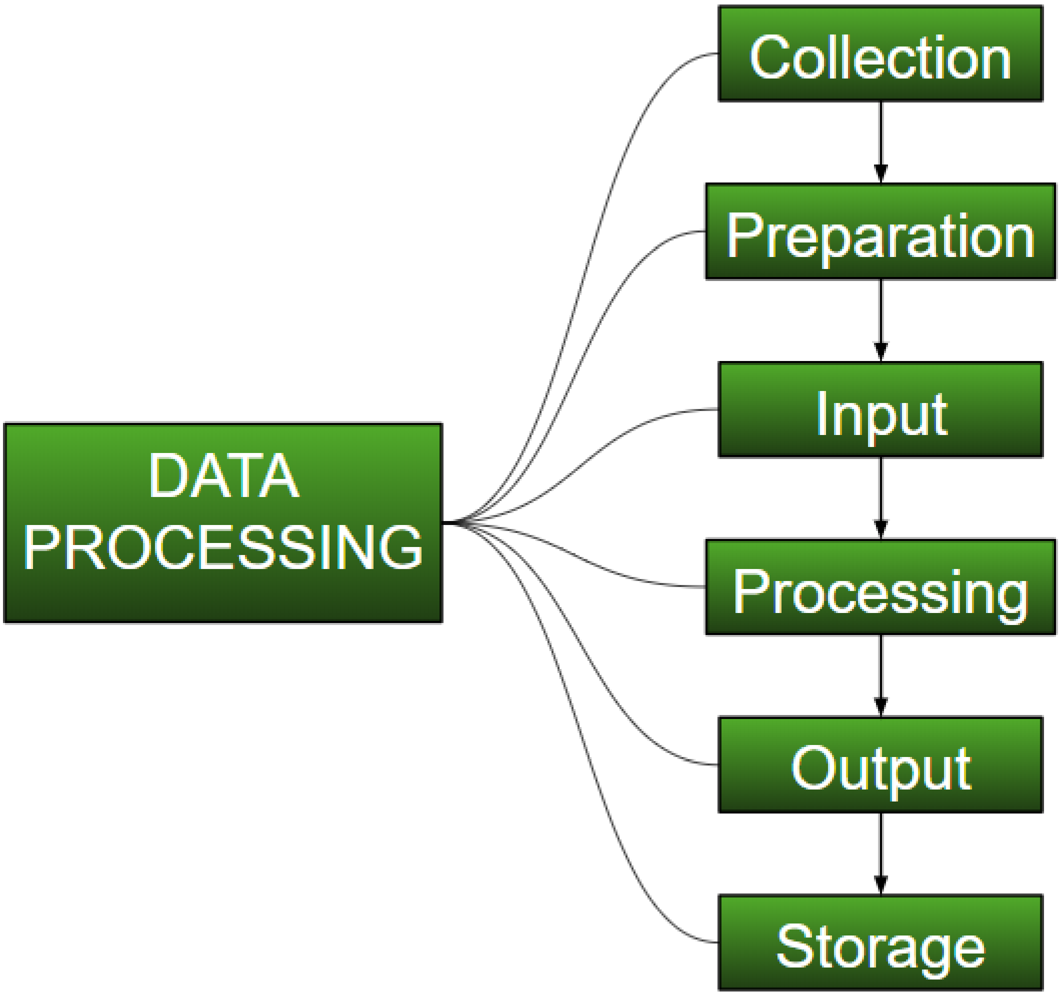
Output:-



X=set of constant values, Y=X

Data Processing

Data Processing is the task of converting data from a given form to a much more usable and desired form i.e. making it more meaningful and informative. Using Machine Learning algorithms, mathematical modeling, and statistical knowledge, this entire process can be automated. The output of this complete process can be in any desired form like graphs, videos, charts, tables, images, and many more, depending on the task we are performing and the requirements of the machine. This might seem to be simple but when it comes to massive organizations like Twitter, Facebook, Administrative bodies like Parliament, UNESCO, and health sector organizations, this entire process needs to be performed in a very structured manner. So, the steps to perform are as follows:



Classification: It is a data analysis task, i.e. the process of finding a model that describes and distinguishes data classes and concepts. Classification is the problem of identifying to which of a set of categories (subpopulations), a new observation belongs to, on the basis of a training set of data containing observations and whose categories membership is known.

Regression:-

When Regression is chosen?
A regression problem is when the output variable is a real or continuous value, such as “salary” or “weight”. Many different models can be used, the simplest is

linear regression. It tries to fit data with the best hyperplane which goes through the points.

Regression Analysis:- it is a statistical process for estimating the relationships between the dependent variables or criterion variables and one or more independent variables or predictors. Regression analysis explains the changes in criteria in relation to changes in select predictors. The conditional expectation of the criteria is based on predictors where the average value of the dependent variables is given when the independent variables are changed. Three major uses for regression analysis are determining the strength of predictors, forecasting an effect, and trend forecasting.

Types of Regression:

- **Linear regression**:- it is used for predictive analysis. Linear regression is a linear approach for modelling the relationship between the criterion or the scalar response and the multiple predictors or explanatory variables. Linear regression focuses on the conditional probability distribution of the response given the values of the predictors. For linear regression, there is a danger of overfitting. The formula for linear regression is: Y' = bX + A.

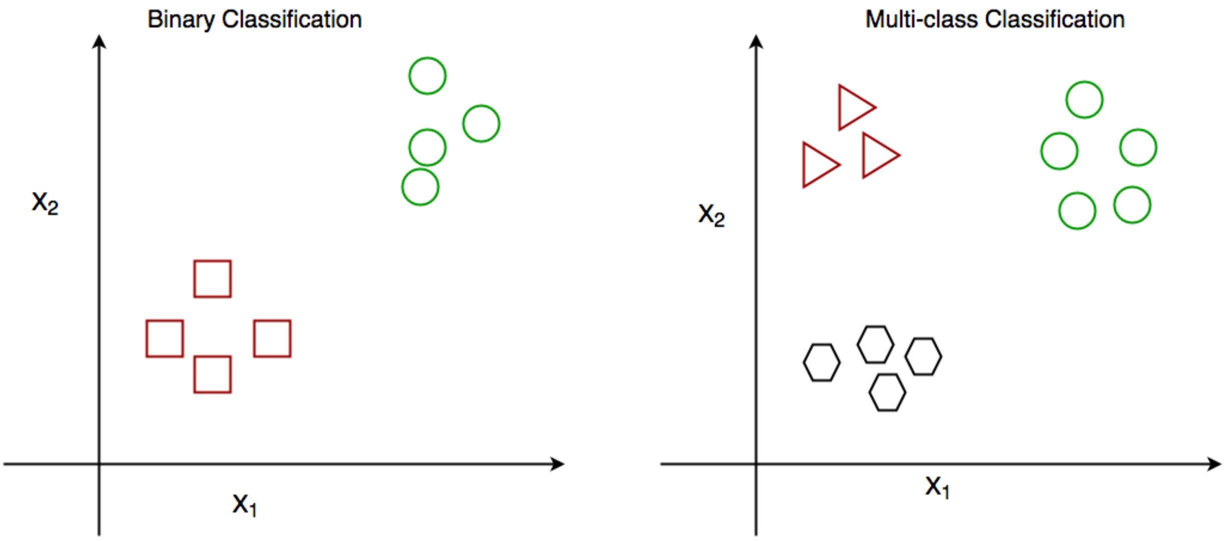
- **Polynomial regression**:- It is used for curvilinear data. Polynomial regression is fit with the method of least squares. The goal of regression analysis is to model the expected value of a dependent variable y in regards to the independent variable x. The equation for polynomial regression is:-

  $l = \beta_0 + \beta_0 x_1 + \epsilon.$

Classification vs Regression

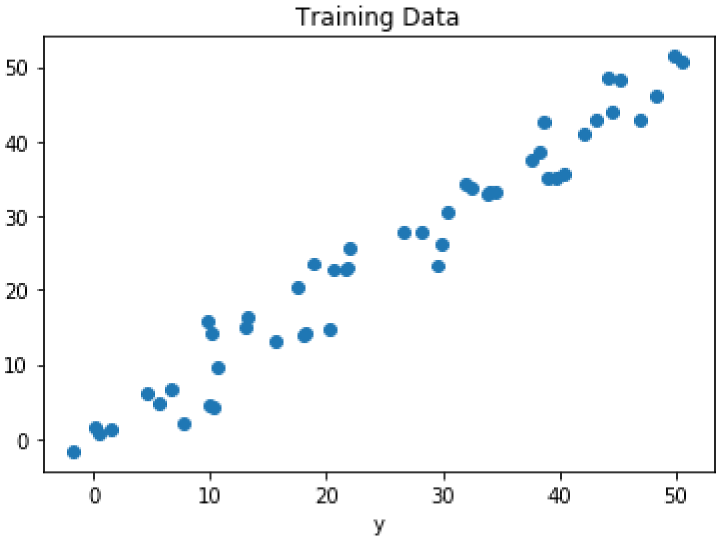
Classification:- In above, Let's take an example, suppose we want to predict the possibility of the winning of a match by Team A on the basis of some parameters recorded earlier. Then there would be two labels Yes and No.

X1, = indendent variables(some parameters related to Yes and No label)
X2= dependent variables



Regression:- Let’s take a similar example in regression also, where we are finding the possibility of rain in some particular regions with the help of some parameters recorded earlier. Then there is a probability associated with the

X



rain.

Fig: y= day,   X=rainfall (in mm)

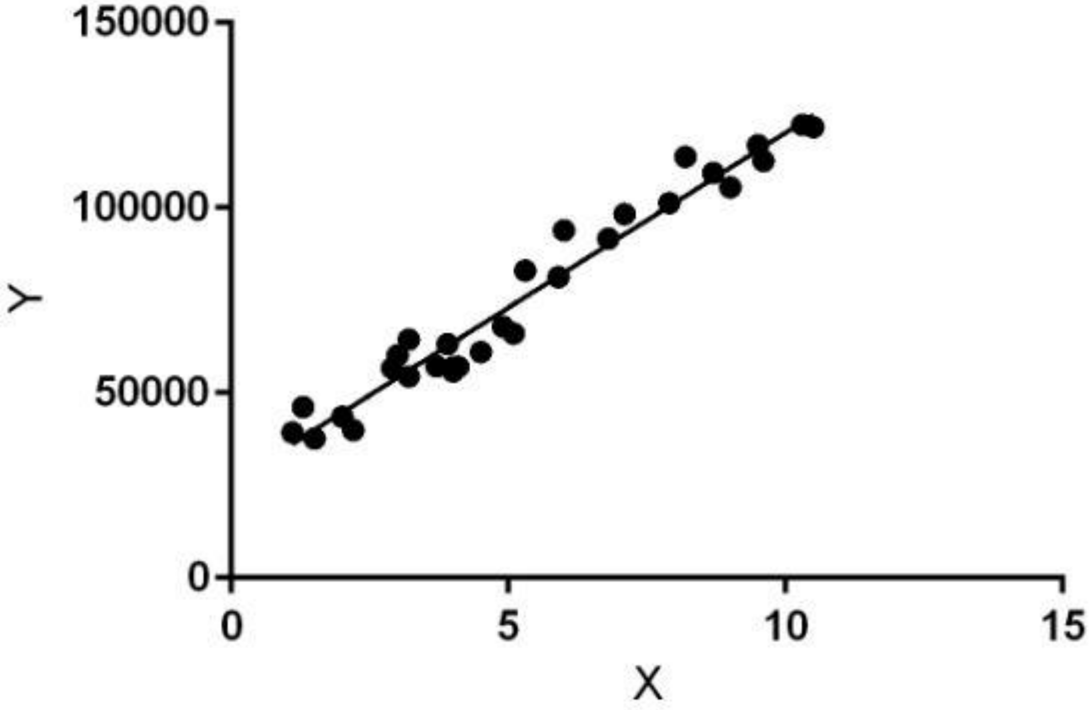
Comparison between Classification and Regression:

Parameter	CLASSIFICATION	REGRESSION
Basic	The mapping function is used for mapping values to predefined classes.	Mapping Function is used for the mapping of values to continuous output.

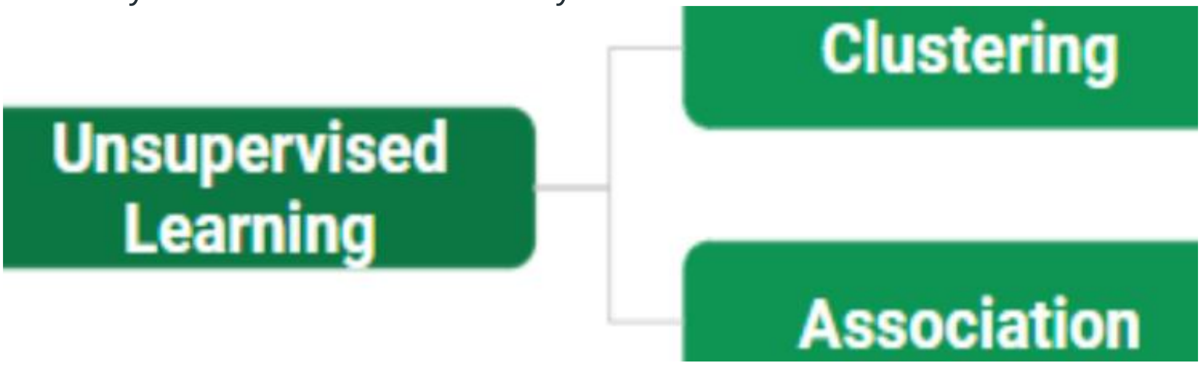
Parameter	CLASSIFICATION	REGRESSION
Involves prediction of	Discrete values	Continuous values
Nature of the predicted data	Unordered	Ordered
Method of calculation	by measuring accuracy	by measurement of root mean square error
Example Algorithms	Decision tree, logistic regression, etc.	Regression tree (Random forest), Linear regression, etc.

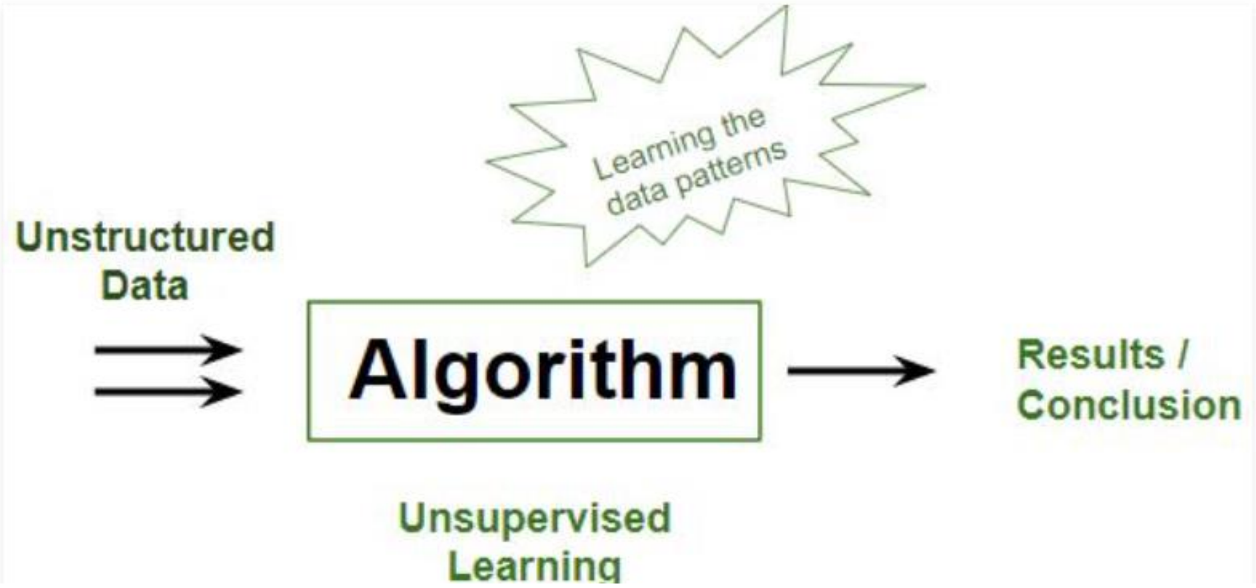
Linear Regression:- **Linear Regression** is a machine learning algorithm based on **supervised learning**. It performs a **regression task**. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used.

X (input) = work experience , Y (output)= salary of a person.



Unsupervised Learning :- It’s a type of learning where we don’t give a target to our model while training i.e. training model has only input parameter values. The model by itself has to find which way it can learn.





Reinforcement learning:- Reinforcement learning is an area of Machine Learning. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behavior or path it should take in a specific situation. Reinforcement learning differs from supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its experience.

Reinforcement learning	Supervised learning
Reinforcement learning is all about making decisions sequentially. In simple words, we can	In Supervised learning, the decision is made on the

Reinforcement learning	Supervised learning
say that the output depends on the state of the current input and the next input depends on the output of the previous input	initial input or the input given at the start
In Reinforcement learning decision is dependent, So we give labels to sequences of dependent decisions	In supervised learning the decisions are independent of each other so labels are given to each decision.
Example: Chess game	Example: Object recognition

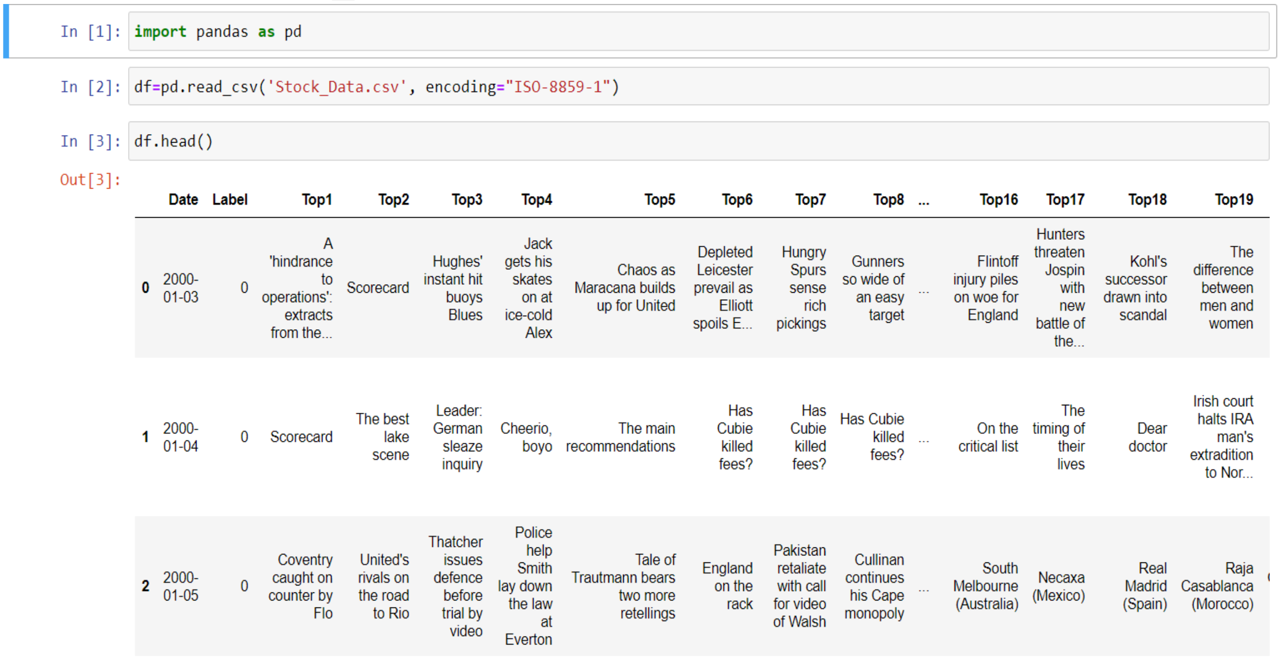
Internship Problem – Stock Prediction:-

Problem - Predict the stock market price of next few days using previous stock market data (equity or indices) using machine learning or Deep learning.

1. Use News headlines as Data for prediction.
2. Use previous Equity data of Day open, close, low, high for prediction.
3. Any other stock Relative data.

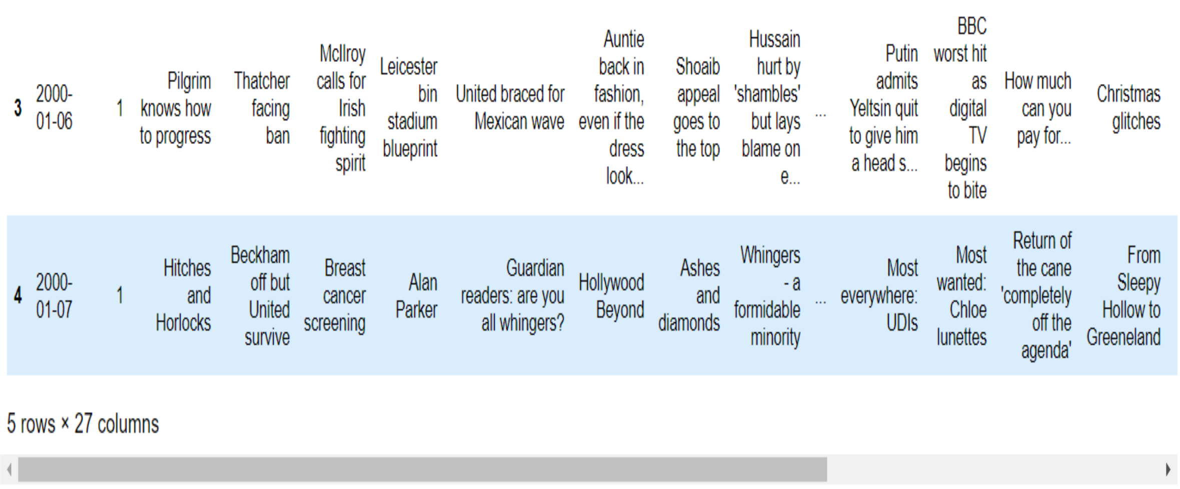
Note: Try to improve the accuracy of the previously built model or implement it from scratch.

SOLUTION:- part_1 screenshot



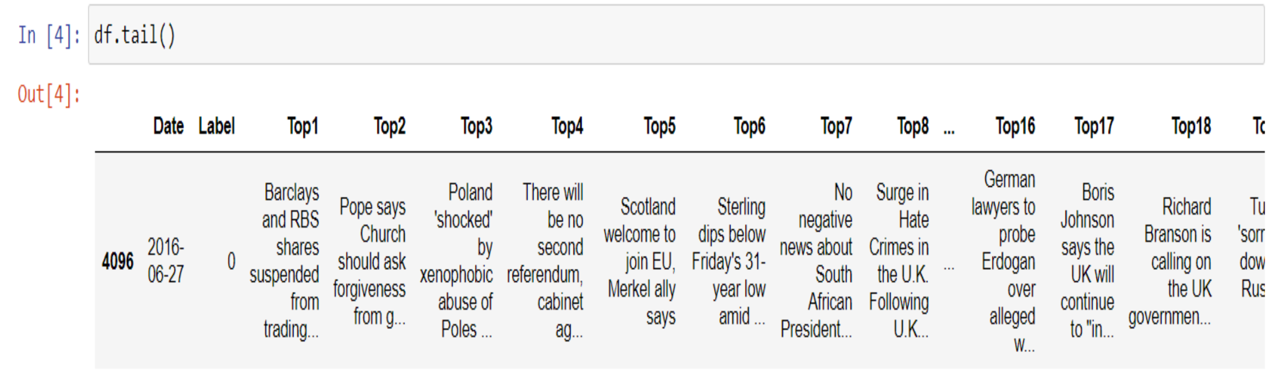
```
In [1]: import pandas as pd

In [2]: df=pd.read_csv('Stock_Data.csv', encoding="ISO-8859-1")

In [3]: df.head()

Out[3]:
```

	Date	Label	Top1	Top2	Top3	Top4	Top5	Top6	Top7	Top8	...	Top16	Top17	Top18	Top19
0	2000-01-03	0	A 'hindrance to operations': extracts from the...	Scorecard	Hughes' instant hit buoys Blues	Jack gets his skates on at ice-cold Alex	Chaos as Maracana builds up for United	Depleted Leicester prevail as Elliott spoils E...	Hungry Spurs sense rich pickings	Gunners so wide of an easy target	...	Flintoff injury piles on woe for England	Hunters threaten Jospin with new battle of the...	Kohl's successor drawn into scandal	The difference between men and women
1	2000-01-04	0	Scorecard	The best lake scene	Leader: German sleaze inquiry	Cheerio, boyo	The main recommendations	Has Cubie killed fees?	Has Cubie killed fees?	Has Cubie killed fees?	...	On the critical list	The timing of their lives	Dear doctor	Irish court halts IRA man's extradition to Nor...
2	2000-01-05	0	Coventry caught on counter by Flo	United's rivals on the road to Rio	Thatcher issues defence before trial by video	Police help Smith lay down the law at Everton	Tale of Trautmann bears two more retellings	England on the rack	Pakistan retaliate with call for video of Walsh	Cullinan continues his Cape monopoly	...	South Melbourne (Australia)	Necaxa (Mexico)	Real Madrid (Spain)	Raja Casablanca (Morocco)



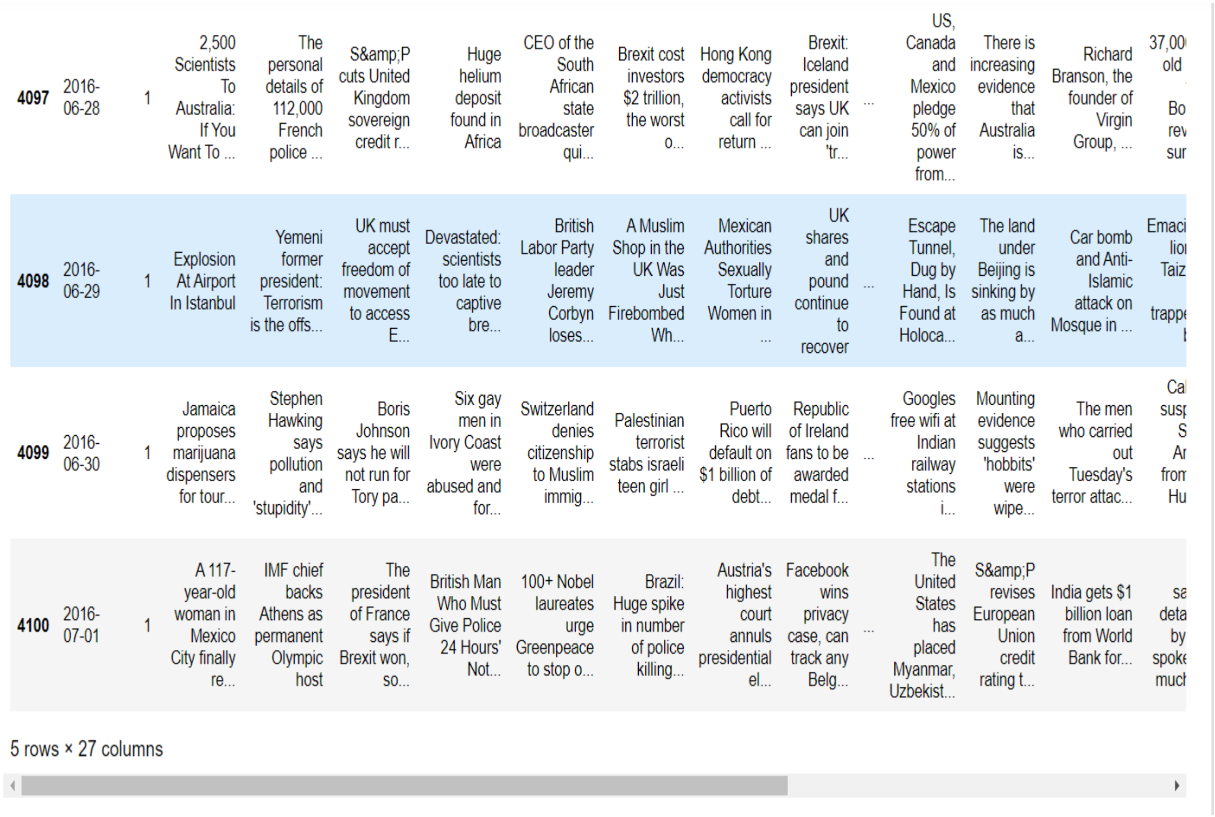
3	2000-01-06	1	Pilgrim knows how to progress	Thatcher facing ban	McIlroy calls for Irish fighting spirit	Leicester bin stadium blueprint	United braced for Mexican wave	Auntie back in fashion, even if the dress look...	Shoaib appeal goes to the top	Hussain hurt by 'shambles' but lays blame on e...	...	Putin admits Yeltsin quit to give him a head s...	BBC worst hit as digital TV begins to bite	How much can you pay for...	Christmas glitches
4	2000-01-07	1	Hitches and Horlocks	Beckham off but United survive	Breast cancer screening	Alan Parker	Guardian readers: are you all whingers?	Hollywood Beyond	Ashes and diamonds	Whingers - a formidable minority	...	Most everywhere: UDIs	Most wanted: Chloe lunettes	Return of the cane 'completely off the agenda'	From Sleepy Hollow to Greeneland

5 rows × 27 columns



In [4]:
```
df.tail()
```

Out[4]:



	Date	Label	Top1	Top2	Top3	Top4	Top5	Top6	Top7	Top8	...	Top16	Top17	Top18	Tc
4096	2016-06-27	0	Barclays and RBS shares suspended from trading...	Pope says Church should ask forgiveness from g...	Poland 'shocked' by xenophobic abuse of Poles ...	There will be no second referendum, cabinet ag...	Scotland welcome to join EU, Merkel ally says	Sterling dips below Friday's 31-year low amid ...	No negative news about South African President...	Surge in Hate Crimes in the U.K. Following U.K...	...	German lawyers to probe Erdogan over alleged w...	Boris Johnson says the UK will continue to "in...	Richard Branson is calling on the UK governmen...	Tu 'sorr dow Rus
4097	2016-06-28	1	2,500 Scientists To Australia: If You Want To ...	The personal details of 112,000 French police ...	S&amp;P cuts United Kingdom sovereign credit r...	Huge helium deposit found in Africa	CEO of the South African state broadcaster qui...	Brexit cost investors $2 trillion, the worst o...	Hong Kong democracy activists call for return ...	Brexit: Iceland president says UK can join 'tr...	...	US, Canada and Mexico pledge 50% of power from...	There is increasing evidence that Australia is...	Richard Branson, the founder of Virgin Group, ...	37,00 old Bo rev sur
4098	2016-06-29	1	Explosion At Airport In Istanbul	Yemeni former president: Terrorism is the offs...	UK must accept freedom of movement to access E...	Devastated: scientists too late to captive bre...	British Labor Party leader Jeremy Corbyn loses...	A Muslim Shop in the UK Was Just Firebombed Wh...	Mexican Authorities Sexually Torture Women in ...	UK shares and pound continue to recover	...	Escape Tunnel, Dug by Hand, Is Found at Holoca...	The land under Beijing is sinking by as much a...	Car bomb and Anti-Islamic attack on Mosque in ...	Emaci lio Taiz trappe
4099	2016-06-30	1	Jamaica proposes marijuana dispensers for tour...	Stephen Hawking says pollution and 'stupidity'...	Boris Johnson says he will not run for Tory pa...	Six gay men in Ivory Coast were abused and for...	Switzerland denies citizenship to Muslim immig...	Palestinian terrorist stabs israeli teen girl ...	Puerto Rico will default on $1 billion of debt...	Republic of Ireland fans to be awarded medal f...	...	Googles free wifi at Indian railway stations i...	Mounting evidence suggests 'hobbits' were wipe...	The men who carried out Tuesday's terror attac...	Ca susp S Ar from Hu
4100	2016-07-01	1	A 117-year-old woman in Mexico City finally re...	IMF chief backs Athens as permanent Olympic host	The president of France says if Brexit won, so...	British Man Who Must Give Police 24 Hours' Not...	100+ Nobel laureates urge Greenpeace to stop o...	Brazil: Huge spike in number of police killing...	Austria's highest court annuls presidential el...	Facebook wins privacy case, can track any Belg...	...	The United States has placed Myanmar, Uzbekist...	S&amp;P revises European Union credit rating t...	India gets $1 billion loan from World Bank for...	sa deta by spoke much

5 rows × 27 columns

In [5]:
```
train=df[df['Date'] < '20150101']
test=df[df['Date'] > '20141231']
```

In [6]:
```
#removing punctuation
data=train.iloc[:, 2:27]
data.replace("[^a-zA-Z]", " ", regex=True, inplace=True)

#removing column name for ease of access
list1=[i for i in range(25)]
new_index=[str(i) for i in list1]
data.columns=new_index

#converting headline to lower case
for index in new_index:
    data[index]=data[index].str.lower()
data.head(1)
```

Out[6]:

	0	1	2	3	4	5	6	7	8	9	...	15	16	17	18	19	20	
0	a hindrance to operations extracts from the...	scorecard	hughes instant hit buoys blues	jack gets his skates on at ice cold alex	chaos as maracana builds up for united	depleted leicester prevail as elliott spoils e...	hungry spurs sense rich pickings	gunners so wide of an easy target	derby raise a glass to strupar s debut double	southgate strikes leeds pay the penalty	...	flintoff injury piles on woe for england	hunters threaten jospin with new battle of the...	kohl s successor drawn into scandal	the difference between men and women	sara denver nurse turned solicitor	diana s landmine crusade put tories in a panic	re c

1 rows × 25 columns

In [7]:
```
#combinnig all the 25 heading
' '.join(str(x) for x in data.iloc[1, 0:25])
```

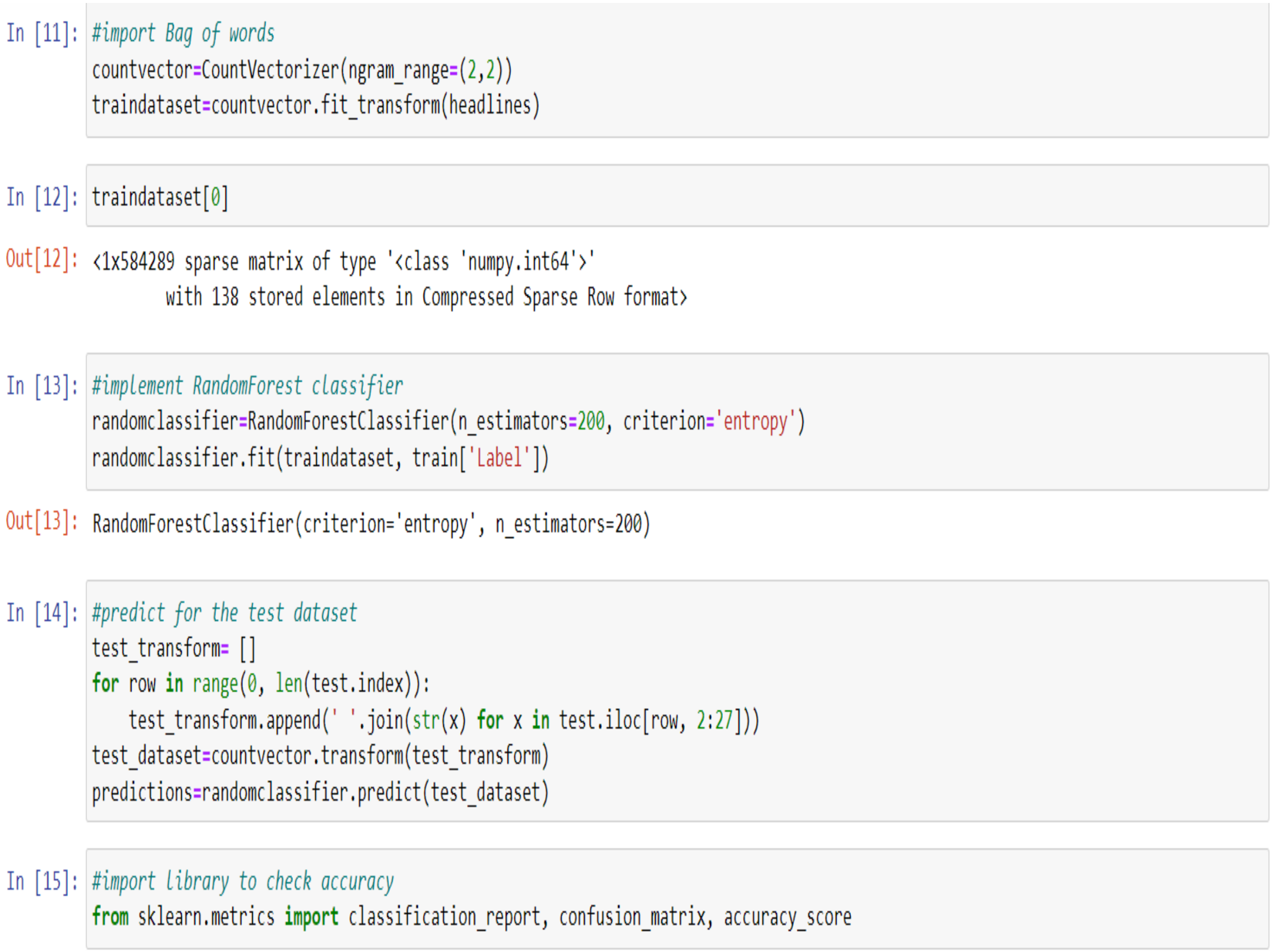
Out[7]:
```
'scorecard the best lake scene leader  german sleaze inquiry cheerio  boyo the main recommendations has cubie killed fees  has
cubie killed fees  has cubie killed fees  hopkins  furious  at foster s lack of hannibal appetite has cubie killed fees  a tale
of two tails i say what i like and i like what i say elbows  eyes and nipples task force to assess risk of asteroid collision h
ow i found myself at last on the critical list the timing of their lives dear doctor irish court halts ira man s extradition to
northern ireland burundi peace initiative fades after rebels reject mandela as mediator pe points the way forward to the ecb ca
mpaigners keep up pressure on nazi war crimes suspect jane ratcliffe yet more things you wouldn t know without the movies mille
nnium bug fails to bite'
```

In [8]:
```
headlines=[]
for row in range(0, len(data.index)):
    headlines.append(' '.join(str(x) for x in data.iloc[row, 0:25]))
```

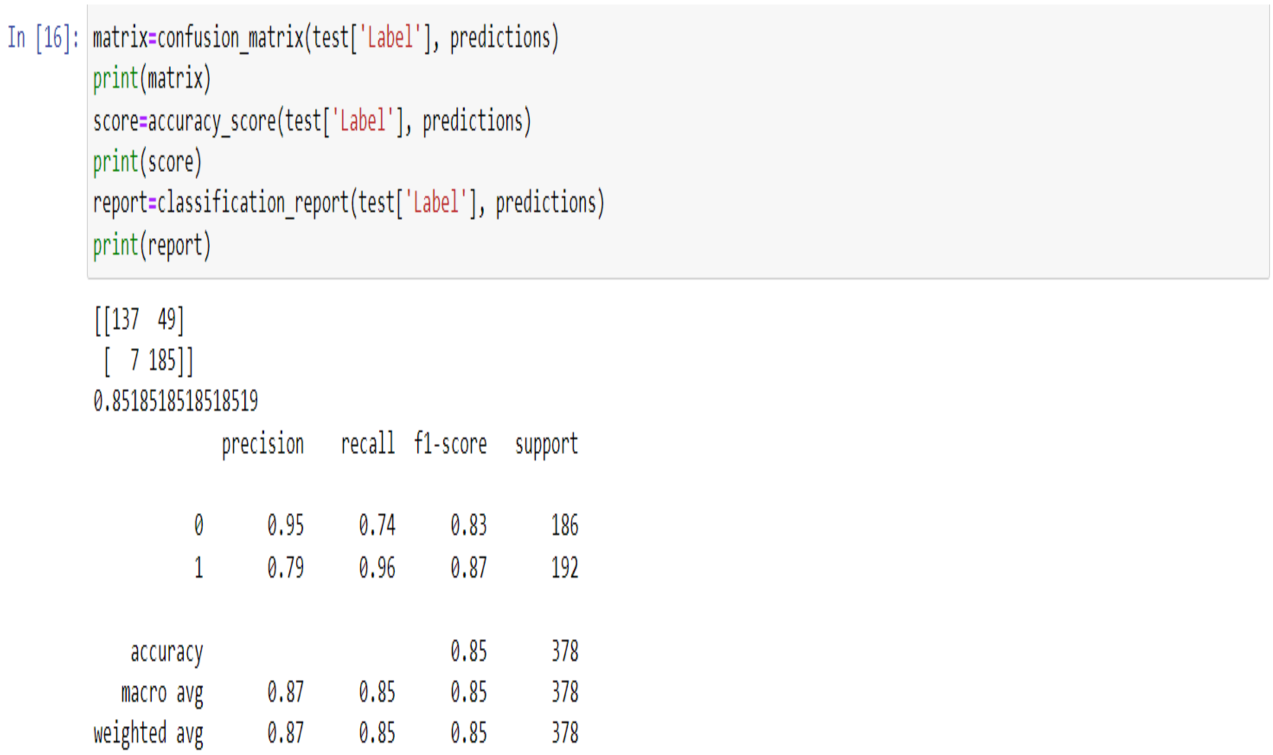
In [9]:
```
headlines[2]
```

Out[9]:
```
'coventry caught on counter by flo united s rivals on the road to rio thatcher issues defence before trial by video police help
smith lay down the law at everton tale of trautmann bears two more retellings england on the rack pakistan retaliate with call
for video of walsh cullinan continues his cape monopoly mcgrath puts india out of their misery blair witch bandwagon rolls on p
ele turns up heat on ferguson party divided over kohl slush fund scandal manchester united  england  women in record south pole
walk vasco da gama  brazil  south melbourne  australia  necaxa  mexico  real madrid  spain  raja casablanca  morocco  corinthia
ns  brazil  tony s pet project al nassr  saudi arabia  ideal holmes show pinochet leaves hospital after tests useful links'
```

In [10]:
```
#CountVectorizer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import RandomForestClassifier
```

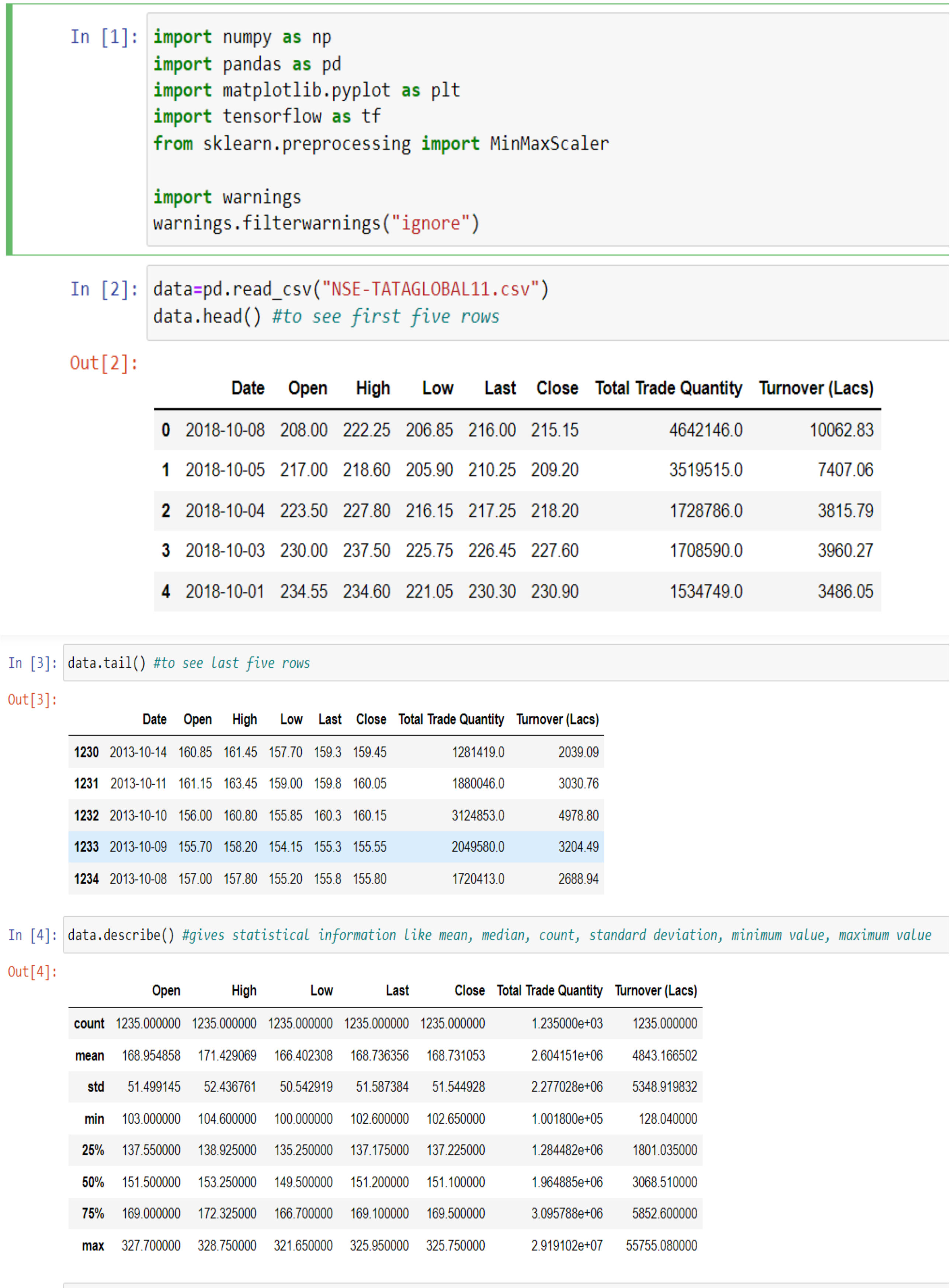


```
In [11]: #import Bag of words
         countvector=CountVectorizer(ngram_range=(2,2))
         traindataset=countvector.fit_transform(headlines)

In [12]: traindataset[0]

Out[12]: <1x584289 sparse matrix of type '<class 'numpy.int64'>'
                 with 138 stored elements in Compressed Sparse Row format>

In [13]: #implement RandomForest classifier
         randomclassifier=RandomForestClassifier(n_estimators=200, criterion='entropy')
         randomclassifier.fit(traindataset, train['Label'])

Out[13]: RandomForestClassifier(criterion='entropy', n_estimators=200)

In [14]: #predict for the test dataset
         test_transform= []
         for row in range(0, len(test.index)):
             test_transform.append(' '.join(str(x) for x in test.iloc[row, 2:27]))
         test_dataset=countvector.transform(test_transform)
         predictions=randomclassifier.predict(test_dataset)

In [15]: #import library to check accuracy
         from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
```



```
In [16]: matrix=confusion_matrix(test['Label'], predictions)
         print(matrix)
         score=accuracy_score(test['Label'], predictions)
         print(score)
         report=classification_report(test['Label'], predictions)
         print(report)

         [[137  49]
          [  7 185]]
         0.8518518518518519
                       precision    recall  f1-score   support

                    0       0.95      0.74      0.83       186
                    1       0.79      0.96      0.87       192

             accuracy                           0.85       378
            macro avg       0.87      0.85      0.85       378
         weighted avg       0.87      0.85      0.85       378
```

part_2 screenshot



In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import tensorflow as tf
from sklearn.preprocessing import MinMaxScaler

import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
data=pd.read_csv("NSE-TATAGLOBAL11.csv")
data.head() #to see first five rows
```

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

In [3]:
```
data.tail() #to see last five rows
```

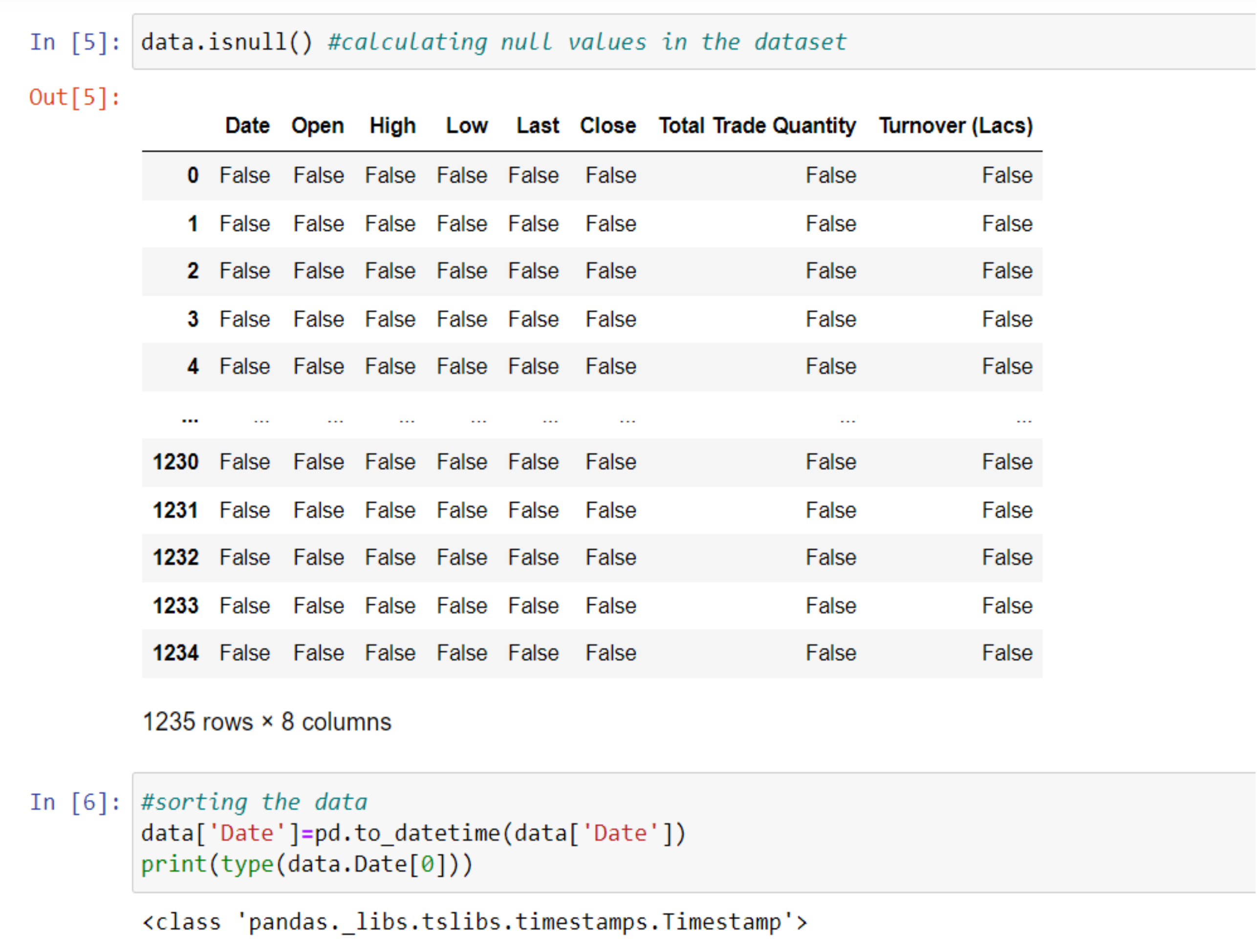
Out[3]:

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
1230	2013-10-14	160.85	161.45	157.70	159.3	159.45	1281419.0	2039.09
1231	2013-10-11	161.15	163.45	159.00	159.8	160.05	1880046.0	3030.76
1232	2013-10-10	156.00	160.80	155.85	160.3	160.15	3124853.0	4978.80
1233	2013-10-09	155.70	158.20	154.15	155.3	155.55	2049580.0	3204.49
1234	2013-10-08	157.00	157.80	155.20	155.8	155.80	1720413.0	2688.94

In [4]:
```
data.describe() #gives statistical information like mean, median, count, standard deviation, minimum value, maximum value
```

Out[4]:

	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
count	1235.000000	1235.000000	1235.000000	1235.000000	1235.000000	1.235000e+03	1235.000000
mean	168.954858	171.429069	166.402308	168.736356	168.731053	2.604151e+06	4843.166502
std	51.499145	52.436761	50.542919	51.587384	51.544928	2.277028e+06	5348.919832
min	103.000000	104.600000	100.000000	102.600000	102.650000	1.001800e+05	128.040000
25%	137.550000	138.925000	135.250000	137.175000	137.225000	1.284482e+06	1801.035000
50%	151.500000	153.250000	149.500000	151.200000	151.100000	1.964885e+06	3068.510000
75%	169.000000	172.325000	166.700000	169.100000	169.500000	3.095788e+06	5852.600000
max	327.700000	328.750000	321.650000	325.950000	325.750000	2.919102e+07	55755.080000



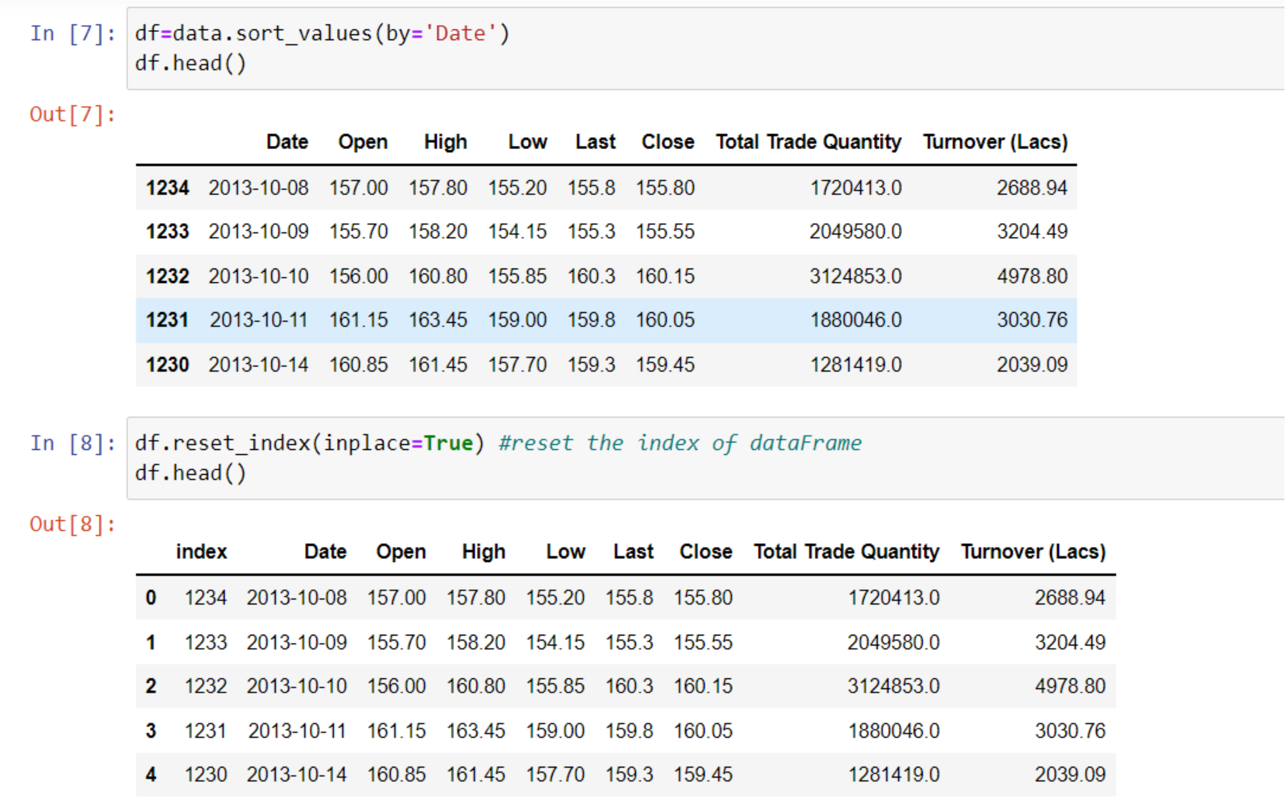
```
In [5]: data.isnull() #calculating null values in the dataset
```

Out[5]:

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
1230	False	False	False	False	False	False	False	False
1231	False	False	False	False	False	False	False	False
1232	False	False	False	False	False	False	False	False
1233	False	False	False	False	False	False	False	False
1234	False	False	False	False	False	False	False	False

1235 rows × 8 columns

```
In [6]: #sorting the data
        data['Date']=pd.to_datetime(data['Date'])
        print(type(data.Date[0]))

        <class 'pandas._libs.tslibs.timestamps.Timestamp'>
```



```
In [7]: df=data.sort_values(by='Date')
        df.head()
```

Out[7]:

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
1234	2013-10-08	157.00	157.80	155.20	155.8	155.80	1720413.0	2688.94
1233	2013-10-09	155.70	158.20	154.15	155.3	155.55	2049580.0	3204.49
1232	2013-10-10	156.00	160.80	155.85	160.3	160.15	3124853.0	4978.80
1231	2013-10-11	161.15	163.45	159.00	159.8	160.05	1880046.0	3030.76
1230	2013-10-14	160.85	161.45	157.70	159.3	159.45	1281419.0	2039.09

```
In [8]: df.reset_index(inplace=True) #reset the index of dataFrame
        df.head()
```

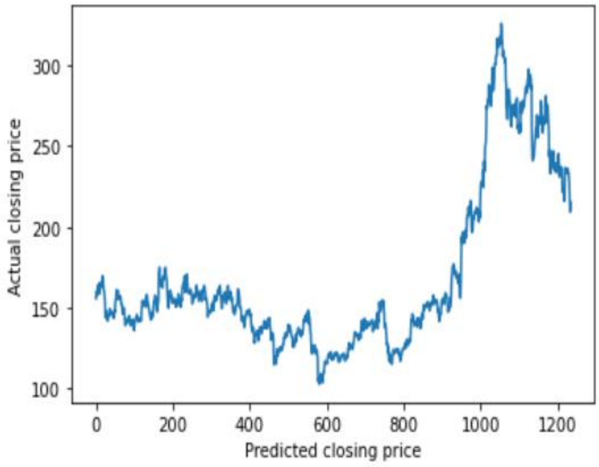
Out[8]:

	index	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	1234	2013-10-08	157.00	157.80	155.20	155.8	155.80	1720413.0	2688.94
1	1233	2013-10-09	155.70	158.20	154.15	155.3	155.55	2049580.0	3204.49
2	1232	2013-10-10	156.00	160.80	155.85	160.3	160.15	3124853.0	4978.80
3	1231	2013-10-11	161.15	163.45	159.00	159.8	160.05	1880046.0	3030.76
4	1230	2013-10-14	160.85	161.45	157.70	159.3	159.45	1281419.0	2039.09

Data Visualization

```
In [9]: plt.plot(df['Close'])
        plt.xlabel("Predicted closing price")
        plt.ylabel("Actual closing price")
```

```
Out[9]: Text(0, 0.5, 'Actual closing price')
```



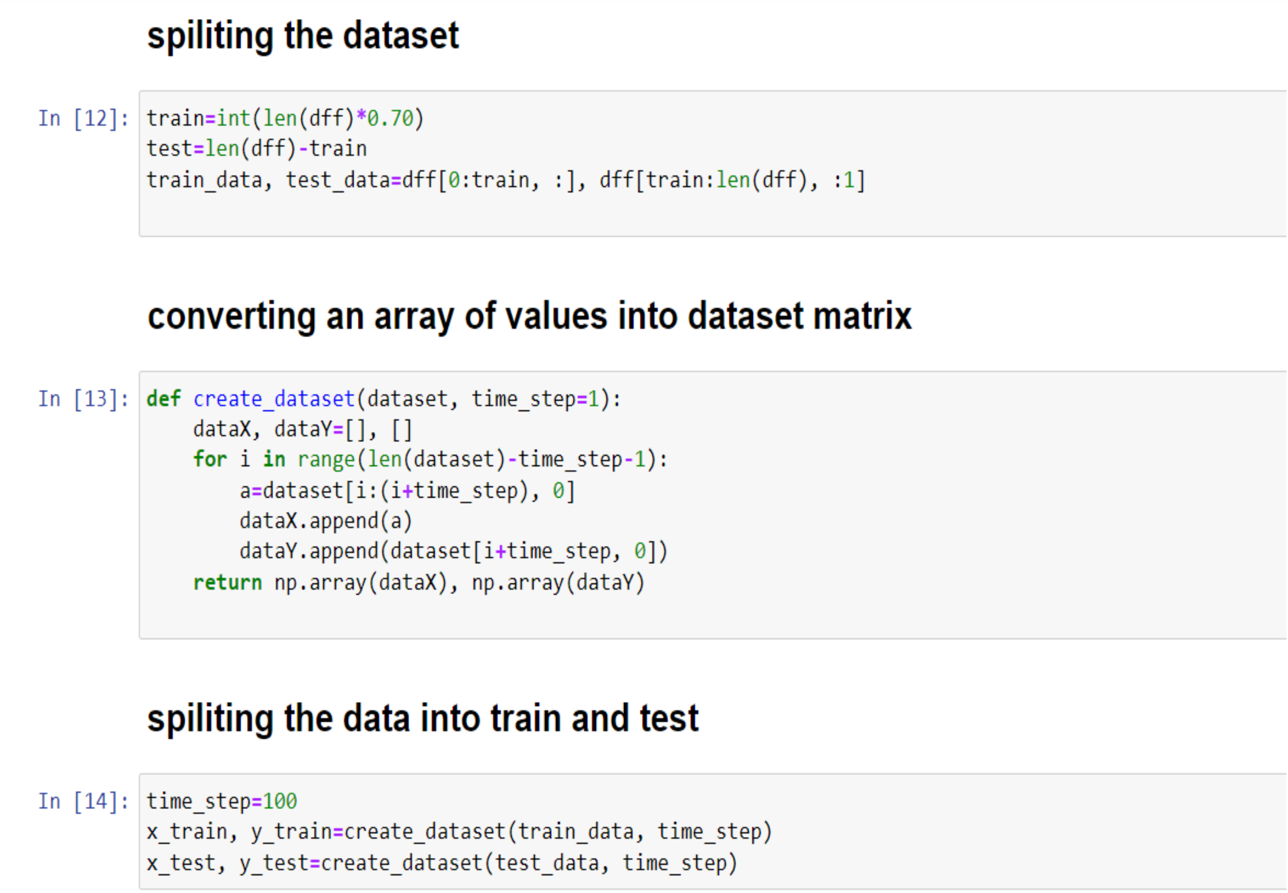
```
In [10]: dff=df['Close']
         dff
```

```
Out[10]: 0       155.80
         1       155.55
         2       160.15
         3       160.05
         4       159.45
                  ...
         1230    230.90
         1231    227.60
         1232    218.20
         1233    209.20
         1234    215.15
         Name: Close, Length: 1235, dtype: float64
```

Min Max Scaler

```
In [11]: scaler=MinMaxScaler(feature_range=(0,1))
         dff=scaler.fit_transform(np.array(dff).reshape(-1, 1))
         dff
```

```
Out[11]: array([[0.23823398],
                [0.2371134 ],
                [0.25773196],
                ...,
                [0.51792918],
                [0.47758853],
                [0.50425818]])
```



spiliting the dataset

In [12]:

```
train=int(len(dff)*0.70)
test=len(dff)-train
train_data, test_data=dff[0:train, :], dff[train:len(dff), :1]
```

converting an array of values into dataset matrix

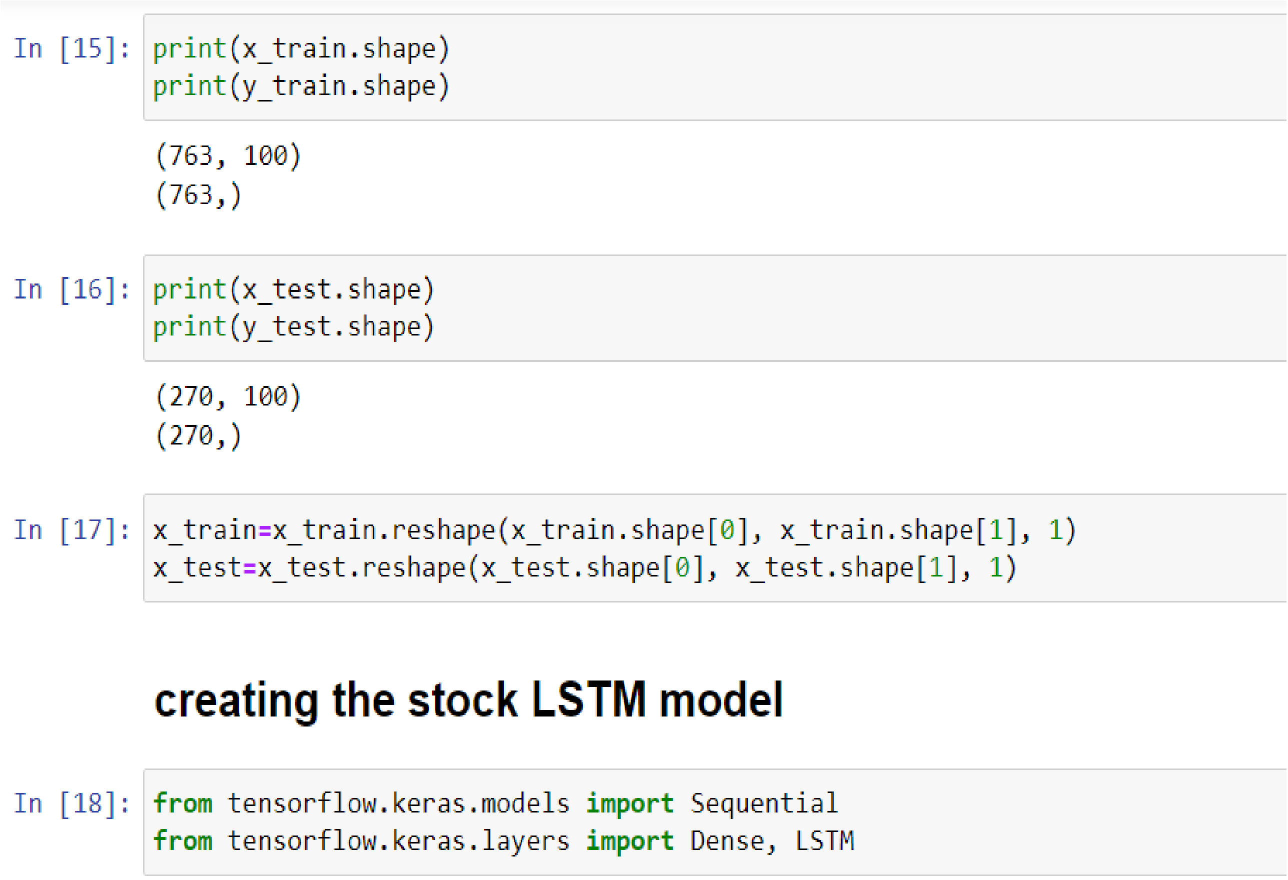
In [13]:

```
def create_dataset(dataset, time_step=1):
    dataX, dataY=[], []
    for i in range(len(dataset)-time_step-1):
        a=dataset[i:(i+time_step), 0]
        dataX.append(a)
        dataY.append(dataset[i+time_step, 0])
    return np.array(dataX), np.array(dataY)
```

spiliting the data into train and test

In [14]:

```
time_step=100
x_train, y_train=create_dataset(train_data, time_step)
x_test, y_test=create_dataset(test_data, time_step)
```



In [15]:

```
print(x_train.shape)
print(y_train.shape)
```

```
(763, 100)
(763,)
```

In [16]:

```
print(x_test.shape)
print(y_test.shape)
```

```
(270, 100)
(270,)
```

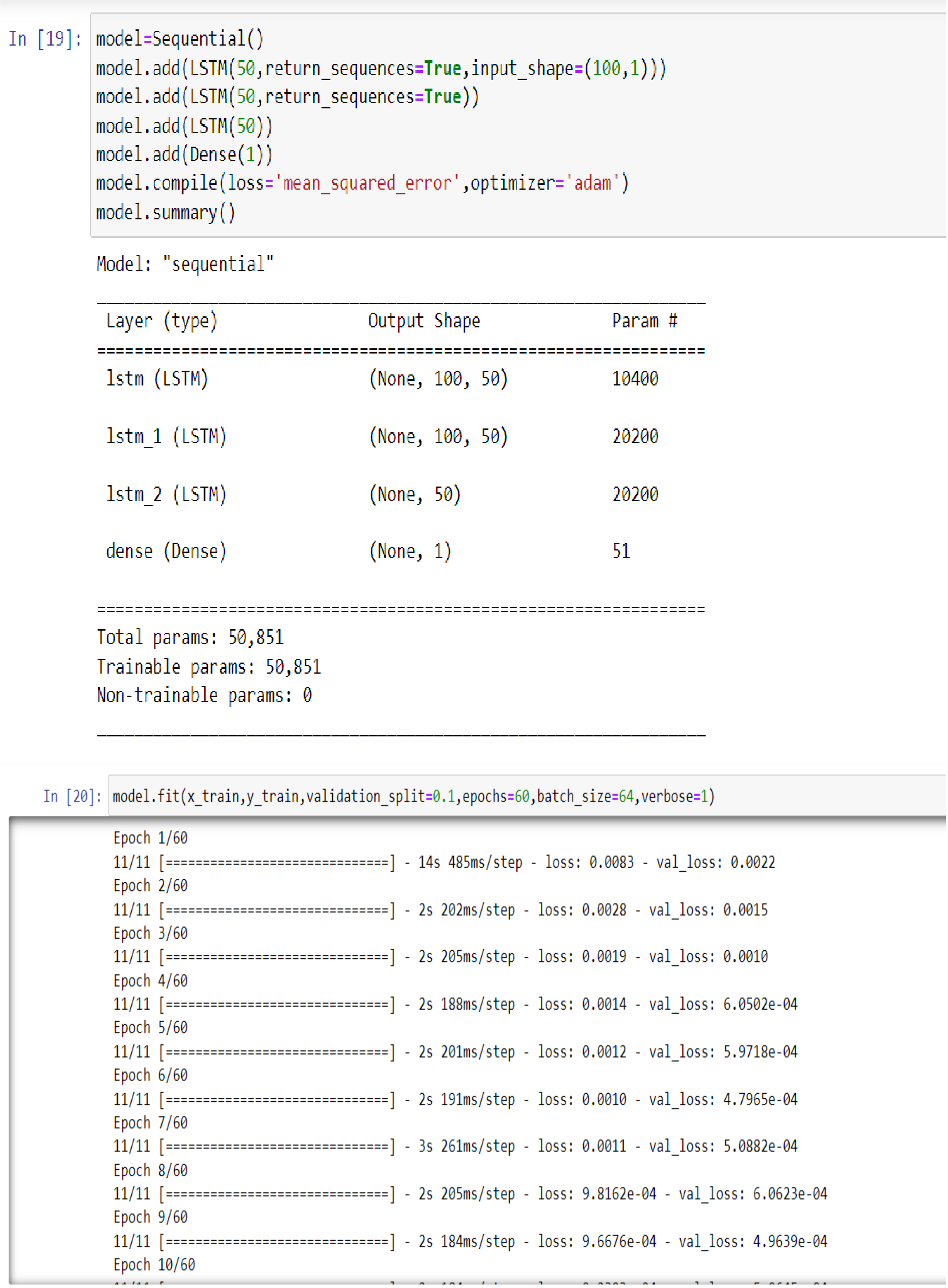
In [17]:

```
x_train=x_train.reshape(x_train.shape[0], x_train.shape[1], 1)
x_test=x_test.reshape(x_test.shape[0], x_test.shape[1], 1)
```

creating the stock LSTM model

In [18]:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM
```

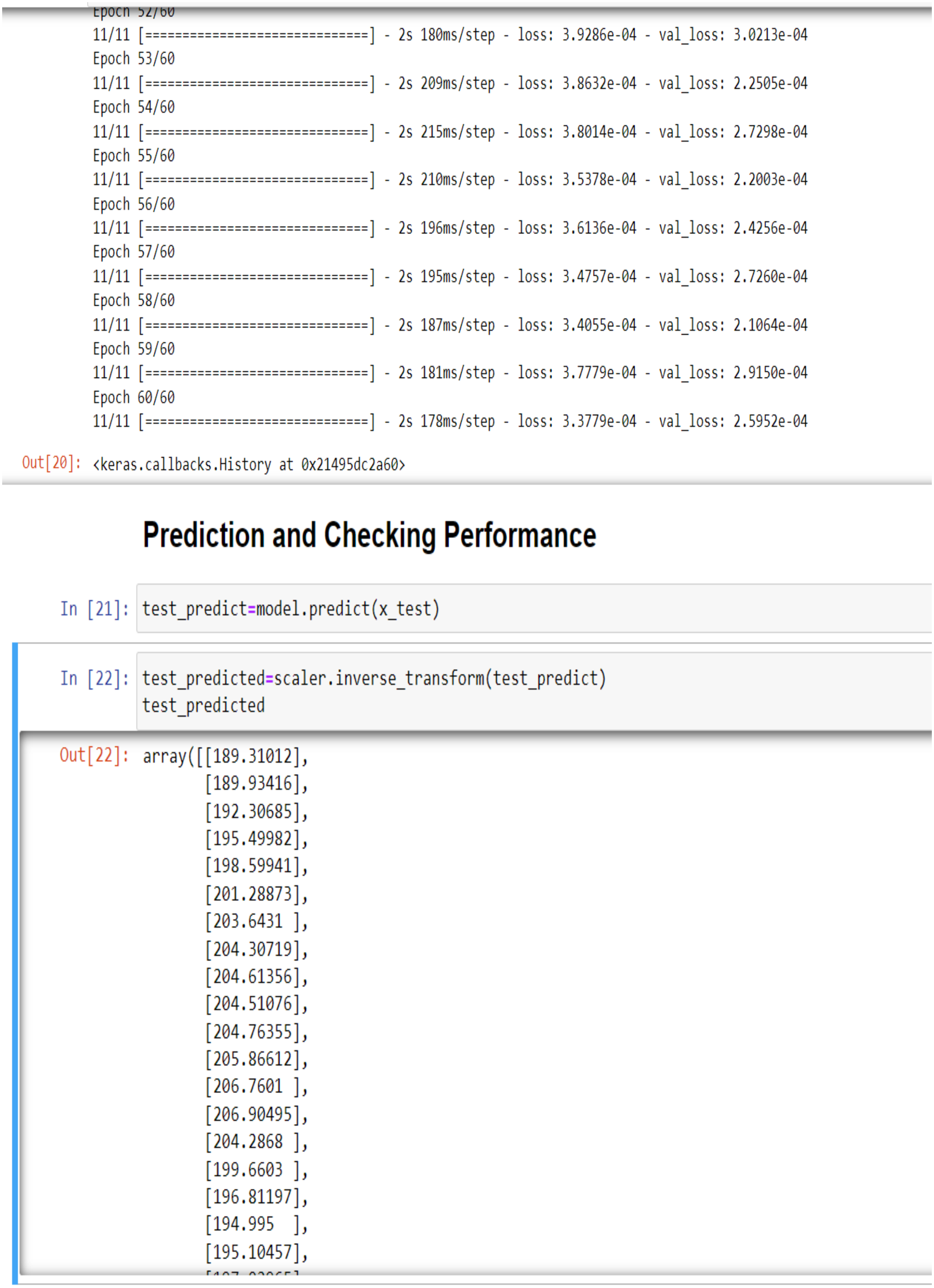


In [19]:
```python
model=Sequential()
model.add(LSTM(50,return_sequences=True,input_shape=(100,1)))
model.add(LSTM(50,return_sequences=True))
model.add(LSTM(50))
model.add(Dense(1))
model.compile(loss='mean_squared_error',optimizer='adam')
model.summary()
```

```
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
_________________________________________________________________
```

In [20]:
```python
model.fit(x_train,y_train,validation_split=0.1,epochs=60,batch_size=64,verbose=1)
```

```
Epoch 1/60
11/11 [==============================] - 14s 485ms/step - loss: 0.0083 - val_loss: 0.0022
Epoch 2/60
11/11 [==============================] - 2s 202ms/step - loss: 0.0028 - val_loss: 0.0015
Epoch 3/60
11/11 [==============================] - 2s 205ms/step - loss: 0.0019 - val_loss: 0.0010
Epoch 4/60
11/11 [==============================] - 2s 188ms/step - loss: 0.0014 - val_loss: 6.0502e-04
Epoch 5/60
11/11 [==============================] - 2s 201ms/step - loss: 0.0012 - val_loss: 5.9718e-04
Epoch 6/60
11/11 [==============================] - 2s 191ms/step - loss: 0.0010 - val_loss: 4.7965e-04
Epoch 7/60
11/11 [==============================] - 3s 261ms/step - loss: 0.0011 - val_loss: 5.0882e-04
Epoch 8/60
11/11 [==============================] - 2s 205ms/step - loss: 9.8162e-04 - val_loss: 6.0623e-04
Epoch 9/60
11/11 [==============================] - 2s 184ms/step - loss: 9.6676e-04 - val_loss: 4.9639e-04
Epoch 10/60
```



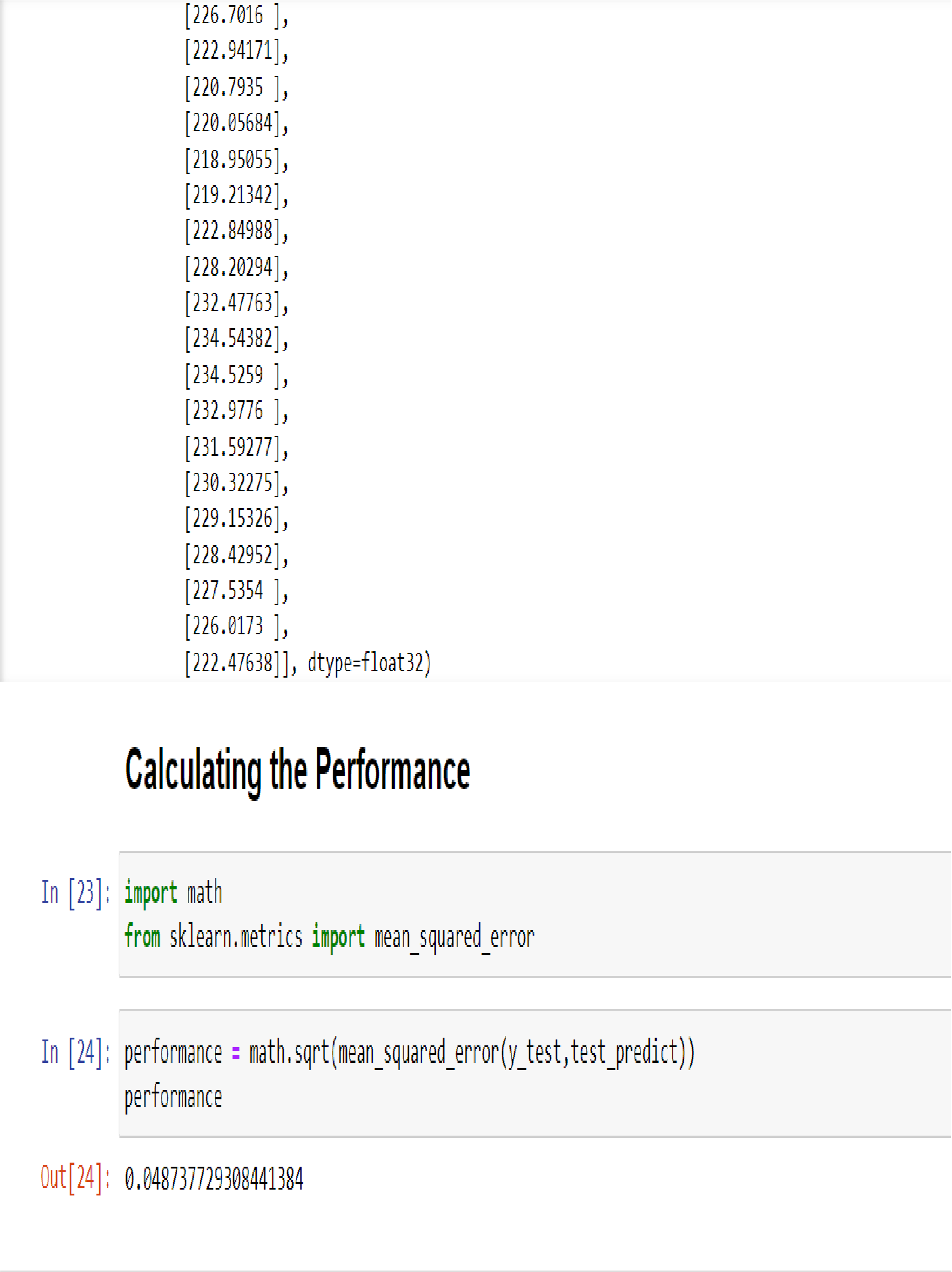
```
Epoch 52/60
11/11 [==============================] - 2s 180ms/step - loss: 3.9286e-04 - val_loss: 3.0213e-04
Epoch 53/60
11/11 [==============================] - 2s 209ms/step - loss: 3.8632e-04 - val_loss: 2.2505e-04
Epoch 54/60
11/11 [==============================] - 2s 215ms/step - loss: 3.8014e-04 - val_loss: 2.7298e-04
Epoch 55/60
11/11 [==============================] - 2s 210ms/step - loss: 3.5378e-04 - val_loss: 2.2003e-04
Epoch 56/60
11/11 [==============================] - 2s 196ms/step - loss: 3.6136e-04 - val_loss: 2.4256e-04
Epoch 57/60
11/11 [==============================] - 2s 195ms/step - loss: 3.4757e-04 - val_loss: 2.7260e-04
Epoch 58/60
11/11 [==============================] - 2s 187ms/step - loss: 3.4055e-04 - val_loss: 2.1064e-04
Epoch 59/60
11/11 [==============================] - 2s 181ms/step - loss: 3.7779e-04 - val_loss: 2.9150e-04
Epoch 60/60
11/11 [==============================] - 2s 178ms/step - loss: 3.3779e-04 - val_loss: 2.5952e-04

Out[20]: <keras.callbacks.History at 0x21495dc2a60>
```

Prediction and Checking Performance

```
In [21]: test_predict=model.predict(x_test)
```

```
In [22]: test_predicted=scaler.inverse_transform(test_predict)
         test_predicted
```

```
Out[22]: array([[189.31012],
                [189.93416],
                [192.30685],
                [195.49982],
                [198.59941],
                [201.28873],
                [203.6431 ],
                [204.30719],
                [204.61356],
                [204.51076],
                [204.76355],
                [205.86612],
                [206.7601 ],
                [206.90495],
                [204.2868 ],
                [199.6603 ],
                [196.81197],
                [194.995  ],
                [195.10457],
```



```
       [226.7016 ],
       [222.94171],
       [220.7935 ],
       [220.05684],
       [218.95055],
       [219.21342],
       [222.84988],
       [228.20294],
       [232.47763],
       [234.54382],
       [234.5259 ],
       [232.9776 ],
       [231.59277],
       [230.32275],
       [229.15326],
       [228.42952],
       [227.5354 ],
       [226.0173 ],
       [222.47638]], dtype=float32)
```

Calculating the Performance

```
In [23]: import math
         from sklearn.metrics import mean_squared_error
```

```
In [24]: performance = math.sqrt(mean_squared_error(y_test,test_predict))
         performance
```

```
Out[24]: 0.048737729308441384
```

Conclusion:-

We have seen basic of ML/DL, some built in module in python for ML(like- numpy, pandas, matplotlib) , linear regression, KNN,LSTM to get better prediction in stock market. We also see **Some time series forecasting techaniques(LSTM, Auto, ARIMA, etc).** LSTM methods gives us a better understanding then others and it also used in widely used for sequence prediction problems and has prooven to extremly effective. We also see some basic of Unsupervised Learning, Reinforcement Learning.

REFERENCES:-

- GeeksforGeeks
    - https://www.geeksforgeeks.org/machine-learning/
- NPTEL machine learning
    - https://youtube.com/playlist?list=PL3pGy4HtqwD2a57wl7Cl7tmfxfk7JWJ9Y
- : Chris Naik channel (reference)
    - https://youtube.com/playlist?list=PLZoTAELRMXVPBTrWtJkn3wWQxZkmTXGwe
- 4: code with harry channel(Hindi + English)

    https://youtube.com/playlist?list=PLu0W_9lII9ai6fAMHp-acBmJONT7Y4BSG