introduction to Machine Learning

numpy

import libs

```
import numpy as np
# importing numpy
```

```
my_list = [1, 2, 3, 4]
# creating a list
my_list
```

```
[1, 2, 3, 4]
```

1-D array

```
list1 = np.array(my_list)
# creating 1-D ndarray from a list
list1
```

```
array([1, 2, 3, 4])
```

```
np.array(range(15))
# creating ndarray using range()
```

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14])
```

```
mylist2 = np.array(range(1, 5))
# creating ndarray using range()
mylist2
```

```
array([1, 2, 3, 4])
```

```
mylist2.ndim
# printing dimensions of ndarray
```

```
1
```

2-D array

```
np.array([range(1, 4), range(4, 7)])
# creating a 2-D ndarray using range()
```

```
array([[1, 2, 3],
       [4, 5, 6]])
```

```
b = np.array([[9, 8, 7], [6, 5, 4]])
# creating a 2-D ndarray
b
```

```
array([[9, 8, 7],
       [6, 5, 4]])
```

```
b1 = np.array([[9.0, 8.0, 7.0], [6.0, 5.0, 4.0]])
# # creating a 2-D ndarray of decimal values
b1
```

```
array([[9., 8., 7.],
       [6., 5., 4.]])
```

3x3 matrix using 2-D ndarrray

```
matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
# creates 3x3 list
matrix
```

```
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

```
a1 = np.array(matrix)
# creates 3x3 matrix / ndarray
a1
```

```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

```
a1.ndim
# checking dimensions of 3x3 matrix
```

```
2
```

```
a1.shape
# checking shape of 3x3 matrix
```

```
(3, 3)
```

```
a1.size
# prints Number of elements in ndarray
```

```
9
```

```
a1.nbytes
# prints number of bytes comsumed by ndarray
```

```
72
```

```
np.array([[True, False], [True, False]])
```

```
array([[ True, False],
       [ True, False]])
```

```
x = np.array([1, 2, 3], dtype='int32')
# creates nd array with explicitly specified data type
x
```

```
array([1, 2, 3], dtype=int32)
```

```
x.dtype
# checks data type of each element of ndarray
```

```
dtype('int32')
```

random number: rand(), randint()

```
np.random.rand(4, 2)
# generates 4x2 ndarray
```

```
array([[0.26123897, 0.00220105],
       [0.88022955, 0.46270049],
       [0.93782603, 0.64366779],
       [0.82134896, 0.44393012]])
```

```
np.random.rand(3)
# equidistant, follows normal distribution
```

```
array([0.10721946, 0.8302901 , 0.71665792])
```

```
np.random.randint(2, 100)
# specify (min, max) and generates a random integers
```

```
35
```

```
np.random.rand(3, 3)
# generate randomn
```

```
array([[0.38960341, 0.09046973, 0.99413171],
       [0.175529  , 0.54600008, 0.64218308],
       [0.7552817 , 0.88648012, 0.08937039]])
```

```
r1 = np.random.randint(1, 10, 8)
# generating ndarray of 8 random values between 1 to 10
r1
```

```
array([3, 1, 4, 8, 1, 4, 1, 9])
```

Aggregate functions

```
r1.max()
# max value from ndarray
```

```
9
```

```
r1.min()
# min value from ndarray
```

```
1
```

```
r1.argmax()
# index of max value from ndarray
```

```
7
```

```
r1.argmin()
# index of min value from ndarray
```

```
1
```

Generating dafault matrices

```
np.zeros(5)
# generates ndarray of zeros
```

```
array([0., 0., 0., 0., 0.])
```

```
np.ones(5)
# generates ndarray of ones
```

```
array([1., 1., 1., 1., 1.])
```

```
np.identity(5)
# identity matrix
```

```
array([[1., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0.],
       [0., 0., 1., 0., 0.],
       [0., 0., 0., 1., 0.],
       [0., 0., 0., 0., 1.]])
```

```
np.random.rand(3, 3) + 5
# creating an ndarray with 3x3 matrix and performing broadcasting operation
```

```
array([[5.09359657, 5.05570999, 5.761124  ],
       [5.35049421, 5.41443224, 5.24340962],
       [5.50363013, 5.93834033, 5.77712611]])
```

Broadcasting operation on ndarray

```
a = np.array([1, 2, 3])
# creating ndarray
a
```

```
array([1, 2, 3])
```

```
a+2
# performing broadcasting operation ndarray of adding 2 to each element
```

```
array([3, 4, 5])
```

```
a-2
# performing broadcasting operation ndarray of subtracting 2 from each element
```

```
array([-1,  0,  1])
```

```
a*2
# performing broadcasting operation ndarray of multiplying 2 to each element
```

```
array([2, 4, 6])
```

```
a/2
# performing broadcasting operation ndarray of dividing 2 from each element
```

```
array([0.5, 1. , 1.5])
```

```
a+a
# performing broadcasting operation of adding two ndarrays
```

```
array([2, 4, 6])
```

```
a ** 3
# performing broadcasting operation of cube of elements of ndarray
```

```
array([ 1,  8, 27])
```

```
a**0.5
# performing broadcasting operation of square root of elements of ndarray
```

```
array([1.        , 1.41421356, 1.73205081])
```

```
np.sqrt(a)
# performing broadcasting operation of square root of elements of ndarray
```

```
array([1.        , 1.41421356, 1.73205081])
```

```
np.exp(a)
# calculate exponent of each element
```

```
array([ 2.71828183,  7.3890561 , 20.08553692])
```

```
np.log(a)
# calculate log of each element
```

```
array([0.        , 0.69314718, 1.09861229])
```

pandas

import libs

```
import pandas as pd
# importing pandas
```

1. Series - 1-D
2. DataFrame - 2-D

Series

```
pd.Series({1:"Harry", 2:"Puttar"})
# creating a Series
```

```
1     Harry
2    Puttar
dtype: object
```

```
obj = pd.Series([1, "John",3.5, "Hey"])
# creating a Series using list
obj
```

```
0       1
1    John
2     3.5
3     Hey
dtype: object
```

```
obj[1]
# accessing 1st index
```

```
'John'
```

pandas index

```
obj.index = ['zero', 'one', 'two', 'three']
# overwriting the indices with a new list
obj
```

```
zero        1
one      John
two       3.5
three     Hey
dtype: object
```

```
obj1 = pd.Series([1, "John",3.5, "Hey"], index=['a', 'b', 'c', 'd'])
# creating a new Series with custom index
obj1
```

```
a       1
b    John
c     3.5
d     Hey
dtype: object
```

```
obj1['b']
# accessing value at index 'b'
```

```
'John'
```

```
obj1.index
# printing the index of Series obj1
```

```
Index(['a', 'b', 'c', 'd'], dtype='object')
```

```
score = {"Jane":90, "Bill":80, "Elon":85, "Tom": 75, "Tim":95}
# creating a dict
score
```

```
{'Jane': 90, 'Bill': 80, 'Elon': 85, 'Tom': 75, 'Tim': 95}
```

```
np.array(score)
# using a dict to create a ndarray,
# not recommended to create ndarray with dict, instead create Series with dict
```

```
array({'Jane': 90, 'Bill': 80, 'Elon': 85, 'Tom': 75, 'Tim': 95},
      dtype=object)
```

```
names = pd.Series(score)
# creating Series using dict
names
```

```
Jane    90
Bill    80
Elon    85
Tom     75
Tim     95
dtype: int64
```

```
names['Tom']
# fetching value at index 'Tom'
```

```
75
```

ops on Series

```
names < 80
# returns boolean Series for condition check on Series
```

```
Jane    False
Bill    False
Elon    False
Tom      True
Tim     False
dtype: bool
```

```
names[names < 80]
# returns elements which satisfy the condition
```

```
Tom    75
dtype: int64
```

```
names.isnull()
# returns boolean Series for condition check on Series
```

```
Jane    False
Bill    False
Elon    False
Tom     False
Tim     False
dtype: bool
```

```
names[names.isnull()]
# returns elements which satisfy the condition
```

```
Series([], dtype: int64)
```

Series: working on nyc_weather data set

import libs

```
import pandas as pd
```

import dataset

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

import os
os.chdir(r'C:\Users\surya\Downloads\PG-DBDA-Mar23\Datasets')
os.getcwd()
```

```
'C:\\Users\\surya\\Downloads\\PG-DBDA-Mar23\\Datasets'
```

```
df = pd.read_csv('nyc_weather.csv')
# reading csv into dataframe
df.head()
```

	EST	Temperature	DewPoint	Humidity	Sea Level PressureIn	VisibilityMiles	WindSpeedMPH	Pr
0	1/1/2016	38	23	52	30.03	10	8.0	

```
df.shape
# df.shape
```

```
(31, 11)
```

aggregation op

```
df['Temperature'].max()
# printing max value from temperature column
```

```
50
```

```
# calculate EST when Event=Rain
df['EST'][df['Events']=='Rain']
```

```
8      1/9/2016
9     1/10/2016
15    1/16/2016
26    1/27/2016
Name: EST, dtype: object
```

```
df['WindSpeedMPH'].mean()
# printing mean value of WindSpeedMPH column
```

```
6.892857142857143
```

interview questions

1. what is meant by learning in context of Machine Learning?
2. What is Machine learning? Give Real-Life Examples
3. List down the types of Machine Learning with an example
4. What are the differences between supervised & unsupervised learning?
5. What is meant by supervised classification?
6. Explain unsupervised learning with an example
7. What do you mean by reinforcement learning? Explain with an example
8. What are the challanges in Machine Learning ?