**TAKENMING GLOBAL INTERNSHIP ON DATA ANALYTICS**

**ASSIGNMENT 1: NUMPY AND PANDAS**

**By Ankit Taklikar**

**Numpy**

Numpy is one of the many libraries used in python, it stands for numerical python. We can assign a name to numpyto avoid typing the whole name of the library function everytime it is used.

Code:

Import numpy as np

**Working with arrays**

We can use various functions with arrays such as shape, dtype, zeros, ones, empty, eye, arrange.

1. **Creating an array**

import numpy as np

list1 = [1,2,3,4]

list2 = [4,3,2,1]

myarray = np.array([list1, list2])

printmyarray

Or

import numpy as np

print np.array([[1,2,3,4],[4,3,2,1]])

Other functions

* Zeros – creates an 1d array with all elements as 0s

Code:

np.zeros(size)

* Shape – tells about the size of array (rows x col)

Code:

print myarray.shape

* Ones – used to create a 2d matrix of all 1s

Code:

np.ones([x,y])

* Empty – similar to zeroes

Code:

np.empty(x)

* Eye – used to create an identity matrix

Code:

np.eye([x, y])

* arange–creates an array after we specify starting element, ending element, and the difference between consecutive elements.

Code:

np.arange(x,y,z)

1. **Scalar operations on array**

Scalar operations can be performed on arrays such as multiplication, exponential multiplication, subtraction and reciprocal.

Code:

import numpy as np

array1=np.array([[1,2,3,4],[5,6,7,8]])

#multiplication

array2=array1\*array1

#exponential multiplication

array3=array1\*\*3

#subtraction

array4=array1-array1

#reciprocal

array6=1/array1

1. **Indexing numpy arrays**

We can use indexing to print a set of elements from an array. For example, printing the 2nd element to the 7th element. And in a similar way we can alse modify the elements of an array.

Code:

import numpy as np

#12 element array starting from 0 to 11

arr=np.arange(0,12)

print arr

#print first 5 elements i.e. 0th element to 4th element

printarr[0:5]

#to replace the first 6 elements by 22

arr[0:6]=22

#to replace all the elements of an array by another number

arr[:]=56

Indexing in case of 2d arrays

Displaying a single row from a 2d array. Displaying the elements from a 2d array using the row and column indexes.

Code:

Import numpy as np

arr2d = np.array([[1,2,3],[4,5,6],[7,8,9]])

#to display the first row of the array

print arr2d[0]

#to display the element of 1st row 3rd column

print arr2d[0][2]

#slices of 2d array

#below instruction will display the first 2 elements of the first 2 rows

slice1 = arr2d[0:2,0:2]

#replace the elements of 0 and 1 row index and 0 and 1 column index with 15

arr2d[0:2,0:2]=15

#we can also use loops to index

arr\_len = arr2d.shape[0]

for I in range(arr\_len):

arr2d[i]=i; #0th row elements will be 0, 1st row elements 1 and 2nd row elements 2

#to display 1st row followed by 0th row

print arr2d[[1,0]]

1. **Premium array operations**

Operations such as sqrt,exponential, addition and maximum

Code:

#array as numbers 0 to 14

A=np.arange(15)

#square root of all elements of array A

B=np.sqrt(A)

#exponent of elements of array

C=np.exp(A)

#addition of arrays

D=np.add(A,B)

#maximum of corresponding elements among both arrays

E=np.maximum(A,B)

1. **Load and Save from external memory**

Load and Save single array:

Code:

import numpy as np

arr=np.arr(10)

#new file created and saved as saved\_array.npy

np.save(‘saved\_array’,arr)

new\_array=np.load(‘saved\_array.npy’)

Load and save multiple arrays

Code:

Np.savez(‘saved\_archive.npz’,x=array\_1,y=array\_2)

Load\_archive=np.load(‘saved\_archive.npz’)

Save array as a textfile:

We cannot physically access the arrays in npy format but can store the arrays in txt format

Code:

#delimiter is used to separate the elements of an array

Np.savetxt(‘xyz.txt’,array\_1,delimiter=’,’)

Load\_txt\_file=np.load.txt(‘xyz.txt’,delimiter=’,’)

1. **Statistical processing and array graphical sketches**

Matplotlib is used for plotting graphs.

Code:

Import numpy as np

import matplotlib as plt

#meshgrid function of numpy library

dx,dy = np.meshgrid(axes\_value,axes\_value)

#create a new array named func which equals the sum of 2\*dx and 3\*dy

func= 2\*dx+2\*dy

print func

1. **Conditional Clauses and Boolean Operations on Numpy**

Code:

X=np.array([100,200,300,400])

Y=np.array([15,15,35,45])

Condition=np.array([true,true,false,false])

Z1=np.where(Condition,X,Y)

#if condition is true(x>0) replace by 0 else replace by 1

Z2= np.where(x>0,0,1)

We can also perform operations such as sum of elements of array, mean, std deviation and variance. Also we looked at the various operators that can be used such as AND, OR .

Sort() function is used to sort the elements of an array.

Eg:

A = np.array([3,9,4,8,2,6])

A.sort() #will sort all the elements

Unique function returns the unique values from the array. In1d function returns true if the elements specified are present in the array else returns false.

**PANDAS**

Pandas provide powerful data structures for data analysis, time series and statistics.

**1.Series Functionality of panda**

Code:

data\_array = np.array([‘a’,’b’,’c’])

s = Series(data\_array)

print s #new series is seen as 1 2 3

#custom index

S = series(data array , index = [45,46,47])

revenue = series([20,45,36,80],index =[‘ola’, ‘uber’, ‘swift’, ‘zoom’] )

#convert series to dictionary

revenue\_dict=revenue.to\_dict()

print revenue\_dict

#shows where the null values are present

pd.isnull(revenue) #in this case there will be no null but if an extra data is added such as lyft and it does not have a revenue attribute then we will get true for lyft

Not null can also be used instead of isnull and it is exactly opposite of isnull.

**2. Dataframes**

Dataframes are in the form of a 2d matrix which contain indexes and column names.

Code:

Import pandas as pd

From pandas import Series, Dataframe

revenue\_df = pd.read\_clipboard()

#display the contents copied from some table

print revenue df

revenue\_df2 = Dataframe(revenue\_df, columns = [‘Rank’,’Name, ‘Sales’])

print revenue df2 #no sales column therefore it will have NaN values

print revenue df.ix[0] #display all the attributes of first row

#We can use head function to display a set of rows from the first row or we can use tail function to do the same for the tuples from the bottom.

* Assigning values to df using numpy

Code:

array1=np.array([1,2,3,4,5])

#1 to 6 will act as profit values of the companies

revenue\_df2[‘profit’]=array1

print revenue\_df2

* Assigning values to df using series

Code:

profit=series([900,100],index=[3,5])

revenue\_df2[‘profit’]=profits

print revenue\_df2

**3. Indexing and re-indexing of series in Pandas**

Code:

Series1=series([10,20,30,40],index=[‘a’,’b’,’c’,’d’])

Index1 = Series1.index

Print index1

Print index1[2] #3rd element of index 1

Print index2[2: ] #from index 2 till last value

Series2=series1.reindex([e,f,g,h,i]) #another method that can be used is fill value

We cannot simply add an index in series. If we want to add another index e in series1 then it cannot be added.

**4. Dropping entries**

Code:

cars=series([‘bmw’,’audi’,’mer’),index = [‘a’,’b’,’c’])

print cars

cars = cars.drop(‘a’)

Using dataframes

Code:

Car\_df=cars\_df.drop(‘BMW’,axis=0) #the bmw row will be deleted

Car\_df=cars\_df.drop(‘pro’,axis=1)

**5. Handling null data**

Code:

#validate null data

Series1.isnull()

#drop not available values

Series1.dropna()

#after creating a dataframe we can perform the following

Df1.dropna(axis=1)

#fill na values with 0

Df2.fillna(0)

**6. Selecting and modifying data in pandas**

Code:

#conditional indexes in series

Print series[2]

Print series[series>50]

#ix function access in dataframe

Print Df.ix[‘bike’] #bike values are printed

print df.ix[1] #instead of specifying bike we specify the row of bike which is 1 and all the attributes of bike are obtained

1. **Coordinate and regulate data**

Addition of series and dataframes. Assigning a row of a dataframe to a series and then subtracting it with the original dataframe.

1. **Ranking and sorting of arrays**

Code:

#sorting according to index

Ser1.sort\_index()

#sorting according to values

Ser1.sort\_values()

#assigning ranks

Ser1.rank()

1. **Statistics and graph sketches with pandas**

Dataframes

Sum, min, max ,idxmax, cumulative sum ( cumsum). All of these are dot functions used in dataframes ( eg: df.sum())

Series

Code:

Ser1.Unique ()

Ser1.valuecounts