# ASSIGNMENT 1

# Image result for numpy and pandas

# On Numpy and pandas

**BY: KSHITIJ SHARMA**

**EMAIL ID: KSHITIJKS301@GMAIL.COM**

# Overview

This is simply documentation of what was learnt from the video lectures prescribes by TakenMind as a part of internship programme. This is part of a series of assignment that has to be submitted. This documentation mainly focuses on numpy and pandas.

# NUMPY

First we need to know if numpy (scientific computing package) is already installed in our iDE. If not then we simply install it going to

files -> settings ->project interpreter -> add numpy and then click install.

Importing numpy

To work with numpy and use it contents we need to import it. To create an array using numpy we need to follow the syntax as specified below

Variable\_name = numpy.array(list)

Where list is the contents of the one dimensional array.

Using lists to create arrays of multi\_dimension

We can use lists to create a multi\_dimension array in numpy

variable\_name = numpy.array(list\_1,list\_2,...)

import numpy

print numpy.array([‘lemon’,’orange’,’grapefruit’,’mandarins’])

# [‘lemon’,’orange’,’grapefruit’,’mandarins’]

Standard array functions

1. shape : it gives the number and columns as an ordered pair (number of rows, number of columns) .

Syntax: array\_name.shape

1. Dtype : it gives the datatype of the contents of the array.

Syntax: array\_name.dtype

1. Zeros: it creates an array with zeros depending on the argument.

Syntax: variable\_name = numpy.zeros(number of zeros)

1. ones : it creates an 2 d array with ones depending upon the arguments.

Syntax: variable\_name = numpy.ones(rows,columns)

1. Eye: it creates an identity matrix on order n\*n where n is the argument.

Syntax: new\_array = numpy.eye(dimension)

1. Empty: it creates an empty array with the elements being approximately zero in value.

Syntax: New\_array = numpy.empty(5)

1. Arrange: it consists of three arguments : start, end, gap it is used to create an array. Syntax : new\_array = numpy.arange(start, end, gap)

It is now clear that numpy can be used to create arrays which are farmor superior than the array data types in other programming languages. Moreover arrays can be handled like integers and performed scalar operations upon like multiplication, addition, reciprocal

Array\_1 = array\_2\*\*3 # performs the cube of the elements of the array.

Several mathematical operations can also be performed on numpy arrays like exponentiation,square roots, maximum element,etc.

A = numpy.arange(4,49,5)

Array\_1 = numpy.exp(A)

Matplotlib

It is a library that is used for plotting graphs in two dimensions . this is achieved by using a module called pyplot which plots various graphs.

Pyplot works by choosing grid points that follow the specific function. For the grid points, we use numpy.

We start by

Import mathplotlib pyplot as p

grid\_values = numpy.arange(-250,250,5)

x,y=numpy.meshgrid()

Plot\_function = x\*x-y\*y

p.imshow(plot\_function)

p.saveimg(‘first\_plotting.jpg’)

The above code generates a grid space of 250\*250 with different gradients across the grid plotting the values of the graphs.

Boolean and logical operations can also be performed in numpy arrays. Operations like sorting, any(if anything is true), all (if everything is true), unique(displays only the unique elements of the array) etc, can be performed on arrays.

# PANDAS

# DataFrames and Series are two modules that we get from panda’s package. DataFrames are used to create two dimensional table.

Series

A series is just an array with its indices labelled and these indices can be changed manually.

Variable\_name = Series( list )

As mentioned earlier we can also change the index of the array as we wish: Series\_name=Series(list,index=’ index\_name’,.......)

The advantage of using series is that it is interconvertible, meaning we can convert series into dictionaries, arrays, dataframes etc.

name\_of\_dictionary = name\_of\_Series.to\_dict()

Sorting of arrays

Sorting of arrays can be of two types

1. By index: series\_name.sort\_index()
2. By values: series\_name.sort\_values()

Sorting is based on the rank of every index. Every element has a rank value associated with it. When the sorting function is called, the function analysis the rank value of the element and arranges in the ascending value of it.

DataFrames

In simple terms dataframes is analogous to sql tables in python. It is a 2 d array with columns of different data types.

There are many methods to create dataframes like using read\_clipboard function(). This function just assigned the content that is copied to clipboard to the variable on the left hand side of the expression.

Functions like head and tail are used to access the first and the last rows of the data table. In order to access specific rows in the data table, we use ix[index\_to\_be\_accessed] function which assigns the row indexed by the argument.

In order to change the index of dataframe, we cannot just change the index using the equal to sign. Instead we need to use the reindexing function.

Syntax: series\_name=series([index\_1\_list],index = [change\_index\_list])

Creating new dataframe using randr

**Syntax:**

Dataframe\_name=DataFrame(rand(total\_number\_of\_ekements\_in\_dataframe).reshape(number\_of\_rows,number\_of\_columns),index = [list\_of\_index],columns = [list\_of\_column\_index])

Dropping an entry from DataFrame

Name\_of\_dataframe = same\_dataframe(‘name of the element’,axis = desired\_row\_number)

Adding two DataFrames

Dataframe\_1.add(DataFrame\_2,filLvalue = 0)

The above statement adds two dataframes and for all the values of dataframe\_1 that does not have any value as dataframe\_2 take it as zero.

Statistics through pandas

There are various functions like sum(),max(),idxmax(),legend(),plot(),describe(),etc. Each of these functions has got its own role to play.

Syntax: dataframe\_name.sum(axis = row\_number)

max() and min() performs similar tasks on the dataframe except that their target element is different.

Syntax: dataframe\_name.min()

cumsum() function calculate the cumulative sum of the elements in a column

Syntax: dataframe\_name.cumsum()

describe() gives the count, mean,std,min,max,etc of the array

Syntax: dataframe\_name.describe()

You can also plot graphs using the matplotlib library and using its pyplot module. Plotting of dataframe is possible through the following command:

Syntax:

dataframe\_name =Dataframe(randn(number\_of\_elements).reshape(rows,columns),index = [list\_1], column = [list\_2])

pyplot.[plot(dataframe\_name).