Machine learning

Python method summary

functions concerning: math, statistics, random
modules: basic python, math, statistics, random, numpy, pandas

חזרה על פקודות python רלוונטיות – basic python

Mathematical function:

x, y, num, base – scalars

arr - iterable or array like element

x ** y

pow(x, y)

abs(x) - Return the absolute value of x

sum(arr) - Return a sum of values in the iterable

min (num1, num2), min (arr)

max (num1, num2), max (arr)

Other functions:

dict(), set(), list() – empty container

len() – number of elements

bool(), int(), float(), str() – casting

type(variable)

all(iterable) - Return True if all elements of the iterable are true.

any(iterable) - Return True if any element of the iterable is true.

sorted(), reversed(), reversed()

חזרה על פקודות python רלוונטיות – The **statistics** module

arr - iterable or array like element

population variance & std:

statistics.pvariance(arr) - variance of arr

statistics.pstdev(arr) - standard deviation of arr

sample variance & std:

statistics.variance() – Sample variance of arr

statistics.stdev(arr) - Sample standard deviation of arr

statistics.median(arr)

statistics.mean(arr)

statistics.mode(arr) - mode (most common value)

For more info on the *statistics* module click: *statistics module* documentation

חזרה על פקודות python רלוונטיות –
The **math** module

x, y, num, base – scalars

arr - iterable or array like element

base_n - the log base, which we want to calculate, e.g., base_n=10 --> log10 and so on.

log_base_n(num) = log(num)/log(base_n)

math.fabs(x) - Return the absolute value of x

math.fsum(arr) - Return an accurate floating-point sum of values in the iterable

math.pi - The mathematical constant $\pi = 3.141592$

math.e - The mathematical constant $e = 2.718281$

math.pow(x, y) = x ^ y

math.sqrt(x) - Return the square root of x

math.exp(x) = e ^ x

math.log(num, base) = ln(num)/ln(base_n)=log_base_n(num)

math.log2(num)

For more info on the *math* module click:

math module documentation

חזרה על פקודות python רלוונטיות –
The **numpy** module (referred as **np**) – math functions

x, y, num, base – scalars
arr, a1, a2 - iterable or array like element

np.abs(arr),

np.log(arr) – ln, np.log2(arr), np.log10(arr)

np.power(arr, y)   x^y (for x in a1)

np.power(a1, a2) x^y (for x in a1, y in a2)
np.sqrt(arr)
np.exp(arr) = e ^ x (for every x in the arr)

np.sum(arr) == nd_arr.sum()
np.prod(arr)

np.dot

np.subtract(a1,a2), np.add (a1,a2),
np.multiply(a1,a2), np.divide (a1,a2)

np.amin(nd_arr) == nd_arr.min()

np.amin(nd_arr, axis=0) == nd_arr.min(axis=0)

np.amin(nd_arr, axis=1) == nd_arr.min(axis=1)

np.amax(nd_arr), nd_arr.max() – same options

np.argmax, np.argmin, np.argsort , etc.

np.pi, np.e, np.nan,

For more info *numpy math* functions click:

numpy math documentation

חזרה על פקודות python רלוונטיות –
The **numpy** module - statistics functions

numpy statistic functions

np.mean(arr) – average

population covariance, variance & std:

np.var(arr), np.std(arr), np.cov(arr1,arr2)

sample covariance, variance & std:

np.var(arr, ddof=1), np.std(arr , ddof=1), np.cov(arr1,arr2 , ddof=1)

For more info *numpy statistics* functions click:

numpy statistics documentation

Pandas DataFrame / Series

df – DataFrame object, sr – Series object

df.mean, df.median

sr. mean, sr. median

sample covariance, variance & std:

df.std, df.var, df.cov

sr.std, df.var, sr.cov

population covariance, variance & std:

df.std(ddof=0), df.var(ddof=0), df.cov(ddof=0)

sr.std(ddof=0), df.var(ddof=0), sr.cov(ddof=0)

חזרה על פקודות python רלוונטיות – random – random vs. np.random module vs. pandas

np.random module

np.random.random- [0, 1) - in a given size

np.random.rand - [0, 1) - in a given shape

np.random.randint (or np.random.random_integers)

np.random.shuffle(arr) - mix arr in place

np.random.permutation(arr) – permutation (as copy)

np.random.choice(arr|size,replace=False)

np.random.seed – reproducible seed

random module

random.randint(a, b) – single number

random.random() – single number

random.choice(arr) – randomly choose element

random. shuffle(arr) - mix arr in place

random.sample(population, k) - without replacement.

random.seed – reproducible seed

Pandas DataFrame / Series (example for df same for series)

df.sample(n|frac, replace=False) – without replacement