**#######################################################**

**#Copyright (c) DeepSphere.AI 2021**

**# All rights reserved**

**# We are sharing this partial code for learning and research, and the idea behind us sharing the source code is to stimulate ideas #and thoughts for the learners to develop their MLOps.**

**# Author: # DeepSphere.AI |** [**deepsphere.ai**](http://deepsphere.ai/) **|** [**dsschoolofai.com**](http://dsschoolofai.com/) **| info@deepsphere.ai**

**# Release: Initial release**

**#######################################################**

**Number-theoretic and representation functions**

As the intrinsic role of mathematics in data science, number-theoretic functions play a fundamental and vital role. The first step to becoming a top-notch data scientist is understanding the data and then applying the problem's solution. Hence Number-theoretic functions and representation functions help us know the numeric data well enough to use any appropriate ML model on the data.

Few examples:

**math.ceil(X)** - returns the next largest integer value around the passed value x.

**math.copysign(X, y)** - returns a float number with the sign of y

**math.fabs(X) -** returns the absolute value of X.

**math.factorial(X) -** returns the factorial of X as an integer. When X is not integral or negative, the function throws ValueError. It accepts only positive integers.

**math.floor(X)** - returns the largest integer less than or equal to X. If X is not a float, then the function should return an Integral value.

**math.fmod(X, y) -** returns fmod(X, y), as defined by the platform C library. fmod(X, y) is precisely equal to X - n\*y for some integer where the result has the sign as X and magnitude less than absolute(y). Function fmod() is mainly preferred when working with floats, while Python's X % y is preferred when working with integers.

**math.fsum(iterate\_val) -** returns an accurate sum of values. The return value is a float without loss of precision.

**math.gcd([integers]) -** returns the greatest common divisor of the specified integer arguments.

**math.isclose(a, b) -** returns True if the values a and b are close to each other and False otherwise.

**math.isfinite(X) -** returns True if X is neither infinity nor a NaN, and False otherwise. (where 0.0 is considered finite.)

**math.isinf(X) -** returns True if X is positive infinity or negative infinity, and False if not.

**math.isnan(X) -** returns True if X is a NaN (not a number) else False .

**math.isqrt(n) -** returns the square root value of the passed non-negative integer. The return value is an integer.

**math.lcm([integers]) -** returns the least common multiple of the specified list of integers. If all the passed values are nonzero, the function returns the smallest positive integer, which is a multiple of all arguments. If any one of the arguments is zero, then it returns 0. lcm() without arguments returns 1.

**math.modf(X) -** returns a tuple that contains the fractional part and integer part of the passed value X. The return values are in float.

**math.nextafter(X, y) -** returns the next floating-point value after X towards y. When X is equal to y, then the function returns y.

**math.perm(n, k=None) -** returns the number of ways to choose k items with the order from n items without repetition.

**math.prod(iterate\_vals, \*, start=1) -** calculates the product of all the elements in the input iterate\_vals. The product's default start value is 1.

**math.trunc(X) -** returns the real value X truncated to an Integral (usually an integer).