Welcome to Linear Algebra Review

by R.J. Surio ©2021

In this module, we are going to review some of the useful equations needed in data science to refresh our skills in Python programming. In this notebook we are going to cover:

- Modulus of a Vector
- The Sigmoid

Practice 1: Modulus of a Vector

The modulus of a vector or the magnitude of a vector can be determined using the Pythagorean theorem. Given the vector A and its scalars denoted as a_n where n is the index of the scalar. So if we have:

$$A = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

We can compute the magnitude as:

$$||A|| = \sqrt{a_1^2 + a_2^2} = \sqrt{1^2 + 2^2} = \sqrt{5}$$

So if we have a matrix with more parameters such as:

$$B = \begin{bmatrix} 2 \\ 5 \\ -1 \\ 0 \end{bmatrix}$$

We can generalize the Pythagorean theorem to compute for the magnitude as:

$$||B|| = \sqrt{b_1^2 + b_2^2 + b_3^2 + ... + b_n^2} = \sqrt{\sum_{n=1}^{N} b_n^2}$$

And this equation is now called a Euclidian distance or the Euclidean Norm.

Code Time:

```
import numpy as np
import math
```

```
def modulus(vec):
    raisesum = 0
    for element in range(len(vec)):  # Accesing the values
        raisesum += vec[element] **2 # raising each number to 2
        sqr = math.sqrt(raisesum)    # Squaring
    return(sqr)

vect1 = [2, 5, 2, 5, 14] #These are the values of vectors
print("Vector's magnitude for vect1 = ", modulus(vect1))
```

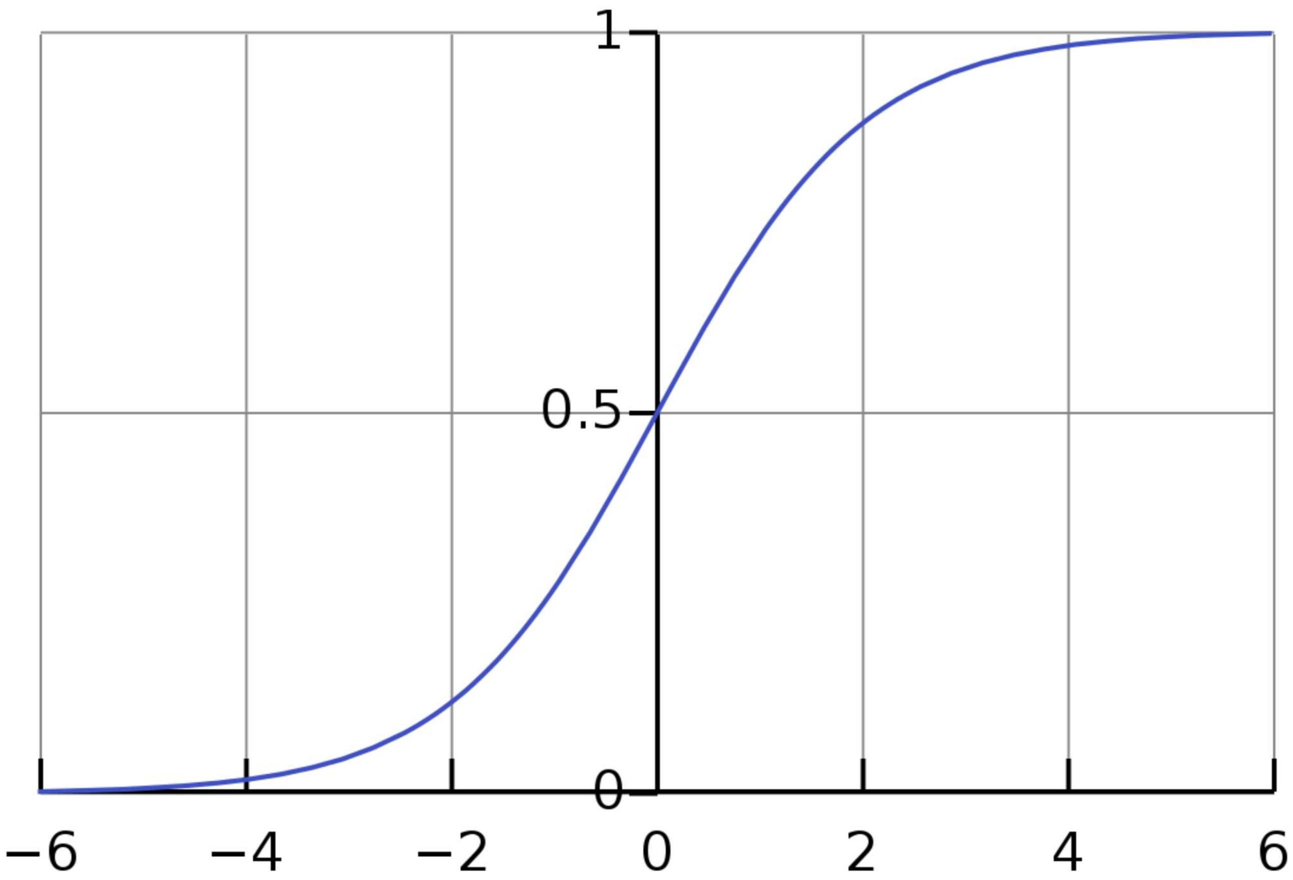
```
Vector's magnitude for vect1 =  15.937377450509228
```

About the code

The function above calculated the magnitude of the vector with 5 elements or values. The variable ‘raisesum’ was set to 0 so that the assignment of this variable is possible at the same time that it was used as referenced. A loop statement was created to access every element inside the vectors created and inside this loop statement are the operations that will compute the modulus of the 5 different vectors. The operations include the summation of the squared elements from the vectors. The sum of these elements was assigned to the ‘raisesum’ variable and the square root of this was assigned to ‘sqr’ which was returned afterward and printed.

Practice 2: The Sigmoid

The sigmoid function is one of the popular Activation Functions which we will discuss later on. The sigmoid is a bounded, differentiable, real function in which its range would be any value from 0 to 1. It is widely used in binary classifications.



If we were to check the equation characterizing this curve in textbooks or journals it would be:

$$sig(x) = \frac{1}{1 + e^{-x}}$$

or

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

So let's try to translate this in NumPy. You might want to use `numpy.exp` for this function.
If you want to read more about the sigmoid function click here.

Code Time:

Standard Function

```
def Sigmoid(x):
    return 1/(1 + np.exp(-x)) # Transformed equation
Sigmoid(3)                    # Using the function
```

```
0.9525741268224334
```

Lambda Function

```
x =  3
s = lambda x: 1/(1 + np.exp(-x))
print(s(x))
```

```
     0.9525741268224334
```

About the code

The equation of sigmoid was transformed into standard and lambda function. The standard function was used by calling the function with values '3' as its argument that passed into the function as its parameter and returns the result of the equation. Meanwhile, the lambda function is the more optimized version of the standard function since the declaration of a function is not needed but still does the job the same as with the standard function.