**VECTORS**

**Euclidean space:**

The space of n dimensions where the distance between any two points is given by the distance formula sqrt(sum(x^2, y^2, ...etc))

**Dot Product:**

* u and v are of the same dimensions
* u.v = ut v

**Unit vectors:**

* Any vector with a magnitude of 1.
* Coordinate axes vectors, i(1,0,0), j(0,1,0), k(0,0,1) etc
* (1/√3,1/√3,1/√3) is also a unit vector as its magnitude is 1
* Any vector with a magnitude != 1 can yield a unit vector if we take the vector and scalar divide it by its own magnitude.

**Projection of one vector onto another:**

* Projyx =

**Angle between two vectors:**

* Cos*θ =*
* Orthogonal vectors θ*= 90o*
  + Cos*θ =* 0
  + = 0
  + (x.y) = 0

**Why do we care about vectors:**

* Very useful to represent features of our dataset as a collection of numbers in a vector format
* For eg: Phone specs as a vector, Image as a collection of numbers in vector form.

**MATRICES**

**Introduction to matrices:**

* It is a collection of column vectors or a collection of row vectors
* m = No. of rows
* n = No. of columns
* Addition of two matrices:
  + m1 == m2
  + n1 == n2

**Multiplying a vector by a matrix:**

* Row-by-column method: dot product of each row of matrix with the column of the vector
* Number of columns in the matrix should be equal to the number of rows in the vector
* mxn \* nx1
  + Yields mx1 dimension vector

**Multiplying a matrix by another matrix**

* Same Row-by-column method: dot product of each row of m1 with each column of m2
* Number of columns of m1 should be equal to the number of rows of m2
* mxn \* nxo
  + Yields mxo dimension matrix

**Alternative way of multiplying two matrices:**

* Linear combination of the columns of the matrix

**Why do we care about matrices:**

* Used to create a matrix of features
* m = No of training examples
* n = Number of features in each training examples