Q1) **Jaccard Distance – The Jaccard Distance tells us the dissimilarity between the two data sets.**

**The Jaccard distance is calculated as (|**A∪B**|-| A∩B|)/|**A∪B**|**

The higher the dissimilarity the greater is the distance between the two.

Example : If we look at the base and target 6 they are very Similar .example The calculated Jaccard Array in the figure 2 shows the distance between them as 0.29 which is less.

base = "Mummy teaches numbers and baby dances"

target = ["Mummy sings Rockabye and baby awakens", "Mummy works and baby eats apple",

"Mummy sings songs and baby dances", "Mummy teaches alphabets or baby dances",

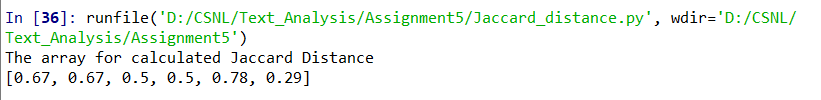
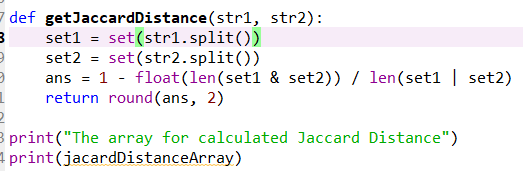
 "Mummy sings lullaby and Mummy sleeps", "Mummy teaches numbers or baby dances"]

Figure 2.

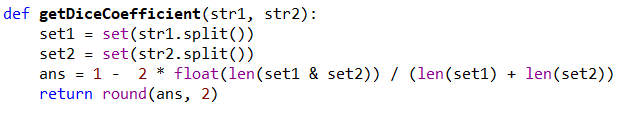
Figure 1.

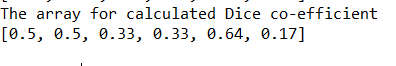
**Triangle Inequality** : The triangle inequality property states that **a+b>=c** i.e. The sum of the two sides is always greater than or equal to the third side.

Now after summing up the 4rd and the 6th value (0.5+0.29) = 0.79, we can conclude that the Triangle Inequality Property holds true for Jaccard Distance because none of the values in the array is greater than the summed up value.

Q1 b) Dice Coefficient : Its calculated **as 2(| A∩B|)/|**A|+|B**|**

**The Dice Co-efficient gives us the similarity between two datasets . But its not regarded as a proper Distance Metric because it doesn’t hold true the triangle inequality property.**

`

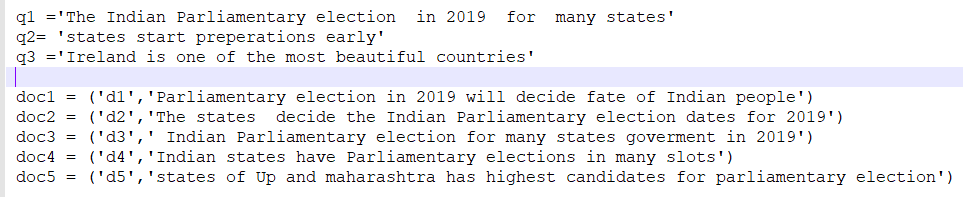
For Example when I sum up the values at position 4th and 6th in the array (0.33 + 0.17) = 0.50 which is less than the 5th value i.e. 0.64 which clearly indicates of **Dice co-efficient not satisfying the triangle in equality property.**

But at times it may satisfy the Triangle inequality property.

For example if we add 1st and 2nd value = 1.0 which is greater than any other value in our array.

Figure : Calculated values for Dice Co-efficient.

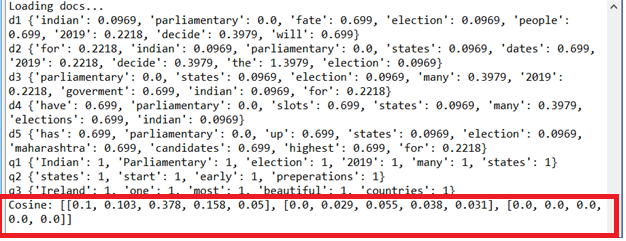
Q2 a ) Data set :

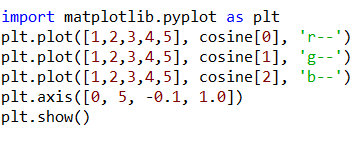
t

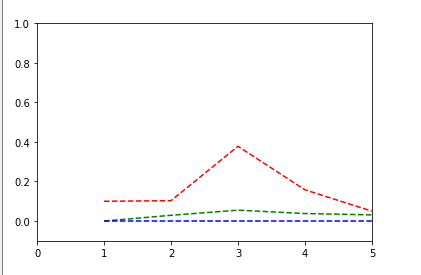
The docs from doc1 to doc5 are the variants of base q1.

The cosine similarity for all docs is calculated with respect to bases q1 , q2 and q3 and are stored in the cosine array highlighted below . Index 0 – q1, 1-q2, 2 -q3.

**The Cosine Similarity states that if the cosine of angle is near one then quantities(here our two documents) are in the same direction** i.e. similar.

So, It can be inferred that the cosine similarity score is highest for q1 and lowest for q3 because all the docs are variants of q1 and not at all related to q3.

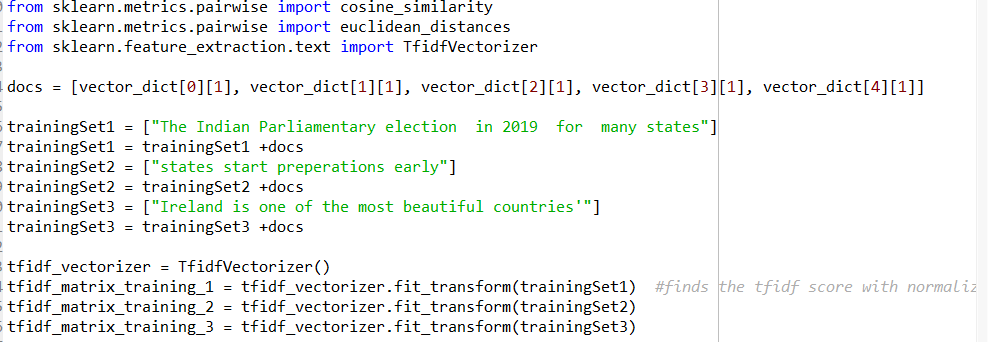
Q2 b) Cosine -plot

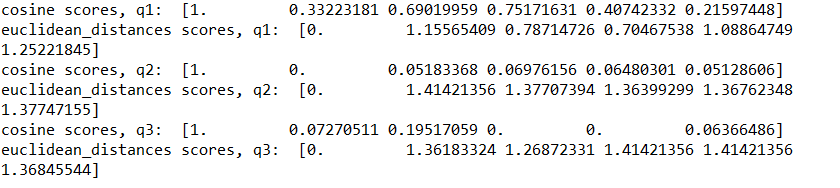
1. Now if we look at the graph below the Red dotted line indicates the plot of q1 vs all the five docs, It shows the highest score amongst all three lines since all the docs are related to q1.
2. The dotted green line shows a plot of q2 vs all five docs ,there are a few spikes for this line due to the inclusion of some similar words, but is very less as compared to q1.
3. The blue dotted line represents a plot of q3 vs all other docs , q3 is totally unrelated to all other docs so the plot shows a straight line which doesn’t move up the zero scale along the y-axis

Q2 c) **Euclidian Distance** - The Euclidian distance is a metric used to find the distance between any two vectors which can be even of varying length.

Higher the distance more dis similarity between the vectors.

**Package used :The sci-kit learn package is used to perform calculations for both cosine and Euclidian distance.**

****



The cosine similarity calculated in both the cases leads to the same conclusion that q1 has the highest similarity score followed by q2 and q3.

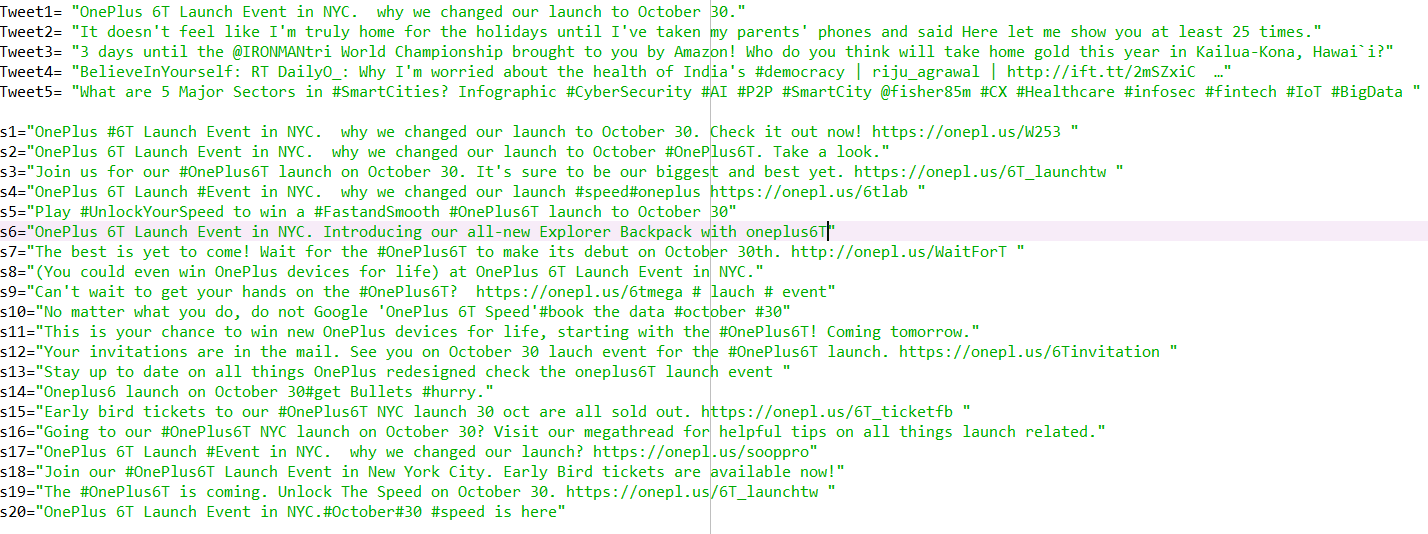
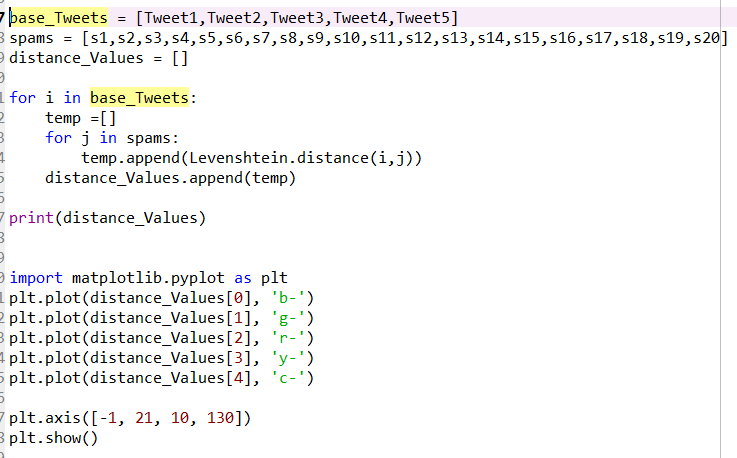
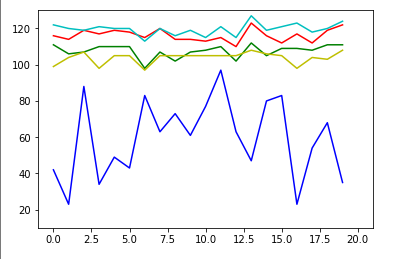
Whereas the Euclidian distance is less for more similar vectors i.e. q1 has small Euclidian distance due to similarity with all 5 docs , than q2 and then longest distance between q3 and all other 5 docs which are totally distinct.

**So we can conclude that the Euclidian distance is inversely proportional to the cosine similarity score.**

**i.e. as the cosine similarity increases the Euclidian distance will tend to decrease.**

References : [stack-overflow :eucledian-distance](https://stackoverflow.com/questions/12118720/python-tf-idf-cosine-to-find-document-similarity),

<https://jakevdp.github.io/PythonDataScienceHandbook/04.01-simple-line-plots.html>

Q3) Data set for 5 normal tweets Tweet 1 to Tweet 5 and Spam tweets from s1 to s10

Code Snippet **Graph for spam vs Normal Tweets**

* The spam tweets were generated using urls, hashtags at various locations and adding some extra textual data.
* We can observe that the blue line shows the comparison between base Tweet1 and all other spam tweets.
* The other colored lines each represent the comparison against the spam tweets.
* I have used the levenshtein package in python to calculate the distance between the two data sets (documents/Tweets)

**Inference :**

The blue line shows **less levenshtein distance because all the 20 tweets are derived from the Tweet1** and all revolve around the topic oneplus6T.

.

The rest all lines show a much higher distance as they are not associated with the spam tweets in all aspects and have a distance from 100 to 120.

**The blue line also shows spikes at times, which indicates that the spam tweets are not obvious, as Levenshtein** **distance is high for some spam tweets this may be because** :

* While calculating levenshtein distance the length of the tweets is also taken into consideration, and in my documents also the tweets are of varying length which can be one of the reasons for the spikes observed in blue line and increase in the distance.
* Also there are spam tweets which have very less mention of the topic oneplus6T and lot involve urls which have no association with the topic which may be the other reason for increase in the levenshtein distance.