**AN ALGORITHM TO EFFICIENTLY FIND DUPLICATES IN A SERIES OF QUESTION DATA.**

From nltk import stop words to remove all the stop words from the series of questions

From nltk import word\_tokenize to convert the questions into token by token.

From lxml import html to Crawler the questions from the websites

Import requests to get the webpage content into our python code

Import tkinter for the Graphical User Interface (GUI) for the application

Create one tkinter named as Top in that window I will create the label and button by clicking on that button it will call the function named as function\_des

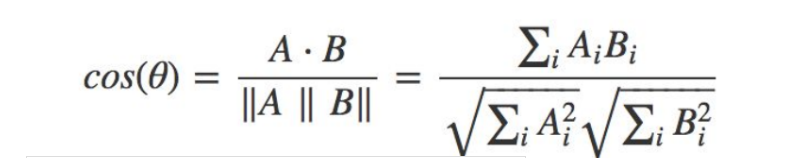
In that function function\_des first and foremost thing destroy the current tkinter window and create a new window names ad ‘root’.

In that window I will create a two label in one label enter the enter the url of the website and in another label enter the class or container name in the website and click Next button that will pass the url and class/container name into res function and destroy that root tkinter window here and the url and the class\_name enter in the label is in the form stringVar want to convert that into string type and get that website content in this code

Next find the similarities of each and every question by passing two questions into function names as fun(str1,str2) in this function by using the cosine similarities I will find the accuracy of the duplication of the questions

Remove the stopwords in the questions and convert question into tokens by using word\_tokenize

**COSINE SIMILARITES FORMULA**



By using this formula calculate the cosine similarities this will give how much the two questions are similar convert this into percentage and split into the buckets finally in the root1 window its show the question bucket by bucket with no duplication of questions .