THE OBJECTIVE IS TO CLASSIFY NEWS ARTICLES INTO TECHNOLOGY RELATED ARTICLES AND NON-TECH ARTICLES

1. CREATE A CORPUS OF NEWS ARTICLES WHICH ARE ALREADY CLASSIFIED INTO TECH AND NON-TECH

DOWNLOAD ALL TECH NEWS ARTICLES FROM NEW YORK TIMES AND WASHINGTON POST AND LABEL THEM AS TECH

DOWNLOAD ALL THE SPORTS ARTICLES FROM BOTH THESE NEWSPAPERS AND LABEL THEM AS NON-TECH

THIS WILL INVOLVE PARSING THE HTML TO REMOVE ALL THE CRUD (DIVS/TAGS)

2. GET A NEW PROBLEM INSTANCE FROM A BLOG - AN ARTICLE THAT NEEDS TO BE CLASSIFIED

3. USE THE K-NEAREST NEIGHBOURS ALGORITHM TO CLASSIFY THE TEST INSTANCE AS TECH OR NON-TECH

REPRESENT EACH ARTICLE AS A VECTOR OF THE 25 MOST IMPORTANT WORDS IN AN ARTICLE

USE NATURAL LANGUAGE PROCESSING FOR THIS : WE HAVE ALREADY DONE IT IN A PREVIOUS EXERCISE

3. USE THE K-NEAREST NEIGHBOURS ALGORITHM TO CLASSIFY THE TEST INSTANCE AS TECH OR NON-TECH

REPRESENT EACH ARTICLE AS A VECTOR OF THE 25 MOST IMPORTANT WORDS IN AN ARTICLE

USE NATURAL LANGUAGE PROCESSING FOR THIS : WE HAVE ALREADY DONE IT IN A PREVIOUS EXERCISE

THE DISTANCE BETWEEN ARTICLES IS CALCULATED USING THE NUMBER OF IMPORTANT WORDS THAT THEY HAVE IN COMMON

FIND THE K-NEAREST NEIGHBOURS AND CARRY OUT A MAJORITY VOTE OF THOSE. ASSIGN THE ARTICLE TO THAT CATEGORY.

SENTIMENT ANALYSIS

GIVEN A PIECE OF TEXT -
A REVIEW OR A TWEET

IS THE SENTIMENT POSITIVE OR NEGATIVE ?

"I LIKE THIS ARTICLE"

POSITIVE SENTIMENTS

"THE FOOD AT THIS RESTAURANT IS OUT OF THIS WORLD"

"THE SERVICE IS VERY BAD"

NEGATIVE SENTIMENTS

"I HATE YOU"

ONE WAY TO PERFORM SENTIMENT ANALYSIS IS TO VIEW IT AS A CLASSIFICATION PROBLEM

CREATE A CORPUS OF TEXT/SENTENCES WHICH ARE LABELLED AS POSITIVE OR NEGATIVE

WHEN A NEW TEXT COMES IN
CLASSIFY IT AS POSITIVE OR NEGATIVE

FIND THE K-NEAREST NEIGHBORS OF AN ARTICLE

```
similarities = {}
 FIND THE SIMILARITY BETWEEN THE TEST INSTANCE AND EACH ARTICLE IN THE TRAINING DATA.
 (SIMILARITY HERE IS THE NUMBER OF WORDS THESE TWO ARTICLES HAVE IN COMMON)
for articleUrl in articleSummaries:
    oneArticleSummary = articleSummaries[articleUrl]['feature-vector']
    similarities[articleUrl] = len(set(testArticleSummary).intersection(set(oneArticleSummary)))
 GET THE 5 NEAREST NEIGHBOURS TO THE TEST INSTANCE
 (THE ARTICLES WITH THE HIGHEST SIMILARITY)
labels = defaultdict(int)
knn = nlargest(5, similarities, key=similarities.get)


RETURN THE LABEL THAT BELONGS TO THE MAJORITY OF THE NEAREST NEIGHBOURS
for oneNeighbor in knn:
    labels[articleSummaries[oneNeighbor]['label']] += 1
nlargest(1,labels,key=labels.get)
```