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 NAIVE BAYES CLASSIFIER 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 NAIVE BAYES CLASSIFIER 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

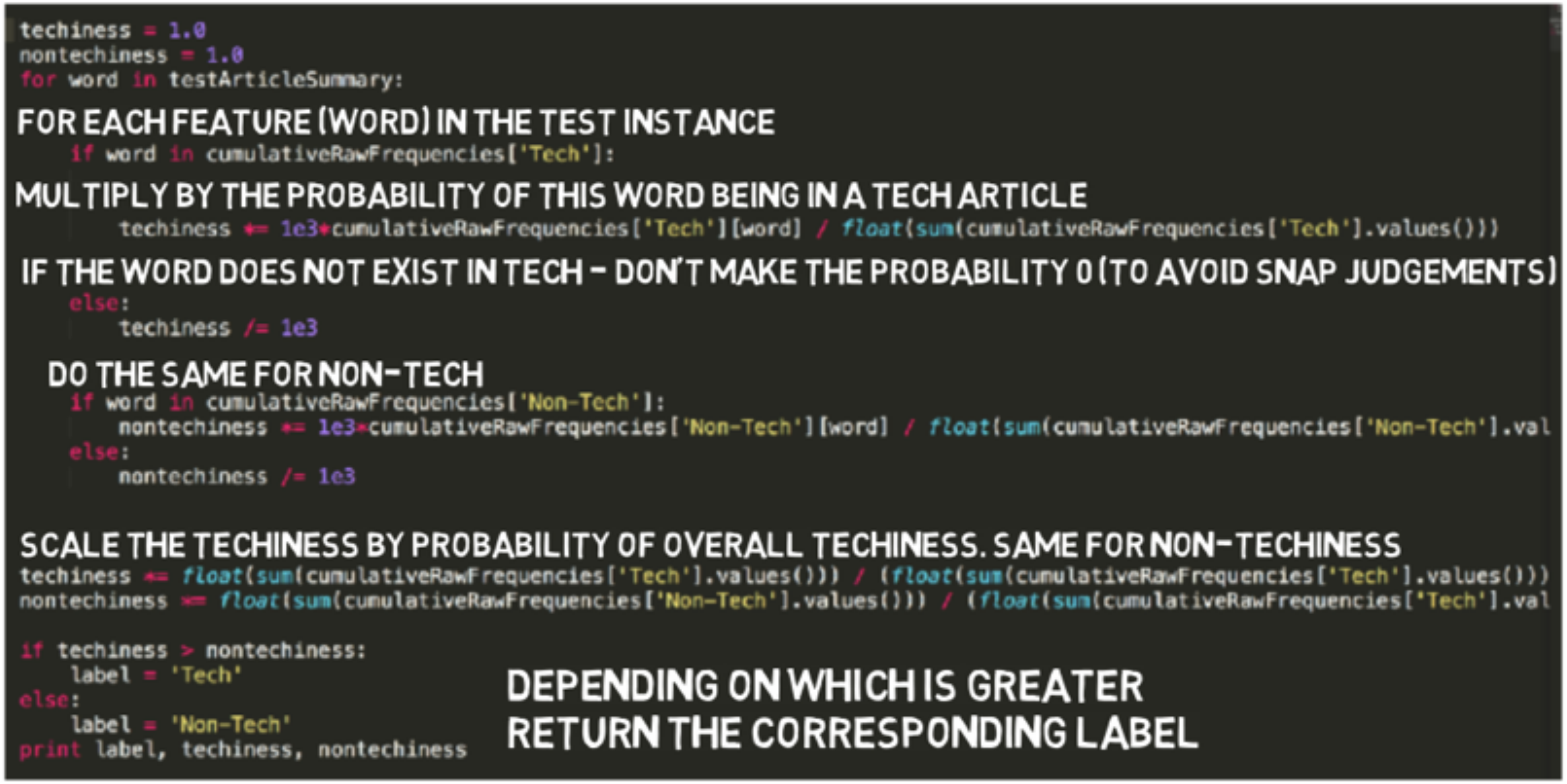
COMPUTE THE TECHINESS AND NON-TECHINESS OF THE ARTICLE (EXACTLY THE WAY WE COMPUTE THE SPAMMINESS AND HAMMINESS IN THE EMAIL EXAMPLE)

IF THE TECHINESS > NON-TECHINESS IT IS A TECH ARTICLE - ELSE IT IS A NON-TECH ARTICLE

COMPUTE THE TECHINESS/NONTECHINESS OF AN ARTICLE

THIS IS HOW YOU CAN COMPUTE THE TECHINESS OF AN ARTICLE (BAYES RULE)
THE NON-TECHINESS WOULD HAVE THE SAME DENOMINATOR - SO JUST COMPUTE THE NUMERATORS

$$Techiness = P(Article\ is\ Tech/Words\ in\ Article) = \frac{P(TEch) \cdot P(\ Word\ 1/Tech) \cdot P(Word2/Tech)........}{P(Words\ in\ Article)}$$



```
techiness = 1.0
nontechiness = 1.0
for word in testArticleSummary:
```

FOR EACH FEATURE (WORD) IN THE TEST INSTANCE

```
    if word in cumulativeRawFrequencies['Tech']:
```

MULTIPLY BY THE PROBABILITY OF THIS WORD BEING IN A TECH ARTICLE

```
        techiness *= 1e3*cumulativeRawFrequencies['Tech'][word] / float(sum(cumulativeRawFrequencies['Tech'].values()))
```

IF THE WORD DOES NOT EXIST IN TECH - DON'T MAKE THE PROBABILITY 0 (TO AVOID SNAP JUDGEMENTS)

```
    else:
        techiness /= 1e3
```

DO THE SAME FOR NON-TECH

```
    if word in cumulativeRawFrequencies['Non-Tech']:
        nontechiness *= 1e3*cumulativeRawFrequencies['Non-Tech'][word] / float(sum(cumulativeRawFrequencies['Non-Tech'].val
    else:
        nontechiness /= 1e3
```

SCALE THE TECHINESS BY PROBABILITY OF OVERALL TECHINESS. SAME FOR NON-TECHINESS

```
techiness *= float(sum(cumulativeRawFrequencies['Tech'].values())) / (float(sum(cumulativeRawFrequencies['Tech'].values()))
nontechiness *= float(sum(cumulativeRawFrequencies['Non-Tech'].values())) / (float(sum(cumulativeRawFrequencies['Tech'].val

if techiness > nontechiness:
    label = 'Tech'
else:
    label = 'Non-Tech'
print label, techiness, nontechiness
```

DEPENDING ON WHICH IS GREATER
RETURN THE CORRESPONDING LABEL