

REPORT

IR Assignment 2

Group No 80

MADIHA TARIQ-MT21125

Q1)JACCARD COEFFICIENT

In jaccard coefficient firstly preprocessing of document has been done as done in assignment 1 than query preprocessing is done again using the similar preprocessing. For jaccard formula has been provided and we had calculated jaccard coefficient using that formula.

a) preprocessing has been done as per assignment 1

b) union and intersection:

```
jaccard_list=[]
for i in tqdm(range(len(d))):
    a=len(set(c)|set(d[i]))           #un
    b=len(set(c)&set(d[i]))           #in
    jaccard=b/a
    jaccard_list.append(jaccard)

100%|██████████| 1133/1133 [00:00<00:00, 115
```

c) sorting to get top 5 document:

```
dic={}
dic_sort={}
for i in range (a):
    dic[b[i]]=jaccard_list[i]
keys=sorted(dic,key=dic.get,rever
for i in keys:
    dic_sort[i]=dic[i]
```

Top 5 Documents are:

```
print(z[0:5])
```

```
['pasta001.sal', 'antimead.bev', 'orgfrost.bev', 'japice.bev', 'montoyo.txt']
```

Ques1(part2)

In question 2 we have to generate tf-idf matrix and score for that. For tf computation 5 weighting scheme was given in

assignment. To do so, we had done preprocessing of document as per assignment 1. vocabulary has been calculated to store term. that are present in document.

Then, we calculated IDF as per the formula given in question-

IDF(word) = $\log(\text{total no. of documents} / \text{document frequency(word)} + 1)$..taking base to be log 10. Term frequency is being calculated using the formula-

$f(t, d) / f(t', d)$. To obtain matrix we multiply tf-idf. The query is being taken from user and then preprocessing has been done. Matrix is generated for query and added them to get tf-idf score. TF-IDF score is calculated and sorting is done to get top five documents. The same procedure is being applied in different weighting scheme just the formula used is different.

a) Binary: tf-weight is (0,1)

PROS: Simple and easy to calculate

CONS: It does not provide enough details on tokens

b) Raw count- formula given is Raw count **$f(t, d)$**

PROS: Good if you want to know details of token, voc

CONS: It is time taking

c) Term-frequency- formula given is **Term frequency $f(t, d) / f(t', d)$**

how often a word appears in a document, divided by how many words there are.

It is calculated as below:

PROS: Easy to compute

Easier to compute similarity between 2 documents

CONS: it takes much longer time to compute

d) Log normalization- formula given is **$-\log(1 + f(t, d))$**

CONS:Time taking

e) Double normalization-The formula given is-- $0.5 + 0.5 * \frac{f(t,d)}{\max(f(t',d))}$

PROS: can tell maximum term present in each token

CONS:Time taking

Q2.Ranked-Information Retrieval and Evaluation .Here we have to evaluate the retrieval system .and work on only quid:4

1. From the given data set create a new dataset that contain only quid:4 so there are only 103 rows that contain quid :4

2. We have made a file that contains max DCG and also need to count how many such files are possible. So here I sort the dataset based on relevance in decreasing order and then count occurrences of each level. There are 4 levels in the data set: 0, 1, 2, and 3. And then take their permutations. These are the total number of files possible with max DCG.

count

```
198934973759383705998260476149053298969368401
705665705882051803127048579926951934824126865
65431050240000000000000000000000000000000000
```

3.

a) nDCG for 5 document

Here i select starting 50 document of newdataset that content qid:4 and calculate DCG then reverse sorted based on relevance and then calculate iDCG and final calculated nDCG for 50 document

Following are the values of DCG,iDCG and nDCG

7.390580969258021

14.067092644997018

0.5253808413557646

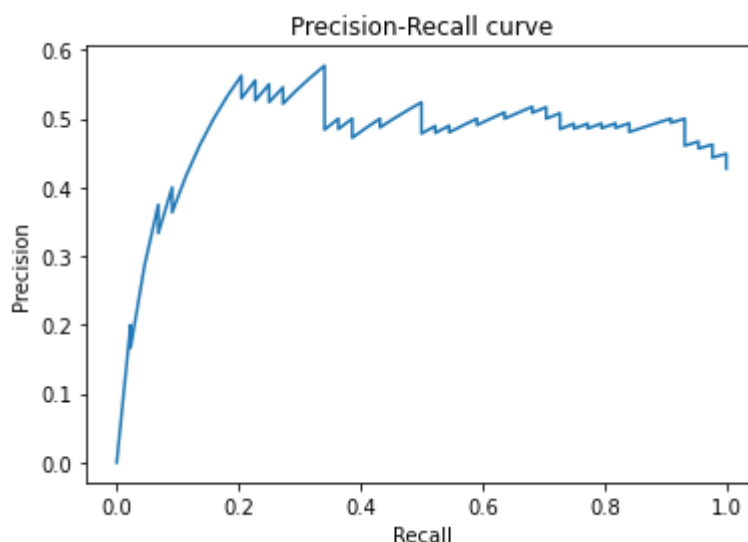
b) nDCG for whole data set so here i follow the same process as for a part and compute DCG,IDCG and nDCG.following are the values of DCG,iDCG and nDCG

12.550247459532576

20.989750804831445

0.5979226516897831

4.In this point we have to plot curve of precision and recall based on the 75 feature value according to relevance and consider all non-zero values as relevance



Q3.In this question we have to implement Naive Bayes algorithm for text classification using TF-IDF,feature selection

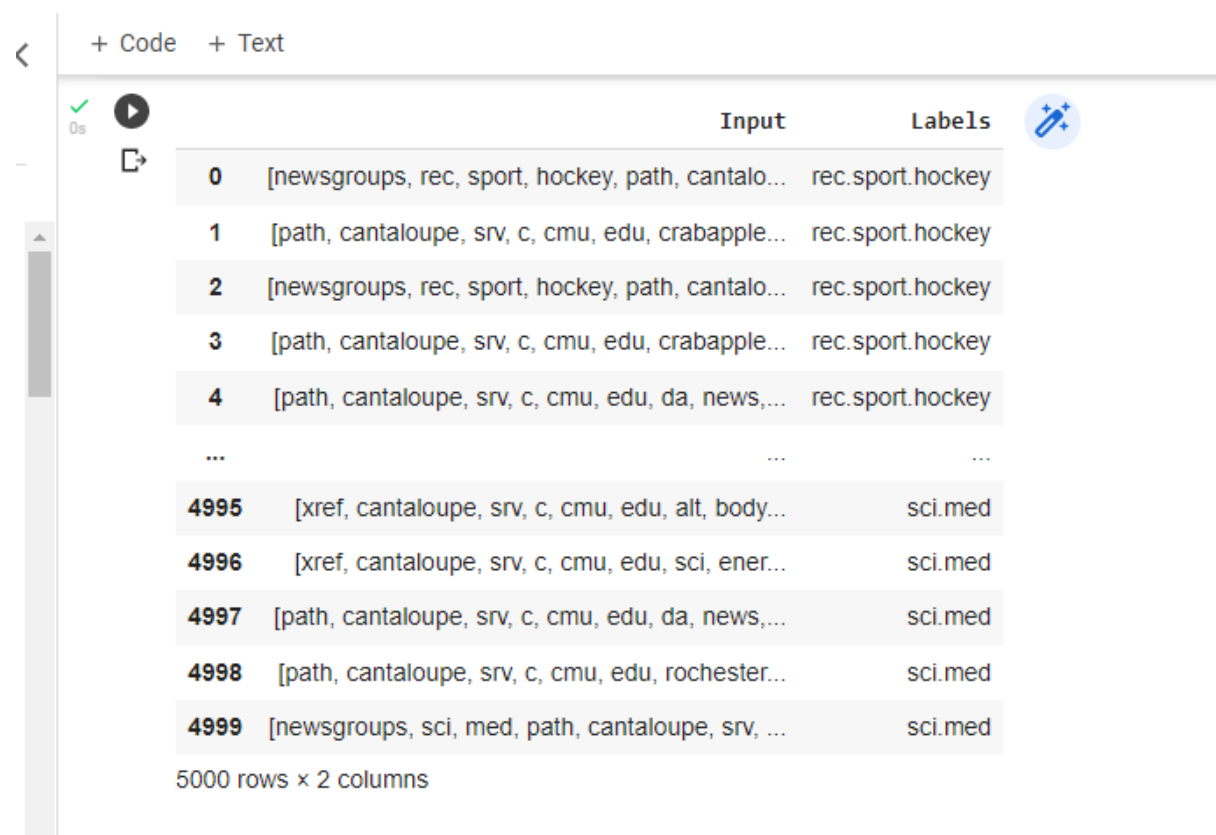
technique. Here we create 4 dictionary 1st dict1 contain count token and nested dictionary which contain name of class for which they that particular belong and count of that token in that class we split data in to ratio of 80:20 as train data and test and 2nd for CF and 3rd for ICF which is computed by using formula mention in slide

Then i create a dataframe that contain 5 column

`['Terms', 'C1', 'C2', 'C3', 'C4', 'C5']` where Terms is token from class and C1 ,C2,C3,C4,C5 are the following class
`['rec.sport.hockey', 'sci.space',
'talk.politics.misc', 'comp.graphics',
'Sci.med']`

Then i create another data frame DF that contain training data

Screenshot of dataset



0s

	Input	Labels
0	[newsgroups, rec, sport, hockey, path, cantalo...	rec.sport.hockey
1	[path, cantaloupe, srv, c, cmu, edu, crabapple...	rec.sport.hockey
2	[newsgroups, rec, sport, hockey, path, cantalo...	rec.sport.hockey
3	[path, cantaloupe, srv, c, cmu, edu, crabapple...	rec.sport.hockey
4	[path, cantaloupe, srv, c, cmu, edu, da, news,...	rec.sport.hockey
...
4995	[xref, cantaloupe, srv, c, cmu, edu, alt, body...	sci.med
4996	[xref, cantaloupe, srv, c, cmu, edu, sci, ener...	sci.med
4997	[path, cantaloupe, srv, c, cmu, edu, da, news,...	sci.med
4998	[path, cantaloupe, srv, c, cmu, edu, rochester...	sci.med
4999	[newsgroups, sci, med, path, cantaloupe, srv, ...	sci.med

5000 rows × 2 columns

Screenshot of training data

```

x_train
1142 [path, cantaloupe, srv, c, cmu, edu, rochester...
2589 [xref, cantaloupe, srv, c, cmu, edu, talk, pol...
1037 [newsgroups, sci, space, path, cantaloupe, srv...
3954 [path, cantaloupe, srv, c, cmu, edu, magnesium...
2420 [xref, cantaloupe, srv, c, cmu, edu, talk, rel...
...
2792 [xref, cantaloupe, srv, c, cmu, edu, alt, acti...
3321 [xref, cantaloupe, srv, c, cmu, edu, sci, med,...
3007 [newsgroups, comp, graphic, path, cantaloupe, ...
951 [newsgroups, rec, sport, hockey, path, cantalo...
1295 [newsgroups, sci, space, path, cantaloupe, srv...
Name: Input, Length: 4000, dtype: object

```

Screenshot of ICF classwise for each term

	Terms	C1	C2	C3	C4	C5
0	path	0.000000	0.000000	0.000000	0.000000	0.000000
1	cantaloupe	0.000000	0.000000	0.000000	0.000000	0.000000
2	srv	0.000000	0.000000	0.000000	0.000000	0.000000
3	c	0.000000	0.000000	0.000000	0.000000	0.000000
4	cmu	0.000000	0.000000	0.000000	0.000000	0.000000
...
46356	44w	2.321928	2.321928	2.321928	2.321928	2.321928
46357	tigrrs_r_us	2.321928	2.321928	2.321928	2.321928	2.321928
46358	1rkaqkinnmpa	2.321928	2.321928	2.321928	2.321928	2.321928
46359	pulverized	2.321928	2.321928	2.321928	2.321928	2.321928
46360	weeniehawks	2.321928	2.321928	2.321928	2.321928	2.321928

46361 rows x 6 columns

the confusion matrix corresponding to the best accuracies are as follows:

for 80:20 split

For 70:30 split:

For 50:50 split:

Analysis:

From the above accuracies we can say that 70:30 split is giving good accuracy out of all. And when we have increased the features upto a limit the accuracy increase and then it stops increasing.