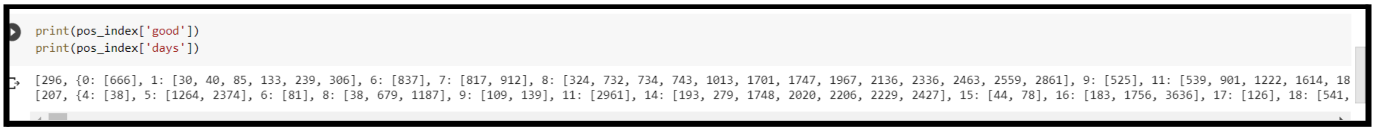
Output

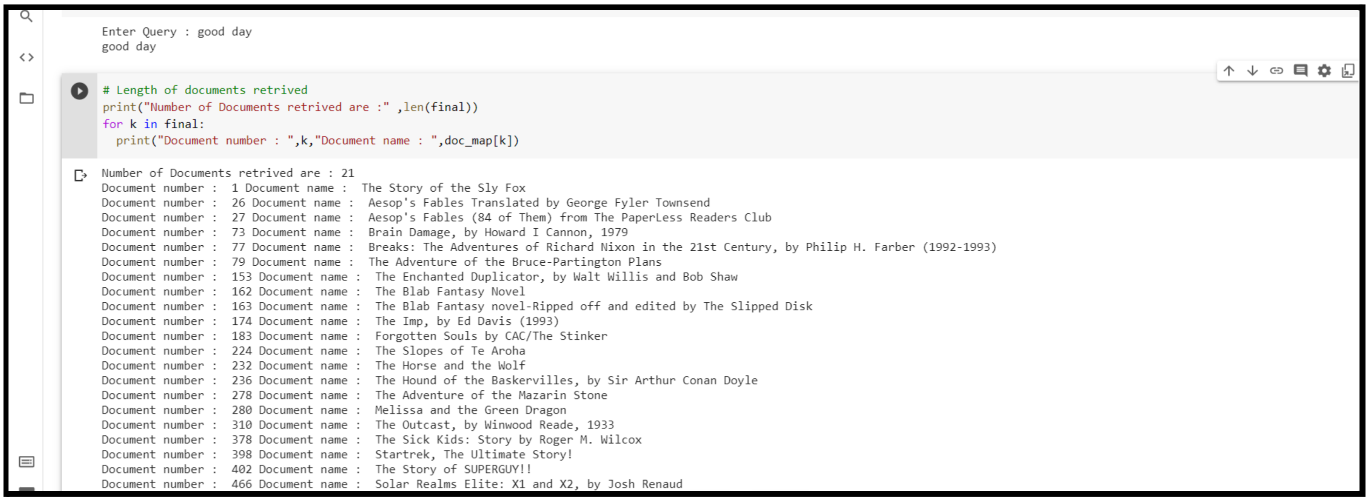
1. Posting index list-



```
print(pos_index['good'])
print(pos_index['days'])
```

```
[296, {0: [666], 1: [30, 40, 85, 133, 239, 306], 6: [837], 7: [817, 912], 8: [324, 732, 734, 743, 1013, 1701, 1747, 1967, 2136, 2336, 2463, 2559, 2861], 9: [525], 11: [539, 901, 1222, 1614, 18
[207, {4: [38], 5: [1264, 2374], 6: [81], 8: [38, 679, 1187], 9: [109, 139], 11: [2961], 14: [193, 279, 1748, 2020, 2206, 2229, 2427], 15: [44, 78], 16: [183, 1756, 3636], 17: [126], 18: [541,
```

Input query- good day



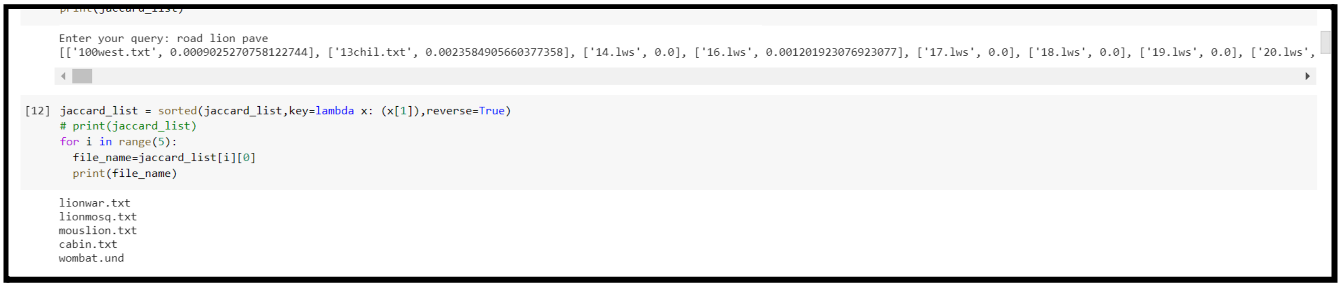
```
Enter Query : good day
good day
```

```
# Length of documents retrived
print("Number of Documents retrived are :" ,len(final))
for k in final:
  print("Document number : ",k,"Document name : ",doc_map[k])
```

```
Number of Documents retrived are : 21
Document number :  1 Document name :  The Story of the Sly Fox
Document number :  26 Document name :  Aesop's Fables Translated by George Fyler Townsend
Document number :  27 Document name :  Aesop's Fables (84 of Them) from The PaperLess Readers Club
Document number :  73 Document name :  Brain Damage, by Howard I Cannon, 1979
Document number :  77 Document name :  Breaks: The Adventures of Richard Nixon in the 21st Century, by Philip H. Farber (1992-1993)
Document number :  79 Document name :  The Adventure of the Bruce-Partington Plans
Document number :  153 Document name :  The Enchanted Duplicator, by Walt Willis and Bob Shaw
Document number :  162 Document name :  The Blab Fantasy Novel
Document number :  163 Document name :  The Blab Fantasy novel-Ripped off and edited by The Slipped Disk
Document number :  174 Document name :  The Imp, by Ed Davis (1993)
Document number :  183 Document name :  Forgotten Souls by CAC/The Stinker
Document number :  224 Document name :  The Slopes of Te Aroha
Document number :  232 Document name :  The Horse and the Wolf
Document number :  236 Document name :  The Hound of the Baskervilles, by Sir Arthur Conan Doyle
Document number :  278 Document name :  The Adventure of the Mazarin Stone
Document number :  280 Document name :  Melissa and the Green Dragon
Document number :  310 Document name :  The Outcast, by Winwood Reade, 1933
Document number :  378 Document name :  The Sick Kids: Story by Roger M. Wilcox
Document number :  398 Document name :  Startrek, The Ultimate Story!
Document number :  402 Document name :  The Story of SUPERGUY!!
Document number :  466 Document name :  Solar Realms Elite: X1 and X2, by Josh Renaud
```

2. Input Query- lion pave road

i). Jaccard Coefficient



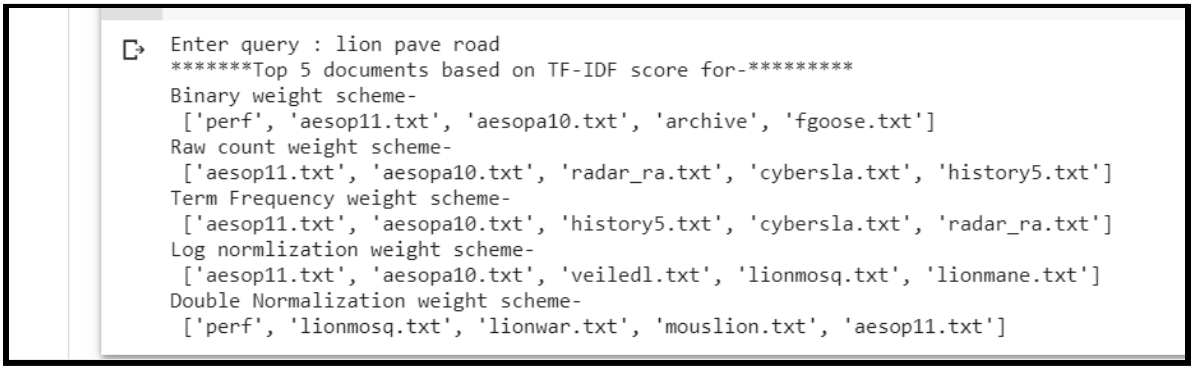
```
print(jaccard_list)
```

```
Enter your query: road lion pave
[['100west.txt', 0.0009025270758122744], ['13chil.txt', 0.0023584905660377358], ['14.lws', 0.0], ['16.lws', 0.0012019230769230077], ['17.lws', 0.0], ['18.lws', 0.0], ['19.lws', 0.0], ['20.lws',
```

```
[12] jaccard_list = sorted(jaccard_list,key=lambda x: (x[1]),reverse=True)
     # print(jaccard_list)
     for i in range(5):
       file_name=jaccard_list[i][0]
       print(file_name)
```

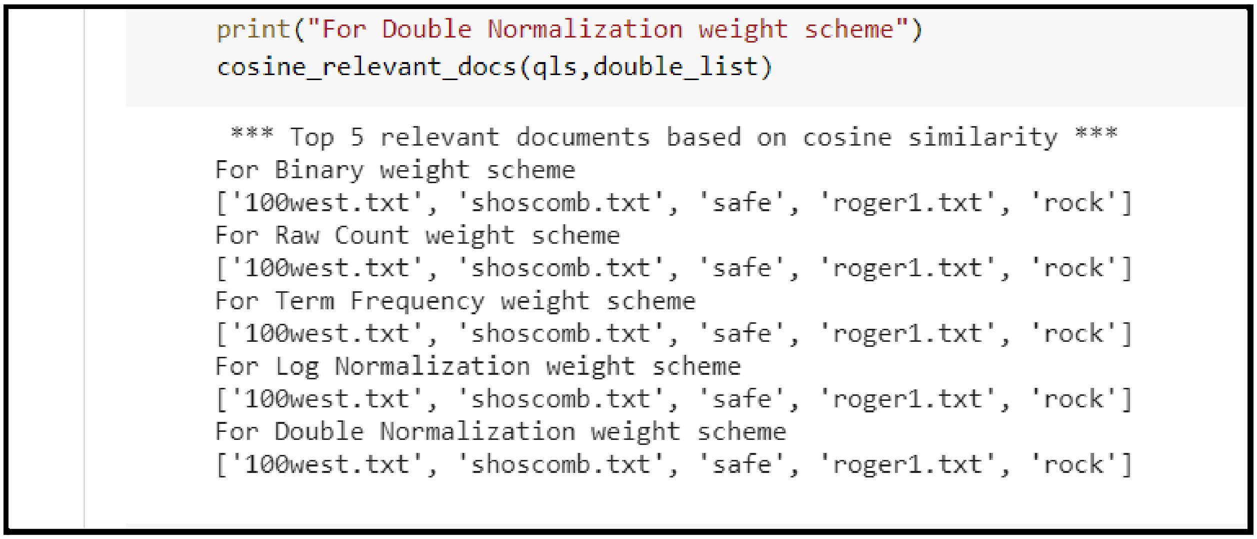
```
lionwar.txt
lionmosq.txt
mouslion.txt
cabin.txt
wombat.und
```

ii). TF-IDF



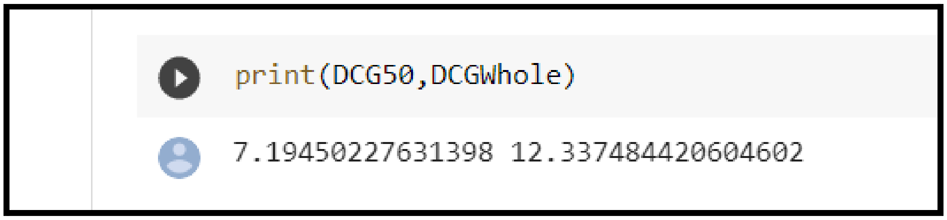
```
Enter query : lion pave road
*******Top 5 documents based on TF-IDF score for-*********
Binary weight scheme-
 ['perf', 'aesop11.txt', 'aesopa10.txt', 'archive', 'fgoose.txt']
Raw count weight scheme-
 ['aesop11.txt', 'aesopa10.txt', 'radar_ra.txt', 'cybersla.txt', 'history5.txt']
Term Frequency weight scheme-
 ['aesop11.txt', 'aesopa10.txt', 'history5.txt', 'cybersla.txt', 'radar_ra.txt']
Log normlization weight scheme-
 ['aesop11.txt', 'aesopa10.txt', 'veiledl.txt', 'lionmosq.txt', 'lionmane.txt']
Double Normalization weight scheme-
 ['perf', 'lionmosq.txt', 'lionwar.txt', 'mouslion.txt', 'aesop11.txt']
```

iii). Cosine Similarity



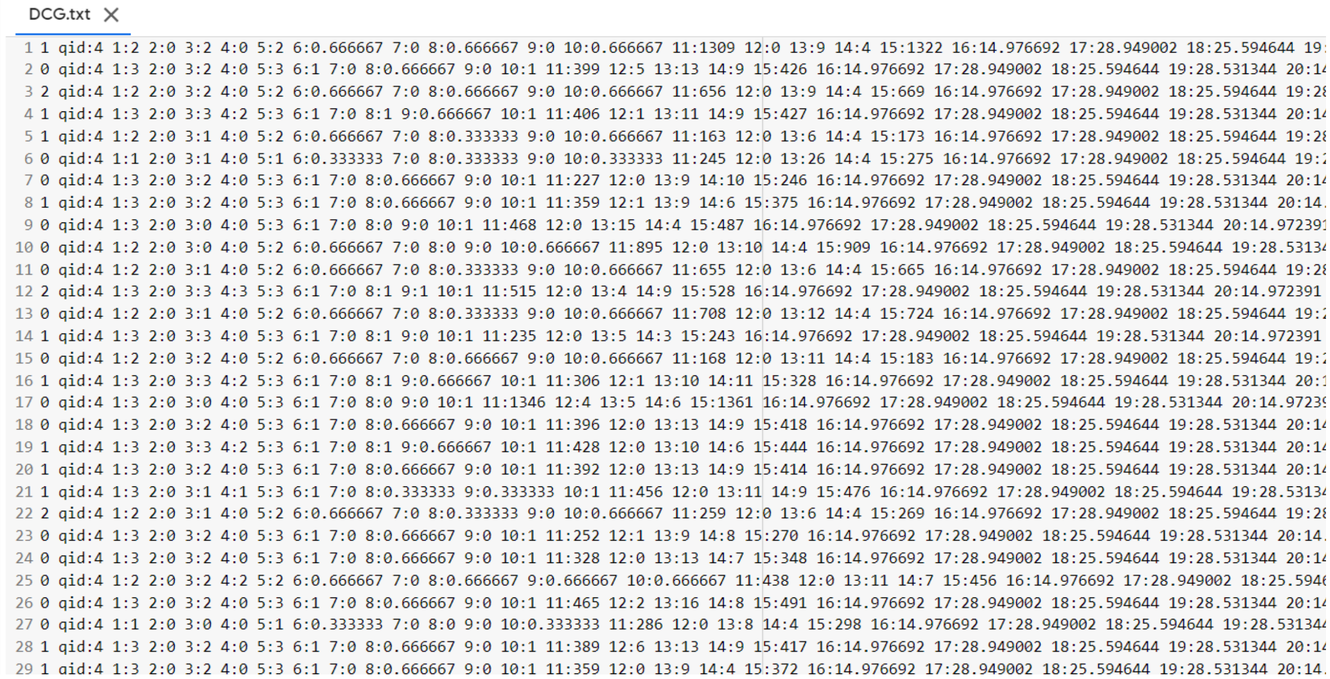
```
print("For Double Normalization weight scheme")
cosine_relevant_docs(qls,double_list)

 *** Top 5 relevant documents based on cosine similarity ***
For Binary weight scheme
['100west.txt', 'shoscomb.txt', 'safe', 'roger1.txt', 'rock']
For Raw Count weight scheme
['100west.txt', 'shoscomb.txt', 'safe', 'roger1.txt', 'rock']
For Term Frequency weight scheme
['100west.txt', 'shoscomb.txt', 'safe', 'roger1.txt', 'rock']
For Log Normalization weight scheme
['100west.txt', 'shoscomb.txt', 'safe', 'roger1.txt', 'rock']
For Double Normalization weight scheme
['100west.txt', 'shoscomb.txt', 'safe', 'roger1.txt', 'rock']
```

3. DCG at 50 and DCG for whole dataset



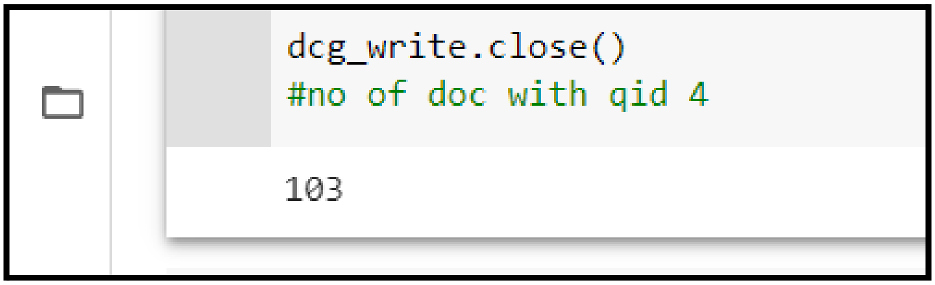
```
print(DCG50,DCGWhole)

7.19450227631398 12.337484420604602
```

File saving data whose qid=4 in DCG.txt



```
1 qid:4 1:2 2:0 3:2 4:0 5:2 6:0.666667 7:0 8:0.666667 9:0 10:0.666667 11:1309 12:0 13:9 14:4 15:1322 16:14.976692 17:28.949002 18:25.594644 19:
0 qid:4 1:3 2:0 3:2 4:0 5:3 6:1 7:0 8:0.666667 9:0 10:1 11:399 12:5 13:13 14:9 15:426 16:14.976692 17:28.949002 18:25.594644 19:28.531344 20:14
2 qid:4 1:2 2:0 3:2 4:0 5:2 6:0.666667 7:0 8:0.666667 9:0 10:0.666667 11:656 12:0 13:9 14:4 15:669 16:14.976692 17:28.949002 18:25.594644 19:28
1 qid:4 1:3 2:0 3:3 4:2 5:3 6:1 7:0 8:1 9:0.666667 10:1 11:406 12:1 13:11 14:9 15:427 16:14.976692 17:28.949002 18:25.594644 19:28.531344 20:14
1 qid:4 1:2 2:0 3:1 4:0 5:2 6:0.666667 7:0 8:0.333333 9:0 10:0.666667 11:163 12:0 13:6 14:4 15:173 16:14.976692 17:28.949002 18:25.594644 19:28
0 qid:4 1:1 2:0 3:1 4:0 5:1 6:0.333333 7:0 8:0.333333 9:0 10:0.333333 11:245 12:0 13:26 14:4 15:275 16:14.976692 17:28.949002 18:25.594644 19:2
0 qid:4 1:3 2:0 3:2 4:0 5:3 6:1 7:0 8:0.666667 9:0 10:1 11:227 12:0 13:9 14:10 15:246 16:14.976692 17:28.949002 18:25.594644 19:28.531344 20:14
1 qid:4 1:3 2:0 3:2 4:0 5:3 6:1 7:0 8:0.666667 9:0 10:1 11:359 12:1 13:9 14:6 15:375 16:14.976692 17:28.949002 18:25.594644 19:28.531344 20:14.
0 qid:4 1:3 2:0 3:0 4:0 5:3 6:1 7:0 8:0 9:0 10:1 11:468 12:0 13:15 14:4 15:487 16:14.976692 17:28.949002 18:25.594644 19:28.531344 20:14.97239
0 qid:4 1:2 2:0 3:0 4:0 5:2 6:0.666667 7:0 8:0 9:0 10:0.666667 11:895 12:0 13:10 14:4 15:909 16:14.976692 17:28.949002 18:25.594644 19:28.5313
0 qid:4 1:2 2:0 3:1 4:0 5:2 6:0.666667 7:0 8:0.333333 9:0 10:0.666667 11:655 12:0 13:6 14:4 15:665 16:14.976692 17:28.949002 18:25.594644 19:28
2 qid:4 1:3 2:0 3:3 4:3 5:3 6:1 7:0 8:1 9:1 10:1 11:515 12:0 13:4 14:9 15:528 16:14.976692 17:28.949002 18:25.594644 19:28.531344 20:14.972391
0 qid:4 1:2 2:0 3:1 4:0 5:2 6:0.666667 7:0 8:0.333333 9:0 10:0.666667 11:708 12:0 13:12 14:4 15:724 16:14.976692 17:28.949002 18:25.594644 19:2
1 qid:4 1:3 2:0 3:3 4:0 5:3 6:1 7:0 8:1 9:0 10:1 11:235 12:0 13:5 14:3 15:243 16:14.976692 17:28.949002 18:25.594644 19:28.531344 20:14.972391
0 qid:4 1:2 2:0 3:2 4:0 5:2 6:0.666667 7:0 8:0.666667 9:0 10:0.666667 11:168 12:0 13:11 14:4 15:183 16:14.976692 17:28.949002 18:25.594644 19:2
1 qid:4 1:3 2:0 3:3 4:2 5:3 6:1 7:0 8:1 9:0.666667 10:1 11:306 12:1 13:10 14:11 15:328 16:14.976692 17:28.949002 18:25.594644 19:28.531344 20:1
0 qid:4 1:3 2:0 3:0 4:0 5:3 6:1 7:0 8:0 9:0 10:1 11:1346 12:4 13:5 14:6 15:1361 16:14.976692 17:28.949002 18:25.594644 19:28.531344 20:14.9723
0 qid:4 1:3 2:0 3:2 4:0 5:3 6:1 7:0 8:0.666667 9:0 10:1 11:396 12:0 13:13 14:9 15:418 16:14.976692 17:28.949002 18:25.594644 19:28.531344 20:14
1 qid:4 1:3 2:0 3:3 4:2 5:3 6:1 7:0 8:1 9:0.666667 10:1 11:428 12:0 13:10 14:6 15:444 16:14.976692 17:28.949002 18:25.594644 19:28.531344 20:14
1 qid:4 1:3 2:0 3:2 4:0 5:3 6:1 7:0 8:0.666667 9:0 10:1 11:392 12:0 13:13 14:9 15:414 16:14.976692 17:28.949002 18:25.594644 19:28.531344 20:14
1 qid:4 1:3 2:0 3:1 4:1 5:3 6:1 7:0 8:0.333333 9:0.333333 10:1 11:456 12:0 13:11 14:9 15:476 16:14.976692 17:28.949002 18:25.594644 19:28.5313
2 qid:4 1:2 2:0 3:1 4:0 5:2 6:0.666667 7:0 8:0.333333 9:0 10:0.666667 11:259 12:0 13:6 14:4 15:269 16:14.976692 17:28.949002 18:25.594644 19:28
0 qid:4 1:3 2:0 3:2 4:0 5:3 6:1 7:0 8:0.666667 9:0 10:1 11:252 12:1 13:9 14:8 15:270 16:14.976692 17:28.949002 18:25.594644 19:28.531344 20:14.
0 qid:4 1:3 2:0 3:2 4:0 5:3 6:1 7:0 8:0.666667 9:0 10:1 11:328 12:0 13:13 14:7 15:348 16:14.976692 17:28.949002 18:25.594644 19:28.531344 20:14
0 qid:4 1:2 2:0 3:2 4:2 5:2 6:0.666667 7:0 8:0.666667 9:0.666667 10:0.666667 11:438 12:0 13:11 14:7 15:456 16:14.976692 17:28.949002 18:25.594
0 qid:4 1:3 2:0 3:2 4:0 5:3 6:1 7:0 8:0.666667 9:0 10:1 11:465 12:2 13:16 14:8 15:491 16:14.976692 17:28.949002 18:25.594644 19:28.531344 20:14
0 qid:4 1:1 2:0 3:0 4:0 5:1 6:0.333333 7:0 8:0 9:0 10:0.333333 11:286 12:0 13:8 14:4 15:298 16:14.976692 17:28.949002 18:25.594644 19:28.531344
1 qid:4 1:3 2:0 3:2 4:0 5:3 6:1 7:0 8:0.666667 9:0 10:1 11:389 12:6 13:13 14:9 15:417 16:14.976692 17:28.949002 18:25.594644 19:28.531344 20:14
1 qid:4 1:3 2:0 3:2 4:0 5:3 6:1 7:0 8:0.666667 9:0 10:1 11:359 12:0 13:9 14:4 15:372 16:14.976692 17:28.949002 18:25.594644 19:28.531344 20:14.
```

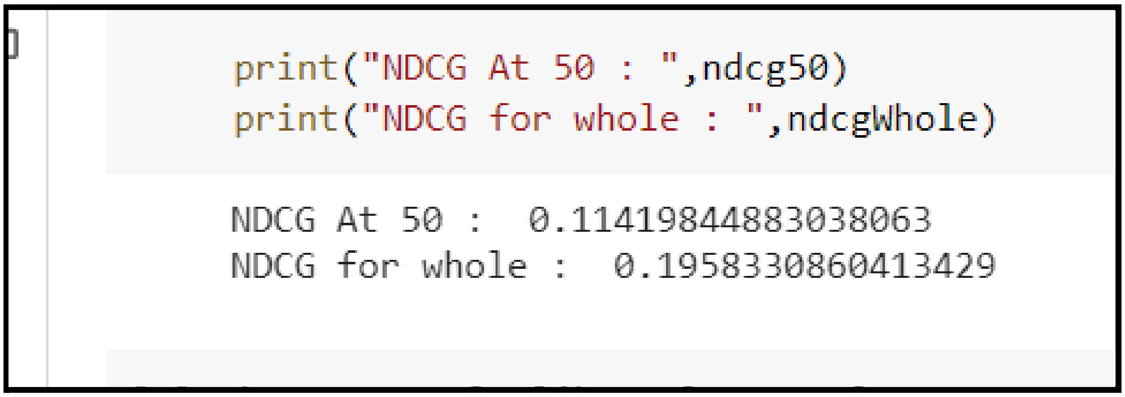
Total Number of documents with qid:4



```
dcg_write.close()
#no of doc with qid 4
```

```
103
```

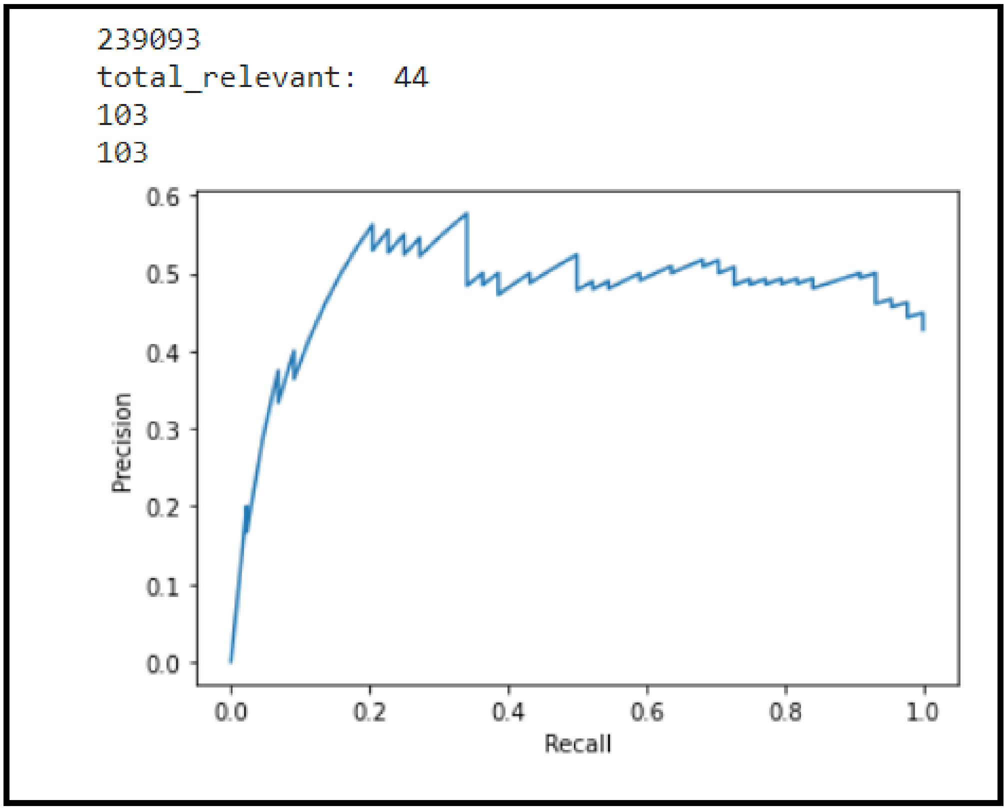
nDCG value at 50 and nDCG value for whole dataset



```
print("NDCG At 50 : ",ndcg50)
print("NDCG for whole : ",ndcgWhole)
```

```
NDCG At 50 :  0.11419844883038063
NDCG for whole :  0.1958330860413429
```

Graph of Precision-Recall curve for qid:4



```
239093
total_relevant:  44
103
103
```