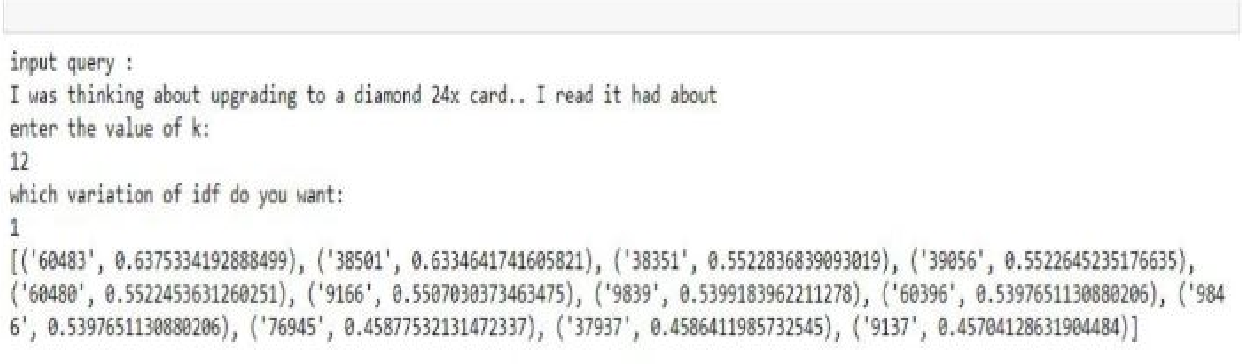
Assignment 3

Submitted by,
Swati Verma
MT19073

Question 1.

1. **Query processing using cosine similarity**

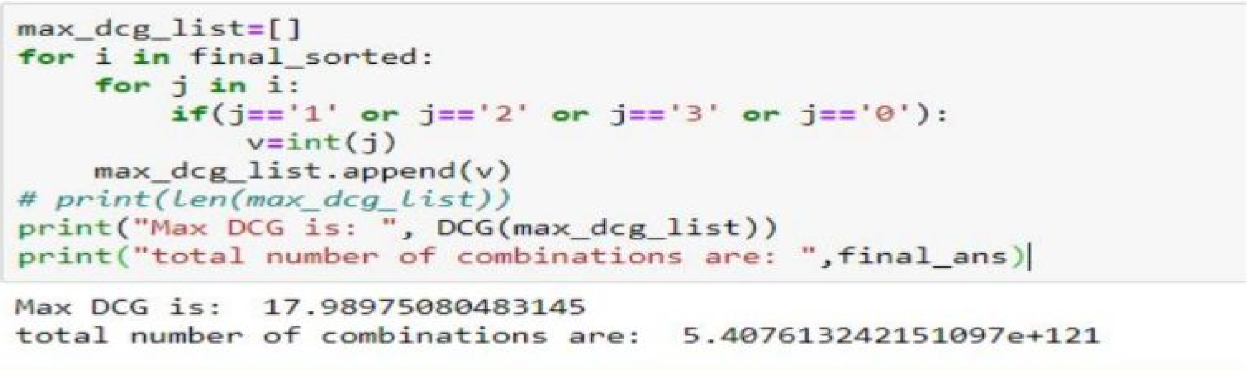


```
input query :
I was thinking about upgrading to a diamond 24x card.. I read it had about
enter the value of k:
12
which variation of idf do you want:
1
[('60483', 0.6375334192888499), ('38501', 0.6334641741605821), ('38351', 0.5522836839093019), ('39056', 0.5522645235176635),
('60480', 0.5522453631260251), ('9166', 0.5507030373463475), ('9839', 0.5399183962211278), ('60396', 0.5397651130880206), ('984
6', 0.5397651130880206), ('76945', 0.45877532131472337), ('37937', 0.4586411985732545), ('9137', 0.45704128631904484)]
```

Analysis for the value of r:
To choose the value for r, I have taken the mean of the length of posting list. This mean came out to be around 20. It can be a good approach because user is mostly concerned about the precision value rather than recall. So, they might be happy with the top 20 relevant documents. This would reduce the number of access to the low list.
The second method for choosing the value of r is taking the mean of square root of the length of posting list. But the value of r came out to be around 4-5 which is very less. Such a small value for r would increase the number of access to the low list which will ruin the idea of using champion list.

Question 2.

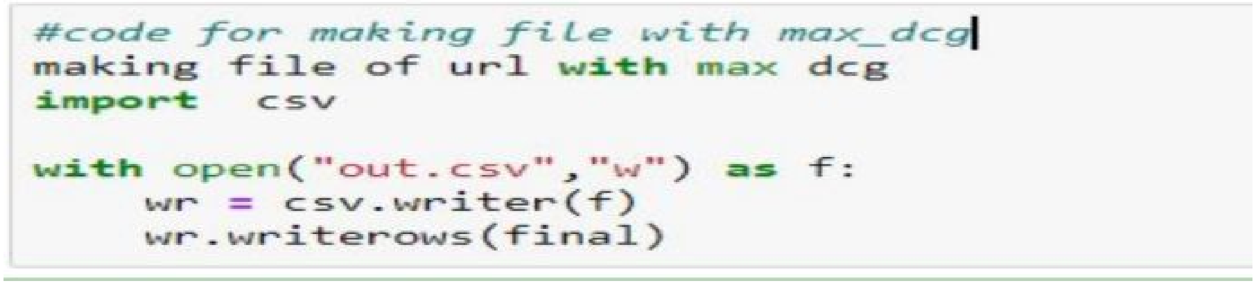
1. **Max dcg and number of possible combinations**



```
max_dcg_list=[]
for i in final_sorted:
    for j in i:
        if(j=='1' or j=='2' or j=='3' or j=='0'):
            v=int(j)
    max_dcg_list.append(v)
# print(len(max_dcg_list))
print("Max DCG is: ", DCG(max_dcg_list))
print("total number of combinations are: ",final_ans)
```

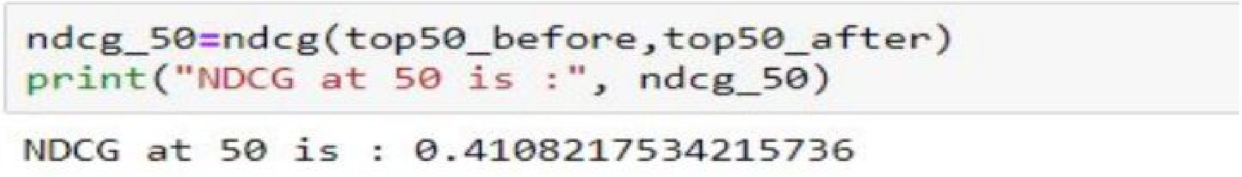
```
Max DCG is:  17.98975080483145
total number of combinations are:  5.407613242151097e+121
```

2. Saving file with max dcg as out.csv



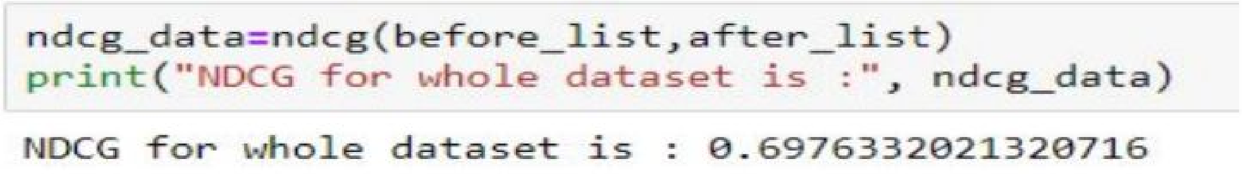
```
#code for making file with max_dcg
making file of url with max dcg
import  csv

with open("out.csv","w") as f:
     wr = csv.writer(f)
     wr.writerows(final)
```

3.NDCG at 50 and NDCG for whole document for qid:4



```
ndcg_50=ndcg(top50_before,top50_after)
print("NDCG at 50 is :", ndcg_50)
```

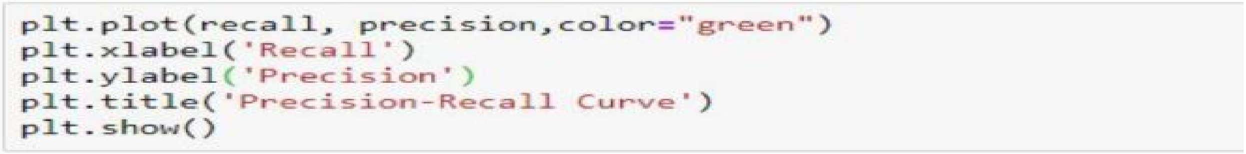
```
NDCG at 50 is : 0.4108217534215736
```



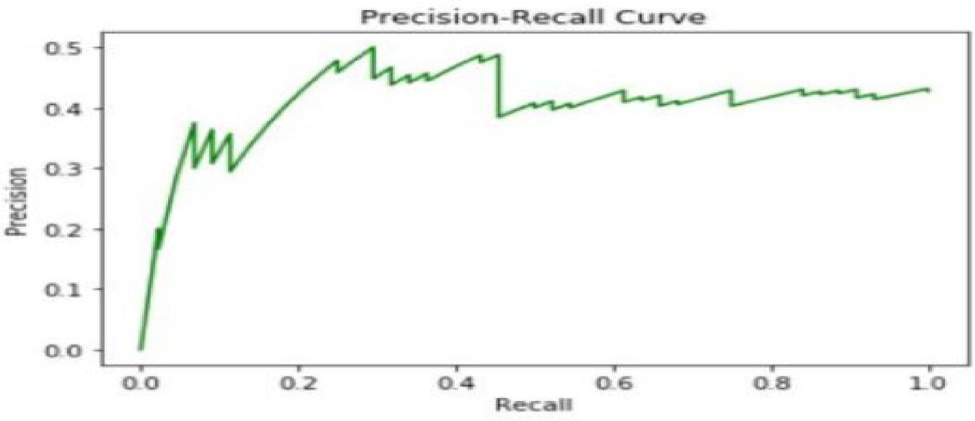
```
ndcg_data=ndcg(before_list,after_list)
print("NDCG for whole dataset is :", ndcg_data)
```

```
NDCG for whole dataset is : 0.6976332021320716
```

4. Precision-Recall Curve



```
plt.plot(recall, precision,color="green")
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.title('Precision-Recall Curve')
plt.show()
```



Ques 3.

1. Explain Relationship between ROC and PR Curve.

Ans: In machine learning, ROC curve is used for performance measurement of the classification problem. ROC curve is plotted between True Positive Rate (sensitive or recall) and False Positive rate(1-specificity). ROC curve tells how much a model is capable of identifying between different classes.

PR curve is plotted between precision and recall. Precision mainly focuses on the correct positive predictions of any classification problem.

So, when the focus is more on positive values then PR curve is more useful but when the focus is on both positive and negative values then ROC curve is more useful.

According to the theorem given in the research paper, **if the dataset consists of fixed number of positive and negative values, then there exists a one-to-one correspondence between a curve in ROC space and a curve in PR space, such that the curves contain exactly the same confusion matrices, if Recall != 0.**

As we know that each point in the ROC curve corresponds to a unique set of TP, TN, FP and FN. But in precision and recall we ignore the true negative values. But if a fixed dataset with known positive and negative values is given then we can easily find the number of true negative values which can again give unique values for the confusion matrix. Hence we can say that there is one to one correspondence between points in the confusion matrix and points in ROC space and same is the case with points in PR space and points in confusion matrix. This implies that we can translate curve in ROC space to curve in PR space and vice versa.

2. Prove that a curve dominates in ROC space if and only if it dominates in PR space.

Proof bycontradiction:

Claim: If curve 1 dominates in the ROC curve than curve 2, then the same curve will

dominate in the PR
curve

If curve 1 dominates in ROC space but does not dominate in PR curve then there must be some point A in PR curve which has same recall as point B in ROC curve but have higher precision i.e. Precision(A)> Precision(B).

We know that TPR=Recall= TP/TP+FP.

If TPR(A)=TPR(B), then TP(A)=TP(B). **But as curve 1 dominates in ROC , hence**

FPR(A)>= FPR(B)

and as we know,

FPR = FP/total negatives

Hence, FP(A)>=FP(B),

We also know that **precision=TP/TP+FP.**

If FP(A)>=FP(B) then, precision(A)<=precision(B) which contradicts our assumption of precision(A)> = precision(B). Hence we can say that if a curve dominates in ROC then it will dominate in PR space also.

Claim 2: If curve 1 dominates in the PR curve than curve 2, then the same curve will

dominate in the ROC curve.

So, Curve 1 doesn't dominate in ROC space means, there exists some point B on 2, such that point A on curve 1, with the same TPR but different (FPR).

i.e FPR(A) < FPR(B) and Recall(A)= Recall(B)

Curve 1 dominates in PR space, so Precision(A) < Precision(B)

And we have, Recall(A) =Recall(B).

Recall(A) = TP(A)/TP+FP

Recall(B)=TP(B)/TP+FP

For Fixed sample ,

TP(A) =TP(B)

And we have established Precision(A) < Precision(B)

Precision(A) = TP(A)/ TP(A) +FP(A)

Precision(B) = TP(B) / TP(B) +FP(B)

Also we have FPR(A) < FPR(B),

then, False Positive(A) < False Positive(B) ,FP(A) < FP(B)

Then it can be implied that:

Precision(A) >Precision(B).

Hence contradicting the claim Precision(A) < Precision(B)**.**

Thus Curve 2 also dominates in PR space.

3. It is incorrect to interpolate between points in PR space. When and why does this happen? How will you tackle this problem?

Ans: No, it is not incorrect to interpolate between points in precision recall space. It is easy and straightforward in ROC to curve to interpolate between two points just by connecting them with a straight line. But interpolating points in PR space may not give correct result sometimes because as recall rises the precision need not to change linearly with recall as False Positive replaces False negative in the denominator of the precision metric

Such a problem rise when two points are not close enough in the Precision Recall curve. In such case the performance of the system can be overestimated by the interpolation.

To overcome the problem of interpolation, we can translate the ROC curve to PR space because the convex hull in the ROC curve gives the best result or we can say it gives us the best area under the curve which can be suitable for interpolation in newly constructed precision recall curve.