Untitled2

May 24, 2018

In [8]:
```
import pandas as pd
import numpy as np

df = pd.read_csv('neural_network.csv')
df = df.drop('Unnamed: 0', axis=1)
df.sort_values('srch_id')
```

Out[8]:

	srch_id	position	ndcg
36899	1	27	0.0
13407	1	34	0.0
40280	21	25	0.0
29653	21	15	0.0
22128	36	27	0.0
15190	45	1	0.0
44932	46	26	0.0
26326	46	13	0.0
23659	47	12	0.0
30755	47	32	0.0
45563	57	31	0.0
40838	72	3	0.0
15417	73	21	0.0
26727	78	6	0.0
38868	81	19	0.0
14675	93	18	0.0
39629	93	20	0.0
43222	102	21	0.0
49526	102	8	0.0
45520	107	33	0.0
42249	111	16	0.0
30754	112	25	0.0
12158	113	33	0.0
27222	115	10	0.0
32621	116	15	0.0
12438	116	22	0.0
8293	128	25	0.0
34354	133	9	0.0
22270	134	32	0.0
48939	152	31	0.0

```
...        ...        ...   ...
33492   332555         24   0.0
36552   332559         18   0.0
32484   332563          8   0.0
27761   332568         37   0.0
8530    332570          4   0.0
22363   332578         20   0.0
31540   332579          7   0.0
20355   332585         16   0.0
43774   332586          3   0.0
9374    332586         32   0.0
49206   332591          8   0.0
20353   332602         18   0.0
46478   332610          2   0.0
16978   332613         26   0.0
28760   332616         22   0.0
7691    332617         10   0.0
6606    332622         12   0.0
4238    332622         18   0.0
5991    332628          7   1.0
4871    332665          8   0.0
25859   332668         19   0.0
35149   332686         33   1.0
33739   332697         18   0.0
47281   332701          3   0.0
15396   332724          4   0.0
36069   332755         14   0.0
3082    332757         34   0.0
28938   332768          7   0.0
5522    332768         25   0.0
46948   332785          1   5.0

[49585 rows x 3 columns]
```

```
In [36]: def dcg(i, rel):

             return (2**rel - 1)/(np.log2(i + 1))


         ##########################################################################################


         # CACULATING DCG
```

```
sum_dcg = 0

array_dcg = []


for k in range(0, len(df) - 1):

    if k != len(df) - 1:

        if(df['srch_id'][k] < df['srch_id'][k + 1]):

            sum_dcg += dcg(df['position'][k],df['ndcg'][k])

            array_dcg.append(sum_dcg)

            sum_dcg = 0

        else:

            sum_dcg += dcg(df['position'][k],df['ndcg'][k])

    else:

        sum_dcg += dcg(df['position'][k],df['ndcg'][k])

        array_dcg.append(sum_dcg)

        sum_dcg = 0




################################################################################


# CALCULATING IDCG


array_idcg = np.array([])

t_idcg = np.array([])

sum_idcg = 0
```

```
# l = 0


for k in range(0, len(df) - 1):
    if k != len(df) - 1:
        if(df['srch_id'][k] < df['srch_id'][k + 1]):
            t_idcg = np.append(t_idcg, df['ndcg'][k])
            t_idcg = np.sort(t_idcg)[::-1]
            for j in range(0, len(t_idcg) - 1):
                sum_idcg += dcg(j + 1, t_idcg[j])
            array_idcg = np.append(array_idcg, sum_idcg)
            sum_idcg = 0
            t_idcg = []
        else:
            t_idcg = np.append(t_idcg, df['ndcg'][k])


    else:
        t_idcg = np.append(t_idcg, df['ndcg'][k])
        t_idcg = np.sort(t_idcg)[::-1]
        for j in range(1, len(t_idcg) - 1):
            sum_idcg += dcg(j, t_idcg[j - 1])
        array_idcg = np.append(array_idcg, sum_idcg)



##############################################################################
```

```
In [37]: print 'nDGC = ', (np.sum(array_dcg)/np.sum(array_idcg))

nDGC =  0.6117436924653229
```