Untitled

May 24, 2018

In [15]:
```
import pandas as pd
import numpy as np
from sklearn.neural_network import MLPRegressor
```

In [2]:
```
df = pd.read_csv('train2.csv')
```

In [9]:
```
df = df.sample(frac=0.1)
```

In [10]:
```
data = ['srch_id',
 'prop_country_id',
 'year',
 'month',
 'prop_score',
 'prop_location_score1',
 'prop_location_score2',
 'diff_hist_price',
 'usd_diff',
 'star_diff',
 'srch_query_affinity_score',
 'orig_destination_distance',
 'prop_review_score',
  'price_rank',
 'stars_rank',
 'score_rank',
 'booking_bool',
 'click_bool',
 'price_per_pers',
 'random_bool',
 'price_usd',
 'total_price',
 'nr_pers',
 'prop_id_counts',
 'comp2_rate',
 'comp5_rate',
 'comp8_rate'
 ]

target = ['position', 'ndcg']
```

```
In [12]: X = df[data]
         y = df[target]
         from sklearn.cross_validation import cross_val_score
```

```
C:\Users\Ryan\Anaconda2\lib\site-packages\sklearn\cross_validation.py:41: DeprecationWarning: Th
  "This module will be removed in 0.20.", DeprecationWarning)
```

1 Neural network on default

```
In [18]: mlp = MLPRegressor()
         mlp.fit(X, y)
         print cross_val_score(mlp, X, y, cv = 10, scoring = 'mean_squared_error')
```

```
C:\Users\Ryan\Anaconda2\lib\site-packages\sklearn\metrics\scorer.py:100: DeprecationWarning: Sco
  sample_weight=sample_weight)
C:\Users\Ryan\Anaconda2\lib\site-packages\sklearn\metrics\scorer.py:100: DeprecationWarning: Sco
  sample_weight=sample_weight)
C:\Users\Ryan\Anaconda2\lib\site-packages\sklearn\metrics\scorer.py:100: DeprecationWarning: Sco
  sample_weight=sample_weight)
C:\Users\Ryan\Anaconda2\lib\site-packages\sklearn\metrics\scorer.py:100: DeprecationWarning: Sco
  sample_weight=sample_weight)
C:\Users\Ryan\Anaconda2\lib\site-packages\sklearn\metrics\scorer.py:100: DeprecationWarning: Sco
  sample_weight=sample_weight)
C:\Users\Ryan\Anaconda2\lib\site-packages\sklearn\metrics\scorer.py:100: DeprecationWarning: Sco
  sample_weight=sample_weight)
C:\Users\Ryan\Anaconda2\lib\site-packages\sklearn\metrics\scorer.py:100: DeprecationWarning: Sco
  sample_weight=sample_weight)
C:\Users\Ryan\Anaconda2\lib\site-packages\sklearn\metrics\scorer.py:100: DeprecationWarning: Sco
  sample_weight=sample_weight)
C:\Users\Ryan\Anaconda2\lib\site-packages\sklearn\metrics\scorer.py:100: DeprecationWarning: Sco
  sample_weight=sample_weight)
C:\Users\Ryan\Anaconda2\lib\site-packages\sklearn\metrics\scorer.py:100: DeprecationWarning: Sco
  sample_weight=sample_weight)
```

```
[-2.70647522e+04 -4.59309149e+03 -1.05928630e+06 -4.23229622e+03
 -1.02742250e+05 -4.90738780e+01 -5.74135621e+03 -1.74382848e+04
 -1.83175451e+06 -1.01414549e+04]
```

```
In [20]: best_solution = df.iloc[list(range(int(len(df)*0.5),int(len(df)*0.6) + 1))]

In [29]: df = best_solution

In [30]: list(df)

Out[30]: ['srch_id',
          'prop_country_id',
```

```
            'year',
            'month',
            'prop_score',
            'prop_location_score1',
            'prop_location_score2',
            'diff_hist_price',
            'usd_diff',
            'star_diff',
            'srch_query_affinity_score',
            'orig_destination_distance',
            'prop_review_score',
            'position',
            'price_rank',
            'stars_rank',
            'score_rank',
            'booking_bool',
            'click_bool',
            'price_per_pers',
            'random_bool',
            'price_usd',
            'total_price',
            'nr_pers',
            'prop_id_counts',
            'comp2_rate',
            'comp5_rate',
            'comp8_rate',
            'ndcg',
            'rank_rel']
```

```
In [31]: def dcg(i, rel):

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

```
In [35]: df = df[['srch_id', 'position', 'ndcg']]
         df.to_csv('neural_network.csv')
```