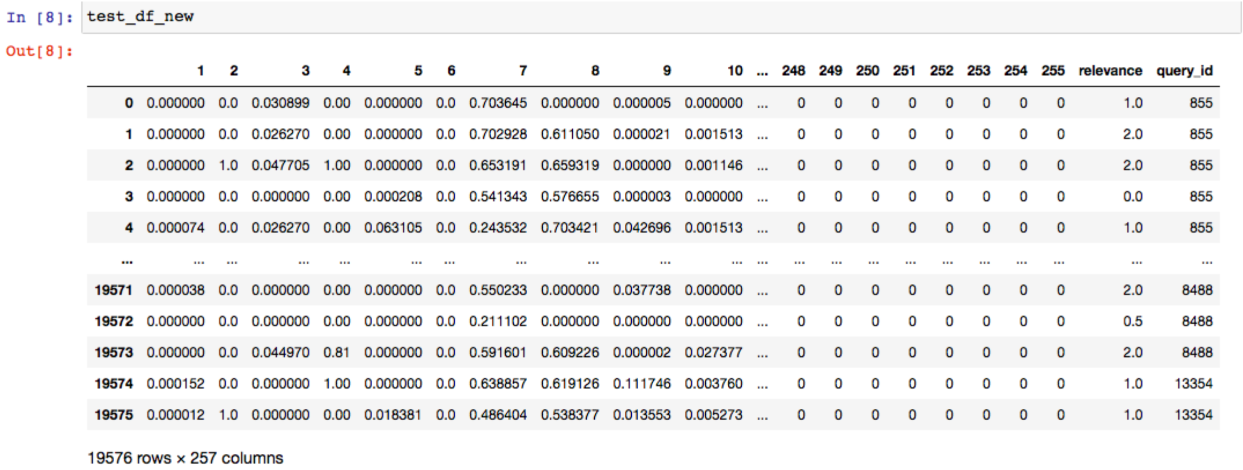
Zhamilya Saparova

IR Assignment 3 - Learning to Rank

In this assignment, I followed the CatBoost tutorial. The dataset was provided, which is Internet Mathematics 2009, Real Yandex data consisting of 245 features. The dataset was splitted on the train set and test set. Relevance labels range from 0 to 4.

Firstly, the dataset was read using the pandas read csv function. Then, I created an empty dictionary with the length of feature numbers. Further, filled with the relevance, query_id and values data the empty dictionary. I converted the dictionary to pandas dataframe and we had a table as below at the end.



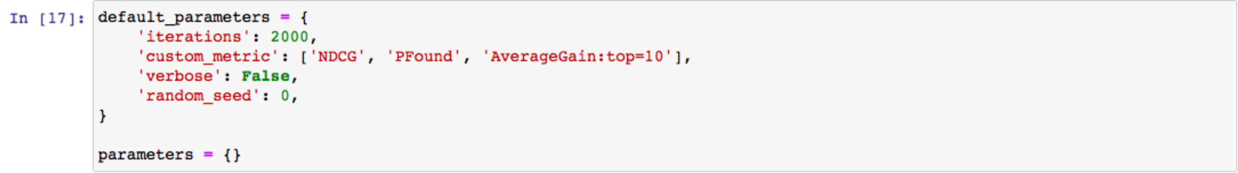
```
In [8]: test_df_new
```

Out[8]:

	1	2	3	4	5	6	7	8	9	10	...	248	249	250	251	252	253	254	255	relevance	query_id
0	0.000000	0.0	0.030899	0.00	0.000000	0.0	0.703645	0.000000	0.000005	0.000000	...	0	0	0	0	0	0	0	0	1.0	855
1	0.000000	0.0	0.026270	0.00	0.000000	0.0	0.702928	0.611050	0.000021	0.001513	...	0	0	0	0	0	0	0	0	2.0	855
2	0.000000	1.0	0.047705	1.00	0.000000	0.0	0.653191	0.659319	0.000000	0.001146	...	0	0	0	0	0	0	0	0	2.0	855
3	0.000000	0.0	0.000000	0.00	0.000208	0.0	0.541343	0.576655	0.000003	0.000000	...	0	0	0	0	0	0	0	0	0.0	855
4	0.000074	0.0	0.026270	0.00	0.063105	0.0	0.243532	0.703421	0.042696	0.001513	...	0	0	0	0	0	0	0	0	1.0	855
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
19571	0.000038	0.0	0.000000	0.00	0.000000	0.0	0.550233	0.000000	0.037738	0.000000	...	0	0	0	0	0	0	0	0	2.0	8488
19572	0.000000	0.0	0.000000	0.00	0.000000	0.0	0.211102	0.000000	0.000000	0.000000	...	0	0	0	0	0	0	0	0	0.5	8488
19573	0.000000	0.0	0.044970	0.81	0.000000	0.0	0.591601	0.609226	0.000002	0.027377	...	0	0	0	0	0	0	0	0	2.0	8488
19574	0.000152	0.0	0.000000	1.00	0.000000	0.0	0.638857	0.619126	0.111746	0.003760	...	0	0	0	0	0	0	0	0	1.0	13354
19575	0.000012	1.0	0.000000	0.00	0.018381	0.0	0.486404	0.538377	0.013553	0.005273	...	0	0	0	0	0	0	0	0	1.0	13354

19576 rows × 257 columns

Further, I assigned X and y, where X is feature values and y is relevance labels. Query_id were assigned to queries_train and queries_test variables.Experiments have been done on four methods of CatBoost, which are ['PairLogit', 'PairLogitPairwise', 'YetiRank', 'YetiRankPairwise']. Parameters for model fitting can be seen below.



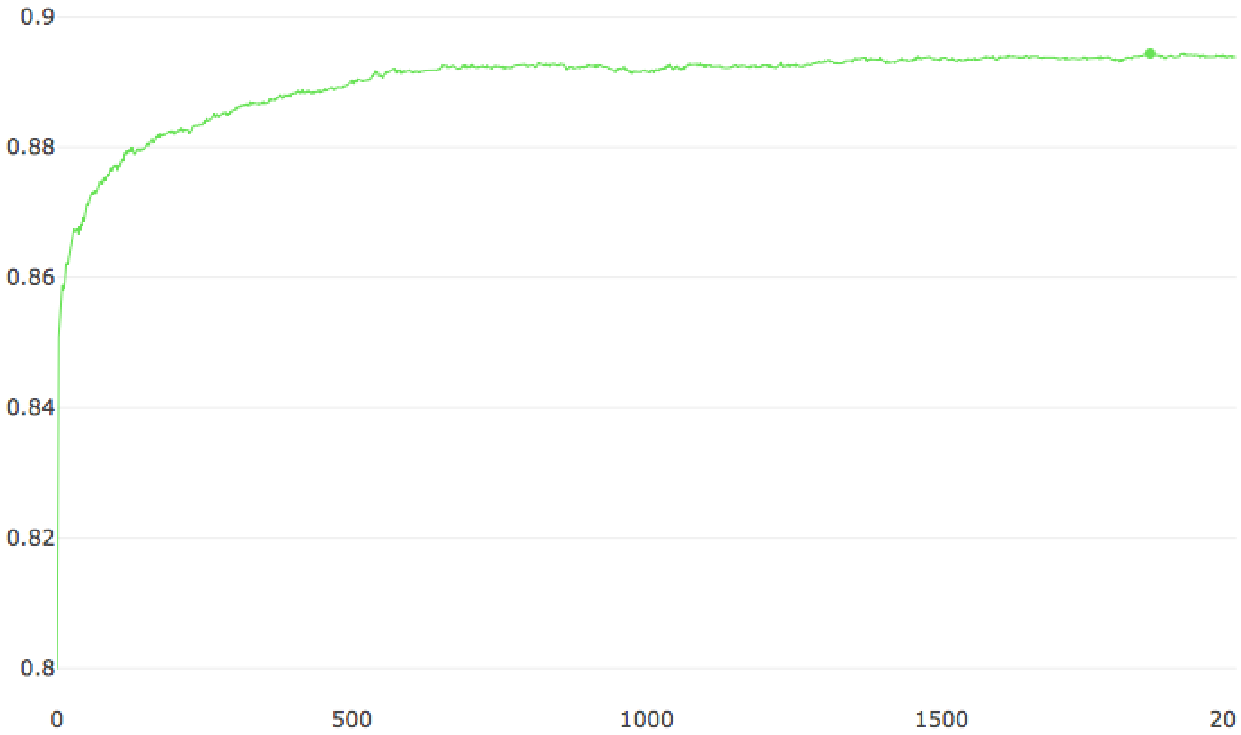
```
In [17]: default_parameters = {
             'iterations': 2000,
             'custom_metric': ['NDCG', 'PFound', 'AverageGain:top=10'],
             'verbose': False,
             'random_seed': 0,
         }

         parameters = {}
```

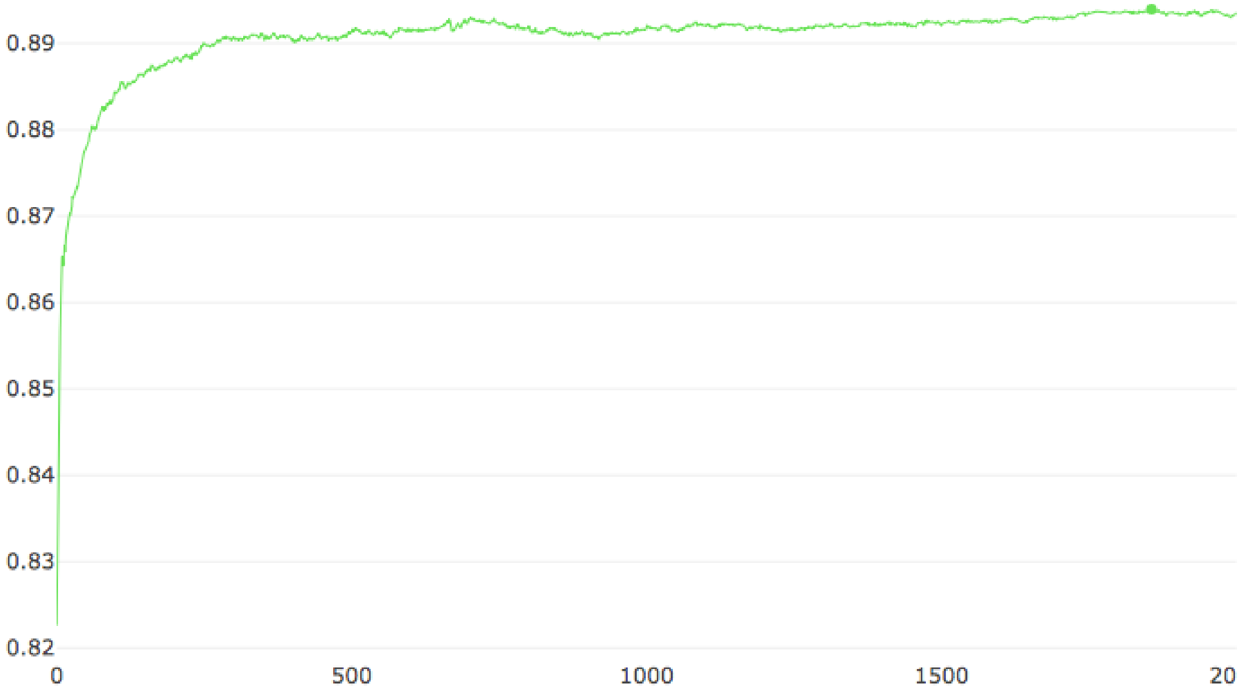
The best score was achieved by YetiRank in terms of time and accuracy. Second best method was PairLogit and third was PairLogitPairwise, but it took more time than YetiRankPairwise. Below can be seen visualization for each method and table of time and accuracies at the end. The ipynb file can be found on github https://github.com/zhamilyaa/ir_assignment.

Zhamilya Saparova

PairLogit NDCG Visualization

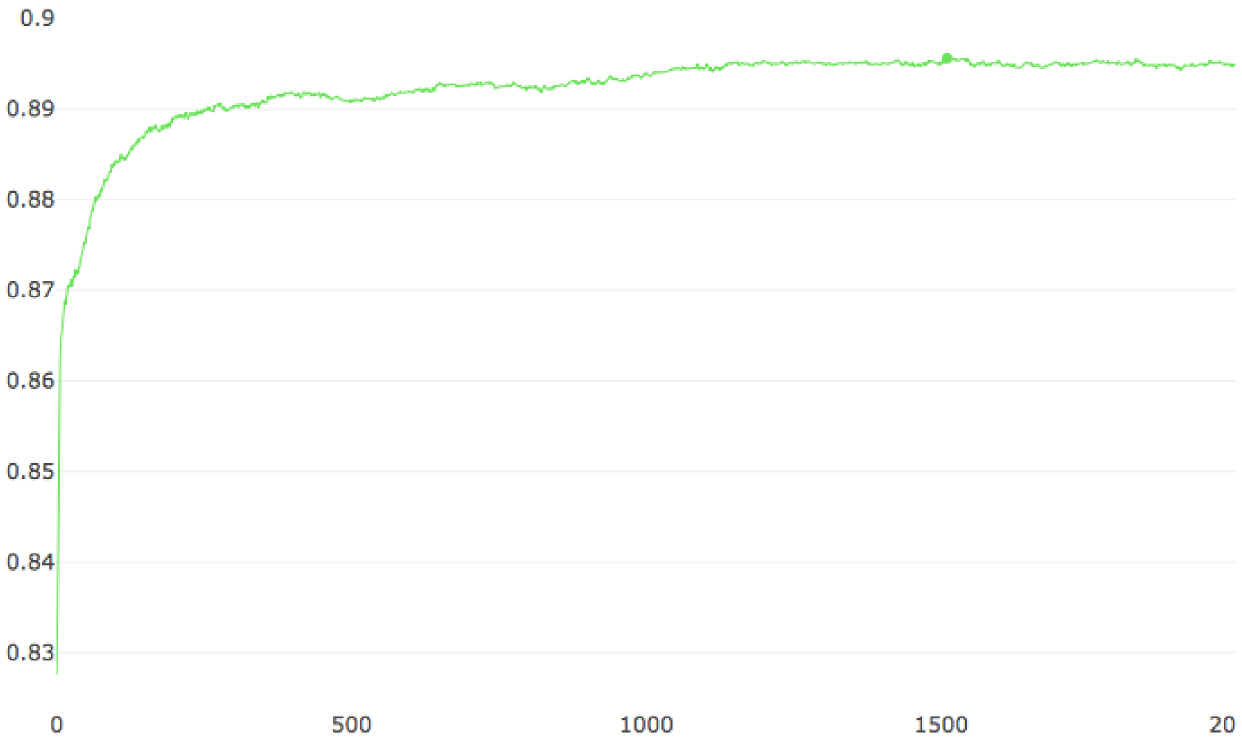


PairLogitPairwise NDCG Visualization

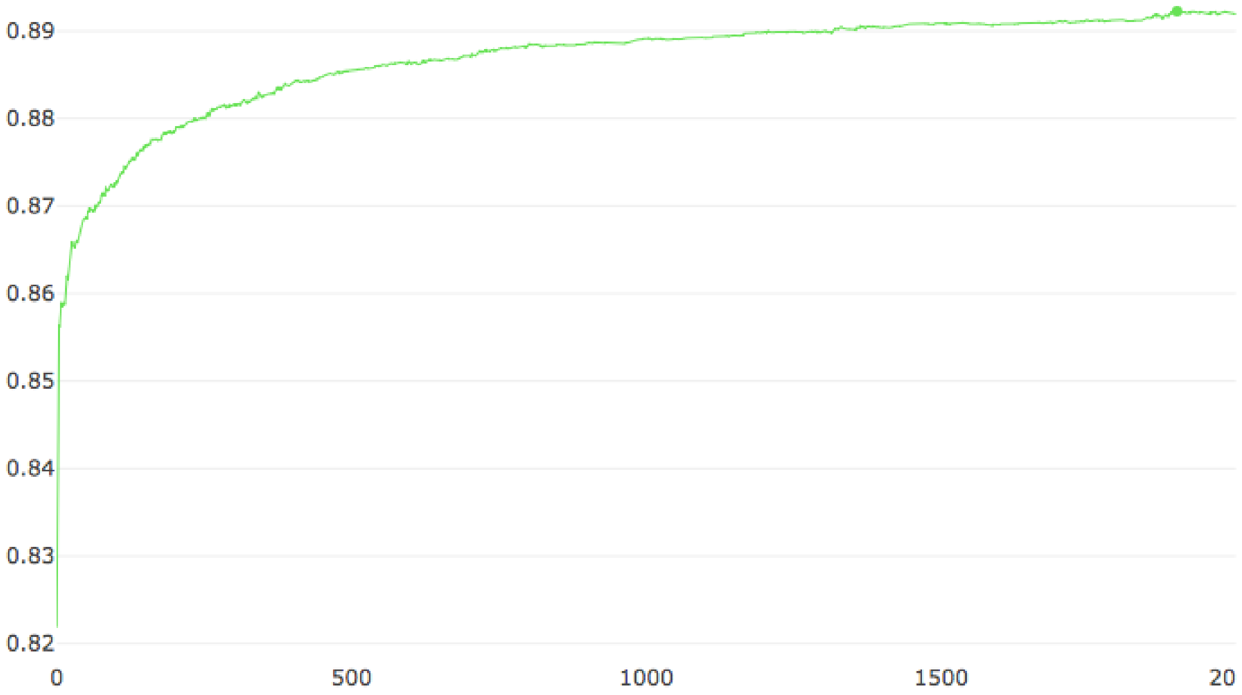


Zhamilya Saparova

YetiRank NDCG Visualization

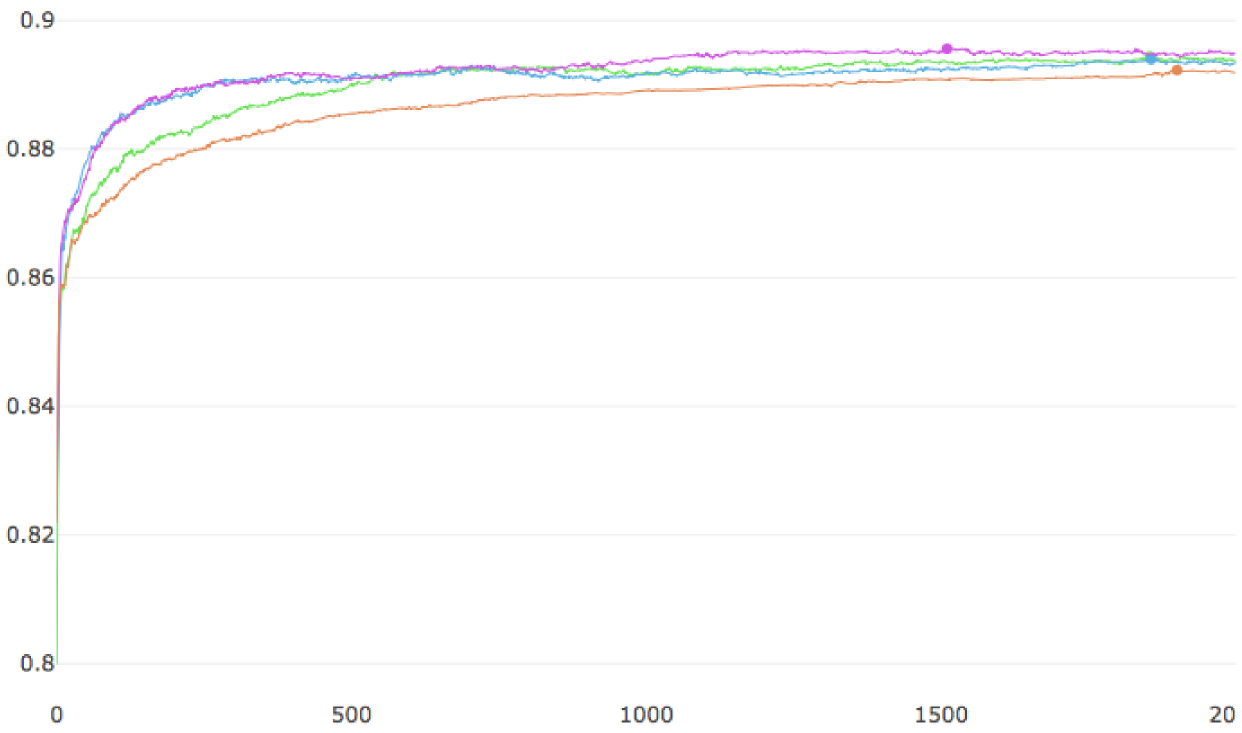


YetiRankPairwise NDCG Visualization

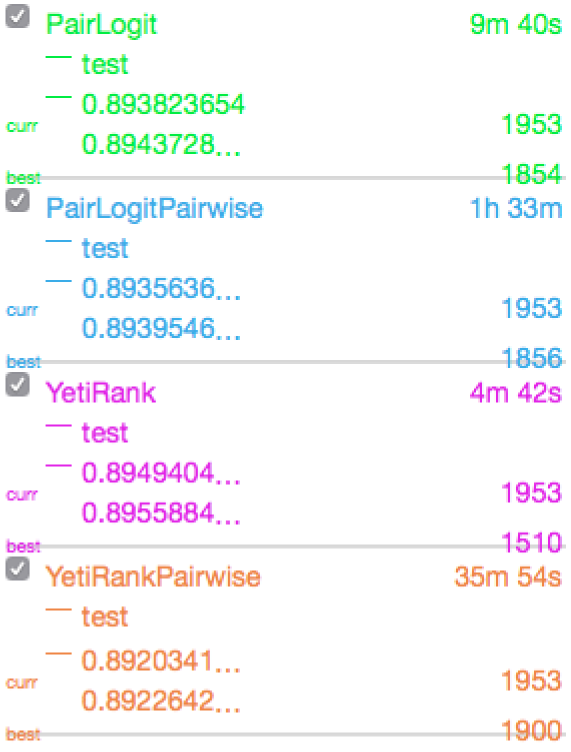


Zhamilya Saparova

Using Metric Visualizer NDCG ('PairLogit' green, 'PairLogitPairwise' blue, 'YetiRank' purple, 'YetiRankPairwise' orange)



Below can be seen time taken for each CatBooster method and test best accuracies.



☑ PairLogit 9m 40s
— test
— 0.893823654
curr 1953
0.8943728...
best 1854

☑ PairLogitPairwise 1h 33m
— test
— 0.8935636...
curr 1953
0.8939546...
best 1856

☑ YetiRank 4m 42s
— test
— 0.8949404...
curr 1953
0.8955884...
best 1510

☑ YetiRankPairwise 35m 54s
— test
— 0.8920341...
curr 1953
0.8922642...
best 1900