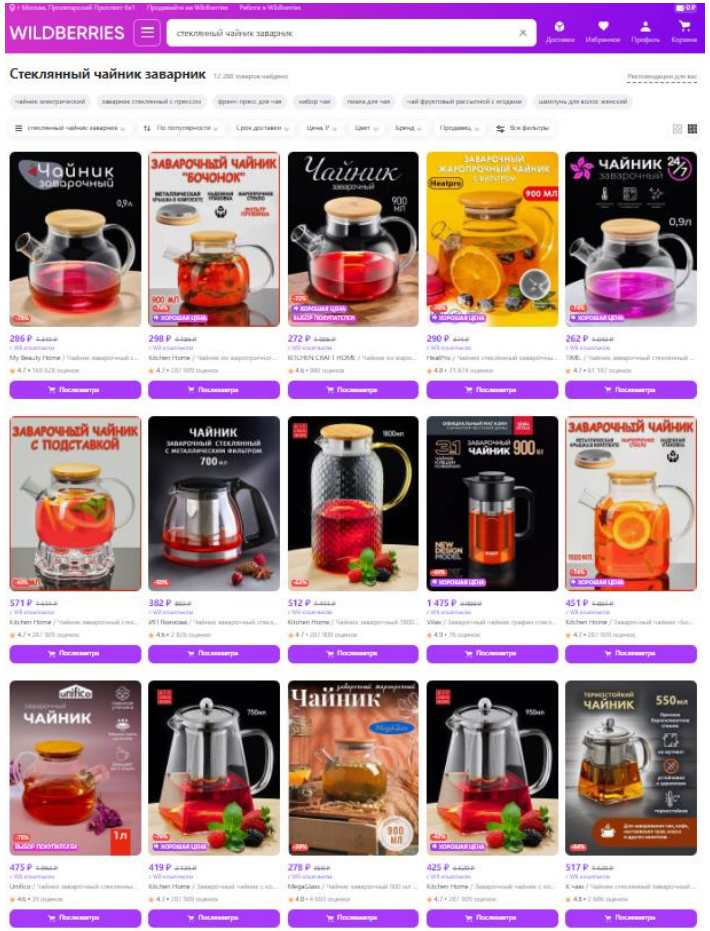
ПРОЕКТ **"РАНЖИРОВАНИЕ В ПОИСКЕ"**

АФАНАСЬЕВ ЛЕОНИД

ЦЕЛИ И ЗАДАЧИ

Ранжирование на WB — это упорядочивание товаров, которое помогает выйти на первые позиции товарам хорошего качества, которые наиболее релевантные запросу в поисковой строке, с наименьшими сроками доставки и от добросовестного продавца.

- Формализации задачи и анализа EDA
- Проработка вариантов решения
- Baseline
- Оптимизация решения



ВАЖНОСТЬ РАНЖИРОВАНИЯ

- **Улучшенный пользовательский опыт:** Алгоритмы LTR направлены на то, чтобы лучше понимать предпочтения и намерения пользователей, что приводит к более релевантным и персонализированным результатам.
- **Повышение вовлеченности и коэффициента конверсии:** Предоставляя пользователям более релевантный контент или рекомендации по продуктам, компании могут повысить вовлеченность пользователей и коэффициент конверсии.
- **Конкурентное преимущество:** обеспечение превосходного пользовательского опыта может стать ключевым отличием от конкурентов.
- **Удержания клиентов:** Когда пользователи постоянно находят релевантный и ценный контент, это может привести к более высоким показателям удержания клиентов и повышению лояльности клиентов с течением времени.
- **Адаптивность и масштабируемость:** Алгоритмы LTR могут адаптироваться к изменяющемуся поведению и предпочтениям пользователей с течением времени, что делает их масштабируемыми решениями для предприятий любого размера.

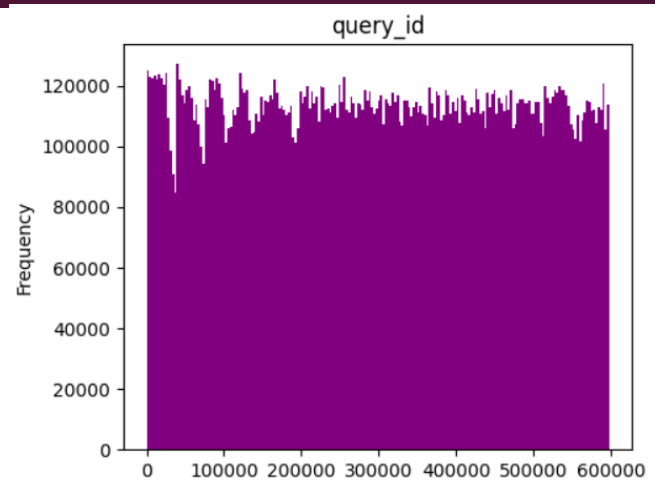
EDA

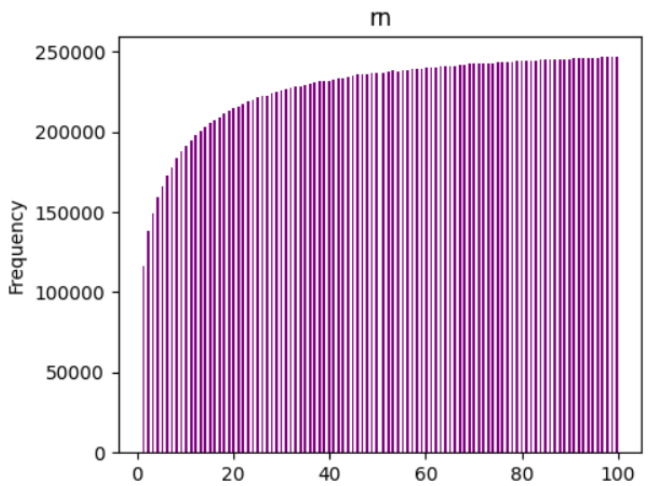
query_id	report_date	target	rn	feature_1	feature_2	feature_3	feature_4	feature_5	feature_6	feature_7	feature_8	feature_9	feature_10	feature_11
u32	datetime[ns]	bool	i64	f64	f64	f64	f64	f64	f64	f64	f64	f64	f64	f64
2	2024-01-11 00:00:00	false	4	0.693	0.1	0.245833	3.376496	0.7777	0.0	2.87166	0.97	1.0	4.6	-0.196707
2	2024-01-11 00:00:00	false	5	0.281	0.1	0.108333	2.909165	0.7882	0.0	1.978092	0.96	0.89	4.8	0.034091
2	2024-01-11 00:00:00	false	8	0.319	0.183333	0.154167	1.725731	0.688	0.5	2.083885	0.95	0.9	4.7	0.286658
2	2024-01-11 00:00:00	false	10	0.281	0.1	0.108333	3.759698	0.7882	1.0	1.898608	0.98	0.81	4.8	0.052632
2	2024-01-11 00:00:00	false	12	0.281	0.1	0.108333	0.026253	0.7055	0.0	0.317658	0.96	0.85	4.7	0.172424

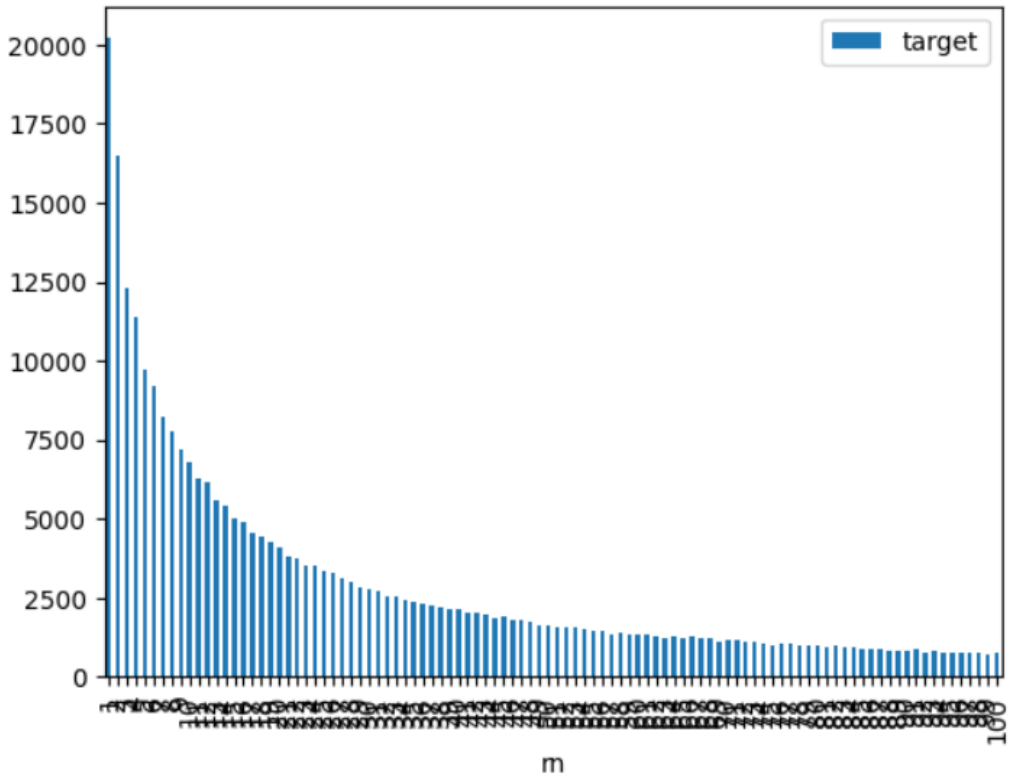
Мы обнаружили следующее:

1. Невысокая доля верных подборов: **1,3%** верных против **98,7%** неверных
2. В датасете имеются 15 признаков: query_id и rn - целочисленные, report_date - время, target - логическая, остальные вещественные.
3. Пропуски имеются в feature_5 и feature_8, 1.4% и 34% соответственно. При этом распределение верных/ложных ответов значительно отличается 0,2% и 0,5% соответственно.
4. Имеются данные с 11 января по 25 января 2024 года- В данных нет дубликатов

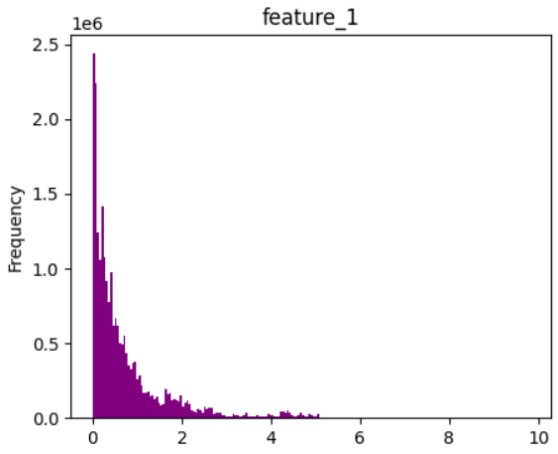
EDA

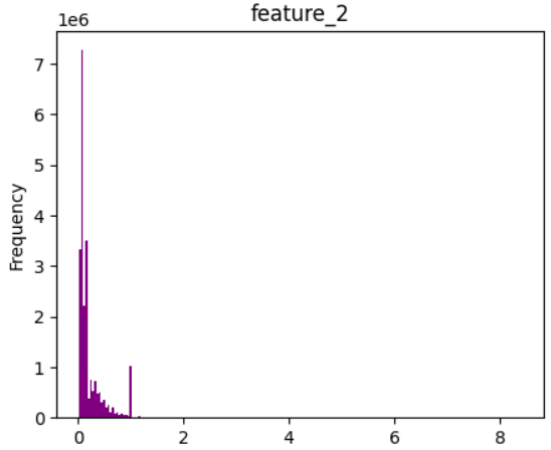

query_id
Frequency
120000
100000
80000
60000
40000
20000
0
0
100000
200000
300000
400000
500000
600000

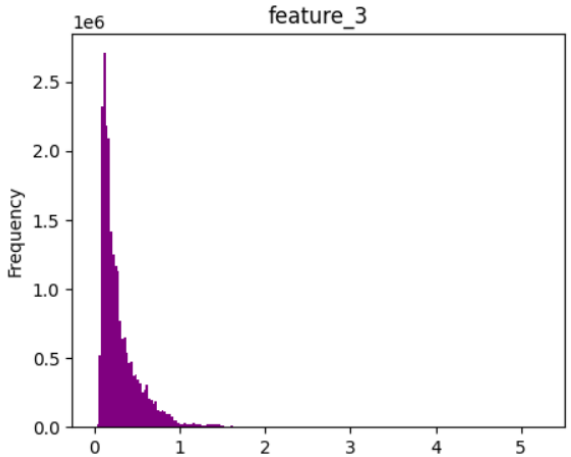

m
Frequency
250000
200000
150000
100000
50000
0
0
20
40
60
80
100

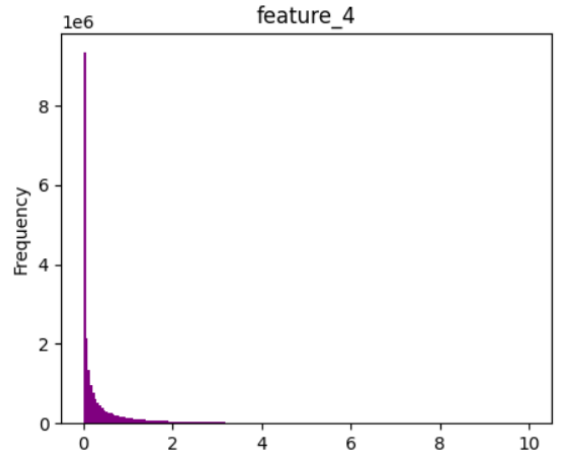

target
20000
17500
15000
12500
10000
7500
5000
2500
0
m

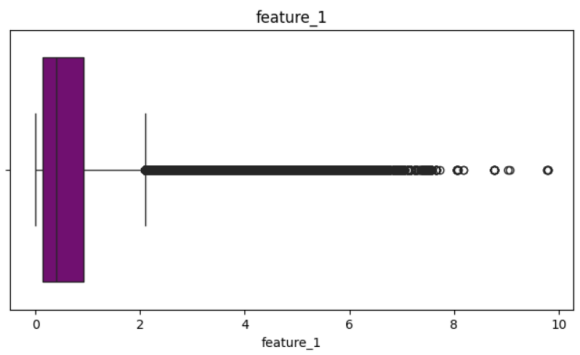
EDA

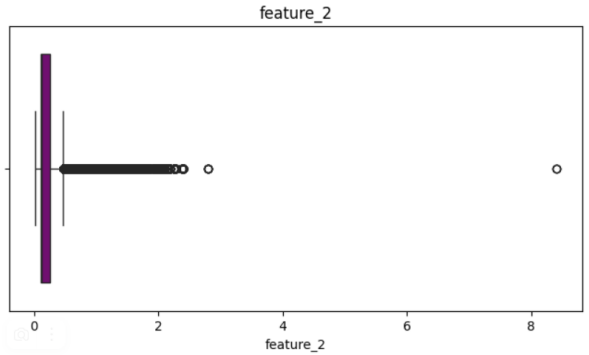

feature_1
1e6
Frequency

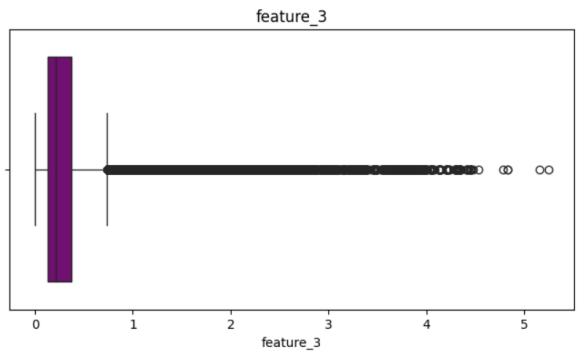

feature_2
1e6
Frequency

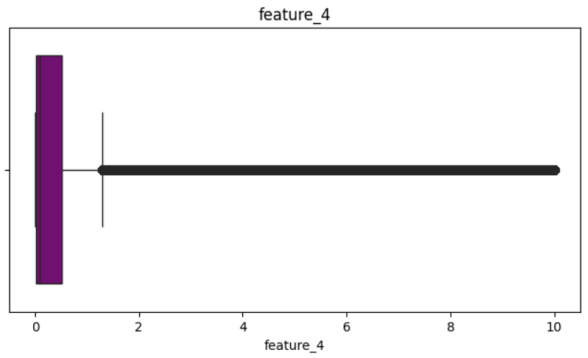

feature_3
1e6
Frequency


feature_4
1e6
Frequency

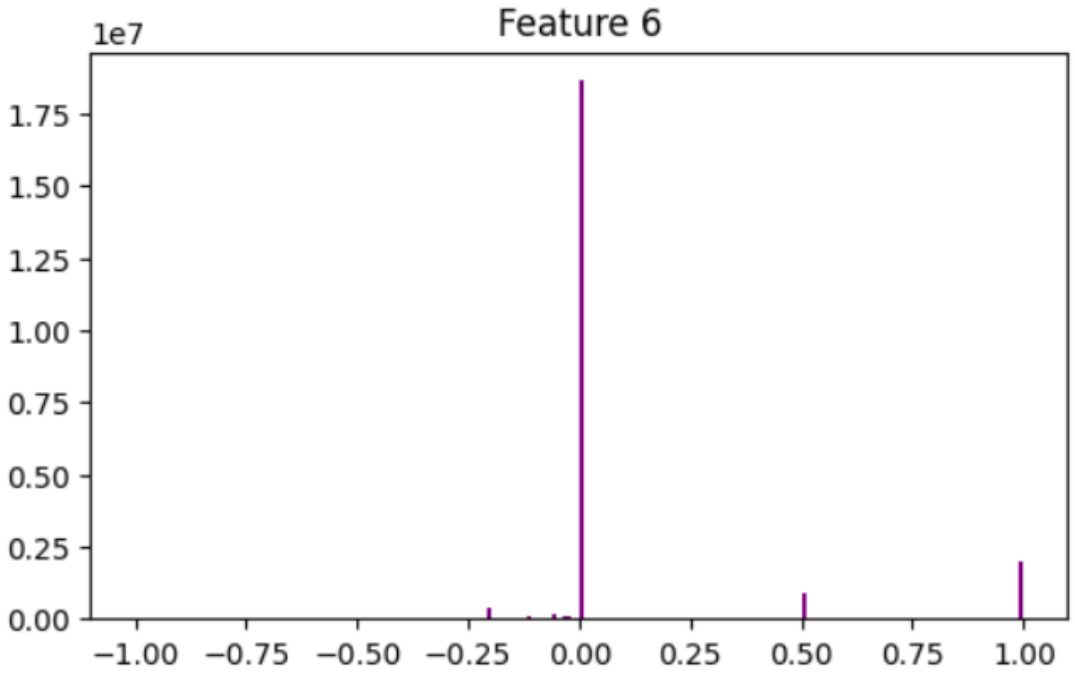

feature_1

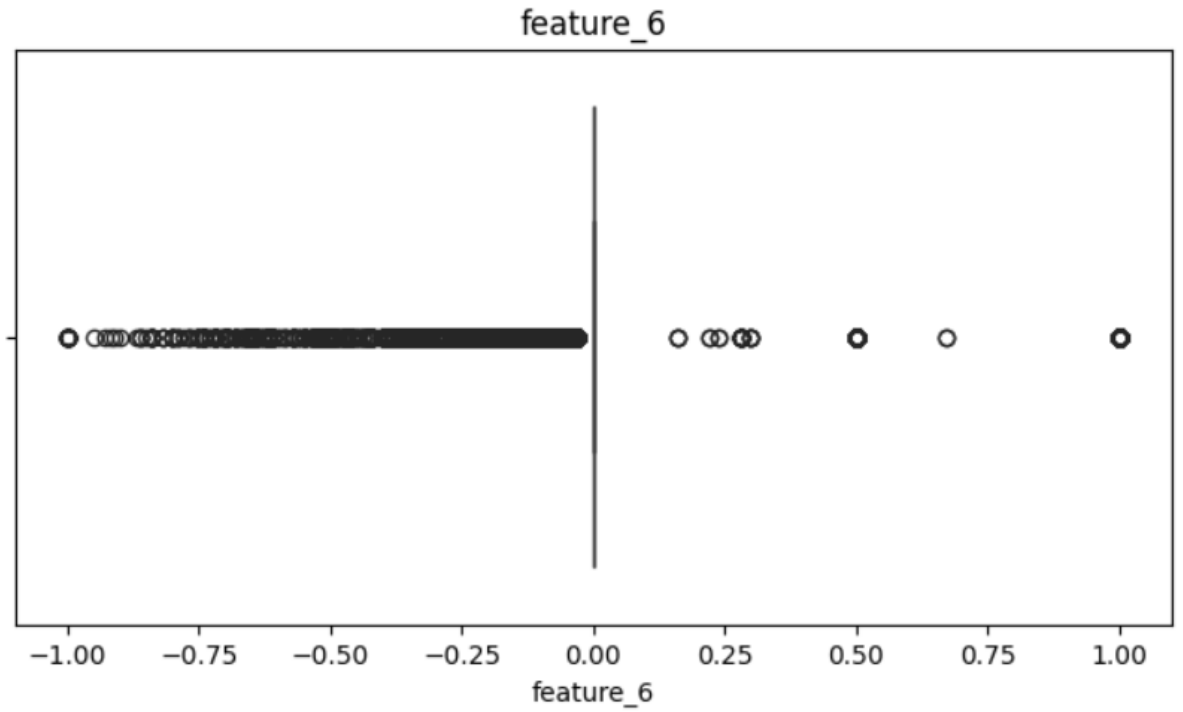

feature_2


feature_3

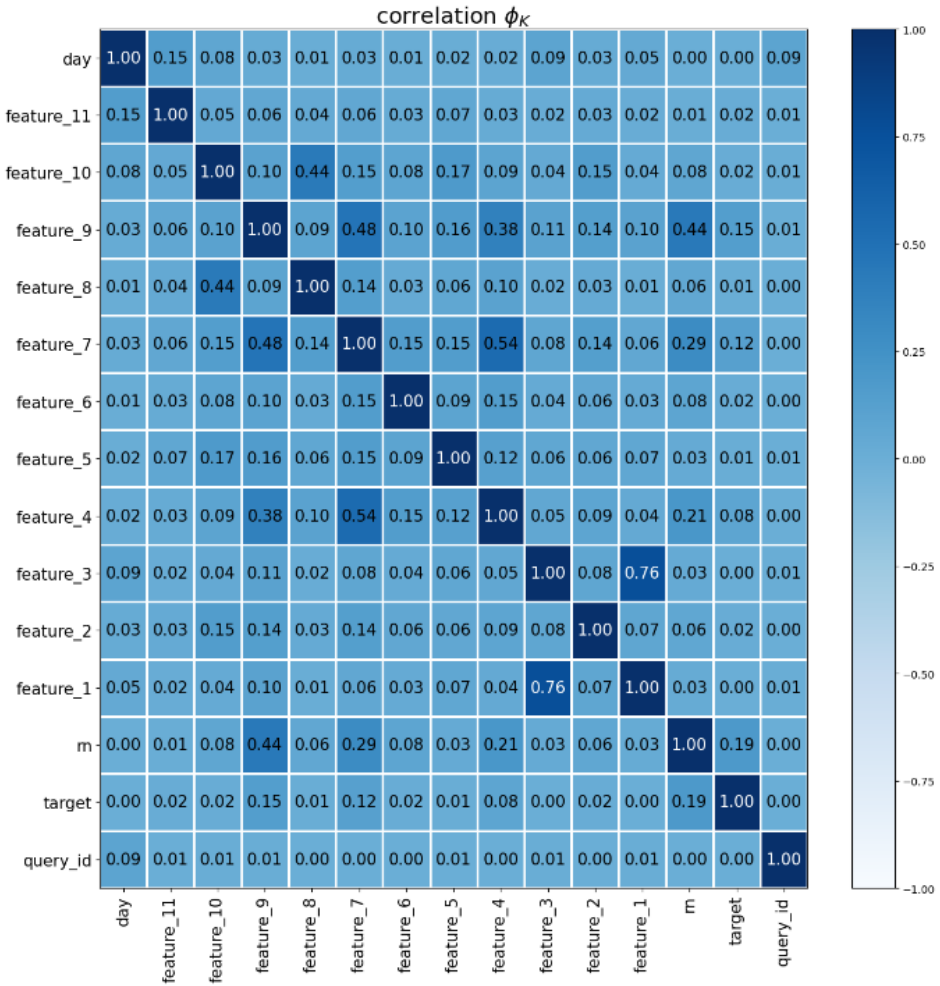

feature_4

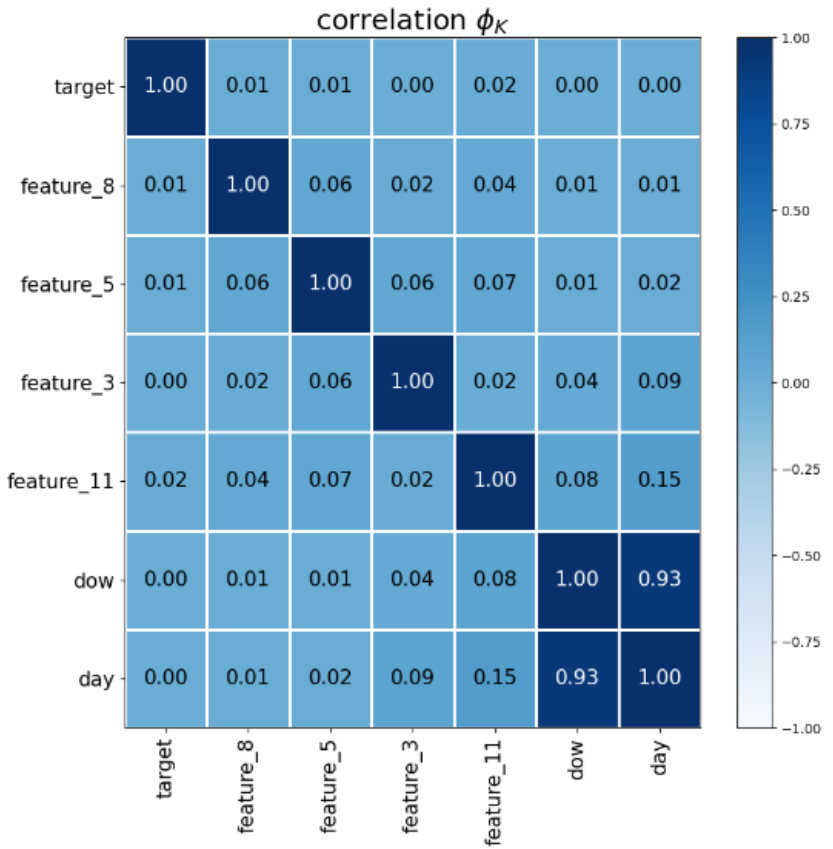
EDA


Feature 6
1e7
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


feature_6
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
feature_6

EDA

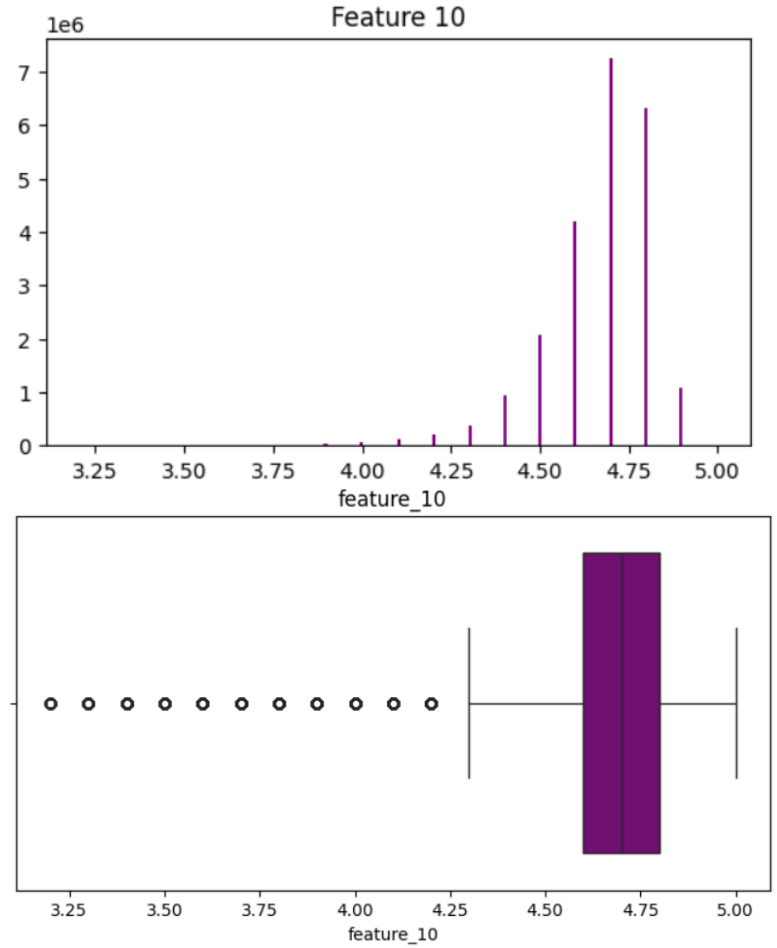

correlation ϕ_K
day 1.00 0.15 0.08 0.03 0.01 0.03 0.01 0.02 0.02 0.09 0.03 0.05 0.00 0.00 0.09
feature_11 0.15 1.00 0.05 0.06 0.04 0.06 0.03 0.07 0.03 0.02 0.03 0.02 0.01 0.02 0.01
feature_10 0.08 0.05 1.00 0.10 0.44 0.15 0.08 0.17 0.09 0.04 0.15 0.04 0.08 0.02 0.01
feature_9 0.03 0.06 0.10 1.00 0.09 0.48 0.10 0.16 0.38 0.11 0.14 0.10 0.44 0.15 0.01
feature_8 0.01 0.04 0.44 0.09 1.00 0.14 0.03 0.06 0.10 0.02 0.03 0.01 0.06 0.01 0.00
feature_7 0.03 0.06 0.15 0.48 0.14 1.00 0.15 0.15 0.54 0.08 0.14 0.06 0.29 0.12 0.00
feature_6 0.01 0.03 0.08 0.10 0.03 0.15 1.00 0.09 0.15 0.04 0.06 0.03 0.08 0.02 0.00
feature_5 0.02 0.07 0.17 0.16 0.06 0.15 0.09 1.00 0.12 0.06 0.06 0.07 0.03 0.01 0.01
feature_4 0.02 0.03 0.09 0.38 0.10 0.54 0.15 0.12 1.00 0.05 0.09 0.04 0.21 0.08 0.00
feature_3 0.09 0.02 0.04 0.11 0.02 0.08 0.04 0.06 0.05 1.00 0.08 0.76 0.03 0.00 0.01
feature_2 0.03 0.03 0.15 0.14 0.03 0.14 0.06 0.06 0.09 0.08 1.00 0.07 0.06 0.02 0.00
feature_1 0.05 0.02 0.04 0.10 0.01 0.06 0.03 0.07 0.04 0.76 0.07 1.00 0.03 0.00 0.01
m 0.00 0.01 0.08 0.44 0.06 0.29 0.08 0.03 0.21 0.03 0.06 0.03 1.00 0.19 0.00
target 0.00 0.02 0.02 0.15 0.01 0.12 0.02 0.01 0.08 0.00 0.02 0.00 0.19 1.00 0.00
query_id 0.09 0.01 0.01 0.01 0.00 0.00 0.00 0.01 0.00 0.01 0.00 0.01 0.00 0.00 1.00
day feature_11 feature_10 feature_9 feature_8 feature_7 feature_6 feature_5 feature_4 feature_3 feature_2 feature_1 m target query_id
1.00 0.75 0.50 0.25 0.00 −0.25 −0.50 −0.75 −1.00


correlation ϕ_K
target 1.00 0.01 0.01 0.00 0.02 0.00 0.00
feature_8 0.01 1.00 0.06 0.02 0.04 0.01 0.01
feature_5 0.01 0.06 1.00 0.06 0.07 0.01 0.02
feature_3 0.00 0.02 0.06 1.00 0.02 0.04 0.09
feature_11 0.02 0.04 0.07 0.02 1.00 0.08 0.15
dow 0.00 0.01 0.01 0.04 0.08 1.00 0.93
day 0.00 0.01 0.02 0.09 0.15 0.93 1.00
target feature_8 feature_5 feature_3 feature_11 dow day
1.00 0.75 0.50 0.25 0.00 −0.25 −0.50 −0.75 −1.00

EDA

Обнаружено следующее:

- Невысокая доля верных подборов: 1,3% верных против 98,7% неверных
- Имеются данные с 1 (четверг)1 января по 25 января 2 для обучающей выборки. С 26 января по 28 января 2024 года для тестовой выборки.
- В данных нет дубликатов
- В столбце Feature 2 найдены выбросы 9 значений больше 2.4. И все они относятся к ложной выдачи. Эти значения отрицательного класса удалены, так как они искажают данные перекосом в большую сторону
- В распределении 5 признака выбивается пик в 0.87. Его природа неясна. Пик не связан с датой
- Feature 6 распределен не равномерно, 83% значений признака = 0. При этом соотношение классов целевого признака разное. Для 0, 1.2%; меньше 0, 1.4%; больше 0, 1.8%
- Ящик с усами показал выбросы Feature 10 ниже 3.6. Однако барплот показал, что такое распределение связано с дискретностью признака
- Таблица корреляций показала, что rn коррелирует feature_9, feature_7, feaure_4, target. Возможно, feature_9 - цена товара
- Так же имеется высокая корреляция между feature_1 и feature_3, feature_4 и feature_7.- feature 11 коррелирует с днем, причина пока также не ясна



ПРОБЛЕМАТИКА

- При поиске товаров на маркетплейсе, покупатель располагает ограниченным количеством времени и терпении (концентрации, интереса и пр.). Поэтому главной целью рандирования является выдача товаров наиболее релевантных запросы. Эти рассуждения идеализированы, так как целью маркетплейса может быть также размещение рекламы с целью заинтересовать схожей категорией товаров или же продвинуть продавцов с привилегиями и пр.

```python
def dcg_at_k(relevance, k):
    relevance = np.asarray(relevance)[:k]
    gains = 2 ** relevance - 1
    discounts = np.log2(np.arange(len(relevance)) + 2)
    return np.sum(gains / discounts)

def ndcg_at_k(relevance, k):
    ideal_dcg = dcg_at_k(sorted(relevance, reverse=True), k)
    if ideal_dcg == 0:
        return 0
    return dcg_at_k(relevance, k) / ideal_dcg
```

- Однако, нашей задаче целью является именно выдача наиболее подходящих товаров. Для того, чтобы иметь объективный показатель, необходимо ввести метрику.

$$NDCG_{@K} = \frac{DCG_{@K}}{IDCG_{@K}}$$

$$IDCG_{@K} = \sum_{i=1}^{K^{ideal}} \frac{G_i^{ideal}}{log_2(i+1)}$$

АНАЛИЗ ДОСТУПНЫХ РЕШЕНИЙ

- **LambdaMART:**

LambdaMART - популярный алгоритм на основе градиентного бустинга деревьев, специально разработанный для задач LTR. Он оптимизирует ранжирование непосредственно путем минимизации функции потерь, учитывающей релевантность пар или групп документов.

- **RankNet:**

RankNet - алгоритм на основе нейронных сетей, который обучается ранжированию, прямо оптимизируя парный порядок документов. Он оценивает вероятность того, что один документ будет ранжирован выше другого.

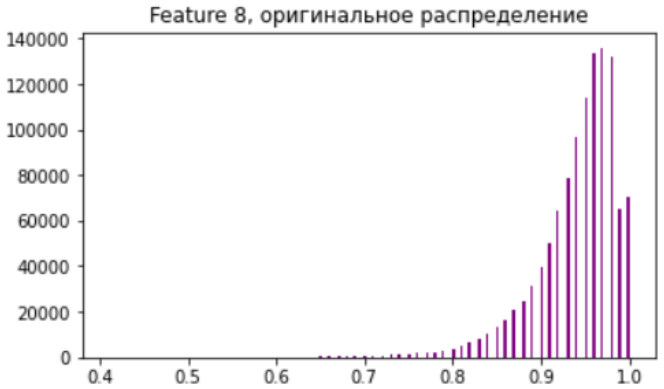
- **ListNet:**

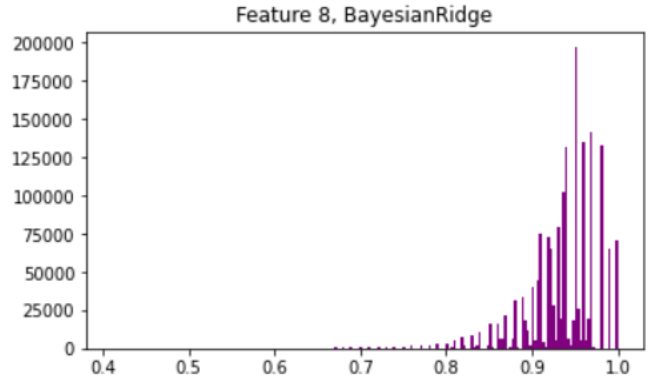
ListNet - также алгоритм на основе нейронных сетей, который обучается ранжированию, однако, в отличии от RankNet работает со всем списком.

RANKNET И LISTNET. ЗАПОЛНЕНИЕ ПРОПУСКОВ

IterativeImputer

Было показано, что заполнение пропусков средним не лучшая стратегия. Поэтому заполним данные используя IterativeImputer - sklearn imputer. KNN является предпочтительным, однако требует больших ресурсов. В работе рассмотрено заполнение Баесом и Лесом.



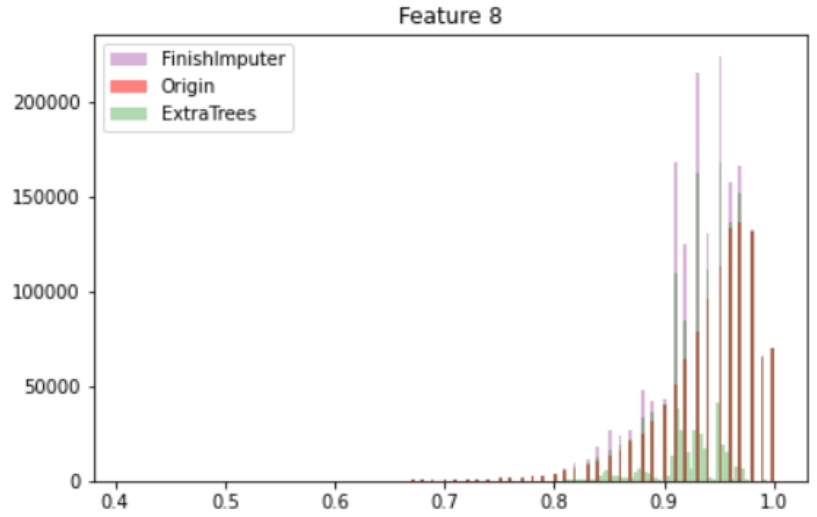


RANKNET И LISTNET. ЗАПОЛНЕНИЕ ПРОПУСКОВ

Важное замечание! Из распределений пропусков признаков 5 и 8, видно, что признак 8 дискретный - 59. У 5 признака дискретность - 3541. Поэтому приведем дискретность 8 признака к такой же оригинальной дискретности.

```python
def find_nearest_value(arr, value):
    """Находим ближайшее значение для элемента из array в value."""
    idx = np.abs(arr - value).argmin()
    return arr[idx]

def replace_with_nearest(series1, series2):
    """меняем значения на ближайшее"""
    replaced_values = [find_nearest_value(series2.values, val) for val in series1.values]
    return pd.Series(replaced_values, index=series1.index)
```



Вывод: полученные данные показывают смещение. Однако, 30% пустые, поэтому распределение так сильно различается от оригинального

RANKNET

Вывод: Модель RankNet результат лучше чем логистическая регрессия. Подбор параметров на кросс валидации показал, что модель особенно чувствительная к lr. Увеличение к lr привело к падению метрик.

	lr	hidden_dim	ndcg_4	ndcg_8	ndcg_12	p_at_1	p_at_4	p_at_12	rr	map_at_1	map_at_4	map_at_12	calculation time
1	0.001	22	0.255360	0.300356	0.327094	0.154388	0.342983	0.560665	0.277314	0.154388	0.231341	0.259217	0 days 00:11:05.436759
2	0.001	33	0.252638	0.298897	0.325120	0.152902	0.338968	0.558275	0.275607	0.152902	0.229135	0.257423	0 days 00:10:57.115775
3	0.005	22	0.249910	0.294443	0.320856	0.151800	0.334049	0.548930	0.273063	0.151800	0.227000	0.254832	0 days 00:11:00.256307
4	0.005	33	0.252309	0.296946	0.323261	0.151800	0.338821	0.553668	0.274459	0.151800	0.228479	0.256209	0 days 00:11:04.658356
5	0.015	22	0.232059	0.279650	0.306610	0.132529	0.317393	0.543135	0.256471	0.132529	0.208645	0.237867	0 days 00:11:05.685582
6	0.015	33	0.250660	0.295486	0.320603	0.152516	0.334228	0.545608	0.274090	0.152516	0.228181	0.255238	0 days 00:11:12.137379

LISTNET

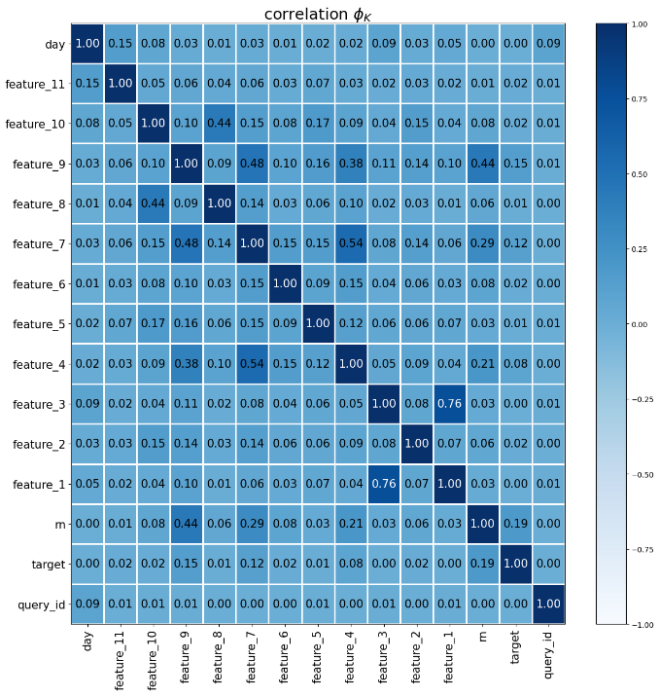
Вывод: ListNet показал прирост показателей по сравнению с бейзлайном и RankNet, особенно, в точности. Недостатком является время обучения, оно учеличилось в 1,5 раза до 15 минут.Лучшим сочетанием параметров является hidden_dim = 33 и lr = 0.001. Остановимся на этой моделе

	lr	hidden_dim	ndcg_4	ndcg_8	ndcg_12	p_at_1	p_at_4	p_at_12	rr	map_at_1	map_at_4	map_at_12	calculation time
1	0.0005	22	0.260020	0.308742	0.335424	0.157031	0.348434	0.576925	0.282778	0.157031	0.235762	0.265041	0 days 00:15:47.872891
2	0.0005	33	0.259892	0.308798	0.334994	0.158132	0.347520	0.574697	0.283185	0.158132	0.236091	0.265211	0 days 00:16:01.208038
3	0.0010	22	0.259117	0.308641	0.334359	0.157031	0.346896	0.574223	0.282173	0.157031	0.234959	0.264456	0 days 00:15:59.267406
4	0.0010	33	0.260125	0.309891	0.335882	0.157362	0.347704	0.576448	0.283376	0.157362	0.236070	0.265574	0 days 00:15:34.388594
5	0.0100	22	0.253550	0.296140	0.310861	0.151856	0.340064	0.509008	0.266450	0.151856	0.229897	0.253189	0 days 00:15:40.245284
6	0.0100	33	0.254110	0.296984	0.310969	0.153067	0.339959	0.506838	0.267301	0.153067	0.231009	0.254057	0 days 00:15:43.196118

LISTNET

Рассмотрим важность признаков. Так как используется Relu, то под важностью будем рассматривать сумму весов для каждого признака.

	feature_1	feature_2	feature_3	feature_4	feature_6	feature_7	feature_9	feature_10	feature_11	feature_8imp	feature_5imp
0	0.044984	0.080929	0.008144	0.007735	0.003126	0.014163	0.076655	0.042473	0.041455	0.024689	0.016852



ИТОГОВЫЙ ВЫВОД

В ходе поиска решения были реализованы на PyTorch и выбраны 2 основных подхода ListNet и RankNet: Вариация и подбор гиперпараметров показал, что модели подвержены переобучению. Так при не правильном выборе lr и количестве слоев, модели показывали худший вариант чем бейзлайн и наоборот, при 1 эпохе достигали лучшего результата (что, в целом, не ошеломляющая новость). Модель ListNet показала лучший результат, почти со всеми гиперпараметрами, с помощью кросс валидации были найдены оптимальные гиперпараметры. Важность признаков следующая для модели ListNet. Как ранее было показано на матрице phi-корреляции, feature_9 имеет самую высокую корреляцию, что и на таблице важности признаков наблюдается.

	4
lr	0.001
hidden_dim	33
ndcg_4	0.260125
ndcg_8	0.309891
ndcg_12	0.335882
p_at_1	0.157362
p_at_4	0.347704
p_at_12	0.576448
rr	0.283376
map_at_1	0.157362
map_at_4	0.23607
map_at_12	0.265574