В нашем проекте рассматривается задача ранжирования с помощью моделей машинного обучения. Выбранный нами датасет представляет собой выпущенный Microsoft крупномасштабный набор данных MSLR-WEB30K с 30 000 запросами (https://www.microsoft.com/en-us/research/project/mslr/) для обучения ранжированию. Набор данных состоит из векторов признаков и меток оценок релевантности, где:

- Столбец '0' - содержит целые числа от 0 до 4, отражающие релевантность документа запросу (где 0 - неактуальный, 4 - идеально релевантный)
- Столбец '1' имеет форму 'qid:int', идентифицирующую запрос.
- Столбцы '2' - '137' содержат признаки для пары запрос-документ в форме 'feature_id:feature_value'.

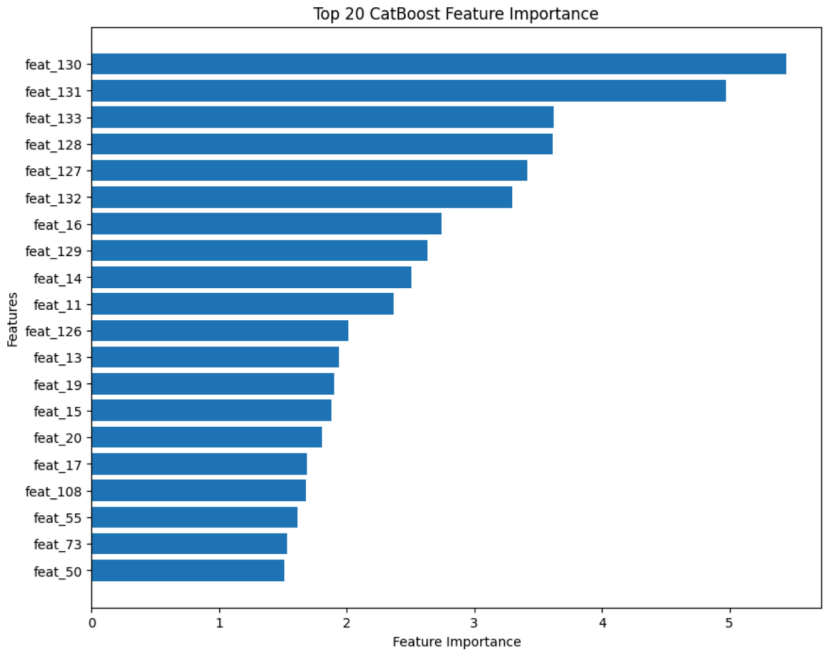
Обучающий набор используется для обучения моделей ранжирования. Валидационный набор используется для настройки гиперпараметров алгоритмов обучения. Тестовый набор используется для оценки эффективности полученных моделей ранжирования.

Предобработка данных

Для начала все данные в столбцах были отформатированы, так как изначально значения признаков записывались в формате: "номер/название столбца: значение признака", и приведены к формату числа с плавающей точкой (float). Также из датасета были удалены все неинформативные признаки, содержащие пропущенные значения. В связи с большим разбросом значений признаков, пролоогорифмируем их. Далее с помощью CatBoostClassifier было найдено 20 лучших признаком для дальнейшей работы, число признаков обоснованно другими работами с подобным датасетом (выбирались в среднем 10-20 лучших из 136). Алгоритм CatBoostClassifier был выбран в связи с большой коррелированностью некоторых из признаков.Благодаря деревьям решений устойчив к мультиколлинеарности. Если признаки сильно коррелируют, модель просто выберет наиболее информативный признак. В топ-20 признаков попали признаки с номерами:

№ признака	Название	Значение
130	Уровень страницы	Оценивает центральность данной веб-страницы на основе веб-ссылок в Интернете. Это обеспечивает успех Google
131	Уровень сайта	Например, «ucsb.edu/pstat» и «ucsb.edu/math» имеют одинаковый уровень.
133	Оценка качества веб-страницы.	Оценка выводится классификатором качества веб-страниц, который измеряет степень плохости веб-страницы.
128	Inlink number	Количество веб-страниц, цитирующих эту веб-страницу.

127	Длина URL	Число символов в URL-адресе
132	Оценка качества веб-страницы.	Оценка выводится классификатором качества веб-страницы.
16	LMIR.DIR для тела	Подход к языковой модели для IRс байесовским сглаживанием с использованием приращений Дирихле.
129	Outlink number	Сколько веб-страниц цитирует этот сайт.
14	Длина потока	Длина URL адреса
11	Длина тела	Длина тела текста
126	Число слэшей в URL адресе	Число слэшей в URL адресе
13	Длина заголовка	Длина заголовка
19	IDF URL	1, деленное на количество документов, содержащих термины запроса.
15	Длина документа	Длина всего документа
20	IDF всего документа	1, деленное на количество документов, содержащих термины запроса.
17	IDF содержимого тега гиперссылки	1, деленное на количество документов, содержащих термины запроса.
108	BM25	Okapi BM25
55	Минимальная длина потока, нормированная на частоту терминов	Минимальная длина потока, нормированная на частоту терминов
73	Сумма TF*IDF заголовка	Сумма произведения количества терминов и IDF для каждого запроса
50	Сумма длины потока, нормированная на частоту терминов для целого документа	Сумма длины потока, нормированная на частоту терминов



NDCG (Normalized Discounted Cumulative Gain)

В данном проекте используется метрика нормализованной дисконтированной кумулятивной суммы. Она оценивает качество ранжирования с учетом позиции релевантных документов. Более высоко релевантные результаты, находящиеся ближе к началу списка, получают больший вес.

Особенности:

- Важно не просто найти релевантные документы, но и ранжировать их так, чтобы наиболее полезные документы находились в топе.
- Используется в поисковых системах и задачах информационного поиска.

$$nDCG@k = \frac{DCG@k}{Ideal\ DCG@k},\ nDCG \in [0,1]$$

Поточечный (pointwise) подход к ранжированию

В этом подходе у нас известны некоторые оценки релевантности каждой пары запрос-документ, и модель учится предсказывать эти оценки. Взаимоотношения между разными документами внутри списка всех документов не рассматриваются.

Функция ошибки по конкретному объекту:

$$\sum_{q,j} I(f(x_j^q),\ r_j^q) \to min$$

Особенность: функция потерь рассчитывается между результатом вычисления $f(x_j^q)$ и мерой релевантности r_j^q.

Модель логистической регрессии

Алгоритм построения модели:

Шаг 1: Определим функцию, принимающую на вход DataFrame и извлекающая из него признаки (features) и целевые значение (labels).
Шаг 2: Разделим набор данных на обучающий и тестовый.
Шаг 3: Выполним масштабирования данных с помощью StandartScaler.
Шаг 4: Обучим модель логистической регрессии с многоклассовой поддержкой.
Шаг 5: Оценка метрики NDCG

В итоге получили на тестовой выборке следующие значения:

```
NDCG@10: 0.1748
NDCG@5: 0.1593
```

Попарный (pairwise) подход к ранжированию
Попарный подход к ранжированию — это метод, который основывается на сравнении объектов парами. При использовании этого метода рассматриваются тройки (q, d1, d2), где q - запрос, а d1, d2 - документы, и каждая пара документов сравнивается друг с другом, чтобы определить, какой из них предпочтительнее. Например, в обучающей выборке есть два документа, и нам известно, какой из них более релевантный по данному запроса. Тогда мы будем штрафовать модель, если она более релевантному поставила прогноз ниже, чем менее релевантному, то есть неправильно сранжировала пару.

Функция ошибки по паре объектов:

$$\sum_{q} \sum_{i,j:r_i^q > r_j^q} I(f(x_i^q) - f(x_j^q)) \rightarrow min$$

CatBoost
CatBoost использует решающие деревья глубины 1 или 2 в качестве базовых моделей. Эти неглубокие деревья, которые иногда называют "котэ", имеют следующие характеристики:

- Каждый узел дерева делает бинарное разбиение на основе значения одной из признаков (К примеру мы решаем задачу классификации, где необходимо определить, будет ли клиент покупать продукт (1) или нет (0). Один из наших признаков - возраст клиента. Решающее дерево CatBoost может разделить клиентов на две группы: те, кто моложе 30 лет, и те, кто старше 30 лет)
- Эти короткие деревья обладают небольшой глубиной, что делает их более устойчивыми к переобучению.
- Способны обрабатывать данные “из коробки”, без предварительного анализа
- Используется повышение градиента — процесс, в котором итеративно строится множество деревьев решений. Каждое последующее дерево улучшает результат предыдущего, что приводит к лучшим результатам.

Особенностью применение алгоритма CatBoost является необходимость создания пулов - классов, в которых содержатся данные, в конструктор pool принимает разные параметры. В качестве функции потерь применялась функция YetiRankPairwise. Метрика, которая записывается в выходные данные при оптимизации YetiRank, зависит от диапазона всех N целевых значений набора данных:

$$target_i \in [0;1] - PFound$$

$$target_i \notin [0;1] - NDCG$$

При обучении модели рассматривались несколько значений различных гиперпараметров. Итоговое значение NDCG на тестовой выборке в несколько раз превышает значения NDCG других моделей:

```
CatBoost Pairwise Ranker Model NDCG@5 Score: 0.8787
CatBoost Pairwise Ranker Model NDCG@10 Score: 0.7333
```

Экстремальный градиентный бустинг

В качестве одной из реализаций попарного подхода к ранжированию мы рассмотрим модель XGBoost (экстремальный градиентный бустинг). В основе XGBoost лежит алгоритм градиентного бустинга деревьев решений. Градиентный бустинг — это техника машинного обучения для задач классификации и регрессии, которая строит модель предсказания в форме ансамбля слабых предсказывающих моделей, обычно деревьев решений.

Алгоритм построения модели

Шаг 1: Определим функцию, принимающую на вход DataFrame и извлекающая из него признаки (features) и целевые значение (labels).

Шаг 2: Определим функцию, принимающую на вход список идентификаторов запросов и подсчитывающую количество повторений каждого уникального идентификатора в наборе данных.

Шаг 3: Определим функцию которая обучает модель XGBoost для задачи ранжирования, используя метрику NDCG (Normalized Discounted Cumulative Gain). Модель экстремального градиентного бустинга имеет следующие параметры: n_estimators (количество деревьев), learning_rate (скорость обучения), objective (метод ранжирования), reg_lambda (параметр регуляризации), early_stopping_round (количество раундов для остановки обучения при отсутствии улучшений), tree_method (метод построения деревьев).

Шаг 4: Обучение модели

Шаг 5: Оценка метрики NDCG

Итоговое значение метрики NDCG представлено ниже:

```
XGBoost Pairwise Ranker Model NDCG@5 Score: 0.4741
XGBoost Pairwise Ranker Model NDCG@10 Score: 0.4877
```

Списочный (listwise) метод ранжирования

В отличие от других подходов ранжирования listwise вместо оптимизации прогнозов модели по отдельным парам запрос - элемент, оптимизирует рейтинг модели для списка в целом.

Функция ошибки на всем списке документов (для конкретного запроса):

$$I(\left\{f(x_j^q)\right\}_{j=1}^{m_q}, \left\{r_j^q\right\}_{j=1}^{m_q}) \rightarrow min$$

Особенность: оценка производится на уровне списка, а не отдельных объектов или пар.

Модель глубокого контекстного анализа (DLCM)

Одним из нетривиальных подходов к решению задачи ранжирования является модель глубокого контекстного анализа по списку DLCM (deep listwise context model). Такой подход основан на взаимосвязи документов в контексте всего списка и реализован с помощью рекуррентной нейронной сети RRN, которая позволяет учитывать контекст запроса, то есть предыдущие входные данные (данные о соседних документах).

Архитектура построенной нами DLCM модели выглядит следующим образом:
С помощью модуля nn.Sequential создается несколько последовательных блоков, объединяющих используемые в каждом конкретном блоке слои. Каждый блок содержит полносвязный линейный слой nn.Linear, который преобразует размерность входного вектора следующим образом:

$$y = W * x + b, \text{ где } W - \text{матрица весов, а } b - \text{смещение}$$

За полносвязным линейным слоем следует слой функции активации nn.LeakyReLU (Leaky Rectified Linear Unit), который добавляет нелинейность в модель и позволяет избежать проблемы затухания градиента в отличие от стандартной функции активации ReLU, которая обнуляет все отрицательные входные значения. Данная функция определена следующим образом:

$$f(x) = max(\alpha x\,;\, x)$$

После функции активации идет слой регуляции nn.Dropout, используемый в нейронных сетях для предотвращения переобучения. Во время обучения данный слой случайным образом отключает (обнуляет) определенный процент нейронов предшествующего слоя.
Применение Dropout к i-нейрону выглядит следующим образом:

$$O_i = X_i\, a\,(\sum_{k=1}^{d_i} w_k x_k + b) = a\,(\sum_{k=1}^{d_i} w_k x_k + b),\ if\ X_i = 1$$

и

$$O_i = X_i\, a\,(\sum_{k=1}^{d_i} w_k x_k + b) = 0,\ if\ X_i = 0$$

Далее с помощью метода forward осуществляется последовательное применение всех слоев к входным данным. Таким образом, модель принимает на вход вектор значений и возвращает предсказанное значение.

```
DLCM NDCG@5: 0.3141
DLCM NDCG@10: 0.2197
```

Опираясь на научную литературу в области DLCM, нужно отметить, что такие низкие значения метрики NDCG характерны для данной модели.

			Microsoft Letor Dataset 30K							
Initial List	Model	Loss Function	nDCG@1	ERR@1	nDCG@3	ERR@3	nDCG@5	ERR@5	nDCG@10	ERR@10
SVMrank			0.301	0.124	0.318	0.197	0.335	0.223	0.365	0.246
SVMrank	DNN	ListMLE	0.337*‡	0.149*‡	0.345*‡	0.224*‡	0.356*‡	0.249*‡	0.382*‡	0.271*‡
		SoftRank	0.388*‡	0.208*‡	0.376*‡	0.279*‡	0.379*‡	0.300*‡	0.395*‡	0.318*‡
		AttRank	0.395*‡	0.198*‡	0.392*‡	0.274*‡	0.396*‡	0.297*‡	0.415*‡	0.316*‡
	LIDNN	ListMLE	0.291	0.122	0.312	0.196	0.331	0.222	0.362	0.245
		SoftRank	0.315*	0.141*	0.326*	0.213*	0.341*	0.238*	0.367*	0.260*
		AttRank	0.306*	0.135*	0.318	0.206*	0.331	0.231*	0.361	0.253*
	DLCM	ListMLE	0.339*‡	0.149*‡	0.346*‡	0.223*‡	0.357*‡	0.248*‡	0.381*‡	0.269*‡
		SoftRank	**0.424***+‡	**0.224***+‡	**0.404***+‡	**0.294***+‡	**0.408***+‡	**0.316***+‡	**0.423***+‡	**0.334***+‡
		AttRank	0.407*+‡	0.206*‡	0.399*+‡	0.281*+‡	0.404*+‡	0.303*+‡	0.422*+‡	0.322*+‡

Данная таблица взята из научной статьи “Learning a Deep Listwise Context Model for Ranking Refinement” Qingyao Ai, Jiafeng Guo, Keping Bi, W. Bruce Croft.
https://www.researchgate.net/publication/326134226_Learning_a_Deep_Listwise_Context_Model_for_Ranking_Refinement

По какой причине может получаться низкое значение метрики NCDG?

1. В списке результатов содержится мало релевантных документов (если большинство элементов имеет низкую оценку релевантности, то итоговое значение метрики будет низким).
2. Использование неинформативных или неэффективных признаков.
3. Высокая сложность задачи и относительная простота модели.

Выводы:
Лучший результат метрики NDCG показала модель CatBoost Pairwise Ranker.