Яндекс

Яндекс ClickHouse

Применение моделей CatBoost в ClickHouse

Николай Кочетов

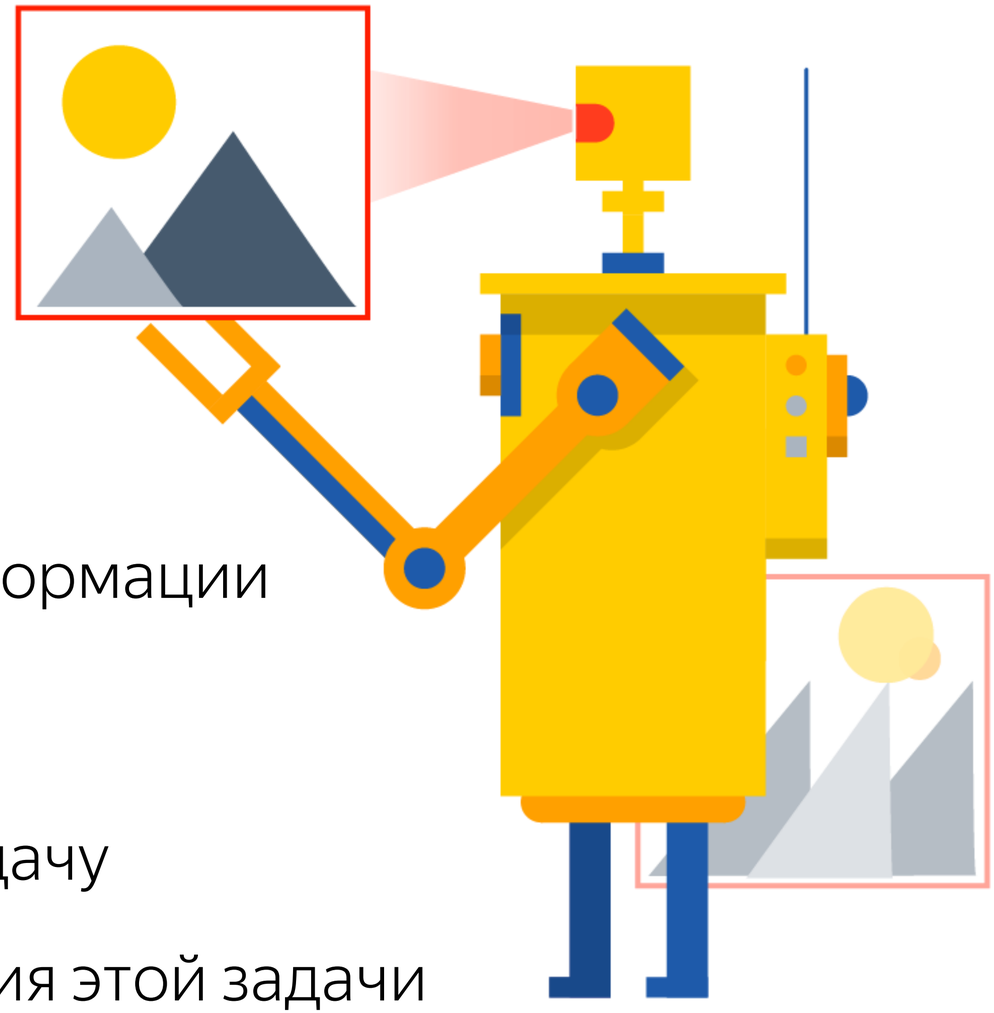
Что такое CatBoost?

CatBoost is an open-source gradient boosting library with categorical features support

Машинное обучение

Предпосылки

- Обилие различных данных
- Рост вычислительной мощности
- Потребность эффективной обработки информации

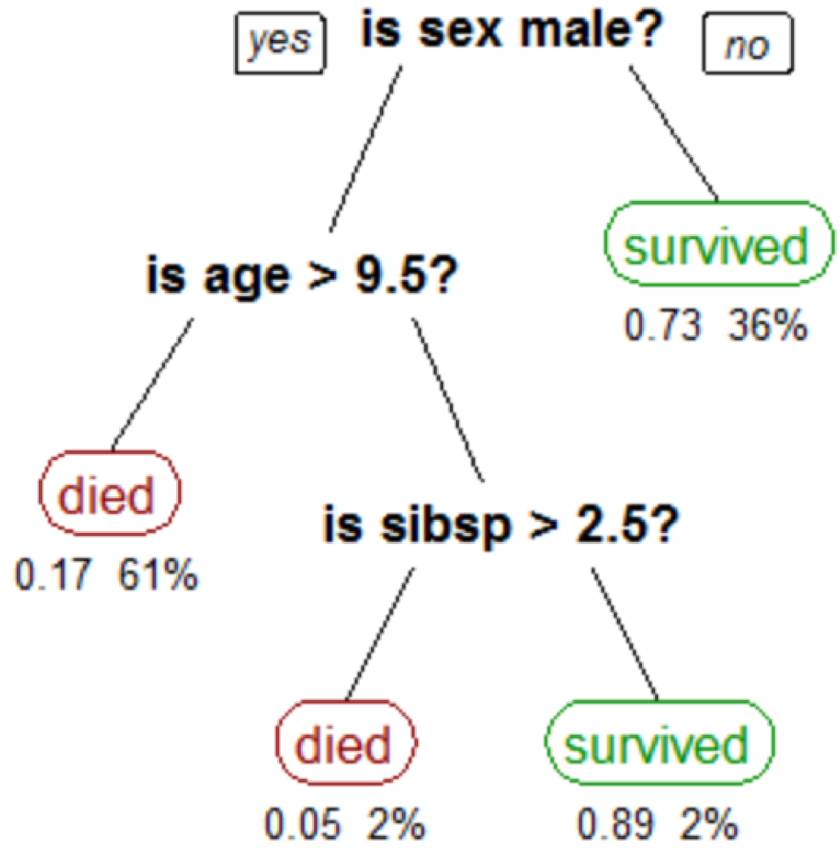


Концепция

- Обучаем алгоритм решать конкретную задачу
- Используем обученную модель для решения этой задачи

Дерево принятия решений

- хорошо интерпретируемая модель - анкета/скрипт
- могут восстанавливать нелинейные зависимости
- легко переобучаются (можно построить дерево с нулевой ошибкой на обучающей выборке)



Градиентный бустинг

Ошибка

Ошибка

Ошибка

Сравнение алгоритмов

	CatBoost	LightGBM		XGBoost		H2O	
Adult	0.269741	0.276018	+ 2.33 %	0.275423	+ 2.11%	0.275104	+ 1.99%
Amazon	0.137720	0.163600	+ 18.79 %	0.163271	+ 18.55%	0.162641	+ 18.09%
Appet	0.071511	0.071795	+ 0.40 %	0.071760	+ 0.35%	0.072457	+ 1.32%
Click	0.390902	0.396328	+ 1.39 %	0.396242	+ 1.37%	0.397595	+ 1.71%
Internet	0.208748	0.223154	+ 6.90 %	0.225323	+ 7.94%	0.222091	+ 6.39%
Kdd98	0.194668	0.195759	+ 0.56 %	0.195677	+ 0.52%	0.195395	+ 0.37%
Kddchurn	0.231289	0.232049	+ 0.33 %	0.233123	+ 0.79%	0.232752	+ 0.63%
Kick	0.284793	0.295660	+ 3.82 %	0.294647	+ 3.46%	0.294814	+ 3.52%

Logloss

Предсказание вероятности покупки

Предсказание вероятности покупки

Задача

Узнать больше информации про пользователей

Цель

По поведению пользователей в прошлом, предсказывать их поведение в будущем. Например, ответить на вопрос — с какой вероятностью пользователь совершит заказ.

Рабочий процесс

Вычисление признаков

Построение выборки и оценка качества

Предсказание вероятности покупки

Сбор данных

Используем данные **Яндекс.Метрики** из **Logs API**

Сырые данные по хитам и визитам

- Предыдущие покупки
- Состояние корзины
- Посещения страниц с описанием товара

Загружаем данные в ClickHouse

Вычисление признаков

- Храним данные в ClickHouse в неагрегированном виде
- Можем получить (почти) любые признаки
- Например, средняя длительность сессии:

```
SELECT
    avg(Duration) as avg_duration
FROM visits_all SAMPLE 1/10 OFFSET 2/10
WHERE StartDate BETWEEN '{start_date}' AND '{end_date}'
GROUP BY FirstPartyCookie
```

Обучение моделей

Собрали выборку и обучили несколько различных моделей:

- SVM
- Logistic Regression
- Random Forest
- XGBoost
- CatBoost

Внедрение обученной модели

В результате обучения получили модель и набор скриптов.

Как с этим жить?

Способ 1: Используем код из обучения

Наводим порядок в куче скриптов:

1. Выгружаем данные раз в неделю (каждый понедельник)
2. Применяем модель также, как и тестировали
3. Загружаем вероятность покупки в таблицу ClickHouse

Способ 1: Используем код из обучения

Преимущества

- Просто, быстро, эффективно. Часть кода уже написана.
- Полезно. Пригодится при переобучении.

Недостатки

- Загрузка и выгрузка данных. Может тормозить.
- Заранее готовим ответы.
  Сложно получить ответ для произвольного периода.

Способ 2: препарируем модель

План работ

- Смотрим, как устроена модель
- Переносим процесс применения в хранилище данных
- Избавляемся от загрузки и выгрузки данных

Чего сможем добиться?

- Избавимся от перекладывания данных
- Будем работать с произвольным множеством данных
- Применение модели — запрос в базу

Способ 2: препарируем модель

- Какие алгоритмы можем перенести в СУБД?

Способ 2: препарируем модель

Линейные классификаторы

Тривиально

```
SELECT ((feature1 * w1) + ... + (featureN * wN)) > threshold
FROM table
```

Логистическая регрессия

Результаты на тестовой выборке

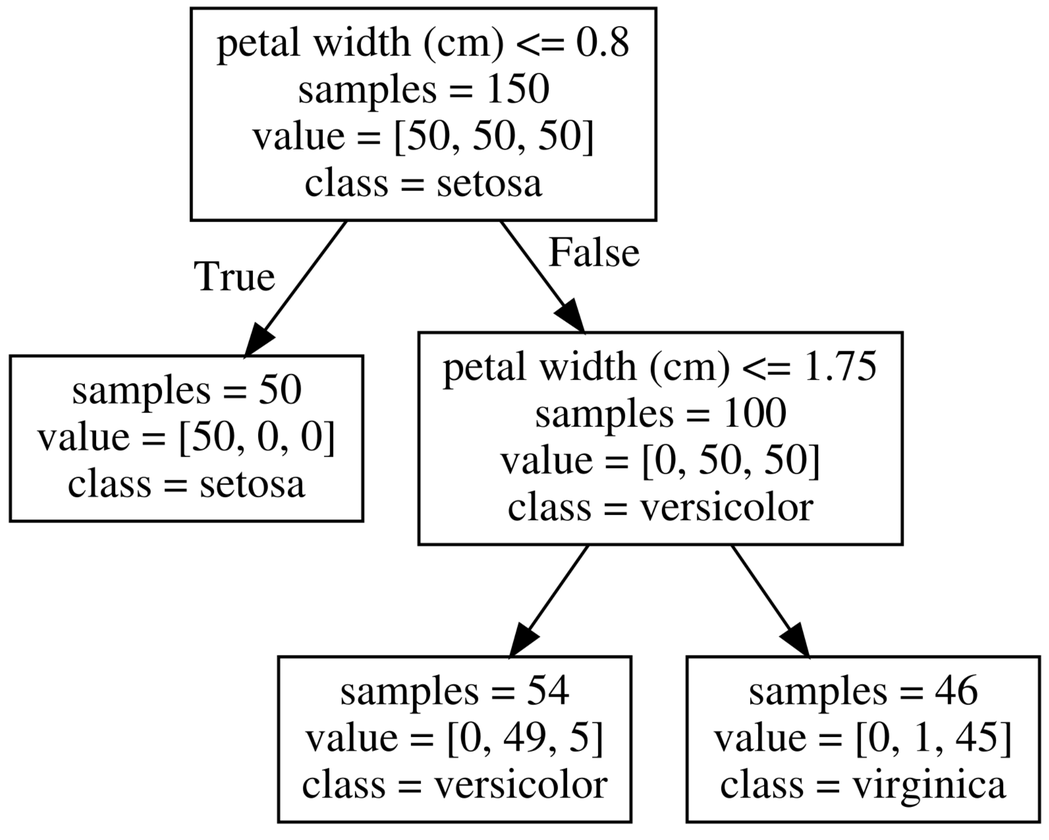
- LogLoss: 0.0411
- 0.44 sec. 441497 rows/sec. 63.16 MiB/sec.

Способ 2: препарируем модель

Дерево принятия решений

Вкладываем дерево в цепочку вызовов условных функций

```
SELECT
    if(petal_width <= 0.8,
       'setosa',
       if(petal_width <= 1.75,
         'versicolor',
         'virginica'))
FROM iris
```



Способ 2: препарируем модель

Лес, бустинг — набор деревьев

```
SELECT arrayReduce('avg',[if(...), ..., if(...)])
```

Random Forest, 100 деревьев глубины 3

Результаты на тестовой выборке

- LogLoss: 0.0437 (0.0395 для глубины 6)
- 2.54 sec. 59450 rows/sec. 1.59 MiB/sec.

Способ 2: препарируем модель

Небольшой лес из 100 деревьев

```
SELECT arrayReduce('sum',[if(f21<0.287376, if(f2<-1.34168, if(f21<-0.0465033, -0.0411413, -0.0168165), if(f18<0.0541038, -0.0486026, -0.0432873)), if(f2<-1.12007, if(f19<0.514323, -0.00154215, 0.0279125), if
(f23<-0.0269176, -0.0391427, -0.0148464))),if(f21<0.287376, if(f2<-1.30474, if(f21<-0.0465033, -0.0407256, -0.0184058), if(f23<-0.0274336, -0.0469912, -0.0290771)), if(f2<-1.157, if(f23<-0.027484, -0.00114219,
0.029407), if(f2<-0.344426, -0.0242555, -0.0423453))),if(f18<0.883746, if(f2<-1.34168, if(f21<-0.00940569, -0.0386349, -0.00968768), if(f23<-0.0274894, -0.0458335, -0.0281316)), if(f2<-1.30474, if
(f23<-0.0273632, -0.00346079, 0.0302068), if(f19<0.514323, -0.0340366, -0.00648281))),if(f21<0.250278, if(f2<-1.34168, if(f21<-0.0465033, -0.0383863, -0.0166084), if(f18<0.0541038, -0.0453193, -0.0402171)), if
(f2<-1.30474, if(f22<0.233904, 0.00176162, 0.0268355), if(f2<-0.639908, -0.0178382, -0.0385038))),if(f18<0.607199, if(f2<-1.34168, if(f21<-0.00940569, -0.0373913, -0.0101554), if(f23<-0.0274336, -0.0439256,
-0.0289171)), if(f2<-1.19394, if(f19<0.514323, -0.00625752, 0.0268741), if(f19<0.514323, -0.0357198, -0.0104513))),if(f21<0.435766, if(f2<-1.34168, if(f21<-0.00940569, -0.035767, -0.0104316), if
(f23<-0.0274894, -0.0428682, -0.0229862)), if(f2<-1.0462, if(f23<-0.0273248, 0.00187816, 0.02685), if(f19<1.63821, -0.032482, 0.00204186))),if(f18<0.607199, if(f2<-1.30474, if(f21<-0.0465033, -0.037346,
-0.0156485), if(f21<-0.0465033, -0.042783, -0.0388954)), if(f2<-1.30474, if(f23<-0.0274931, -0.00521804, 0.0276861), if(f23<-0.0275324, -0.0332554, -0.00768356))),if(f21<0.250278, if(f2<-1.30474, if
(f21<-0.0465033, -0.0354773, -0.015733), if(f18<0.0541038, -0.0418548, -0.0373441)), if(f2<-1.30474, if(f19<0.514323, 0.00189302, 0.0275378), if(f2<-0.824584, -0.0146903, -0.0342196))),if(f21<0.21318, if
(f2<-1.30474, if(f21<-0.0465033, -0.0347172, -0.0169444), if(f19<0.514323, -0.04073, -0.0248928)), if(f2<-1.30474, if(f22<0.233904, 8.23854E-4, 0.0253576), if(f23<-0.027183, -0.0319395, -0.00700244))),if
(f21<0.287376, if(f2<-1.30474, if(f21<-0.0465033, -0.0343388, -0.014464), if(f23<-0.0268591, -0.0399331, -0.0230654)), if(f2<-1.19394, if(f22<0.233904, -0.00125408, 0.0236391), if(f23<-0.0271985, -0.031504,
-0.0054035))),if(f21<0.21318, if(f2<-1.30474, if(f21<-0.00940569, -0.0328916, -0.0128814), if(f23<-0.0274492, -0.0392849, -0.0231176)), if(f2<-1.08313, if(f19<0.514323, -0.00200433, 0.02365), if(f2<-0.307491,
-0.0222099, -0.0358909))),if(f18<0.607199, if(f2<-1.34168, if(f21<0.027692, -0.0317689, -0.00479046), if(f23<-0.0274336, -0.0386082, -0.0233539)), if(f2<-1.30474, if(f23<-0.0274876, -0.00522193, 0.0275162), if
(f19<0.514323, -0.0297226, -0.00283392))),if(f18<0.607199, if(f2<-1.37861, if(f21<-0.00940569, -0.0310768, -0.00633344), if(f18<0.0541038, -0.0384042, -0.033431)), if(f2<-1.12007, if(f22<0.233904, -0.00648516,
0.0217163), if(f19<0.514323, -0.0309866, -0.00797589))),if(f18<0.607199, if(f2<-1.30474, if(f21<-0.00940569, -0.0317836, -0.00930418), if(f21<-0.0465033, -0.0380014, -0.0341582)), if(f2<-1.157, if
(f22<0.233904, -0.00607711, 0.0222084), if(f19<0.514323, -0.0298831, -0.00679007))),if(f18<0.607199, if(f2<-1.30474, if(f21<-0.00940569, -0.0312764, -0.00765295), if(f21<-0.0465033, -0.0374954, -0.0336616)), if
(f2<-1.30474, if(f19<0.514323, -0.00339929, 0.025132), if(f23<-0.0275324, -0.0277916, -0.00536361))),if(f18<0.607199, if(f2<-1.37861, if(f21<-0.0465033, -0.0306154, -0.00996313), if(f23<-0.0274894, -0.0363659,
-0.0205428)), if(f2<-1.12007, if(f19<0.514323, -0.00647451, 0.0239892), if(f19<1.63821, -0.028755, 0.00906169))),if(f21<0.138985, if(f2<-1.157, if(f18<0.0541038, -0.0311921, -0.0167023), if(f18<0.0541038,
-0.0365831, -0.0329484)), if(f2<-1.37861, if(f23<-0.0275224, 0.00278828, 0.0268101), if(f2<-0.639908, -0.0124602, -0.0313422))),if(f21<0.138985, if(f2<-1.37861, if(f21<-0.0465033, -0.0286743, -0.0120908), if
(f23<-0.0274794, -0.0354306, -0.0190915)), if(f2<-1.0462, if(f22<0.233904, -0.00441485, 0.0187175), if(f23<-0.026721, -0.0302931, -0.00990793))),if(f21<0.138985, if(f2<-1.12007, if(f23<-0.0275315, -0.0287219,
0.0120843), if(f18<0.0541038, -0.0356779, -0.0321881)), if(f2<-1.26781, if(f23<-0.0273504, -6.72319E-4, 0.0238452), if(f2<-0.602973, -0.0144265, -0.0304927))),if(f21<0.101887, if(f2<-1.30474, if(f18<0.0541038,
-0.0294119, -0.0142636), if(f2<-0.676843, -0.0320512, -0.035213)), if(f2<-1.08313, if(f22<0.233904, -0.0050053, 0.0185778), if(f2<-0.0489445, -0.0208435, -0.0326259))),if(f21<0.101887, if(f2<-1.12007, if
(f21<-0.0465033, -0.0296804, -0.0166378), if(f23<-0.0274474, -0.0345859, -0.0216888)), if(f2<-1.26781, if(f22<0.233904, -0.00305679, 0.0205645), if(f2<-0.270556, -0.0169205, -0.0314051))),if(f21<0.21318, if
(f2<-1.157, if(f21<-0.0465033, -0.0290345, -0.0139274), if(f18<0.0541038, -0.0343114, -0.0300956)), if(f2<-1.0462, if(f23<-0.0274025, -0.00231972, 0.0219966), if(f23<-0.026871, -0.0280293, -0.00744155))),if
(f18<0.330651, if(f2<-1.37861, if(f21<-0.00940569, -0.0267404, -0.00563836), if(f2<-0.824584, -0.0304892, -0.0340051)), if(f2<-1.12007, if(f22<0.233904, -0.00695099, 0.0184827), if(f23<-0.0275041, -0.0273197,
-0.00706744))),if(f18<0.330651, if(f2<-1.30474, if(f21<-0.0465033, -0.0283736, -0.0114656), if(f21<-0.0465033, -0.0335919, -0.0301803)), if(f2<-1.19394, if(f23<-0.0275324, -0.0056847, 0.0212156), if
(f23<-0.0271985, -0.0267871, -0.00349806))),if(f18<0.607199, if(f2<-1.37861, if(f21<-0.0465033, -0.0267457, -0.0066822), if(f21<-0.0465033, -0.0331048, -0.028648)), if(f2<-1.12007, if(f23<-0.0270814,
-0.00400424, 0.020147), if(f23<-0.0274565, -0.0254219, -0.0043865))),if(f21<0.0647896, if(f2<-1.30474, if(f16<-0.252381, -0.0281698, -0.0169334), if(f18<0.0541038, -0.0329024, -0.0290634)), if(f2<-1.08313, if
(f23<-0.0274867, -0.00479253, 0.0186878), if(f2<-0.270556, -0.017792, -0.029774))),if(f18<0.330651, if(f2<-1.34168, if(f21<-0.0465033, -0.0269778, -0.00866721), if(f2<-0.824584, -0.029273, -0.0326748)), if
(f2<-1.12007, if(f23<-0.027484, -0.00552632, 0.018317), if(f23<-0.0274904, -0.0259033, -0.00592927))),if(f18<0.330651, if(f2<-1.30474, if(f21<-0.0465033, -0.0268872, -0.0101706), if(f23<-0.0274364, -0.0319173,
-0.0183374)), if(f2<-1.12007, if(f19<0.514323, -0.0064463, 0.0195015), if(f19<0.514323, -0.0256998, -0.00468003))),if(f21<0.176083, if(f2<-1.30474, if(f18<0.0541038, -0.0256026, -0.0108268), if(f18<0.0541038,
-0.0318308, -0.0270271)), if(f2<-1.37861, if(f22<0.233904, 0.00221522, 0.0221127), if(f2<-0.307491, -0.010695, -0.027506))),if(f21<0.0647896, if(f2<-1.37861, if(f19<0.514323, -0.021889, 0.0214266), if
(f18<0.0541038, -0.0315352, -0.0268925)), if(f2<-1.08313, if(f22<0.233904, -0.00340782, 0.0172627), if(f2<-0.0489445, -0.0175592, -0.02915))),if(f21<0.101887, if(f2<-1.0462, if(f18<0.0541038, -0.0264361,
-0.01378), if(f3<1.01952, -0.0315176, -0.0273609)), if(f2<-1.34168, if(f22<0.233904, 1.76803E-4, 0.0191606), if(f2<-0.344426, -0.011713, -0.0270393))),if(f18<0.330651, if(f2<-1.30474, if(f21<-0.0465033,
-0.0256311, -0.00864613), if(f2<-0.824584, -0.0273856, -0.0311522)), if(f2<-1.37861, if(f22<0.233904, -6.7716E-4, 0.0203041), if(f19<0.514323, -0.0217005, 1.10617E-4))),if(f21<0.101887, if(f2<-1.30474, if
(f17<-0.145753, -0.0257026, -0.0135729), if(f18<0.0541038, -0.0307452, -0.0259101)), if(f2<-1.0462, if(f23<-0.0274025, -0.00314162, 0.0178064), if(f2<-0.0489445, -0.016502, -0.0281161))),if(f18<0.607199, if
(f2<-1.30474, if(f21<-0.0465033, -0.0246073, -0.00673528), if(f21<-0.0465033, -0.030584, -0.0260979)), if(f2<-1.0462, if(f19<0.514323, -0.00417125, 0.0196098), if(f2<-0.307491, -0.0127647, -0.0256753))),if
(f21<0.0647896, if(f2<-1.12007, if(f21<-0.083601, -0.0263189, -0.0161315), if(f19<0.514323, -0.0302128, -0.0146063)), if(f2<-1.0462, if(f23<-0.0274876, -0.0036788, 0.0171262), if(f23<-0.0273257, -0.0243853,
-0.00539893))),if(f18<0.330651, if(f2<-1.34168, if(f21<-0.083601, -0.025844, -0.0115585), if(f23<-0.0274336, -0.0296326, -0.0135145)), if(f2<-1.37861, if(f15<0.313696, -0.0057402, 0.0147234), if(f19<0.514323,
-0.0208168, 0.00104059))),if(f21<0.0647896, if(f2<-1.0462, if(f19<0.514323, -0.0229019, 0.0143177), if(f3<1.67772, -0.0299549, -0.0240689)), if(f2<-0.861519, if(f22<0.233904, -0.00472233, 0.0151651), if
(f2<0.246537, -0.0193216, -0.0284809))),if(f18<0.330651, if(f2<-1.0462, if(f21<-0.0465033, -0.0251667, -0.0104095), if(f23<-0.027387, -0.0295698, -0.0176517)), if(f2<-1.37861, if(f15<0.213168, -0.0062054,
0.0120493), if(f2<-0.307491, -0.0110783, -0.0265832))),if(f18<0.330651, if(f2<-1.30474, if(f21<-0.0465033, -0.023067, -0.00776957), if(f23<-0.0274364, -0.0289648, -0.0132435)), if(f2<-1.0462, if
(f23<-0.0274876, -0.0051898, 0.0171133), if(f2<-0.0489445, -0.0154874, -0.0273422))),if(f21<0.0647896, if(f2<-1.157, if(f15<-0.170228, -0.0246292, -0.013444), if(f3<1.82221, -0.0291237, -0.0217857)), if
(f2<-1.0462, if(f22<0.233904, -0.00316204, 0.0154117), if(f2<-0.0489445, -0.0146096, -0.0264161))),if(f21<0.027692, if(f2<-1.34168, if(f17<-0.145753, -0.0235227, -0.0111151), if(f2<-0.492167, -0.0259396,
-0.0293583)), if(f2<-1.0462, if(f23<-0.0274775, -0.00389493, 0.0165575), if(f2<-0.0120093, -0.01558, -0.0263846))),if(f21<0.0647896, if(f2<-1.12007, if(f21<-0.083601, -0.0243, -0.0136216), if(f23<-0.0274336,
-0.0286324, -0.0145423)), if(f2<-1.0462, if(f22<0.233904, -0.00365398, 0.0153315), if(f19<1.63821, -0.02102, 0.0238225))),if(f18<0.330651, if(f2<-1.34168, if(f21<-0.083601, -0.0236564, -0.00883573), if
(f23<-0.0274336, -0.0280848, -0.0111912)), if(f2<-0.639908, if(f22<0.233904, -0.00724257, 0.0137201), if(f19<0.514323, -0.0240014, -0.00753991))),if(f18<0.330651, if(f2<-1.30474, if(f21<-0.0465033, -0.0219351,
-0.00640791), if(f23<-0.0274336, -0.0279327, -0.0108776)), if(f2<-1.00926, if(f23<-0.0273632, -0.00480692, 0.0170223), if(f19<0.514323, -0.0213351, -0.00180769))),if(f18<0.0541038, if(f2<-1.37861, if
(f21<-0.083601, -0.0235011, -0.00839891), if(f2<-0.824584, -0.0248607, -0.0285964)), if(f2<-0.824584, if(f22<0.233904, -0.00835252, 0.013313), if(f23<-0.027483, -0.0240317, -0.00831159))),if(f21<0.101887, if
(f2<-1.30474, if(f23<-0.0275324, -0.0179696, 0.0198129), if(f18<0.0541038, -0.0279481, -0.0223771)), if(f2<-0.861519, if(f23<-0.0275123, -0.00265762, 0.0164488), if(f3<0.762661, -0.0237467, -0.012149))),if
```

Способ 2: препарируем модель

Недостатки

- Сложность преобразования модели в запрос
- Ограниченная применимость
- Не для всех алгоритмов машинного обучения
- Ограничения со стороны СУБД
- Проблемы с производительностью

Способ 3: встраиваем применение в базу

- Используем библиотеку машинного обучения внутри базы
- Перекладываем на базу работу по преобразованию данных
- Применяем модель как вызов встроенной функции

```
SELECT modelEvaluate('iris', sepal_width, petal_width)
FROM iris
```

Способ 3: встраиваем применение в базу

Преимущества

Те же, что и у предыдущего способа, но

- Нет неоправданных проблем с производительностью
- Основная работа — на базе данных
- Оптимизация работы внутри библиотеки машинного обучения

Недостатки

- База должна поддерживать работу с конкретным алгоритмом
- Различия в версиях и форматах хранения

Способ 3: встраиваем применение в базу

CatBoost, 100 деревьев глубины 6

Результаты на тестовой выборке

- LogLoss : 0.0362
- 3.96 sec. 19467 rows/sec. 2.78 MiB/sec.

XGBoost

Результаты на тестовой выборке

- 100 деревьев глубины 6 — LogLoss: 0.0376

Способ 3: встраиваем применение в базу

Результаты

Алгоритм	LogLoss	Время	Скорость
Логистическая регрессия	0.0411	0.44 sec.	63.16 MiB/sec.
Лес, 100 дер. глубины 3	0.0437	2.54 sec.	1.59 MiB/sec.
Лес, 100 дер. глубины 6	0.0395		
XGBoost, 100 дер. глубины 6	0.0376		
CatBoost, 100 дер. глубины 6	0.0362	3.96 sec.	2.78 MiB/sec.

Интеграция ClickHouse и CatBoost

Модели CatBoost в ClickHouse

1. Описываем конфигурацию модели

```
<models>
    <model>
        <!-- Тип модели. Сейчас только catboost. -->
        <type>catboost</type>
        <!-- Имя модели. Для modelEvaluate(). -->
        <name>purchase_model</name>
        <!-- Путь к обученной модели. -->
        <path>clickhouse/models/model.cbm</path>
        <!-- Период обновления. 0 — не обновляем. -->
        <lifetime>0</lifetime>
    </model>
</model>
```

Модели CatBoost в ClickHouse

1. Описываем конфигурацию модели
2. В config.xml добавляем путь к конфигурации и путь к CatBoost

```
<!-- catboost/catboost/libs/model_interface репозитория CatBoost -->
<!-- Сборка: ../../../ya make -r -->
<catboost_dynamic_library_path>
    /path/to/libcatboostmodel.so
</catboost_dynamic_library_path>
<!-- Используем '*' для маски поиска -->
<models_config>
    clickhouse/models/model*.xml
</models_config>
```

Модели CatBoost в ClickHouse

1. Описываем конфигурацию модели
2. В config.xml добавляем путь к конфигурации и путь к CatBoost
3. Используем функцию

```
modelEvaluate('model_name', feature1, ..., featureN)
```

```
SELECT
    modelEvaluate('purchase_model', ...) AS prediction
FROM table
```

Сначала перечисляем числовые признаки, затем категориальные.

Чтение из CatBoost Pool

Формат входных данных для обучения CatBoost — CatBoost Pool

1. Описание столбцов — TSV файл вида

```
column_id     data_type     feature_id (optional)
```

   Пример для двух признаков и Target

```
0       Categ   is_yabrowser
1       Num     viewed_products
2       Target
```

2. Описание датасета — TSV файл с данными

Чтение из CatBoost Pool

Чтобы быстро протестировать работу обученной модели в ClickHouse, добавлена возможность читать данные сразу из пула CatBoost.

> **Табличная функция catBoostPool.**

Параметры — пути к файлам с описанием столбцов и датасета.

```
catBoostPool('/path/to/column/description', '/path/to/dataset/description')
```

Создает временную таблицу с движком File('TSV').

Файлы должны находиться в директории данных сервера.

Чтение из CatBoost Pool

Описание столбцов пула CatBoost:

```
0	Categ	is_yabrowser
1	Num	viewed_products
2	Target
```

Описание столбцов в catBoostPool

```
DESCRIBE TABLE catBoostPool('test.cd', 'test.csv')
```

```
┌─name──────────────┬─type────┬─default_type─┬─────────────┐
│ Num0              │ Float64 │              │             │
│ Categ0            │ String  │              │             │
│ Target            │ Float64 │              │             │
│ is_yabrowser      │ String  │ ALIAS        │ Categ0      │
│ viewed_products   │ Float64 │ ALIAS        │ Num0        │
└───────────────────┴─────────┴──────────────┴─────────────┘
```

Чтение из CatBoost Pool

Описание столбцов пула CatBoost:

```
0       Categ  is_yabrowser
1       Num    viewed_products
2       Target
```

Описание столбцов в catBoostPool

```
DESCRIBE TABLE
(
    SELECT *
    FROM catBoostPool('test.cd', 'test.csv')
)
```

```
┌─name─────┬─type────┬─default_type─┬─default_expression─┐
│ Num0     │ Float64 │              │                    │
│ Categ0   │ String  │              │                    │
└──────────┴─────────┴──────────────┴────────────────────┘
```

Использование обученной модели

- Предсказываем вероятность покупки

```
SELECT
    modelEvaluate('purchase_model', *) AS prediction,
    1. / (1. + exp(-prediction)) AS probability
FROM catBoostPool('test.cd', 'test.csv')
```

Метрика LogLoss

Зафиксируем ответы классификатора на выборке. Какова вероятность получить правильный ответ при заданном классификатором распределении?

Для удобства логарифмируем условную вероятность и меняем знак

$$-\log P(y|p) = -(y \log p + (1-y) \log (1-p))$$

Усредняем значение по всей выборке

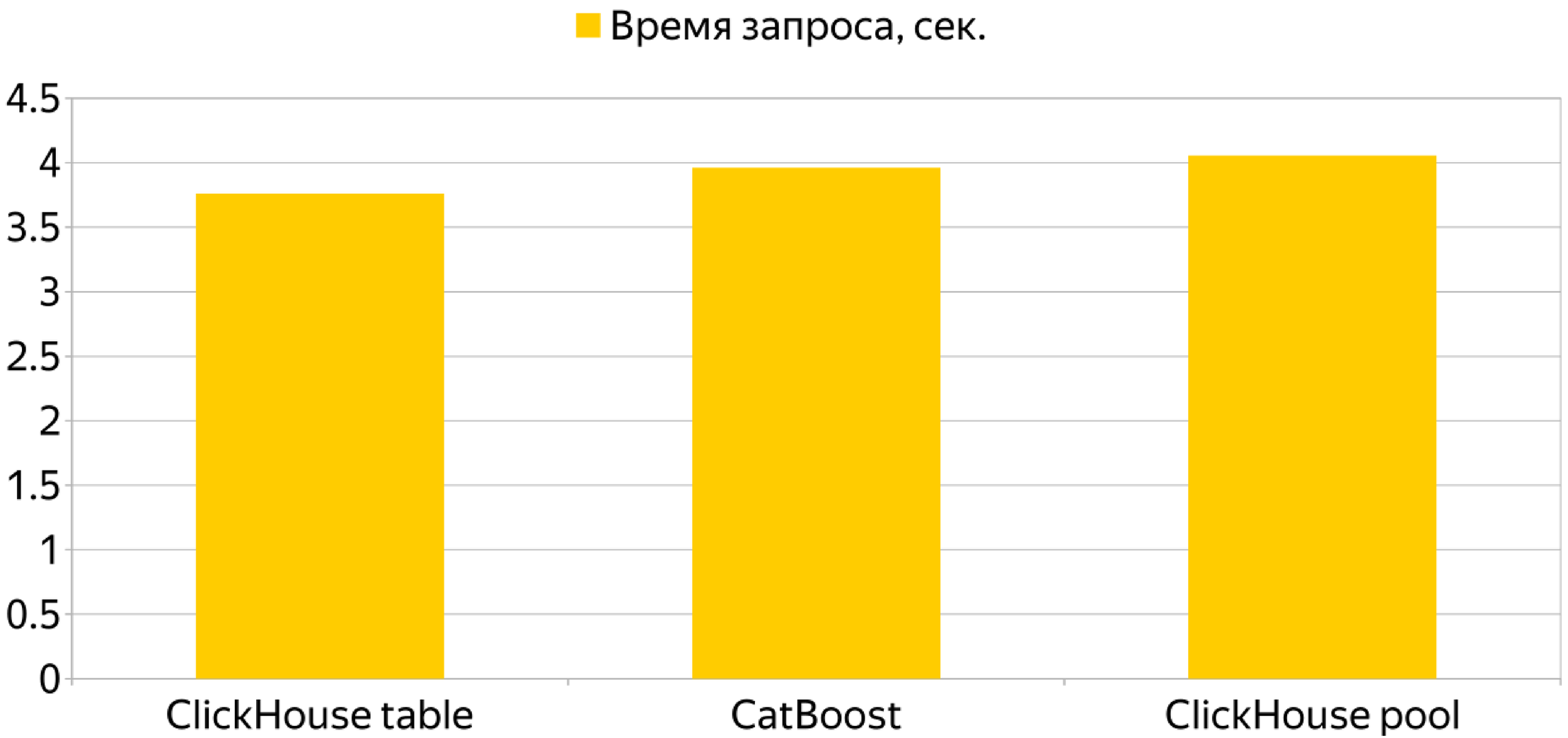
Использование обученной модели

Считаем ошибку на тестовой выборке по метрике Logloss

```
SELECT -avg((Target * log(prob)) +
            ((1. - Target) * log(1. - prob))) AS logloss
FROM
(
    SELECT
        modelEvaluate('purchase_model', *) AS pred,
        1. / (1. + exp(-pred)) AS prob,
        Target
    FROM catBoostPool('test.cd', 'test.csv')
)
```

```
┌─────────logloss─┐
│ 0.0362389241496 │
└─────────────────┘
```

Использование обученной модели



Итоги

Интеграция ClickHouse и CatBoost

- Применение обученных моделей
- Чтение данных из пула

Дальнейшие планы

- Другие форматы моделей
- Встроенное обучение моделей — ?

Спасибо!


https://github.com/yandex/ClickHouse/


https://clickhouse.yandex


https://groups.google.com/forum/#!forum/clickhouse


https://telegram.me/clickhouse_ru


https://telegram.me/clickhouse_en


clickhouse-feedback@yandex-team.com