



Faculty
of
Computer
science
Higher School of Economics

Tabr: Tabular Deep Learning + Nearest Neighbours

Подготовил:
Казадаев Максим, БПМИ202

План



- Данные – на чем учимся
- Препроцессинг – как предобработываем данные
- Пайплайн обучения – как тюним гиперпараметры
- MLP + CatBoost
- TabR + CatBoost
- Визуализация Nearest Neighbors + интерпретация TabR

Данные

Kaggle Соревнование от American Express

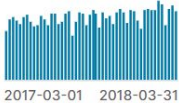
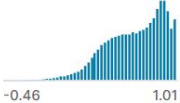




<https://www.kaggle.com/competitions/amex-default-prediction/data>

- Строчки – клиенты (500k клиентов)
- Признаки – 190 анонимных численных полей
- Каждый клиент имеет историю длины ≤ 13
- Target – бинарная величина, будет ли у клиента дефолт в будущем

train_data.csv (16.39 GB)

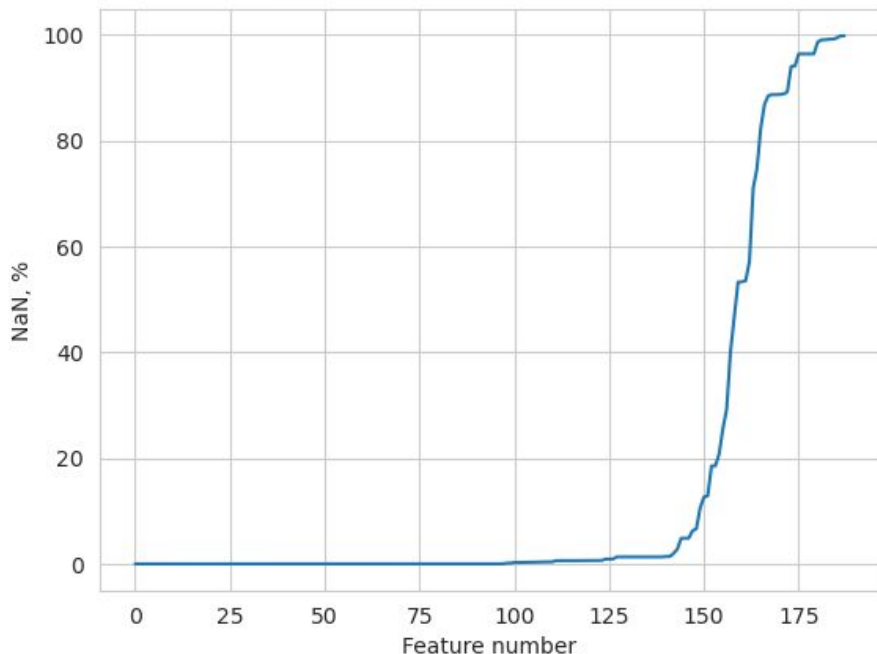
Detail Compact Column

10 of 190 columns

customer_ID	S_2	P_2	D_39	B_1	B_2	R_1
458913 unique values						
0000099d6bd597052cdc da90ffabf56573fe9d7c 79be5fbac11a8ed792fe b62a	2017-03-09	0.9384687191272548	0.0017333390041739	0.0087244509498605	1.0068382339663076	0.0092277222
0000099d6bd597052cdc da90ffabf56573fe9d7c 79be5fbac11a8ed792fe b62a	2017-04-07	0.9366646050988444	0.0057754430691282	0.0049233526310337	1.0006531959897804	0.0061513089

Предобработка данных

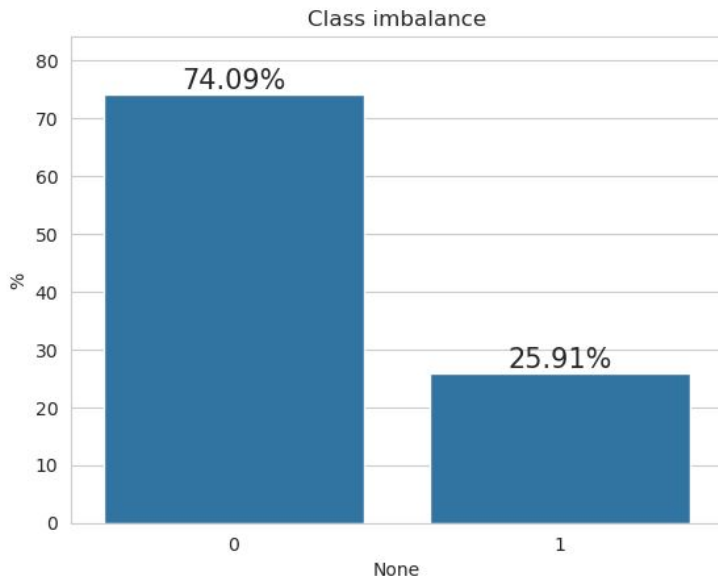
NaN ratios for different features



- Для каждого клиента берем только **последнее наблюдение**.
- Удалим признаки с пропусками. Останется **78 признаков**.
- Уменьшим размер выборки:
 - Train: 32000
 - Validation: 8000
 - Test: 10000

Предобработка данных

	D	S	P	B	R
Original	97	22	3	40	28
Preprocessed	20	14	1	20	25



Поля:

- D – Правонарушения
- S – Расходы
- P – Платежи
- B – Баланс
- R – Показатели риска

Для Deep Learning моделей:

- Стандартизируем признаки
- Добавим циклические эмбединги:
 $\cos(ax)$, $\sin(ax)$

Обучение



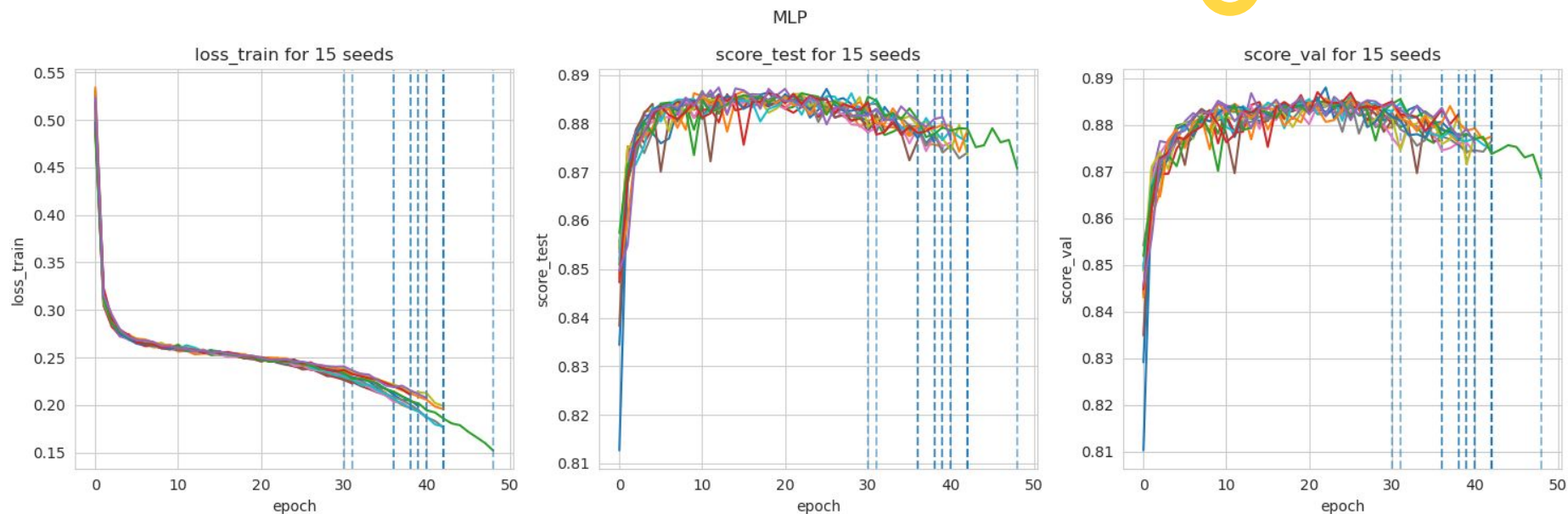
Train:

1. Перебираем **гиперпараметры** по accuracy на валидации с помощью **optuna**
2. Обучаем **15 моделей** с разными random seed с оптимальными параметрами
3. Из 15 моделей составляем **3 ансамбля** по 5 моделей

Test:

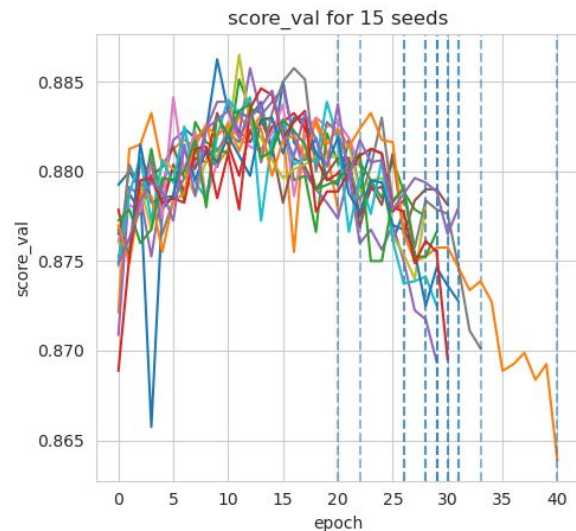
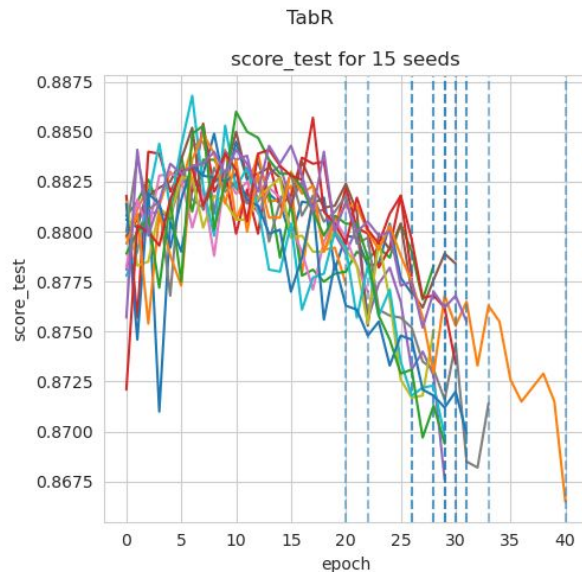
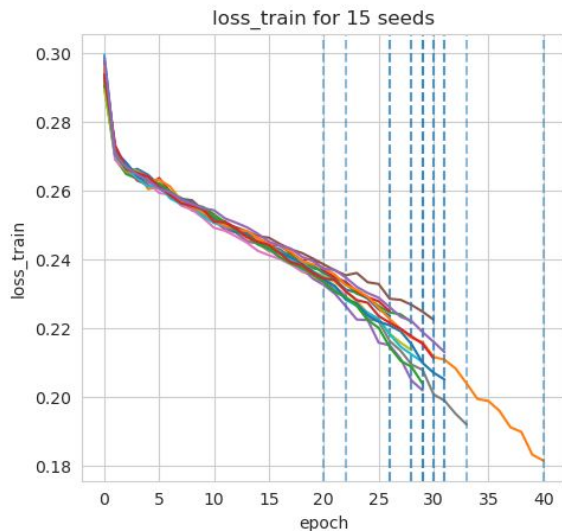
1. Считаем **метрики** для ансамблей: ROC-AUC, Accuracy, Precision, Recall
2. Считаем **стандартное отклонение метрик** у ансамблей
3. Считаем то же самое для **ансамблей из разных моделей**

MLP



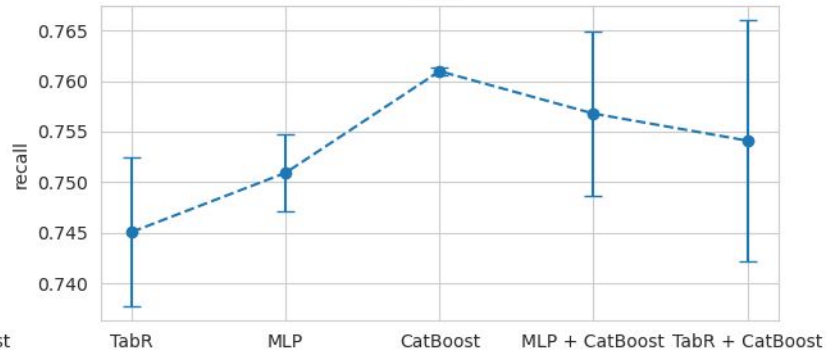
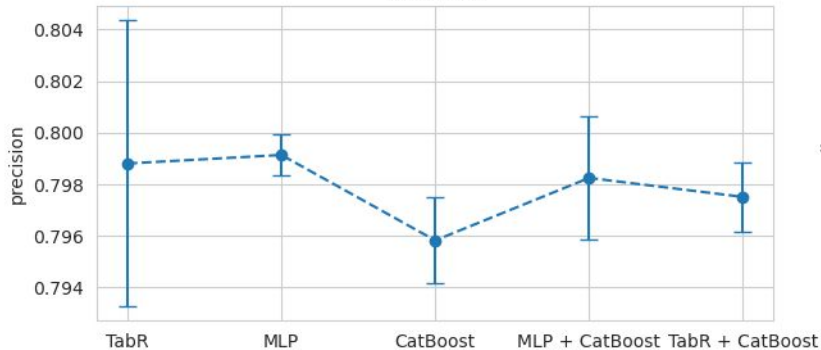
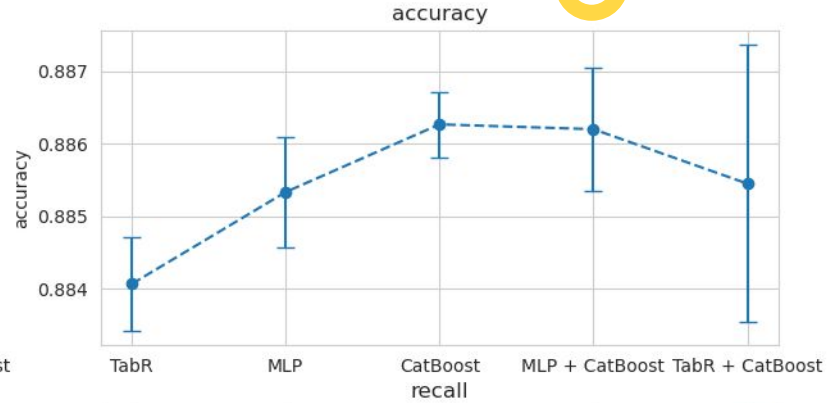
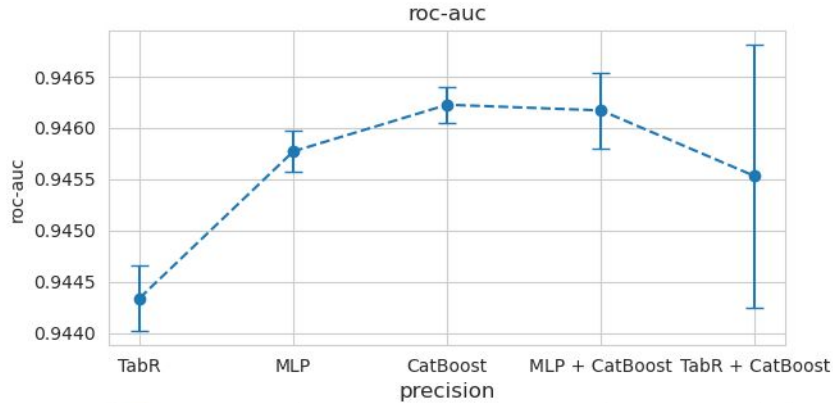
- **15 секунд** на 1 обучение на GPU. **100 перезапусков** для подбора гиперпараметров. Всего **30 минут**.
- С ~20 эпохи MLP переобучается. Метрики на валидации и тесте падают

TabR



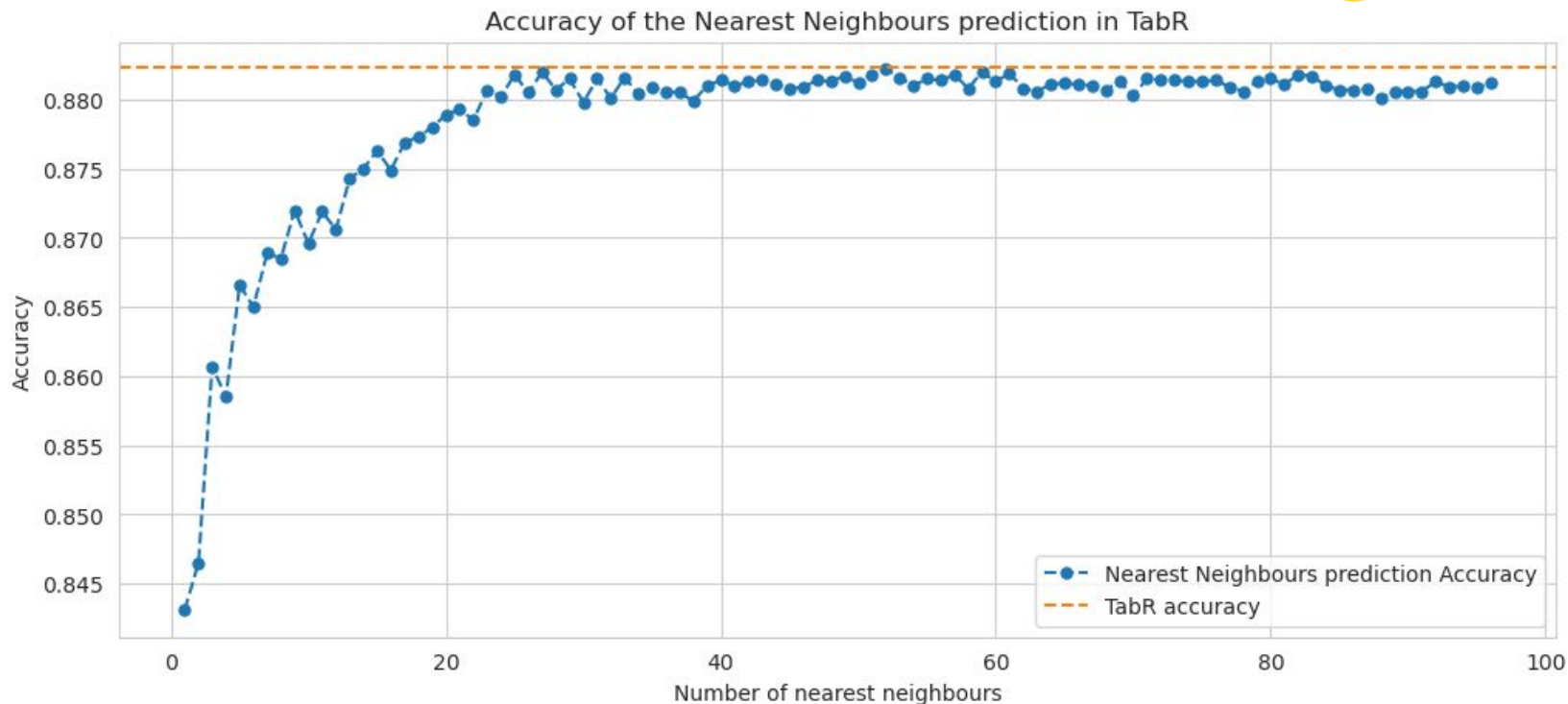
- **5-7 минут** на 1 обучение на GPU. **15 перезапусков** для подбора гиперпараметров. Всего **2 часа**.
- С ~10 эпохи TabR переобучается. Метрики на валидации и тесте **нестабильны**

MLP | TabR | CatBoost



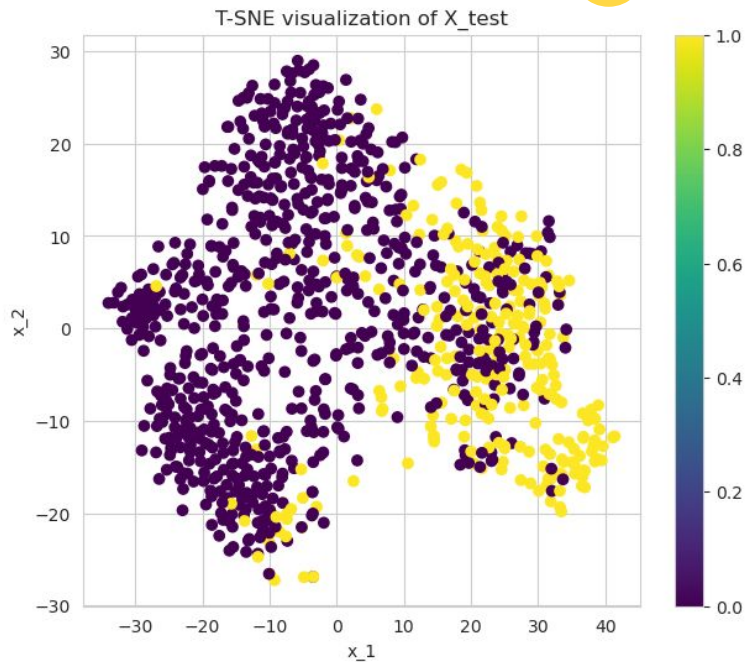
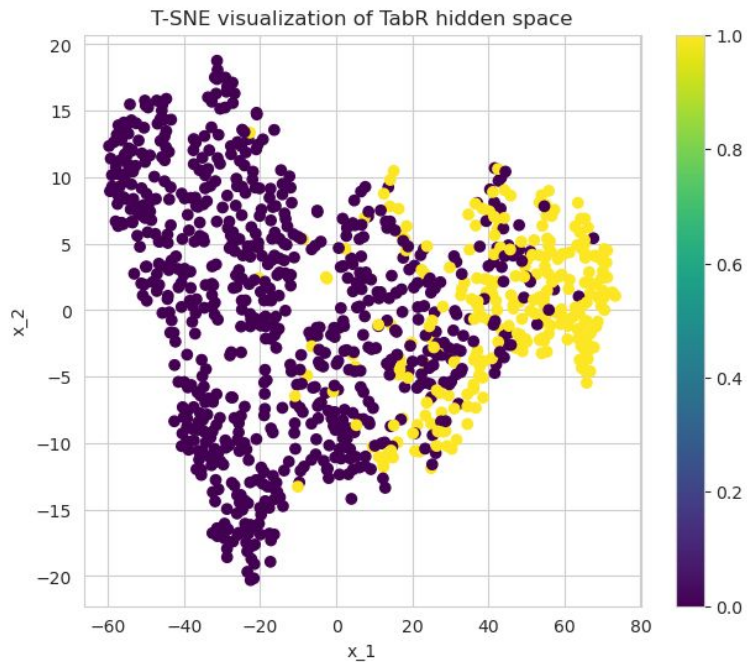
- **CatBoost + MLP ~ CatBoost > MLP > TabR**

TabR Nearest Neighbors



- Берем N ближайших соседей из TabR. Усредняем их предсказания

TabR Nearest Neighbors



- Визуализируем эмбединги, по которым ищутся расстояния в TabR

Выводы

- Deep Learning обычно работает хуже **на табличных данных**, чем градиентный бустинг, но добавление его в ансамбль моделей может улучшить качество.
- **TabR долго учить**: в 5 раз дольше, чем CatBoost, и в 30 раз дольше, чем MLP.
- TabR выучивает пространство для внутренних представлений объектов. На рассмотренной задаче **он оказался лучше, чем KNN**.
- Плюс TabR еще в том, что можно **добавлять новые примеры** в retrieval блок после обучения модели

Материалы

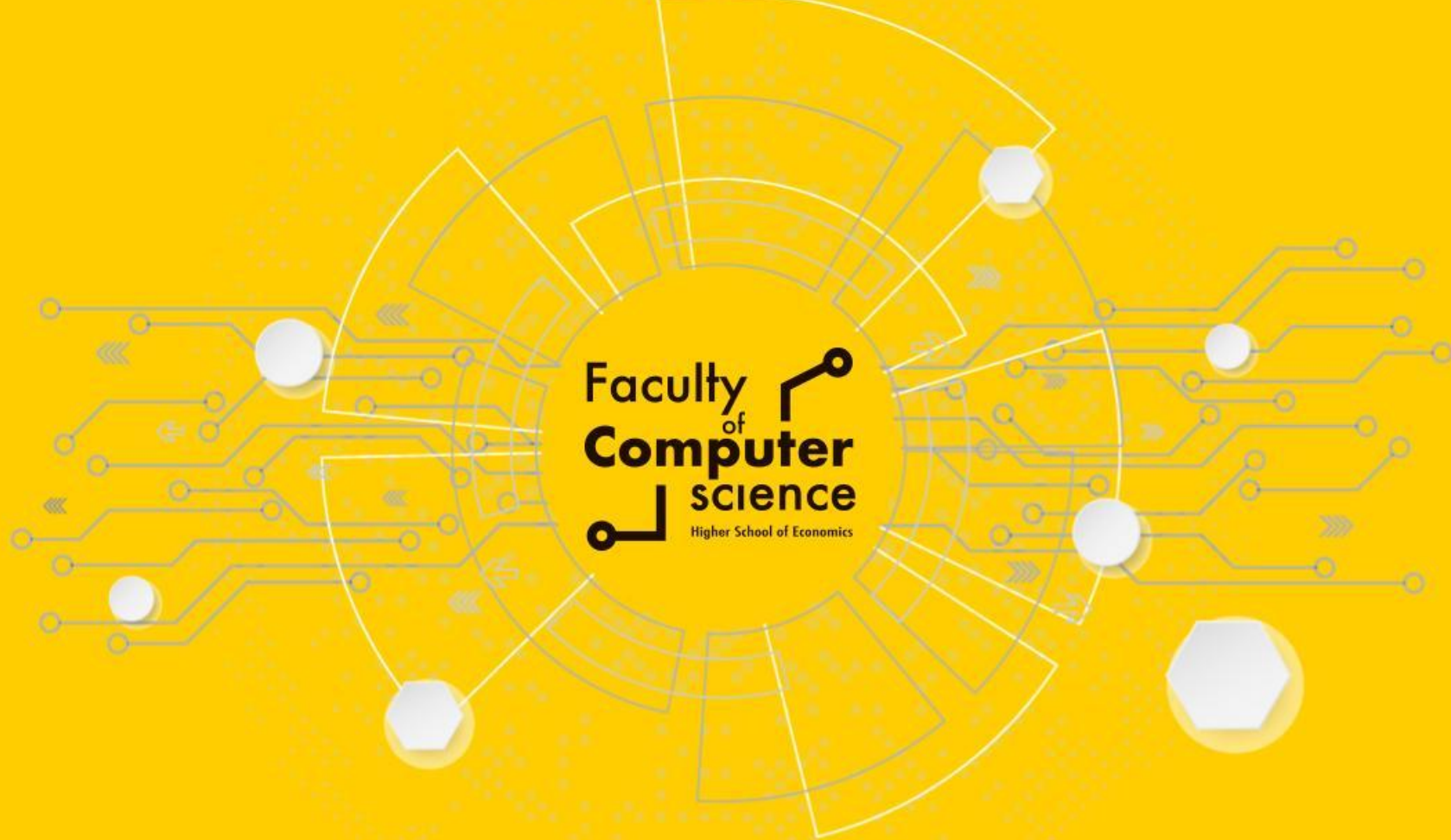


GitHub от Yandex research

<https://github.com/yandex-research/tabular-dl-tabr>

Оригинальная статья

<https://arxiv.org/abs/2307.14338>



mskazadaev@edu.hse.ru

CatBoost Feature Importance

