

WNS Analytics Wizard 2018

Прогнозирование повышения работника

Дмитрий Симаков

Никита Чуркин

Зачем?



Впервые заняли призовое место на индийской площадке

Хотим поделиться опытом участия и призываем участвовать вас

Особенности площадки I



Много участников (в среднем больше, чем на российских платформах)



72 часа на решение задачи



Разрешается много посылок в день (10-15)



Data-лики во многих задачах, использовать которые запрещено

Особенности площадки II



Нет форума и кернелов в «стиле kaggle» (есть канал в slack, участники могут шарить свой код посредством лидерборда после дедлайна)

1011

В конце соревнования надо загружать работающий код

Особенности площадки III



- <https://datahack.analyticsvidhya.com>
- <https://analyticsvidhya.slack.com>



- Ближайший хакатон: 18/11/2018 - AmExpert 2018
- Призы: “MacBooks and Interview Opportunities with American Express”

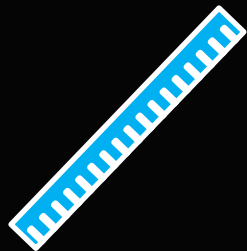
Особенности задачи



14 - 16 сентября



3846 участников (1300 отправили код на проверку)



Метрика – F1-Score, train 55к, test 23к, баланс классов 8.5%

Данные

- Департамент сотрудника
- Регион
- Уровень образования
- Пол
- Recruitment channel
- Количество тренингов (soft skills, hard skills и т.д.)
- Возраст сотрудника
- Количество отработанных в компании лет
- «Рейтинг» работника в пред. год
- KPI > 80%
- Факт получения наград
- Средний “training score”
- Таргет: рекомендован ли сотрудник к повышению

Топовые оригинальные признаки

- Департамент сотрудника
- Регион
- Уровень образования
- Пол
- Recruitment channel
- Количество тренингов (soft skills, hard skills и т.д.)
- Возраст сотрудника
- Количество отработанных в компании лет
- «Рейтинг» работника в пред. год
- KPI > 80%
- Факт получения наград
- Средний “training score”
- Таргет: рекомендован ли сотрудник к повышению

Инструменты

Использовали стандартный набор

Microsoft
LightGBM

+

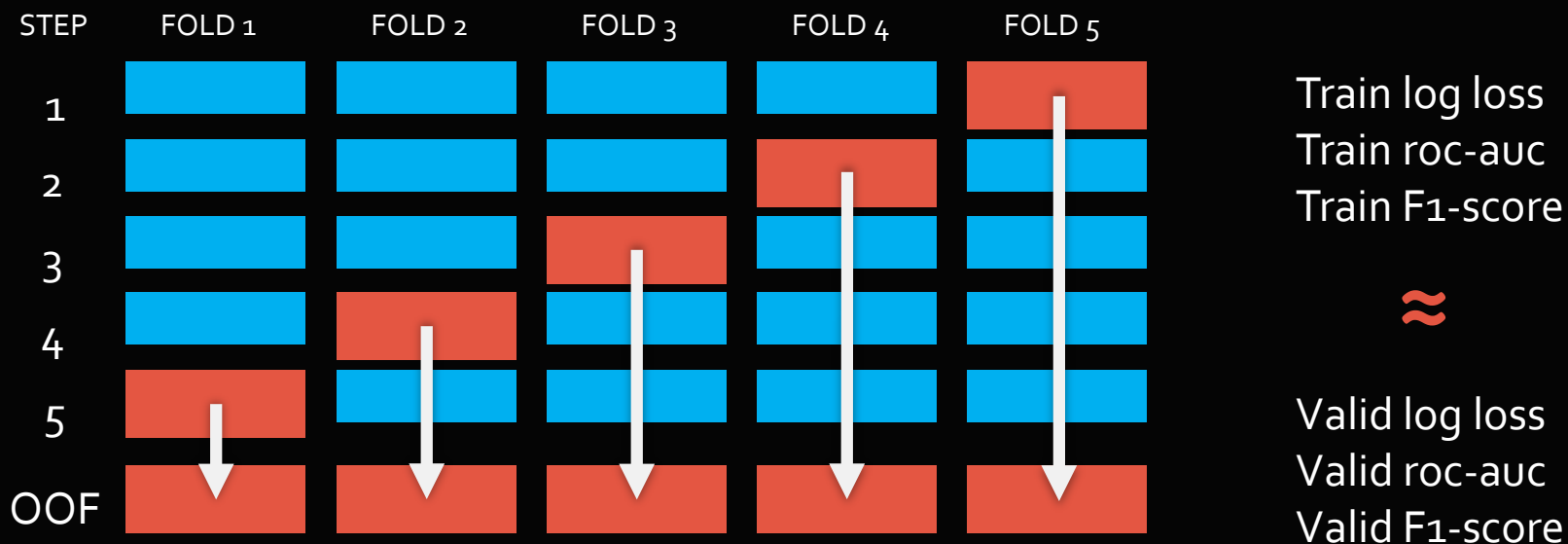


+

 PyTorch

Валидация*

Таргет очень несбалансированный: всего 8.5% положительных ответов



* Не используем стратификацию

Стараемся минимизировать разрыв

Общее решение

Модель = $\sim 0.25 \times \text{LightGBM } 5 \text{ CV}$ + $\sim 0.75 \times \text{LightGBM } 5 \text{ CV}$



21 признак, тюнинг
гиперпараметров руками

82 признака, тюнинг
гиперпараметров через
Bayesian Opt

Решение 1

- Вычисляем permutation importance от оригинальных признаков (метрика – log loss)
- Оставляем только «хорошие» признаки
- Генерируем арифметические комбинации признаков (+, -, *, /)
- Снова отбираем только «хорошие» признаки
- Сильно регуляризируем модель (мало листьев, относительно много l1 регуляризации)
- Не сработал target encoding
- Не сработало создание более сложных признаков, поиск аномалий и т.д.

Решение 2

- Обогащаем датасет огромным количеством разномастных новых признаков

12 признаков → **4.6K+** признаков

- Проводим отбор признаков

Рассмотрим это решение подробнее

Генерация признаков I

Все переменные – unsupervised – на объединенном train + test

- KNN
- Взаимодействия непрерывных признаков
- Простые интеракции категориальных и бинарных признаков
- Агрегаты и статистики
- Reconstruction error

Генерация признаков II

KNN

- K: 5, 10, 50, 200, 1000

Категориальные признаки:

- Совпадает ли класс
- Наиболее вероятный класс
- Вероятность наиболее правдоподобного класса
- Вероятность реального класса
- Расстояние до наиболее правдоподобного класса
- Расстояние до наименее правдоподобного класса

Непрерывные признаки:

- Расстояние до соседей
- Отношение к соседям
- Соседи

Генерация признаков III

Агрегаты

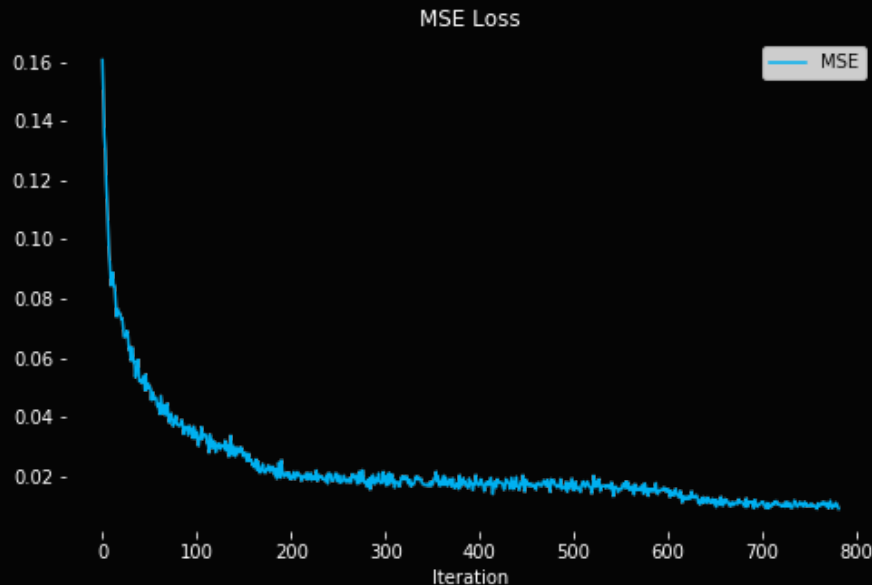
- Группировка по категориальной переменной
- Агрегаты min, max, mean, sum, count для непрерывных внутри группы
- Отношения текущих значений к агрегатам по группе
- Расстояния текущих значений и агрегатов

4290 признаков

Генерация признаков IV

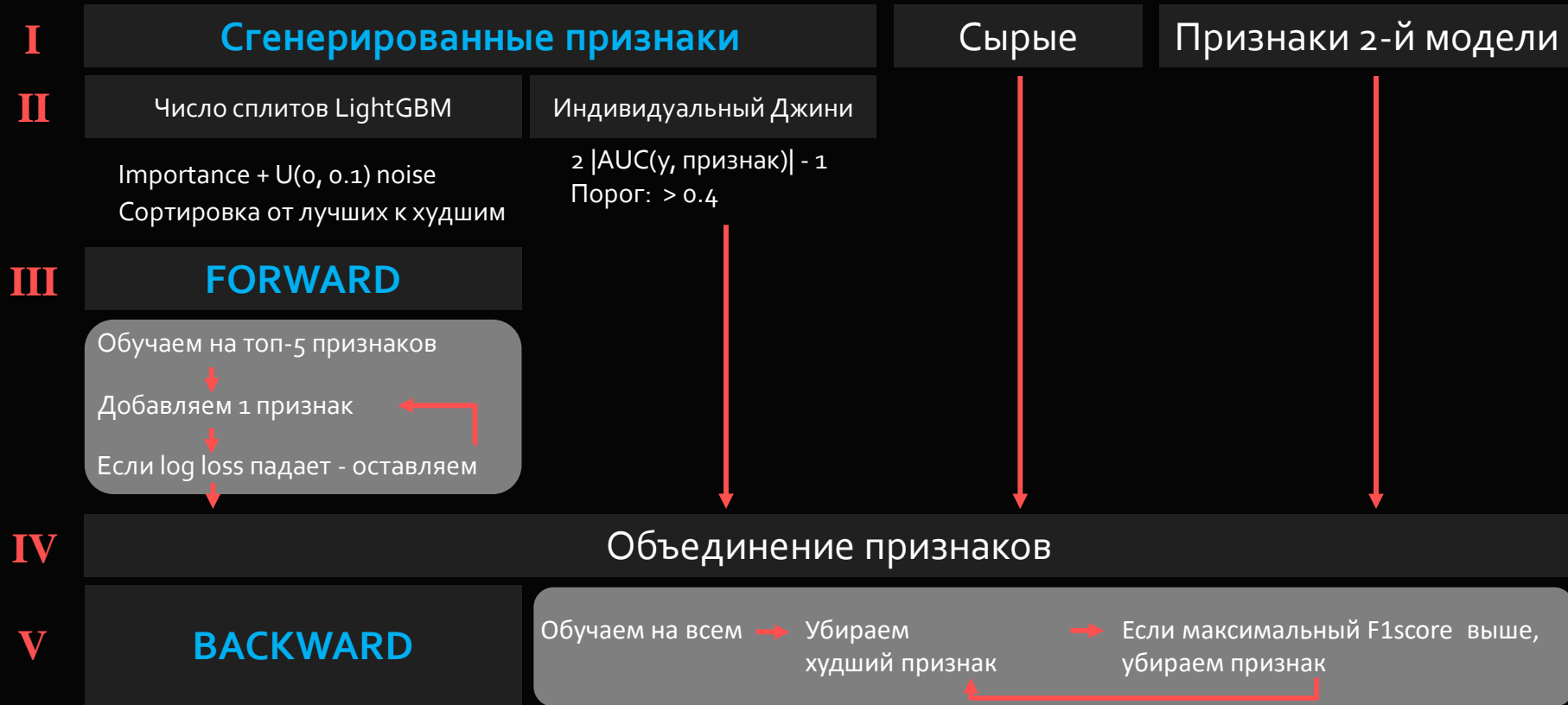
Автоэнкодер

- Исходные признаки
- NaN на -1
- MinMax scaling
- 3 линейных слоя (размерность / 2)
- MSE
- Адам, lr 0.02, bs 200
- 2 эпохи
- Reconstruction Error (как MSE и MAE по строке)



2 признака

Отбор признаков I



Отбор признаков II

FORWARD

- + Сильно сокращает признаки
- + Относительно быстр
- + Оптимизирует любую метрику
- Чувствителен к порядку
- Нестабилен
- Качество обычно незначительно падает

BACKWARD

- + Качество обычно выше
- + Оптимизирует любую метрику
- Чувствителен к порядку
- Долгий
- Удаляет мало признаков

Отбор признаков III

Важен правильный порядок!

Наш выбор **Permutation Importance**

Преимущества:

- + Интерпретируемый
- + Считается по любой метрике
- + Обрабатывает коррелирующие фичи
- + Относительно вычислительно легок*

Недостатки:

- Большое число неподвижных точек
- 0 PI у переменных с числом сплитов = 0
- Плох, если мало уникальных значений
- Все же недостаточно быстр

*: относительно LOCO; на 4.5к фичах, 40к наблюдений, 1 перестановка, 2 фолда – 18 часов на 50 потоках, можно считать группами переменных, можно выкидывать с нулевым количеством сплитов

Лучшие признаки I

Группа	% от числа	% в модели	% по вкладу
Исходные	100	15.6	4.4
KNN	3.6	15.6	10.2
Агрегаты	0.9	48.6	71.4
Интеракции	2.8	4.7	3.6
RE	100	3.4	1.8
Из другой модели	100	12	8.4

Лучшие признаки II

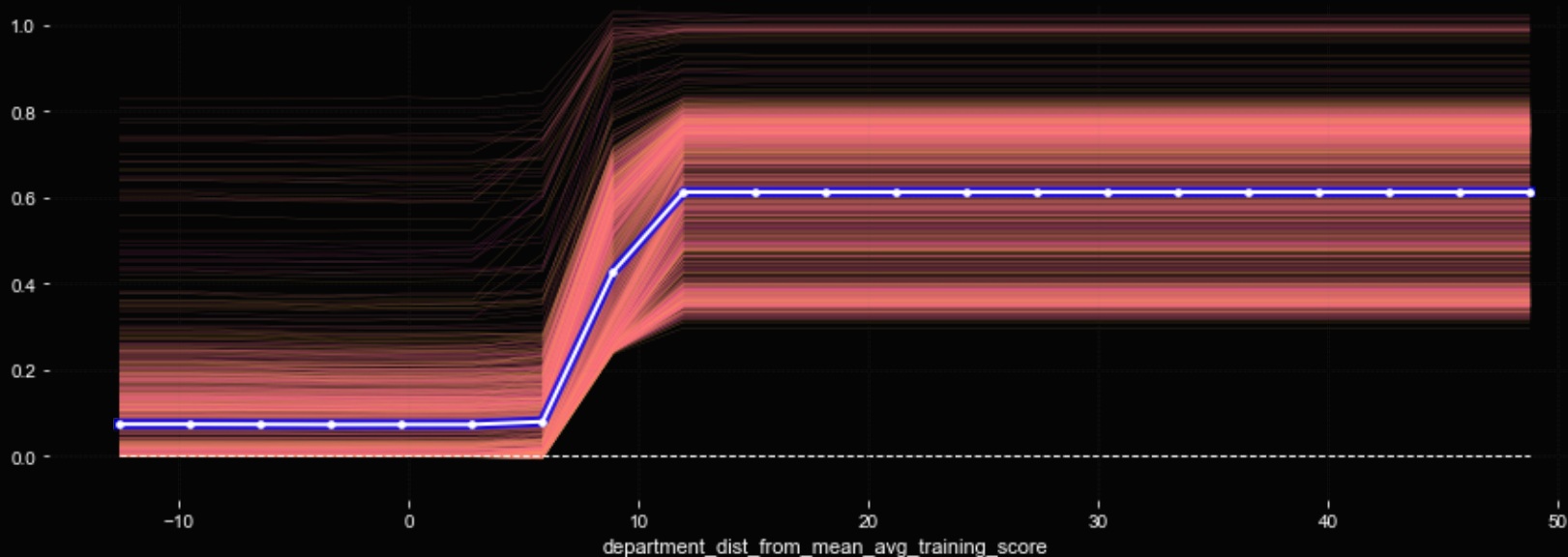
PI F1 score	Переменная
0.074664	Расстояние среднего training score до среднего по департаменту
0.014005	Расстояние среднего training score до минимального по департаменту
0.005294	Вероятность «awards won?» по 200 ближайшим соседям
0.003428	Средний рейтинг по департаменту
0.003272	Средний training score, умноженный на рейтинг за предыдущий год

Что можем еще сделать?

Можем посмотреть, насколько предсказания согласуются с логикой!

PDP-ICE

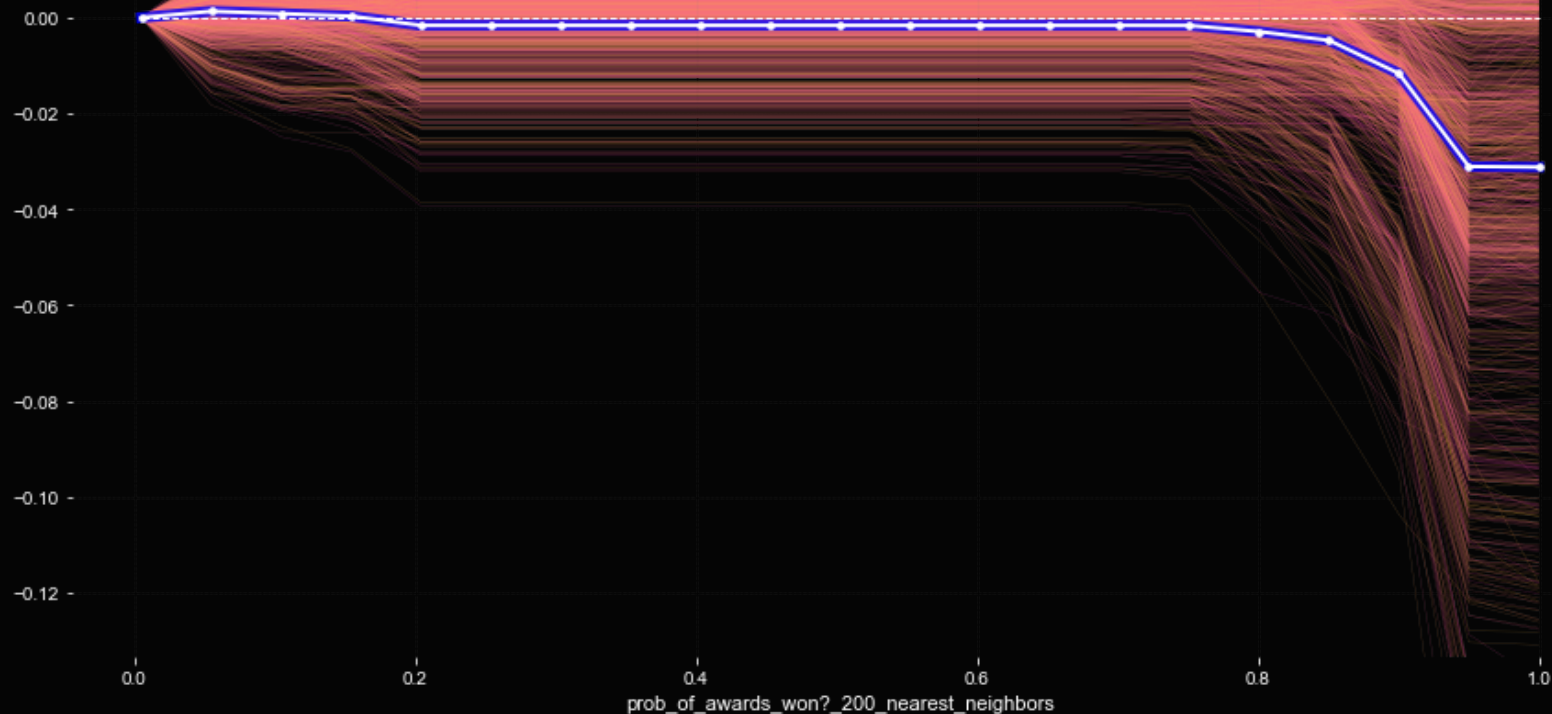
Расстояние среднего training score до среднего по департаменту



Если твой скор сильно лучше среднего – то тебя вероятнее повысят

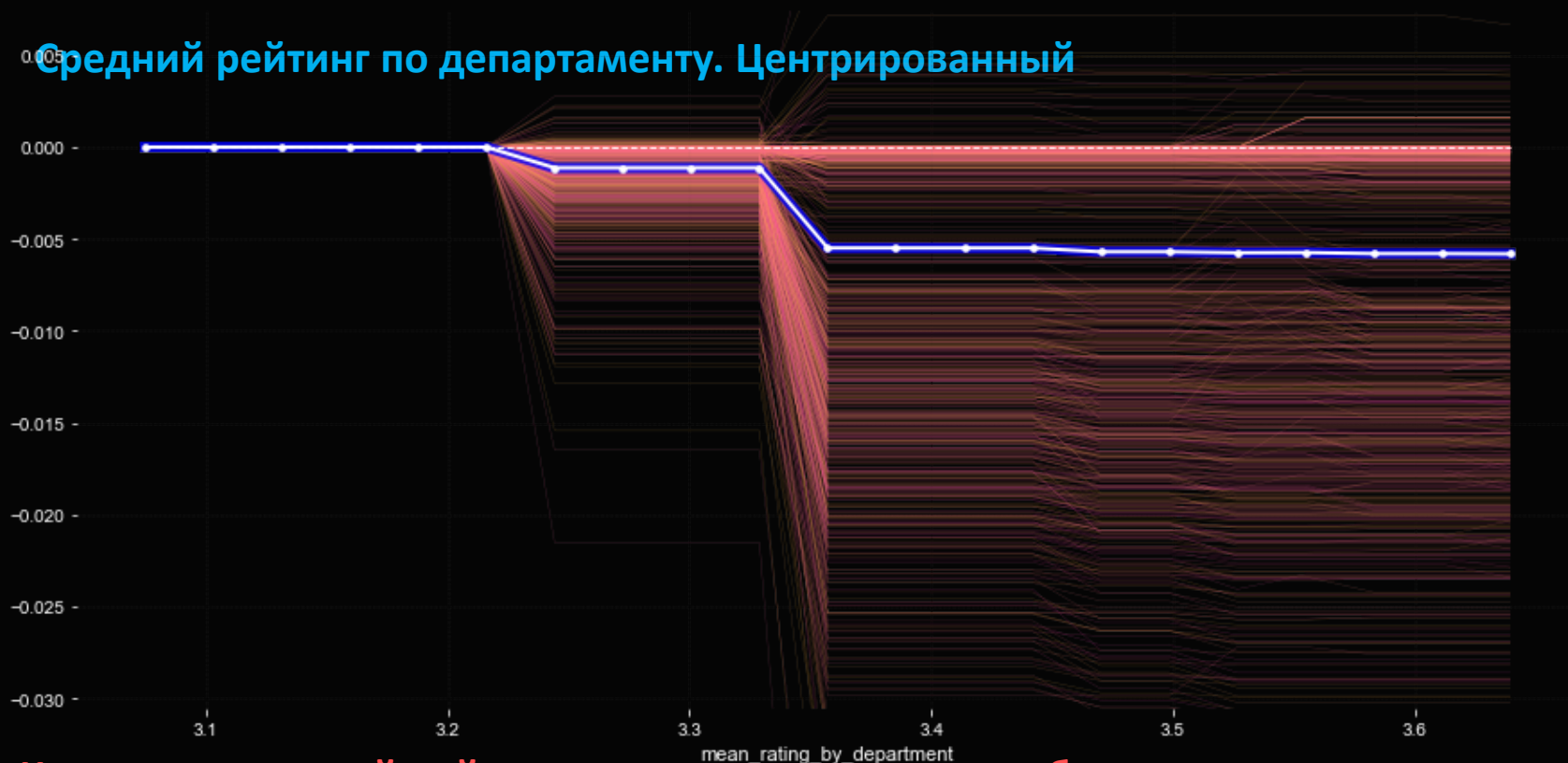
PDP-ICE

Вероятность «awards won?» по 200 ближайшим соседям. Центрированный



Если твои «соседи» выигрывали призы – то тебя менее вероятно повысят

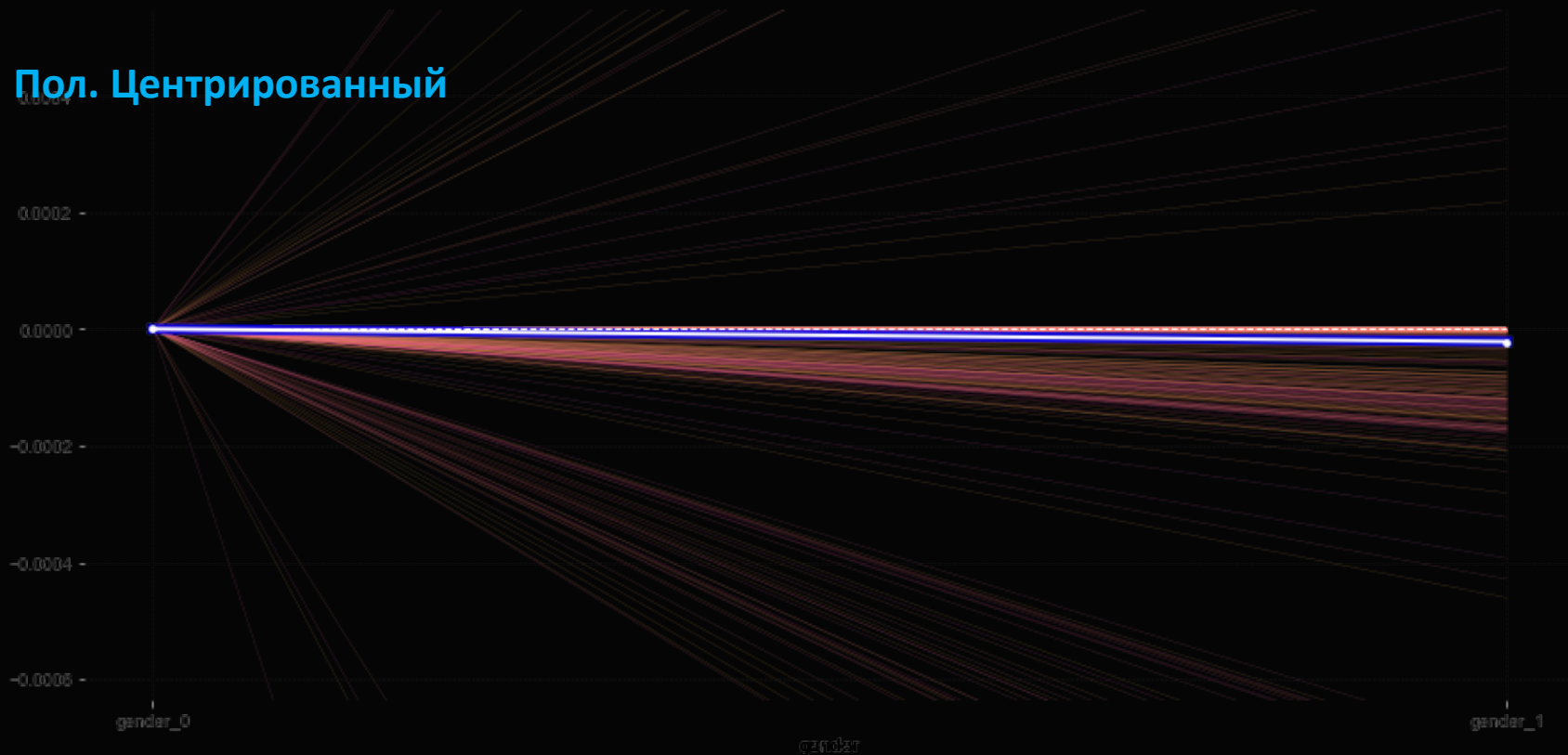
PDP-ICE



Чем выше средний рейтинг в департаменте – тем тебя менее вероятно повысят

PDP-ICE

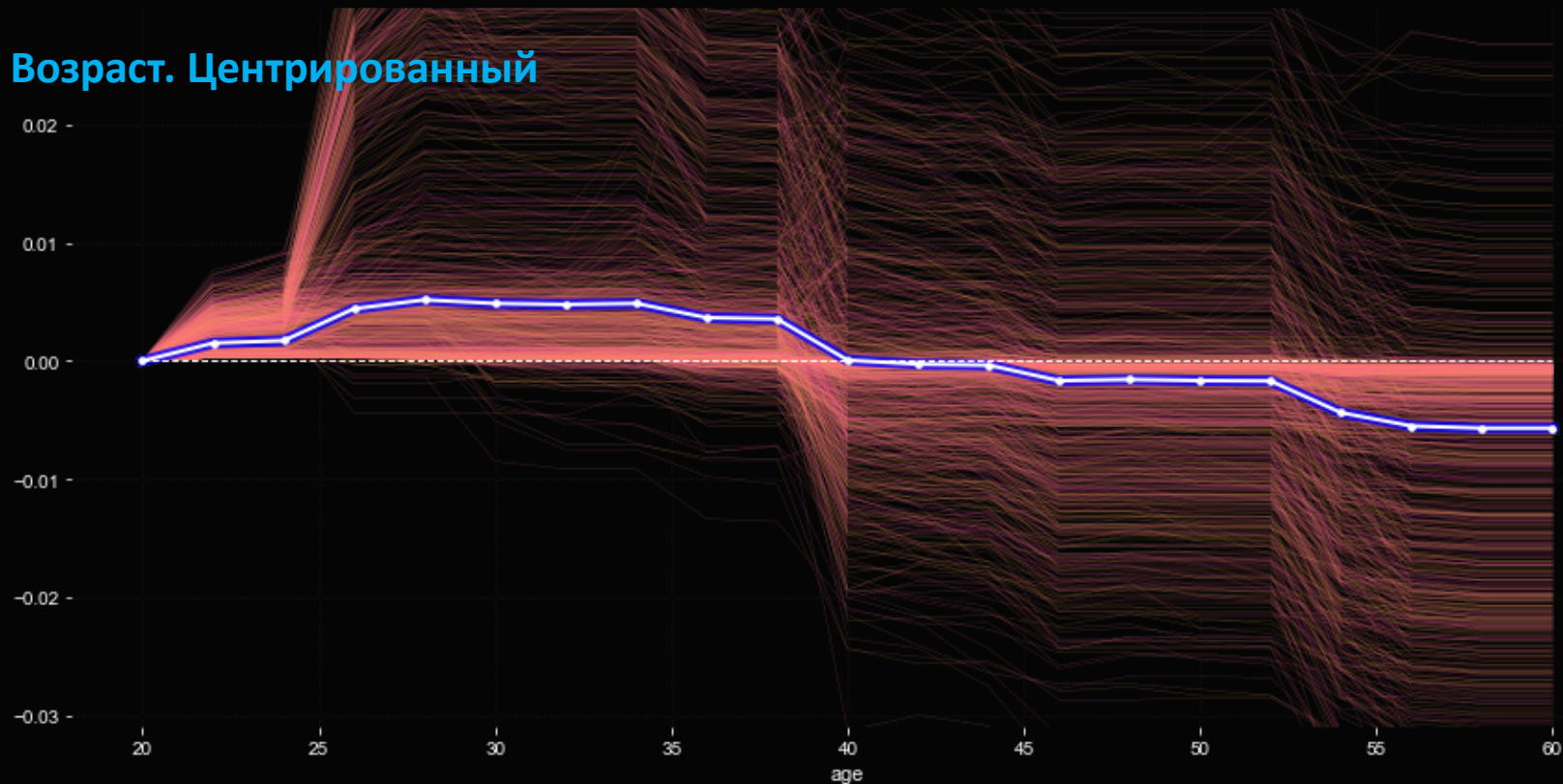
Пол. Центрированный



Нет дискриминации по полу!

PDP-ICE

Возраст. Центрированный



А вот по возрасту немного, но дискриминируют.

Выставление меток

Подбор веса для смешивания и cut-off: по out-of-fold предсказаниям

Алгоритм:

- Смешиваем два oof предсказания с весом W
- Выставляем cutoff
- Считаем F-score
- Максимизируем метрику с помощью scipy по двум параметрам












Советы

- Для таких непродолжительных конкурсов нужно иметь готовый код, чтобы быстрее проверять идеи
- Не стоит доверять публичному ЛБ слишком сильно в задачах классификации с дисбалансным таргетом
- В подобных задчах аккуратно подбирайте гиперпараметры
- Всегда пробуйте отбор признаков
- Пишите чистый, воспроизводимый код

Приватный ЛБ

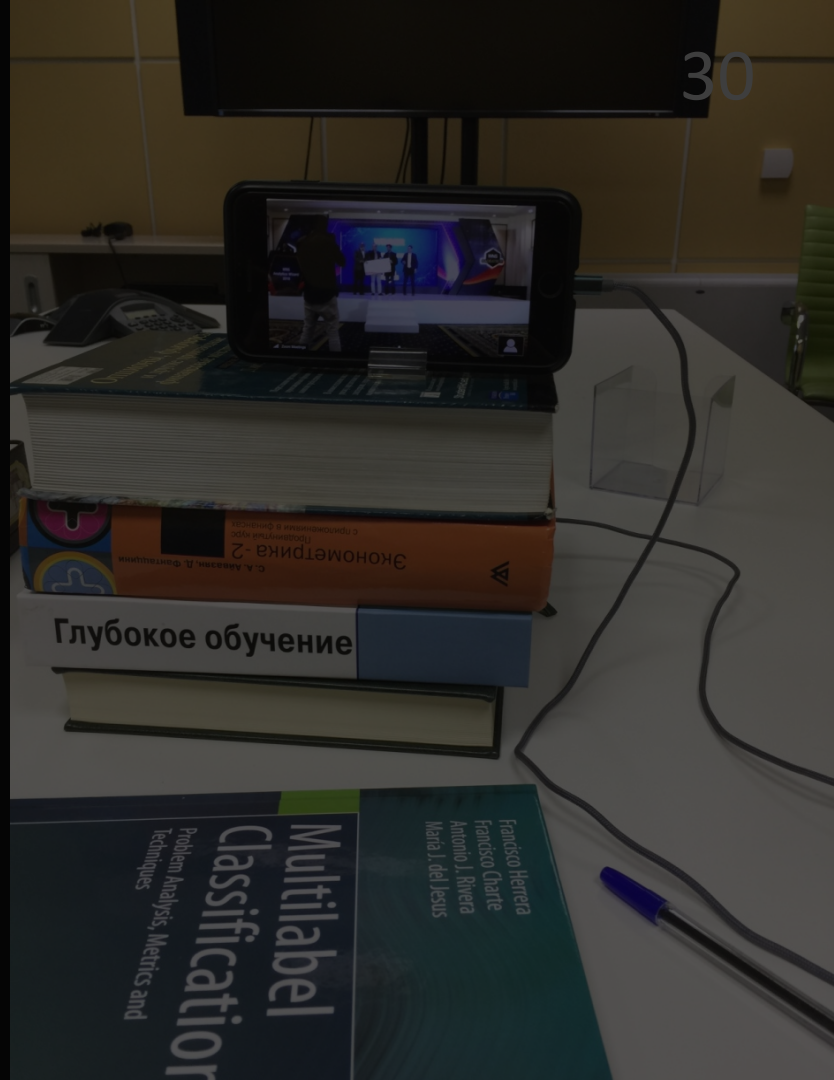
Вжух: 59 -> 2

Private Leaderboard - WNS Analytics Wizard 2018 (Machine Learning Hackathon)

	#	Name	Score	Participant's Code	Participant's approach
 0	1	 siddharth3977	0.5346232179	---	
 0	2	 Cheburek	0.5338386573	---	
 0	3	 kamakals	0.5333333333	---	
 5	4	 tearth	0.5318107667	---	Read Participant's approach 
 0	5	 Deviants	0.5313531353	---	

Церемония награждения

- Проходила в Индии
- Мы связывались через Zoom
- Организаторы проводили несколько «репетиций» с нами перед трансляцией
- Обещали интервью с нами
- В итоге просто «вручили» чек через экран



Подход Дениса Воротынцева



Денису удалось занять 4-е место в данном конкурсе

Он изложил свое решение в посте на Medium:

<https://towardsdatascience.com/how-to-save-hrs-time-with-machine-learning-b6f2226b789d>

Спасибо за внимание

ODS Slack: Дмитрий Симаков
Mail: dmitryevsimakov@gmail.com

НИУ ВШЭ: ФЭН, ФКН
ПАО Сбербанк

ODS Slack: nikita_churkin
Mail: nikita1994175@ya.ru

МГУ: ММ; НИУ ВШЭ: ФКН
ПАО Сбербанк

И вдруг кому-то интересно...

Валидация

train log_loss: 0.1507
train roc_auc: 0.9343
train f1_score: 0.5766



oof log_loss: 0.1629
oof roc_auc: 0.9120
oof f1_score: 0.5341

Model 2

Блэнд

f1_score: 0.537
recall_score: 0.422
precision_score: 0.737

```
{'app': 'binary',  
 'learning_rate': 0.01,  
 'num_leaves': 10,  
 'feature_fraction': 0.5,  
 'lamda_l1': 3,  
 'metric': 'binary_logloss'}
```

 Model 1

```
{'bagging_fraction': 0.9458564289234204,  
 'feature_fraction': 0.5004666960515116,  
 'lambda_l2': 0.022577930769472343,  
 'min_data_in_leaf': 99,  
 'num_leaves': 13}
```

Model 2