Applied AI Challenge

Fintech Case
8-21 April 2024
Online

НЦКР ИТМО точка
сильный ИИ в промышленности ИТМО ai

План Хакатона

- Нужно было решить проблему предсказания сочетаний «продавец – потенциальный клиент»
  - Метрикой был f1-score, в сумме было 42 различных признака, включающие в себя признаки клиента, продавца, компании и другие.
  - Первый этап – анализ данных на пропуски, выбросы и потенциальные новые фичи.
  - Второй этап – преобразование данных на основании результатов первого.
  - Третий этап – тренировка и оптимизация модели.

Что было сделано (1)

- Анализ данных
  - Именно пропуски встретились только в столбце ‘time_tz_diff’, в остальных отсутствующие значения заменялись либо на ‘unknown’, либо на значения вне ожидаемого интервала (например, **employed_days** == -1).
  - Большая часть признаков имела незначительную корреляцию с целевой переменной, значения выше 0.01 в основном наблюдались у конкретных значений категориальных признаков.
  - Общее число признаков после OneHot кодирования общей матрицы составляло 771
  - Соотношение числа положительных взаимодействий к негативным – 1:84

Что было сделано (1)

position_position_51	0.045185
egrul_reg_months_ago_0	0.033466
role_role_9	0.030693
grade_grade_0	0.028378
grade_grade_18	0.027848
grade_grade_10	0.027224
division_division_8	0.027097
has_new_phones	0.020949
position_unknown	-0.020236
position_position_33	-0.023484
grade_grade_12	-0.027645
days_since_ors_avg_unknown	-0.029192
role_role_3	-0.030474
grade_grade_4	-0.033204

Name: target, dtype: float64

Топ по корреляции с целевой переменной

Что было сделано (2)

- Работа с данными
  - Для каждого блока информации были созданы новые признаки, средняя корреляция которых с целевой переменной составляла 0.01
  - Были удалены признаки, у которых наблюдалась слабая корреляция и множественные пропуски, в результате которых получилось снизить количество признаков с 771 до 388, что ускорило процесс обучения в 2.3 раза без значительных потерь в качестве.
  - Удалены признаки, у которых стандартное отклонение равнялось 0 (константные).
  - Стратифицированное разбиение на тренировочную, валидационную и тестовые выборки.

Что было сделано (3)

- Тренировка модели
  - В качестве основной модели было решено взять CatBoostClassifier, бейзлайном выступало оригинальное решение с весами классов, обратными к их частоте (1, 84), публичный скор которого составил 0.05.
  - Модель тренировалась на train части выборки, оптимальные параметры находились при помощи библиотеки optuna, оптимизация происходила по результатам модели на валидационной выборке.
  - Сначала были найдены оптимальные веса классов – 1 и 25 для негативного и положительного соответственно, это позволило увеличить публичный скор до 0.065.
  - Затем было замечено, что модель быстро (200-300 итераций) оверфитит тренировочную выборку, о решении этой проблемы дальше.

Что было сделано (3)

- Уменьшение переобучения
  - Для определения того, что модель переобучилась на тренировочной выборке я использовал встроенный в CatBoost overfitting detector, который завершал обучение, если в течение фиксированного числа итераций не наблюдалось улучшения на валидационном множестве.
  - После этого я применил общие методы уменьшения переобучения для алгоритмов градиентного бустинга: уменьшение глубины дерева и увеличение коэффициента L2 регуляризации.
  - Кроме того, был использован параметр random_strength, который к скорам разделений в дереве прибавляет случайную величину с нулевым матожиданием и уменьшающейся во время ожидания дисперсии, которая и умножается на этот коэффициент.

Завершение

- Финальные штрихи
  - Итоговые параметры были подобраны через общую оптимизацию оптуны, для репродукции результатов сид модели был зафиксирован. Таким образом и был получен последний скор, который хорошо себя показал и на публичной, и на приватной части соревнования.
  - Самым важным решением оказалось удаление лишних признаков, без этого не получилось бы добиться такой точности решения из-за большего времени на обучение и оптимизацию.