GeekBrains

Мой первый ML проект по задаче классификации

.

Подтема

Предсказание невыполнения кредитных обязательств.

Дмитрий Яковлев

Закончил факультет искусственного интеллекта GeekBrains..
Закончил курсы повышения квалификации в МГТУ им. Баумана.

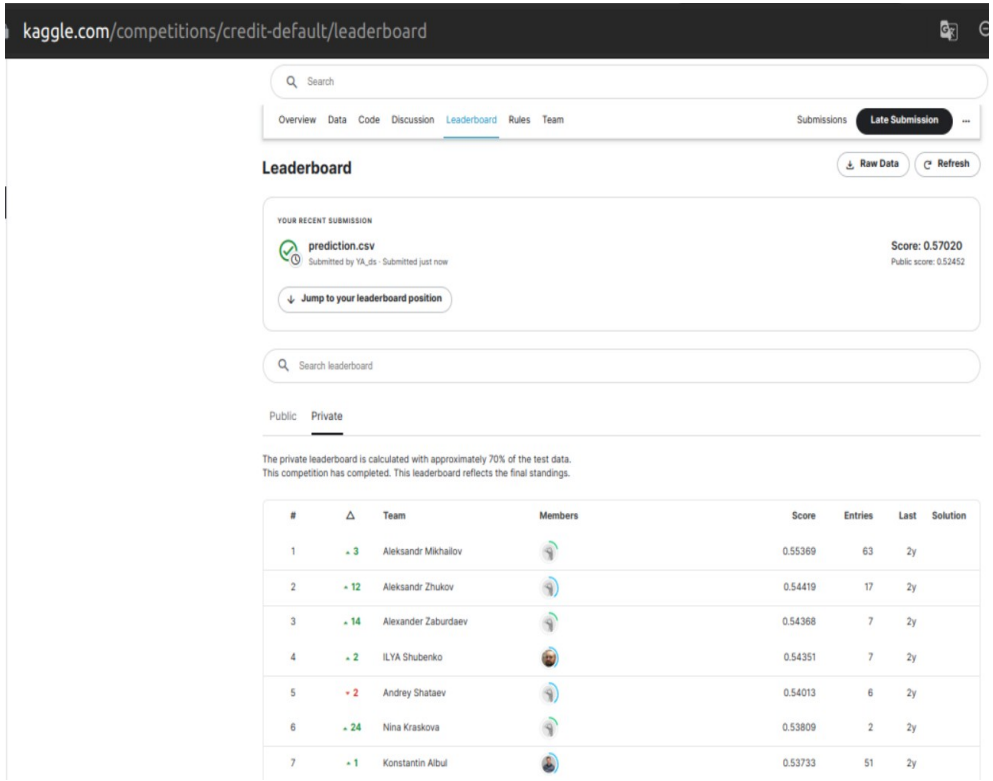
Немного о себе.

Инженер по работе с клиентами и проектировщиками на предприятии, производящем высоковольтное оборудование.

План проекта

1. Пройденный этап: DataPreprocessing
2. Пройденный этап: EDA
3. Пройденный этап: FeatureGeneration
4. Пройденный этап: настройка регрессоров
5. Пройденный этап: создание общего пайплайна
6. Пройденный этап: создание product ready приложения
7. Будущий этап: доработка проекта
9. Будущий этап
10. Будущий этап
11. Будущий этап
12. Будущий этап
13. Будущий этап
14. Будущий этап
15. Будущий этап
16. Будущий этап

Достигнутые цели



Удалось найти наилучшее предсказание на лидерборде соревнования на Каггл по этому проекту.

Поставил для себя задачу достичь результата на базовом классификаторе (на небольших датасетах лучше всего Catboost) и без фиксации зерна рандомных функций. random_state при разделении тренировочного датасета пришлось зафиксировать, так как лидерборд не достаточно хорошо валидировался кроссвалидацией модели обучения. Пришлось train.csv делить на тренировочную и валидационную выборки. Но в классификаторе, хоть и забыл убрать random_state, подбирал гиперпараметры внутренним гиперпараметром Катбуста «use_best_model», а им манипулировать не возможно.

Решение задачи / План работы

- Иследование данных на платформе Kaggle.
- Исследование рабочего датасета: поиск аномалий, анализ распределений, корреляций, исследование методов работы с количественными и категориальными признаками, исследование выбросов, поиск оптимального алгоритма построения классификатора, определение полезных признаков.
- Генерация новых признаков, исследование их корреляции, попытка нахождения полезных кластеров.
- Разработка классификатора.

Трудности

Очень значимое количество пропусков в признаках. Попытка заполнения пропусков с помощью ML-моделей или KNN_Imputer привела к искажению распределений данных, подгонке данных под решатель. Поэтому пытался заполнить пробелы используя статистику. Выбросы для задачи классификации не так важны как для регрессии, тем не менее в этой задаче в выбросах имелась дополнительная информация.

Также в датасете имеется значительный дисбаланс классов по целевому признаку, который удалось исправить нахождением большого кластера с нулями по целевому признаку, генерацией искусственных наблюдений с помойщью нейронной сетки GAN и методом undersampling.

Проблема для этого датасета оказалась также в не возможности найти более-менее значимые дополнительные признаки. Судя по низкой метрике качества, не удалось их найти и другим участника соревнования. Возможно это связано с большим количеством пропусков в наиболее значимых признаках, и восстановить исходное распределение по этим признакам очень трудная задача.

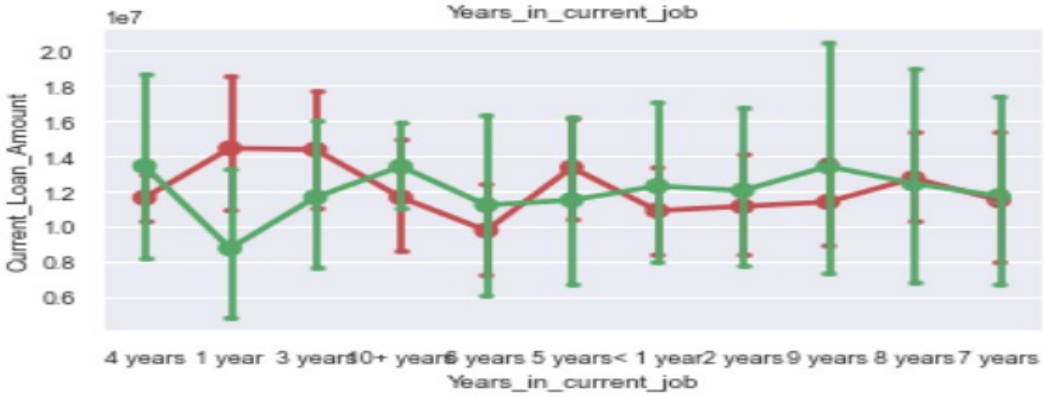
Были небольшие трудности с валидацией. Одна лишь кроссвалидация не справилась с этой задачей. Потребовалась ее слаженная работа с выделенной валидационной выборкой. Тем не менее валидацию удалось наладить с помощью обычной функции train_test_split. Чем проще решение тем надежней.

Ну и конечно проблему составляет небольшое количество признаков в датасете. Если сравнивать с одним из лучших по этой теме соревнований «Home Credit Default Risk», то здесь на порядок меньше признаков.

Baseline

Более-менее качественного baseline-решения для этого датасета не нашлось. Самое качественное деление таргета происходит по признаку Credit score, а конкретно по пропущенным значениям в этом признаке, по которым таргет равен единице.

EDA



Проверил все признаки на пропущенные значения, соответствие распределений на трейне и тесте, на распределение значений внутри категориальных признаков, на выбросы и наличие взаимной информации между признакми и таргетом.

FeatureGeneration

Полезным оказался только признак, разделяющий признак “Credit score” на действительные значения и пропущенные. В пропущенных значениях целевой признак принимает значение 1. Видимо в пропущенных значениях имеется в виду, что у клиента кредитный рейтинг отсутствует.

Только наличие этого признака в модели почти обнуляет feature importance классификатора остальных признаков. Руководствуясь этим и тем, что бустинговые модели очень чувствительны к глубине деревьев, я разделил датасет по этому признаку вручную.

Выбор класификатора

```
model = catb.CatBoostClassifier(iterations=12000,
                                grow_policy="Depthwise",
                                #loss_function = 'RMSE',
                                eval_metric = 'AUC',
                                early_stopping_rounds = 1000,
                                random_state=42,
                                cat_features=cat_feats,
                                #task_type="GPU", # закомменти
                                #devices='0', # закомментирова
                                verbose = False,
                                use_best_model=True
                               )
```

В качестве базового решения выбрал решатель CatBoostClassifier, и не менял его до самого конца, Катбуст в качестве базового классификатора интересен тем, что более устойчив относительно других бустеров без подбора гиперпараметров. А также тем, что имеет достаточно качественный внутренний энкодер для категориальных признаков. Подбор гиперпараметров осуществлял внутренним гиперпараметром Катбуста «use_best_model». Более продвинутые методы не пригодились. eval_metric = 'AUC' использовал, потому что она не зависит от порога, в отличие от F1.

Предложения по проекту

В первую очередь более качественно проработать предобработку данных. Из-за отсутствия опыта я его не достаточно качественно проработал, а при доработке проекта не стал переделывать, т.к. в этом случае проект нужно делать с нуля. Проработать решения на базе нейронных сетей, но это тоже надо с нуля прорабатывать решение, т.к. действующее решение ориентировано на решающие деревья. Усложнить структуру решателя. Проработать сложный таргет энкодинг.