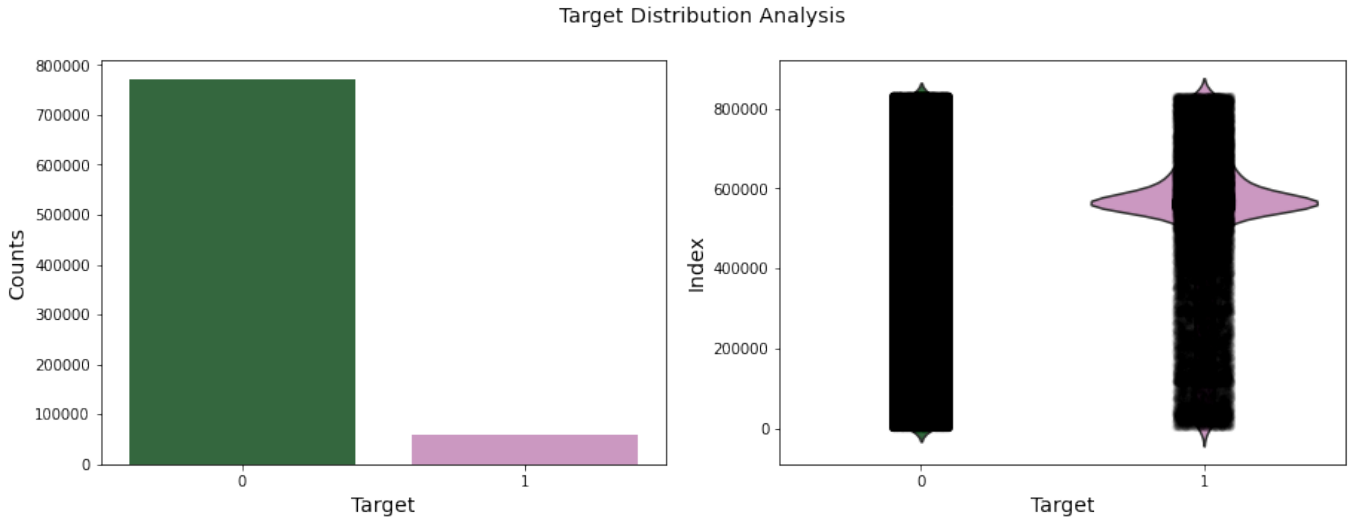
EDA

Анализ целевой переменной

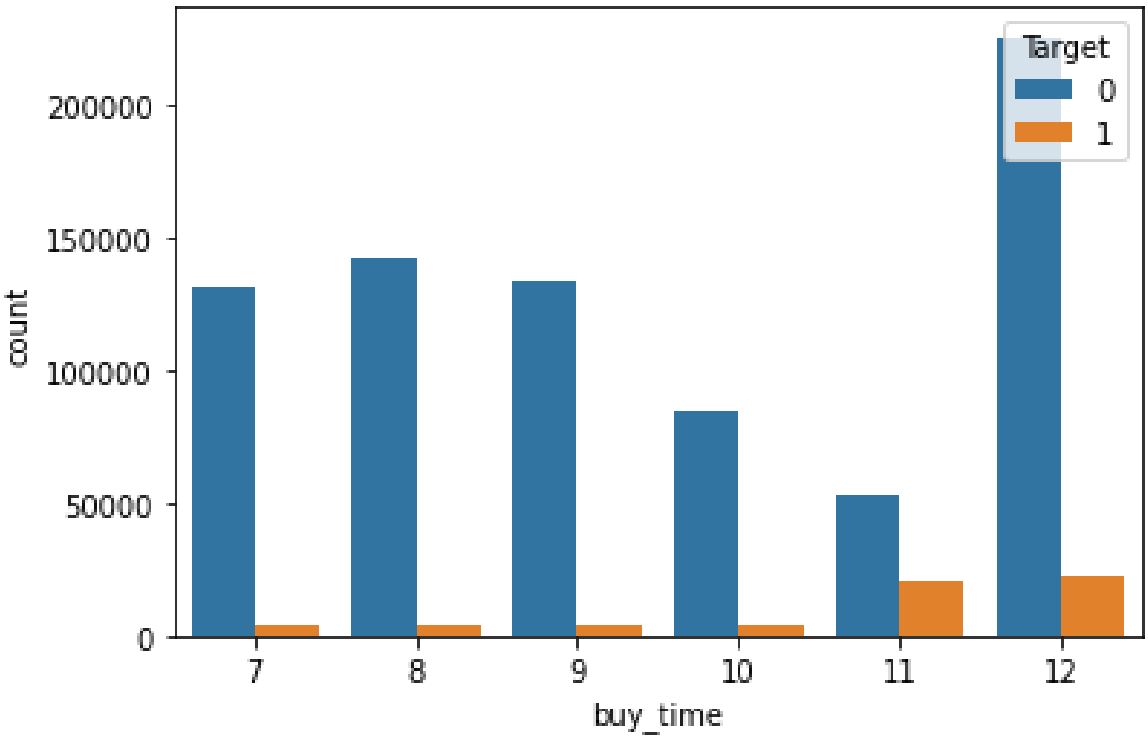


Выводы о целевой переменной:

- Мы будем решать задачу бинарной классификации с дисбалансом целевой переменной. Количество клиентов, которые подключили услугу, намного меньше, чем клиентов, которые не подключили.

- На правом графике мы видим, что целевая переменная распределена равномерно по индексам датафрейма, кроме одного периода. Возможно здесь присутствует сезонная компонента.

Анализ переменных, связанные со временем



Временные тренировочные переменные представляют один год 2018, предложения делали по воскресеньям, в тренировочном сете 6 месяцев в тестовом данные за 2019год за первый месяц.

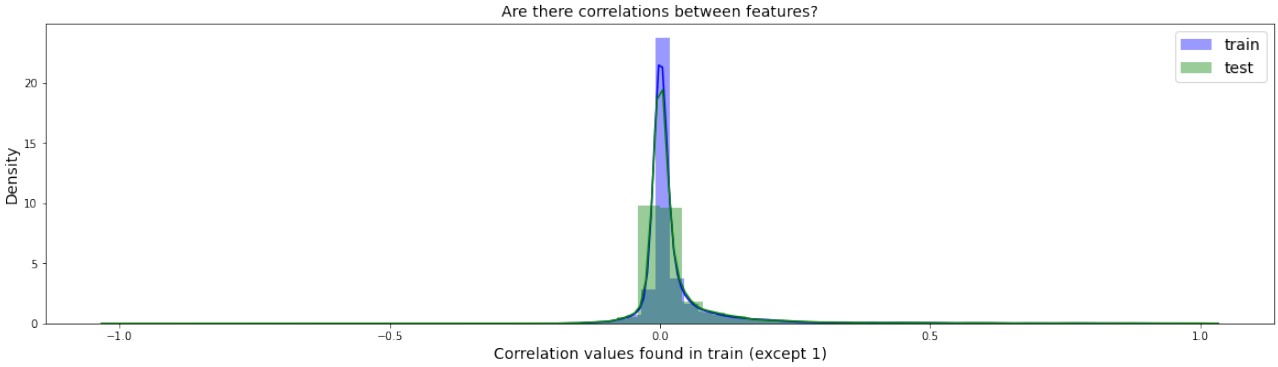
Анализ дискретных признаков показал, что их 44 и признаки 139,203 содержат в себе константы. Такие признаки мы удаляем. 20 признаков является бинарным. Мы их можем поменять на 0,1 и сделать их категориальными.

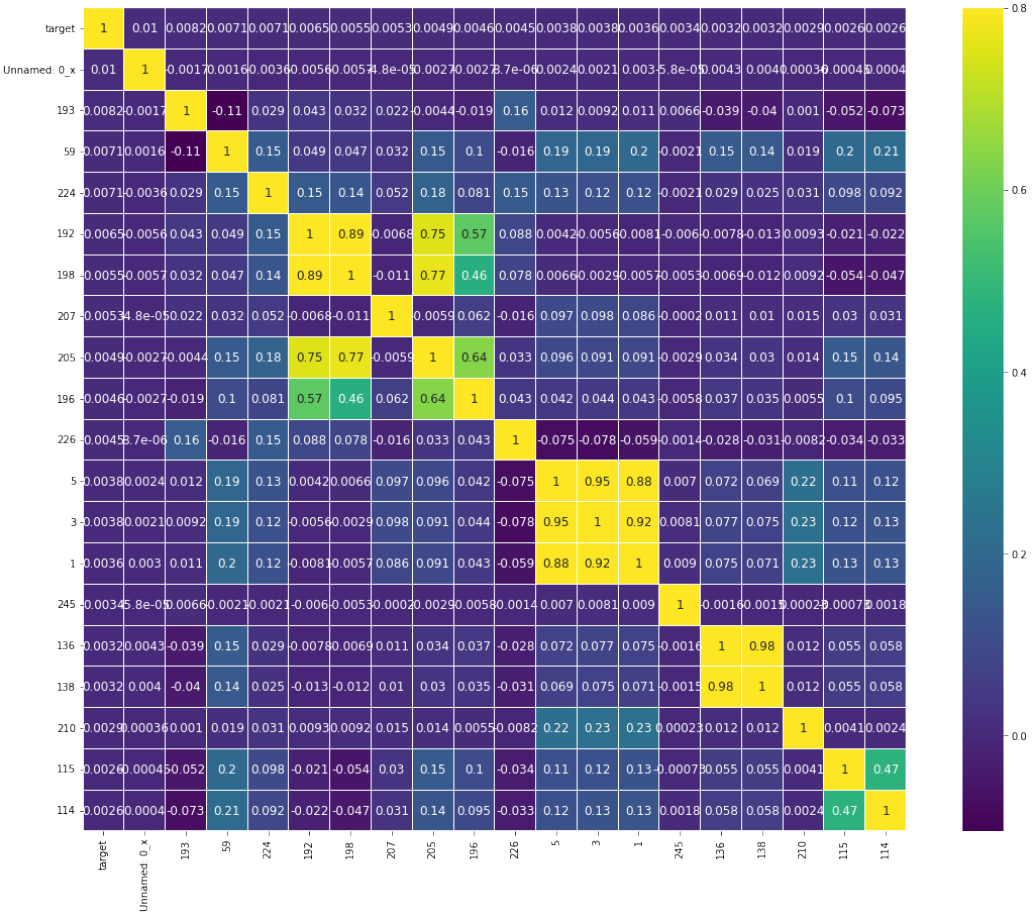
Анализ непрерывных признаков показал, что большинство признаков имеют равномерное распределение, что может негативно сказаться на распределяющей способности деревьев.
Распределение данных на тренировочной и тестовой выборке совпадает не по всем признакам.

Из категориальных признаков у нас только vas_id.

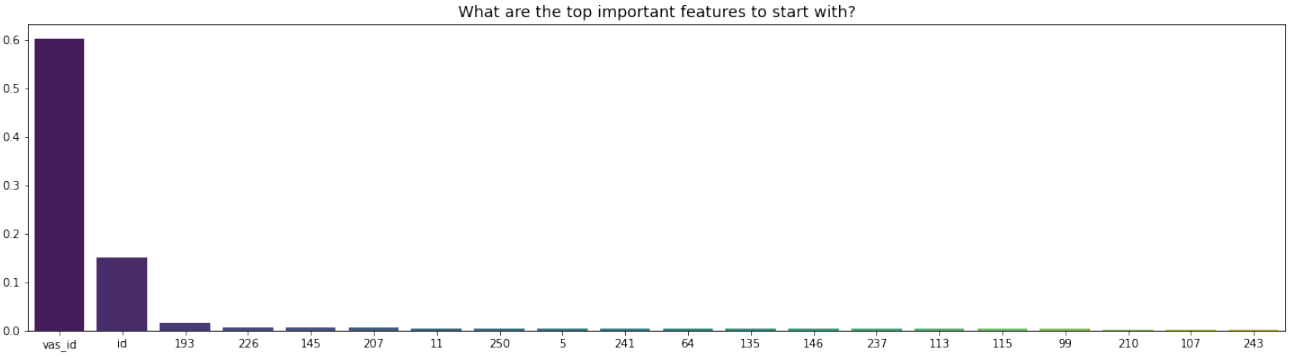
Тренировочные и тестовые дата сеты пропусков не имеют.

Линейная корреляция





Нелинейная связь между признаками



Короткие выводы:

- Между признаками и target отсутствует сильная линейная связь

- Распределение корреляции между тренировочными и тестовыми данными не совпадают

- Высокая линейная корреляция наблюдается между признаками 5,3,1 и 136, 138. Следовательно, в нашем анализе нам нужна только одна из этих переменных (мы можем оставить 5, поскольку ее корреляция с target выше и 5).

- Наибольшей нелинейной связью с целевой переменной обладают vas_id, id и 193.

Выбор модели

Отбор моделей производился между LGBMClassifier, CatBoostClassifier и XGBClassifier. Наилучший результат показал CatBoostClassifier.

	Y	Model_idx	Model_name	f1	CalcTime
0	target	0	<class 'lightgbm.sklearn.LGBMClassifier'>	0.675345	39.764027
0	target	1	<class 'catboost.core.CatBoostClassifier'>	0.691931	265.911360
0	target	2	<class 'xgboost.sklearn.XGBClassifier'>	0.659387	389.774913

Отбор признаков и параметров

Так как у нас большой дисбаланс в таргете, то мы применяем параметр scale_pos_weight в CatBoostClassifier. Для отбора признаков я использовал eli5, который использует метод permutation_importance.

Топ 20 признаков

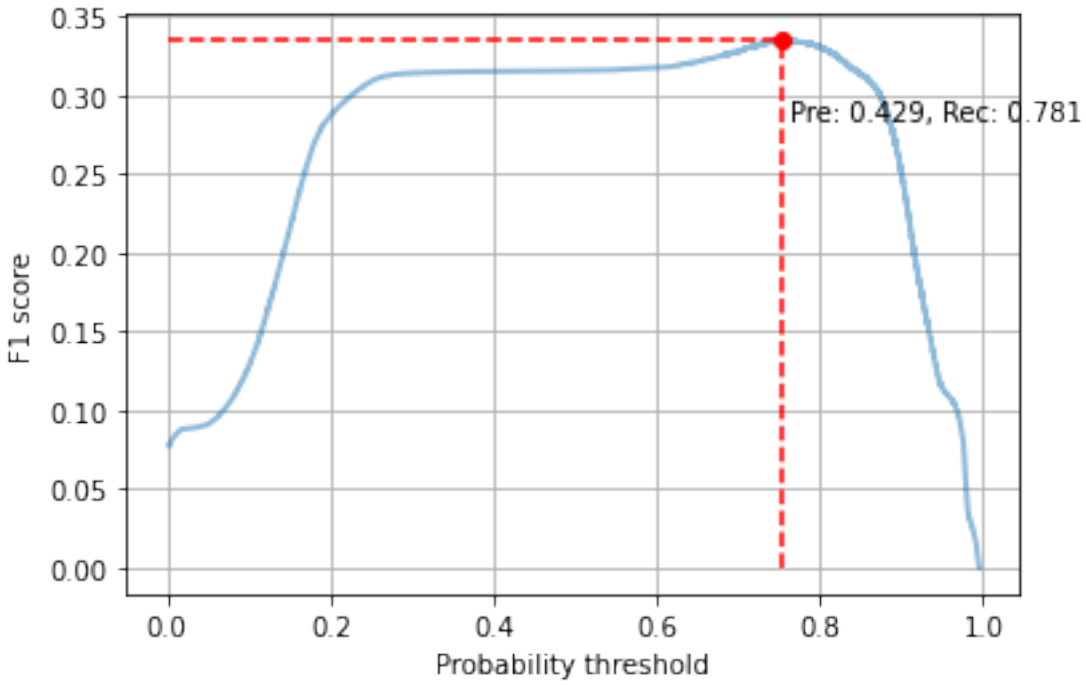
Weight	Feature
0.3059 ± 0.0010	vas_id
0.0949 ± 0.0004	buy_time
0.0019 ± 0.0003	247
0.0017 ± 0.0002	id
0.0008 ± 0.0002	229
0.0007 ± 0.0002	117
0.0007 ± 0.0001	164
0.0006 ± 0.0001	222
0.0006 ± 0.0000	207
0.0006 ± 0.0002	226
0.0006 ± 0.0001	241
0.0005 ± 0.000	138
0.0005 ± 0.000	155
0.0005 ± 0.0001	145
0.0005 ± 0.0001	244
0.0004 ± 0.000	036
0.0004 ± 0.0001	146
0.0004 ± 0.000	162
0.0004 ± 0.000	147
0.0004 ± 0.0001	238

При сочетании разных признаков из файла features.csv лучшим результатом был: 0.71367 при кросс-валидации на трех фолдах с группировкой по id, для уменьшения переобучения и ликов данных.

При использовании признаков из файла data_train.csv 0.71469.

Метрики различаются не сильно, но время выполнения кода и использование объема памяти сокращаются во много раз при использовании признаков только из data_train.csv.

Подбор порога вероятности



Из данного графика мы видим, что наилучший скор модель показывает при вероятности 0,79. Опытным путем мы подбираем порог вероятности 0,85.

После этого я подобрал оптимальные параметры с помощью optuna. Дефолтные параметры показывали лучший результат.

Итог

В итоге данная модель показала результ **0.76** при кросс-валидации на трех фолдах с группировкой по id. **Recall** данной модели составлял **0,9**, т. е. алгоритм с 90% вероятностью определяет тех абонентов которые нуждаются в данной услуге. **Precision** составила **0,35** , таким образом мы ошибочно предложили наши услуги 65% абонентам так, как одну услугу мы предлагаем только 1 раз, то получается, что излишние затраты мы понесли единожды. Зато те клиенты которые подключили нашу услугу будут оплачивать нам ее ежемесячно. В этом случае для нас важно максимизировать именно Recall.