Итоговая работа по курсу «Мегафон»

Модель предсказания вероятности подключения услуги

Факультет: Искусственный интелект
Группа: GU_AI_1445 (24.05.2021)
Студент: Гусев Александр

1. Исходные данные

Задача: построить алгоритм, который для каждой пары пользователь-услуга определит вероятность подключения услуги.

Исходные данные: В качестве исходных данных доступна информация об отклике абонентов на предложение подключения одной из услуг.

Датасеты:
Тренировочный датасет **data_train.csv** (id, vas_id, buy_time, target)
Датасет с набором признаков абонента **features.csv** (id, feature_list)
Тестовый датасет **data_test.csv** (id, vas_id, buy_time)

id	vas_id	buy_time	target
540968	8.000	1537131600	0.000
1454121	4.000	1531688400	0.000
2458816	1.000	1534107600	0.000
3535012	5.000	1535922000	0.000
1693214	1.000	1535922000	0.000

Расшифровка параметров:
id - идентификатор абонента
vas_id - подключаемая услуга
buy_time - время покупки, представлено в формате timestamp
target - целевая переменная, где 1 означает подключение услуги, 0 — абонент не подключил услугу соответственно

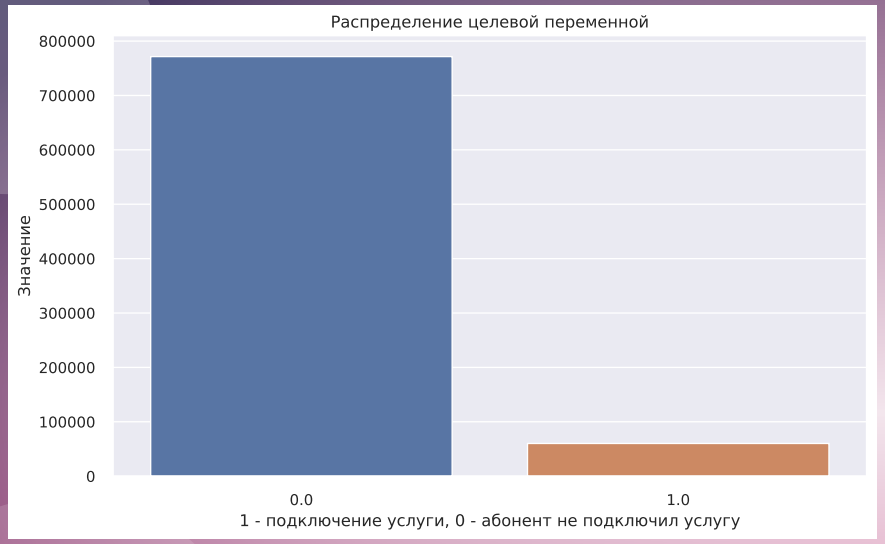
2. Анализ исходных данных

Целевая переменная имеет неравномерное распределение. Значение 0 составляет 92.76%, а значение 1 всего 7.24%.

Дисбаланс классов негативно сказывается на регрессионных моделях, но в меньшей степени влияет на точность моделей, основанных на деревьях решений

После добавления в data_train.csv признаков из features.csv имеем:

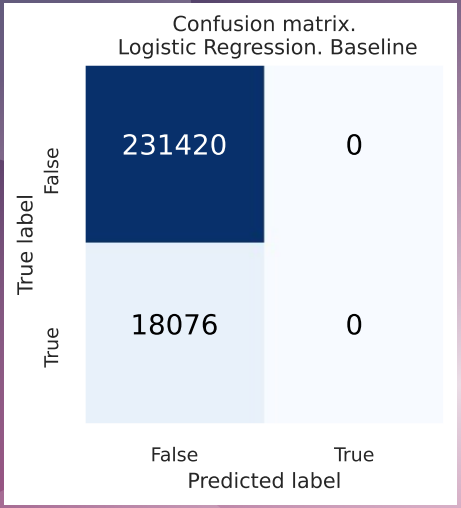
Всего признаков: 256
Временных признаков: 2
Константных признаков: 5
Категориальных признаков: 1
Вещественных признаков: 248

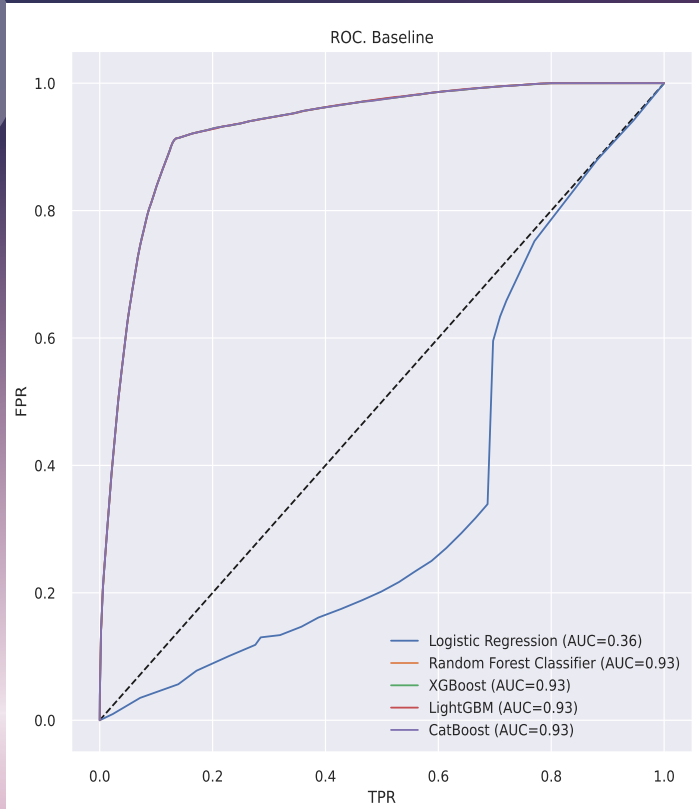


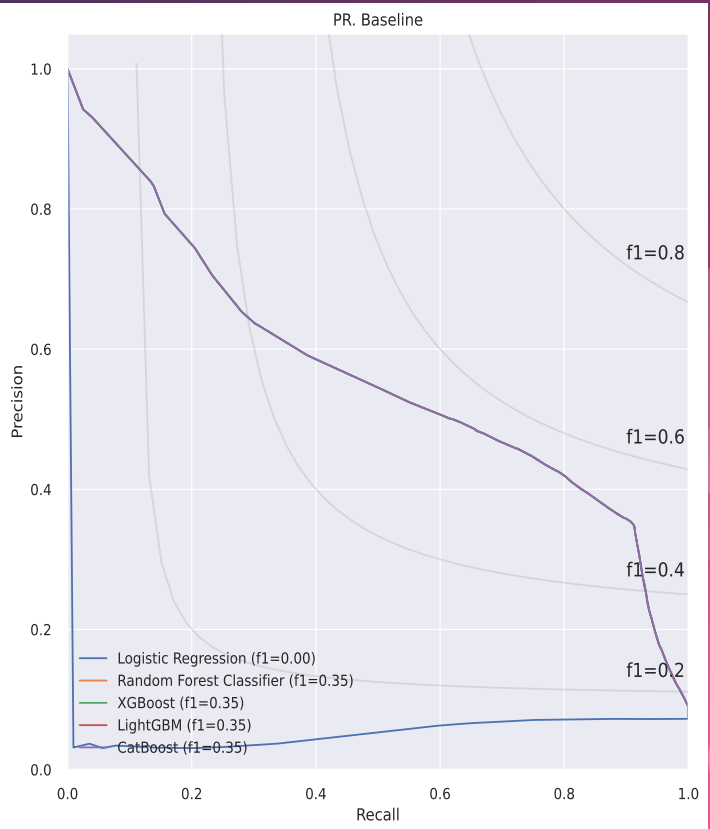
3. Baseline

Построение базовых моделей Logistic Regression, Random Forest Classifier, XGBClassifier, LGBMClassifier и CatBoostClassifier

На датасете **data_train.csv** (id, vas_id, buy_time, target) хуже всех сработала модель Logistic Regression (сказался дисбаланс классов). Остальные модели показали одинаковый результат.

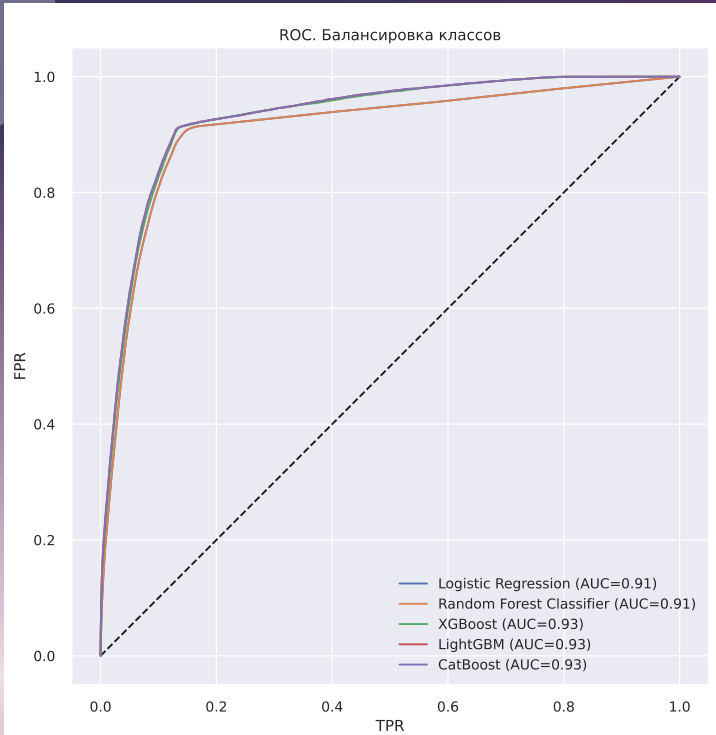


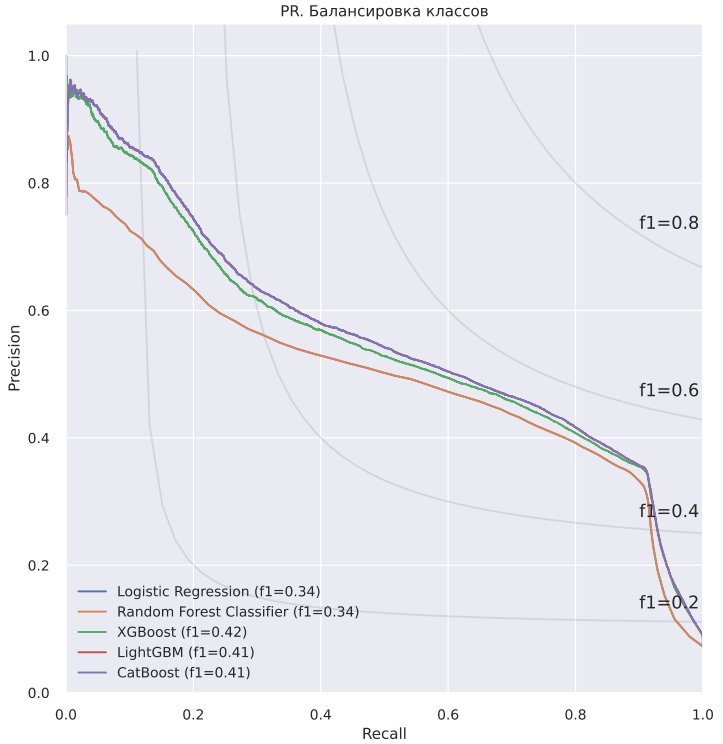




4. Добавление признаков и балансировка классов

Добавление признаков из features.csv и последующая балансировка классов с помощью SMOTE дало положительный результат. Все модели улучшили свои результаты.
Хуже всех себя показали Logistic Regression и Random Forest Classifier. Остальные модели показали одинаковый результат. XGBoost немного вырывается вперед, в зависимости от random_state.
При прочих равных, остановился на LigthGBM, т.к. она работает быстрее остальных

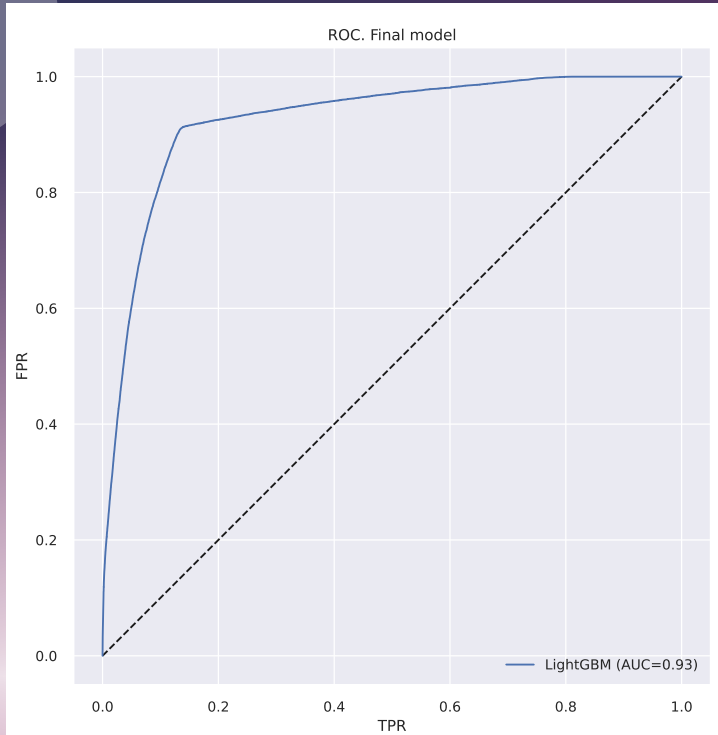


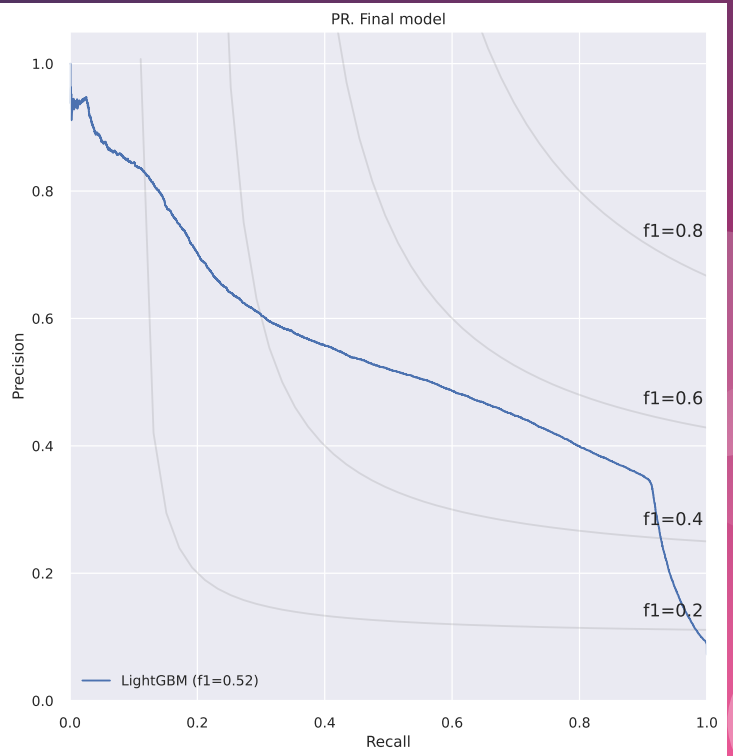


5. Тюнинг финальной модели (LGBMClassifier)

С помощью Randomized Search был осуществлён подбор оптимальных гиперпараметров для модели LGBMClassifier:

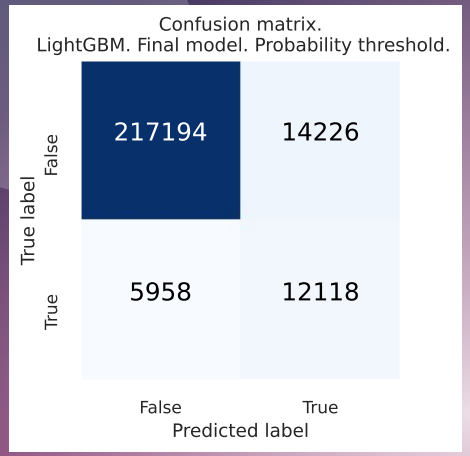
model_parameters =
{'SelectFromModel__threshold': 1e-05,
'subsample': 0.70,
'reg_lambda': 0.05,
'reg_alpha': 0.95,
'num_leaves': 25,
'n_estimators': 350,
'min_child_weight': 0.946,
'max_depth': 6,
'learning_rate': 0.21,
'class_weight': 'balanced'}

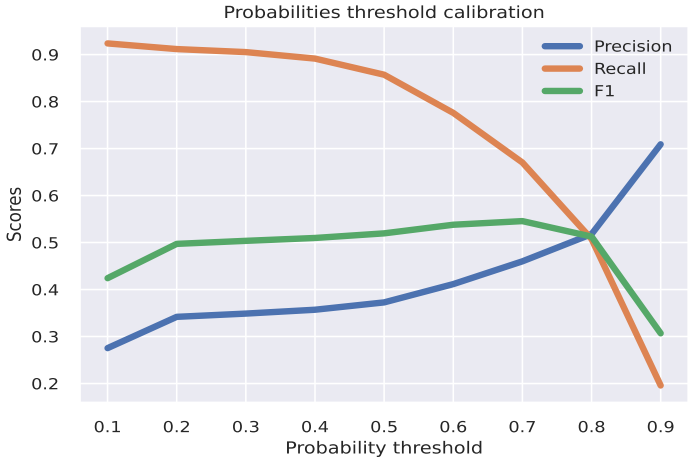


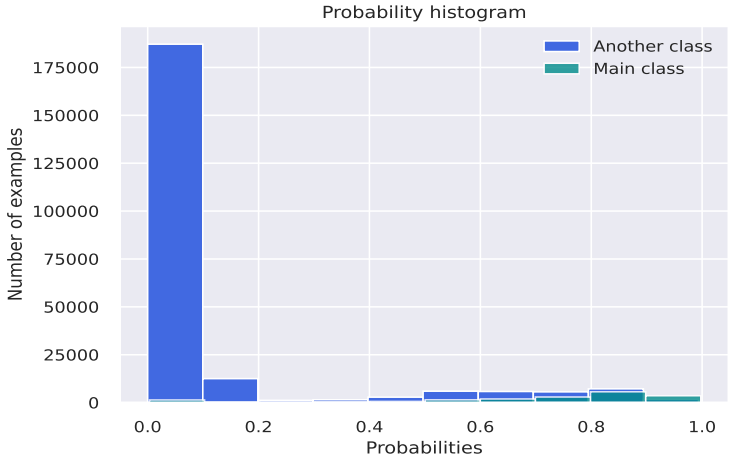


6. Выбор порога для определения класса

Итоговая модель имеет метрику f1 = 0,546 при пороге определения класса, равном 0,7







f1	precision	recall	probability
0.546	0.46	0.67	0.7
0.538	0.412	0.776	0.6
0.519	0.373	0.857	0.5
0.513	0.518	0.507	0.8
0.51	0.357	0.891	0.4
0.504	0.349	0.905	0.3
0.497	0.342	0.912	0.2
0.424	0.275	0.924	0.1
0.307	0.709	0.196	0.9