Презентация к курсовому проекту от «МегаФон»

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

28 декабря 2021

Данные. Метрика.

Данные

В качестве исходных данных доступна информация об отклике абонентов на предложение подключения одной из услуг. Каждому пользователю может быть сделано несколько предложений в разное время, каждое из которых он может или принять, или отклонить.
Отдельным набором данных будет являться нормализованный анонимизированный набор признаков, характеризующий профиль потребления абонента. Эти данные привязаны к определенному времени, поскольку профиль абонента может меняться с течением времени.
Данные train и test разбиты по периодам – на train доступно 6 месяцев, а на test отложен последующий месяц.

Итого, в качестве входных данных будут представлены:
data_train.csv: id, vas_id, buy_time, target
features.csv.zip: id, <feature_list>

И тестовый набор:
data_test.csv: id, vas_id, buy_time

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

Метрика

Скоринг будет осуществляться функцией **f1**, невзвешенным образом, как например делает функция sklearn.metrics.f1_score(…, average='macro').

Информация о модели, ее параметрах, особенностях и основных результатах

1. Объединил данные data_....csv с features.csv по id путем поиска ближайшего времени событий двух таблиц

```
df = pd.merge_asof(data, df_f, on="buy_time", by="id", direction='nearest')
df_test = pd.merge_asof(data_test, df_f, on="buy_time", by="id", direction='nearest')
```

2. Добавил 2 дополнительные фичи:
   diff – разница в днях между событиями двух таблиц data и features
   is_no_first – признак того, что это не первая запись о предложении услуги

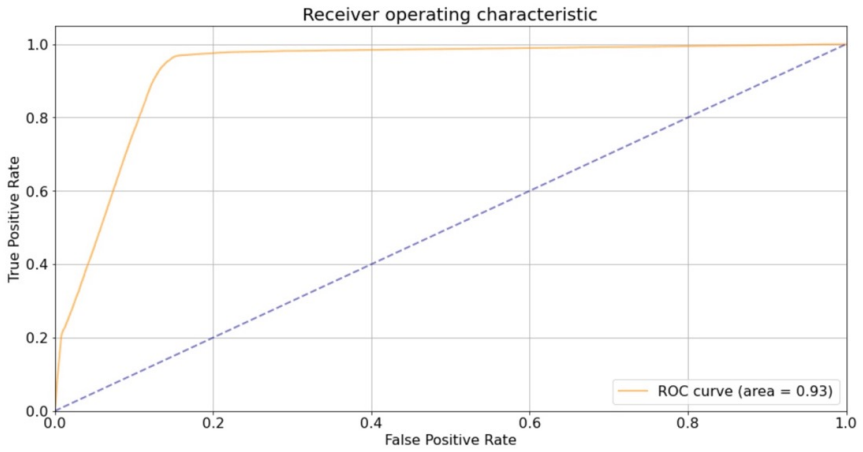
3. Разбил данные на тренировочные и валидационные
   train: июль – ноябрь 2018
   test: декабрь 2018

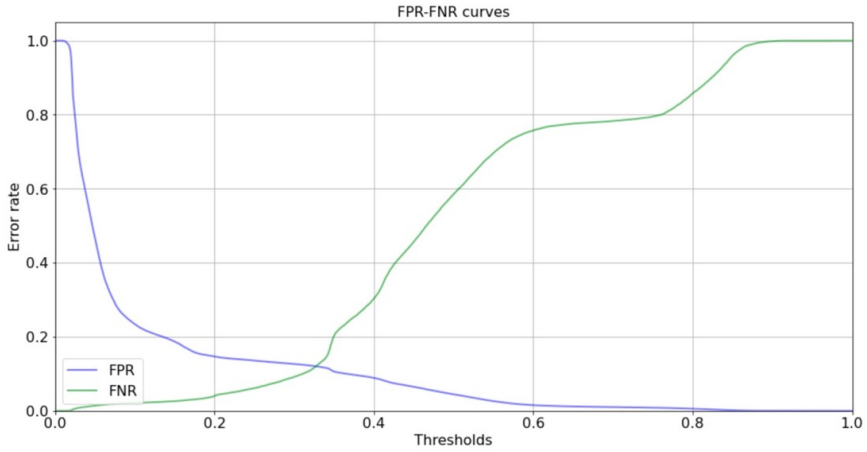
4. В качестве модели использовал CatBoostClassifier

5. Сократил количество признаков до топ-22
   для облегчения модели и тестового датасета с признаками

```
model = CatBoostClassifier(iterations=2500,
                           depth=10,
                           learning_rate=0.1,
                           loss_function='Logloss',
                           eval_metric='AUC',
                           early_stopping_rounds=50,
                           grow_policy='Depthwise',
                           random_state=42,
                           thread_count=4,
                           use_best_model=True,
                           verbose=True)
```

Информация о модели, ее параметрах, особенностях и основных результатах





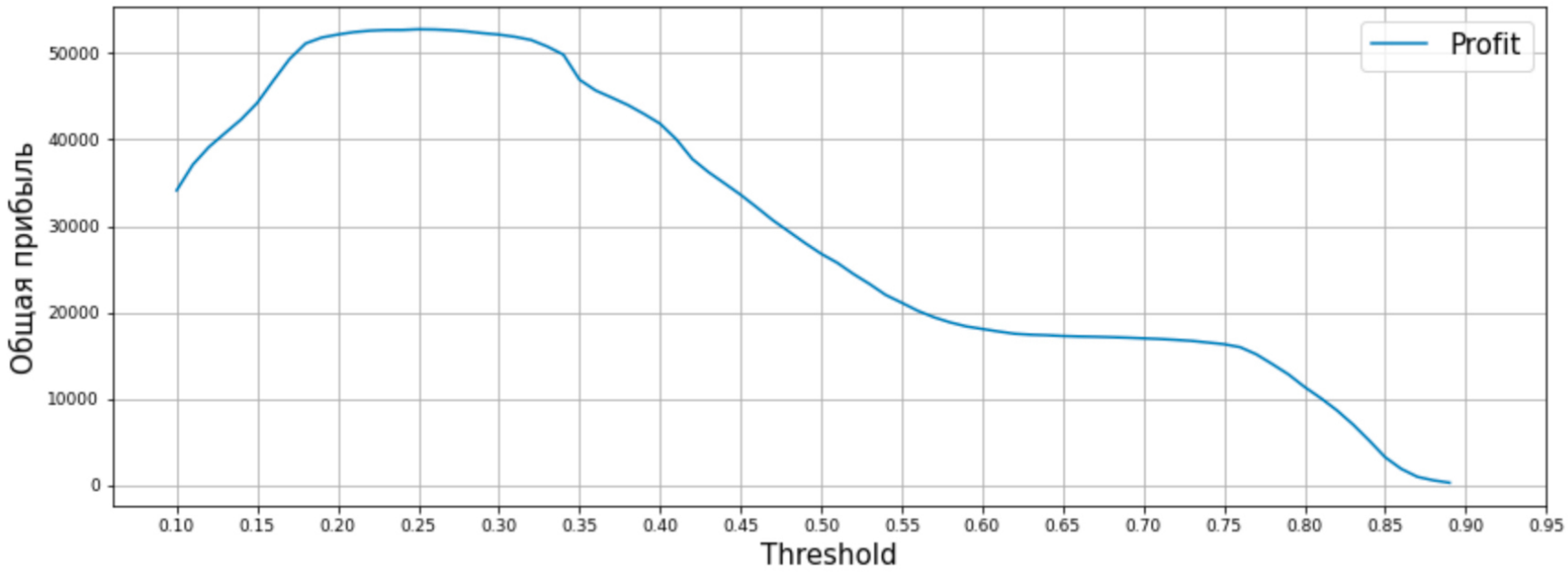
Результаты на валидации

```
f1_score test: 0.6966461052577112
roc_auc_score test: 0.9290454703125743
```

	feature_importance	feature_names
0	49.657020	vas_id
1	30.915755	is_no_first
255	10.806501	diff
224	2.015439	222
203	0.418318	201
225	0.318537	223
250	0.280479	248
252	0.257411	250

Рекомендации к выбору значения threshold, в зависимости от бизнес-показателей

По итогу теста на данных за декабрь 2018 (в среднем по всем услугам)



SCR=5
PCC=1

Используемая формула для расчета предполагаемой прибыли

$$Profit = TP \cdot SCR - (TP + FP) \cdot PCC$$

Profit – Общая прибыль со всех пользователей, подключивших услугу по предложению
TP – Количество успешных предложений (купили услугу)
FP – Количество не успешных предложений (не купили услугу)
SCR – Ожидаемый доход от услуги **в расчете на одного клиента**, сделавшего покупку
PCC – Затраты на **одно предложение** услуги клиенту (успешное или не успешное)

На примере значений **SCR=5 и PCC=1**
(расход на отправку одного предложения равен 1 у.е., а доход от одного пользователя при подключении услуги равен 5 у.е.)

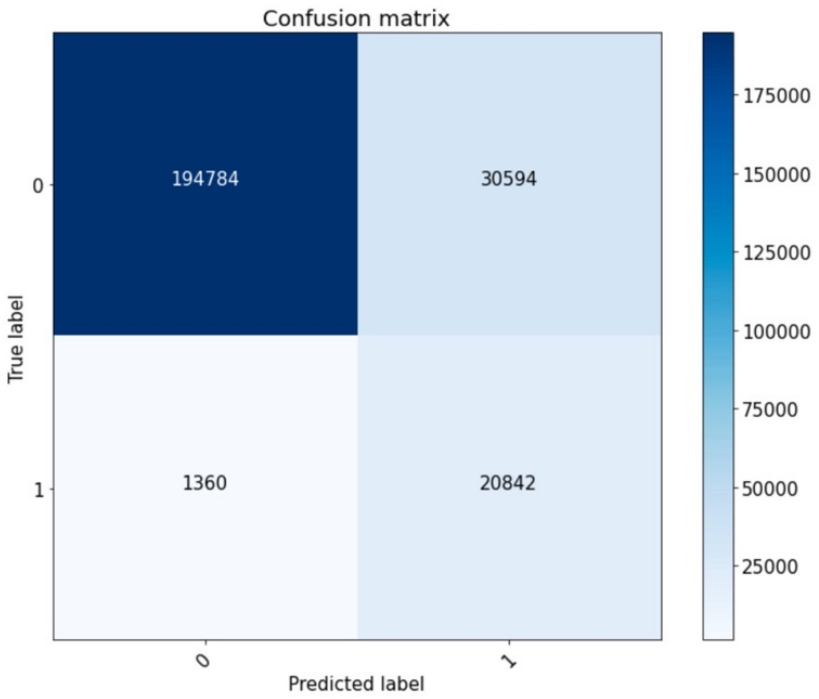
На графике видно, как зависит величина общей прибыли от значения threshold. Оптимальным значением threshold в данном примере является 0.25, что дает прибыль 52774 у.е. при полноте в 94% от всех возможных положительных решений.

Рекомендации к выбору значения threshold, в зависимости от бизнес-показателей

По итогу теста на данных за декабрь 2018 (в среднем по всем услугам)

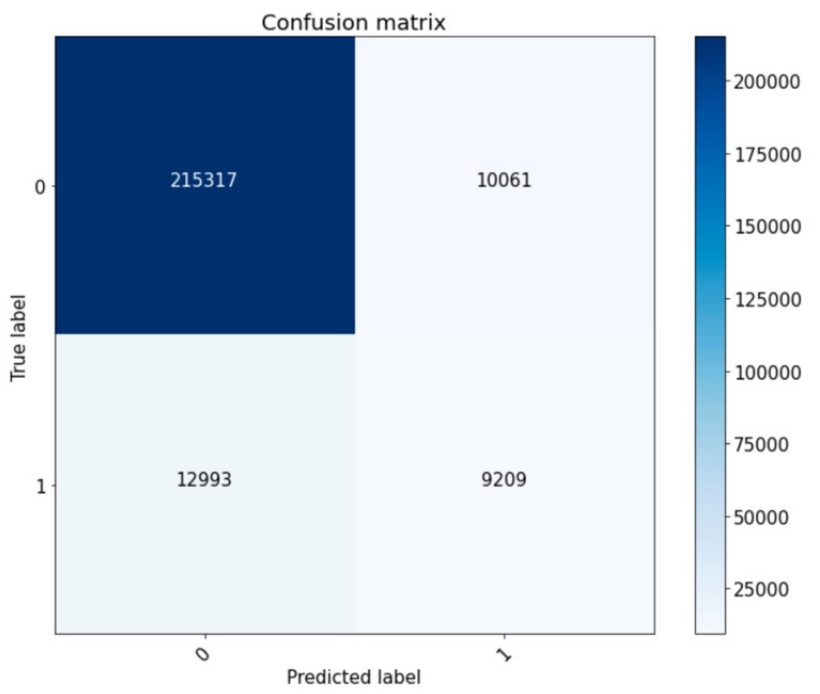
Confusion matrix, without normalization
[[194784 30594]
 [1360 20842]]

Treshold = 0.25



Confusion matrix, without normalization
[[215317 10061]
 [12993   9209]]

Treshold = 0.5



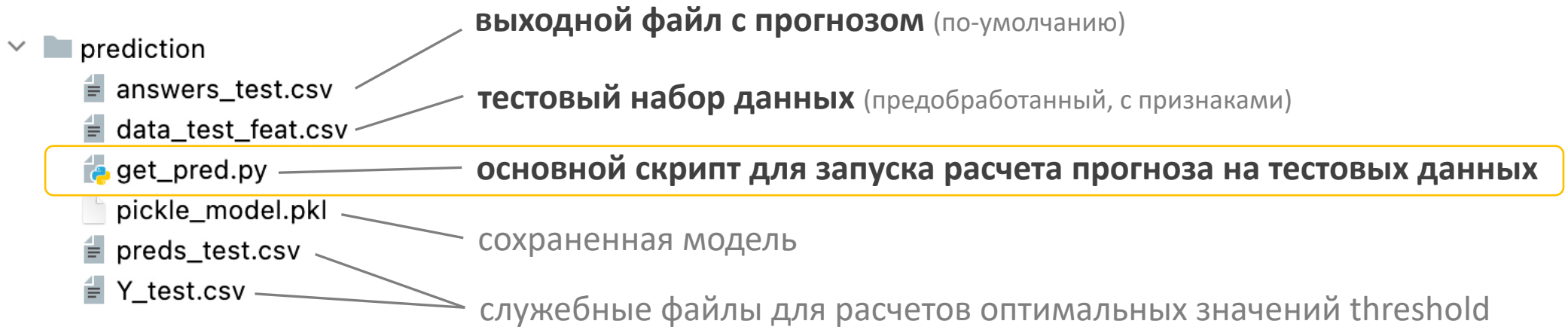
На примере значений **SCR=5 и PCC=1**

(расход на отправку одного предложения равен 1 у.е., доход от одного пользователя при подключении услуги равен 5 у.е.)

- ✓ **Treshold = 0.25**
- ✓ **Полнота верно предсказанных подключений 94%**
- ✓ **Profit = 52774 у.е.**

- ✓ Treshold = 0.5
- ✓ Полнота верно предсказанных подключений 41%
- ✓ Profit = 26775 у.е.

Описание файлов в папке проекта prediction. Применение модели.



*По-умолчанию при запуске скрипта без параметров включается **интерактивный режим**, в котором можно указать ожидаемые значения дохода от услуги одному клиенту и расхода на доставку предложения одному клиенту. На основании этих данных модель выберет **оптимальный threshold для максимизации прибыли** и сделает итоговый прогноз по всему датасету.*

Пример запуска интерактивного режима получения прогноза (меток класса) с автоматическим подбором значения threshold для максимизации прибыли

```
$ cd prediction
$ python get_pred.py
```

Описание файлов в папке проекта prediction

Возможные параметры для получения прогноза вручную

- --file входной файл для прогноза (должен включать дополнительные фичи)
- --to выходной файл для сохранения результатов прогноза (csv)
- --threshold ручное выставление значения порога при определении класса (при активном значении автоподбор с максимизацией прибыли отключается)
- --pproba при значении 1 модель возвращает не итоговые классы, а вероятности (если указано --pproba 1, то параметр threshold игнорируется)

```
$ cd prediction
$ python get_pred.py --file data_test_feat.csv --to answers_test.csv --threshold 0.48
```

Пример запроса, для получения вероятностей вместо итоговых классов

```
$ cd prediction
$ python get_pred.py --pproba 1
```

Возможные улучшения и доработки

- ✓ **В данных обнаружена аномальная активность по покупкам всех услуг, приходящаяся на 19 ноября 2018 года.** Стоит уточнить у «бизнеса», в чем может быть причина: возможно данные вносились не равномерно или было какое-то событие, акция... Поняв причины всплеска покупок, можно улучшить модель.

- ✓ **Все расчеты производились в среднем по всем услугам**. Для более точной оценки имеет смысл проработать прогноз каждой услуги отдельно. Возможно, разделить на 2 модели: 1) услуги 4, 6, 9 (как самые покупаемые); 2) все остальные.

- ✓ **Добавить pipeline** для автоматической предобработки исходных тестовых данных и добавления признаков из features.csv

Спасибо

zotov@adinweb.ru