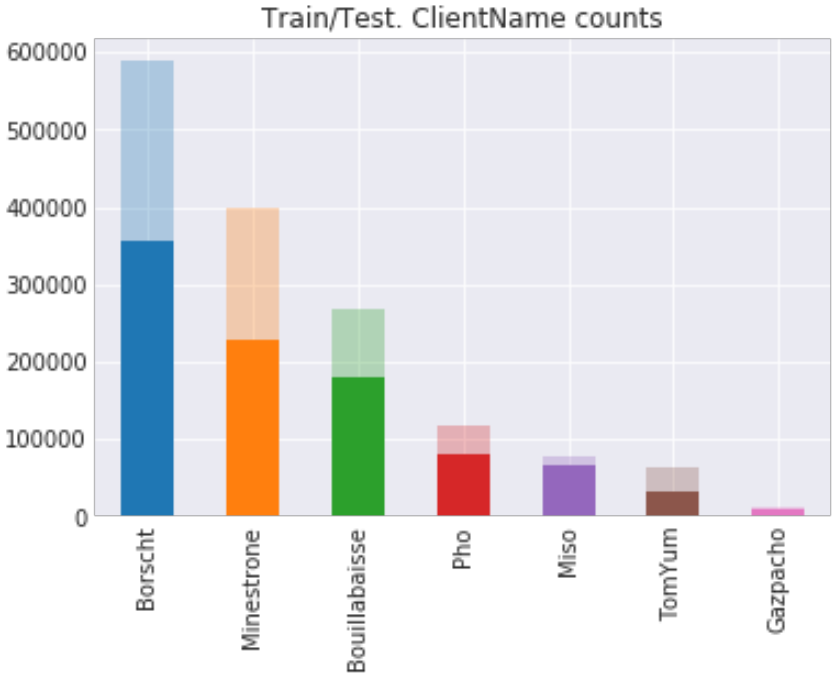
БАБУШКИН СУП

Онищенко Елена 1 место

Содержание

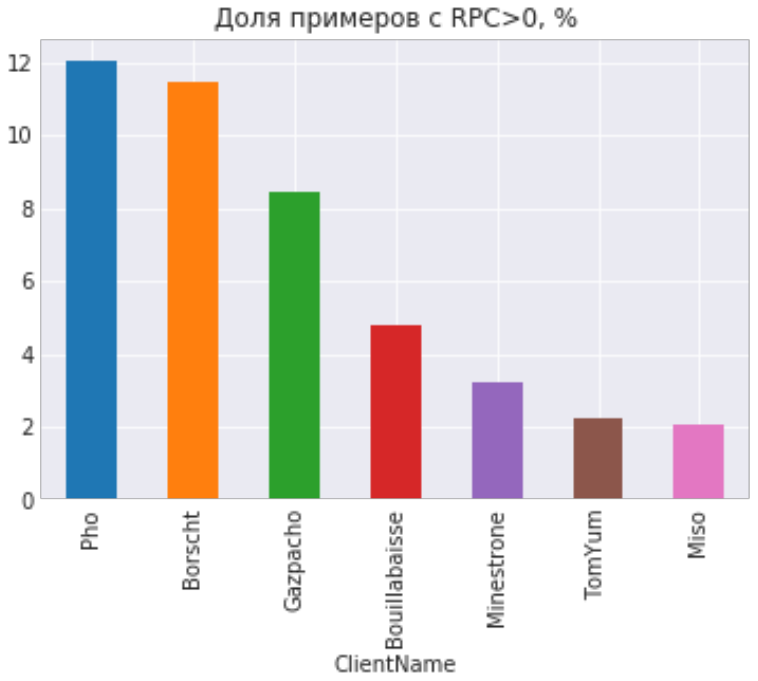
Данные



ClientName	RPC_ mean	Train counts %	Test counts %
Gazpacho	56,87	0,7	0,9
Pho	25,79	7,7	8,4
Train	7.15		
Borscht	6,73	38,5	37,5
Bouillabaisse	5,15	17,5	18,9
TomYum	4,88	4,2	3,3
Minestrone	3,84	26,1	23,9
Miso	0,53	5,1	6,8

train.RPC.mean < **test**.RPC.mean

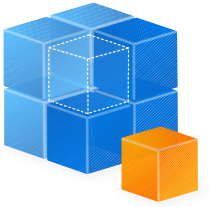
Undersampling



weight = 0

ClientName	count
Borscht	26 067
Bouillabaisse	27 693
Minestrone	11
Miso	4 738
	58 509
	3.8% train

Не учитываем при обучении (weight=0) примеры, для которых количество всех конверсий == 0, а ценность конверсий > 0 и RPC ==0



Признаки

1. На основе даты
2. Конкатенированые признаки
3. Для рекламных компаний
4. Для групп рекламных объявлений
5. И ещё… «странный» доход

Кодирование:

1. LabelEncoding — для базовых признаков
2. FrequencyEncoding— для базовых и конкатенированных признаков

Признаки на основе даты

1. День месяца
2. День недели*
3. Флаг праздника*
4. Флаг weekend
5. Количество дней до выходных (и праздников)
6. Другие — для рекламных компаний и групп объявлений

* Признаки «флаг праздника» и «день недели» использованы только в конкатенированных признаках

Признаки для рекламных компаний

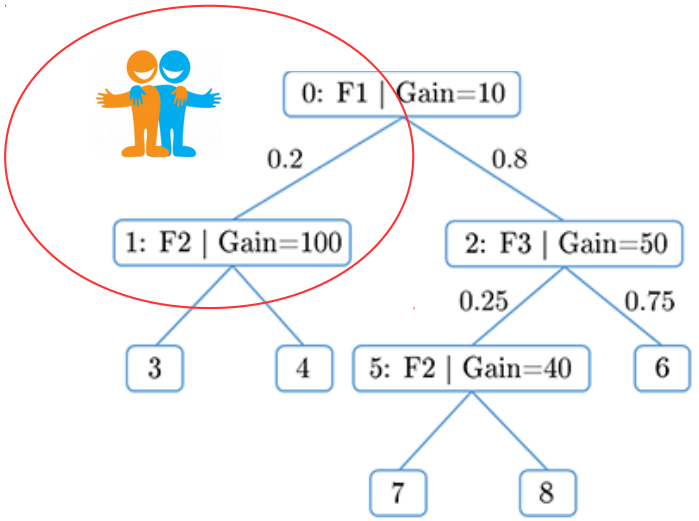
1. Доля примеров с RPC > 0
2. Доля примеров *AdNetworkType2* == SEARCH
3. Количество *KeywordId , AdGroupId*, уникальных слов (для всех - log)
4. Количество дней с даты первого клика
5. Количество дней с даты первого дохода
6. Доля рекламной компании, которая уже прошла

Признаки для групп рекламных объявлений

1. Количество *KeywordId* , уникальных слов (для всех - log)
2. Количество дней с даты первого клика
3. Флаг отсутствия группы объявлений в данных файла '*extra_kw_structure.csv*'

Остальные признаки, аналогичные признакам для рекламных компаний, были исключены по причине маленького количества примеров для некоторых групп и переобучения.

Производные признаки - конкатенированные



$\text{FScore}_{F2} = 2 \text{ (appears 2x)}$
$\text{Gain}_{F2} = \text{Gain}_1 + \text{Gain}_5 = 100 + 40 = 140$

Метод отбора — с помощью XGBfir*

Interaction	FScore	Gain
CampaignId | CampaignId	359	477 504 852
…		
CampaignId | Device	152	177 154 945
AdGroupId | Device	91	105 391 355
…		

*Xgbfi - XGBoost Feature Interactions & Importance project

https://github.com/Far0n/xgbfi
https://github.com/limexp/xgbfir

Производные признаки - конкатенированные

1. 'ClientName', 'CampaignId', 'AdGroupId' - с *'Device', днём недели, флагом праздника*
2. *'*Device*' - с днём недели, флагом праздника*
3. *'*ClientName*' - с 'RegionCriteriaId'*
4. *'*ClientName+dayofweek*' - с 'CountryCriteriaId'*

* Идея : генерация на основе базовых и производных признаков новых признаков - статистик по целевой переменной — не взлетело.

https://alexanderdyakonov.wordpress.com/2016/08/03/python-категориальные-признаки/

Лучшая соло-модель

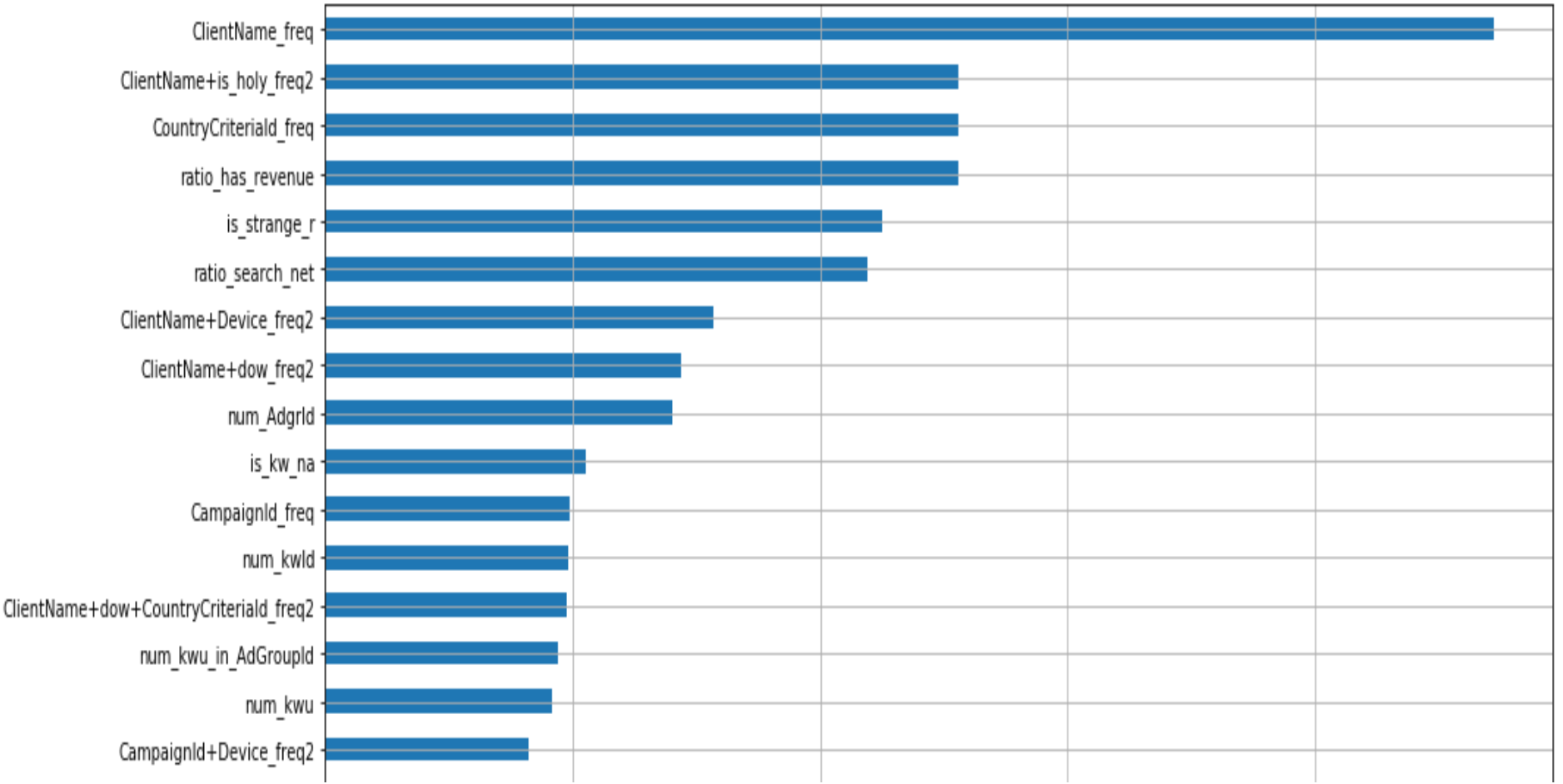
38 признаков, Xgboost - pub LB 66.090

parametrs*	
max_depth	7
min_child_weight	39
colsample_bytree	0.4
subsample	1
gamma	20
lambda	2.2933
eta	0.0025
num_round	1335

dtrain = xgb.Dmatrix(X_tr, y, **weight=X_tr.weight**)

*Тюнинг — ручками - скриптом по мотивам
https://www.analyticsvidhya.com/blog/2016/02/complete-guide-parameter-tuning-gradient-boosting-gbm-python/

Feature Imporance (Gain)

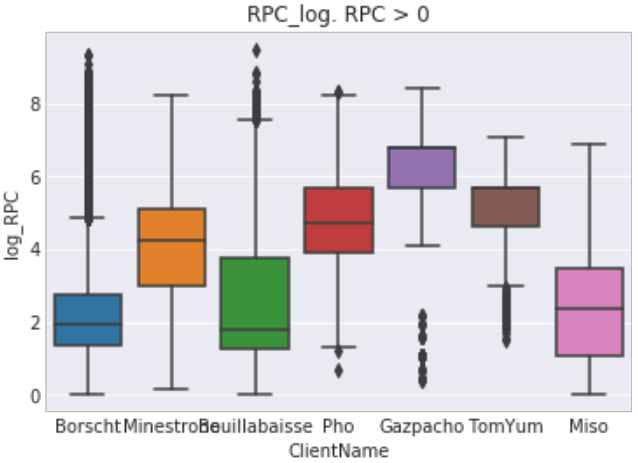


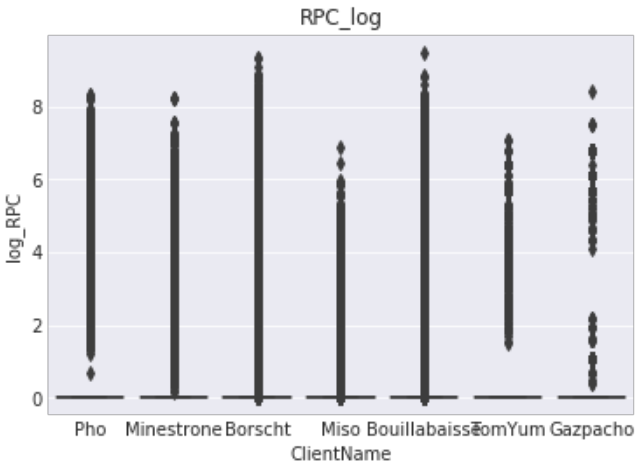
Ансамбль

	Public LB	Privat LB
Лучшая соло-модель	66.090	?
Смесь 4 лучших [соло] сабмитов	66.083	?
- " - + 6 xgb по мотивам лучшей модели	66.082	67.444

Для моделей использованы датасеты с **разными признаками** и/или **по-разному закодированными признаками** (34-38), а также **разные гиперпараметры** XGBoost (seed, num_round, [+ max_depth, min_chaild_weight, gamma, lambda — для 4х лучших сабмитов]).

Что ещё планировалось





1. Ещё признаки:
   - Бинаризация

     RPC, clicks, sum_num_conversion / clicks, sum_value_conversion / clicks

   + конкатенация с базовыми и производными признаками (отбор!)
   + FrequencyEncoding

2. Отдельные модели по крупным клиентам

3. Оптимизация весов и гиперпараметров моделей в ансамбле

Summary

Принципы построения решения:

1. Признаки, признаки, признаки
2. Не плодить лишних сущностей без необходимости
3. Тюнить xgboost

32 Gb RAM, Intel Core i7-7820X (8 ядер)

СПАСИБО ЗА ВНИМАНИЕ!

slack ODS @maatkara