Сервис прогнозирования целевых действий на сайте СберАвтоПодписка

Специализация: ML

Плотникова Александра • 16.09.2022

План презентации

Постановка задачи

Данные: подготовка к моделированию

Данные: закономерности

Выбор модели прогнозирования

Итоговая модель и ее показатели, подбор порога.

Структура итогового сервиса

Демонстрация работы сервиса и краткий инструктаж по развертыванию

Постановка задачи

Исходные данные

- ga_sessions.pkl - инфо об уникальных сессиях пользователей: 1860042 x 18;
- ga_hits-001.pkl - инфо о действиях пользователей: 15726470 x 11;
- Информация, какие действия считать целевыми;
- Сессию считаем конверсионной, если хотя бы 1 действие в рамках нее было целевым.

Задача:

- Создать сервис прогнозирования совершения целевого действия пользователем сайта;
- На вход сервиса поступают данные, аналогичные строке датасета ga_sessions.pkl;
- Ответ сервиса - прогноз совершения целевого действия данным пользователем.

Постановка задачи: требования к сервису

ROC AUC

не менее 0,65

Формат ответа

0/1

Скорость ответа сервиса

не более 3 сек.

Сервис

(минимум) .py-скрипт с инструкцией по локальному запуску или (максимум) localhost web app.

Данные: подготовка к моделированию

Необходимые данные собраны в 1 датафрейм:

- к рабочему файлу ga_sessions добавлен столбец ‘y’ (0/1): из файла ga-hits, сгруппированного по номеру сессии, добавлен признак конверсионности сессии согласно заданию
- итоговый рабочий файл имеет размерность 1732266 x 19: для части сессий информации в файле ga_hits не найдено - число строк уменьшилось. Дублей нет.

Баланс классов:

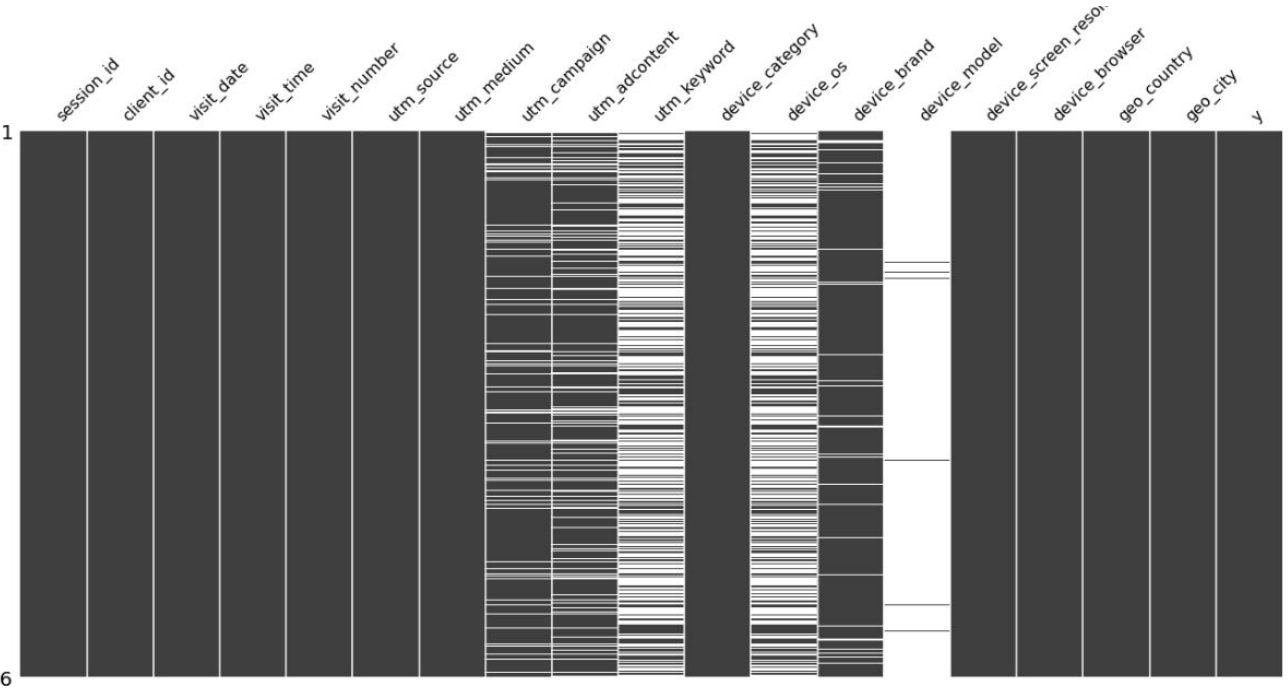
- Конверсия: 2,9%,
- Классы очень разбалансированы.

```
|: df.y.value_counts()

|: 0    1681952
   1      50314
   Name: y, dtype: int64
```

Данные: подготовка к моделированию

Полнота данных:



% пропущенных значений:

device_model	99.130503
utm_keyword	58.925823
device_os	58.533966
utm_adcontent	17.557177
utm_campaign	11.273500
device_brand	6.358030
utm_source	0.004387
geo_city	0.000000
geo_country	0.000000
device_browser	0.000000
device_screen_resolution	0.000000
session_id	0.000000
device_category	0.000000
client_id	0.000000
utm_medium	0.000000
visit_number	0.000000
visit_time	0.000000
visit_date	0.000000

Данные: подготовка к моделированию

Заполнение пропусков в данных:

- 'device_brand', 'utm_campaign', 'utm_adcontent', 'utm_source', 'utm_keyword': пропуски заменены на ‘other’
- 'device_os':
  - если ‘device_brand’ == ‘Apple - > ‘iOS’
  - если ‘device_browser’ == ‘Edge’ или ‘Internet Explorer’ или ‘IE’ - > ‘Windows’
  - в остальных случаях -> ‘Android’
- ‘device_model’ - > отбрасываем признак как непоказательный

Данные: подготовка к моделированию

Объединение мелких категорий в категориальных признаках:

- utm_campaign: сокращаем с 407 до 191 категории. Принцип - хотя бы ~100 элементов в категории. Остальные относим к ‘rare’.
- utm_keyword: 1193 -> 151 кат. Принцип: от 200 элементов в категории. Или ‘rare’.
- utm_source: 281 - > 79 категорий. Принцип: от 50 элементов в категории. Или ‘rare’.

Приведение типов:

- Категориальные признаки приводим к типу ‘category’
- ‘visit_date’ - приводим к типу ‘datetime’

Данные: подготовка к моделированию

Feature engineering:

#	New feature	Source features	Formula	Range
1	month	visit_date	df.visit_date.dt.month	1-12
2	day	visit_date	df.visit_date.dt.day	1-31
3	day_of_week	visit_date	df.visit_date.dt.dayofweek	0-6
4	hour	visit_time	[x.hour for x in df.visit_time]	0-23
5	device_browser_short	device_browser	df.device_browser.str.split(' ').str[0], type -> ‘category’	31 categories
6	screen_width	device_screen_resolution	df.device_screen_resolution.str.split('x'). str[0].astype(int)	0-5924
7	screen_height	device_screen_resolution	df.device_screen_resolution.str.split('x'). str[1].astype(int)	0-20000

Данные: подготовка к моделированию

Feature engineering:

#	New feature	Source features	Formula	Range
8	pixels_sum	screen_height, screen_width	df['screen_height'] * df['screen_width']; ограничение сверху: 150000	0-150000
9	pixels_sum_category	pixels_sum	0.0 if x.pixels_sum < 280800 (q25) else (1.0 if x.pixels_sum > 376980 (q75) else 0.5)	0 / 0.5 / 1
10	organic	utm_medium	1 if x.utm_medium in ['organic', 'referral', '(none)'] else 0	0 / 1
11	advertisment	utm_source	1 if x.utm_source in ['QxAxdyPLuQMEcrdZWdWb', 'MvfHsxITijuriZxsqZqt', 'ISrKoXQCxqqYvAZICvjs', 'IZEXUFLARCUMynmHNBGo', 'PIbkrSYoHuZBWfYjYnfw','gVRrcxiDQubJiljoTbGm'] else 0	0 / 1

Данные: подготовка к моделированию

Feature engineering:

#	New feature	Source features	Formula	Range
12	lat	geo_city + координаты городов, полученные с помощью пакета geopy	есть данные - берем координату населенного пункта, нет данных - берем координату страны	[-46, 79]
13	long			[-176, 177]
14	country_lat	geo_country + координаты стран, полученные с помощью пакета geopy	координата страны из файла с выкачанными данными с пом. geopy	[-41.5, 79]
15	country_long			[-176, 173]
16	Russia	geo_country	1 if x.geo_country == 'Russia' else 0	0 / 1

Данные: подготовка к моделированию

Оставляем фичи для моделирования:

visit_number, utm_source, utm_medium, utm_campaign, utm_adcontent, utm_keyword, device_category, device_os, device_brand, **device_browser_short, month, day, day_of_week, hour, screen_width, screen_height, pixels_sum, pixels_sum_category, organic, advertisment, lat, long, country_lat, country_long, Russia.**

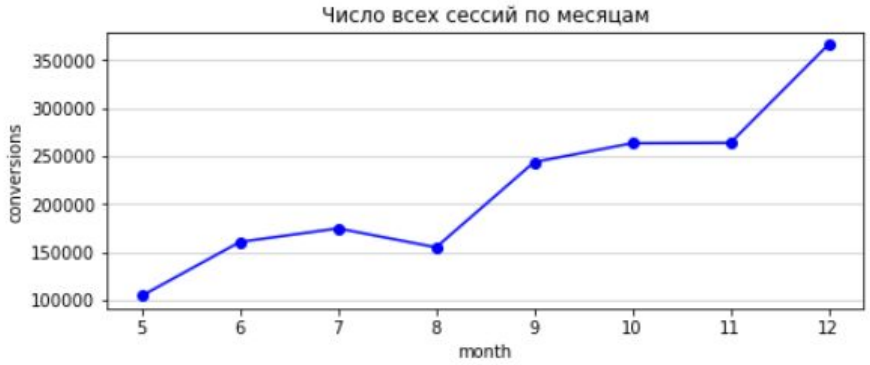
Остальные фичи - удаляем.

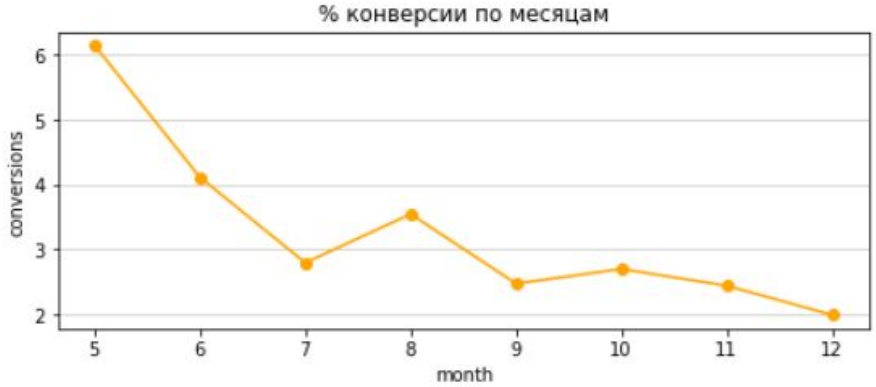
Применяем OneHotEncoder(pandas get_dummies), Standard Scaler к соотв.фичам.

Сохраняем 2 датасета для подбора будущей модели:

1. Датасет до кодирования и скалирования - для CatBoostClassifier.
2. Датасет с примененными pd.get_dummies и StandardScaler - для RandomForestClassifier, LogisticRegression, MLPClassifier.

Данные: закономерности. Сессии по месяцам.

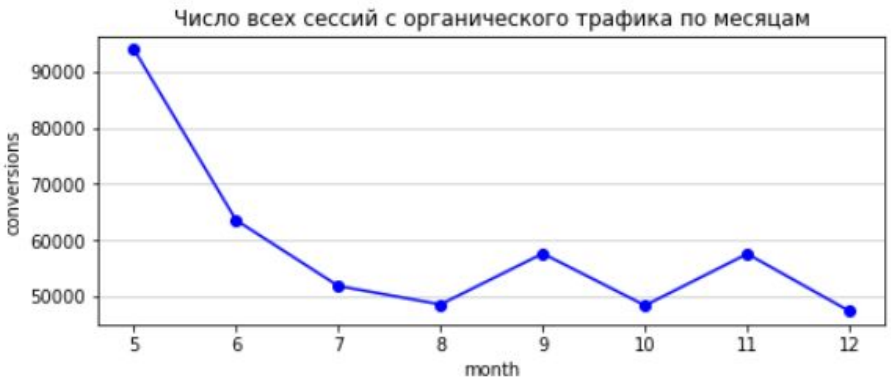


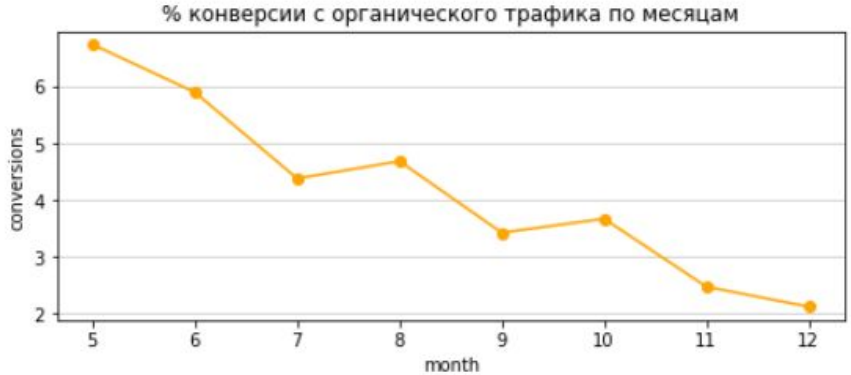


Общий трафик в течение года рос, % конверсии снижался. Число целевых сессий имеет большой разброс.

Июль, вероятно, - низкий сезон.

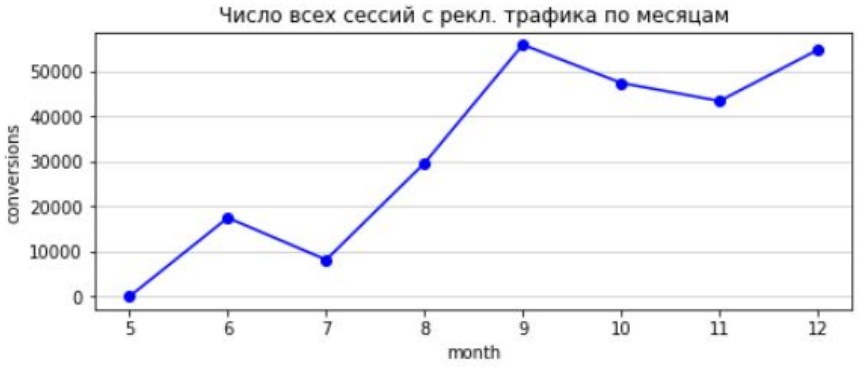
Данные: закономерности. Органический трафик.





Органический трафик весь период падал по всем параметрам. Вероятно, реклама перетянула на себя весь трафик.

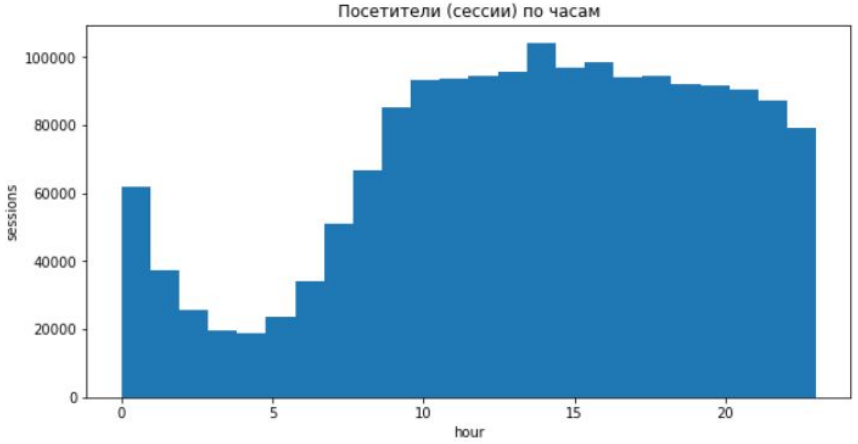
Данные: закономерности. Рекламный трафик.

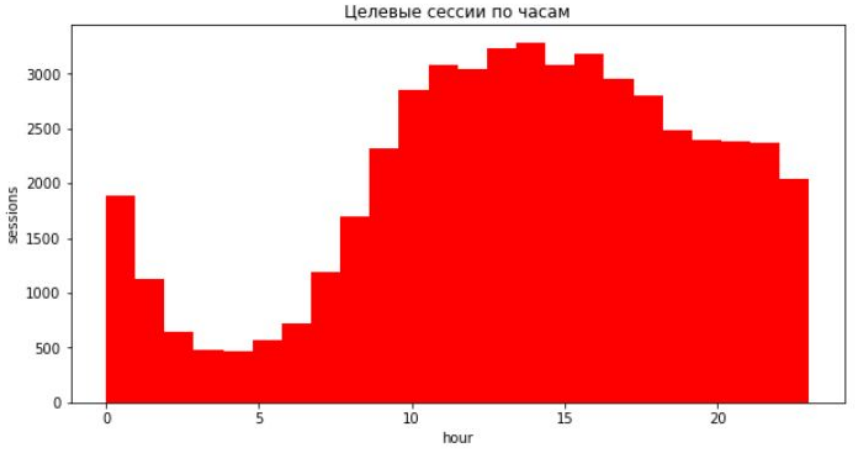


Рекламный трафик рос в абсолютных значениях весь период.

% конверсии с рекламы с августа почти стабилен.

Данные: закономерности. Сессии по часам.



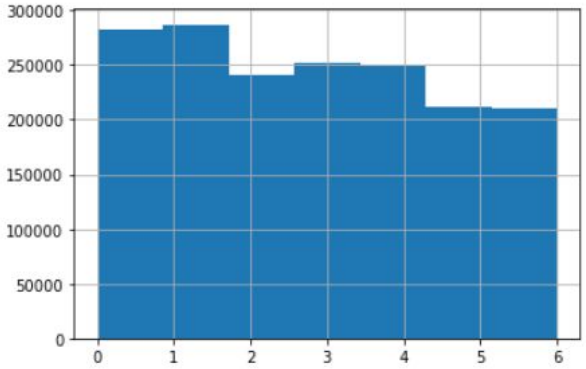


В течение суток интерес к сайту и продукту меняется волнообразно.

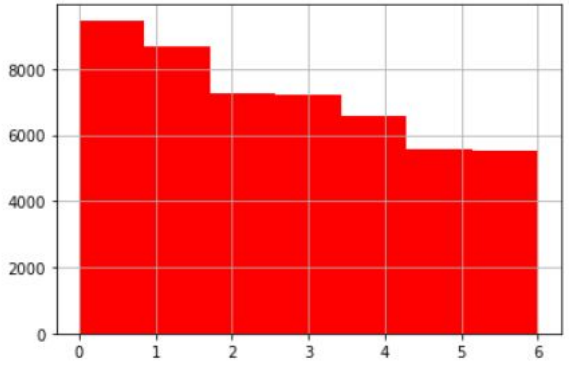
Конверсионность по часам имеет чуть более резкие контуры, чем общее число посещений сайта. (более широкий диапазон относительно пика - спуск до ⅙, против спуска до ⅕ у общего числа сессий).

Данные: закономерности. Дни недели.

Все сессии, пн-вс.



Конверсионные сессии, пн-вс.

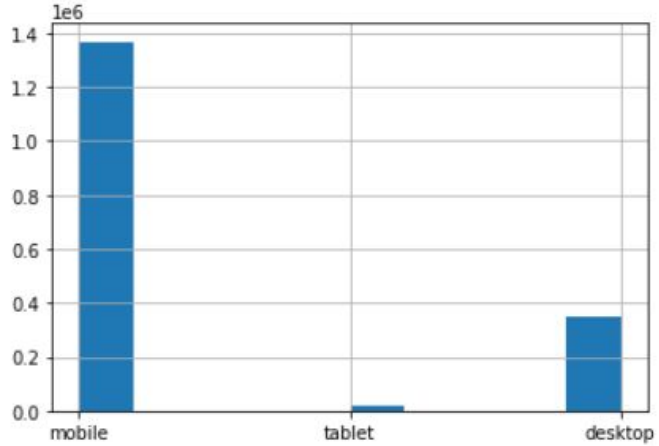


Понедельник и вторник: наиболее высокий интерес к сайту и продукту.

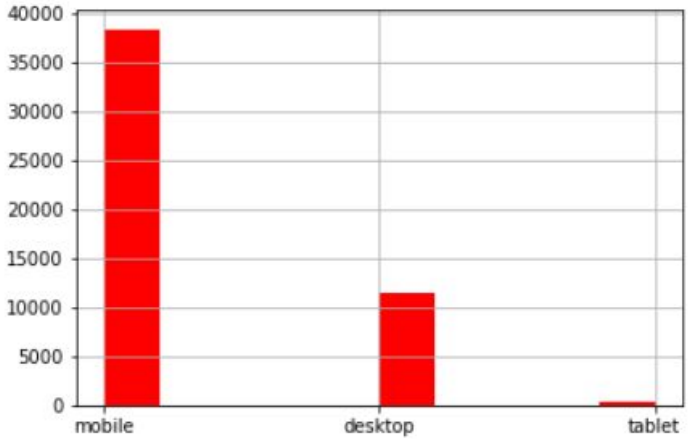
Выходные: меньше посетителей и меньше конверсионных сессий, чем в другие дни.

Данные: закономерности. Тип устройства.

Все сессии

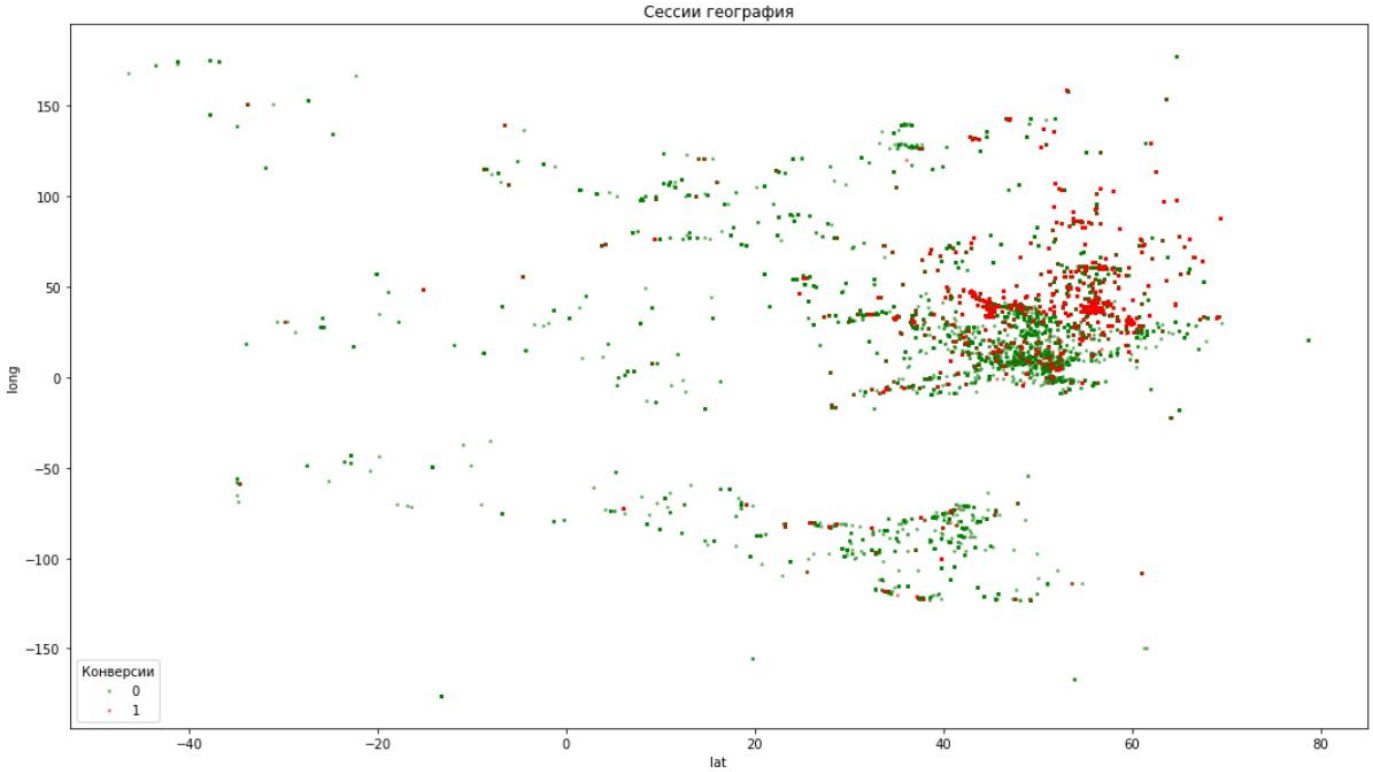


Конверсионные сессии



Основной траффик и конверсии идут с мобильных устройств.

Данные: закономерности. География.

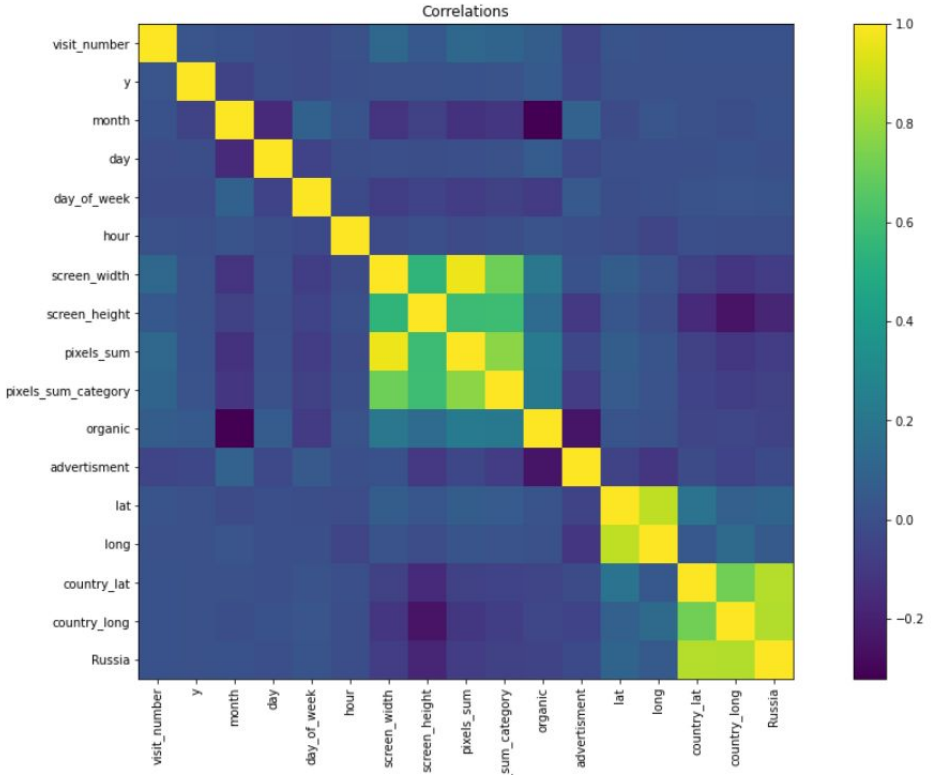


Конверсии концентрируются по городам в некоторых областях.

Нецелевые сессии более широко раскиданы по карте.

Закономерности явно есть.

Данные: закономерности. Корреляции.



Целевая переменная линейно не коррелирует ни с какими фичами.

Видим некоторую взаимосвязь месяца и параметра органического трафика.

Заметно коррелируют вручную созданные фичи, связанные с параметрами экрана.

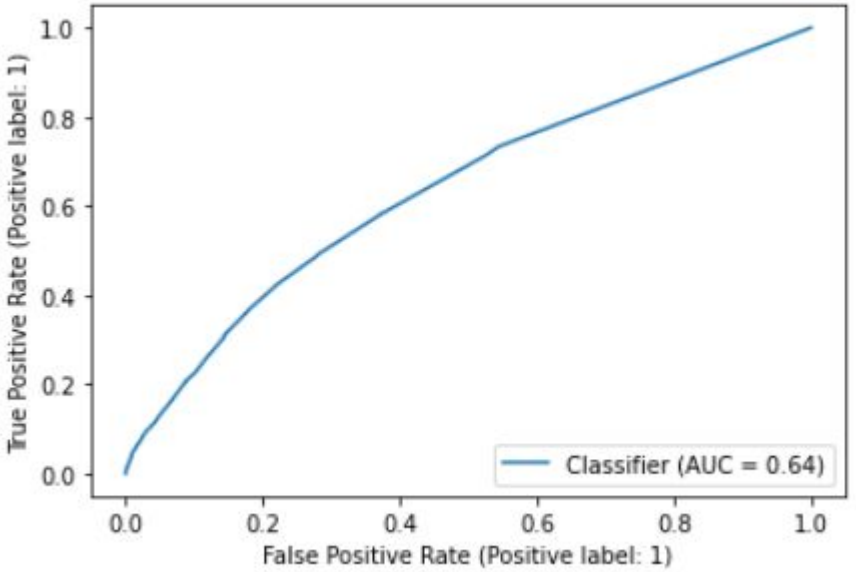
Видим корреляцию гео-параметров между собой.

Видим взаимосвязь органического и рекламного параметров.

Выбор модели прогнозирования

Random Forest Classifier

- 0.637 - roc_auc_score - ниже целевого.
- Время обучения 36 мин.
- Варианты усиления AUC ROC на 1-2%:
  - перебалансировка классов и разбивка датасета для обучения гибридной модели нескольких лесов.
  - тюнинг параметров.
- Риски дальнейшей работы по усилению модели: скорость работы сервиса, излишняя сложность пайплайна, эффект не достаточный (точность), потеря времени.



Выбор модели прогнозирования

Logistic Regression

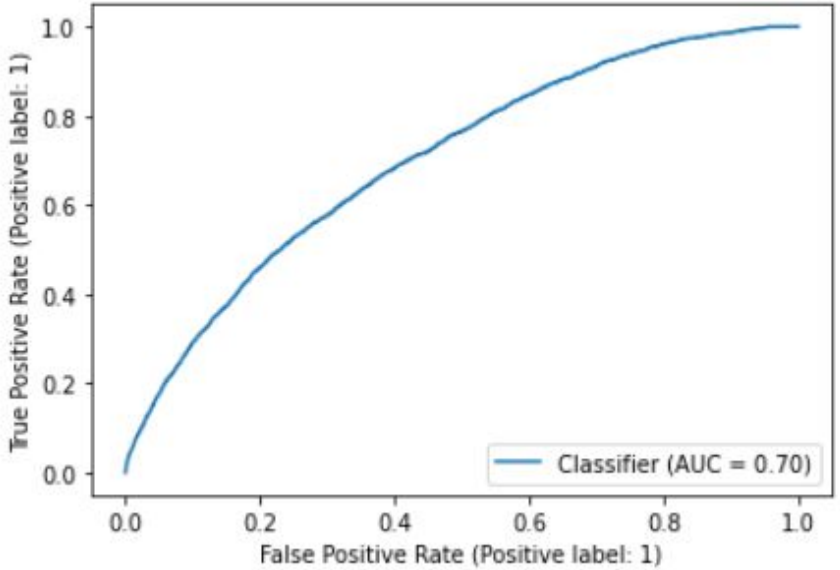
- Алгоритм и память не справились с объемами данных.
- В лучшем случае обучать гибрид на кусках данных и сокращать фичи.

- Предыдущие эксперименты на том же самом перебалансированном датасете с по-другому обработанными данными, разбитом на несколько частей, показали, что точность логистической регрессии близка к точности случайного леса на таких же кусках данных. Попробуем пока другие алгоритмы. Возможно, разбивка датасета не потребуется.

Выбор модели прогнозирования

MLP Classifier

- 0.702- roc_auc_score.
- Время обучения 56 мин. Процесс вручную остановлен.
- Результат лучше целевого на 5%, но слишком большое время обучения.
- Провести кросс-валидацию на таком тяжелом процессе обучения, возможно, не получится.
- Время обучения затруднит подбор параметров.

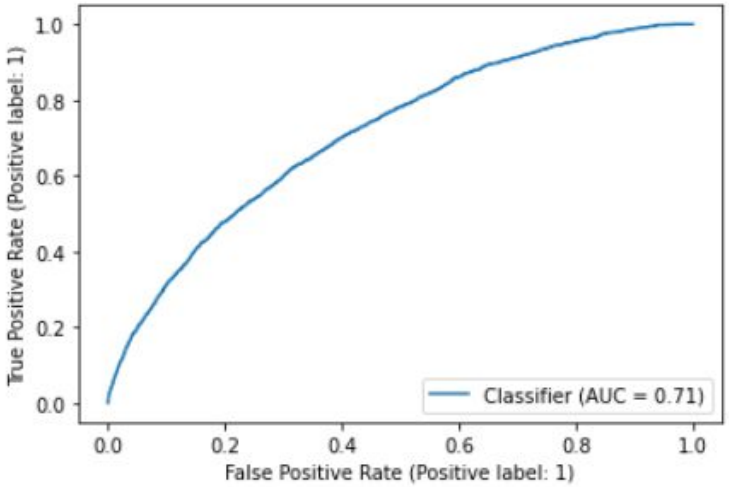


- Проверим еще вариант другой модели.

Выбор модели прогнозирования

CatBoost Classifier

- AUC ROC: 0.713 - 0.714 (несколько экспериментов). Лучший результат.
- Время обучения: 31-34 мин.
- Подбор learning rate: 0.236



Остановим выбор на Catboost Classifier:

- сокращенная обработка данных - снизит время работы сервиса.
- более высокая точность модели.
- приемлемое время обучения алгоритма: кросс-валидация и обучение пайплайна не должны вызвать проблем.

Параметры Catboost Classifier: важность фичей

	Feature Id	Importances
0	utm_source	11.680995
1	month	10.803101
2	visit_number	9.331320
3	utm_campaign	7.619588
4	organic	6.985890
5	screen_height	6.093336
6	utm_adcontent	5.557279
7	day	4.510646
8	utm_medium	4.389380
9	screen_width	4.315856
10	hour	3.901677
11	long	3.795505
12	lat	3.313127

	Feature Id	Importances
13	device_brand	2.805102
14	pixels_sum	2.693780
15	advertisment	2.479032
16	utm_keyword	2.133310
17	device_browser_short	1.903784
18	device_os	1.568455
19	day_of_week	1.496405
20	country_long	1.124886
21	device_category	0.759137
22	country_lat	0.554056
23	pixels_sum_category	0.104115
24	Russia	0.080238

Не вижу необходимости сокращать фичи для обучения итоговой модели, т.к.:

1. ни одна из них не получила коэффициент на уровне “шума” (0,0001 и ниже). Худший коэффициент важности - 0,08.
2. Здесь нет горячего кодирования и фичи не разбиты на сотни и тысячи отдельных, из которых часть можно было бы отбросить. Каждый столбец содержит массу информации для обучения.

Параметры CatboostClassifier: cv

Cross-Validation

- Параметры:
  'loss_function': 'Logloss',
  'custom_loss': 'AUC',
  'random_seed': 42,
  'learning_rate':0.236
- Число фолдов: 5

	iterations	test-Logloss-mean	test-Logloss-std	train-Logloss-mean	train-Logloss-std	test-AUC-mean	test-AUC-std
0	0	0.384561	0.000027	0.384554	0.000059	0.565144	0.003379
1	1	0.247000	0.000080	0.246983	0.000060	0.609562	0.003331
2	2	0.186192	0.000059	0.186169	0.000045	0.618264	0.002541
3	3	0.157895	0.000067	0.157868	0.000045	0.629955	0.003617
4	4	0.143847	0.000076	0.143811	0.000043	0.632893	0.003079
...	...	...	...	...	...	...	...
995	995	0.122370	0.000269	0.118114	0.000111	0.711486	0.003577
996	996	0.122369	0.000269	0.118110	0.000113	0.711498	0.003576
997	997	0.122370	0.000268	0.118107	0.000113	0.711480	0.003565
998	998	0.122370	0.000268	0.118105	0.000112	0.711494	0.003568
999	999	0.122370	0.000267	0.118103	0.000114	0.711498	0.003556

- Средний test AUC ROC на 999 итерации: **0,711498**
- test AUC std (999): **0,003556**

Переобучения нет.

Параметры Catboost Classifier: подбор порога

Поиск оптимального порога принятия решения на основе predict_proba.

Алгоритм:

- Разбиваем отрезок [0, 1] На 1000 делений - вариантов порога принятия решения, проходимся циклом по делениям - считаем прогнозы.
- Для каждого порога и прогноза проверяем ключевые метрики: f1, precision, recall, accuracy, и произведения метрик: recall*precision, f1*recall*precision.
- Сохраняем в отдельную переменную для каждой метрики/произведения метрик оптимальный порог
- для каждого из оптимальных порогов оцениваем confusion matrix и остальные показатели.
- Выбираем лучший вариант.

Параметры Catboost Classifier: подбор порога

Пороги (q), дающие лучшие показатели по тем или иным метрикам:

- best **accuracy score**: 0.97, **q = 0.349**

  [[97103 26]

  [2837 34]] модель почти везде предполагает 0.

  Полнота - очень плохая.

- best **f1**: 0.138, **q = 0.073**

  [[92733 4396]

  [2331 540]]

  Более смелая модель. Полнота - лучше, но все еще очень низкая

- best **recall**: 1.0, **q = 0.001**

  [[816 96313]

  [0 2871]] no comments

- best **precision**: 0.875, **q = 0.838**

  [[97128 1]

  [2864 7]] no comments

На следующем слайде - лучшие произведения метрик.

Параметры Catboost Classifier: подбор порога

Смотрим произведения метрик для подбора порога:

- best **recall*precise**: 0.0289, **q = 0.001**

  [[816 96313]

  [0 2871]] тоже плохой вариант. Прогноз почти всегда 1.

- best **f1*recall*precise**: 0.0038 **q = 0.045**

  [[82041 15088]

  [1673 1198]]

  Достаточно высокое число угаданных единиц.

  При этом не слишком зашкаливают ложноположительные.

Параметры Catboost Classifier: подбор порога

Выбираем порог между лучшим f1 и лучшим f1*recall*precise:

- best **f1**: 0.138,

  q = 0.073

  [[92733 4396]

  [2331 540]]

  точность: 540/4936 = 0.1094

  полнота: 540/2871 = 0.188

  accuracy: 93273/100000 = 0.93

- best **f1*recall*precise**: 0.0038,

  q = 0.045

  [[82041 15088]

  [1673 1198]]

  точность: 1198/16286 = 0.736

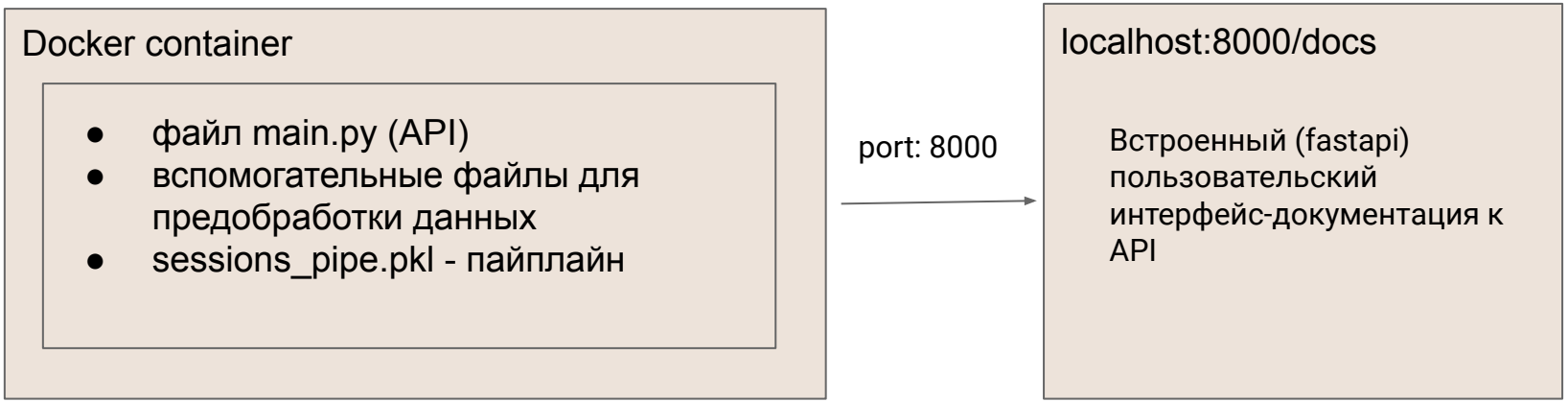
  полнота: 1198/2871 = 0.417

  accuracy: 83239/100000 = 0.83

Выбор: q= 0,045

В сервисе будет использоваться данный порог.

Структура сервиса прогнозирования



Get:
- root
- status
- version
- feature_importances

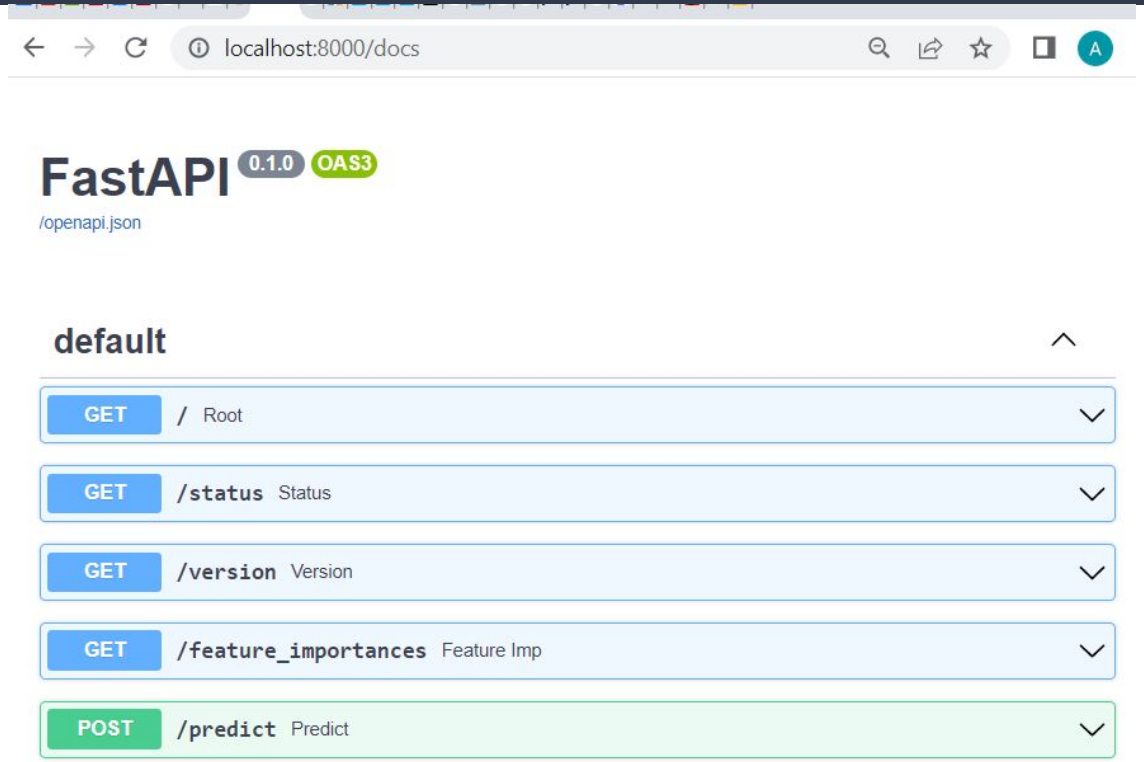
Post:
- predict

Инфо для генерации образа - в Dockerfile;
Инструкция по развертыванию и использованию сервиса: readme.txt.

Обработка запроса Post: predict

Время отклика: 0.056 сек. - 0.28 сек.

Демонстрация работы сервиса



localhost:8000/docs

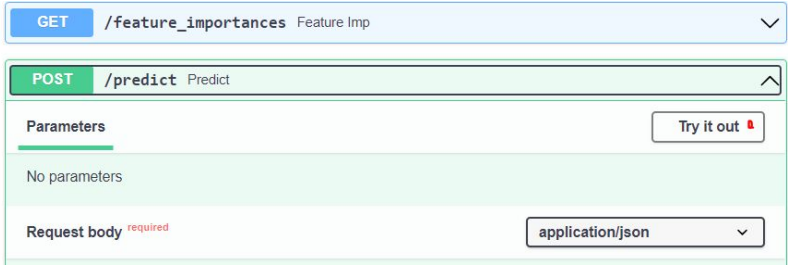
Встроенный интерфейс FastAPI
- Документация
- Рабочий функционал

Каждый вид запросов можно раскрыть и протестировать - увидеть ответ сервиса.

Демонстрация работы сервиса (предикт)

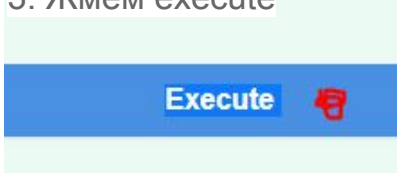
Post: predict. Время отклика: 0.056 сек. - 0.28 сек.

1. Раскрываем нужный запрос. Try it out.



2. Подаем данные о сессии на вход (json)

3. Жмем execute



4. Получаем ответ.

Спасибо за внимание!

Вопросы?