Анализ сайта - “СберАвтоподписка”

Леонард Стучилин

школа Skillbox

Цель проекта:

Научиться предсказывать совершение целевого действия (ориентировочное значение ROC-AUC ~ 0.65) — факт совершения пользователем целевого действия.

Итог проекта:

Упакованная в сервис модель, берущая на вход все атрибуты, типа utm_*, device_*, geo_*, и предсказывающая на выходе 0/1 (1 — если пользователь совершит любое целевое действие).

Методология исследования данных:

CRISP DM

Предоставленные данные:

1. датасет GA Sessions

2. датасет GA Hits

Работа выполнена :

Python,
Jupyter_notebook,
PyCharm

Приступая к проекту

Главной целью для меня в данном проекте, было научиться самостоятельной реализации подобного задания.

Дополнительно я получил прекрасную возможность разобраться во всех этапах подготовки и реализации проекта. От понимания предоставленных данных до реализации готового сервиса.

При реализации проекта мною были применены полученные на курсе знания и навыки.

Так-же была использована техническая документация библиотек и модулей используемых в проекте.

Что позволило улучшить навыки в поиска, освоения и закрепления новых знаний.

Data Understanding

Исследование датасета GA Sessions

Исследование датасета GA Hits

Извлечение целевой переменной - event_value из датасета GA Hits

Добавление целевой переменной к датасету GA Session

Data Preparation

Удаление дубликатов

Удаление неинформативных признаков, с пропусками в значениях более 45%

Заполнение пропусков в категориальных переменных введением дополнительного значения, ввиду предположения, что пропуски в данных признаках обусловлены отсутствием события

Data Cleaning

очистка данных

Удаление выбросов в числовых переменных, заменой значений выбросов, значением следующим за граничным значением квантиля, в данном случае==2

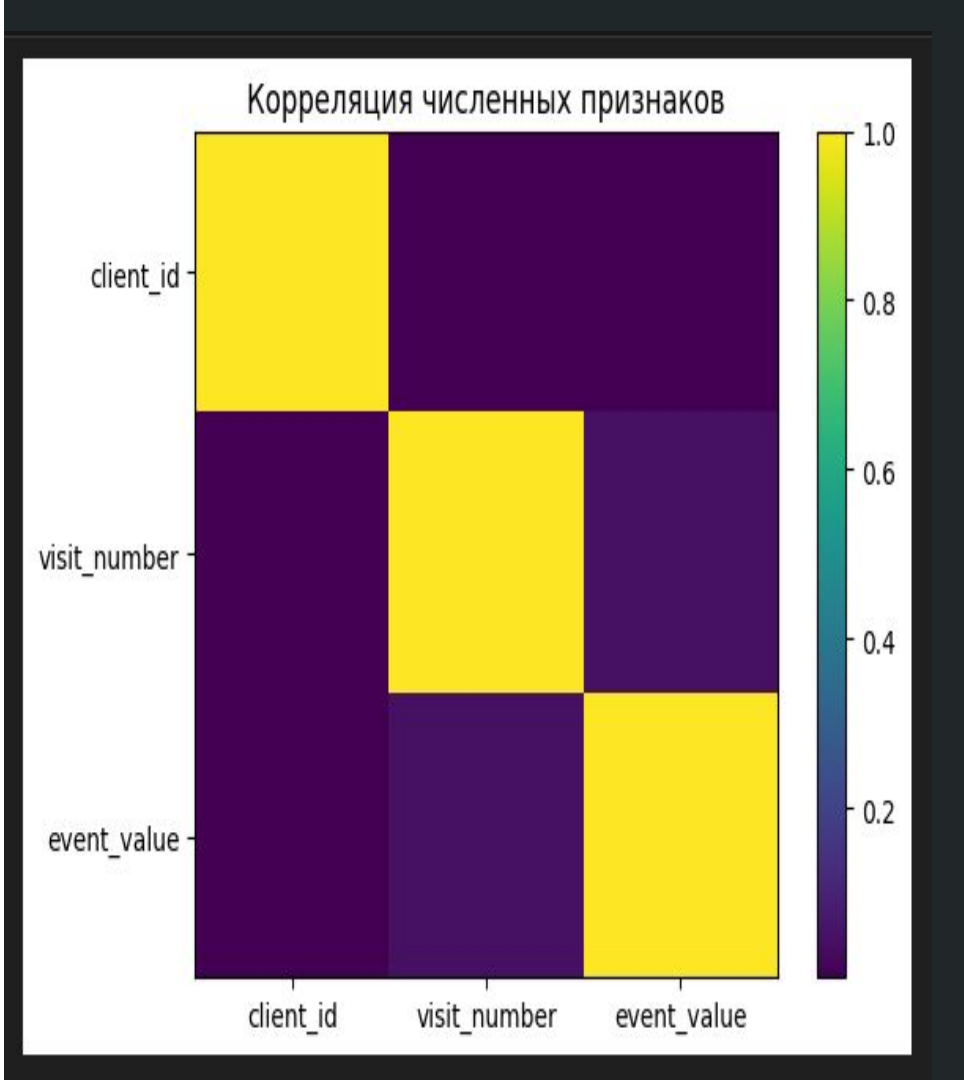
Преобразование значений в категориальных признаках. Взяты первые 2-5 букв значения в признаках: utm_source, utm_campaign, utm_adcontent, с сохранением количества уникальных значений

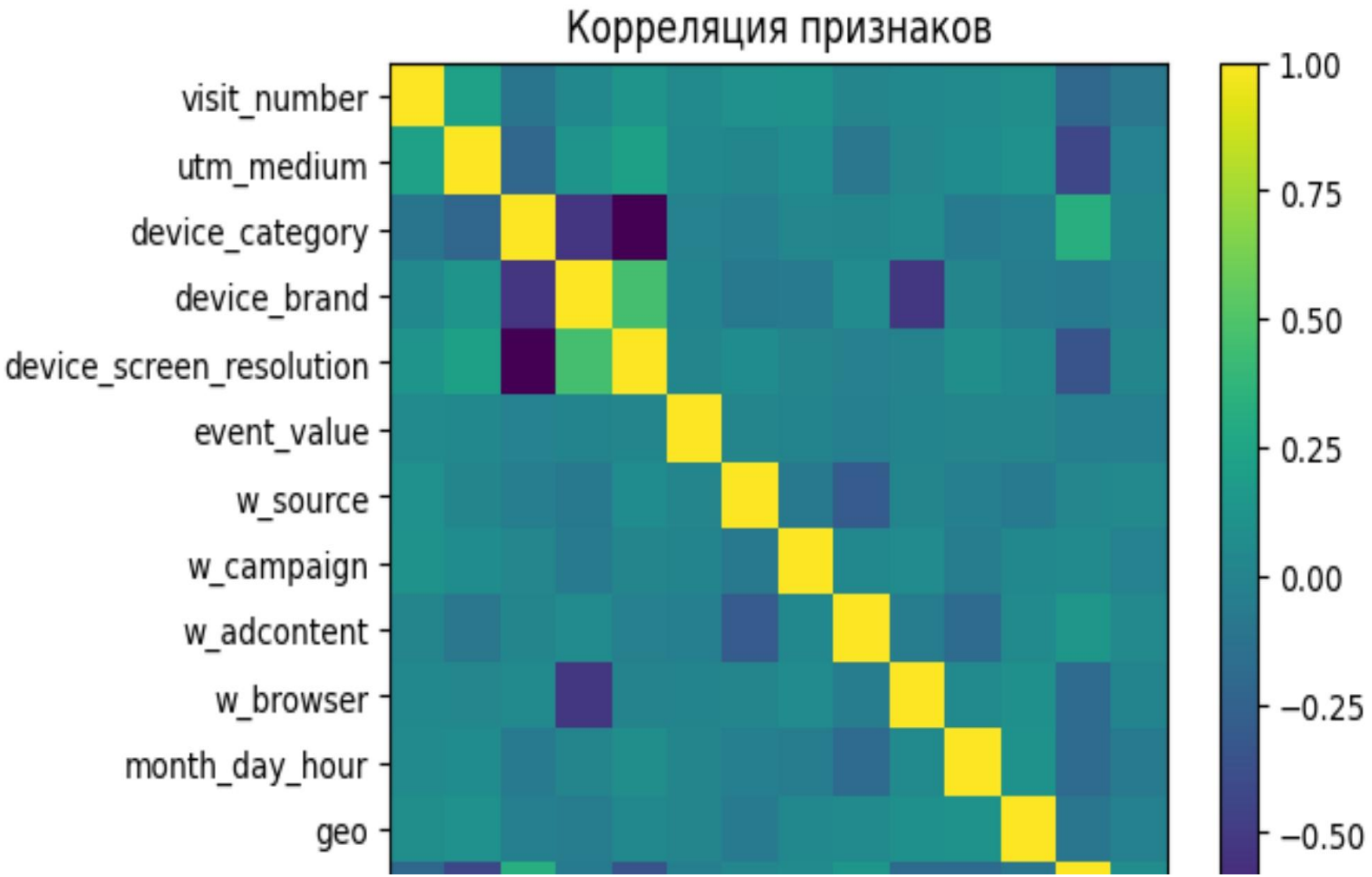
Объединение колонок содержащих дату и время, приведение к типу Datetime

Data Visualization

просмотр распределений признаков и их влияния на целевую переменную

Создание дополнительного датафрейма с преобразованием данных по типу '.astype('category').cat.codes', для просмотра корреляций признаков (включая категориальные строчные значения) к целевой переменной




Корреляция признаков
visit_number
utm_medium
device_category
device_brand
device_screen_resolution
event_value
w_source
w_campaign
w_adcontent
w_browser
month_day_hour
geo
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50

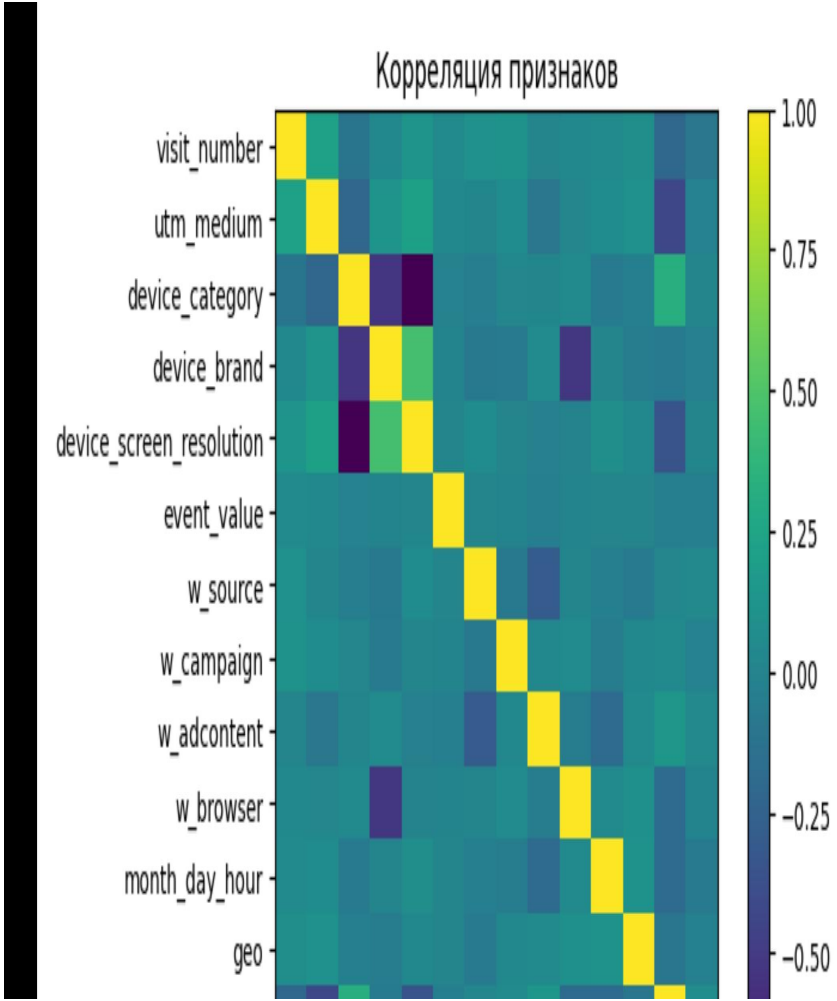
Что удалось выяснить

Исходя из исследований, можно предполагать следующее:

Данные слабокоррелированы с целевой переменной.

Присутствует дисбаланс классов.

Присутствуют шумы которые ухудшат качество модели.



Feature Engineering

были созданы следующие фичи:

date_time - дата и время события

month_day_your - месяц, день недели и час события

geo - численная переменная указывающая на локацию (от перевода к координатам отказался ввиду значительного времени обработки)

utm_organic - переменная указывающая на тип трафика

utm_path - составная переменная состоящая из нескольких признаков типа utm

Балансировка классов

Проводились эксперименты по балансировке классов методами Взвешивания классов и Undersampling.

Впоследствии от идеи балансировки классов отказался.

Поскольку результаты обучения моделей на подготовленных данных были стабильны и TruePositiveRate была достаточного уровня.

Что позволило сохранить естественные данные и не допустить их искусственности.

Modeling

Для моделирования использовались следующие модели, подходящие для задач бинарной классификации:

LogisticRegression()

RandomForestClassifier()

KNeighborsClassifier()

Pipeline

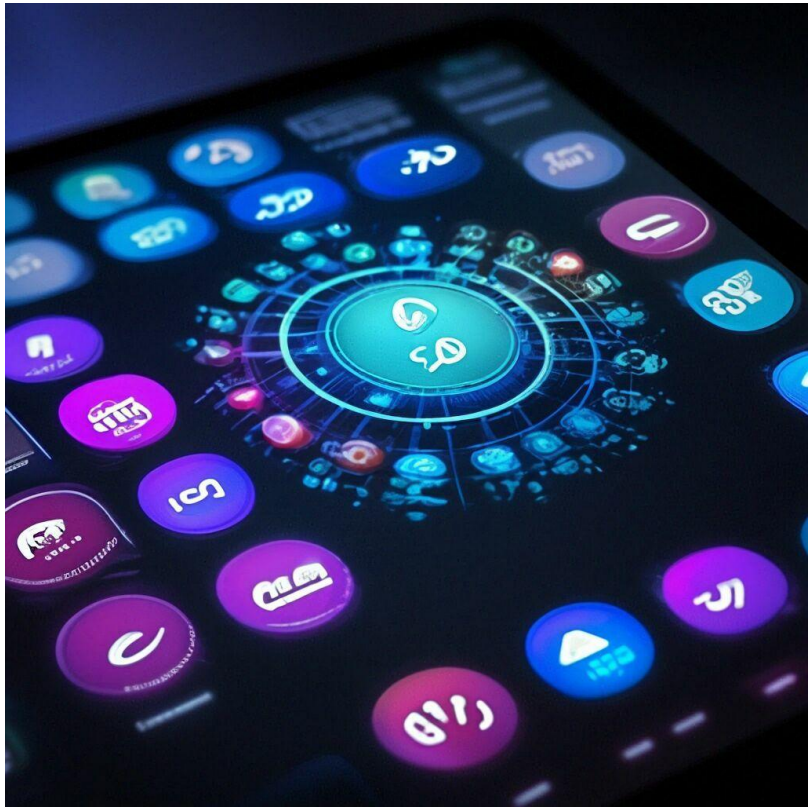
Создание конвейера, включающего в себя все этапы обработки данных

Так-же частью конвейера является кодирование признаков в

OneHotEncoder и StandardSkalern

Последним этапом pipeline является обучение моделей и получение метрики оценки модели roc_auc

Evaluation



Данный этап проекта явился наиболее насыщенным экспериментами

поскольку как и предполагалось на этапе Data Visualization, в данных присутствуют шумы и дисбаланс классов

Вследствии проведенных экспериментов и дополнительной работы с данными были уменьшены факторы снижающие метрики roc_auc модели

По итогам этапа Evaluation

для обучения модели были оставлены следующие признаки

visit_date

visit_number

utm_path

utm_organic

device_screen_resolution

geo

count_duplicates

event_value - целевая переменная

Deployment

Сервис для обращения к модели по API построен на FastApi
Сервис принимает следующие запросы:
get/status - ответ сервиса, о своем статусе
get/version - выводит метаданные модели, включая метрику roc_auc
get/readmi - выводит информацию о форме запроса для предсказания модели
post/predict - предсказание модели на приложенный json запрос

Deployment

Сервис работающий в автоматическом режиме построен на Apache AirFlow

Порядок работы сервиса:

Сервис на основе новых данных, обучает новую модель (предыдущая модель при этом удаляется)

Записывает в файл предсказания на json файлы запросов к модели

Структура проекта:

sber_auto:

/data: исходные данные

data/to_pipeline: данные для обучения модели

/modules: модули с кодом PyCarm для работы проекта:**take_data** - модуль подготовки датасета из исходных данных **pipeline** - модуль конвейера для дообработки данных и обучения модель

predict_api - модуль для предсказаний модели по api
predict - модуль для работы серверного сервиса. **def true_positive_rate**, выводит tpr и confusion_matrix, обученной модели
result_out - модуль вывода предсказания при работе серверного сервиса

/dags: модуль для работы с AirFlow (необходимо скопировать в scheduler)

/jupyter: jupyter_notebook с исследованиями и подготовкой данных

/models: обученная модель проекта

/data/predictions - папка для json файлов запросов предсказаний