Предсказание подключения услуг пользователями Megafon

Geekbrains

Факультет Искусственного интелекта

Чиняев Алексей Викторович

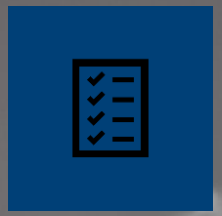
Курсовая работа.

Задача курсовой работы

ПОСТРОИТЬ И ОБУЧИТЬ МОДЕЛЬ-КЛАССИФИКАТОР ДЛЯ ПРЕДСКАЗАНИЯ ПОДКЛЮЧЕНИЯ УСЛУГИ

План обучения модели

ЗАГРУЗКА
И ИСЛЕДОВАНИЕ
ДАННЫХ

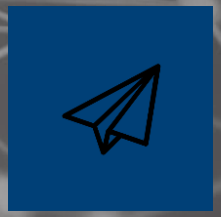


ТРАНСФОРМАЦИЯ И
ДОБАВЛЕНИЕ НОВЫХ
ПРИЗНАКОВ

АНАЛИЗ
ЭФФЕКТИВНОСТИ
ПРИЗНАКОВ

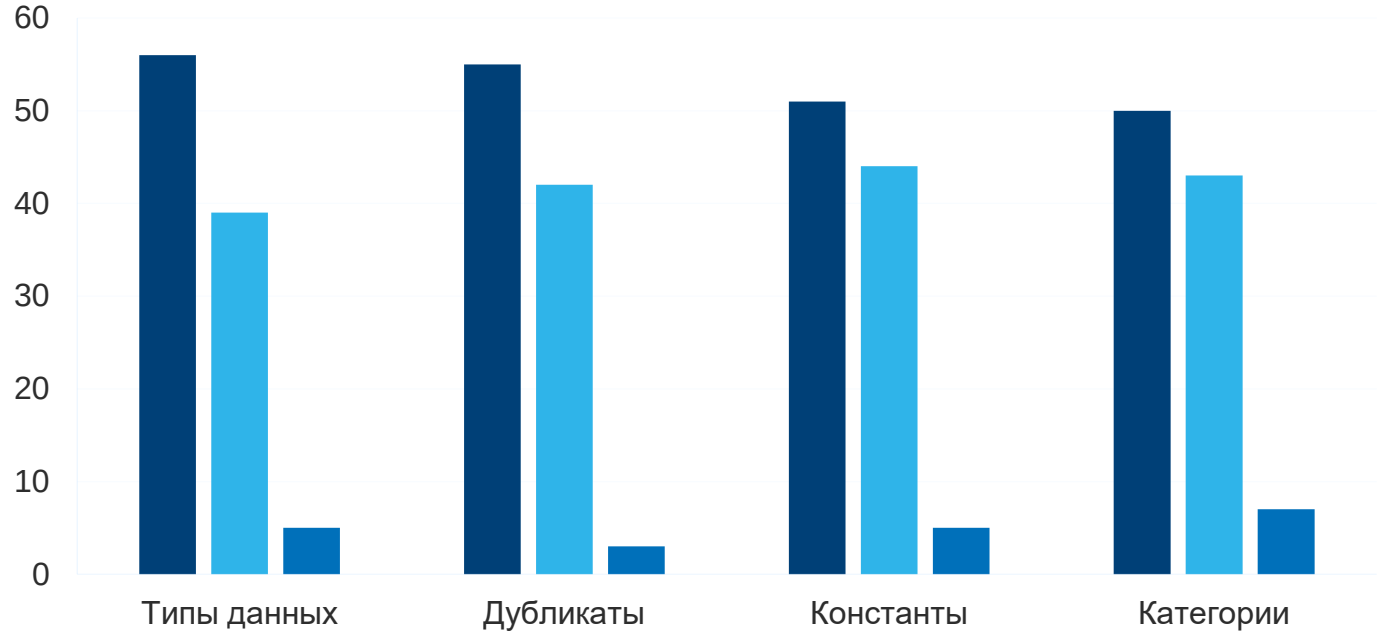
ВЫБОР МОДЕЛИ

ОБУЧЕНИЕ ПОДБОР
ГИПЕРПАРАМЕТРОВ

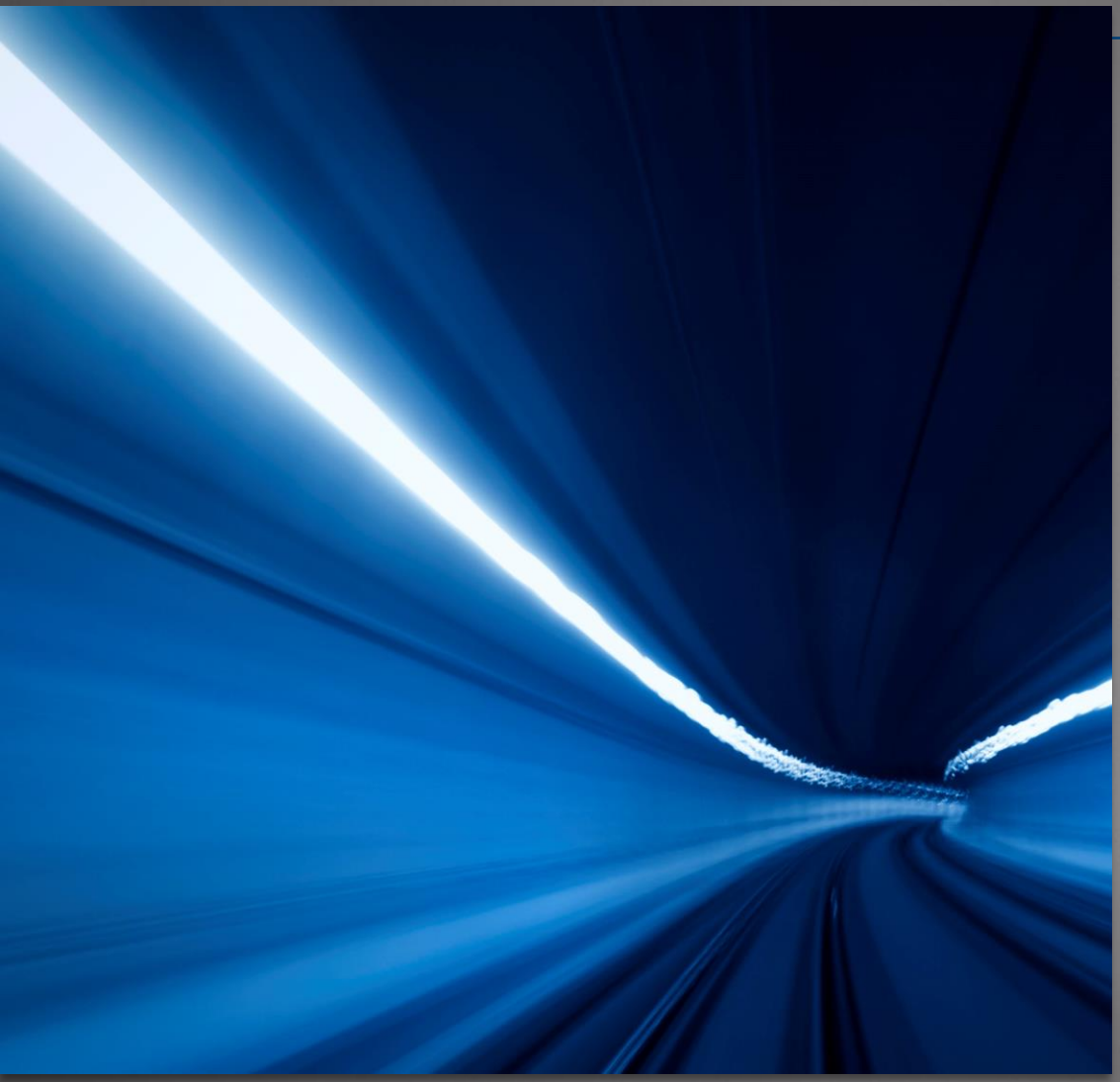


ИТОГИ

Загрузка и первичный анализ данных



Оптимизируем типы данных, удаляем дубликаты, удаляем константы, отмечаем категорийные признаки



Трансформация и добавление новых признаков

Добавляем категорийные признаки, трансформируем существующие, удаляем дубликаты.

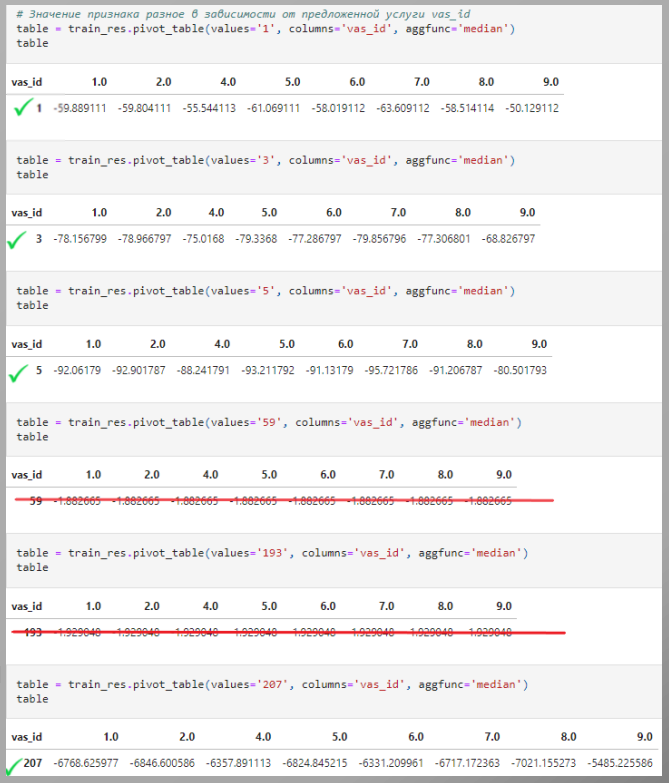
Анализ эффективности признаков

Для анализа эффективности используем «**SelectKBest**» от SK-Learn и «**Feature Selector**» от [Will Koersena]

В начале отбираем 15 лучших признаков с помощью **SelectKBest**, из них отбираем пятерку лучших и проверяем как ведут себя медианные значения признаков для всех предложений услуг, обираем только изменяющиеся, статичные отбрасываем. По оставшимся добавляем новые признаки с медианными значениями.

Для получившего датасета выполняем балансировку.

Используем **Feature Selector.** Отбираем признаки с нулевой важностью и с низкой, удаляем их из датасета.



```
# Значение признака разное в зависимости от предложенной услуги vas_id
table = train_res.pivot_table(values='1', columns='vas_id', aggfunc='median')
table
```

vas_id	1.0	2.0	4.0	5.0	6.0	7.0	8.0	9.0
1	-59.889111	-59.804111	-55.544113	-61.069111	-58.019112	-63.609112	-58.514114	-50.129112

```
table = train_res.pivot_table(values='3', columns='vas_id', aggfunc='median')
table
```

vas_id	1.0	2.0	4.0	5.0	6.0	7.0	8.0	9.0
3	-78.156799	-78.966797	-75.0168	-79.3368	-77.286797	-79.856796	-77.306801	-68.826797

```
table = train_res.pivot_table(values='5', columns='vas_id', aggfunc='median')
table
```

vas_id	1.0	2.0	4.0	5.0	6.0	7.0	8.0	9.0
5	-92.06179	-92.901787	-88.241791	-93.211792	-91.13179	-95.721786	-91.206787	-80.501793

```
table = train_res.pivot_table(values='59', columns='vas_id', aggfunc='median')
table
```

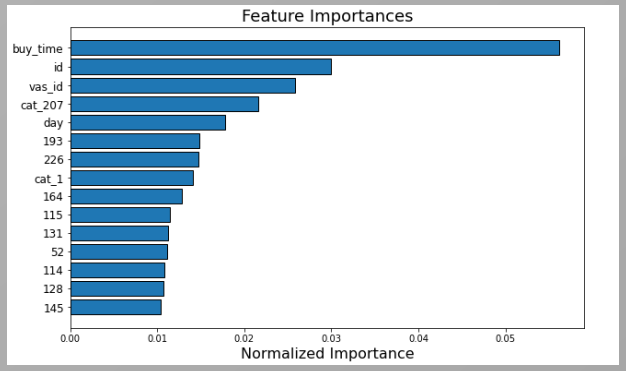
vas_id	1.0	2.0	4.0	5.0	6.0	7.0	8.0	9.0
59	-1.882665	-1.882665	-1.882665	-1.882665	-1.882665	-1.882665	-1.882665	-1.882665

```
table = train_res.pivot_table(values='193', columns='vas_id', aggfunc='median')
table
```

vas_id	1.0	2.0	4.0	5.0	6.0	7.0	8.0	9.0
193	-1.929648	-1.929648	-1.929648	-1.929648	-1.929648	-1.929648	-1.929648	-1.929648

```
table = train_res.pivot_table(values='207', columns='vas_id', aggfunc='median')
table
```

vas_id	1.0	2.0	4.0	5.0	6.0	7.0	8.0	9.0
207	-6768.625977	-6846.600586	-6357.891113	-6824.845215	-6331.209961	-6717.172363	-7021.155273	-5485.225586



Выбор модели

Выбирал из трех моделей, после перебора параметров, получил следующие результаты:

	Precision	Recall	F-score
LogisticRegression	0.689	0.908	0.783
XGBClassifier	0.868	0.920	0.893
CatBoostClassifier	0.887	0.912	0.900

Лучшие результаты для подготовленного датасета показала модель CatBoost, выбор на нее выпал так как она отрабатывала гораздо быстрее чем схожая по результатам XGB.

Итоги

Проведя обработку на тестовых данных получил следующие результаты для модели CatBoost:

	Precision	Recall	F-score
CatBoostClassifier	0.506	0.617	0.556

Спасибо за внимание!

ACHINYAEV@GMAIL.COM