**Центр онлайн-обучения «Нетология-групп»**

Дипломный проект

Определение паттернов потребления онлайн-школы «Фоксфорд».

Выполнил:

Пастушенко Юрий Павлович

Ментор:

Сачко Егор Геннадьевич

Москва 2017

# Постановка задачи

## Описание заказчика

Фоксфорд – это веб-ресурс, предоставляющий образовательные услуги учащимся средних школ. Основными продуктами компании являются платные курсы, состоящие из вебинаров, в ходе которых учащиеся могут получить домашние задания, проверяемые преподавателями курса.

После регистрации нового пользователя ему предоставляется небольшое количество бесплатных услуг, после исчерпания которых пользователь либо оплачивает курс, либо прекращает пользование сервисом.

Ежедневно на веб-ресурсе происходит в среднем более четырех тысяч регистраций, из которых менее 5% процентов клиентов в последствии воспользуются хотя бы одной платной услугой. Типовая задача отдела продаж состоит в повышении конверсии из потенциальных клиентов в клиенты путем взаимодействия с потенциальным клиентом. Отделу продаж Фоксфорда доступны две формы взаимодействия: электронные письма и телефонные звонки. Электронные письма не входят в область ответственности данного диплома.

## Формулировка задачи

При приросте потенциальных клиентов в 4 тысячи человек в сутки и конверсии в 5%, для покрытия всех клиентов телефонными звонками, пришлось бы обзвонить примерно 3800 человек в сутки. На текущий момент отдел продаж не способен справится с таким количеством звонков, поэтому руководством была поставлена задача: сформировать выборку клиентов, которые с большой вероятностью купят продукт, если с ними свяжется сотрудник отдела продаж.

Таким образом цель проекта – построение модели, предсказывающей клиентов, которые с большей вероятностью готовы оплатить подписку, если с ними свяжется сотрудник отдела продаж.

Необходимо учесть, что среди клиентов есть те, которые сами заказали обратный звонок. Этих клиентов нужно исключать из рассмотрения.

## Как решается задача сейчас

На текущий момент прямого обзвона потенциальных клиентов Фоксфорда не производится ввиду большого объема и низкой конверсии звонков.

## Стратегия решения

В информационной системе Фоксфорда существуют следующие базы данных (о которых нам известно):

* отчетность на движке «Metabase»
* «Big query», содержащая данные о трафике на веб-ресурсе foxford.ru
* AmoCRM, содержащая в том числе данные о звонках потенциальным клиентам.
* ERP, в которой ведется управленческий учет

Для работы над проектом был предоставлен доступ к базе Metabase. Доступ к остальным базам ввиду технических сложностей предоставлен не был.

Т.к. общая цель задачи – разгрузить отдел продаж путем предоставления для обзвона только тех клиентов, которые не приобрели бы курс без звонка сотрудника отдела продаж, то для получения качественного результата, нужно было бы изучить влияние того насколько повлиял звонок сотрудника отдела продаж на решение о покупке курса. Однако, так как доступа к базе, содержащей данные о звонках предоставлено не было, пришлось выдвигать гипотезы, которые могли бы помочь выдавать преимущественно нужных для обзвона клиентов.

В ходе работы над проектом было решено свести задачу к задаче бинарной классификации клиентов на **оплативших в течение нескольких дней с момента регистрации** и на **не оплативших в течение этого же времени**. Такая постановка позволит использовать существующие стабильные алгоритмы классификации и сконцентрироваться на проработке параметров, которые могут влиять на класс клиента. Результатом работы модели будет являться вероятность отнесения клиента к классу оплативших в первые несколько дней после регистрации.

## Метрики качества

Типовым инструментом для контроля качества результатов работы бинарной классификации являются ROC-кривая и кривая Precision-Recall. Обе этих кривые будут использованы при анализе результатов работы модели.

Т.к. в качестве результата работы модели планируется вероятность отнесения клиента к одному из двух классов, то нет смысла использовать метрики, работающие только для одного значения порога отсечения вероятности (такие как Accuracy или F-score). В качестве числового показателя качества была выбрана метрика ROC-AUC (площадь под ROC-кривой). Эта метрика позволяет оценить качество работы всей модели и напрямую зависит от ROC-кривой, которая также будет использована для анализа результатов работы модели.

# Анализ

### Количество данных

В базе Metabase содержаться данные о клиентах, зарегистрировавшихся на foxford.ru начиная c 2010 года. Как видно из графика (рис.1) активный рост количества регистраций начался только в 2015м году. Также необходимо учитывать тот факт, что правила ведения учета клиентов неоднократно менялись с 2015 года. В ходе консультаций с сотрудниками фоксфорда было установлено, что пригодными для анализа можно считать данные с сентября 2016 года. Однако сама система Metabase плохо работает с большими выгрузками, поэтому начальная дата отсечения данных была установлена на 01.01.2017. Сама дата выведена эмпирически.

Также необходимо учесть, что на сайте foxford.ru у пользователя есть возможность заказать обратный звонок, то есть если пользователь введет свой телефон и имя, то ему перезвонит сотрудник отдела продаж и выяснит его потребности. Таких пользователей нам нужно исключить из рассмотрения, т.к. обзванивать их нам нет никакого смысла, т.к. они не входят в нашу целевую группу.

С учетом описанных выше факторов, итоговая выборка клиентов для анализа составляет **более 500 тысяч человек**.

## Целевая переменная

Так как модель должна быть нацелена на предсказания клиентов, которые готовы оплатить какую-либо платную услугу, нам нужно представление о том, сколько клиентов вообще оплатили услуги. На графике (рис. 2) представлено количество зарегистрировавшихся клиентов и доля клиентов, воспользовавшихся хотя бы одной платной услугой. Из графика можно заключить, что в целевой выборке доля клиентов, оплативших хотя бы одну услугу составляет от 1,5 до 17 процентов.



Общее количество клиентов, купивших хотя бы одну услугу составляет более 27ми тысяч человек.

Также важным параметром для модели может выступать время, которое прошло с момента регистрации до момента первой оплаты. Для клиентов, которые купили хотя бы одну платную услугу, представим зависимость количества клиентов от количества полных суток, прошедших с момента регистрации до момента первой покупки на графике (рис. 3).



Из графика видно, что более половины клиентов покупают платную услугу в первые сутки после регистрации. Клиентов, которые купили первую платную услугу на вторые сутки после регистрации на порядок меньше. Крайняя правая колонка на графике показывает клиентов, которые совершили первую покупку через 30 или более суток, то есть у этого распределения довольно тяжелый хвост.

После консультаций с заказчиком было установлено, что после звонка сотрудника отдела продаж, сотруднику засчитают бонус за клиента, только в том случае, если клиент оплатит в течение 30 дней после звонка. Таким образом несмотря на тяжелый хвост, **целевой переменной будем считать признак того, что клиент купил какую-либо услугу в течение 30 суток после регистрации**. В целевой выборке количество таких событий **более 22х тысяч** (из более 500 тысяч), что составляет примерно 4,5% от всей выборки.

Таким образом классы клиентов, оплативших в течение 30 дней и не оплативших в течение этого времени, получаются несбалансированными.

Таблица 1. Признаки для расчета целевой переменной

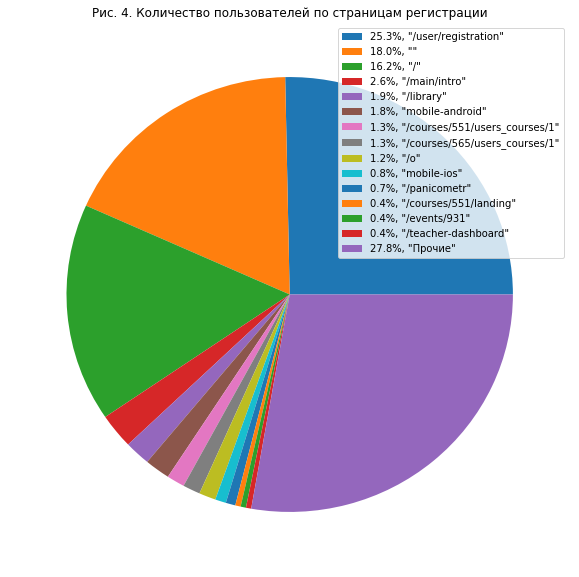
|  |  |  |  |
| --- | --- | --- | --- |
| **Название** | **Описание** | **Тип** | **Обработка** |
| Реферер | Адрес ссылки. Реферер, с которого пользователь попал на страницу регистрации. | Категориальный | WOE |
| Цель обучения | Идентификатор. Указывается пользователем при регистрации. | Категориальный | Бинаризация |
| Планирует ли пользователь заниматься с репетитором | Идентификатор. Указывается пользователем при заполнении «входного теста». | Категориальный | Бинаризация |
| Сколько времени готов посвящать обучению | Идентификатор. Указывается пользователем при заполнении «входного теста». | Категориальный | Бинаризация |
| Причина, по которой хочет дополнительных занятий | Идентификатор. Указывается пользователем при заполнении «входного теста». | Категориальный | Бинаризация |
| Зарегистрировался в будний или в выходной день | Да/нет. Вычисляется на из даты регистрации | Бинарный | Нет |
| Зарегистрировался в ночное время | Да/нет. Вычисляется на из даты регистрации | Бинарный | Нет |
| Группа маркетинговых источников | Идентификатор. Обобщенный канал продвижения | Категориальный | Нет |
| Класс ученика | Идентификатор. Указывается при регистрации на сайте | Категориальный | Бинаризация |
| Пользователь указал дату рождения | Да/нет. На основании действий пользователя. | Бинарный | Нет |
| Пользователь является родителем ученика | Да/нет. Указывается при регистрации на сайте | Бинарный | Нет |
| Пользователь является преподавателем | Да/нет. Указывается при регистрации на сайте | Бинарный | Нет |
| Пользователь ввел номер телефона | Да/нет. На основании действий пользователя. | Бинарный | Нет |
| Подтвердил номер телефона | Да/нет. На основании действий пользователя. | Бинарный | Нет |
| Регион пользователя | Идентификатор региона в таблице в базе Metabase. Есть только у пользователей, указавших адрес | Категориальный | Бинаризация |
| Пол пользователя | Мужской/Женский. Вычисляется на основе данных стороннего сервиса по имени. | Бинарный | Пустые значения обрабатываются как отдельная категория |

## Признаки (фичи) для обучения модели

Использованные признаки перечислены в таблице 1. Большую их часть удается получить из базы Metabase простым запросом. Некоторые признаки требуют дополнительной обработки. Далее каждый признак будет разобран подробнее. Анализ признаков, в основном, проводился с использованием SQL и не документировался. Код построения приведенных здесь графиков содержится в файле [charts.ipynb](https://github.com/pastushenkoy/DataScientist/blob/master/dip/charts/Charts.ipynb).

### «Реферер»

Страница регистрации представляет собой реферер, с которого пользователь перешел на страницу регистрации. Сейчас в списке рефереров содержится более 10 тысяч уникальных элементов. Бинаризовывать такое количество уникальных элементов довольно затруднительно. Пирог встречаемости рефереров показан на рис. 4.



Из графика видно, что несмотря на то, что есть несколько доминирующих рефереров, хвост настолько огромный, что не получится просто объединить все рефереры, встречаемость которых меньше определенного значения в одну группу, т.к. тогда мы потеряем больше 25% данных. Вследствие этого, при обработке этого признака, для каждого реферера будет рассчитано значение показателя «weight of evidence» (WOE). Это значение позволит нам сохранить всю информацию и не потерять точность. Минусом WOE по сравнению с бинаризацией является трудность интерпретации результатов. Формула для расчета WOE:

Здесь N – общее количество элементов в выборке.

Также из графика видно, что примерно у 18% клиентов реферер не заполнен. Незаполненные значения будут интерпретированы как отдельная категория и для нее также будет рассчитан WoE.

### «Цель обучения»

Обязательное при регистрации поле. Содержит идентификатор цели обучения, то есть является категориальным признаком. Заполнено у всех пользователей целевой выборки. Бинаризуется перед обучением модели.

### «Планирует ли пользователь заниматься с репетитором»,

### «Сколько времени готов посвящать обучению», «Причина, по которой хочет дополнительных занятий»

Эти признаки пользователь вводит при заполнении входного теста на foxford.ru. Все они хранятся в одном и том же поле таблицы базы данных в json-формате. Каждый из них представляет собой идентификатор соответствующей категории. Эти идентификаторы не хранятся в Metabase, а потому не интерпретируемы.

Поле, в котором хранятся эти признаки заполнено не более чем у 50 тысяч клиентов, однако даже сам факт наличия значения в этом поле может повлиять на результат обучения модели, поэтому решено было оставить эти признаки.

Каждый из этих признаков бинаризуется перед запуском модели. Помимо очевидного назначения, бинаризация также решает проблему с большим количеством незаполненных значений.

### «Зарегистрировался в будний или в выходной день», «зарегистрировался в ночное время»

Эти два признака получаются прямым преобразованием даты регистрации, которая заполнена у всех клиентов. Оба они бинарные и не требуют дополнительной обработки.

### «Группа маркетинговых источников»

Этот признак представляет собой идентификатор канала продвижения. Таких каналов в системе 53, из них только по 35ти есть пользователи. На рис. 5 показан пирог распределения пользователей по маркетинговым источникам. Видно, что есть несколько доминирующих источников и небольшой хвост. При таком небольшом количестве источников, мы можем спокойно бинаризовывать этот признак. Группа маркетинговых источников заполнена у всех пользователей целевой выборки.



### «Класс ученика»

Один из самых важных показателей – это идентификатор класса ученика. Класс заполняют клиенты при регистрации на сайте. Он заполнен у всех клиентов целевой выборки. В списке классов содержаться все 11 классов средней школы и одно специальное значение «Выпускник», означающее, что ученик уже закончил школу. Признак бинаризуется перед запуском модели.

### «Пользователь указал дату рождения», «является родителем ученика», «является преподавателем», «ввел номер телефона», «подтвердил номер телефона»

Все эти признаки – это значения одной из таблиц базы Metabase. Все они бинарные, заполнены у всех пользователей и не требуют дополнительной обработки.

### «Регион»

Этот признак – это идентификатор региона в собственной таблице в базе Metabase, причем это не обязательно регион России. Регион есть только у тех пользователей, которые ввели адрес. Таких пользователей около 27ми тысяч. Самих регионов в таблице 162. Такое количество позволяет нам применить бинаризацию этого признака.

### «Пол клиента»

Пола в явном виде нет в базе данных. Для его определения пришлось спарсить сайт kakzovut.ru и получить списки мужских и женских русских имен. Путем поиска в этих списках был определен пол примерно у 80% клиентов целевой выборки. Значения пола: 0 – женский, 1 – мужской. Не заполненные значения интерпретируются как отдельная категория.

Код парсинга сайта kakzovut.ru содержится в файлу [name\_parse.ipynb](https://github.com/pastushenkoy/DataScientist/blob/master/dip/name_parse.ipynb).

### «Резульаты ЕГЭ для школы» (исключенный признак)

Была предпринята попытка обогатить данные результатами ЕГЭ для московских школ. Данные о результатах ЕГЭ есть в открытом доступе в датасете «Результаты ЕГЭ» на портале открытых данных по адресу <https://data.mos.ru/opendata/7719028495-rezultaty-ege-dogm>. В этом наборе данных одна строка соответствует результату ЕГЭ для одной школы. Результаты ЕГЭ представлены в 2х колонках:

* «Количество обучающихся, набравших более 220 баллов на трех экзаменах»
* «Количество обучающихся, набравших не менее 160 баллов на трех экзаменах»

Так как количество обучающихся в школах различно, нужно нормировать значения в каждой из этих колонок на значение в колонке «Общее количество обучающихся».

К сожалению, в таблице с результатами ЕГЭ отсутствует какая-либо информация, которая позволила бы нам связать школу из открытых данных со школой из базы данных Metabase, однако такие данные есть в соседнем датасете «Образовательные учреждения города Москвы», расположенном по адресу <https://data.mos.ru/opendata/7719028495-obrazovatelnye-uchrejdeniya-goroda-moskvy>. В этой таблице есть контактная информация (телефоны, адреса электронной почты), которая также содержится в базе Metabase.

Связать эти два датасета между собой можно по полю ОГРН.

После связывания наборов данных и удаления ненужных колонок, мы получили возможность дополнить наши данные результатами ЕГЭ для московских школ. Удалось связать примерно 600 школ из 1200. Однако, к сожалению, эти данные практически никак не повлияли на результат в метрике ROC-AUC, а для работы с ними, пришлось бы отдельно обучать модель для московских и региональных школ, что довольно затруднительно. Поэтому решено было отказаться от использования этих данных на раннем этапе.

## Выбор модели для классификации

При выборе модели оценивались 3 классификатора:

* RandomForestClassifier
* XGBoostClassifier
* DecisionTreeClassifier с глубиной 10.

Для оценки применялось случайное разбиение целевой выборки на тренировочную и тестовую. В качестве метрики использовалась ROC-AUC. Результаты представлены в таблице 2.

Таблица 2.

|  |  |  |  |
| --- | --- | --- | --- |
| **Классификатор** | **Номер сэмпла** | **Test ROC-AUC** | **Время выполнения, с** |
| RandomForestClassifier | 1 | 0.78603 | 61 |
| 2 | 0.85294 | 69 |
| 3 | 0.78928 | 65 |
| 4 | 0.85723 | 71 |
| XGBoostClassifier | 1 | 0.85539 | 149 |
| 2 | 0.92743 | 149 |
| 3 | 0.85610 | 144 |
| 4 | 0.92821 | 148 |
| DecisionTreeClassifier | 1 | 0.84678 | 54 |
| 2 | 0.90447 | 57 |
| 3 | 0.84365 | 46 |
| 4 | 0.91074 | 51 |

Точнее всех оказался XGBoostClassifier, но он же обучался в 2 раза дольше остальных. Лучшее время обучения и приемлемую точность показал DecisionTreeClassifier.

Для дальнейшего использования был выбран XGBoostClassifier.

# Методика решения

## Последовательность действий при работе модели

1. Выполнение запроса к базе Metabase, который сформирует файл csv с описанными выше признаками. Файл сохраняется в каталог рядом с моделью.
2. Модель загружает файл в Pandas Dataframe.
3. Удаляются колонки, которые предполагается использовать в будущем.
4. Дата регистрация пользователя переводится в стандартный формат даты и времени из timestamp’а.
5. Определяется пол каждого пользователя.
6. На основе даты регистрации и значения количества дней между датой регистрации и датой оплаты рассчитывается целевая переменная.
7. Данные разделяются на учебную и рабочую (тестовую или боевую) выборки.
8. Для тренировочной выборки, рассчитываются значения показателя WOE для соответствующих колонок.
9. Для тренировочной выборки, бинаризуются соответствующие категориальные признаки.
10. В тренировочной выборке незаполненные к этому моменту значения заполняются медианами.
11. В тренировочной выборке взвешиваются классы.
12. **Модель обучается на тренировочной выборке**
13. В рабочую выборку подставляются значения WOE, рассчитанные для тренировочной.
14. В рабочей выборке бинаризуются соответствующие категориальные колонки.
15. Состав колонок тренировочной и рабочей выборки приводятся в соответствие:
    1. Из рабочей выборки убираются колонки, которых не было в тренировочной
    2. В рабочую выборку добавляются колонки, которые есть в тренировочной, но нет в рабочей. Эти колонки заполняются значениями по умолчанию, принятыми для их типов.
16. В тестовой выборке незаполненные к этому моменту значения заполняются медианами.
17. **Выполняется предсказание на рабочей выборке.**

## Описание модели

Итоговая модель представляет собой класс-декоратор, в который через конструктор можно передать любой классификатор, поддерживающий интерфейс sklearn (мы будем пользоваться XGBoostingClassifier). Сам класс модели выполняет предобработку тренировочной и рабочей выборок. Модель описана в файле [new\_model.ipynb](https://github.com/pastushenkoy/DataScientist/blob/master/dip/model_new.ipynb) в разделе 2.

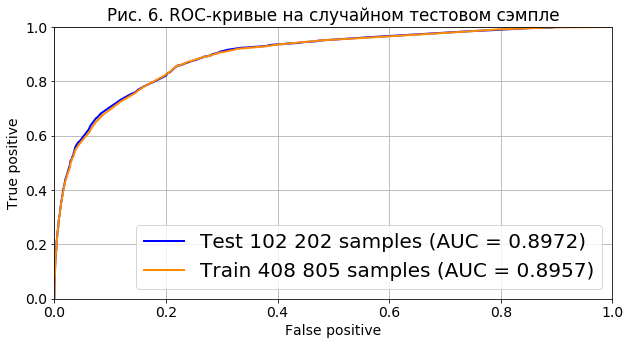
Перед выполнением, модели надо указать какие колонки как обрабатывать. Это делается через параметры конструктора, куда передаются следующие списки колонок:

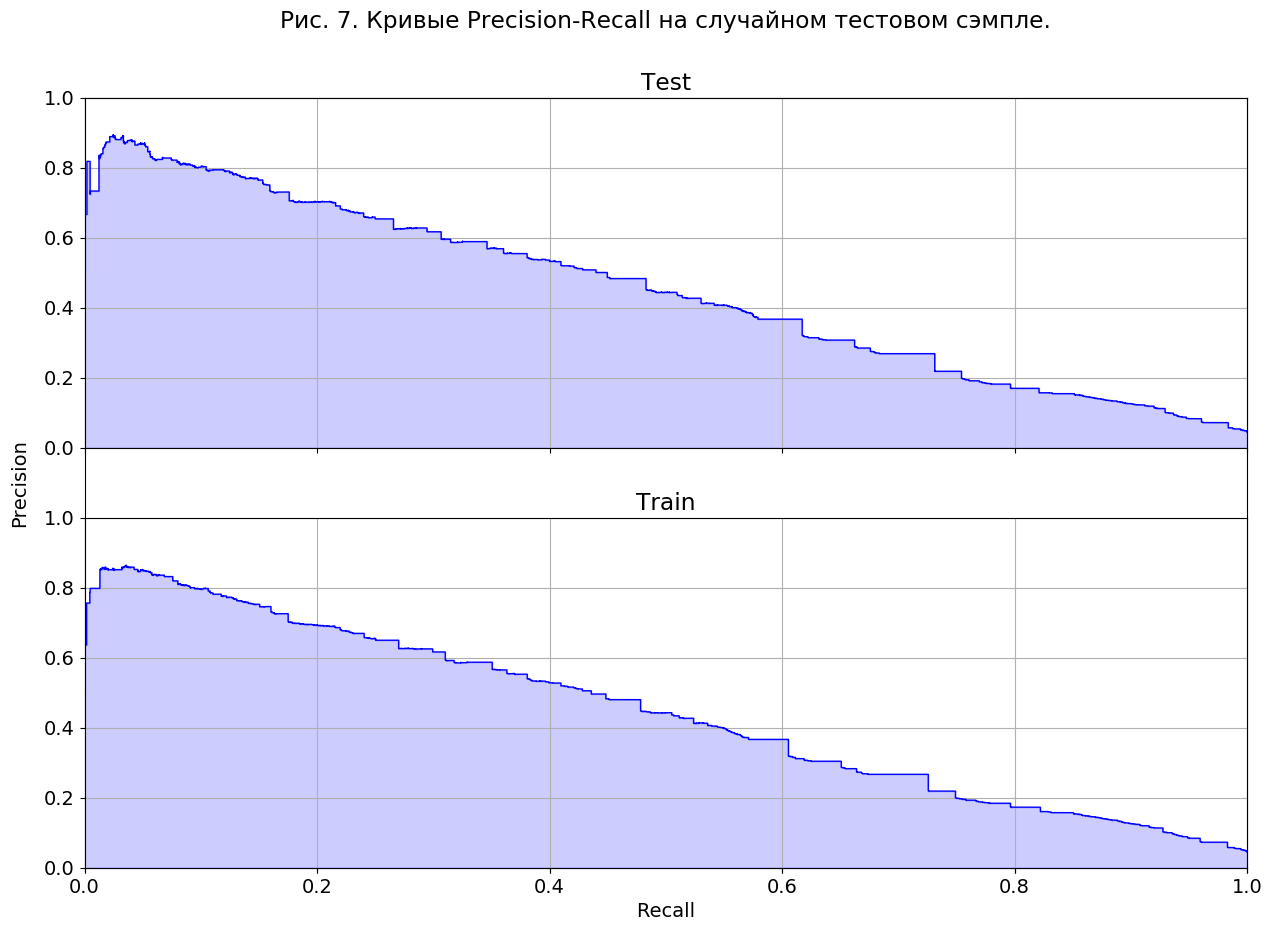
* Колонки, которые нужно преобразовать в WOE
* Колонки, которые нужно бинаризовать
* Колонки, которые нужно удалить

### Тестирование на случайной выборке

Код тестирования содержится в файле [new\_model.ipynb](https://github.com/pastushenkoy/DataScientist/blob/master/dip/model_new.ipynb) в разделе 3.

Результаты тестирования на случайной выборке представлены на рис. 6 и 7. Размер тренировочной выборки: 408805 строк; тестовой: 102202 строки. Метрика ROC-AUC для тестовой выборки составила 0.8972. Кривая precision-recall равномерно убывает.





Результаты работы модели на случайной выборке говорят о хорошем качестве результатов модели.

### Тестирование на выборке, отсеченной по времени

Код тестирования на выборке, отсеченной по времени содержиться в файле [new\_model.ipynb](https://github.com/pastushenkoy/DataScientist/blob/master/dip/model_new.ipynb) в разделах 3 и 4.

Несмотря на то, что модель показывает хорошие результаты на случайной тестовой выборке, интересно посмотреть, как будет вести себя модель, если тестовую выборку ограничить по времени.

время

Дата регистрации  
последнего клиента в целевой выборке

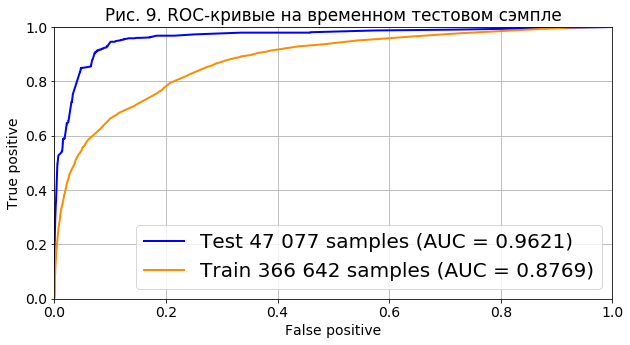
30 суток

Тестовая выборка 7 суток

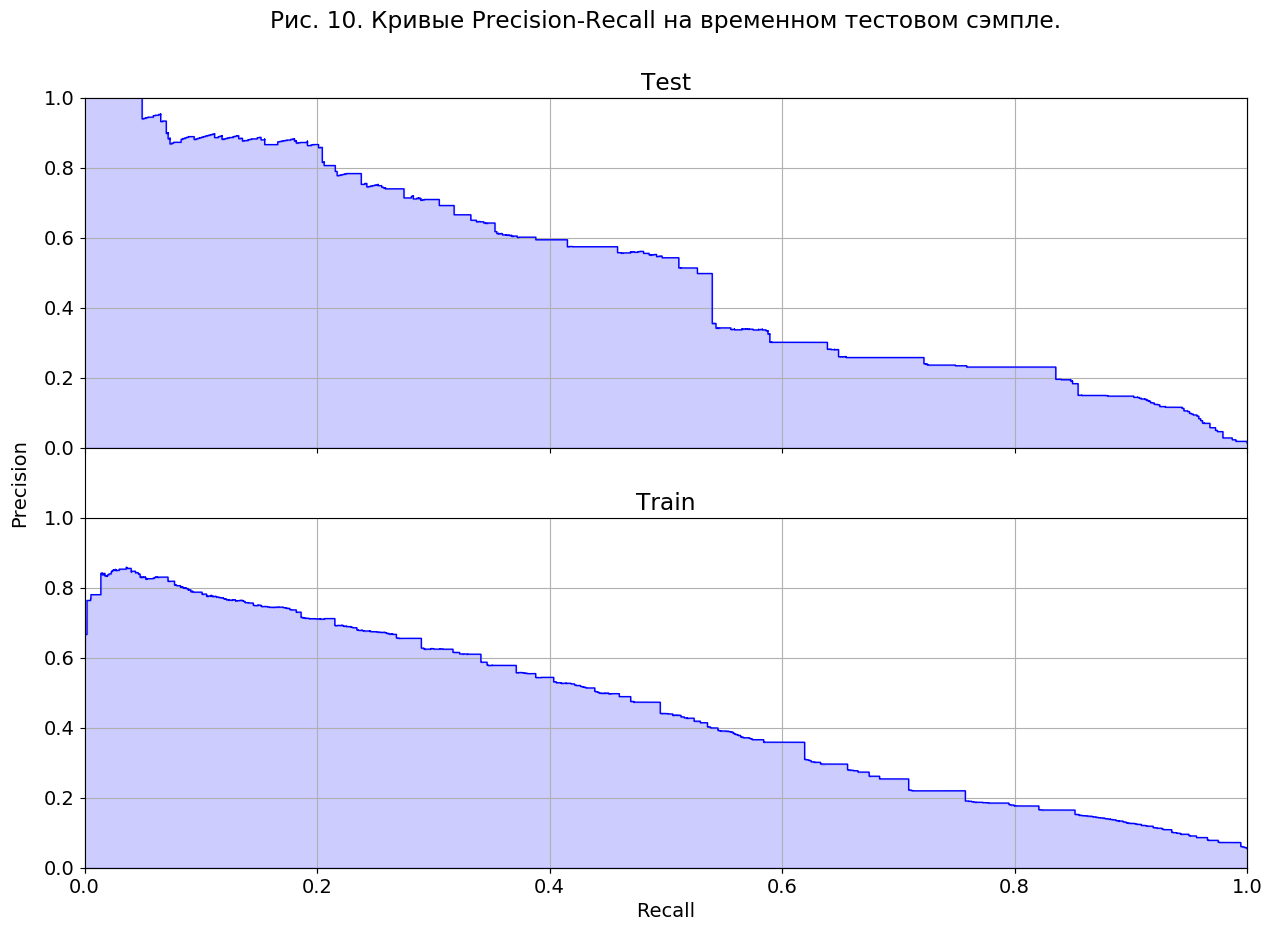
Рис. 8. Визуализация получения тестового сэмпла

Целевой переменной для модели является «купит ли клиент хотя бы одну платную услугу в течение 30 дней после регистрации». Таким образом тестовый сэмпл должен оставлять минимум 30 дней с момента регистрации последнего клиента в сэмпле, чтобы иметь полную информацию о том к какому классу относится клиент. Составим тестовый сэмпл следующим образом: возьмем 7 суток, которые истекли 30 суток назад начиная с даты регистрации последнего клиента в целевой выборке (рис. 8).

Тестовая выборка содержит 47 077 элементов, среди которых 626 целевых событий (1,3%).  
Тренировочная выборка содержит 366 642 элемента, среди которых 19 969 целевых событий (5,4%).



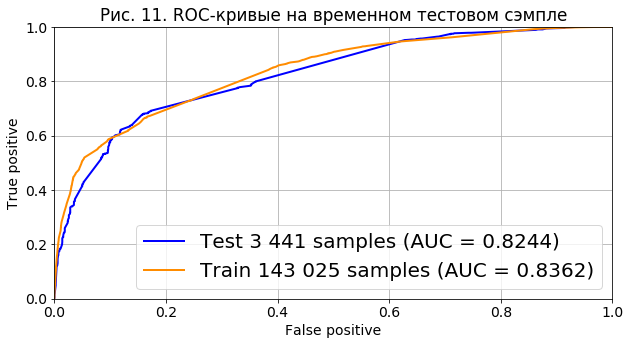
Результаты тестирования на таком сэмпле представлены на рис. 9 и 10. Качество на тестовой выборке получилось значительно выше, чем качество на тренировочной. Это объясняется тем, что количество целевых событий в сформированной тестовой выборке в несколько раз ниже, чем в тренировочной, а так как классы изначально были не сбалансированы, модель предсказывает наступление целевого события значительно реже, чем наступление.



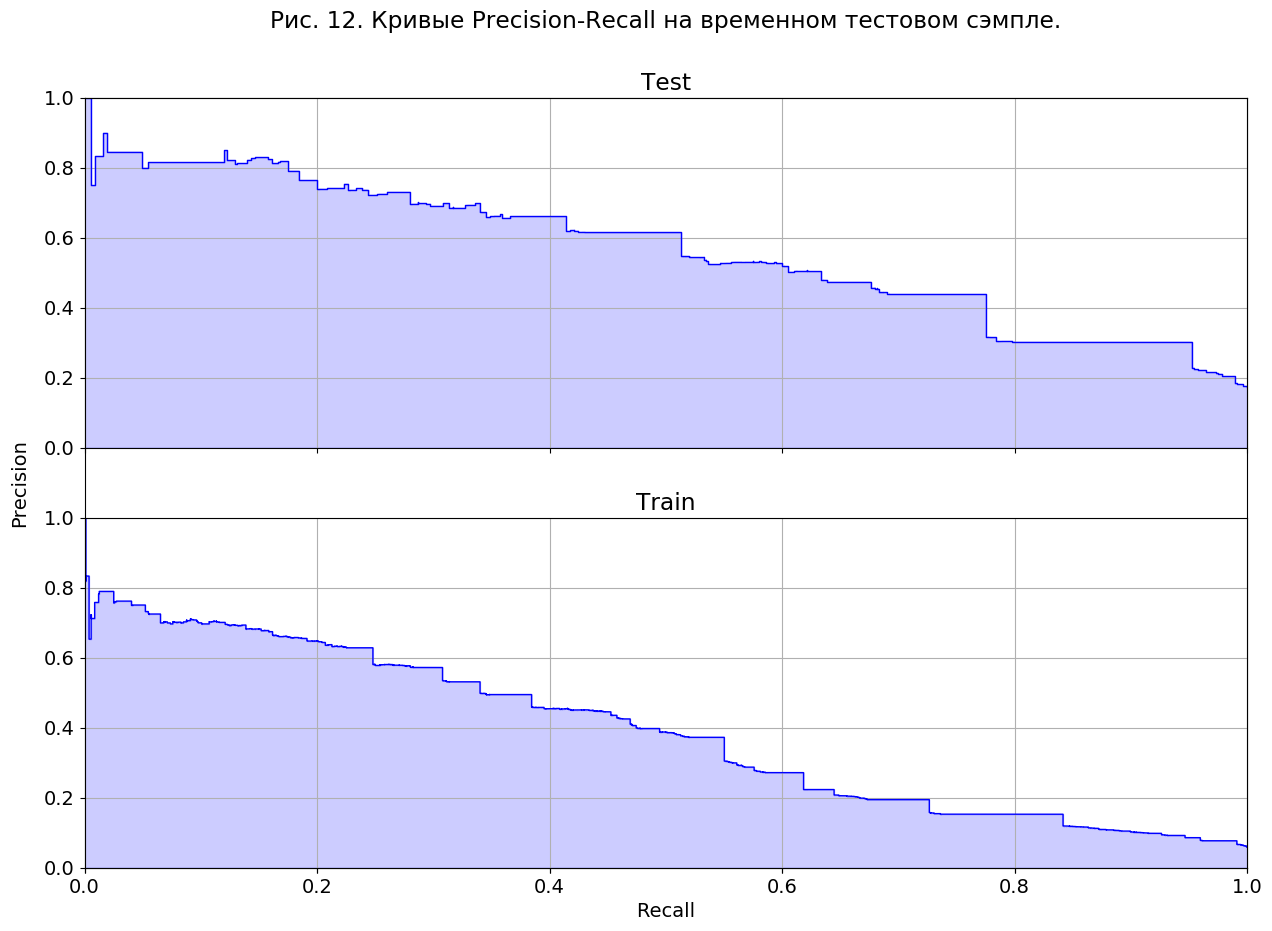
Для проверки этой гипотезы, можно сформировать тестовую выборку таким образом, чтобы в ней было значительно больше целевых событий, чем в тренировочной. Вернемся к графику на рис. 1. Из него видно, что максимальный процент целевых событий был в июне. Сформируем тестовую выборку аналогично предыдущему случаю, но таким образом, чтобы 7 суток тестовой выборки начались 15 июня, а закончились, соответственно 22 июня.

Такая тестовая выборка содержит 3 441 элементов, среди которых 565 целевых событий (16,4%).  
Тренировочная выборка содержит 134 724 элемента, среди которых 8 301 целевых событий (5,8%).

Результаты такой проверки показаны на рис. 11 и 12. Из графиков видно, что тестовая кривая теперь находится значительно ближе, чем в предыдущем случае.



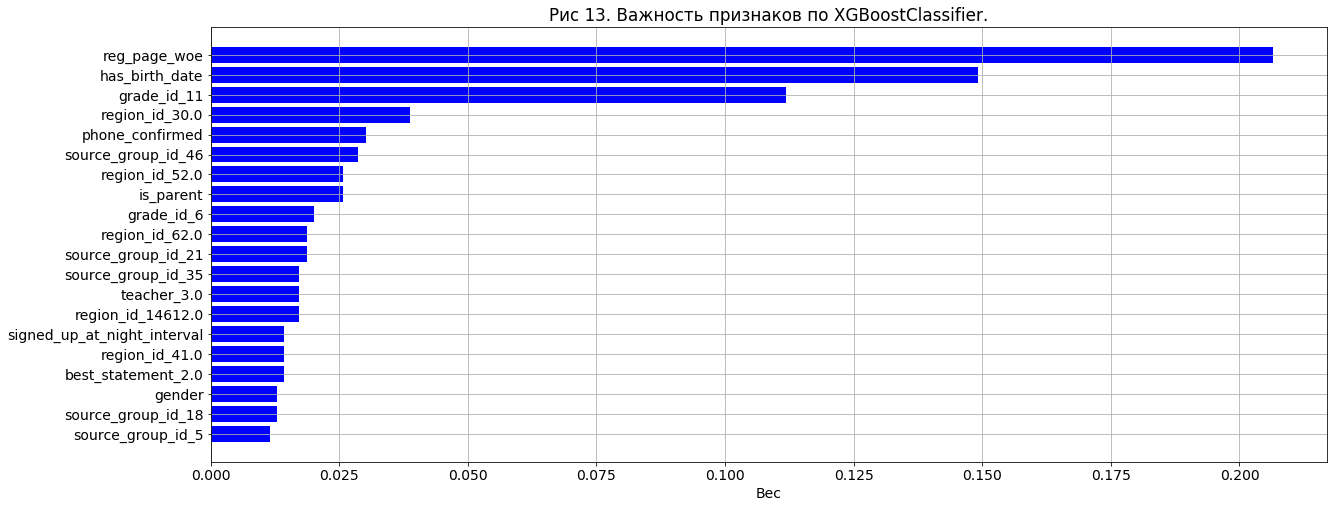
Кроме того, эти графика подтверждают стабильность модели, т.к. несмотря на сильное изменение процента целевых событий и размера выборки, результат для тестовой выборки не сильно хуже, чем для тренировочной.



# Результаты

Созданная модель позволяет с хорошей точностью предсказывать вероятность того, что клиент купит хотя бы одну платную услугу в течение 30 дней. Устойчивость модели к шумам была подтверждена кросс-валидацией в момент выбора классификатора. Косвенно устойчивость подтверждается результатами тестирования на временном сэмпле.

Важность признаков по версии XGBoostClassifier представлена на рис. 13.



Самыми важными признаками оказались:

1. Реферер (reg\_page\_woe)
2. Наличие даты рождения в профиле (has\_birth\_date)
3. Клиент является выпускником (grade\_id\_11)
4. Клиент из Xабаровского края (region\_id\_30)
5. Клиент подтвердил телефон (phone\_confirmed)
6. Канал продвижения – профориентация (source\_group\_id\_46)
7. Клиент из Московской области (region\_id\_52)
8. Клиент является родителем (is\_parent)
9. Клиент обучается в 11ом классе (grade\_id\_6)

К сожалению, на внедрение модели в работу не хватило времени. В данный момент прорабатывается алгоритм, по которому будет производится тестирование.

## Выдача для обзвона

Мы получили модель, способную предсказывать вероятность покупки, однако изначально цель создания модели была поставлена как выдача клиентов, которые с большей вероятностью готовы купить услугу, если с ними свяжется сотрудник отдела продаж. Если мы будем каждый день просто выдавать наиболее теплых клиентов, то есть возможны две проблемы:

1. Один и тот же клиент может появится в выборках для нескольких дней подряд, а т.к. у нас нет данных о звонках, мы никак это не учтем.
2. Клиент может купить услугу сам, без помощи отдела продаж и, в этом случае, звонок будет совершен зря.

Чтобы избежать указанных выше проблем, мы решили выдавать для обзвона только клиентов, которые зарегистрировались не ранее, чем сутки назад, но еще не купили ни одной платной услуги. Такой подход имеет основания, т.к. из рис. 1 видно, что больше половины клиентов покупает платную услугу в первые сутки после регистрации. Следовательно, если клиент не купил услугу в первые сутки, но модель считает его теплым, то ему следует позвонить.

## Методика тестирования

Тестировать модель предполагается по методу A/B-тестирования. Выделенные сотрудники отдела продаж будут разделены на две группы. Группа A будет обзванивать клиентов, по выборке, сформированной руководителем отдела продаж. Группа B будет обзванивать клиентов, которых выдаст модель.

Для проверки эффективности звонков, каждая из выборок A и B будет разделена на две: выборку для обзвона и тестовую выборку. В тестовую выборку будет попадать каждый 4ый клиент. В последствии, доля оплативших клиентов из тестовой выборки будет сравниваться с долей оплативших клиентов выборки для обзвона для выяснения того на сколько звонок повлиял на решение клиента о покупке платной услуги.

Клиенты, присутствующие в выборках A и B должны учитываться в обеих выборках, но обзванивать их будут только сотрудники группы A. Такое решение принято из-за сложности организации процесса тестирования.

Алгоритм процесса тестирования в настоящий момент обсуждается. На текущий момент он выглядит так:

1. В 5:00 каждые сутки данные поступают в базу Metabase.
2. Руководитель отдела продаж создает выборку для обзвона для группы А (без использования модели):
   1. выполняет отчет, который формирует стандартную выборку для обзвона клиентов.
   2. передает список для обзвона сотрудникам группы А
   3. сохраняет выборку в файл csv в папку с моделью (для того, чтобы модель могла исключить из обзвона звонки из «ручной» выборки)
   4. сохраняет выборку в выделенную папку с результатами
3. Руководитель отдела запускает модель и формирует выборку для группы B (с использованием модели):
   1. заходит в Metabase и выполняет отчет по запросу, который я предоставлю.
   2. сохраняет результат запроса в файл unload.csv в папку с моделью
   3. запускает файл execute.bat (запуск расчета модели). После окончания расчета модель выдаст 3 файла: *test\_<текущая дата>.csv*, *prod\_<текущая дата>.csv* и call\_<текущая дата>.csv.
   4. сохраняет все три файла в выделенной папке куда складываются все результаты для анализа
   5. формирует в Metabase некий отчет, которые выводит данные клиента, которые нужны для звонка сотруднику отдела продаж. В этот отчет в качестве параметра передает пользователей, которые содержаться в файле *call\_<текущая дата>.csv*
   6. передает результаты выполнения отчета сотрудникам отдела продаж из группы B
4. Сотрудники отдела продаж обеих групп:
   1. обзванивают клиентов своих выборок и каким-то образом отмечают клиентов, с которыми состоялся разговор
   2. передают список клиентов руководителю отдела продаж
5. Руководитель отдела продаж
   1. сохраняет переданные ему списки клиентов, с которыми состоялся разговор в выделенную папку

# Выводы

В ходе работы над проектом была проанализирована информационная система онлайн-школы Фоксфорс и создана модель, которая позволяет с хорошей точностью предсказывать вероятность покупки клиентом платной услуги.

Была разработана методика для тестирования модели, которая, при удовлетворительных результатах, легко трансформируется в алгоритм рабочего процесса.

В данный момент ведется обсуждение с руководством Нетологии о введении модели в продакшн.

## Развитие модели

В настоящий момент в модели не были учтены данные трафика, содержащиеся в базе BigQuery и данные о звонках пользователям, содержащиеся в базе AmoCRM. Одним из важных шагов в развитии модели могло бы быть получение доступа к этим базам и их анализ.

Также в ходе работы над моделью появились идеи об использовании некоторых фич из базы Metabase, например, «Участвовал ли пользователь в олимпиаде Фоксфорда», «Занял ли пользователь какое-либо призовое место на олимпиаде Фоксфорда» или «Положил ли пользователь какие-либо курсы в корзину». Использование этих фич также могло бы дать прирост качества.

# Список источников

1. Joel Grus. «Data Science from Scratch: First Principles with Python» 2015 O’Reilly Media
2. Python documentation. <https://docs.python.org/3/>
3. Documentation of scikit-learn. <http://scikit-learn.org/stable/documentation.html>
4. XGBoost documentation. <http://xgboost.readthedocs.io/en/latest/>