Федеральное государственное автономное образовательное учреждение
высшего образования

«Национальный исследовательский университет

«Высшая школа экономики»

Факультет компьютерных наук
ООП «Прикладная математика и информатика»

Отчет о прохождении учебной практики

Студент: Когтенков Алексей Александрович

Группа: 151

Место прохождения
учебной практики: ООО "Вебгеймс", ул. Ленинская Слобода, 19, Департамент маркетинга

Руководитель:

Ведущий аналитик-исследователь,
Цымбалов Евгений Алексеевич

Москва, 14 июля 2017

Оглавление

Часть 1

Описание задач проекта и реализации

1.1 Постановка задач проекта

1) Ознакомиться с инструментарием для анализа данных;
2) Проанализировать данные о игроках и построить модель для предсказания количества дней в следующие дни.

В качестве более конкретной задачи была взята следующая: по данным об игроке в первый месяц после регистрации предсказать, сколько дней он будет играть в течение следующих двух недель после первого месяца.

Реализация модели была осуществлена на языке Python.

1.2 Описание таблиц с данными

Модель составляется на основе данных трёх таблиц. Каждая из этих таблиц состоит из строк, каждая строка содержит информацию о событиях:

1. Таблица pgr-сообщений:
    - Идентификатор игрока (id);
    - Страна (country);
    - Количество PGR-сообщений (их шлет устройство во время игры) (pgr);
    - Дата события (actdate).
2. Таблица агрегированных данных о деятельности игрока за день:
    - Идентификатор игрока (id);
    - Дата события (actdate);
    - Дата регистрации игрока (regdate);
    - Сколько кластеров (кластеры состоят из квестов) основной сюжетной линии прошел игрок (clusters);
    - Прибыль с игрока (revenue);
    - Количество платежей (transactions);
    - Сколько побочных квестов прошел игрок (quests);
    - Сколько квестов основной сюжетной линии прошел игрок (m_quest);
    - Сколько кристаллов (в определенной нормировке) потратил игрок (crystals);
    - Сколько раз игрок зашел в магазин (store_enters).

3. Таблица сессий:

   - Идентификатор игрока (id);
   - Порядковый номер сессии игрока (session_number);
   - Длина сессии (session_length);
   - Время начала сессии (session_start);
   - Время окончания сессии (session_end);
   - Дата события (actdate).

Особенностью таблиц является наличие небольшого количества несоответствий: например, в первой таблице могут быть данные о том, что игрок что-то делал, а во второй и третьей соответствующих данных может не быть. Однако, такие случаи нечасты и происходят, если игрок почти ничего не делает, поэтому не должны влиять на результат.

Также по каким-то причинам могло получаться, что дата регистрации наступает позже даты события; такие "бракованные" строки я выкидывал.

1.3 Описание основной модели

Структура модели

В качестве основной модели для предсказания я взял двухуровневую модель.

1) Первый уровень – классификатор активности (далее в некоторых случаях – просто классификатор). Игрок считается активным, если в течение двух недель играл в игру больше, чем 1 день. Классификатор возвращает 1, если игрок считается активным, и 0 иначе.

2) Второй уровень – предсказатель количества дней (далее в некоторых случаях – просто предсказатель), которые игрок проведет в игре за две недели. Предсказатель возвращает вещественное число.

Итоговый результат вычисляется как $predict\_active \cdot predict\_days$, где $predict\_active$ - результат классификатора, $predict\_days$ - результат предсказателя дней. Такая двухуровневость позволяет улучшить качество предсказаний за счет классификатора, обнуляющего результат предсказателя для неактивных игроков и таким образом снимающего часть нагрузки с предсказателя дней.

Определение качества модели

Для определения качества данной модели введём следующие способы измерения ошибочности предсказаний:

1) Общая ошибка модели - это среднее отклонение предсказанного количества дней от реального. Здесь под отклонением подразумевается модуль разности кол-ва предсказанных и реально проведенных в игре дней.

2) Ошибка предсказателя дней - это среднее отклонение предсказанного количества дней от реального для активных игроков. Эта ошибка позволяет отследить улучшение качества предсказателя дней и дает понимание о том, насколько хорош предсказатель.

3) Ошибка классификатора активности – это отношение неверно сделанных предсказаний классификатора к общему количеству предсказаний. Эта ошибка позволяет отследить улучшение качества классификатора. Для улучшения качества анализа работы классификатора были также добавлены дополнительные показатели: ошибка 1-го и ошибка 2-го рода. Ошибка 1-го рода – это отношение неверных предсказаний в тех случаях, когда игрок был активен, к общему количеству предсказаний. Ошибка 2-го рода – это отношение неверных предсказаний в тех случаях, когда игрок не был активен, к общему количеству предсказаний. Данный показатель позволяет отследить изменения качества классификатора, а ошибки 1-го и 2-го рода помогают проанализировать эти изменения.

4) Ошибка категории. Как известно, игроки могут играть нерегулярно, даже если им нравится игра. Поэтому если мы предсказываем, что игрок будет играть 10 дней за две недели, а он на самом

деле играет только 8 дней, то наше предсказание можно считать вполне хорошим: погрешность составляет всего лишь 2 дня из 14. Иными словами, если мы ошибаемся часто, но на 1-2 дня, то это лучше, чем если мы ошибаемся реже, но ошибки в этих случаях будут больше. Поэтому в игровой статистике игроков принято разделять на категории [сделать ссылку]. Исходя из этих соображений, была создана формула для ошибки категории. Будем считать, что если отклонение предсказанного количества дней от реального лежит на полуинтервале [0; 2.5), то для этого игрока мы верно предсказали категорию; если в полуинтервале [2.5; 5.5), то мы ошиблись на одну категорию; если в [5.5; 8.5) – то на две; если в [8.5; 11.5) – то на три; если в [11.5; 14] – то на четыре.

Плюсы данной ошибки как показателя:

позволяет оценить не только среднее отклонение, но и соотношение малых и больших отклонений

за счет того, что малые отклонения перестают учитываться, позволяет качественно оценить точность модели

Минусы:

размеры временных интервалов в этом показателе составляют примерно 3 дня. Это означает, что если мы возьмем два предсказателя, и между предсказанными ими значениями будет разница около 6 дней, то для обоих предсказателей ошибка категории может оказаться одной и той же, хотя 6 дней – это почти 50% от двух недель. Однако уменьшить длины полуинтервалов нецелесообразно, поскольку иначе категории будут слишком мелкими.

Часть 2

Обучение, результаты и выводы

2.1 Обучение модели

В качестве первоначальных фичей (свойств) для обучения модели было решено взять данные за 4 последние недели месяца (то есть данные с 3 по 30 дни включительно). Это связано с идеей о том, что обычно в первые несколько дней происходят выбросы активности: игрок только начинает играть в игру, поэтому его активность значительно выше нормы вне зависимости от его дальнейшего отношения к игре. Для каждой недели я ввёл формулы получения новых свойств:

1) Среднее за неделю. Среднее значение выбранной фичи за неделю.

2) Максимум за неделю. Макимальное значение выбранной фичи за неделю.

3) Среднее / максимум.

4) Среднее отклонение от среднего.

Причины, по которым я вводил эти новые фичи:

1) Данные необходимо как-то усреднять. Поскольку жизнь людей в какой-то степени периодична с периодом 1 неделя, то решено было усреднять по каждой неделе.

2) Необходимо получать информацию о выбросах. Например, если человек играет каждый день по 20 минут, то это не то же самое, что человек играет два дня в неделю по часу. Для этого были добавлены фичи максимума и среднего отклонения от среднего. Фича среднее/максимум - это очень хорошая фича, заменяющая при обучении фичу максимума, поскольку принимает значения от 0 до 1.

3) Фича минимума добавлена не была, поскольку у абсолютного большинства игроков это был бы 0.

Получив по вышеприведенным формулам значения новых фичей для показателей из таблиц, я обучил модель, используя только их.

Вес фичей

Регрессоры библиотеки sklearn позволяют просмотреть важность фичей – то есть как сильно каждая из фичей влияет на результат. Выяснилось, что самые важные свойства – это количество игровых дней за 4-ю неделю (имеет максимальный вес, в несколько раз превосходящий вес любой другой фичи), средние количества pgr-сообщений за все недели, а также показатели, связанные длиной сессий и количеством пройденных квестов. Наименее важными свойствами оказались показатели, связанные с платежеспособностью игрока и страной проживания. При этом из трех типов свойств (среднее, среднее/максимум, среднее отклонение) самым бесполезным оказалось среднее отклонение: все такие показатели оказались в нижней половине списка. Фича максимума оказалась бесполезной из-за сильной зависимости от других фичей, поэтому ее добавление при обучении модели ничего не меняет.

Полученные веса свойств вполне легко можно объяснить. Например, платежеспособность (то есть прибыль с игрока, количество транзакций и т.п.) хоть и влияет на его поведение, но не сильно. Логично предположить, что если игрок потратил сколько-то денег на игру, то он будет в нее играть в ближайшее время, однако я случайно обнаружил среди данных случаи, когда этого не происходило.

Среди так называемых "платящих" игроков относительно часты случаи, когда игрок платит, но потом через несколько дней прекращает играть. С другой стороны, если игрок довольно много играл в течение каждой из 4-х недель месяца, то логично предположить, что он будет так же и играть и на 5-ю неделю.

2.2 Эксперименты

В результате экспериментов с типами регрессий в качестве классификатора активности был взят gradient boosting regressor библиотеки sklearn; в качестве предсказателя количества дней – extra-trees regressor: это сочетание регрессий дало наилучший результат.

Были проделаны следующие эксперименты и попытки улучшить качество модели:

1) Ансамбльный классификатор и предсказатель. Данные разбиваются на несколько частей, каждая часть подается для обучения своему собственному регрессору. Результатом является среднее арифметическое результатов регрессоров. Однако добиться никакого качественного улучшения таким способом не удалось.

2) Искусственное решающее дерево. Идея заключается в том, чтобы исходя из логических соображений самому сделать простенькую модель, которая будет немного улучшать качество на основании одной-двух фич. Пример такой модели: если игрок за последнюю неделю первого месяца не играл ни одного дня, то мы будем считать его неактивным в наших предсказаниях. Добавление этой модели к классификатору незначительно уменьшило его ошибку 2-го рода, практически не изменив ошибку 1-го рода.

3) Одноуровневая модель. Изначально была опробована модель, в которой не было классификатора, а был только один предсказатель количества дней. В отличие от двухуровневой модели, здесь предсказатель обучался на всех данных (а не только на активных игроках). Качество этой модели было заметно хуже и сопоставимо с качеством предсказателя дней двухуровневой модели (без применения классификатора).

4) Полные данные как фичи. Был проведен эксперимент: взяты данные по каждому дню первого месяца (то есть по 30 чисел для каждой фичи) для всех фичей, и на этих данных была обучена модель. Несмотря на полную неурезанность данных, результат обучения был сопоставим по качеству с результатом при обучении на гораздо меньшем количестве фич.

5) Страна игрока как свойство. Помимо различных количественных данных, исходные таблицы содержат информацию о стране игрока. Я перевел данные о стране в бинарный вид: каждому игроку поставил в соответствие массив длины n, где n – это количество различных стран; на i-м месте стоит 1, если игрок – из этой страны, и 0 – иначе. Эти n новых фичей я тоже сначала использовал для обучения. Однако выяснилось, что они имеют довольно малый вес: у редко встречающихся стран это были почти нулевый коэффициенты веса, но даже у стран типа США вес был недостаточен (и был примерно на порядок меньше максимального веса фич). Поэтому в дальнейших своих экспериментах я не использовал данные о стране игрока.

6) Imbalanced learning. Я попробовал использовать методы для балансировки обучения (используя соответствующую часть библиотеки sklearn), но эти методы работают слишком долго для такого объема данных.

2.3 Результаты

Для проверки своей модели я использовал две различных таблицы с данными, но все вышенаписанное верно для обеих таблиц.

"Платящие" игроки

Фирмы, как правило, интересуются не количеством играющих игроков, а тем, будут ли игроки, которые потратили деньги на какие-то бонусы в игре, дальше в нее играть (и, возможно, и дальше приносить прибыль). Поэтому первая таблица с данными, которую я использовал, содержала в себе

только сведения о тех игроках, которые в первый месяц после регистрации потратили сколько-то денег. В результате обучения на 60% этих данных (это 16000 игроков), на проверочной выборке (оставшиеся 40%) модель показала следующий результат:

Ошибка классификатора активности: 10.634%

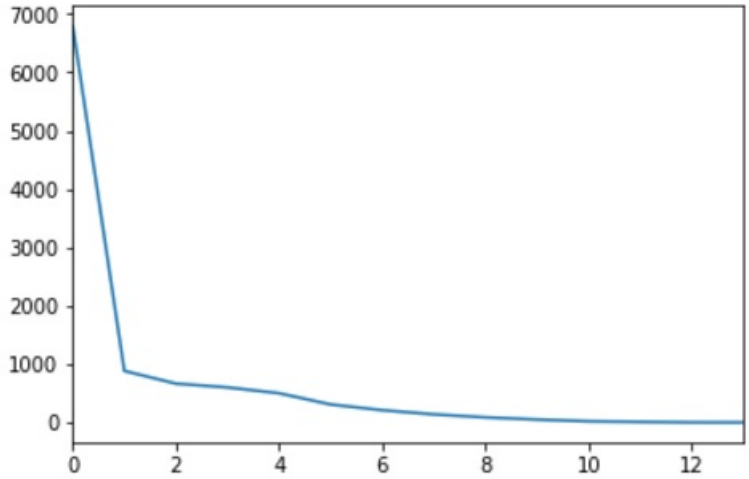
Ошибка классификатора 1-го рода: 6.419%

Ошибка классификатора 2-го рода: 4.215%

Ошибка предсказателя дней: 2.6455 дней

Общая ошибка модели: 1.1446 дней

Ниже приведен график количества предсказаний с данным отклонением в зависимости от значения отклонения (в днях).



Ошибка категории измерялась как для всех игроков, так и только для игроков, которые были активными во время двух недель после первого месяца. Это было сделано для того, чтобы оценить, насколько хорошо работает модель для активных игроков (которые, прежде всего, и интересуют компанию).

Ошибка категории	Все игроки	Активные игроки
0 (от 0 до 3 дней)	8957 (80%)	2324 (57.4%)
1 (от 3 до 6 дней)	1642 (14.7%)	1348 (33%)
2 (от 6 до 9 дней)	510 (4.6%)	337 (8.3%)
3 (от 9 до 12 дней)	101 (0.9%)	65 (1.6%)
4 (от 12 до 14 дней)	12 (0.1%)	7 (0.17%)

Видно, что для активных игроков наша модель дает результат не такой хороший, как для всех игроков в совокупности. Это различие легко объясняется тем, что, предсказав для неактивных игроков их неактивность, мы автоматически получаем нулевое отклонение, тогда как для активных необходимо предсказывать количество дней.

Активные игроки

Однако помимо игроков, которые в первый месяц успели принести сколько-то денег компании, в игру играет гораздо больше людей, которые этого не сделали, и для них тоже хочется уметь предсказывать количество дней. Однако в подавляющем большинстве случаев игрок перестает играть в игру в течение первой-второй недели, если игра ему не понравилась. Поэтому для таких игроков мы заранее может предсказать, что они не будут играть в следующие две недели после первого месяца: момент их отвала уже произошел. Значит, обучать модель на данных таких игроков не имеет смысла по следующим причинам. Во-первых, если игрок зарегистрировался, поиграл, а потом довольно быстро прекратил играть, то достаточно написать "ручной" классификатор, который по этим признакам сразу говорит, что игрок больше не будет играть. Во-вторых, из-за большого количества таких игроков (а их подавляющее количество) наша модель может переобучиться. В-третьих, опять же, из-за большого количества таких игроков наша модель будет казаться лучше, чем есть на самом деле: она будет выдавать большой процент ответов верно, но действительно важные предсказания

(про игроков, для которых непонятно, сколько они будут играть) могут быть неправильными.

На основе вышеприведенной идеи была составлена таблица с данными игроков, которые за последние две недели первого месяца хоть один день играли. Таких игроков получилось примерно в 2.3 раза больше, чем "платящих" игроков. Стоит отметить, что далеко не все "платящие" игроки попали в эту таблицу. На 55% данных этой таблицы была обучена модель, которая на отложенной выборке (оставшиеся 45%) показала следующий результат:

Ошибка классификатора активности: 19.861%

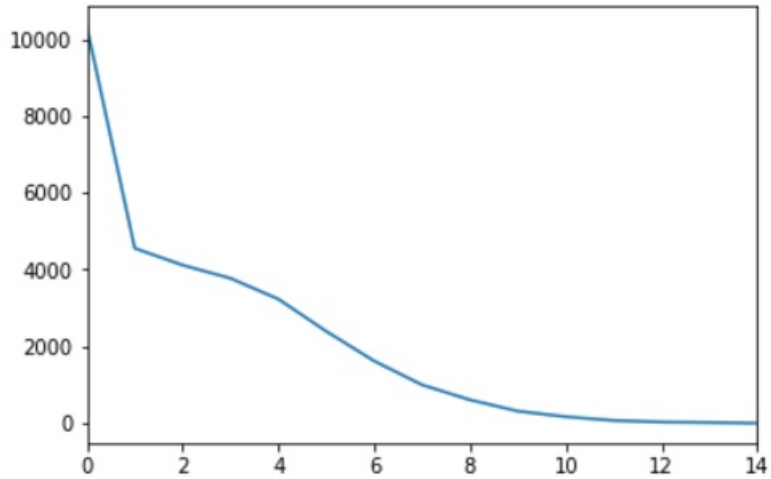
Ошибка классификатора 1-го рода: 9.474%

Ошибка классификатора 2-го рода: 10.388%

Ошибка предсказателя дней: 2.727 дней

Общая ошибка модели: 2.389 дней

Ниже приведен график количества предсказаний с данным отклонением в зависимости от значения отклонения (в днях).



Так же, как и для предыдущей таблицы данных, ошибка категории измерялась не только для всех игроков, но и отдельно для активных игроков.

Ошибка категории	Все игроки	Активные игроки
0 (от 0 до 3 дней)	22774 (59.9%)	13655 (54.9%)
1 (от 3 до 6 дней)	11029 (29%)	8747 (35.1%)
2 (от 6 до 9 дней)	3559 (9.4%)	2108 (8.5%)
3 (от 9 до 12 дней)	605 (1.6%)	324 (1.3%)
4 (от 12 до 14 дней)	66 (0.17%)	53 (0.2%)

Видно, что результат для этой таблицы данных получился гораздо хуже. Причем если точность предсказателя дней осталась примерно на том же уровне, то точность классификатора упала в два раза. Причина, из-за которой это могло произойти, на мой взгляд, всего одна – это уже упомянутая выше проблема игроков, которые после первых дней игры уже отвалились, но простота предсказаний количества дней для них повышает совокупное качество модели.

2.4 Замечания

2.5 Выводы

2.6 Заключение (о практике)

2.7 Источники

1) "Предсказание ухода лояльных игроков в MMO" – http://www.progamedev.ru/2012/08/data-mining-predicting-veterans-churn.html

2) "Choosing the right estimator" (http://scikit-learn.org/stable/tutorial/machine_learning_map) и другие вспомогательные источники информации про обучение моделей, связанные с библиотекой sklearn.