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

Введение

1.1 О компании

Компания "Webgames" занимается разработкой и продвижением игр для мобильных устройств и социальных сетей. Ее продукция является достаточно известной среди определенных пользователей: например, в 2013 году одна из ее игр вошла в топ-20 на Facebook. "Webgames" также занимается сбором данных об игроках и их анализом. На основании полученных результатов анализа делаются соответствующие выводы и изменения в выпускаемой продукции.

Одной из задач такого анализа является предсказание активности игрока в игре.

1.2 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 с использованием библиотек pandas и sklearn.

Часть 2

Описание данных и модели для предсказания

2.1 Описание таблиц с данными

Модель составляется на основе данных из нижеприведенных 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 внутриигровой валюты, – "кристаллов", – (в определенной нормировке) потратил игрок (crystals);
   - Сколько раз игрок зашел в магазин (store_enters).
3. Таблица сессий (то есть данные о временах входа и выхода игрока в игре):
   - 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 был активен в какой-то день, а во второй и третьей соответствующих данных может не быть. Однако такие случаи редки (менее 5% всех данных) и происходят, если игрок почти ничего не делает в игре, поэтому они не влияют на результат.

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

2.2 Описание основной модели

2.2.1 Структура модели

В качестве основной модели для предсказания я взял двухуровневую модель. Напомню, с помощью нее мы хотим предсказывать количество дней, проведенных игроком в игре в течение двух недель через месяц после регистрации.

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. Результат работы классификатора обозначим как $predict\_active$.

2. Второй уровень — предсказатель количества дней (далее в некоторых случаях – просто предсказатель), которые игрок проведет в игре за две недели. Предсказатель возвращает вещественное число, обозначим его как $predict\_days$.

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

2.2.2 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. Данный показатель позволяет отследить улучшение качества классификатора. Для улучшения качества анализа работы классификатора были также добавлены дополнительные показатели:

   - 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.* 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 [1]. 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 3

Обучение, результаты и выводы

3.1 Свойства данных для обучения модели

3.1.1 Выбор свойств данных для обучения

В качестве первоначальных признаков для обучения модели было решено взять данные за 4 последние недели первого игрового месяца (то есть данные с 4-го по 31-й дни после регистрации 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 рассчитал следующие новые признаки:

1. *Среднее за неделю*. Среднее значение выбранного свойства за неделю.
2. *Максимум за неделю*. Максимальное значение выбранного свойства за неделю.
3. *Среднее / максимум* за неделю.
4. *Среднее отклонение от среднего* за неделю.

Причины, по которым я вводил эти новые признаки:

- Данные необходимо как-то усреднять. В предположении, что жизнь людей в какой-то степени периодична с периодом 1 неделя, решено было усреднять по каждой неделе.
- 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 признаки максимума и среднего отклонения от среднего. Фича среднее/максимум – это очень хорошее свойство, заменяющее при обучении фичу максимума, поскольку принимает значения от 0 до 1.
- 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.

3.1.2 Важность (вес) свойств данных

Регрессоры библиотеки sklearn позволяют просмотреть важность свойств – то есть как сильно каждый из признаков влияет на результат предсказания. Выяснилось, что самые важные свойства – это количество игровых дней за 4-ю неделю (имеет максимальный вес, в несколько раз превосходящий

вес любого другого признака), средние количества pgr-сообщений за все недели, а также показатели, связанные с длиной сессий и количеством пройденных квестов. Наименее важными свойствами оказались показатели, связанные с платежеспособностью игрока. При этом из трех типов свойств (среднее, среднее/максимум, среднее отклонение) самым бесполезным оказалось среднее отклонение: все такие показатели оказались в нижней половине списка важности свойств. Признак максимума оказался бесполезным из-за сильной зависимости от других свойств, поэтому его 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, возможно, не сильно. Хотя логично предположить, что если игрок потратил сколько-то денег на игру, то он будет в нее играть в ближайшее время, я случайно обнаружил среди данных случаи, когда этого не происходило. Кроме того, 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.

3.2 Обучение и 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:

- Ансамблевый 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.

- 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.

- 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).

- 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.

- Страна игрока как свойство. Помимо различных количественных данных, исходные таблицы содержат информацию о стране проживания 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 нулевые коэффициенты важности, но даже

у стран типа США вес был недостаточен (и был на порядок меньше максимального веса фич). Поэтому в дальнейших своих экспериментах я не использовал данные о стране игрока.

- Балансировка классов. Абсолютное большинство игроков перестает играть в игру (становятся неактивными) через короткое время после начала, даже если они потратили деньги на покупку внутриигровой валюты. Я попробовал использовать методы балансировки классов активных и неактивных игроков для обучения (используя соответствующую часть библиотеки sklearn), но эти методы работают слишком долго для такого объема данных.

3.3 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 какой-то момент времени (не обязательно даже в первый месяц) потратили деньги на покупку внутриигровой валюты. Классы активных и неактивных игроков в этой таблице по количеству людей имели соотношение примерно 3:1. В результате обучения на 60% этих данных (это 16000 игроков), на проверочной выборке (оставшиеся 40%) модель показала следующий результат:

Ошибка классификатора активности: 10.634%

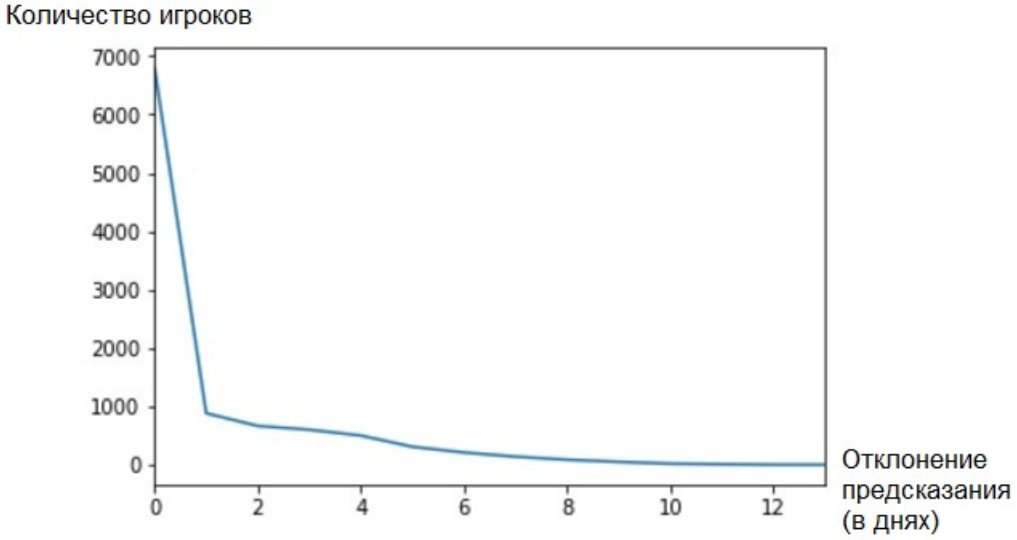
Ошибка классификатора 1-го рода: 6.419%

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 прибыль 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 большинство) и, как следствие, сильной несбалансированности классов активных и неактивных игроков, 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 3 раза больше, чем "платящих" игроков. Стоит отметить, что далеко не все "платящие" игроки попали в эту таблицу. Классы активных и неактивных игроков в этой таблице по количеству людей имели соотношение примерно 1:1. На 40% данных этой таблицы была обучена модель, которая на отложенной выборке (оставшиеся 60%) показала следующий результат:

Ошибка классификатора активности: 19.861%

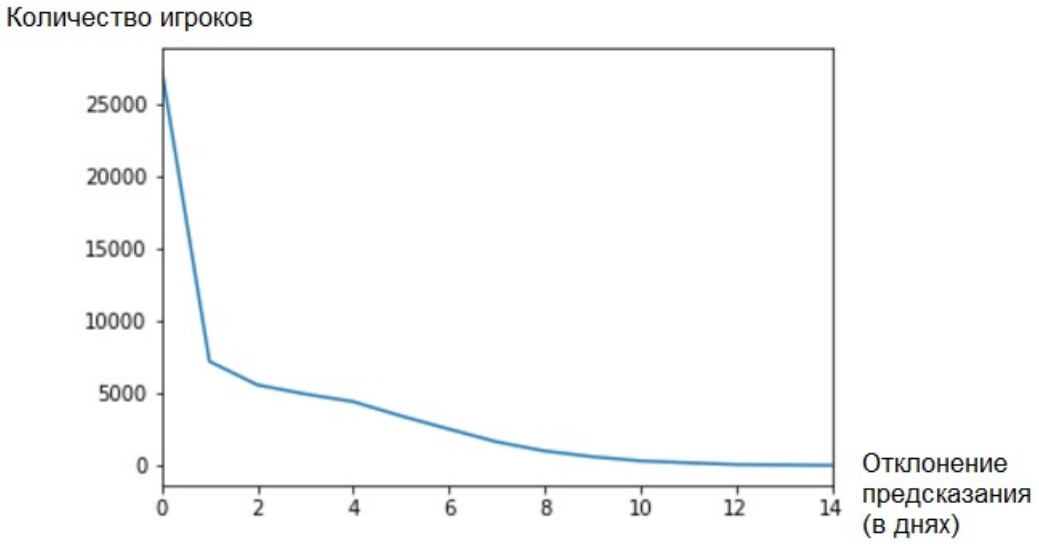
Ошибка классификатора 1-го рода: 9.474%

Ошибка классификатора 2-го рода: 10.388%

Ошибка предсказателя дней: 2.727 дней

Общая ошибка модели: 1.9 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)	45084 (66.9%)	19267 (53.8%)
1 (от 3 до 6 дней)	15278 (22.7%)	12730 (35.5%)
2 (от 6 до 9 дней)	5750 (8.5%)	3291 (9.2%)
3 (от 9 до 12 дней)	1164 (1.7%)	466 (1.3%)
4 (от 12 до 14 дней)	108 (0.16%)	62 (0.17%)

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 большая сбалансированность классов активных и неактивных игроков, в результате которой классификатору сложнее разделять их (например, для первой таблицы можно было сделать классификатор, который всегда возвращает 1, и его ошибка была бы всего 25%).

Интересен тот факт, что качество модели для этой таблицы можно улучшить путем поднятия порога классификатора (то есть увеличения ошибки 1-го рода и уменьшения ошибки 2-го рода): тогда общую ошибку модели можно снизить примерно на 0.2 дня.

Таким образом, хотя на первой таблице модель показала весьма хороший результат, качество ее работы на действительно интересующих нас данных (то есть тех, для которых нельзя сразу же определить, что игрок вообще не будет играть) заметно ниже, хотя даже на них в среднем все еще дает относительно неплохие результаты.

3.4 Заключение (о практике)

Итак, я построил и обучил модель для предсказания количества дней, проведенных игроком в игре в течение двух недель через месяц после регистрации, и таким образом попробовал решить одну из задач игровой аналитики. При этом были опробованы различные подходы к решению проблемы и изучены необходимые библиотеки языка Python. Также я узнал основные задачи игровой аналитики и идеи, связанные с ними. Поскольку игровая индустрия продолжает развиваться, полученные навыки могут мне пригодиться в дальнейшем.

3.5 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.