ADSC
Applied Data Science Center

Индустриальный анализ данных. Статья 1

О цикле

Данный цикл статей основан на семинарских занятиях открытого курса Data Mining in Action по направлению “индустриальный анализ данных”. На семинарах (и, соответственно, в статья) мы будем говорить о том, как правильно формализовать задачи, в расплывчатой форме поставленные заказчиком, понимать, какие данные нужны, и строить решения, применимые в бизнесе.

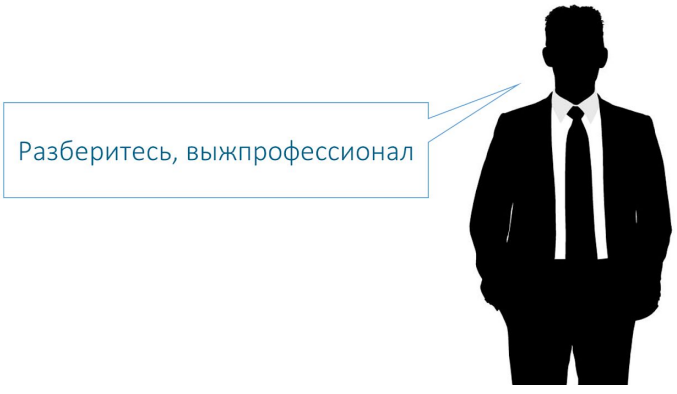
Кейс 1: “очереди в магазине”

Вступление

Представьте, что вам приходит какой-то представительный человек, владелец крупной сети супермаркетов, и обрисовывает вам проблему следующим образом. Ему кажется, что у него проблемы и что он теряет из-за них деньги. Он не знает, сколько. Он не знает, какие проблемы. Но у него чуйка: он теряет деньги*.

Он говорит вам следующее: есть подозрение, что это связано с очередями в магазинах. И добавляет коронную фразу в конце: разберитесь, вы ж профессионал.

Итак, сейчас ваша цель - понять проблему. Задавайте вопросы. Получайте ответы. Узнайте все необходимое, чтобы решить задачу.



* Да, большинство людей действительно не знают, какой эффект они хотят. А еще бывают заказчики, который хотят не тот эффект. И их не переубедить.

Общение с заказчиком

Вопрос: Сколько у нас всего магазинов?
Ответ: Много. Штук семь.
Вопрос: А у конкурентов упали продажи?
Ответ: Мы не знаем, мы за ними не следим.
Вопрос: А какая проблема? Она вообще есть?
Ответ: Мы не знаем, есть ли проблема. Нам кажется, что она есть.
Вопрос: С чего вы взяли, в таком случае, что она есть?
Ответ: Нам кажется, что мы теряем деньги. Мы не уверены. Кажется, у нас бывают очереди. И в этот момент происходит что-то плохое. Но что конкретно - мы не знаем.
Вопрос: А что вы продаете?
Ответ: Продаем еду и сопутствующие товары - зубные щетки, например.
Вопрос: Сколько у вас клиентов?
Ответ: Много. Но мы не знаем, сколько. Мы не измеряем.
Вопрос: А в магазинах есть камеры?
Ответ: В некоторых - да, есть.
Вопрос: А вы можете предоставить материалы с камер?
Ответ: Потенциально можем. Но зачем? Если глазами людей считать, то это плохая идея. А если нейросети обучать на это, то выйдет невероятно дорого и чрезвычайно сложно.
Вопрос: А что у вас вообще есть?
Ответ: У нас есть кассы. Мы можем отгрузить, когда и сколько было куплено. И есть счетчики на входе: мы можем сказать, сколько вошло и сколько вышло.
Вопрос: Есть ли данные кассиров?
Ответ: Есть их расписание.
Вопрос: Как считают клиенты, у вас есть очереди?
Ответ: Клиентов мы не опрашивали.
Вопрос: А продавцы?
Ответ: Продавцы говорят, что им бывает тяжеловато, когда очереди.
Вопрос: Есть местоположения магазинов? Близость к метро?
Ответ: Это неважно. В этой задаче нет зависимости от географии в этой задаче, (иначе это будет слишком сложно).
Вопрос: А в какое время есть очереди? И сколько они длятся по времени?
Ответ: Мы не знаем, мы не посмотрели.

Понимание бизнес-проблемы

Теперь, исходя из полученных сведений, необходимо ответить на следующие вопросы:

- В чем состоит проблема?

Ответ: в том, что, возможно, магазин недополучает деньги в час-пик.

- Как отследить, когда очереди?

Ответ: спросить у кассиров.

- Какие доступные данные у нас вообще есть?

Ответ: Кассовые чеки (в которых можно увидеть список покупок, счет и дату совершения покупки) и количество входов и выходов из магазина.

- Какие есть гипотезы, почему это происходит?

Ответ: мы знаем, сколько людей вошло и знаем, сколько из них купило что-то. То есть, у нас есть отношение: сколько купило / сколько вошло. Необходимо построить график. Чтобы узнать, что происходит с потоком покупателей, когда в магазине образуется очередь, необходимо количество тех людей, которые купили что-то, разделить на количество людей, которые вошли в магазин.

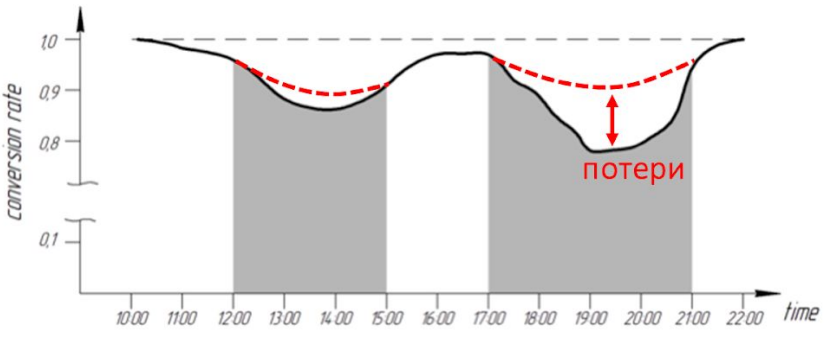
А вот смотреть только на количество людей - это плохая идея, ведь вошедших в час-пик появляется больше, но и людей, которые ничего не покупает, тоже становится больше: магазин недозарабатывает.

- Какую конверсию нужно смотреть, чтобы учитывать непосредственную экономическую выгоду магазина? Как понять, сколько денег приносит вошедший человек? Если мы будем смотреть конверсию, сколько вошло и сколько купило, это не будет столь информативно.

Предположение: Количество покупок?
Ответ: Нет. Может, он купил телегу, а может хлебушек на вечер.
Предположение: Средний чек?
Ответ: уже ближе. А еще более точно - в экономике есть специальный термин - маржа товара. Потому что проблема в том, что она не у всех константная.

- Как нам оценить упущенную выгоду?

Предположение: Посадить много кассиров? Поставить новую кассу?
Ответ: Нет, это плохой вариант. Это значит, что мы предлагаем клиенту потратить деньги за знание, которое может еще и не помочь. Нет, для того, чтобы что-то сделать, сначала нужно дать какое-то обоснование.
Ключевой момент: очереди есть не во всех супермаркетах, более того, очереди у них не в одно и то же время. Соответственно, вы потенциально можете посмотреть, как ведет себя ваша конверсия там, где нет очереди. Это бы дало вам возможность предсказать с помощью модели ML следующее: что будет с конверсией, если бы очереди не было. Мы можем представить это в виде графиков и проинтерполировать.



Показатель конверсии в магазине за будний день

То есть, это означает, что потенциально вы теперь можете осознать, сколько именно недополучаете.

- Вот это расстояние, обозначенное красной стрелкой, - что оно означает?

Ответ: Сколько денег получил магазин с пришедшего человека. То есть, если бы мы поставили дополнительную кассу, то это - **максимальное** количество денег, которые принесут все люди в месяц (с учетом потерь), из чего нужно будет вычесть затраты на кассира и на открытие новой кассы.
Если затраты на нового кассира и на открытие новой кассы больше, чем прибыль, то ставить ее точно нет смысла и нужно искать более дешевые способы улучшить ситуацию.

- Еще одна идея - экспресс-касса. Как можно было бы оценить, сколько денег принесет она?

Для этого нужно посмотреть по чекам, сколько людей совершают маленькие покупки. В идеале мы можем ожидать, что люди с маленькими покупками начнут проходить на экспресс-кассу. Это и есть те деньги, которые касса будет зарабатывать. Соответственно, нужно вычесть эти деньги из других касс и вложить в эту.
Важный момент: нам нужно посчитать, сколько дополнительных денег заработает наша новая касса именно в час-пик. В те часы, когда у вас нет очереди, вы не заработаете новых денег. Условно вы можете посмотреть, сколько в среднем денег приносит касса в час-пик, и считать, что ровно столько денег вы и заработаете с новой кассой. И нужно помнить: то, что мы построим новую кассу, не значит, что очередей не будет. В лучшем случае они станут меньше.

В общем, мы сравниваем расходы с нашей цифрой - максимальным доходом, и думаем, подходит ли нам этот вариант или нет.

Важные детали

Необходимо проговорить некоторые “подводные камни”:

- Нам нужно обучить модель, которая предсказала бы нам ситуацию в магазине, если бы очередей в нем не было. На чем мы можем обучить ее? Есть несколько вариантов: на всех магазинах или на части из них.
    - Если мы будем обучаться на всех магазинах, то тогда наша модель должна будет научиться интерполировать на тот участок, который мы никогда не наблюдали.
    - Если же мы решим брать только часть магазинов, возникает вопрос: какие магазины мы возьмем? Можем взять только те, где очередей не было. Но они могут принципиально отличаться чем-то от тех, где эти очереди есть (к примеру, там у кассиров другое расписание).
- Необходимо учитывать разные детали:
    - например, люди, которые заходят в магазин, и люди, которые остаются стоять очередь, могут приносить магазину разное количество денег: покупатель, набравший полную тележку продуктов, вряд ли покинет очередь, в то время как студент, зашедший вечером купить хлеб, скорее всего, ждать не будет.
    - Кроме этого, можно учитывать наличие альтернатив поблизости. Если в нашем магазине очередь, то человек может пойти в соседний.
    - Можно учитывать, что при очереди товары, которые лежат у кассы, продаются, возможно, чаще.
    - Можно учитывать среднюю работу кассиров в разное время: может быть, кто-то не исполняет свои обязанности усердно?

Все это может повлиять на работу нашей модели.

Применение модели

Да, в этой статье мы не будем рассматривать, какие алгоритмы ML мы используем для обучения нашей модели, так как методы мы еще не прошли.

Но вот, предположим, мы все это учли и обучили нашу модель. Теперь она может взять магазин и сказать, нужно ли что-то в нем сделать (открыть кассу, экспресс-кассу или что мы решили) или нет. Но как оценить ее качество на исторических данных? Никто не даст вам провести эксперимент, пока вы не покажете красивых чисел, которые говорят о чем-то. Необходимо доказать заказчику, что его вложение в новую кассу принесет определенное количество денег.

Мы потратили еще какое-то время и, наконец, оценили работу нашей модели и показали красивые цифры: насколько больше мы станем зарабатывать. Заказчик обрадовался и позволил проводить эксперимент с реальной кассой (экспресс-кассой, новыми продавцами - всем тем, что вы ему предложили).

- Как спроектировать такой эксперимент?

Объектами нашего эксперимента является магазины. И в соответствии с предсказаниями нашей модели мы делаем что-то из заранее определенного списка (установить кассу, установить экспресс-кассу, нанять нового кассира…) в половине из наших магазинов.

- Как выбрать половину? Как разбить выборку на контрольную и тестовую?

Можно случайно. Но в таком случае они могут быть не равноправными.

К примеры, у нас есть сеть из ста магазином. Мы берем некоторым образом пятьдесят случайных магазинов. Как их выбрать? Например, возьмем хэш от адреса и разделим на два. Применяем к одной половине нашу модель и выполняем действие, которое она рекомендует.
Кстати, как понять, удачно ли мы разбили? Вы смотрите исторические показатели и проверяете, что статистической разницы по набору признаков в вашей тестовой группе и контрольной нет. Например, у магазина может быть признак: есть рядом метро или нет. Очевидно, что нужно, чтобы в контрольных группах таких магазинов было более-менее одинаковое количество. Так мы и получаем два репрезентативных множества.

Еще одно замечание: в реальной жизни заказчик озвучивает сумму, которую вы можете потратить. И после этого вы понимаете, в каком соотношении можете разбить свою выборку. Разбивать 70 \ 30 (применяем \ не применяем) кажется плохой идеей: мы еще не знаем наверняка эффектность модели в реальности.

После запуска

Наконец, мы запустили тест.

- На какой эффект нам нужно обращать внимание?

Проверяем, сколько выручки получилось у магазинов разной выборки. При этом смотреть стоит на то, сколько заработали в отношение на какого-то человека: нас все же интересует суммарная маржа, деленная на количество суммарно вошедших людей. Если смотреть на общую сумму, то можно ошибиться: вдруг во время теста по какой-то причине в магазины в целом стало ходить больше людей? Статистика слетит. Именно поэтому нормированные величины более устойчивы ко всяким выбросам, за счет чего можно оценить статистику адекватнее.

Поздравляем, вы только что успешно справились со своей первой задачей и помогли заказчику разобраться с очередями. Удачи со следующим кейсом!

Разбор домашнего задания

Вопросы по BigData

1.Немного шуточный вопрос для разогрева. Есть некий сервис, предлагающий отличить, является ли слово названием покемона или же разработкой в области BigData. Выберите BigData:

- Feebas
- Flink

- Gorebyss
- Arbok
- Azurill
- Seahorse
- Atlassian

Правильные ответы: Flink, Seahorse, Atlassian.

2. Вы реализуете какой-то алгоритм обучения на map reduce. Какие способы разбивания на батчи являются наиболее предпочтительными? (вы хотите получать независимые разбиения с воспроизводимостью результата)

- Взять хэш от объекта, по остатку от деления определить в соответствующий кусочек
- Взять хэш от номера объекта, по остатку от деления определить в соответствующий кусочек
- Использовать random без указания seed
- Использовать random с указанием seed
- Разбивать по значению признака (например по году)

Разбор:

1) Взять хэш от объекта, по остатку от деления определить в соответствующий кусочек

Если у вас есть совпадения значений признаков объектов, то они будут попадать в одну и ту же часть. Это означает, что гипотетически два одинаковых объекта не могут попасть в разные батчи, а две модели не могут увидеть один и тот же объект. Значит, независимости* нет. Вариант не подходит.

2) Взять хэш от номера объекта, по остатку от деления определить в соответствующий кусочек

У каждого объекта свой номер. Поэтому если выберем хэш от номера, то это будут независимые попарно величины. Такой вариант нам подходит.

3) Использовать random без указания seed

Random без указания seed использовать не стоит - нам нужна воспроизводимость результата. Не подходит.

4) Использовать random с указанием seed

Да, все воспроизводимо, так можно.

5) Разбивать по значению признака (например по году)

В таком случае эти части не независимые. Значения одинаковых признаков попадает в один и тот же батч.

Правильный ответ: хэш от номера и random с указанием seed.

***Про независимость:** если два события не могут наступать одновременно (не совместны), то они не независимы.

Задачи на статистику

3. Какова вероятность выбросить 8 и больше орлов из 10 бросков монеты? Ответ округлите до 3 знака

Правильный ответ: 0.055

4. Какова вероятность выбросить 53 и больше орлов из 100 бросков монеты? Ответ округлите до 3 знака

Правильный ответ: 0.309

Обе задачи решаются одинаково:
1) найти что-то вроде scipy.stats
2) в ней найти функцию плотности биномиального распределения,
3) подсчитать значения пары точек.

Задачи на ML

5. Вы обучаете некоторым методом модель a(x), предсказывающую среднее время до поломки детали по ее параметрам. Теперь вам дают исходные параметры детали. Вы, используя свою модель, можете их немного скорректировать (очень слабо изменить), чтобы повысить время до поломки. Какие из нижеперечисленных методов подходят для такого использования в своей классической версии, т.е. без дополнительных улучшений и комбинирований с другими методами? Можете ориентироваться на sklearn.

- линейные модели
- решающие деревья
- метод ближайших соседей
- случайный лес
- градиентный бустинг

Разбор:
Вы хотите использовать вашу модель, чтоб поверх нее производить какую-то оптимизацию (слабо изменить значение признаков). Какие же методы ML позволяют такое сделать корректно?

1) Линейные модели

Вполне: есть линейная зависимость от каждого коэффициента. Вы немного изменяете значение, и ответ тоже изменяется немного. Вариант подходит.

2) Решающие деревья

Решающее дерево - это кусочно-постоянная функция. Если вы немного измените значение, то у вас либо появится гигантский скачок, либо не будет изменения вообще. Никакого адекватного эффекта вы не получите. Вариант не подходит.

3) Метод ближайших соседей

В том же sklearn можно задавать разные веса разным соседям. Таким образом, небольшое изменение приведет к изменению расстоянию до ближайших объектов, и в этом случае целевое значение функции изменится. Это значит, что даже при малейшем изменении значение функции будет меняться. Нам подходит.

4) Случайный лес
5) Градиентный бустинг над решающими деревьями

В чем отличие случайного леса и градиентного бустинга над решающими деревьями от обычных деревьев? Деревьев просто много. Поэтому в этих алгоритмах тоже адекватно изменяться значение функции не будет.

Правильные ответы: линейные модели, метод ближайших соседей.

6. Вы обучали модель предсказывать количество проданного товара в следующем месяце. Однако при применении модели появляются отрицательные предсказания, хотя в обучающей выборке все целевые переменные положительны. Какие методы обучения могут давать такой эффект?

- линейные модели
- решающие деревья
- метод ближайших соседей
- случайный лес
- градиентный бустинг

Разбор:

Итак, прогнозирование спроса: сколько товара будет продано? Отрицательное количество - вряд ли может. Но вот вы обучили модель на обучающей выборке, сделали предсказания на тестовой и наблюдаете, что в тестовой выборке есть отрицательные предсказания, хотя в обучающей таргеты только положительные, даже нулевых нет. Вопрос: у каких методов могут проявляться такие эффекты?

1) Линейные модели

Могут: вы приблизили ответ к какой-то прямой. Признаки объекта комбинируются с определенным коэффициентом, из-за чего и могут появляться отрицательные значения.

2) Решающие деревья

Мы проверяем какие-то значения, затем предсказываем значение из листа. Опускаемся по дереву, приходим в лист и в качестве ответа дает некую константу, которая задается в этом листе. Однако, когда мы задаем константу в листе, мы даже в задаче регрессии будем усреднять значение таргетов. Таким образом, если все таргеты на обучающей выборке были положительны, то ответ в листе отрицательным

быть не может. Соответственно, независимо от того какой объект мы берем, все возможные варианты ответов будут положительными. Этот вариант нам не подходит.

3) Метод ближайших соседей

Гипотетически kNN усредняет значение своих значений. И если у соседей были положительные таргеты, значит усреднение будет тоже положительным. Не подходит.

4) Случайный лес

Если усреднять значения решающих деревьев, у которых только положительные прогнозы, то усреднение тоже будет положительное. Вариант не подходит.

5) Градиентный бустинг над решающими деревьями

В то время как случайный лес усредняет, градиентный бустинг складывает с какими-то коэффициентами, которые подбирает с какими-то сложными предположениями и логикой, о которой вам будет подробно рассказано на лекции, а мы на следующем семинаре (или через один) эту логику разберем. И в таком случае отрицательные значения появляться могут.

Правильные ответы: линейные модели и градиентный бустинг над решающими деревьями.

Практические задачи

7. Что выведет этот код? Ответом на эту задачу является выведенное число, округлённое до 4-го знака, дробная часть отделяется точкой.

```
from sklearn.datasets import load_breast_cancer
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score

X_data, y_data = load_breast_cancer(return_X_y=True)

print(cross_val_score(RandomForestClassifier(criterion='entropy', n_estimators=42, random_state=42), X_data, y_data, cv=3).mean())
```

Правильный ответ: 0.9648

8. Напишите свою функцию определения качества модели по следующей метрике: максимальный precision, при условии, что $recall>0.5$ и $recall>0.5$, и определите наилучшее ее значение, перебирая гиперпараметры по предложенной сетке. Ответом на эту задачу является максимальное значение качества по предложенной метрике, округлённое до 4-го знака, дробная часть отделяется точкой.

```
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import make_scorer
```

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import load_breast_cancer

param_grid = {
    'n_estimators': [10, 20, 30, 40, 50],
    'max_depth': [None, 5, 10, 15, 20],
    'criterion': ['entropy', 'gini']
}

X_data, y_data = load_breast_cancer(return_X_y=True)

estimator = RandomForestClassifier(random_state=42)

print('Accuracy best params and score')
result = GridSearchCV(estimator, param_grid, cv=3, scoring='accuracy').fit(X_data,
y_data)
print('\tParams:', result.best_params_)
print('\tScore:', result.best_score_)


scorer = # TODO

print('Custom loss best params and score')
result = GridSearchCV(estimator, param_grid, cv=3, scoring=scorer).fit(X_data,
y_data)
print('\tParams:', result.best_params_)
print('\tScore:', result.best_score_)
```

Разбор:

У вас была сетка параметров и была кастомная метрика: не просто precision, а precision с дополнительными ограничениями. В случае когда у нас есть какая-то модель, которая говорит не 1 или 0, а какую-то степень уверенности, у нас появляется не одно значение precision и recall, а целая зависимость, и в таком случае вы можете перебирать пороги. То есть, если предсказание промежуточной модели - мера ее уверенности, и она больше 0.1, то объекту будет присвоен, например, класс 1, а в противном случае - класс 0.

Но почему именно 0,1? Этот порог можно перебирать, и для каждого значения порога мы будем получать свое значение precision и свое значение recall. Соответственно, нам надо перебрать все эти пороги, получить множество различных комбинаций precision и recall, выбрать из них те, которые удовлетворяют указанным ограничениям, и взять среди них максимальный precision.

Как это нужно было сделать?

Используя функцию **make_scorer** для того, чтобы реализовать собственную метрику. В sklearn есть функция **precision_recall_curve,** которая принимает истинные метки, ваши предсказания (вот это вот меру уверенности, по которой потом назначаются

пороги) и после этого возвращает вам всевозможные пары комбинаций precision и recall при различных значениях порога.
После этого нужно было взять только те пары комбинаций, где выполнены наши условия: precision < 1.5 recall, а recall > 0.5. Наконец, вы взяли precision, удовлетворяющий таким ограничениям, и среди всех таких значений precision выбрали максимальное.
Кроме этого, можно было совершить следующую ошибку: не указать в функции **make_scorer** параметр **needs_proba = true**. Потом, когда нужно будет оптимизировать модель с вашей метрикой, потребуется, чтобы в качестве y_pred передавались непосредственно мера уверенности. А по умолчанию туда передаются классы: 0 и 1. Тогда вы будете считаеть **precision_recall_curve** просто по 0 и 1, а это не то, что вам нужно.

Правильный ответ: необходимо было написать функцию, сделать из нее **scorer**, указав, что чем больше, тем лучше, и что нужно передавать в нее параметр **needs_proba = true** (то есть, вероятность, а не итоговые классы). А после этого запустить **greed search** и получить ответ, который будет существенно отличаться от значения accuracy и от максимального перебора precision.

9. Какова минимальная сторона квадрата с параллельными осям сторонами, содержащего все точки из x?
Ответом на эту задачу является число, округлённое до 2-го знака, дробная часть отделяется точкой.

```
from sklearn.datasets import load_breast_cancer

data = load_breast_cancer()
X = data.data[:, :2]
```

Разбор:
В исходном варианте задачи нужно было найти квадрат, который полностью покрывал бы все точки.
Стоит заметить, что квадрат обязан касаться самой левой и самой правой точек или самой верхней и самой нижней точек. Таким образом, ответ - это максимум из двух расстояний: между самой верхней и самой нижней - или самой левой и самой правой точек.

При решении слушатели задали вопрос: квадрат можно поворачивать? Мы провели опрос, поняли, что большинство людей такой вариант решать не хочет, и не стали вводить его.

Правильный ответ: 29.57