Мой ответ на все любят пиццу

Большое домашнее задание по курсу «Введение в анализ данных» № 2

16 июня 2019 г.

Общая постановка задания

Мистер Пануччи владеет сетью пиццерий. Для доставки пиццы мистер Пануччи использует довольно большую сеть из курьеров. Он, как мудрый менеджер, понимает, что его текущая система работает неоптимально. В одних районах курьеры регулярно простаивают, в других не хватает рук, и заказы опаздывают. Хочется с помощью машинного обучения решить эту проблему, и каждую неделю перераспределять силы курьеров так, чтобы они не простаивали, и все заказчики получали свою пиццу вовремя.

Вам предлагается подумать о том, как эту проблему можно было бы решить. Для того, чтобы решить эту задачу, вам нужно ответить на следущую серию вопросов:

Как можно измерить, сколько денег сейчас теряет мистер Пануччи из-за неоптимальности курьерской сети?

Тут нужно было написать как минимум два пойнта:

- Если заказ просрочен, то мы теряем стоимость этого заказа.
- Если курьер простаивает, то мы теряем деньги эквивалентные его зарплате.

Обе эти цифры мы можем чётко посчитать по базе, потому что мы платим деньги курьерам и мы устанавливаем цены на продукты. Некоторые хотели считать это через средние зарплаты и средние чеки. Зачем? Это же приблизительное число, а мы можем посчитать точное.

Крутой идей в нескольких работах было попробовать оценить потери в лояльности из-за опозданий и измерить их в деньгах. Тут предлагали вешать на клиентов, которые засветились в базе несколько раз (например, больше 5) ярлык постоянных клиентов, и смотреть перестали ли такие клиенты делать заказы после просрочки. Тут как раз можно было бы использовать для денежных оценок средний чек. Идея очень хорошая, но для её реализации нужна довольно большая база и довольно большое число просрочек.

Какую именно целевую переменную надо научиться прогнозировать? Что за задачу мы решаем? Какие объясняющие переменные взять?

Целевая переменная - спрос (либо суммарная стоимость заказа из района, либо число заказов из района). Решаем задачу регрессии. Тут надо было поднять на поверхность вопрос, а с какой частотностью спрос лучше всего пытаться прогнозировать? Каждый день, каждый час, раз в неделю? Раз в неделю точно нельзя, потому что между выходными и будними днями точно в формировании спроса есть разница. Можно ли обойтись одним прогнозом в день? Тоже явно нет, потому что между утренним, дневным, вечерним и ночным временем точно есть разница в спросе. Логично строить прогнозы несколько раз в сутки, например, на каждый час по каждому району.

Мы прогнозируем спрос по районам. Значит в качестве объясняющих переменных нам надо брать как минимум два вида переменных: временные и пользовательские.

Время:

- Дамми на дни недели
- Дамми на время дня (утро день ночь и тп)
- Дамми на сезон года, месяц
- Дамми на праздники
- Тренд (так как пиццерия постоянно растёт и спрос тоже должен по идее расти)
- Количество заказов за прошлый час (это называется запаздывающей переменной), логично же, что если за прошлый час купили меньше, то в следущий могут купить больше

Пользовательские:

- Население района
- Насколько район престижный (можно прикинуть по средней стоимости квартиры в районе
- Есть ли метро
- Число заведений, где можно поесть (можно выгрузить информацию из яндекс-карты)
- Дамми на район, которая впитает в себя какие-то его отдельные особенности.
- Какая-то информация, которая есть у нас о пользователе, который делает заказ. Например, какие-то данные вроде возраста, которые он вбил в приложение, через которое делает заказ.
- Сюда же можно добавить какую-то историю пользовательских заказов из приложения и посчитать разные переменные по ней.
- Дамми на то, является ли клиент постоянным (сделал ли он больше 10 заказов или 20 заказов), засечку в данных надо подобрать эмпирически по данным.

Как надо использовать прогнозы, чтобы решить проблемы пиццерии? Чтобы ответить на этот пункт, нужно описать схему того, как ваши прогнозы помогают оптимально распределить курьеров по районам.

Тут были очень разные ответы. Чаще всего была какая-нибудь формула в стиле ”один курьер - один заказ”. Спрогнозировали где сколько заказов, раскидали, разрешили курьерам в случае чего нарушать границы соседних районов. В принципе я тут ожидал чего-то подобного.

В паре работ произошёл очень правильны синтез первой домашки и второй. Предлагали следущее:

1. Спрогнозировали на завтра на каждый час число заказов по всем районам.
2. Построили на утро, день, вечер и ночь гистограмму с загруженностью районов.
3. Сегментировали районы по квантилям на районы с высоким, средним, низким спросом.
4. Разбили всех курьеров на смены. Раскидали их по пиццериям в разных районах в зависимости от того, где какая загруженность в соответствии с нашей сегментацией.

И это придумал не я, это придумали вы! И это круто :)

Какие этапы предобработки нужно провести с данными перед тем, как обучать для них модель?

О чём написали:

- Пропуски в данных. Надо будет либо выкинуть наблюдения с пропусками, либо выкинуть слишком разреженные переменные, либо надо будет заполнить все пропуски какими-нибудь нейтральными значениями. В случае непрерывных переменных можно заполнить медианой или средним. В случае катогриальных модой. Либо ничем не заполнять, а просто при One hot encoding поставить отдельное дамми на то, что был сделан пропуск. Про это ещё поговорим.
- Выбросы. Можно очистить данные от них разными способами. Если выброс в числе заказов из-за праздника, на праздник можно поставить дамми-переменную. Потому что праздник - это как толстый друг на шашлыке. Кроме того, можно делать фильтрацию по квантилям, можно попытаться сгладить длинный хвост в данных логарифмированием. Ещё можно поставить везде вместо выбросов NA, а потом заполнить эти NA по какой-нибудь стратегии заполнения пропусков. Но тут надо быть осторожным, так как можно потерять кусок данных.
- Предобработка категориальных переменных. Если категории встречаются часто, делаем их one hot encoding. Если категории встречаются редко, можно объединить их в одну большую категорию ”другое”.
- Зачем-то писали тут про переобучение. Когда я уточнял это в формулировке задания, я хотел увидеть кое-что другое... Переобучение это свойство не предобработки данных, а конкретного алгоритма. Например, случайный лес не переобучается...

О чём мало кто написал:

- У нас данные зависят от времени. Из-за этого есть несколько проблем. Во-первых, надо аккуратно делать разбиение выборки на обучающую и тестовую. Будет не очень правильно проверять качество модели на лете, а учить её на зиме. Разным временам года присуща сезонность, которая будет искажать нашу оценку результативности модели. Некоторые писали, что в таких ситуациях для тестовой выборки нужен как минимум год. Они правы. Можно попробовать отдельно моделировать сезонность и тогда в тест можно брать и более маленькие промежутки.
- Из-за того, что у нас данные зависят от времени, мы можем использовать в качестве объясняющих переменных значения, которые мы наблюдали вчера. Если очень неаккуратно

перемешать данные при разбиение на тренировочную и тестовую выборки, мы будем случайно заглядывать моделью в будущее. Это не очень круто.

- Видел очень крутую идею у ребят, которые поняли метрики регрессии. Они написали, что для нас могут быть важнее часы, когда спрос находится на пике. И в эти моменты для нас важнее всего не налажать с заказами. Они предложили учить модель регрессии, минимизируя MAE, взвешенное на время суток. При этом для каждого времени суток вес предлагалось находить как стоимость среднюю стоимость просрочек. Ну например, вечером заказывают много еды. Средняя просрочка будет очень дорогой. Днём еды заказывают меньше, средняя просрочка будет поменьше. Можно попробовать каждой ошибке прогноза таким образом задать вес и получить на выходе модель, которая попыталась бы учесть такие нюансы. Конечно же веса для обучения можно было бы ставить и по-другому. Способов оценить гадости от просрочки довольно много.

О чём никто не написал:

- Погода. Некоторые брали её в качестве объясняющей переменной. Давайте представим себе ситуацию. Мы взяли и обучили по погоде модель предсказывать спрос на продукты. Как построить прогноз на завтра? Мы же не знаем какая завтра будет погода? Мы можем только использовать прогноз погоды. Но тогда получится, что мы учим модель на реальной погоде, а для прогнозирования спроса используем прогноз погоды. Происходит подмена переменной, и качество модели может из-за этого довольно сильно просесть. Чтобы такой проблемы не было, нужно сразу же учить модель не на настоящей погоде, а на прогнозах погоды.

Как измерить пользу от внедрения модели?

Тут важно понимать, что метрики регрессии (RMSE, MAE и тп) это не пользовательские метрики, отвечающие за качество бизнеса. Это технические метрики, которые говорят нам насколько хорошие прогнозы строит модель. Нельзя смотреть на них, если мы хотим понять насколько хорошо работает бизнес. На них можно смотреть только для того, чтобы понять насколько хорошо строит прогнозы наша модель. Если кто-то предлагал смотреть насколько упало RMSE, он сразу же получал по шее.

Какие хорошие пользовательские метрики люди придумали:

- Выросла прибыль пиццерии (то есть выручка минус издержки)
- Упало количество жалоб и неоплаченных заказов
- Выросло среднее число удачно выполненных заказов
- Упало время простоя курьеров
- Выросла лояльность клиентов (например, число тех, кто за последний месяц сделал больше 10 заказов)
- Можно встроить в приложение пиццерии опросник, где люди после заказа будут ставить обслуживанию какое-то количество звёздочек и отвечать на вопрос порекомендовали бы они делать заказы тут друзьям. Если средняя оценка работы выросла, это хорошо.
- Считаем среднее LTV (lifetime value). Если оно растёт, хорошо.

Пользовательских метрик можно придумать уйму. На какие обращать внимание в первую очередь? Зависит от того, какая проблема (простой или просрочка) для вас важнее.

Другой момент заключается в том, что нельзя просто взять и посмотреть, выросло ли какое-то из чисел. Этого мало. Такой рост может быть случайным. Надо проводить АБ-тест. С ним будет много сложностей. Во-первых. Представим себе, что мы выкатываем новую систему в продакшн летом. А все метрики берут и растут просто из-за того, что наступило лето, более благоприятный сезон для поедания пиццы. А мы берём и списываем этот рост метрик на нашу крутую новую модель. То есть мы обманываем сами себя.

Плохо получается. Нужно чётко понимать какие метрики к такой проблеме чувствительны, и надо их как-то корректировать на сезонность. Можно попробовать сделать это статистическими методами. Либо надо обустраивать АБ-тест так, чтобы сезонность своего вклада не вносила.

Первый вариант - держать АБ-тест год. Но это не очень весёлое занятие. Другой вариант: надо как-то грамотно поделить районы на две части так, чтобы в каждой из выборок оказались сопоставимые по характеристикам районы. После надо на одну часть раскатить систему, а на вторую нет. Но далеко не ϧакт, что такой тест окажется чистым. Собирать адекватный дизайн офлайн-теста, это отдельная и довольно сложная проблема. Про неё тут тоже стоило упомянуть.