Семинар 5: Регрессия

> «Нарисуй линию вдоль моих точек. Да, это искусственный интеллект.»
>
> Герман Греф

В этом семинаре мы впервые столкнёмся с настоящим машинным обучением и попробуем понять что стоит за его магией. В ручной части семинара мы пойдём по следующему плану:

1. разберёмся чем классификация отличается от регрессии, сформулируем задачу регрессии и поймём её специфику;
2. поймём с помощью каких метрик можно оценить качество прогноза в случае регрессии и попробуем разобраться какой смысл стоит за этими метриками;
3. разберёмся как выглядит простейшая линейная модель регрессии;
4. на пальцах прикинем как она обучается.

Упражнение 1 (ставим задачу)

Представьте себе, что у вас есть паблик с мемами. **Вы — Хозяин мемов.** Как и любой другой Хозяин мемов, вы любите лайки под мемами. Возникает желание привлечь в паблик целевую аудиторию, которая будет ставить под мемы лайки. Для этого вы хотите запустить рекламную кампанию паблика. Ясное дело, что рекламу хочется показывать не всем подряд, а только подходящим людям.

У вас есть данные по профилям всех тех людей, которые уже ставили в паблике лайки. По этим данным вам хочется построить модель, которая могла бы предсказать подходит ли конкретный человек для вашей рекламной компании (поставил бы ли он в паблик лайк, если бы был на него подписан).

а) Сформулируйте задачу машинного обучения. Какой должна быть целевая переменная, чтобы перед вами была задача классификации. Какой должна быть целевая переменная, чтобы это была задача регрессии?

б) Какие факторы из профилей вы бы использовали, чтобы спрогнозировать подходит ли человек для рекламной кампании?

в) Приведите ещё парочку примеров задачи классификации и задачи регрессии.

Решение:

Если мы будем пытаться спрогнозировать факт лайка (пользователь поставил хотя бы один лайк в паблик), то мы будем решать задачу классификации, так как мы стараемся предсказать бинарную переменную (либо поставил, 1, либо нет, 0). Если мы будем пытаться спрогнозировать непрерывную переменную: количество лайков, которое пользователь поставил в паблике, то мы будем решать задачу регрессии.

В качестве факторов для прогноза можно использовать абсолютно любую информацию из профилей: пол, возраст, есть ли аватар, как часто человек что-то репостит, на какие другие похожие паблики он подписан и т.п. Правда не факт, что все эти переменные окажутся полезными.

Классификация: предсказание оттока клиентов, вернёт ли человек кредит, болен ли человек, содержит ли письмо спам, мошенническая ли транзакция, сделает ли человек клик, поставит ли лайк и т.д.

Регрессия: предсказание цен, спроса, выручки, валютного курса, ВВП страны, инфляции, качества вина, уровня преступности (число преступлений на душу населения) и т.д.

Упражнение 2 (качество прогноза)

Добрыня, Алёша и Илья смотрят мемы и ставят на них лайки. Мы пытаемся предсказать сколько лайков они оставят под мемами на основе поведения их однокурсников. Для этого мы оценили регрессию. Ну и она нам напредсказывала, что парни поставят 4, 20 и 110 лайков. В реальности они поставили 5, 10 и 100 лайков. Возникает вопрос: насколько сильно наша модель ошиблась в прогнозировании.

Что такое MAE, MSE, RMSE и MAPE? Посчитайте для модели все четыре метрики качества.

Решение:

Попробуем посчитать основные метрики, которые встречаются для регрессии.

- **MAE (mean absolute error), средняя абсолютная ошибка**

  Первой очевидной метрикой качества будет просто взять и просуммировать все ошибки модели. Так, в случае Добрыни ошибка оказалась $|5-4| = |1| = 1$. Модуль от ошибки берётся, потому что можно ошибаться в разные стороны. Например, если бы не было модуля, для Алёши ошибка составила бы $10-20 = -10$. Потом, чтобы посчитать среднюю ошибку, нам надо было бы сложить два числа и мы получили бы 9. Ошибка в 9 лайков. А это неправда, потому что мы ошиблись в 11 лайков (в одном случае предсказали 1 лишний, а в другом потеряли 9). Поэтому берётся модуль.

  Средняя абсолютная ошибка для парней составит:

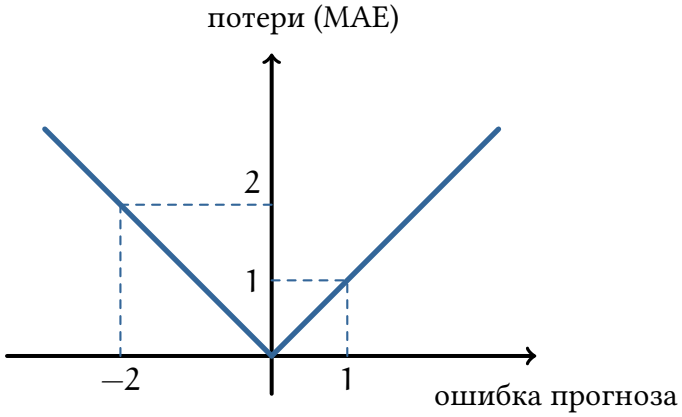
  $$\frac{1}{3} \cdot (|5-4| + |10-20| + |100-110|) = 7$$

  В среднем мы ошибаемся на 7 лайков. Формула для поиска средней абсолютной ошибки в общем виде выглядит вот так:

  $$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|.$$

  Можно нарисовать MAE на картинке. По оси x отложим ошибку прогноза. В случае Добрыни это $5-4=1$. По оси y будем откладывать то, насколько сильный штраф мы накладываем за такую ошибку. В случае MAE штраф за ошибку в 1 равен 1. То есть мы получа-

ем прямую под углом в 45 градусов. В отрицательную сторону ошибка также штрафуется один к одному. График выглядит, как галочка.



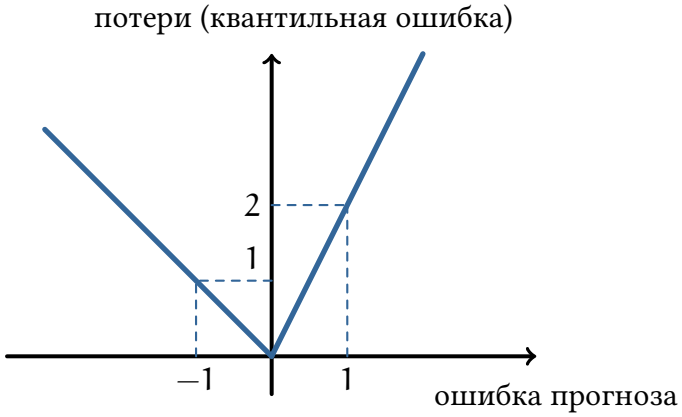
- **квантильная ошибка**

Один из минусов MAE в том, что мы одинаково нелюбим перепрогноз и недопрогноз. В реальности цена этих двух ошибок может быть разной. Мы можем это учесть. Представим себе ситуацию, что Хозяин мемов очень сильно обижается, если мы прогнозируем ему больше лайков, чем получается в реальности. Если мы прогнозируем меньше, чем в реальности, он также обижается, но чуть-чуть. В общем, когда мы делаем перепрогноз, он обижается в 2 раза сильнее. Тогда ошибку можно посчитать по формуле:

$$\frac{1}{3} \cdot (|5-4| + 2 \cdot |10-20| + 2 \cdot |100-110|) = 13.6$$

То есть мы умножили перепрогнозы на два. На самом деле, в реальности коэффициенты могут быть и другими. Чаще всего они берутся из всяких денежных соображений. В случае, если бы мы прогнозировали не лайки, а продажи в магазине, недопрогноз спроса мог бы для нас быть серьёзнее из-за потери кучи денег в виде лояльных клиентов. Насколько он страшнее мы могли бы попытаться померить в деньгах. Такая неравномерная ошибка обычно называется **квантильной ошибкой.**

Если мы решим нарисовать такую ошибку на картинке, наша галочка чуть-чуть покорёжится.



Увидели? Теперь, если мы ошибаемся на единицу направо, то есть происходит перепрогноз, мы несём потери размером 2. Если мы ошибаемся на единицу налево, то есть про-

исходит недорогноз, мы несём потери в 1. Получается, что любой сдвиг вправо в плане ошибки для нас в два раза хуже, чем влево. Чем больше коэффициент перед ошибкой, тем резче дисбаланс в ошибках для нас. В общем виде квантильную ошибку можно записать вот так:

$$QE = \frac{1}{n}\sum_{i=1}^{n} \rho(y_i - \hat{y}_i),$$

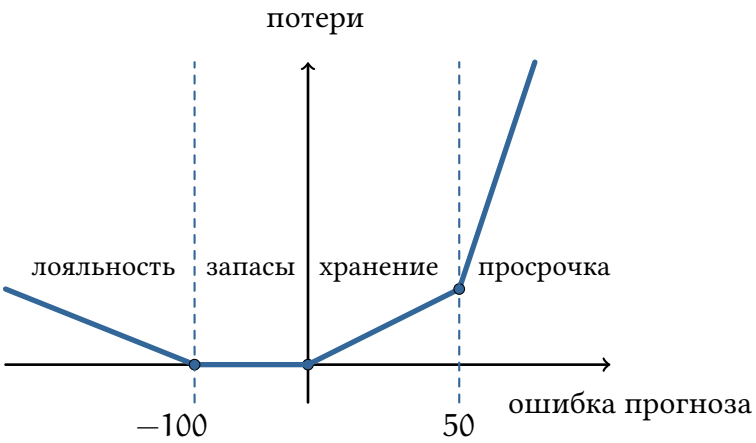
где функция $\rho(y_i - \hat{y}_i)$ это

$$\rho(y_i - \hat{y}_i) = \begin{cases} \alpha_1 \cdot (y_i - \hat{y}_i), \text{ если } y_i \geqslant \hat{y}_i \\ \alpha_2 \cdot (\hat{y}_i - y_i), \text{ если } y_i < \hat{y}_i \end{cases}.$$

Буквы α_1 и α_2 означают какие-то штрафы. В нашем случае это 1 и 2.

- **Кусочно-линейная функция потерь**

  Можно немного модернизировать квантильную ошибку и превратить её в кусочно-линейную функцию потерь. Такая функция неплохо подойдёт для задачи прогнозирования продаж.



Если мы спрогнозировали слишком маленький спрос, мы произведём мало товара и его не хватит всем покупателям. Идём по оси x налево. Поначалу наш косяк смогут закрыть резервы со склада. И мы не будем терпеть никаких убытков. Но рано или поздно они подойдут к концу. Предположим, что на складе хранится 100 единиц товара. Тогда, если мы пробьём своей ошибкой его объём, получится не очень хорошая ситуация.

Приходят к нам потребители и говорят: мы хотим телевизор. А у нас нет. На складе пусто. Ну потребитель и говорит нам, что пошёл в другой магазин, раз у нас пусто. Мы теряем лояльность потребителя и, скорее всего, в следующий раз он пойдёт сразу в другой магазин. Это страшно. Поэтому начиная с ошибки в −100, у нас появляются потери. Насколько крутыми они будут, зависит от того, как быстро тает лояльность пользователя. Это нужно как-то оценивать по данным. И это отдельная задача.

Если мы спрогнозировали слишком большой спрос, мы наклепаем лишнего товара. Движемся по оси x направо. Если мы произвели не особо много лишнего, можно положить весь товар на склад до лучших времён. Мы будем нести издержки на хранение. У кривой

потерь будет один угол. Если товара было произведено слишком много, часть испортится. Мы должны будем выкидывать товар на помойку и угол у потерь будет более крутым. Например, на картинке, мы нарисовали потери так, что если перепрогноз был больше, чем на 50 единиц, то будет просрочка. Откуда взять это число? Опять же надо посмотреть на реальные данные и понять начиная с какой отметки товар точно будет портиться.

Для кусочно-линейной функции потерь можно придумать довольно большое количество разных ситуаций и углов в зависимости от бизнесовой составляющей вашей задачки.

- **MSE (mean squared error), средняя квадратичная ошибка.**

Все три метрики, о которых мы уже поговорили, линейно накидывают потери за ошибку. А что если мы хотим штрафовать большие потери ещё сильнее, может быть, даже нелинейно? Тут нам на помощь приходит штука под названием **средняя квадратичная ошибка.** Считается по формуле

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2.$$

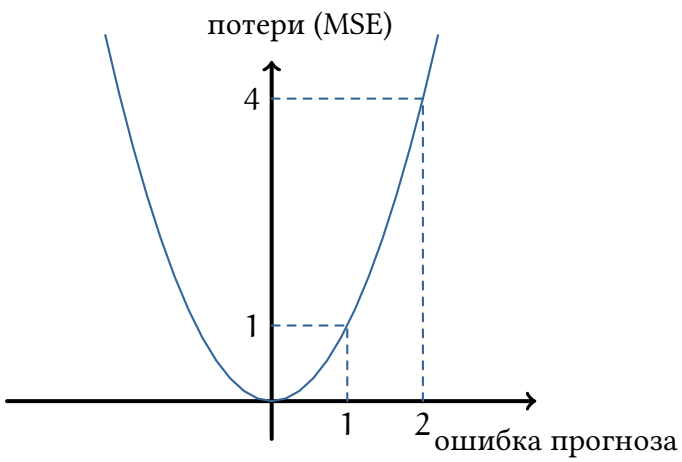
Для нашей ситуации она составит

$$\text{MSE} = \frac{1}{3}\cdot((5-4)^2 + (10-20)^2 + (100-110)^2) = 67$$

Посчитали, на формулу посмотрели. Теперь давайте разбираться какой в этом смысл. Смысл в том, чтобы штрафовать за большие ошибки сильнее, чем за маленькие. Если мы ошиблись на 5 лайков, то в потери войдёт 25. Если мы ошиблись на 10 лайков, то в потери войдёт 100. Чем выше ошибка, тем сильнее потери.

Вы можете сказать мне: «А зачем? Квантильная ошибка делает то же самое!» Не совсем. В примере, на который мы смотрели выше, мы каждый раз умножали квантильную ошибку на одно и то же число, 2. То есть за ошибку в 5 лайков мы бы понесли потери размером 10. За ошибку в 10 лайков мы бы понесли потери размером 20. То есть в два раза больше. В случае MSE потери получились 25 и 100. То есть в 4 раза больше.

В квантильной ошибке пропорция между потерями всегда одинаковая, а в квадратичной ошибке она постоянно увеличивается. Давайте нарисуем на картинке.



Ошиблись на один лайк? Потери 1. На два лайка? Потери 4. На три лайка? Потери 9. Потери каждый раз всё больше. Такова природа этой ошибки. Если вы ещё не забыли, в дисперсии была такая же логика.

У MSE есть проблемы с выбросами. Она очень резко реагирует на них и штрафует за их наличие очень сильно. Если вы хотите использовать MSE и знаете, что у вас в данных выбросы, от них нужно избавиться заранее.

Вопрос: а чувствительна ли к выбросам MAE?

Ответ: нет, потому что сумма по модулю и не так жёстко штрафует за увеличение отклонения.

- **RMSE (root mean squared error)**

  Когда мы говорили про MAE, мы выяснили, что потери в её случае измеряются в лайках (ну или в телевизорах). Для MSE мы каждое слагаемое возводим в квадрат, и итоговая сумма измеряется в квадратных лайках. Или в квадратных телевизорах. Ну или на худой конец в квадратных попугаях. Можно извлечь из MSE квадратный корень, чтобы избавиться от квадрата **(снова как в случае дисперсии и среднего квадратичного отклонения).** Тогда получится новая ошибка, RMSE. Посчитаем её для нашей ситуации:

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{67} \approx 8.19$$

  Из-за того, что более большие ошибки для нас страшнее, RMSE обычно получается больше, чем MAE.

- **MAPE (mean absolute percentage error)**

  Последний герой нашей задачки про метрики. Часто для нас принципиальным является не то, на сколько лайков мы ошиблись, а то, на сколько процентов мы ошиблись. Метрика, которая отлавливает процентную ошибку, называется MAPE (mean absolute percentage error), средняя абсолютная процентная ошибка.

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i}$$

  Если вы предсказали 1, а в реальности было 10 — это не то же самое, что вы предсказали 1000, а в реальности было 1009: в одном случае вы ошиблись в 10 раз, а в другом — совсем чуть-чуть. С точки зрения MAE или MSE, это две совершенно одинаковые ошибки: $|1009 - 1000| = |10 - 1| = 9$. А если вас интересует относительная ошибка, т.е. на сколько процентов вы ошибаетесь, то следует использовать MAPE. В первой ситуации мы ошиблись на $100 \cdot \frac{|1-10|}{10} = 90$ процентов от реального результата. Во второй ситуации на $100 \cdot \frac{|1000-1010|}{1009} \approx 1$ процент от реального результата.

  Давайте посчитаем метрику для нашей выборки.

$$\text{MAPE} = 100 \cdot \frac{1}{3} \cdot \left(\frac{|5-4|}{5} + \frac{|10-20|}{10} + \frac{|100-110|}{100}\right) = 43\%$$

  В среднем при каждом прогнозе мы ошибаемся на 43% от реального результата.

Часто MAPE используют в финансах, поскольку там важен процент, который мы получаем в качестве дохода, а не абсолютное значение.

Этого набора метрик нам для начала хватит. На практике часто используются разные другие метрики. Про них можно подробнее почитать, например? в книге Дьяконова (тут гиперссылка, тыкнете её и перейдёте в книгу). Правда, там может встретиться довольно заковыристая для вас математика.

Упражнение 3 (как выглядит модель)

Предположим, Олег хочет купить автомобиль и считает, сколько денег ему нужно для этого накопить[1]. Он пересмотрел десяток объявлений в интернете и увидел, что новые автомобили стоят около 20000, годовалые — примерно 19000, двухлетние — 18000 и так далее.

В уме Олег-аналитик выводит формулу: адекватная цена автомобиля начинается от 20000 и падает на 1000 каждый год, пока не упрётся в 10000. Олег сделал то, что в машинном обучении называют регрессией — предсказал цену по известным данным. Давайте попробуем повторить подвиг Олега.

а) Как выглядит формула в случае Олега?

б) За сколько продать старый айфон? Придумайте формулу для предсказания. Проинтерпретируйте каждый коэффициент в ней.

в) Сколько одежды брать с собой в путешествие? Придумайте формулу для предсказания. Проинтерпретируйте каждый коэффициент в ней.

г) Сколько шашлыка брать на дачу? Как выглядит формула?

д) Сколько брать шашлыка, если есть друг-вегетарианец? Как можно назвать этого друга в терминах машинного обучения? Испортит ли вегетарианец формулу?

Было бы удобно иметь формулу под каждую проблему на свете. Но взять те же цены на автомобили: кроме пробега есть десятки комплектаций, разное техническое состояние, сезонность спроса и ещё столько неочевидных факторов, которые Олег, даже при всём желании, не учёл бы в голове. Люди тупы и ленивы — надо заставить вкалывать роботов.

Решение:

а) Формула Олега: $y_i = 20000 - 1000 \cdot x_i$, где y_i— цена машины, x_i — её возраст. Если бы мы собрали данные о машинах и загнали их в компьютер, нам нужно было бы оценить модель

$$y_i = \beta_0 + \beta_1 \cdot x_i.$$

[1]Сделано по мотивам статьи “Машинное обучение для людей”, прочтите её: https://vas3k.ru/blog/machine_learning/

Коэффициент β_0 отражает базовую стоимость машины, а β_1 то, насколько она дешевеет с каждым годом. Вообще говоря, дополнительно следует добавить ограничение снизу (машина вряд ли когда-то станет стоить меньше 10000):

$$y_i = max(10000; 20000 - 1000 \cdot x_i)$$

Но это условие, в целом, можно учесть и при последующей обработке предсказаний.

б) Я не знаю, к чему мы пришли при обсуждении на семинаре, но, скорее всего, к чему-то похожему на случай Олега.

в) Это зависит как минимум от двух вещей: длительности поездки и времени года. Зимой обычно нужно больше вещей. Формула для обучения может выглядеть, например, вот так:

$$y_i = \beta_0 + \beta_1 \cdot x_i + \beta_2 \cdot winter_i,$$

где x_i — срок поездки, а $winter_i$ принимает значение 1, если поездка совершалась зимой. Тогда коэффициент β_0 будет говорить, какое базовое количество одежды нужно в любой поездке (например, тот минимум, который будет надет на нас). Коэффициент β_1 — сколько дополнительной одежды надо взять, если срок путешествия увеличивается на один день. Коэффициент β_2 будет говорить сколько одежды надо взять с собой дополнительно, если вы решили путешествовать в холодное время года. То есть летом, по-любому, надо взять с собой β_0 одежды, а зимой — $\beta_0 + \beta_2$ одежды.

Можно ещё сильнее усложнить формулу и получить

$$y_i = \beta_0 + \beta_1 \cdot x_i + \beta_2 \cdot winter_i + \beta_3 \cdot winter_i \cdot x_i,$$

Коэффициент β_3 в таком уравнении будет говорить, на сколько единиц одежды надо взять больше на каждый дополнительный день, если поездка пришлась на зиму.

в) Это зависит от числа людей и числа дней, на которое мы едем на дачу. Наверное, логично было бы брать полкило на человека в день, то есть $y_i = 0.5 \cdot x_i \cdot z_i$, где x_i — число человек, z_i — число дней.

Да, да. Это тоже модель! И она нелинейная. Никто не обещал, что будет легко. Можно при желании превратить её в линейную (линеаризовать). Обычно это делается с помощью логарифмирования:

$$\ln y_i = \ln 0.5 + \ln x_i + \ln z_i.$$

В данном случае мы подобрали все коэффициенты из головы, задействовав свой природный оцениватель. Другой путь: собрать данные о поездках на дачу и заставить компьютер оценить модель:

$$\ln y_i = \beta_0 + \beta_1 \cdot \ln x_i + \beta_2 \cdot \ln z_i.$$

Такие модели, записанные в логарифмах интерпретируются чуть сложнее линейных. Они интерпретируются в процентах. Коэффициент β_1 отражает то, на сколько процентов будет расти количество необходимого шашлыка, при росте числа людей на 1%. Коэффициент β_2 будет говорить, на сколько процентов будет расти количество необходимого шашлыка, при увеличении числа дней на 1%. Немного подробнее про это будет в задачках ниже.

Иногда мы будем брать логарифмы от факторов при оценивании моделей. Это будет позволять нам бороться с выбросами и получать на выходе более адекватную модель. Про это читайте в Ещё задачах!

г) Если есть друг-вегетарианец, он не ест мясо. В терминах наших данных это статистический выброс. Если использовать модель выше для этого друга, то часть мяса рискует испортиться за ненадобностью. Если же заново оценивать модель по выборке, включающей вегетарианца, то она подстроится под него и будет выдавать плохие прогнозы для обычных людей: скорее всего, среднее количество мяса на человека в нашей формуле уменьшится.

Можно модернизировать нашу модель и ввести на этого друга дамми-переменную, которая будет принимать значение 1, если рассматриваемое наблюдение — он, и 0, если кто-то другой. Модель тогда будет выглядеть:

$$\ln y_i = \beta_0 + \beta_1 \cdot \ln x_i + \beta_2 \cdot \ln z_i + \beta_3 \cdot veg_i.$$

Тогда после оценивания модели коэффициент β_3 будет отражать то, на сколько шашлыка надо взять меньше с учётом друга-вегетарианца. То есть коэффициент β_3, скорее всего, будет отрицательный.

Очень важно понимать, что такая **интерпретация коэффициентов верна только в тех случаях, когда мы изменяем какую-то одну переменную при фиксированных других (при прочих равных). Более того, на реальных данных все эти изменения верны в среднем,** а не для каждого конкретного случая. То есть, если мы подогнали под наши данные модель

$$y_i = \beta_1 \cdot x_i + \beta_2 \cdot z_i$$

реальное значение y в среднем (не всегда) увеличится на β_1, при увеличении x_i на 1, если при этом z_i останется неизменной (при прочих равных).

Упражнение 4 (как обучаются модели)

Давайте попробуем совсем-совсем на пальцах почувствовать, как модели обучаются. Пусть у Хозяина мемов есть две переменные: x — возраст подписчика и y — число лайков, которое он оставил. Хозяин мемов хочет оценить регрессию $y = \beta \cdot x$, то есть он хочет попытаться предсказать число лайков по возрасту подписчика. Хозяин собрал два наблюдения для оценивания модели: $x_1 = 15, y_1 = 10$ и $x_2 = 22, y_2 = 2$.

Теперь хозяину надо подобрать коэффициент β так, чтобы ошибка прогноза, измеряемая с

помощью MSE оказалась поменьше.

1. Пусть $\beta = 1$. Какие значения нам спрогнозирует модель? Какая у неё будет ошибка?
2. Пусть $\beta = 0.5$. Найдите прогнозы и ошибку модели.
3. Какое значение для β нам больше подходит? Как можно найти оптимальное β?

Решение:

При $\beta = 1$ получаем прогнозы $\hat{y}_1 = 1 \cdot 15 = 15$ и $\hat{y}_2 = 1 \cdot 22 = 22$. Находим ошибку: $\text{MSE} = (15 - 10)^2 + (22 - 2)^2 = 25 + 400 = 425$.

При $\beta = 0.5$ получаем прогнозы $\hat{y}_1 = 7.5$ и $\hat{y}_2 = 11$. Ошибка составит $\text{MSE} = (10 - 7.5)^2 + (2 - 11)^2 = 6.25 + 81 = 87.25$.

В случае $\beta = 0.5$ ошибка ниже. Если перебрать кучу разных β, можно найти самое классное! Именно этим и занимается компьютер, пока мы не видим. Конечно же он не перебирает в лоб все возможные значения[2]. Он делает перебор по-умному. Обычно находят производную функции ошибки по параметру β и по ней понимают куда надо сдвинуться и какое значение β надо "проверить на оптимальность" следующим. **Такой умный перебор называется градиентным спуском.** Но о нём мы поговорим подробнее как-нибудь в другой раз.

Упражнение 5 (одинокий дуб)

Для того чтобы решать задачу регрессии и прогнозировать что-нибудь, можно пытаться искать коэффициенты в уравнениях, которые мы выписывали выше. Это один из вариантов модели. Он называется **линейной регрессией.** Линейной, потому что мы пытаемся провести через облако точек линию. Можно пробовать оценивать и какие-то другие, более сложные, нелинейные модели. Например, можно построить **регрессионное дерево.** Было бы нечестно бросать вас не обучив ручками ни одной модели. Давайте обучим!

Миша работает в маленькой кофейне. Харио Малабар Монсун — фирменный напиток этой кофейни. Мише интересно узнать, как именно ведёт себя количество заказов напитка y_i в зависимости от температуры за окном t_i. Четыре дня Миша записывал свои наблюдения:

t_i	y_i
21	1
19	2
12	8
8	8

Сегодня он решил обучить регрессионное дерево. В качестве функции потерь он использует

$$\text{MSE} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2.$$

[2] он же не бесполезный кусок мяса

а) Обучите регрессионное дерево.

б) Какой прогноз на сегодня сделает дерево Миши, если за окном 13 градусов?

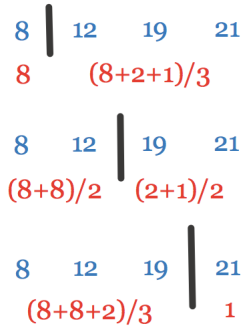
в) Можно ли для обучения дерева использовать MAE?

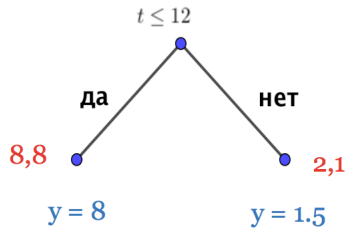
Решение:

Давайте посадим дерево! Будем ли мы на следующем строить дом и рожать ребёнка — большой вопрос. Мы должны по переменной t спрогнозировать переменную y.

Для этого нужно обучить дерево. Учить мы его будем по-жадному: будем смотреть, какое разбиение по переменной t сильнее всего уменьшает ошибку между предсказанным значением y и реальным, и выбирать его. В качестве предсказания будем использовать просто среднее значение будет по тем данным, которые попали в одну кучку.

На первом шаге у нас есть три способа сделать разбиение по переменной t:





- Мы можем отправить в левую вершину все ситуации, где температура меньше либо равна 8 градусам. В таком случае, когда мы идём по дереву налево, мы будем прогнозировать, что потребители выпьют 8 чашек кофе. Когда мы идём в правую вершину, мы будем прогнозировать, что потребители выпьют 3.6 чашек кофе. Это среднее всех y, попавших в правую вершину. Давайте посчитаем ошибку, которую при этом будет допускать дерево.

$$(8-8)^2 + (8-3.6)^2 + (2-3.6)^2 + (1-3.6)^2 = 28.68.$$

- Мы можем отправить в левую вершину все ситуации, где температура меньше либо равна 12. В таком случае слева прогноз будет 8, а справа 1.5. Найдём ошибку:

$$(8-8)^2 + (8-8)^2 + (2-1.5)^2 + (1-1.5)^2 = 0.5.$$

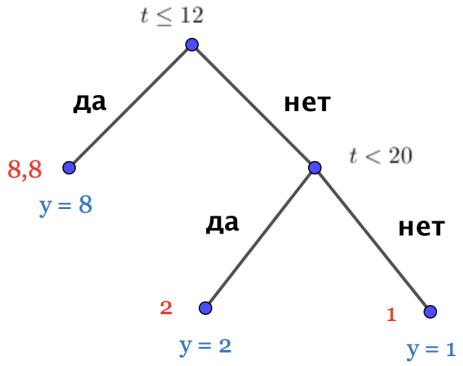
- В третьей ситуации получаем, что

$$(8-6)^2 + (8-6)^2 + (2-6)^2 + (1-1)^2 = 24.$$

Оптимальным для разбиения оказывается второй вариант. Он сильнее всего уменьшает ошибку. Выбрав его, мы отправляем в левую вершину две восьмёрки и получаем в ней нулевую

ошибку. В правую вершину мы отправляем двойку и единицу.

В правой вершине нужно сделать ещё одну итерацию, чтобы отделить двойку от единицы. Тогда обучение дерева будет окончено. Итоговое дерево будет иметь вид:



Ура! Конец. Компьютер обучает деревья ровно так, как мы сейчас сделали это руками. Правда обычно данных ему мы скармливаем намного больше.

Сделаем прогноз для 13 градусов. Для этого пройдёмся по дереву от корня к одному из листов. На улице меньше или равно 12 градусов? Нет. Идём направо. На улице меньше 20 градусов? Да. Идём налево. В кофейне купят 2 чашки.

Можно ли для обучения использовать MAE? А почему, собственно, нет?! Попробуйте дома проделать это руками самостоятельно. Всё что изменится — это способ оценки ошибки при каждом разбиении.

Обратите внимание, что дерево идеально запомнило обучающую выборку. Оно слишком сильно фрагментировало её. Такая ситуация, когда модель идеально вылизывает обучающие данные, называется **переобучением**. Чтобы деревья не переобучались и не вылизывали выборку, обычно останавливают обучение деревьев досрочно:

- Когда в вершине оказалось не менее 10 объектов
- Когда дерево построилось до 20 листьев.
- Когда глубина дерева оказалась равна 5.

Конечно же, конкретные цифры здесь для примера. Их можно (и нужно) подбирать по данным. Их, кстати говоря, называют **гиперпараметрами**.

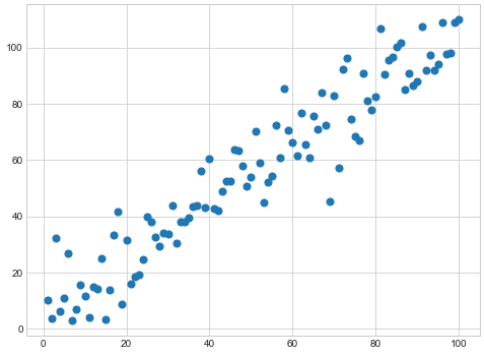
Ещё, чтобы деревья не переобучались, их обычно объединяют в леса. Именно этим мы с вами на следующей паре и займёмся, но уже на компьютере.

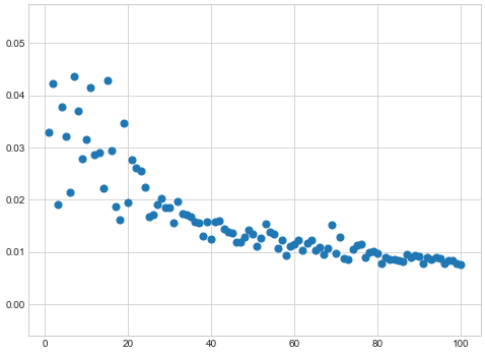
Ещё задачи

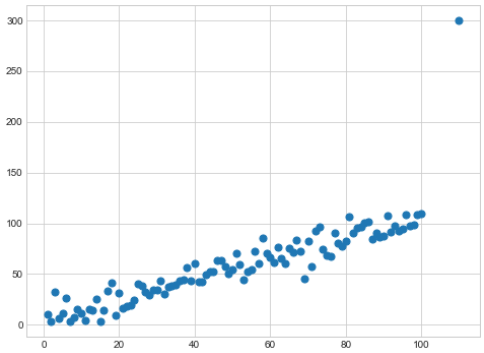
Тут находится несколько задачек, о которых вам нужно подумать самостоятельно. Не исключено, что похожие задачи попадутся вам на самостоятельной работе.

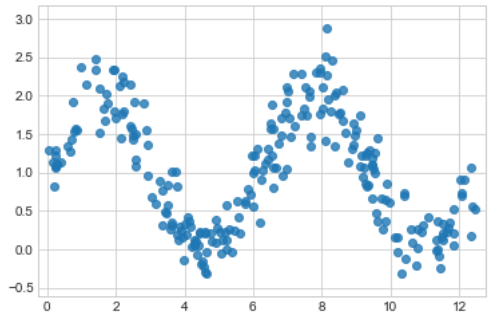
Упражнение 6

Вот несколько ситуаций, как, на ваш взгляд, будут выглядеть оптимальные линии регрессии? Да, это тоже машинное обучение. Но обычно кривые рисуем не мы, а комплюхтер.









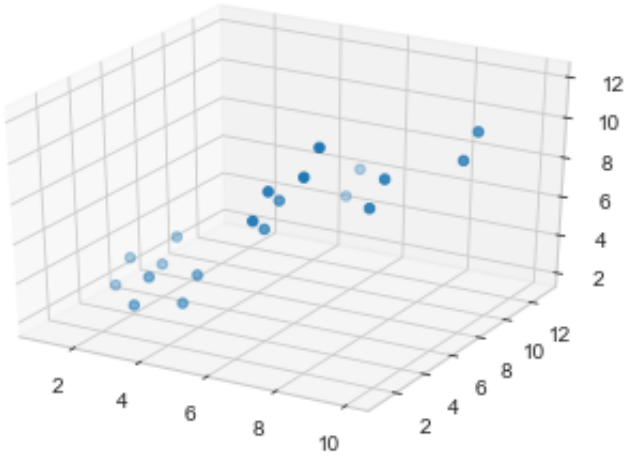
а) Нарисуйте на каждой из картинок линию регрессии.

б) Как выглядят уравнения регрессии в этих ситуациях? Какие параметры в них нам нужно обучить?

в) В чём проблема на картинке слева снизу? Приведите пример ситуации, когда может наблюдаться такая картинка.

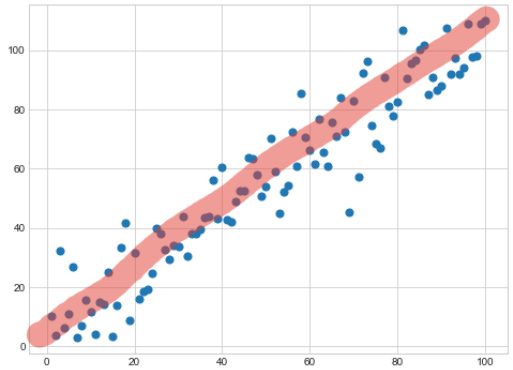
г) В четвёртой ситуации мы выбрали для обучения полином. А почему бы не взять его в каждой ситуации и не обучить через каждую точку?

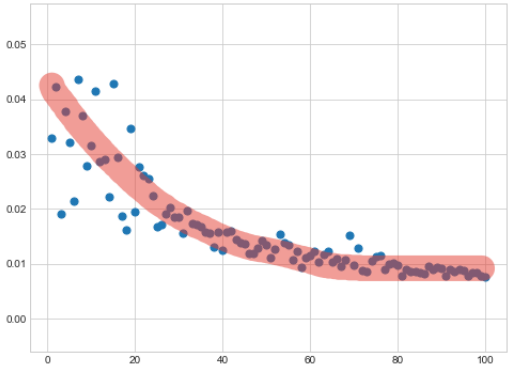
д) Ещё одна, на этот раз трёхмерная картинка! Слабо дополнить её также, как мы делали это выше? Как будет выглядеть уравнение регрессии?

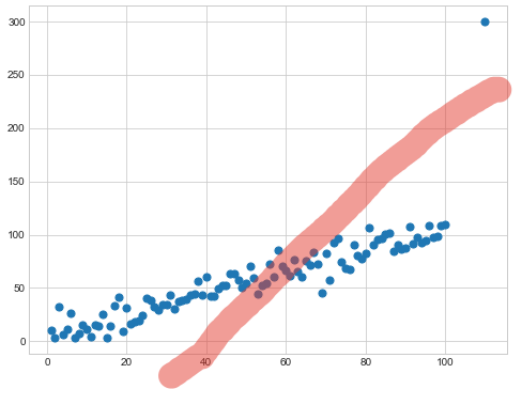


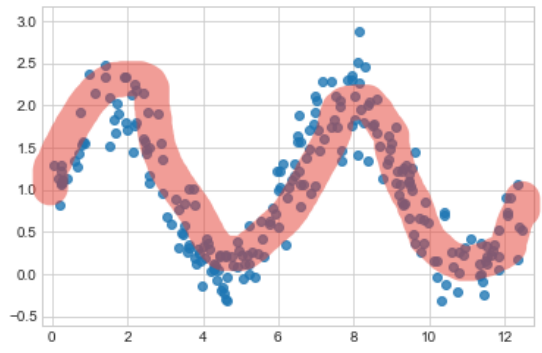
Решение:

а) Берём и рисуем!









б) В первой ситуации это обычная линейная модель $y_i = \beta_0 + \beta_1 x_i$. Во второй ситуации перед нами нелинейная модель. Внешне картинка похожа на гиперболу. Можно попробовать обучить модель $y_i = \frac{1}{\beta_0 + \beta_1 x_i}$. Однако на практике обычно поступают иначе. Если взять от x_i логарифм, то модель станет линейной, и можно будет обучить $y_i = \beta_0 + \beta_1 \ln x_i$. В третьей ситуации это снова обычная линейная модель. В четвёртой ситуации это либо многочлен, либо какой-нибудь косинус. Об этих двух ситуациях мы поговорим подробнее ниже.

в) В случае шашлыков это был бы кто-то, кто очень много ест. Он портит обучение модели и прямая вместо того, что бы пройти через облако точек, подстраивается под него. Такие

ситуации обычно **называют выбросами**. Если последовать рецепту из первого упражнения и наложить на этого друга дамми, то ситуация нормализируется, и красная прямая пройдёт сквозь облако также как и в первой ситуации. Это эквивалентно тому, что мы выбрасываем друга из выборки и работаем с ним отдельно.

Другой путь: использовать модели, которые нечувствительны к выбросам. Например, использовать для обучения модели MAE!

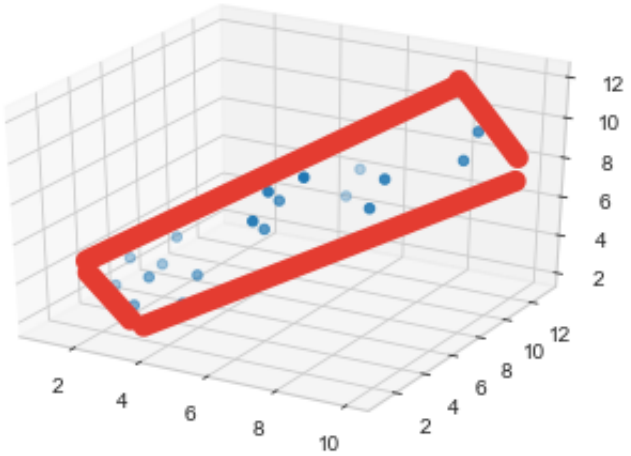
г) В четвёртой ситуации мы взяли полином. Возможно, у вас возник соблазн обучить и в первых трёх ситуациях модель, которая пройдёт через все возможные точки. Это неправильно. В таком случае наша модель слишком сильно вылизывает данные. Обычно в данных много шума, и модель подстраивается под него, вместо того, чтобы вычленить сигнал, **то есть переобучается.**

д) В этой ситуации мы строим модель не на одну переменную (y на x), а на две (y на x и на z). Уравнение будет иметь вид

$$y_i = \beta_0 + \beta_1 \cdot x_i + \beta_2 \cdot z_i.$$

В алгебре такое уравнение описывает двумерную плоскость в трёхмерном пространстве. Новость: в трёхмерном случае мы учим не линию, а плоскость!

Интерпретируется ситуация довольно просто: например, Олег при покупке машины решил учитывать не только её возраст, но и средний пробег.



Упражнение 7

Драгомир пытается предсказать продажи видео-игр. Для моделирования он использует две переменные: x_1 — возраст игры, x_2 — на кого она ориентирована. Если на мужчин, $x_2 = 1$, если на женщин, $x_2 = 0$. Целевая переменная y — количество проданных экземпляров игры. Драгомир оценил линейную регрессию:

$$y = 1000 - 100 \cdot x_1 + 200 \cdot x_2.$$

Проинтерпретируйте полученные коэффициенты. Предположим, что мы выпускаем на рынок свежую игру для женщин. Спрогнозируйте наши продажи.

Решение:

Получается, что $\beta_0 = 1000$ — это базовые продажи, $\beta_1 = -100$ — насколько падают продажи игры с каждым новым годом её присутствия на рынке, $\beta_2 = 200$, насколько выше продажи игр, которые ориентированны на мужчин.

Чтобы сделать прогноз, просто подставим в уравнение интересующие нас условия:

$$\hat{y}_{new} = 1000 - 100 \cdot 0 + 200 \cdot 0 = 1000$$

Получится 2000 экземпляров игры.

Упражнение 8

Мстислаполк, конкурент Драгомира, тоже пытается предсказать продажи видео-игр. Для моделирования он использует две переменные: x — возраст игры. Целевая переменная y — сумма продаж. Мстислаполк оценил линейную регрессию:

$$\ln y = 5 - 6 \cdot \ln x.$$

Проинтерпретируйте полученный коэффициент. Предположим, что мы отгружаем на рынок новую партию игры, выпущенной в прошлом году. Сколько экземпляров этой игры будет продано?

Решение:

Начнём с прогноза:

$$\ln \hat{y}_{new} = 5 - 6 \cdot \ln 1 = 5 - 0 = 5,$$

то есть 5 логарифмов экземпляров игры. Чтобы перейти от логарифмов игр просто к играм, надо сделать действие обратное логарифмированию. То есть взять от 5 экспоненту:

$$\hat{y}_{new} = \exp(5) \approx 148,$$

То есть мы продадим 148 экземпляров игры. Теперь давайте проинтерпретируем коэффициенты. Базовый логарифм продаж это $\beta_0 = 5$. При росте возраста игры на 1%, продажи будут падать *примерно* на 6%. За это отвечает коэффициент $\beta_1 = 6$.

Давайте ради интереса убедимся в последнем утверждении. Возраст игры у нас был равен 1. Если игра станет на процент старше, получится 1.01. Спрогнозируем y. Несложно посчитать, что это будет:

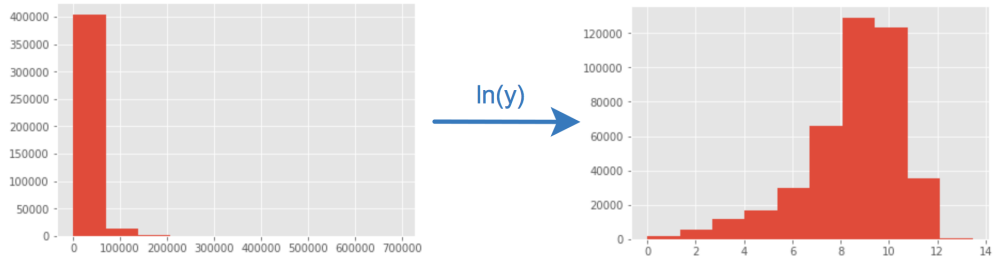
$$\hat{y} = \exp(5 - 6 \cdot \ln 1.01) \approx 140.$$

Если подставить все данные без округлений, получается, что продажи упали на $100 \cdot$

$\frac{\hat{y}_{new} - \hat{y}}{\hat{y}_{new}} \approx 5.7\%$.

Упражнение 9

Логарифмирование позволяет сгладить длинные хвосты распределений, кишащие выбросами. Из-за этого на практике переменные довольно часто логарифмируют. На картинке ниже изображена гистограмма продаж в супермаркетах Wallmart. По оси x отложена сумма продаж, по оси y число продаж на такую сумму.



Понятное дело, что люди делают покупки на огромные суммы, но в маленьком количестве. Отсюда у распределения появляется огромный хвост. Если прологарифмировать продажи, распределение станет няшным.

Попробуйте посмотреть как именно происходит это сглаживание. Предположим, что в магазинах продали $y_1 = 100$, $y_2 = 200$, $y_3 = 300$ и $y_4 = 1000$ игр. Посчитайте разницу между соседними наблюдениями. Прологарифмируйте их. Что стало с этой разницей?

Решение:

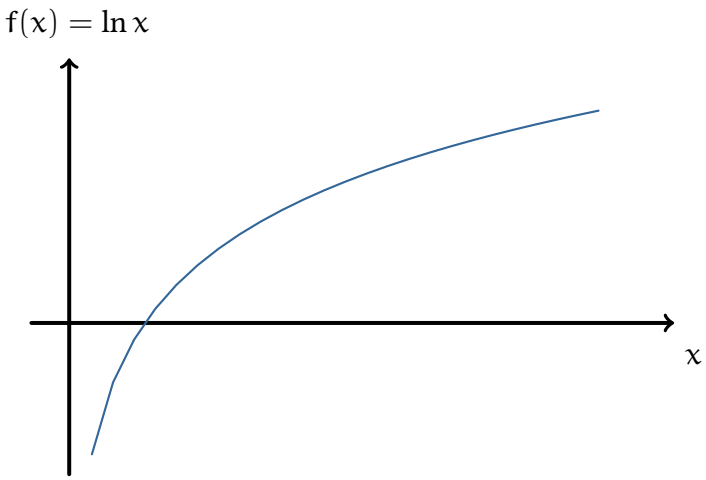
Возьмём логарифмы:

$$\ln y_1 = 4.6, \ln y_2 = 5.3, \ln y_3 = 5.7, \ln y_4 = 6.9.$$

Наблюдения стали намного плотнее прилегать друг к другу. Выброс y_4 довольно сильно сгладился. Именно это мы видим на гистограмме выше. Посмотрим на то, что происходит с разностями между соседними наблюдениями:

$$\begin{aligned} y_2 - y_1 &= 100 \qquad & \ln y_2 - \ln y_1 &= \ln 200 - \ln 100 = 0.69 \\ y_3 - y_2 &= 100 \qquad & \ln y_3 - \ln y_2 &= \ln 300 - \ln 200 = 0.4 \\ y_4 - y_3 &= 900 \qquad & \ln y_4 - \ln y_3 &= \ln 1000 - \ln 300 = 1.2 \end{aligned}$$

Между y_1, y_2 и y_2, y_3 до логарифмирования была одинаковая разность (100). Теперь, из-за того, что логарифм сгладил хвост, y_3 ближе к y_2, чем y_2 к y_1. В принципе, если построить график с логарифмом, эти свойства видны на нём. Чем больше x, тем меньший прирост по y происходит.



Такие вот у логарифма классные свойства. Конечно же, **логарифмирование не всегда спасает от длинных хвостов,** но иногда оно бывает довольно полезным преобразованием.

Упражнение 10

В один прекрасный день Маша проснулась в своей кровати и поняла, что она и есть та самая маша, которой принадлежит лёрнинг. Она решила посвятить машин лёрнингу всю свою жизнь и стала коллекционировать модели.

Вчера она пообщалась с Мишей. Он тоже коллекционер. Он спросил у неё, какое у её моделей качество. Маша не смогла ответить. Ей было очень стыдно[3]. Она решила проверить качество. У неё есть три наблюдения y_i. Она для каждого построила прогнозы. Найдите для её прогнозов MAE, MSE, RMSE и MAPE.

настоящие y_i	1	2	3
прогнозы нейросети	2	3	1
прогнозы регрессии	2	3	4
прогнозы случайного леса	1	1	1

Упражнение 11

Объясните мемас:

[3]Прям как вам после самостоятельной на следущей паре

Упражнение 12

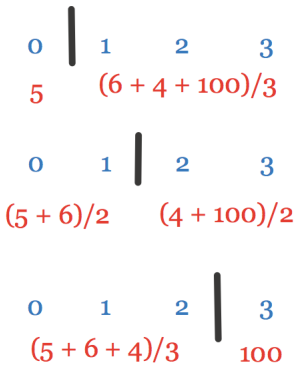
Выращиваем регрессионное дерево в домашних условиях! Вот вам выборка для этого:

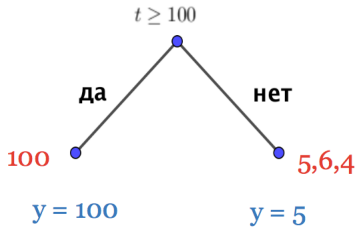
x_i	y_i
0	5
1	6
2	4
3	100

Критерий деления вершины — минимизация квадратичной функции потерь (MSE). Критерий остановки — три листа. Зачем нужен критерий остановки? Как дерево ведёт себя с выбросами?

Решение:

У нас есть три способа раздробить по x дерево.



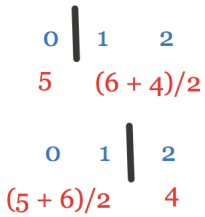


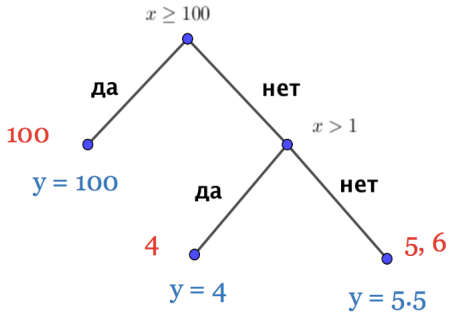
Посчитаем для каждого способа квадратичную ошибку:

- $(5-5)^2 + (6-36.6)^2 + (4-36.6)^2 + (100-36.6)^2 = 6018.68$
- $(5-5.5)^2 + (6-5.5)^2 + (4-52)^2 + (100-52)^2 = 4608.5$
- $(5-5)^2 + (6-5)^2 + (4-5)^2 + (100-100)^2 = 2$

Выгоднее всего оказывается обособить первым же отсечением выброс. Это нормальная ситуация. На практике так происходит регулярно. **Деревья изолируют выбросы в отдельные вершины, и они никак не портят работу с основной выборкой. Такое свойство называется нечувствительностью к выбросам или робастностью к выбросам.**

Сделаем второй шаг разбиения.





Посчитаем для каждого способа квадратичную ошибку:

- $(5-5)^2 + (6-5)^2 + (4-5)^2 = 2$
- $(5-5.5)^2 + (6-5.5)^2 + (4-4)^2 = 0.5$

Понятно, что дробить нужно, обосабливая четвёрку. После этого нужно остановиться. По условию задачи критерий остановки — три листа у дерева. Ошибка бы продолжила убывать для тренировочной выборки. На тестовой она бы возрастала. Обычно подобные критерии ранней остановки помогают избежать переобучения.

Кстати говоря, именно благодаря тому, что деревья на первом же шаге изолируют выбросы, случайный лес можно из прогнозной модели модернизировать в модель, которая неплохо справляется с поиском аномалий. Подумайте на досуге о том, как именно можно сделать это.

Упражнение 13

Бернард не очень хорошо умеет в маркетинг и управление бизнесом, а ещё он — владелец книжного магазина. Он обратил внимание, что чем чаще в магазин заходят покупатели, тем чаще он пьёт. Чай. Ещё бывают постоянные покупатели, которые, в общем-то, ничего не покупают, а только делятся проблемами, выпивают чай и уходят.

Бернард хочет прикинуть свои расходы на чай в следующем месяце. Какую информацию ему надо добыть и где её достать? Какие метрики использовать при построении модели? Какие метрики использовать при оценке её качества?