В прошлый раз...

- Поговорили о задаче классификации
- Научились генерировать синтетические данные
- Узнали про три семейства моделей
- Вывели общий процесс подготовки моделей машинного обучения

Вопросы?

План занятия

- Как обучаются модели
- Линейные модели
- Предобработка данных
- Более подробно о метриках
- О требованиях бизнеса к решению задач методами машинного обучения

Материал для самоподготовки, повторения

- Открытый курс по машинному обучению
  - Текст
  - Видео
- Метрики в задаче классификации
- Метрики в задаче регрессии
- ODS про линейные модели
- ODS базовые принципы машинного обучения на примере линейных моделей

Освежим в памяти

Что делают алгоритмы машинного обучения?

Освежим в памяти

Что делают алгоритмы машинного обучения?

Имея ограниченный набор данных, восстанавливают генеральную зависимость

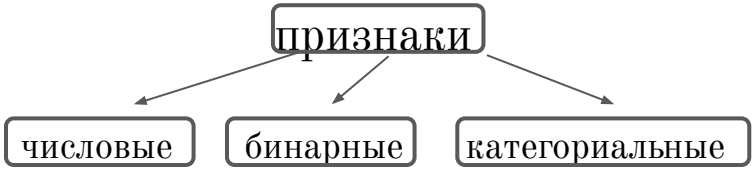
Освежим в памяти

Как представляются данные для алгоритмов машинного обучения?

Освежим в памяти

Как представляются данные для алгоритмов машинного обучения?

Объект x задаётся *признаковым описанием*



$f_1, f_2, \ldots, f_k$ — признаки (features) объекта x

$$\begin{bmatrix} f_1(x_1), & f_2(x_1), \ldots, & f_k(x_1) \\ f_1(x_2), & f_2(x_2), \ldots, & f_k(x_2) \\ & \ldots & \\ f_1(x_n), & f_2(x_n), \ldots, & f_k(x_n) \end{bmatrix}$$

— матрица “объекты-признаки”
объект, пригодный для применения
алгоритмов машинного обучения

Освежим в памяти

Что могут и не могут алгоритмы машинного обучения?

Освежим в памяти

Что могут алгоритмы машинного обучения?

- Найти оптимальную взвесь признаков, и *точнее* человека определить давать ли человеку кредит
- Предсказывать спрос на услуги
- Находить *похожие* объекты на основе признакового описания
- Прогнозировать нагрузку на информационную систему
- Рекомендовать релевантные темы
- Найти все дорожные знаки на изображении
- Рефлексировать, действовать за рамками поставленной задачи

Освежим в памяти

Что такое машинное обучение с учителем?

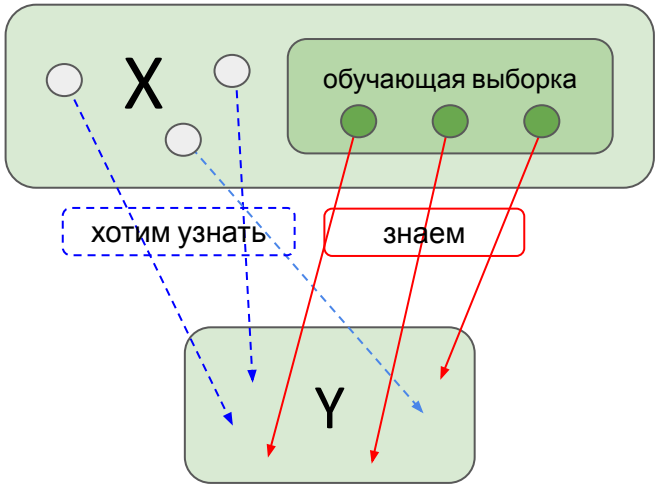
Освежим в памяти

X — множество *объектов*

Y — множество *ответов* (например, два класса или произвольные числа)

$y: X \to Y$ — неизвестная закономерность

Дано: *обучающая выборка*, $\{x_1, x_2, \ldots, x_n\}$ — подмножество множества X
Цель: подобрать *алгоритм*, приближающий функцию $y(x)$.



Освежим в памяти

Какие задачи решают алгоритмы машинного обучения с учителем?

Освежим в памяти

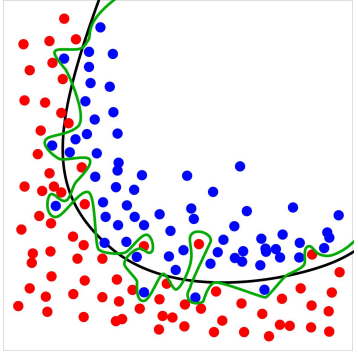
Какие задачи решают алгоритмы машинного обучения с учителем?

Существует два типа контролируемых алгоритмов машинного обучения: регрессия и классификация. Первый прогнозирует непрерывные выходные значения, а второй - дискретные выходные. Например, прогнозирование цены дома в долларах является проблемой регрессии, тогда как прогнозирование того, является ли опухоль злокачественной или доброкачественной, является проблемой классификации.

Освежим в памяти

Что такое переобучение?

Освежим в памяти



Что такое переобучение?

Переобучение — это результат чрезмерной подгонки параметров модели к зависимостям, содержащимся в обучающем множестве. Если происходит переобучение, то модель не приобретает способности к обобщению — возможности распространять обнаруженные на обучающем множестве зависимости и закономерности на новые данные.

Освежим в памяти

Каковы этапы построения модели? Опишите типичный рабочий процесс

Освежим в памяти

Каковы этапы построения модели? Опишите типичный рабочий процесс

1. Загрузка данных
2. Постановка задачи. Определение метрик качества
3. Визуальный анализ данных
4. Представление данных в корректном для алгоритма виде
5. Разделение на обучающую и тестовую выборки
6. Подбор модели на *кросс-валидации* ← уменьшаем разброс результатов, контролируем переобучение
7. Сохранение промежуточных результатов. Предположения о возможных проблемах, возврат к шагу 2, либо переход к шагу 8
8. Сохранение модели, переход к её эксплуатации и поддержке

Как обучаются модели?

Понятия, обозначения

- X — пространство объектов
- Y — пространство ответов
- x = (x1, ..., xd) — признаковое описание объекта
- X = (xi, yi) i=1..L — обучающая выборка
- a(x) — алгоритм, модель
- Q(a, X) — функционал ошибки алгоритма a на выборке X
- Обучение: $a(x) = \operatorname{argmin}_{a \in A} Q(a, X)$

Функционал ошибки Q

По другому называется loss. Чем меньше значение Q, тем лучше обучена модель

Примеры:

- Для регрессии
  - *Может* совпадать с метрикой качества: MSE
- Для классификации
  - Логистическая функция потерь (logloss, кросс-энтропия)
  - Экспоненциальная функция потерь
  - ...

Отличие loss (Q) от метрики

- loss используется для подбора параметров модели на тренировочном наборе
- метрика используется для оценки модели на тестовом наборе
- они могут быть как одинаковыми функциями, так и разными
- метрика не обязательно должна быть дифференцируема
- метрик может быть несколько, а loss у модели один

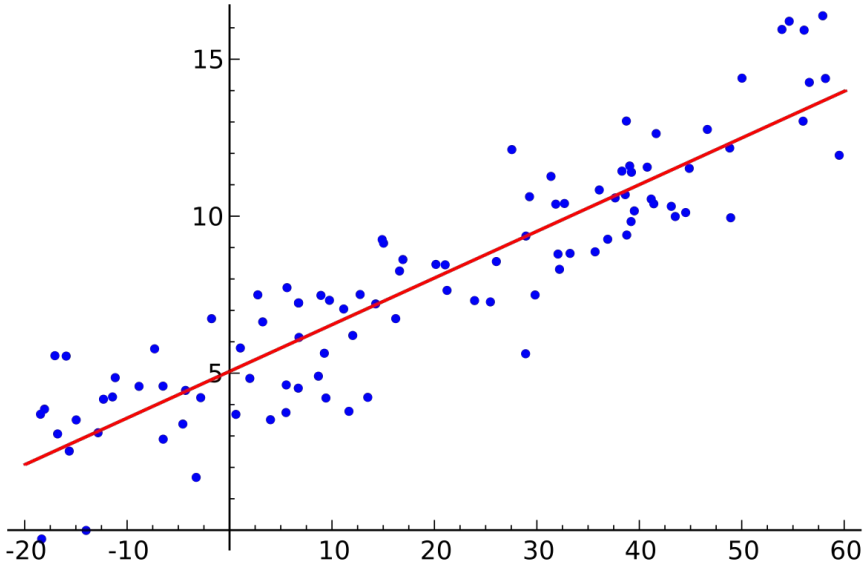
Вспомним

- Задача классификации (classification) отличается тем, что множество допустимых ответов конечно. Их называют метками классов (class label). Класс — это множество всех объектов с данным значением метки.
- Задача регрессии (regression) отличается тем, что допустимым ответом является действительное число или числовой вектор.

В прошлый раз говорили про классификацию, сегодня поговорим про регрессию

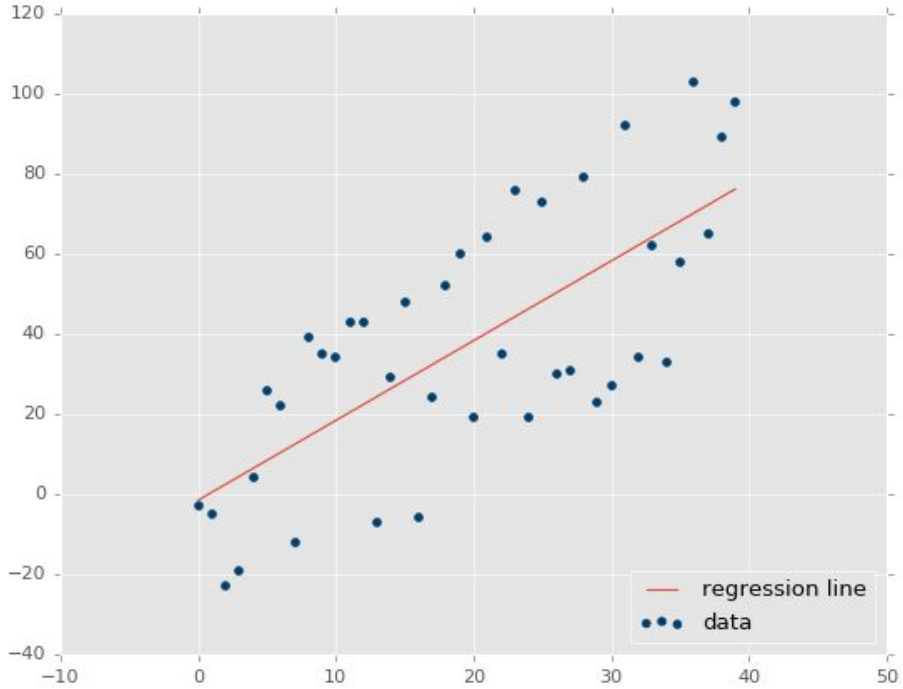
Линейная регрессия

Линейная регрессия. Также известна как МНК - метод наименьших квадратов.



Линейная регрессия

Y= mx + b



Colab? Colab!

Метод наименьших квадратов

Линейная зависимость:

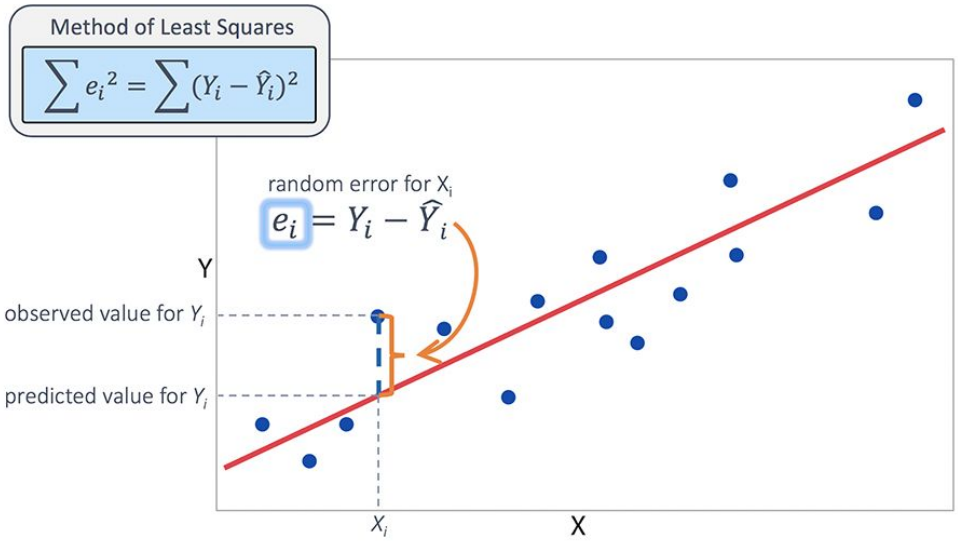
$$\hat{y}_i = wx_i + b$$

Где w и b нужно подобрать.

Среднеквадратичная ошибка:

$$\mathcal{L} = \sum_i (wx_i + b - y_i)^2$$

Будем называть это loss, или “функция потерь”



Линейные модели

$$a(x) = w_0 + \sum_{j=1}^{d} w_j x^j$$

свободный коэффициент — w_0; w_j — веса; x^j — признаки

$$Q(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} (a(x_i) - y_i)^2$$

для регрессии

Точное решение

В случае многих переменных:

$$\hat{y}_i = W\vec{x}_i + b$$

$$\mathcal{L} = \sum_i (W\vec{x}_i + b - y_i)^2$$

Целевая переменная скаляр (число)

Нахождение минимума сводится к решению системы линейных уравнений:

$$\frac{\delta\mathcal{L}}{\delta W_j} = \frac{\delta\mathcal{L}}{\delta b} = 0$$

Или

$$\sum_i \vec{x}_i(W\vec{x}_i + b - y_i) = 0$$

Перепишем:

$$\sum_i \vec{x}`_i(\vec{x}`_i W` - y_i) = 0$$

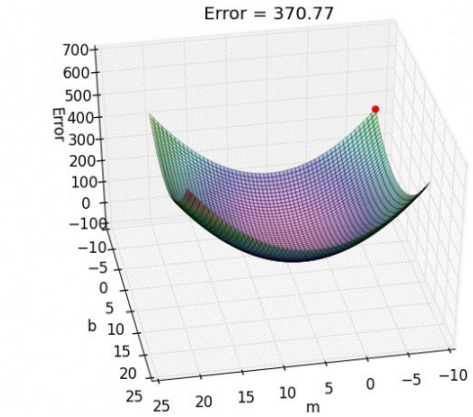
Еще перепишем:

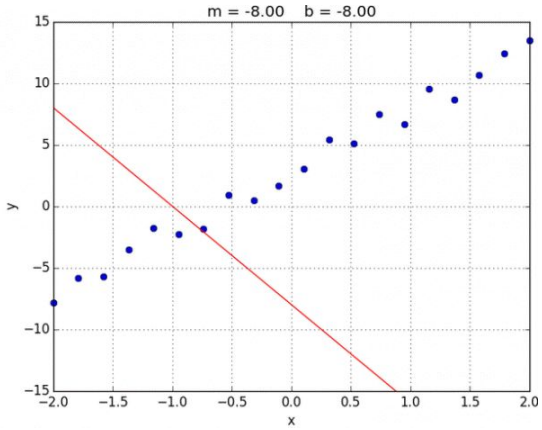
$$X`^T X`W` - X`^T Y = 0$$

$$W = (X^T X)^{-1} X^T Y \quad <-$$

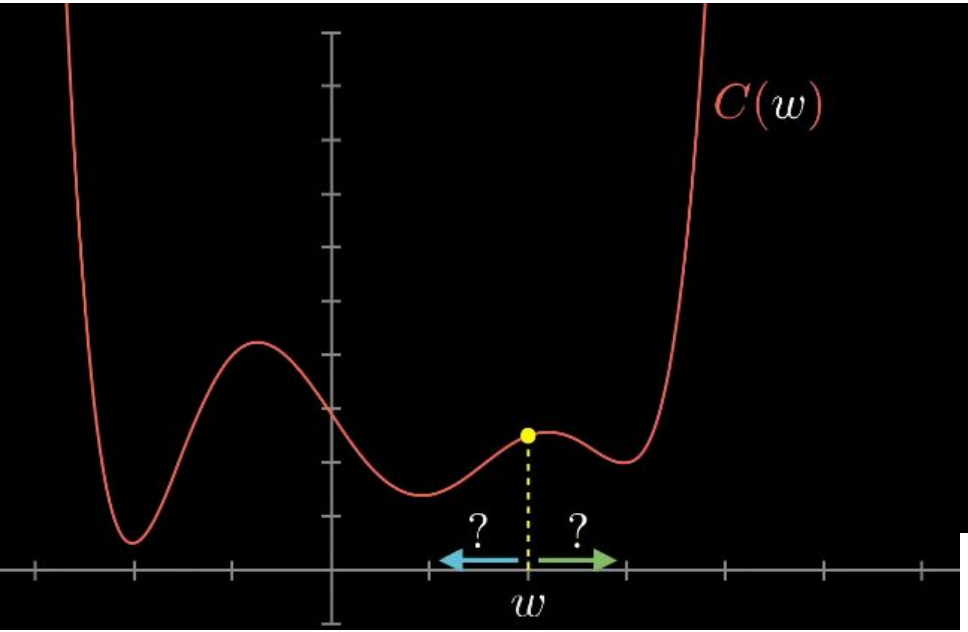
Градиентный спуск

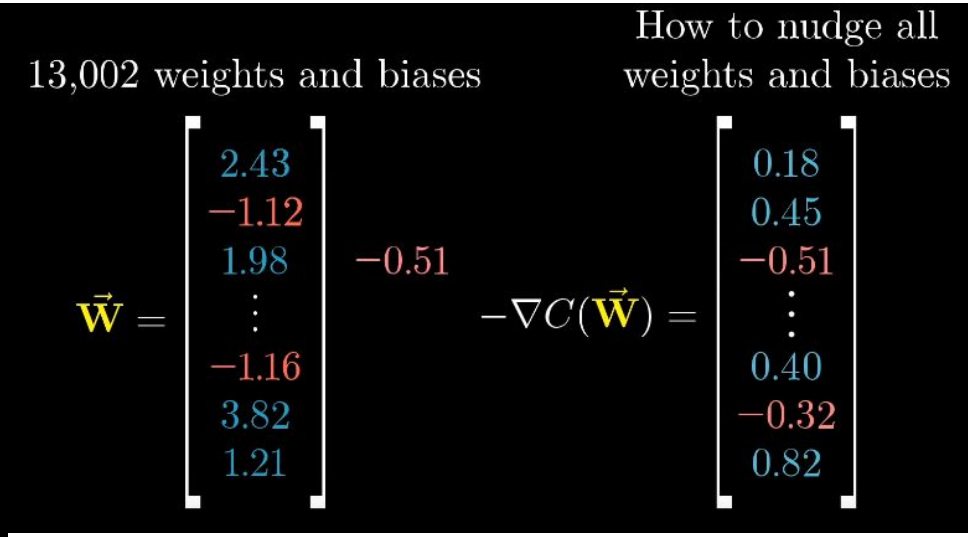
- Точное решение существует и единственно.
- Однако при большом кол-ве параметров и данных искать его аналитически становится невыгодно.
- Градиентный спуск часто оказывается быстрее.
- Минимум по-прежнему один и находится быстро





Градиентный спуск



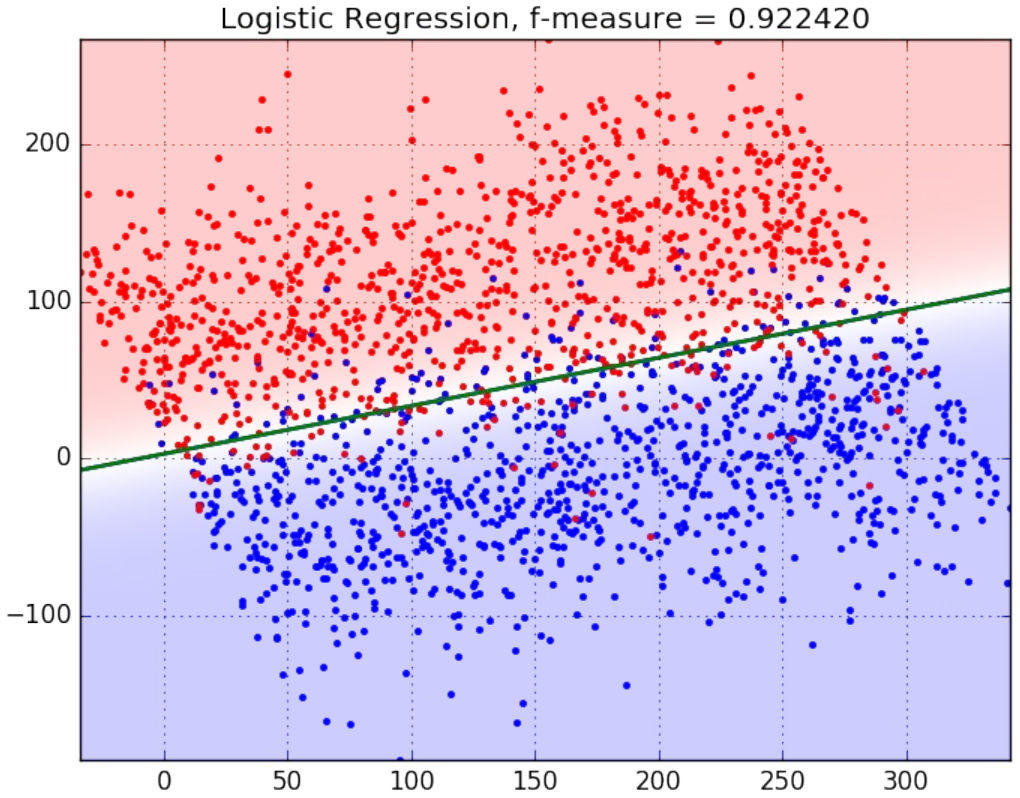


- https://www.youtube.com/watch?v=aircAruvnKk

Colab? Colab!

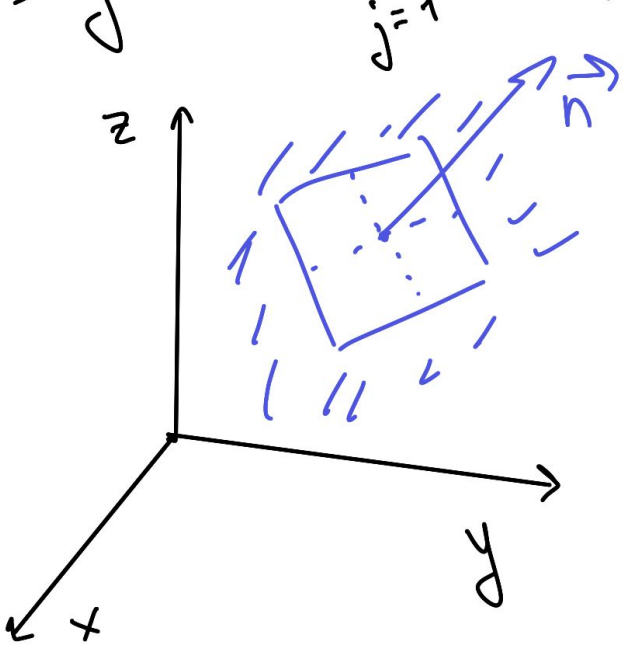
Logistic regression

- Логистическая регрессия - линейный метод **классификации**
- Название исторически сложилось, т.к. этот метод предсказывает вероятность
- Также называют линейным классификатором



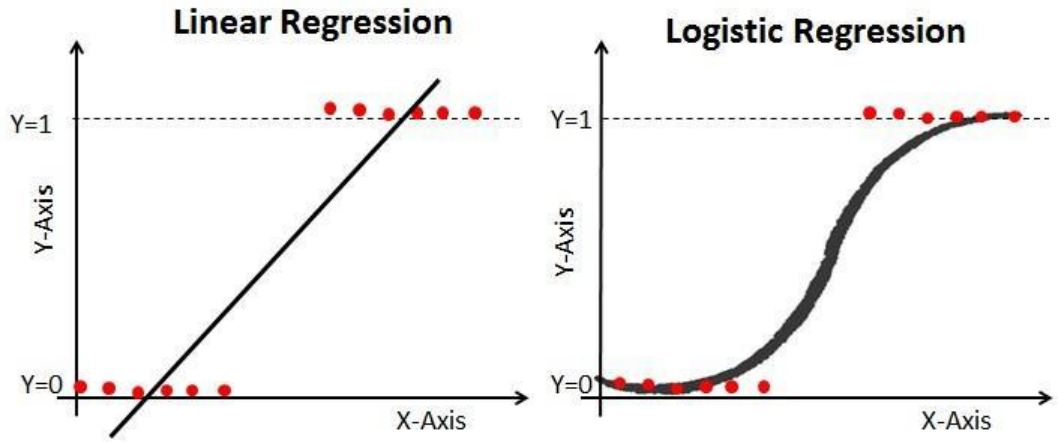
Линейный классификатор

$$a(x) = sign\left(w_0 + \sum_{j=1}^{d} w_j x^j\right)$$



гиперплоскость!

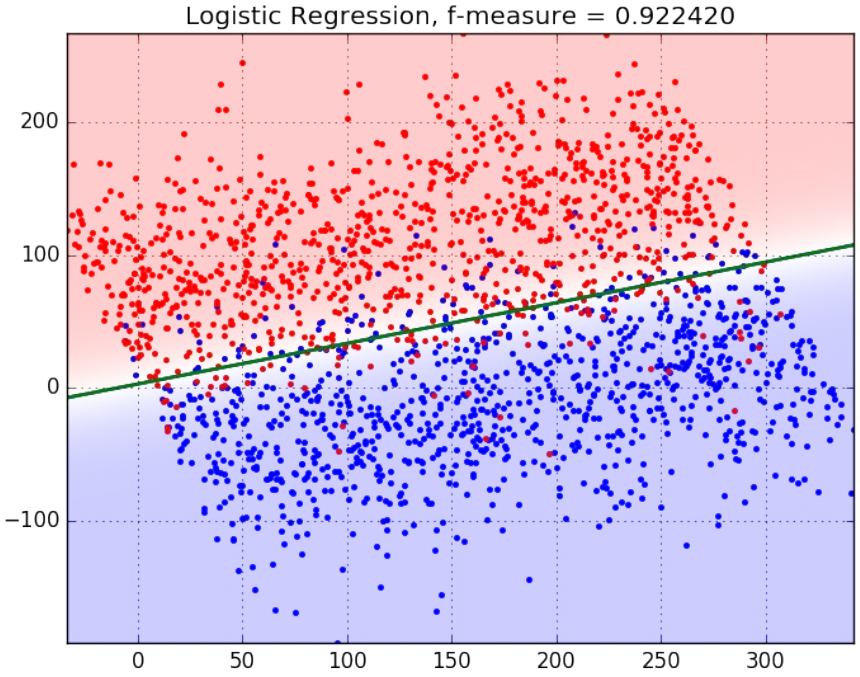
Логистическая функция



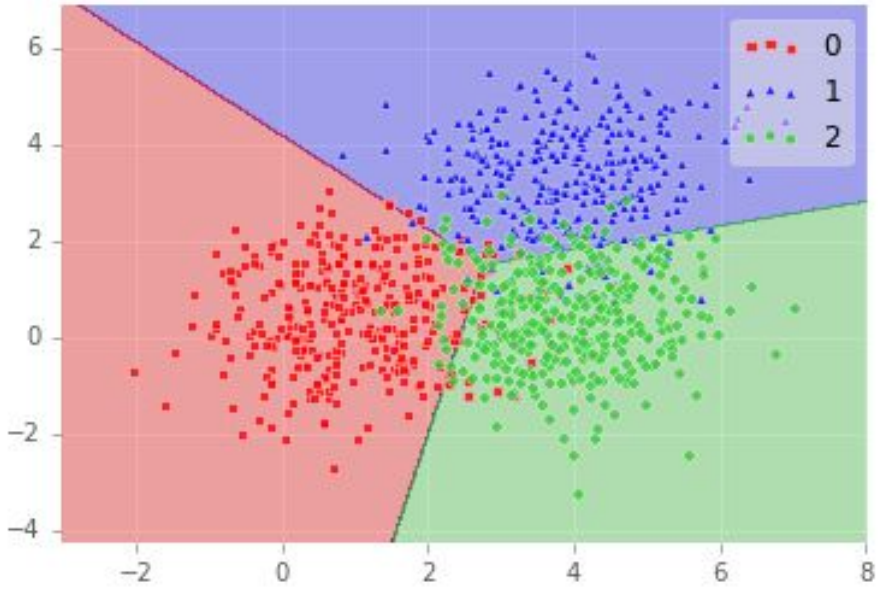
$$y = \frac{1}{1 + e^{-(W\vec{x}+b)}}$$

Визуализация в 2d

$$y = sigmoid(W\vec{x})$$



$$\vec{y} = softmax(W\vec{x})$$



А что если...

Данные линейно неразделимы?

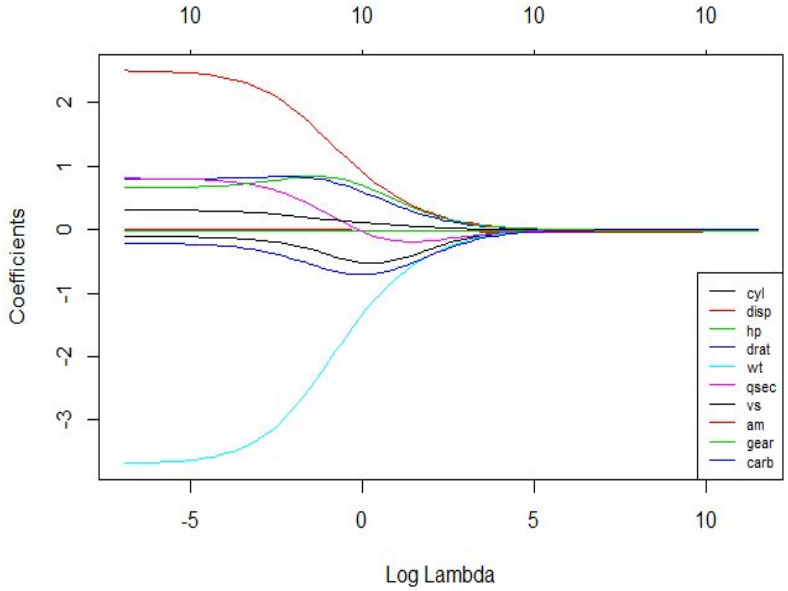
Существует техника под названием Kernel Trick, но она остается за рамками нашего курса

Регуляризация

L2 регуляризация

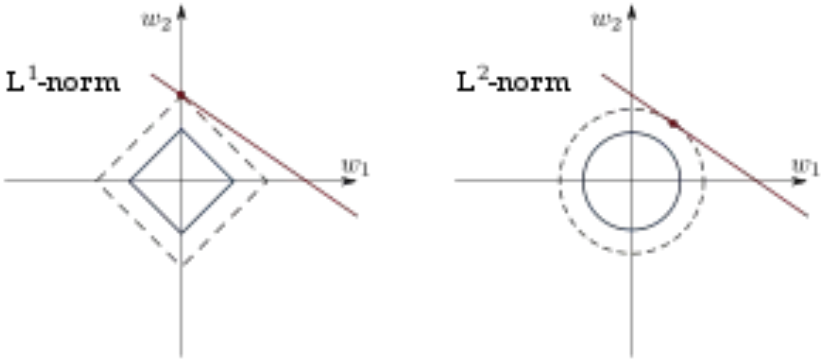
$$\mathcal{L}' = \mathcal{L} + \lambda \cdot ||w||_{L_2}$$

- Некоторые признаки линейно зависимы
- По некоторым признакам статистика представлена мало
- Можно добавить веса к функции потерь, и тогда модель будет получать штраф за большие веса
- В случае линейной регрессии такая модель называется ridge regression
- Помогает бороться с переобучением, делает разброс значений весов меньше

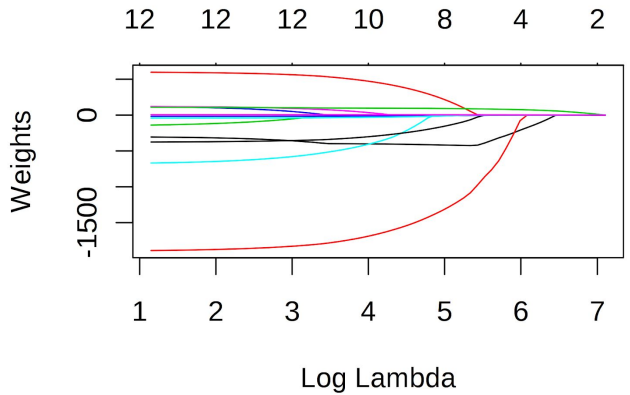


L1 регуляризация

- Норма L1 тоже применяется, и ведет себя по-другому.
- Модель линейной регрессии с такой регуляризацией называется lasso regression
- LASSO - least absolute shrinkage and selection operator
- Полезна для отбора признаков, так как стремится “занулить” менее значимые признаки



$$\mathcal{L}' = \mathcal{L} + \lambda \cdot ||w||_{L_1}$$



Предобработка данных

Кодировка категориальных признаков

Два основных подхода: LabelEncoder и OneHotEncoding

LabelEncoder

	Bridge_Types	Bridge_Types_Cat
0	Arch	0
1	Beam	1
2	Truss	6
3	Cantilever	3
4	Tied Arch	5
5	Suspension	4
6	Cable	2

OneHotEncoding

id	color
1	red
2	blue
3	green
4	blue

id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

Масштабирование признаков

- Стоит масштабировать признаки для ускорения сходимости градиентного спуска
- Можно проводить стандартизацию: вычитать среднее и делить на стандартное отклонение признака
- Можно масштабировать на отрезок [0,1]

Масштабирование признаков

Чем еще может быть полезно масштабирование?

Возраст	Зарплата	Стоимость машины
30	100000	1000000
20	30000	200000
25	80000	300000
40	200000	2000000

стоимость =
<коэф. возраст> * <возраст> + <коэф. зарплата> * <зарплата> + <свободный коэффициент>

В результате оптимизации среднеквадратичного функционала ошибки мы получим коэффициенты в разных масштабах, а значит не сможем с точностью сказать какой фактор оказывает доминирующее влияние

Бинарные признаки можно не трогать, а категориальные можно перевести в бинарные с помощью one hot encoding (sklearn)

Colab? Colab!

Ещё раз про метрики

Применение метрик качества

- Иногда ими задают функционал ошибки (loss)
- Подбор гиперпараметров моделей на кросс-валидации
- Оценивание итоговой модели

Среднеквадратичная ошибка

- Сильно штрафует за большие ошибки, поскольку отклонения возводятся в квадрат
- Анализ выбросов!

Средняя абсолютная ошибка

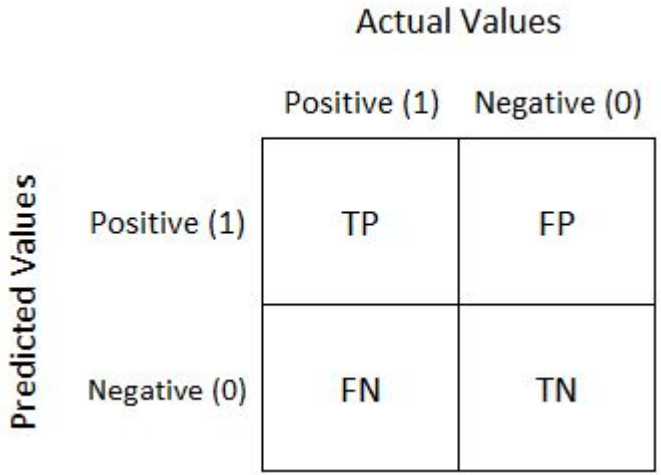
- Сложнее минимизировать, так как у модуля производная не существует в нуле
- Более устойчив к выбросам

Доля правильных ответов

- Другое название accuracy
- Не очень подходит, если классы несбалансированные

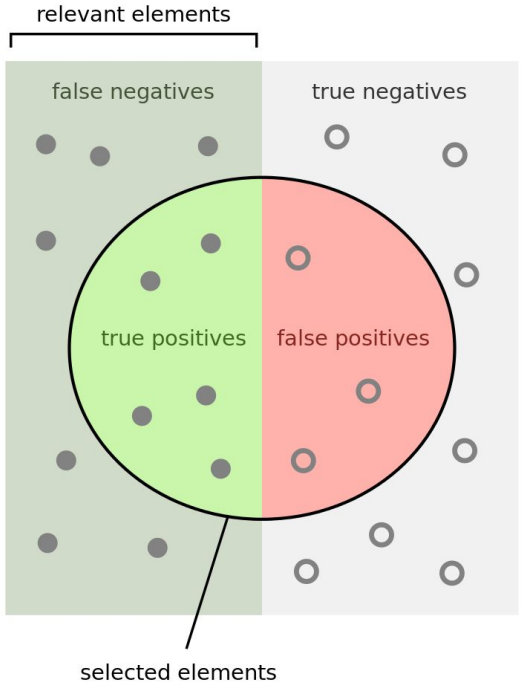
Точность и полнота

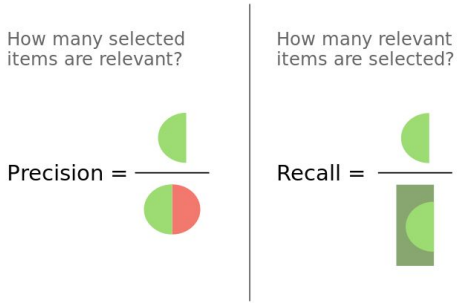
Какова цена
ошибки?



Точность и полнота

- Вычисляются для одного из классов
- Точность: precision. Сколько из выбранных элементов действительно принадлежат данному классу?
- Полнота: recall. Сколько объектов из данного класса алгоритму удалось найти?





F-мера

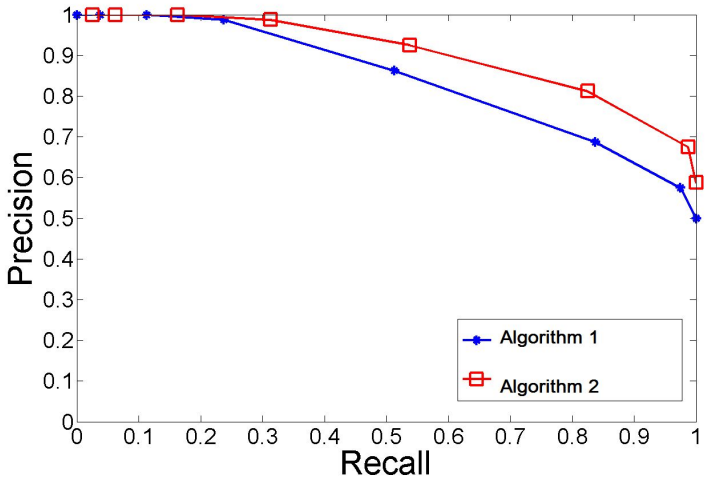
$$F = \frac{2 * (Precision * Recall)}{Precision + Recall}$$

$$F = (\beta^2 + 1)\frac{Precision * Recall}{\beta^2 Precision + Recall} \qquad \beta^2 \in [0, \infty]$$

Где β принимает значения в диапазоне 0 <β <1, если Вы хотите отдать приоритет точности,
а при β> 1 приоритет отдается полноте.
При β=1 формула сводится к предыдущей и вы получаете сбалансированную F-меру (также ее называют F_1)

PRC-AUC

- Оценивает качество самой оценки алгоритма, не фиксируя порог
- Строится в координатах precision и recall
- В идеальном случае будет проходить через точку (1,1)
- Начинается в точке (0,0), заканчивается в точке (1,r), где r доля объектов класса 1

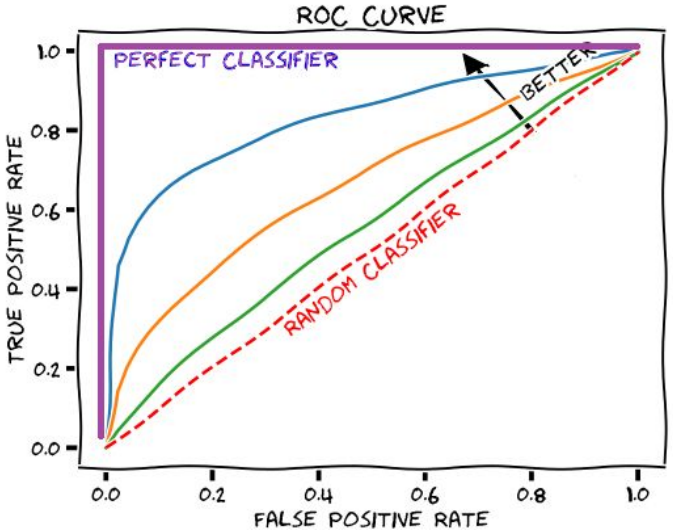


ROC AUC

$$FPR = \frac{FP}{FP + TN}$$

$$TPR = \frac{TP}{TP + FN}$$

- Оценивает качество самой оценки алгоритма, не фиксируя порог
- [Очень рекомендую к прочтению]
- Доля пар объектов вида (объект класса 1, объект класса 0), которые алгоритм верно упорядочил
- Строится в координатах FPR и TPR
- 1 идеально, 0.5 рандом, 0 ужасно



Colab? Colab!

Резюме

- Обсудили более подробно что из себя представляет обучение моделей
- Разобрали подробно класс линейных моделей
- Поговорили о предобработке данных
- Разобрали метрики более подробно
- Узнали, что оптимальное значение метрики не всегда ведет к оптимальному результату для бизнеса

Спасибо за внимание!