

# Week 02

## Lecture 02

# Цели

1. Краткий обзор прошлой лекции
2. Прогнозирование дефектов программного обеспечения
  - прогнозирование количества дефектов
  - Рассмотрите эту задачу в контексте ML
3. Что такое линейная регрессия? Какова ее целевая функция? Как она мотивируется?
4. Вывод решения в замкнутой форме для линейной регрессии

# Отражение

- Что именно означает для машины научиться чему-то?
- Зачем использовать машинное обучение в таких приложениях, как фильтрация спама?
- Какие проблемы относятся к проблемам обучения под наблюдением?
- Почему оценка цены дома с учетом его характеристик является задачей регрессии, а фильтрация спама - задачей классификации?
- Чем контролируемое машинное обучение на основе данных похоже на оценку функций?
- Что такое недоучет и переучет и как их обнаружить?
- Всегда ли сложная модель будет подходить больше, а простая - меньше?

# Обзор (1)

## What is Machine Learning?

- A subfield of artificial intelligence
- Computer programs that improve their performance at some task through experience
- Examples: object recognition, spam detection, disease prediction, weather forecasting, etc.

## Parametric Models

$$y = f(x; \text{parameters})$$

$$y = f(x; w)$$

$$y = f(x; w_0, w_1) = w_0 + w_1 x$$

## Goal of Learning

- Learning or inferring a “functional” relationship between predictors and target

$$D = \{x_i, y_i\}_{i=1}^N$$

$$x \in \mathbb{R}^d$$

$$\hat{f} \approx f \quad \text{Goal of learning}$$

$$y = f(x)$$

## Classification and Regression

Country	Age	Salary	Purchased
France	44	72000	No
Spain	27	48000	Yes
Germany	30	54000	No
Spain	38	61000	No
Germany	40		Yes
France	35	58000	Yes
Spain		52000	No
France	48	79000	Yes
Germany	50	83000	No
France	37	67000	Yes

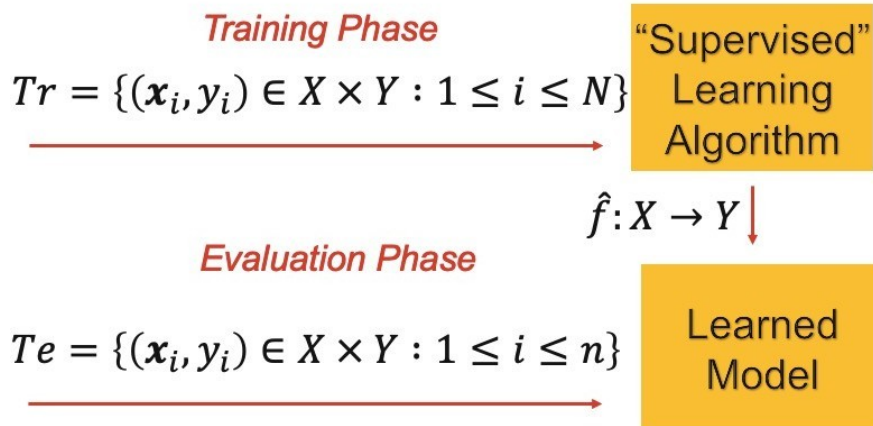
Classification

YearsExperience	Salary
1.1	39343
1.3	46205
1.5	37731
2	43525
2.2	39891
2.9	56642
3	60150
3.2	54445
3.2	64445

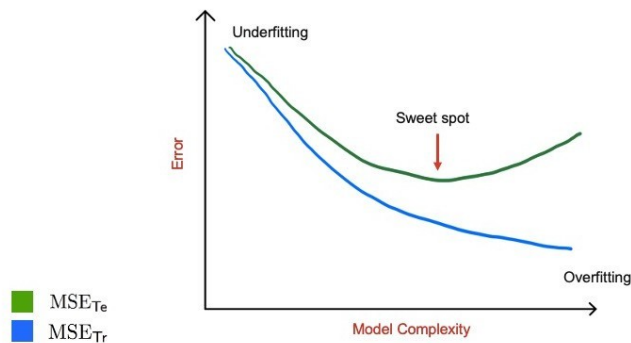
Regression

# Обзор (2)

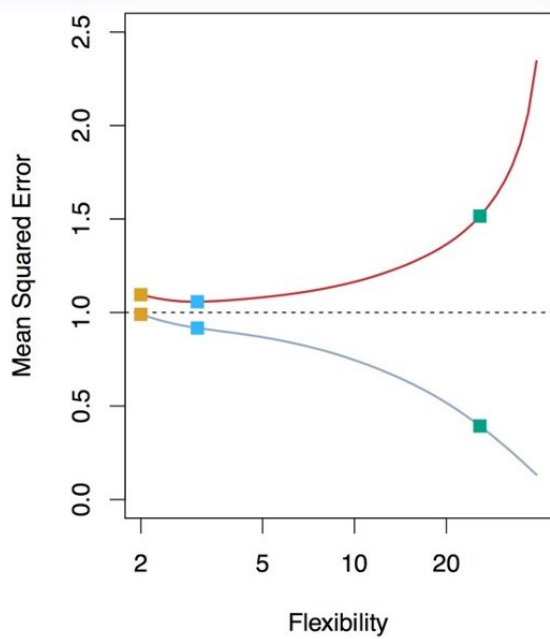
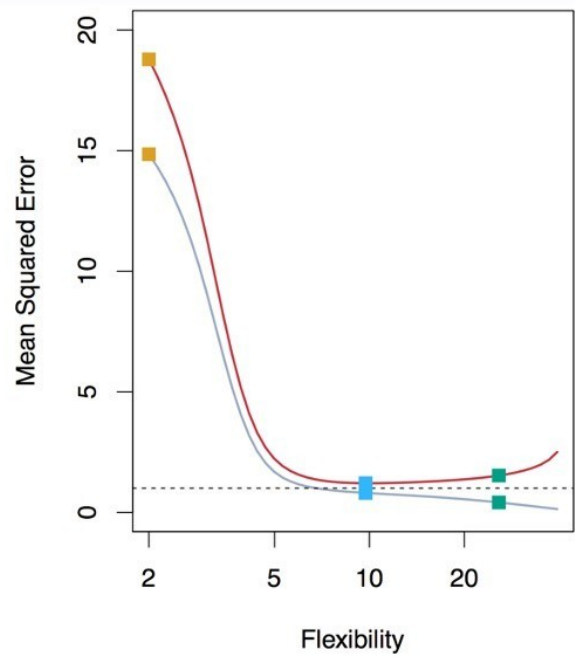
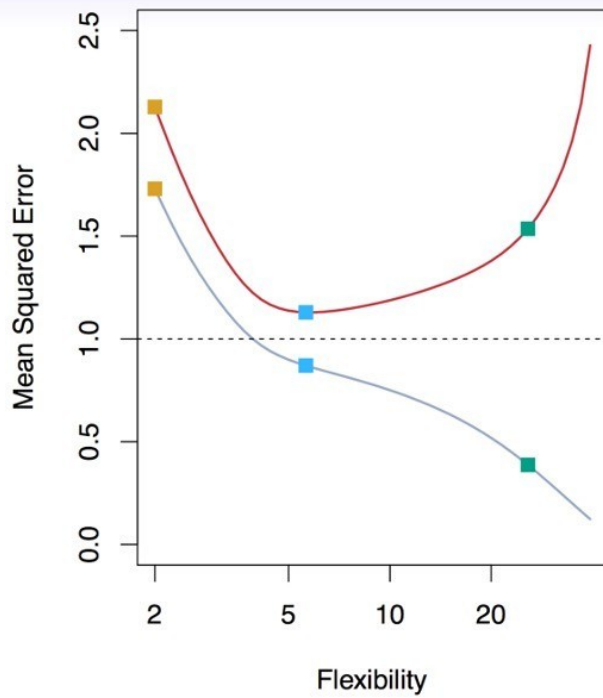
## How do we implement it?



## Model Complexity or Flexibility



# Недооценка и переоценка



# Дефекты программного обеспечения

- Также известен как
  - Жучки
  - Проблемы
  - Ошибка
  - Аномалия
  - ...
- Мы говорим, что программное обеспечение имеет дефекты, если
  - Он делает то, что не должен
  - Он не делает того, что должен
  - ...

# Источники проблем

- Определение требований
- Дизайн
- Реализация
- Неадекватное тестирование
- ...



# Негативные последствия зараженного программного обеспечения

- **Здравоохранение:** гибель людей, утечка данных и т.д.
- **Коммуникации:** Потеря данных и т.д.
- **Защита:** Ошибочное опознание цели и т.д.
- **Электроэнергия:** перебои в подаче электроэнергии  
травмы и т.д.
- **Управление деньгами:**  
мошенничество, закрытие биржи и т.д.
- ...

# Программное обеспечение без ошибок

- Можете ли вы гарантировать, что программные системы, которые вы или ваша команда будете разрабатывать, не будут содержать ошибок?
- Даже если мы будем очень осторожны, все равно очень трудно сделать программное обеспечение без ошибок, потому что
  - *По мере того как программное обеспечение приобретает все больше функций и поддерживает все больше платформ, становится все труднее сделать его без ошибок.*

# Обнаружение против предсказания

- Обнаружение дефектов программного обеспечения
  - Выявление дефектов
  - Исправьте их
- Но обычно ошибки, обнаруженные позже, обходятся дороже.
- Прогнозирование дефектов программного обеспечения
  - Предварительная информация о возможных дефектах
  - .. Количество дефектов ...

# Теперь вы знаете...

1. Что такое ошибки в программном обеспечении?
  2. Каковы их источники?
  3. Каковы их побочные эффекты?
  4. Насколько маловероятно создать программное обеспечение без ошибок?
  5. Насколько важно уметь предсказывать информацию, связанную с дефектами?
- ❖ Теперь давайте посмотрим, как можно предсказать **количество дефектов** в программном обеспечении с помощью машинного обучения.

# Вспомнить: Цель обучения

- Изучение или вывод "функциональной" связи между предикторами и целью

$$D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$$

$$\mathbf{x} \in \mathbb{R}^d$$

$$\hat{f} \approx f \quad \textit{Goal of learning}$$

$$y = f(\mathbf{x})$$

# Прогнозирование количества дефектов с точки зрения ML

Дана компьютерная программа,  
скажем,  $p_i$

1. Каким будет  $x_i$
2. Каким будет  $y_i$

Таким образом, целью обучения является  
оценка следующей функциональной  
зависимости

$$\underbrace{\# \text{ of defects in } p_i}_{y_i} = \underbrace{f(\text{features of } p_i)}_{x_i}$$

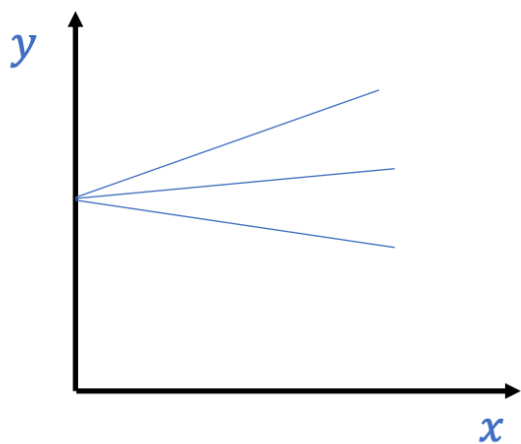
# Уравнение прямой линии

Наклон

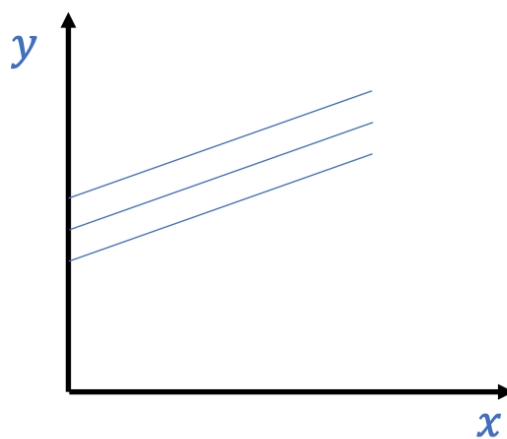
Перехват


$$y = ax + b$$

# Различные наклоны и перехваты



Различные склоны



Различные перехваты



# Вернемся к нашей проблеме регрессии

*# of defects in  $p_i = f(\text{features or behavior of } p)_i$*

- Предположим, что есть только одна функция,
- Тогда мы можем записать приведенное выше выражение в виде

$$y = w_1 x + w_0$$

- Это то же уравнение, что и уравнение прямой линии
- Именно поэтому мы называем ее "простой линейной регрессией".

# В общем, Линейная регрессия

$$y = w_0 + w_1 x_1 + w_2 x_2 + \cdots w_p x_p$$

- Переменная ответа - **количественная**
- Предполагается, что связь между откликом и предикторами **линейна** по входу
- Таким образом, мы ограничиваемся пространством гипотез линейных функций

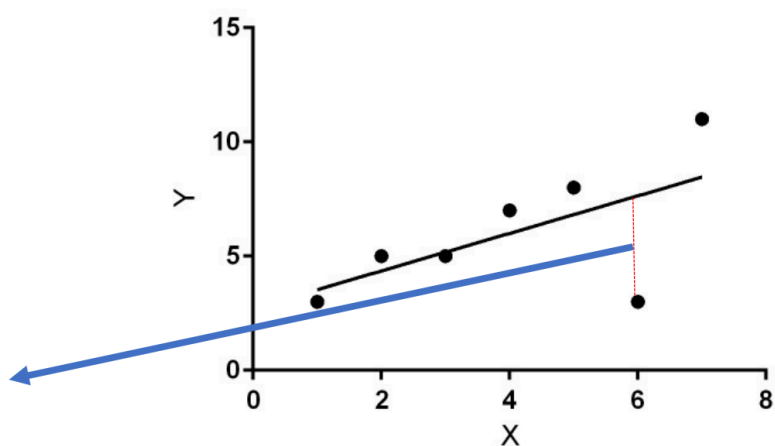
# Зачем нужна линейная регрессия

- Хотя линейная регрессия может показаться слишком упрощенной, она чрезвычайно полезна.
  - Легко делать выводы
  - Служит хорошей отправной точкой для более мощных и сложных подходов

# Как обучить модель линейной регрессии

$$f(x_i) = w_0 + w_1 x_i$$

$$e_i = y_i - f(x_i)$$



# Средняя квадратичная ошибка (MSE)

$$f(x_i) = w_0 + w_1 x_i$$

$$e_i = y_i - f(x_i)$$

$$\mathcal{L}(w_0, w_1) = \frac{1}{n} \sum_{i=1}^n (y_i - (w_0 + w_1 x_i))^2$$

Нам нужно найти  
значение параметров,  
которые  
минимизируют эту  
функцию затрат или  
потерь.

# Объективная функция

$$f(x_i) = w_0 + w_1 x_i$$

$$e_i = y_i - f(x_i)$$

$$\mathcal{L}(w_0, w_1) = \frac{1}{n} \sum_{i=1}^n (y_i - (w_0 + w_1 x_i))^2$$

$$\underset{w_0, w_1}{\operatorname{argmin}} \mathcal{L}(w_0, w_1)$$

Термин **argmin** - это сокращение от "найти аргумент, который минимизирует ...".

# Производные

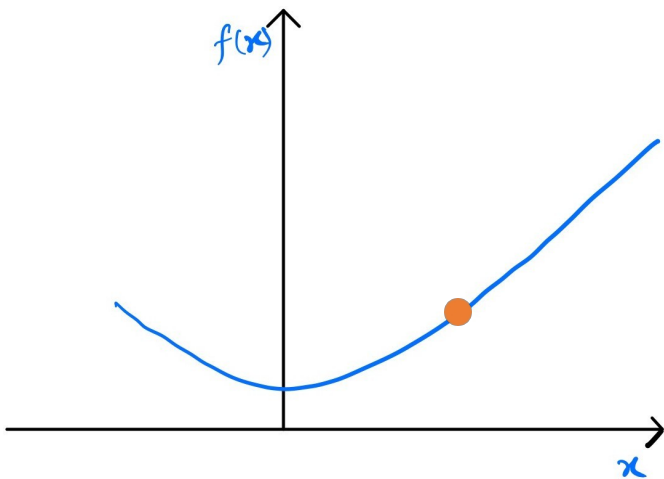
- Производная - сердце исчисления
- Производная функции одной переменной определяется как

$$f'(x) = \lim_{dx \rightarrow 0} \frac{f(x + dx) - f(x)}{dx}$$

- Но вопрос в том, для чего мы можем его использовать?

# Использование Производных

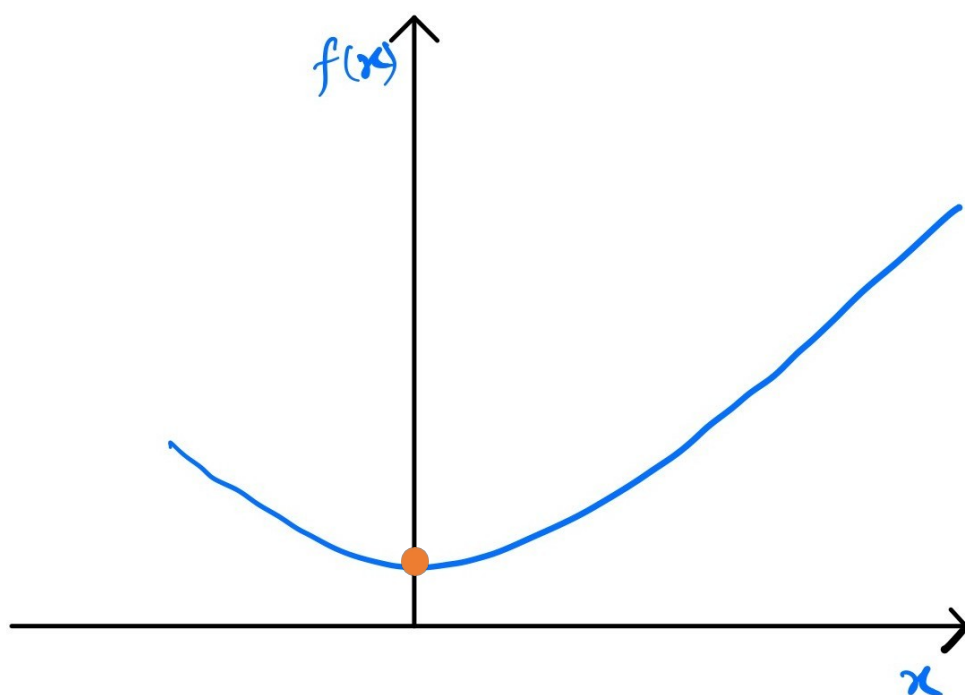
- Допустим, мы стоим в этой точке.
- Мы хотели бы знать, что произойдет, если увеличим  $x$ .
- То есть, увеличится или уменьшится значение функции?



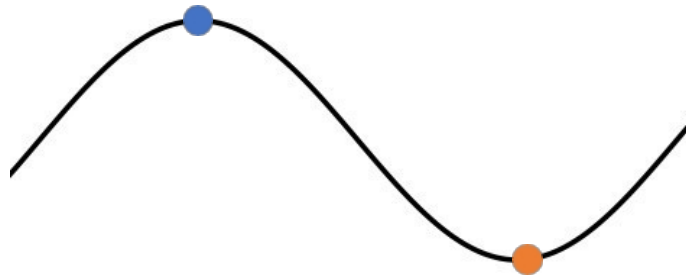
- **Это** можно найти, взяв производную функции в этой точке!



Что будет  $f'(x)$  в этот момент?

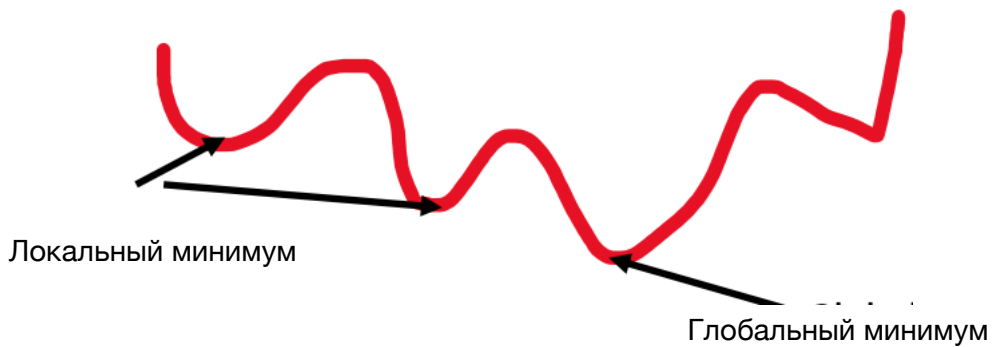


# Максимум и минимум

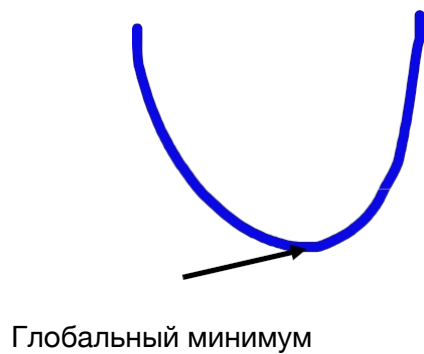


Точка $x$	$\#f$ немного левее $x$ .	$\#f$ at $x$	$\#f$ немного правее $x$ .
Максимальный ●	$> 0$	$0$	$< 0$
Минимум ●	$< 0$	$0$	$> 0$

# Выпуклые и невыпуклые



Множество точек минимума - локальный и глобальный минимум



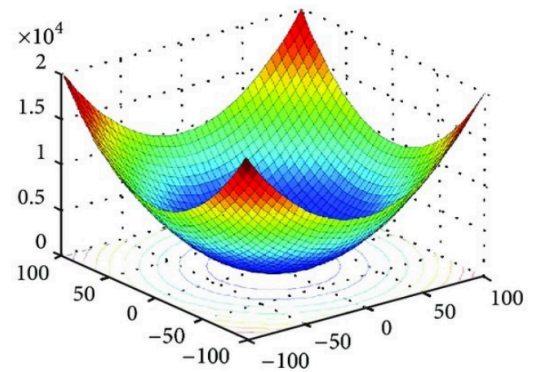
- Уникальный минимум - его глобальный минимум

# Отзыв

1. **Целевая функция** (которую мы хотим оптимизировать)
2. **Производная** (математический инструмент, позволяющий определить, возрастает или убывает функция при незначительном увеличении ее входных данных)
3. **Точки максимума и минимума** (где производная равна 0)
4. **Выпуклые функции** (мы любим их за то, что у них есть только одна глобальная точка минимума, что упрощает их использование в задачах оптимизации)

# Вернёмся к нашей целевой функции

$$\mathcal{L}(w_0, w_1) = \frac{1}{n} \sum_{i=1}^n (y_i - (w_0 + w_1 x_i))^2$$



MSE является  
выпуклой: в  
единственном  
минимуме нашей  
функции потерь ее  
"частичная"  
производная по  
отношению к  $w_0$  и  $w_1$   
будет равна нулю!

# Решение по методу наименьших квадратов

$$\mathcal{L}(w_0, w_1) = \frac{1}{n} \sum_{i=1}^n (y_i - (w_0 + w_1 x_i))^2$$

1. Вычислите частные производные функции потерь по  $w_0$  и  $w_1$
2. Установите их на 0
3. И решите для  $w_0$  и  $w_1$

# Решение по методу наименьших квадратов (2)

$$\mathcal{L}(w_0, w_1) = \frac{1}{n} \sum_{i=1}^n (w_1^2 x_i^2 + 2w_1 x_i (w_0 - y_i) + w_0^2 - 2w_0 y_i + y_i^2)$$

- Возьмем частные производные функции потерь по отношению к  $w_0$ ,
- Мы можем начать с удаления терминов, которые не включают  $w_0$

$$\frac{1}{n} \sum_{i=1}^n (w_0^2 + 2w_1 x_i w_0 - 2w_0 y_i)$$

# Решение по методу наименьших квадратов (3)

$$\frac{1}{n} \sum_{i=1}^n (w_0^2 + 2w_1 x_i w_0 - 2w_0 y_i)$$

- Переставьте члены, не имеющие индекса  $n$ , пределы суммирования,

$$= w_0^2 + 2w_1 w_0 \frac{1}{n} \left( \sum_{i=1}^n x_i \right) - 2w_0 \frac{1}{n} \left( \sum_{i=1}^n y_i \right)$$



### Basic Properties and Formulas

If  $f(x)$  and  $g(x)$  are differentiable functions (the derivative exists),  $c$  and  $n$  are any real numbers,

$$1. (cf)' = cf'(x)$$

$$5. \frac{d}{dx}(c) = 0$$

$$2. (f \pm g)' = f'(x) \pm g'(x)$$

$$6. \frac{d}{dx}(x^n) = nx^{n-1} - \text{Power Rule}$$

$$3. (fg)' = f'g + fg' - \text{Product Rule}$$

$$7. \frac{d}{dx}(f(g(x))) = f'(g(x))g'(x)$$

$$4. \left(\frac{f}{g}\right)' = \frac{f'g - fg'}{g^2} - \text{Quotient Rule}$$

This is the **Chain Rule**

### Common Derivatives

$$\frac{d}{dx}(x) = 1$$

$$\frac{d}{dx}(\csc x) = -\csc x \cot x$$

$$\frac{d}{dx}(a^x) = a^x \ln(a)$$

$$\frac{d}{dx}(\sin x) = \cos x$$

$$\frac{d}{dx}(\cot x) = -\csc^2 x$$

$$\frac{d}{dx}(e^x) = e^x$$

$$\frac{d}{dx}(\cos x) = -\sin x$$

$$\frac{d}{dx}(\sin^{-1} x) = \frac{1}{\sqrt{1-x^2}}$$

$$\frac{d}{dx}(\ln(x)) = \frac{1}{x}, \quad x > 0$$

$$\frac{d}{dx}(\tan x) = \sec^2 x$$

$$\frac{d}{dx}(\cos^{-1} x) = -\frac{1}{\sqrt{1-x^2}}$$

$$\frac{d}{dx}(\ln|x|) = \frac{1}{x}, \quad x \neq 0$$

$$\frac{d}{dx}(\sec x) = \sec x \tan x$$

$$\frac{d}{dx}(\tan^{-1} x) = \frac{1}{1+x^2}$$

$$\frac{d}{dx}(\log_a(x)) = \frac{1}{x \ln a}, \quad x > 0$$

# Решение по методу наименьших квадратов (4)

$$w_0^2 + 2w_1w_0 \frac{1}{n} \left( \sum_{i=1}^n x_i \right) - 2w_0 \frac{1}{n} \left( \sum_{i=1}^n y_i \right)$$

- Возьмем частную производную по отношению к  $w_0$ ,

$$\frac{\partial \mathcal{L}}{\partial w_0} = 2w_0 + 2w_1 \frac{1}{n} \left( \sum_{i=1}^n x_i \right) - 2 \frac{1}{n} \left( \sum_{i=1}^n y_i \right)$$

# Решение по методу наименьших квадратов (5)

$$\frac{\partial \mathcal{L}}{\partial w_0} = 2w_0 + 2w_1 \frac{1}{n} \left( \sum_{i=1}^n x_i \right) - 2 \frac{1}{n} \left( \sum_{i=1}^n y_i \right)$$

- Теперь приравняйте частную производную к нулю,

$$2w_0 + 2w_1 \frac{1}{n} \left( \sum_{i=1}^n x_i \right) - 2 \frac{1}{n} \left( \sum_{i=1}^n y_i \right) = 0$$

$$2w_0 = 2 \frac{1}{n} \left( \sum_{i=1}^n y_i \right) - 2w_1 \frac{1}{n} \left( \sum_{i=1}^n x_i \right)$$

# Решение по методу наименьших квадратов (6)

$$2w_0 = 2 \frac{1}{n} \left( \sum_{i=1}^n y_i \right) - 2w_1 \frac{1}{n} \left( \sum_{i=1}^n x_i \right)$$

$$w_0 = \frac{1}{n} \left( \sum_{i=1}^n y_i \right) - w_1 \frac{1}{n} \left( \sum_{i=1}^n x_i \right)$$

$$w_0 = \bar{y} - w_1 \bar{x}$$

# Решение по методу наименьших квадратов (7)

$$w_0 = \bar{y} - w_1 \bar{x}$$

- Теперь мы должны проделать тот же процесс для  $w_1$

# Решение по методу наименьших квадратов (8)

$$\mathcal{L}(w_0, w_1) = \frac{1}{n} \sum_{i=1}^n (w_1^2 x_i^2 + 2w_1 x_i (w_0 - y_i) + w_0^2 - 2w_0 y_i + y_i^2)$$

- Теперь  
возьмем  
частные  
производные  
функции  
потерь по  
отношению к  
 $w_1$ ,
- Мы можем начать с  
удаления терминов, которые  
невключают  $w_1$

$$\frac{1}{n} \sum_{i=1}^n (w_1^2 x_i^2 + 2w_1 x_i w_0 - 2w_1 x_i y_i)$$

You must do this yourself!

# The Least Square Solution (Summary)

$$\mathcal{L}(w_0, w_1) = \frac{1}{n} \sum_{i=1}^n (y_i - (w_0 + w_1 x_i))^2$$

$$w_0 = \bar{y} - w_1 \bar{x}$$

$$w_1 = \frac{\overline{xy} - \bar{x} \bar{y}}{\overline{x^2} - (\bar{x})^2}$$

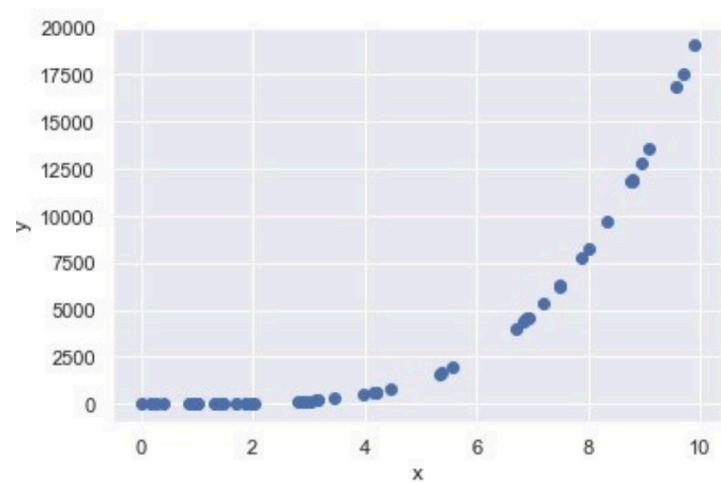
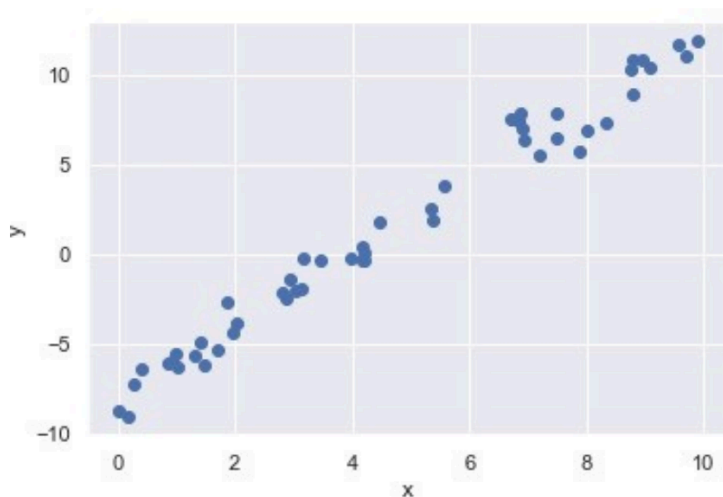
# Альтернативы

- Вы только что узнали, как оценить параметры LR с помощью метода наименьших квадратов
- Но есть и другие способы сделать это, особенно если мы имеем дело с данными, которые не помещаются в памяти
- Одним из таких и очень важных методов является *градиентный спуск*



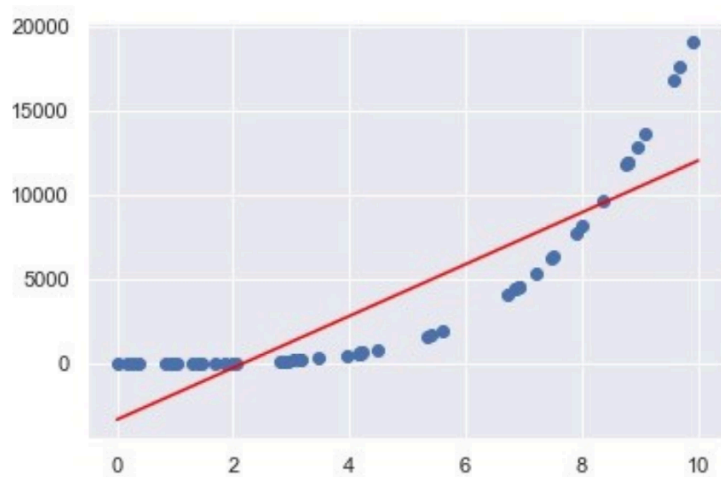
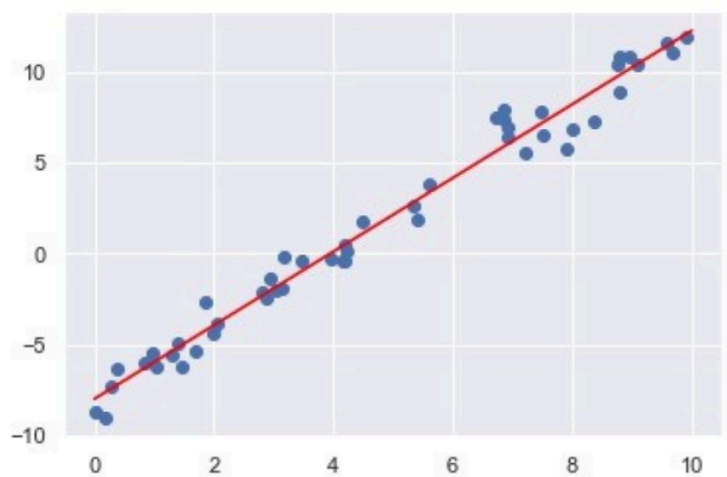
# Расширение линейной регрессии

Нелинейная связь между предикатом и ответом



•

## Нелинейная связь между предикатом и ответом (2)



# Полиномиальная регрессия

- Используя ту же схему, которую мы изучили, можно подогнать семейство более сложных моделей с помощью преобразования предикторов

- Линейная модель имеет следующий вид

$$y = w_0 + w_1 x$$

- Она линейна как по предиктору ( $x$ ), так и по параметрам ( $w_0, w$ ).<sup>1</sup>
- Давайте оставим его линейным по параметрам, но сделаем квадратичным по предикторам

# Полиномиальная регрессия (2)

- Это

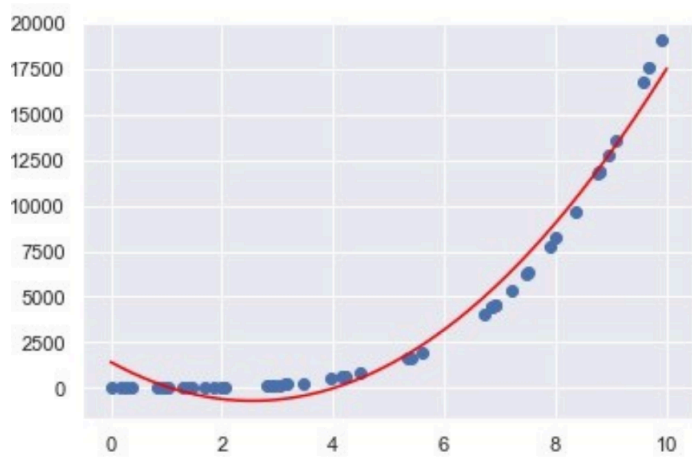
$$y = w_0 + w_1 x + w_2 x^2$$

- В  
целом,

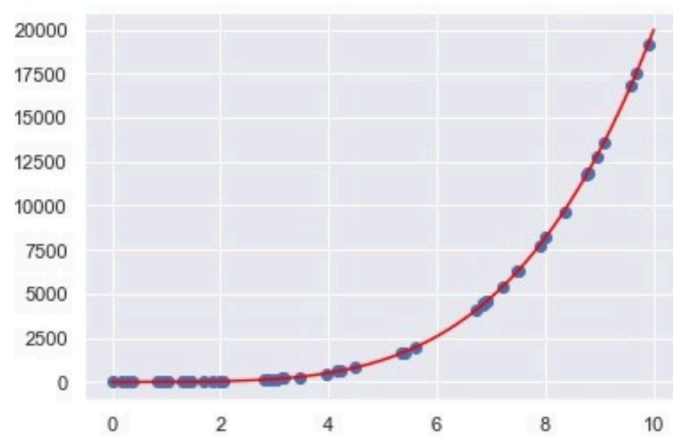
$$y = w_0 + w_1 x + w_2 x^2 + \dots + w_d x^d$$

- Не забывайте, что "модель по-прежнему линейна по параметрам".

# Полиномиальная регрессия (3)

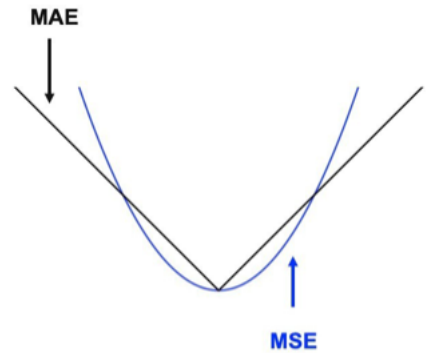


Порядок или степень



Порядок или степень 4

# MSE против MAE



- **MAE**
  - **Хорошо:**  
когда в данных могут быть отклонения
  - **Плохо:**  
потому что его производная везде одинакова
- **MSE**
  - **Хорошо:**  
градиент велик при больших потерях и уменьшается по мере приближения потерь к 0
  - **Плохо:**  
когда в данных есть отклонения
- **Тогда существует функция потерь Губера**
  - **Хорошо в двух случаях**
    - но имеет дополнительный гиперпараметр, требующий перебора.

# Резюме

- Важность прогнозирования (количества) дефектов в программном обеспечении
- Проанализируйте задачу с точки зрения ML - чтобы убедиться, что это задача регрессии
- Формулировка цели обучения
- Решение задачи
  - Решение по методу наименьших квадратов
- Следующая лекция:
  - Градиентный спуск

# Отражение

- Почему прогнозирование количества дефектов в программном обеспечении является проблемой регрессии?
- Что означают слова "простая", "линейная" и "регрессия" в книге "Простая линейная регрессия"?
- Почему мы хотим минимизировать среднюю квадратичную ошибку (MSE) относительно параметров нашей регрессионной модели?
- Какова роль производной в этой задаче минимизации?
- Что такое выпуклые функции и почему мы рады, что MSE - выпуклая функция?
- Что такое нелинейная регрессионная задача?
- Почему линейная регрессия не подходит для решения нелинейных задач?
- Что такое полиномиальная функция и как она может помочь нам в решении задач нелинейной регрессии?