«Градиентный спуск»

ML Professional

Цели вебинара | После занятия вы узнаете

1. Recap: линейная регрессия
2. Градиентный спуск
3. Регуляризация
4. Применение на практике

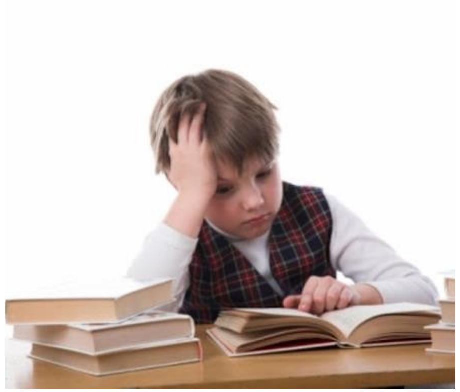
Задачи машинного обучения

С учителем
Supervised learning

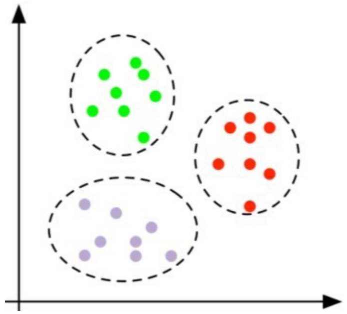
Классификация

Регрессия

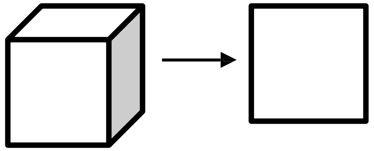
Без учителя
Unsupervised learning



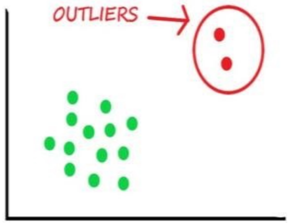
Кластеризация



Снижение размерности



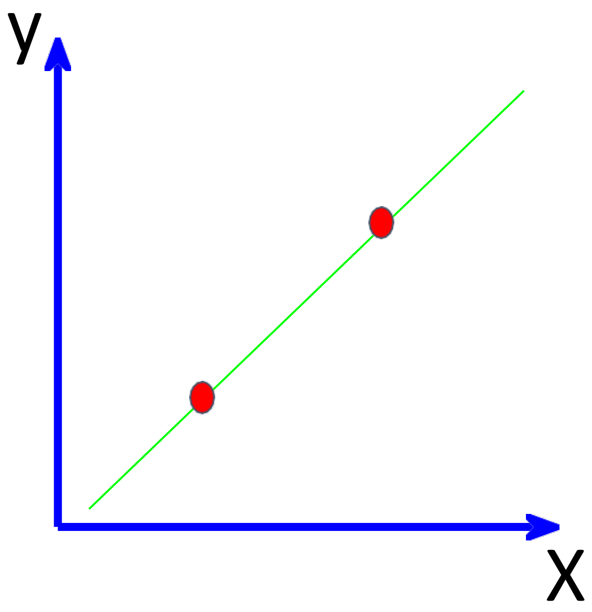
Поиск аномалий



Recap: линейная регрессия

Линейная регрессия

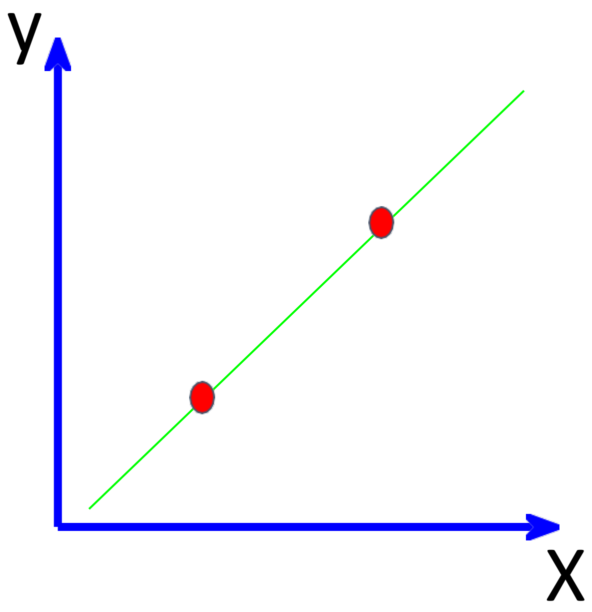
Через две точки на плоскости можно легко провести прямую и тольку одну



$$y = \omega_0 + \omega_1 x$$

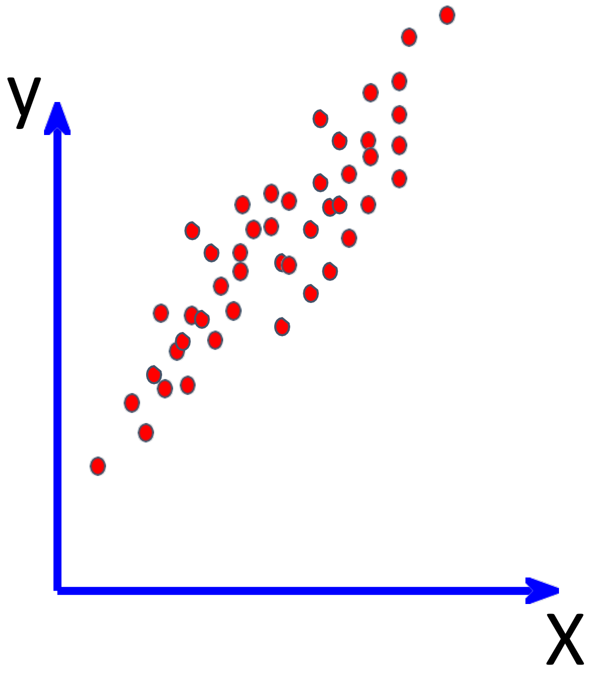
Линейная регрессия

Через две точки на плоскости можно легко провести прямую и тольку одну



$$y = \omega_0 + \omega_1 x$$

А если точек три или более?

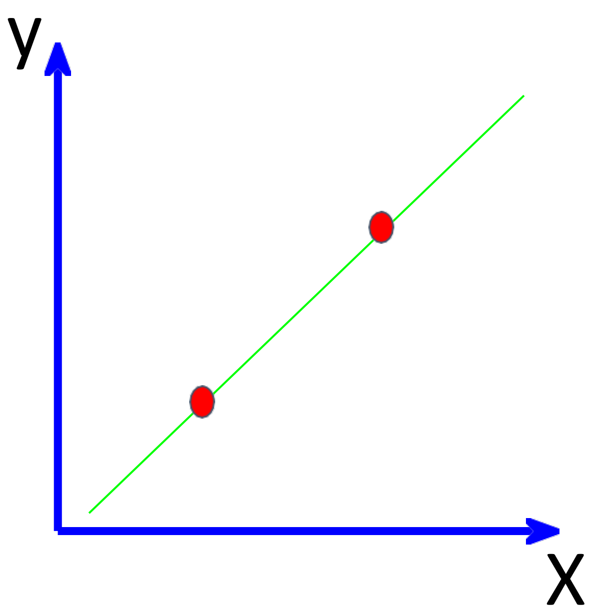


Линейная регрессия

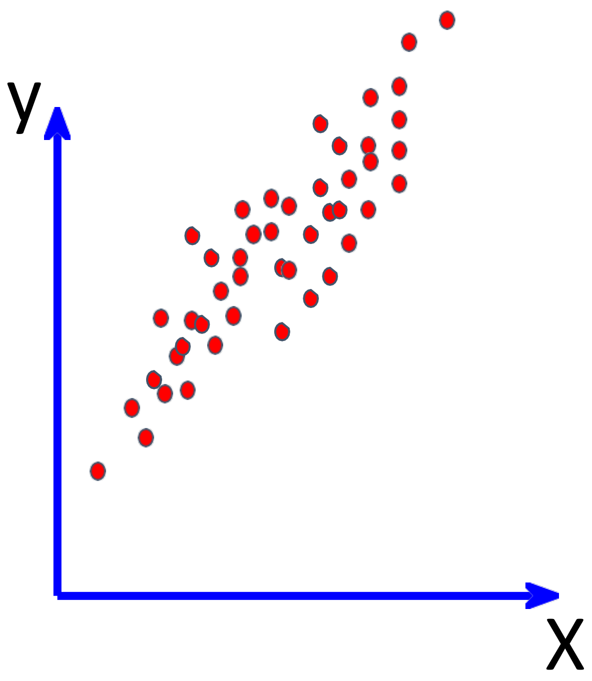
Через две точки на плоскости можно легко провести прямую и только одну

А если точек три или более?

$$y = \omega_0 + \omega_1 x + \epsilon$$



$$y = \omega_0 + \omega_1 x$$

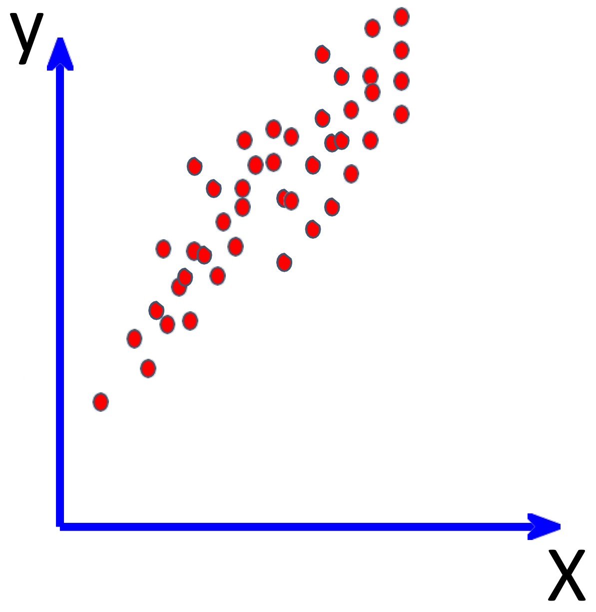


Линейная регрессия

Картина мира:

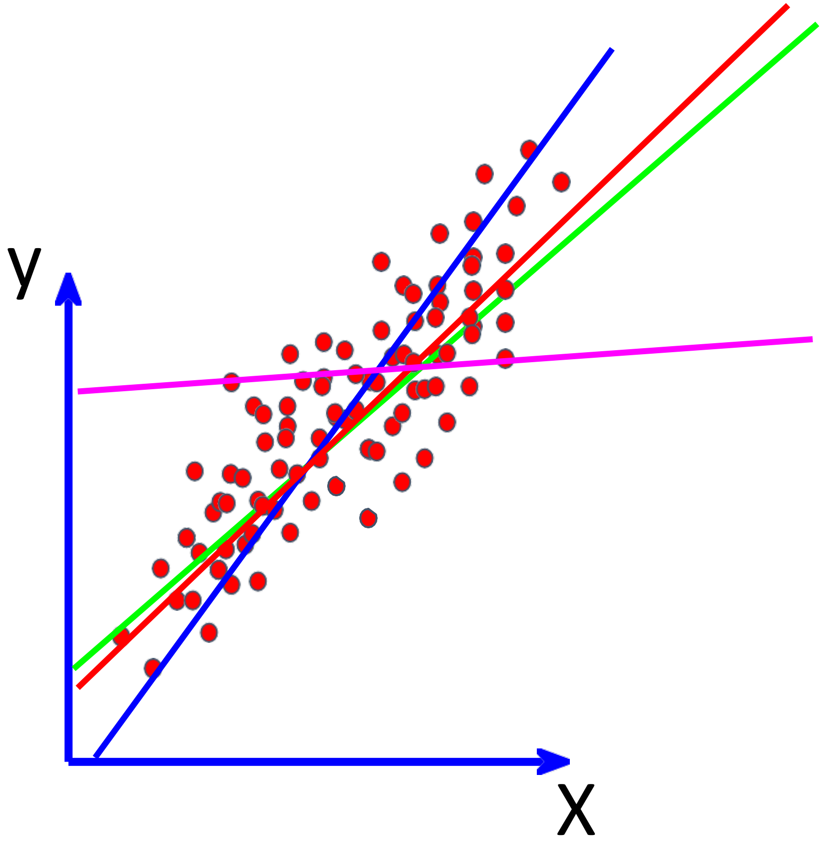
$$y = \omega_0 + \omega_1 x + \epsilon$$

- матожидание случайных ошибок равно нулю: $\forall i : \mathbb{E}\left[\epsilon_i\right] = 0;$
- дисперсия случайных ошибок одинакова и конечна, это свойство называется гомоскедастичностью: $\forall i : \mathrm{Var}\left(\epsilon_i\right) = \sigma^2 < \infty;$
- случайные ошибки не скоррелированы: $\forall i \neq j : \mathrm{Cov}\left(\epsilon_i, \epsilon_j\right) = 0.$



Линейная регрессия

А если точек три или более?

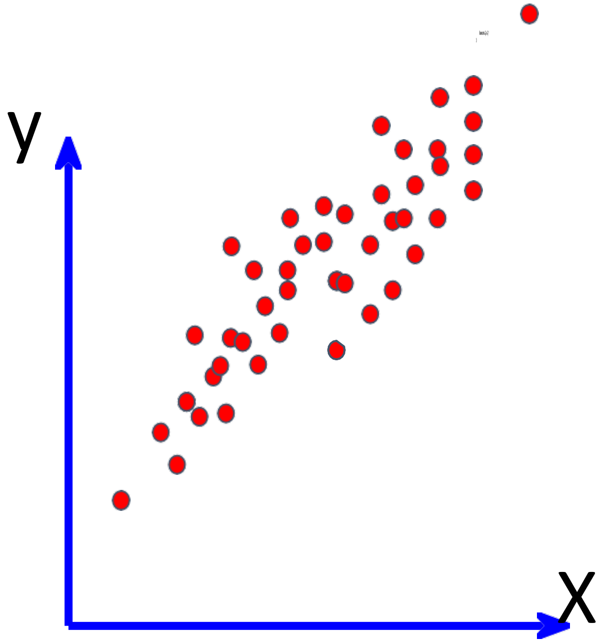


Можно конечно попробовать провести ее руками, опираясь зрительно.

Но как понять, что вы проверили ее хорошо?

Линейная регрессия

А если точек три или более?



Можно сделать умнее.
Пусть зависимость линейная,
тогда прямая будет иметь вид:

$$\hat{y} = \hat{\omega}_0 + \hat{\omega}_1 x$$

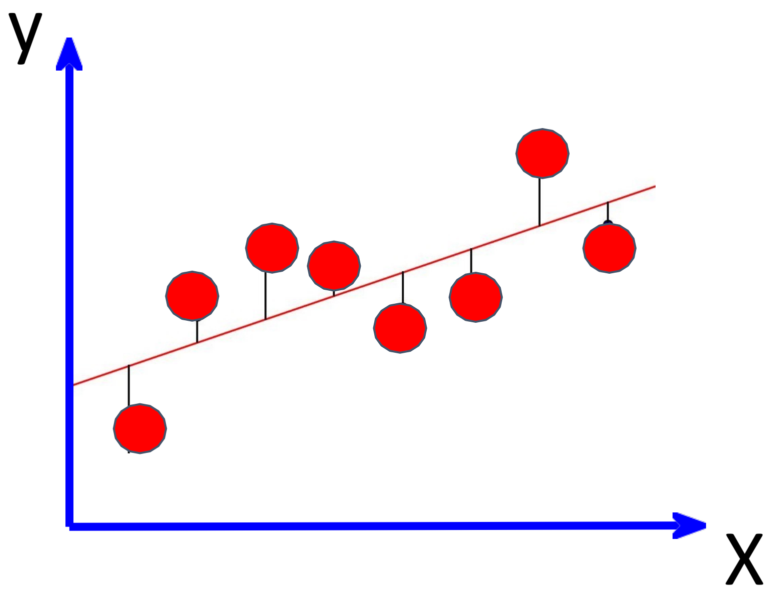
Какие взять $\widehat{\omega_0}$ и $\widehat{\omega_1}$?

Линейная регрессия

Нам нужен какой-то показатель качества проведенной прямой

Метод наименьших квадратов

$$\mathcal{L}(y, \hat{y}) = \sum_{i=1}^{n} (y_i - \hat{y}_i) \rightarrow min$$

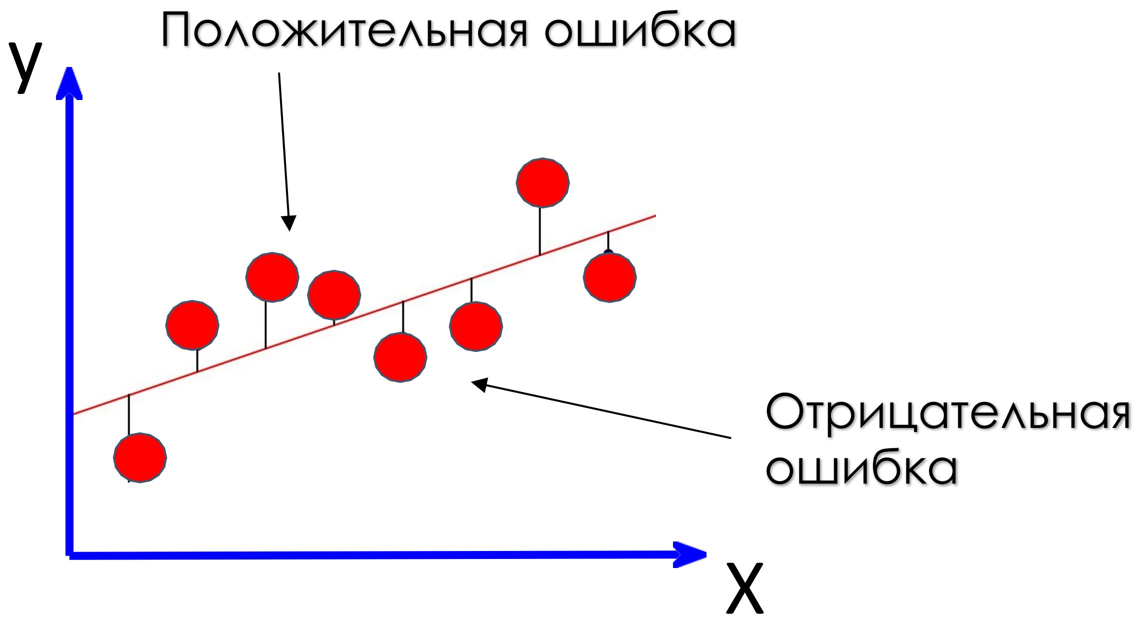


В чем проблема с такой функцией ошибок?

Метод наименьших квадратов

$$\mathcal{L}(y, \hat{y}) = \sum_{i=1}^{n}(y_i - \hat{y}_i) \rightarrow min$$

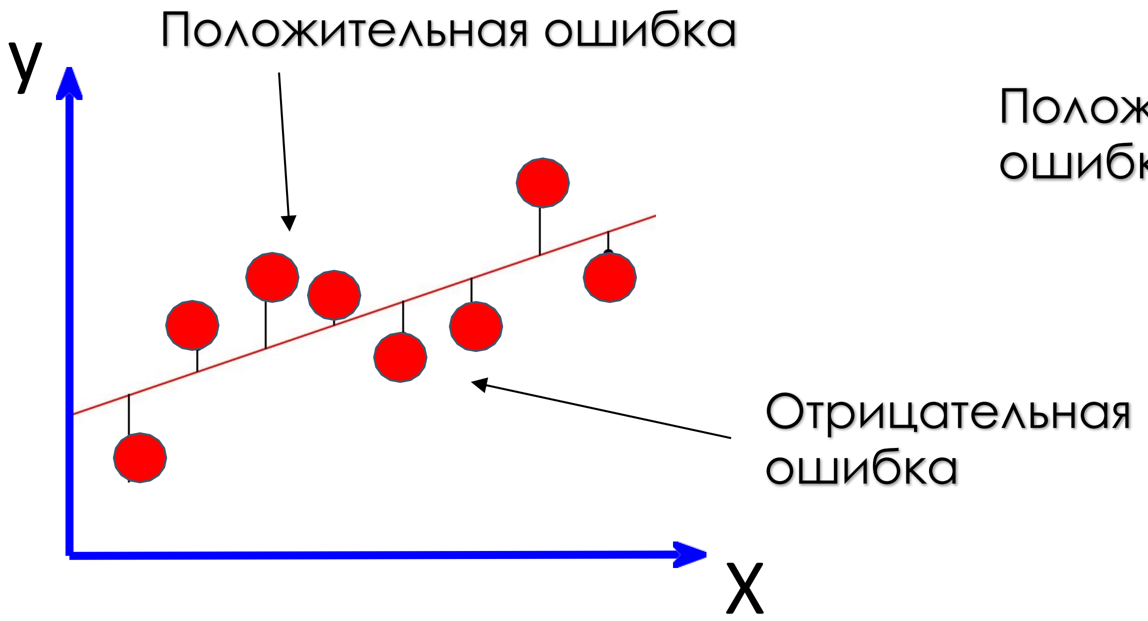
В чем проблема с такой функцией ошибок?



Метод наименьших квадратов

$$\mathcal{L}(y, \hat{y}) = \sum_{i=1}^{n}(y_i - \hat{y}_i) \rightarrow min$$

В чем проблема с такой функцией ошибок?



Положительная ошибка + Отрицательная ошибка ≈ 0

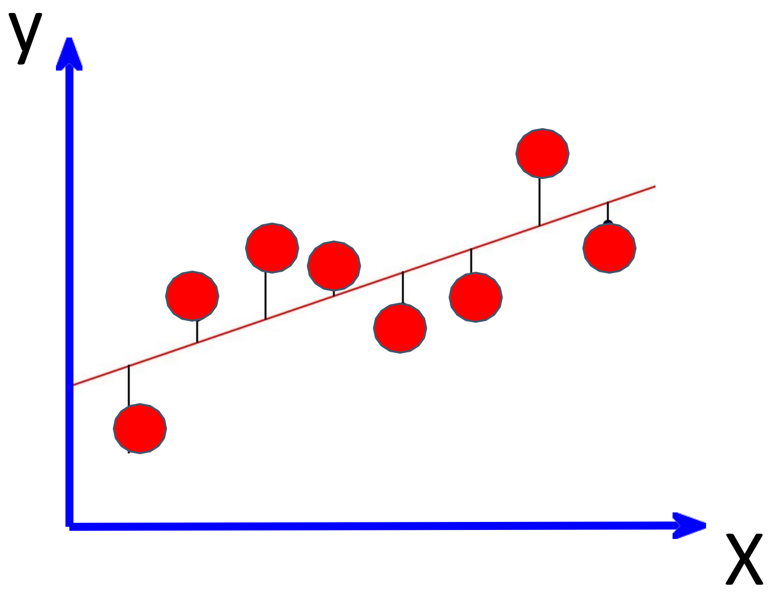
Метод наименьших квадратов

Идея: брать квадраты!

Метод наименьших квадратов

$$RSS = \mathcal{L}(y, \hat{y}) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \rightarrow min$$

RSS - Residual Sum of Squares (сумма квадратов ошибок)

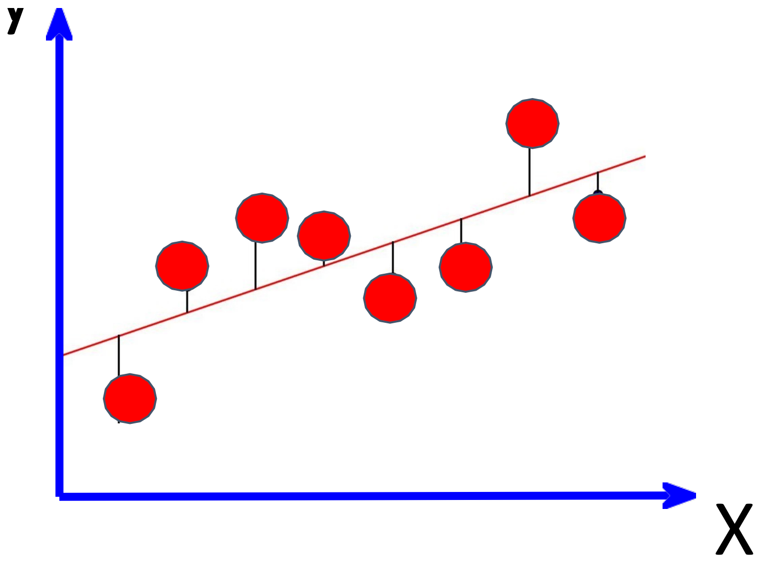


Простыми словами - это сумма квадратов разности **между точкой и значением проведенной прямой**

Метод наименьших квадратов

$$RSS = \mathcal{L}(y, \hat{y}) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \rightarrow min$$

$$RSS = \mathcal{L}(y, \hat{y}) = \sum_{i=1}^{n}(y_i - (\hat{\omega}_0 + \hat{\omega}_1 x))^2 \rightarrow min$$

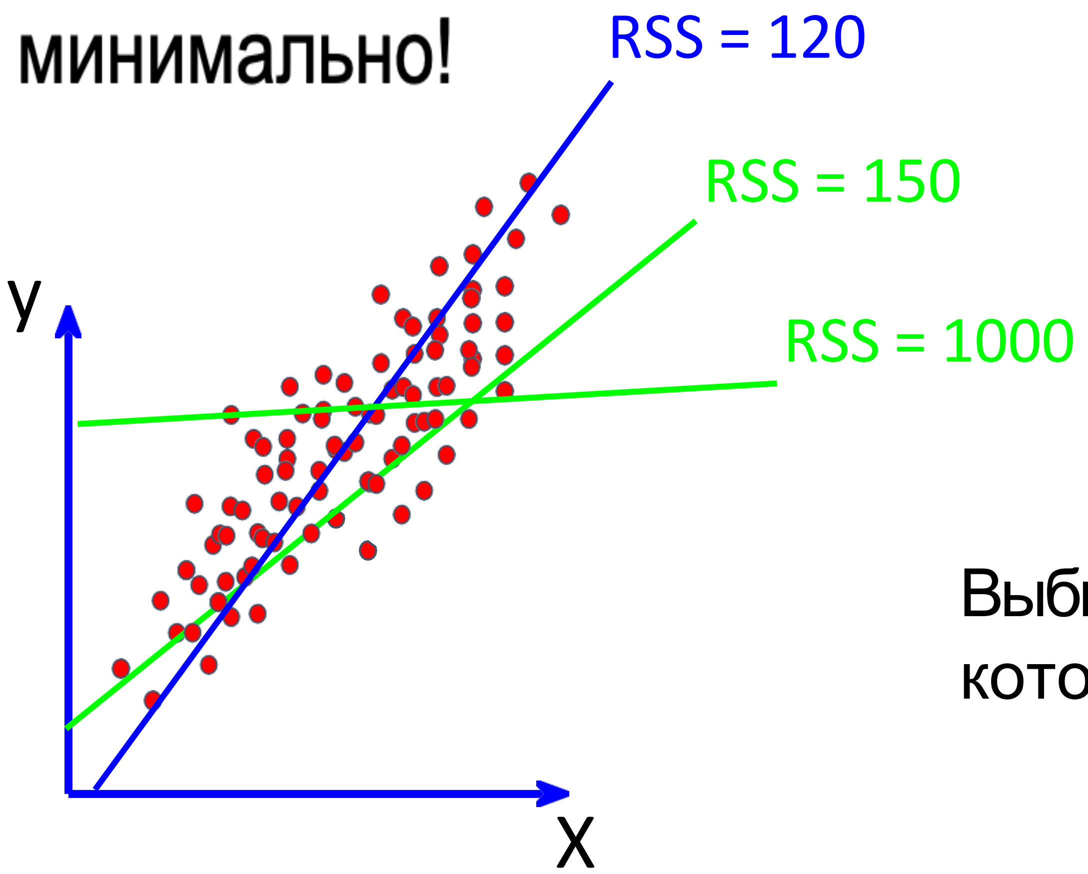


RSS - Residual Sum of Squares (сумма квадратов ошибок)

Простыми словами - это сумма квадратов разности **между точкой и значением проведенной прямой**

Метод наименьших квадратов

Выбираем $\widehat{\omega_0}$ и $\widehat{\omega_1}$, для которых RSS минимально!



Выбираем $\widehat{\omega_0}$ и $\widehat{\omega_1}$, для которых RSS минимально!

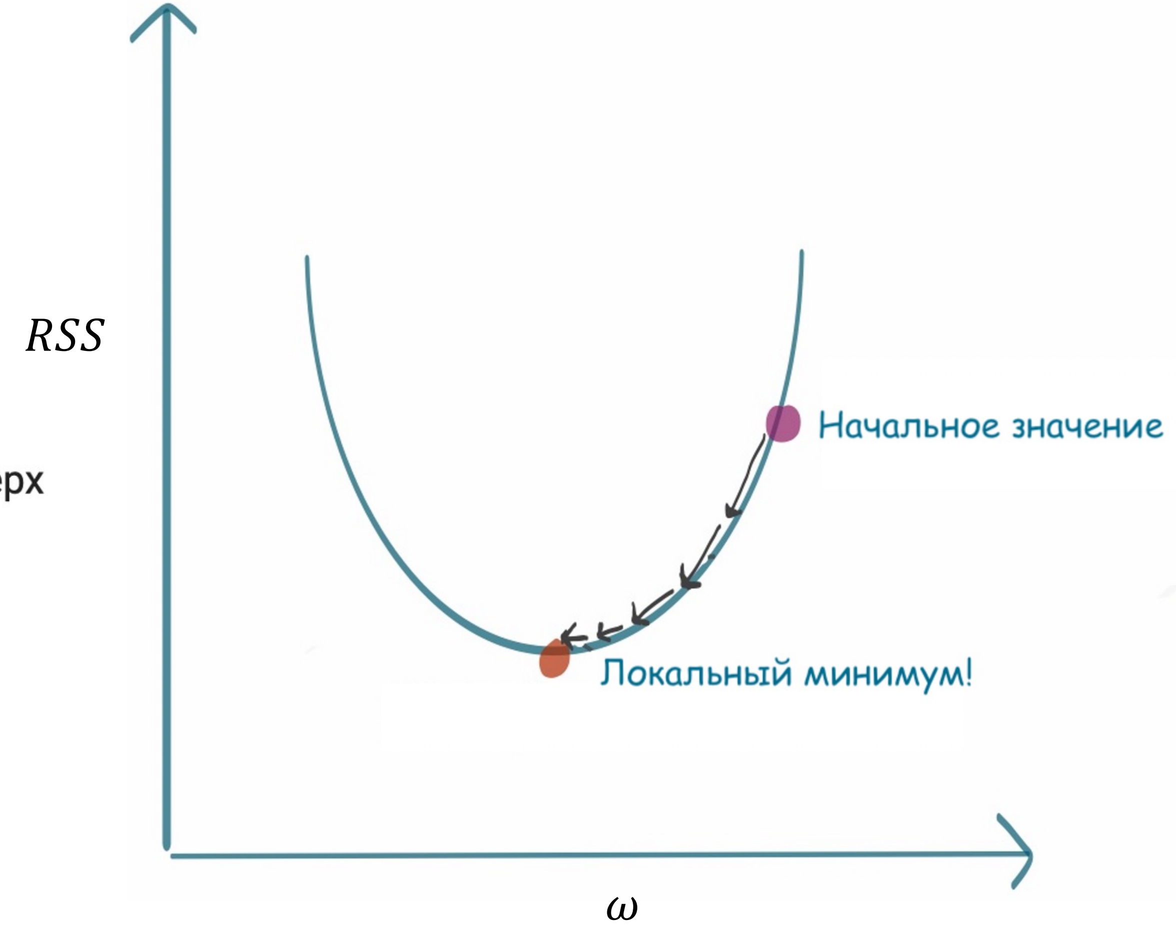
Градиентный спуск

Оптимизируем RSS **градиентным спуском**

$$RSS = \mathcal{L}(y, \hat{y}) = \sum_{i=1}^{n}(y_i - (\hat{\omega}_0 + \hat{\omega}_1 x))^2 \rightarrow min$$

$$RSS = \mathcal{L}(y, \hat{y}) = \sum_{i=1}^{n}((\hat{\omega}_0 + \hat{\omega}_1 x) - y_i)^2 \rightarrow min$$

Квадратичная функция от ω_0 и ω_1 - парабола ветвями вверх



Градиентный спуск

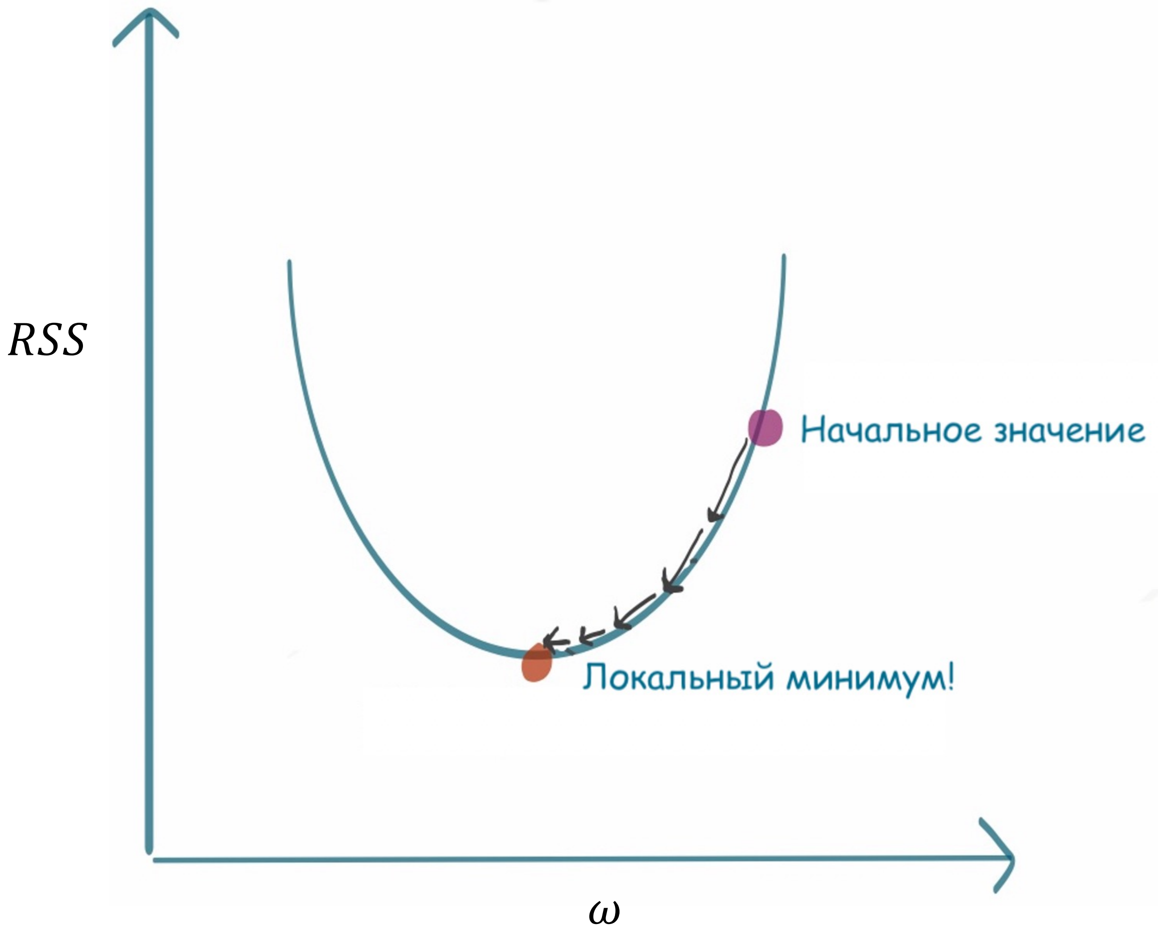
Оптимизируем RSS **градиентным спуском**

$$RSS = \mathcal{L}(y, \hat{y}) = \sum_{i=1}^{n}(y_i - (\hat{\omega}_0 + \hat{\omega}_1 x))^2 \to min$$

$$RSS = \mathcal{L}(y, \hat{y}) = \sum_{i=1}^{n}((\hat{\omega}_0 + \hat{\omega}_1 x) - y_i)^2 \to min$$

Квадратичная функция от ω_0 и ω_1 - парабола ветвями вверх

$$w^{(t)} = w^{(t-1)} - \eta L'(w^{(t-1)})$$



Градиентный спуск

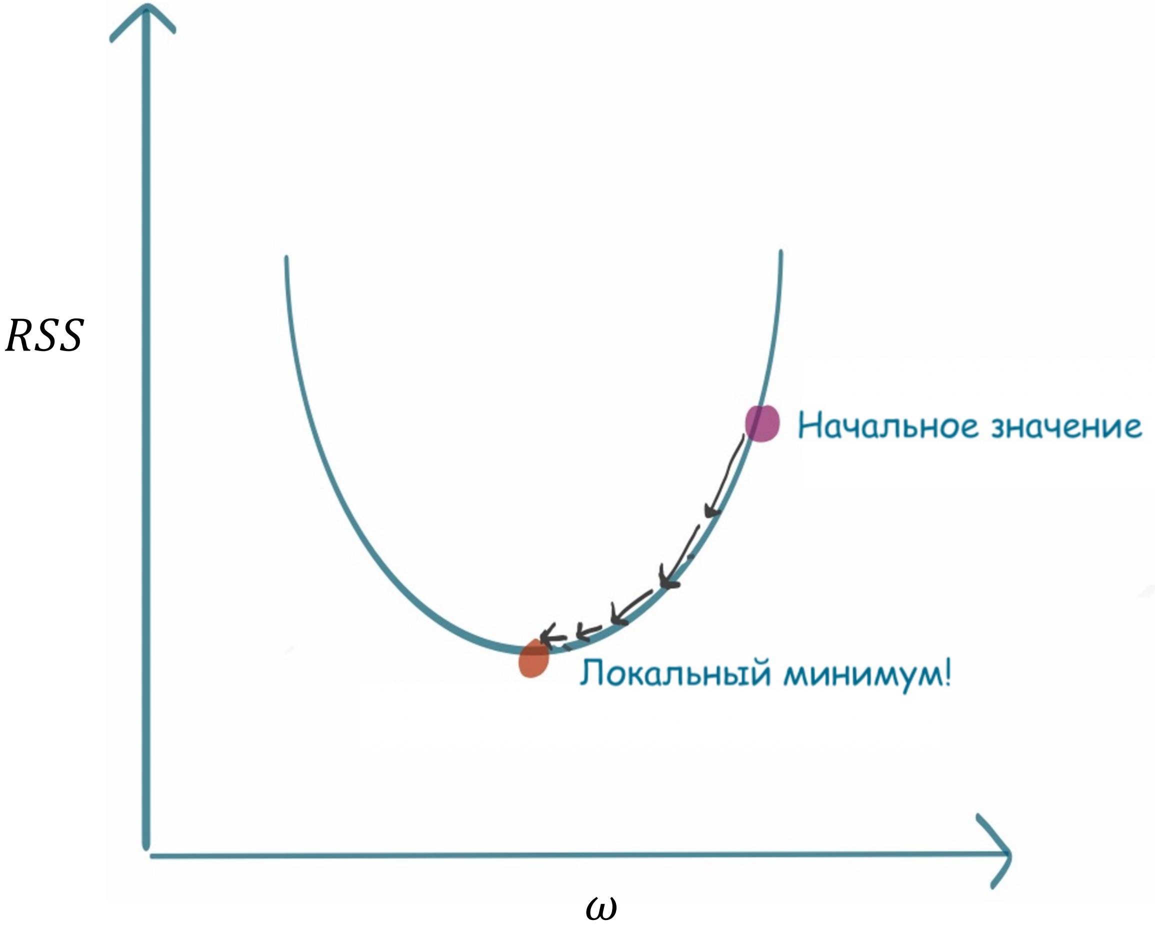
Оптимизируем RSS **градиентным спуском**

$$RSS = \mathcal{L}(y, \hat{y}) = \sum_{i=1}^{n}(y_i - (\hat{\omega}_0 + \hat{\omega}_1 x))^2 \rightarrow min$$

$$RSS = \mathcal{L}(y, \hat{y}) = \sum_{i=1}^{n}((\hat{\omega}_0 + \hat{\omega}_1 x) - y_i)^2 \rightarrow min$$

Квадратичная функция от ω_0 и ω_1 - парабола ветвями вверх

$$w^{(t)} = w^{(t-1)} - \eta \nabla L(w^{(t-1)})$$

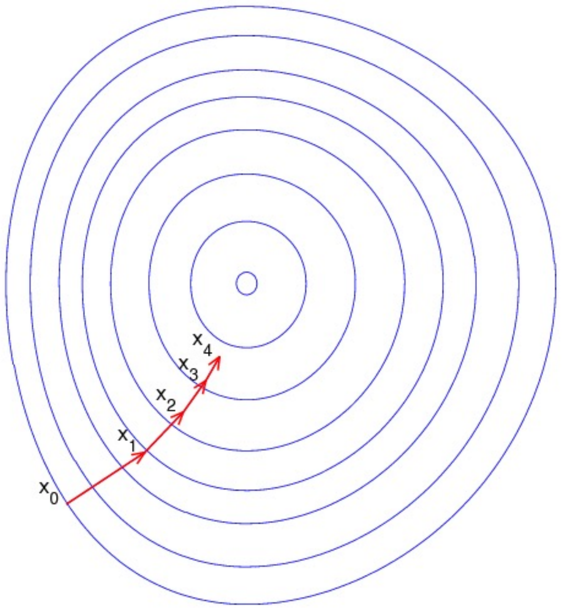


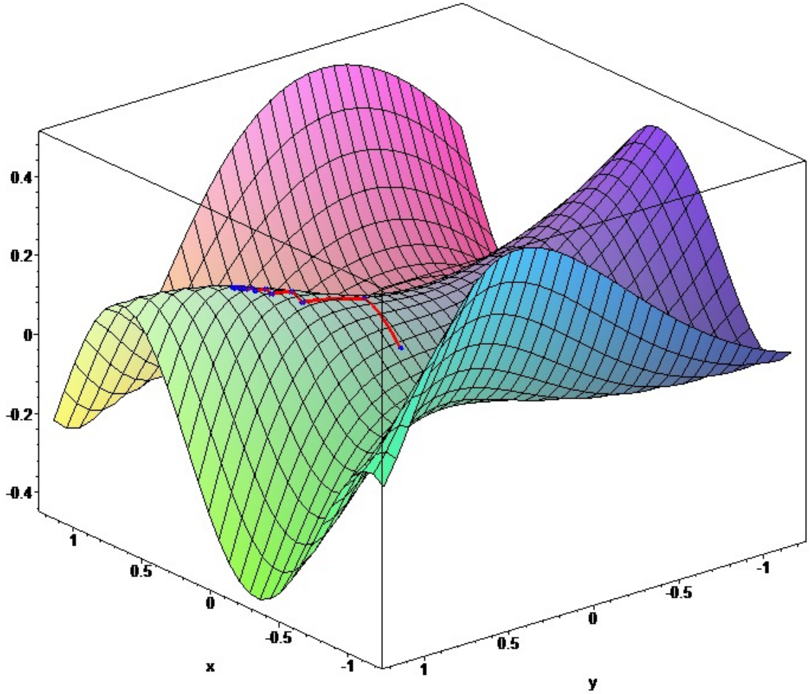
Градиентный спуск

Градиентный спуск работает и когда признаков много!

$$\hat{y} = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \ldots + \omega_p x_p = x^T \omega$$

$$w^{(t)} = w^{(t-1)} - \eta \nabla L(w^{(t-1)})$$





Метрики качества

Метрики качества

Средняя абсолютная ошибка

$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

Средняя квадратичная ошибка

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

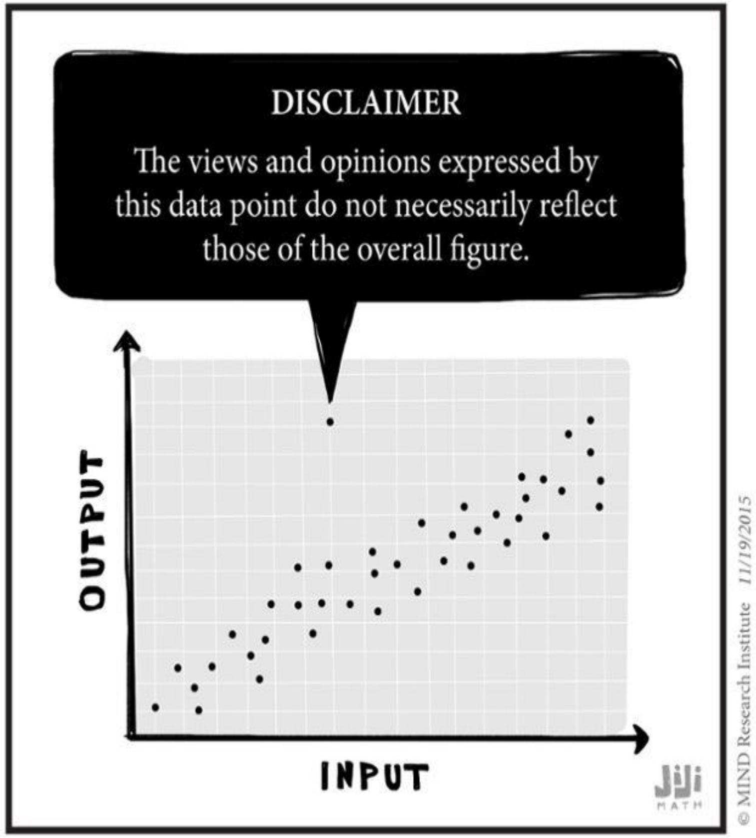
Корень из средней квадратичной ошибки

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$


python™

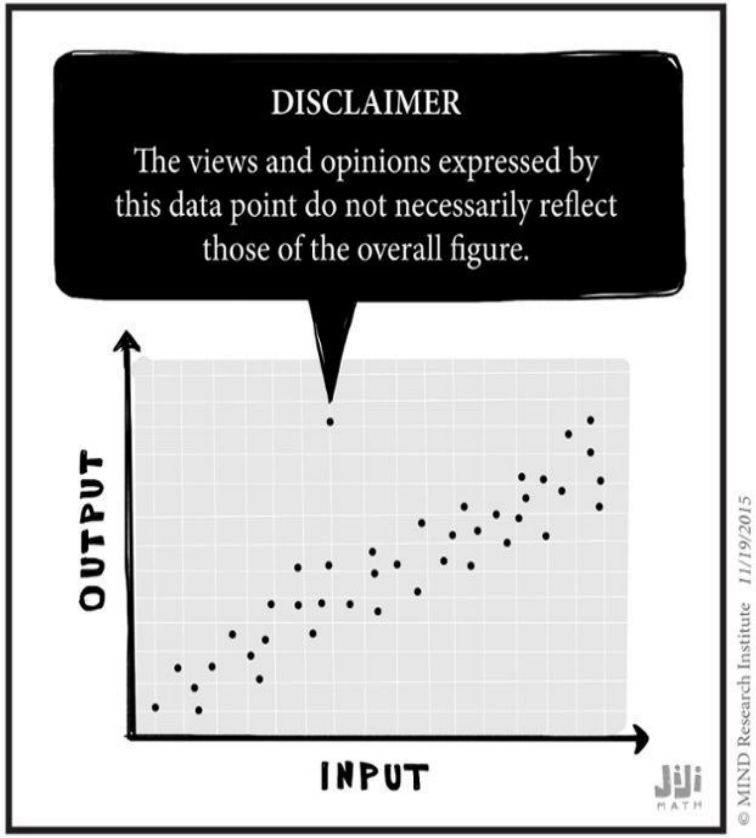
Регуляризация

Регуляризация



Проблема: выбросы и переобучение

Регуляризация



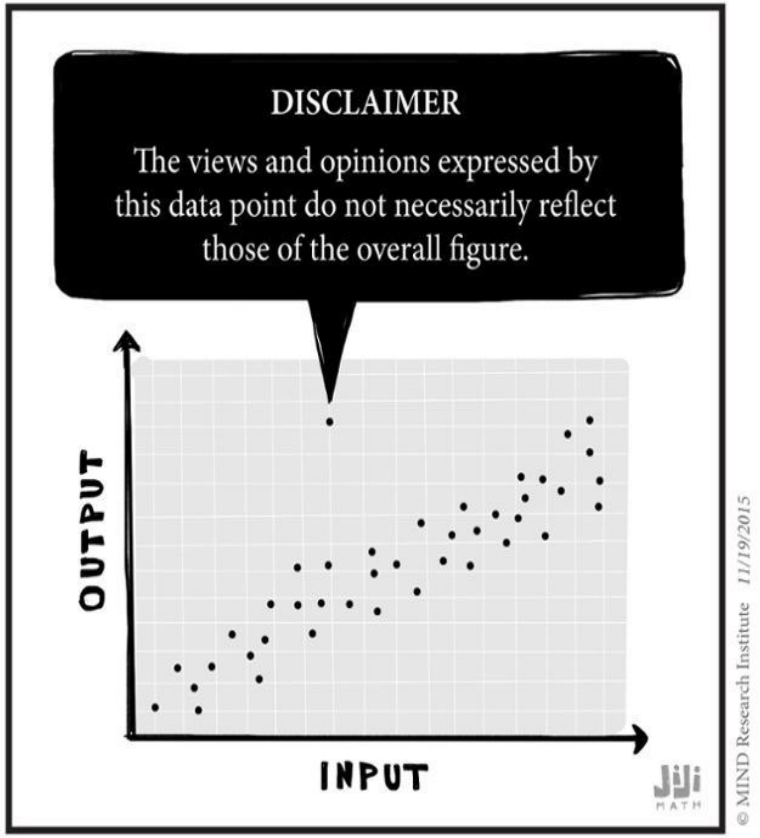
Проблема: выбросы и переобучение

Идея: ввести штраф за излишнюю сложность

Регуляризация – ограничение на норму вектора весов

$$\sum_{n=1}^{N} \left(x_n^T \omega - y_n\right)^2 + \lambda R(\omega) \to \min_{\beta}$$

Регуляризация



Регуляризация – ограничение на норму вектора весов

$$\sum_{n=1}^{N} \left(x_n^T \omega - y_n\right)^2 + \lambda R(\omega) \to \min_{\beta}$$

$R(\omega) = ||\omega||_1$ (L1) Lasso regression

$R(\omega) = ||\omega||_2^2$ (L2) Ridge regression


python ™