

«Машинное обучение»

Линейные методы (часть 1): линейная и логистическая регрессии

Александр Дьяконов

План

Линейная регрессия

Решение проблемы вырожденности

Регуляризация, гребневая регрессия, LASSO, Elastic Net

Устойчивая регрессия

Orthogonal matching pursuit

Градиентный метод обучения

Линейные скоринговые модели в задаче бинарной классификации

Логистическая регрессия

Линейные решающие модели в задаче бинарной классификации

Идея максимального зазора

SVM

Линейный дискриминант Фишера

Линейная регрессия

**Гипотеза о линейной зависимости целевой переменной,
ищем решение в виде:**

$$a(X_1, \dots, X_n) = w_0 + w_1 X_1 + \dots + w_n X_n$$

Практика:

- **часто неплохо работает и при монотонных зависимостях**
- **хорошо работает, когда есть много «однородных» зависимостей:**

цель – число продаж

признак 1 – число заходов на страницу продукта

признак 2 – число добавлений в корзину

признак 3 – число появлений продукта в поисковой выдаче

...

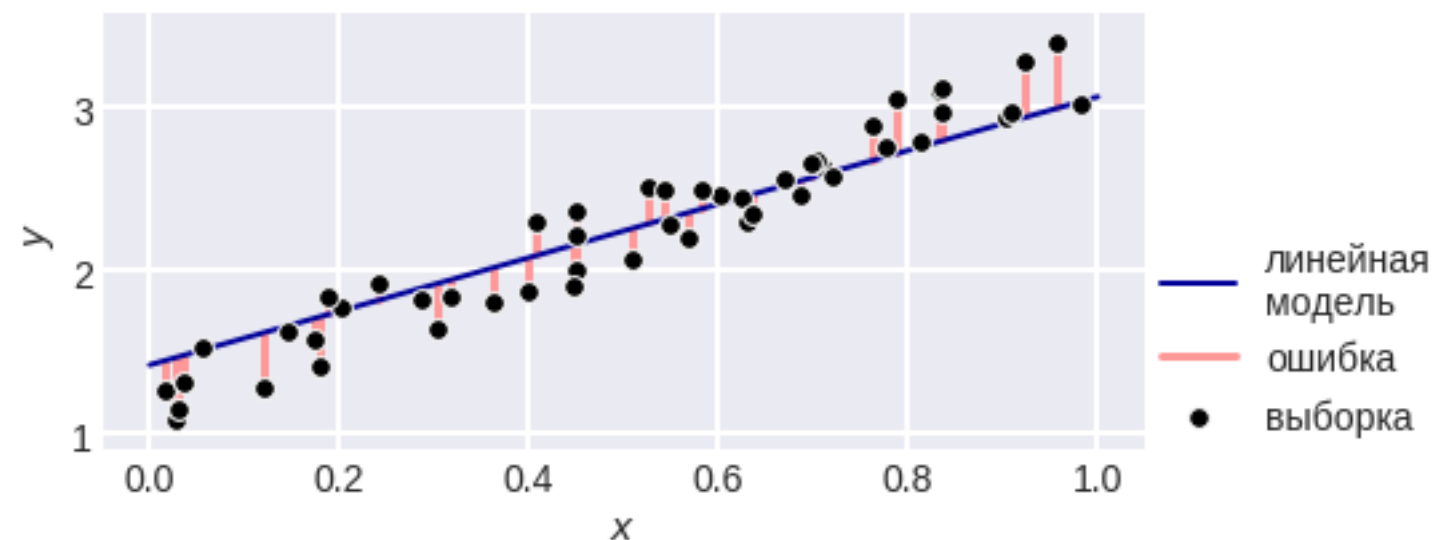
Линейная регрессия от одной переменной

$$a(X_1) = w_0 + w_1 X_1$$

обучение: $\{(x_1, y_1), \dots, (x_m, y_m)\}$, $x_i \in \mathbb{R}$

хотели бы...

$$\begin{cases} w_0 + w_1 x_1 = y_1 \\ \dots \\ w_0 + w_1 x_m = y_m \end{cases}$$



невязки / отклонения (residuals):

$$e_1 = y_1 - w_0 - w_1 x_1$$

...

$$e_m = y_m - w_0 - w_1 x_m$$

Линейная регрессия от одной переменной

Задача минимизации суммы квадратов отклонений (residual sum of squares)

$$\text{RSS} = e_1^2 + \dots + e_m^2 \rightarrow \min$$

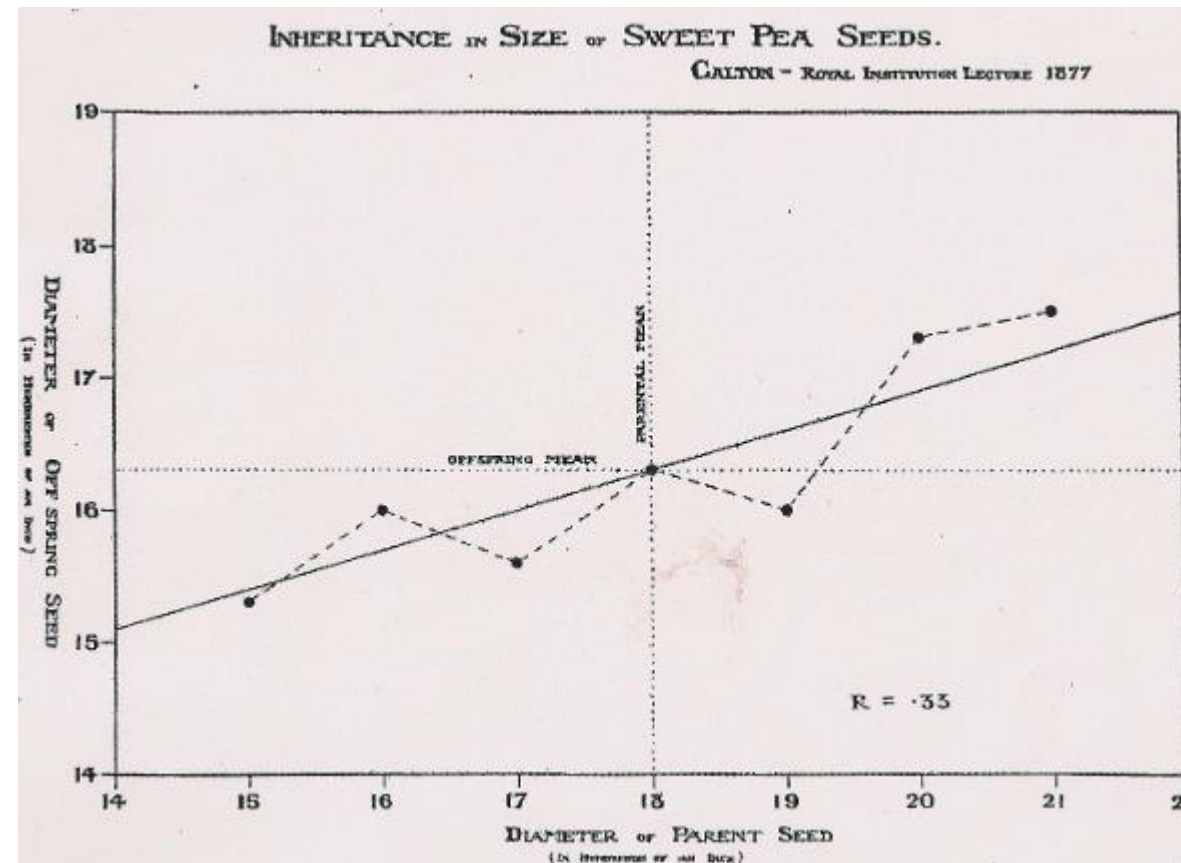
На это можно смотреть как на минимизацию эмпирического риска
по параметрам $w = (w_0, w_1)$

$$L(w) = \sum_{i=1}^m (y_i - a_w(x_i))^2 = \sum_{i=1}^m (y_i - (w_0 + w_1 x_i))^2$$

тут конкретная функция ошибки

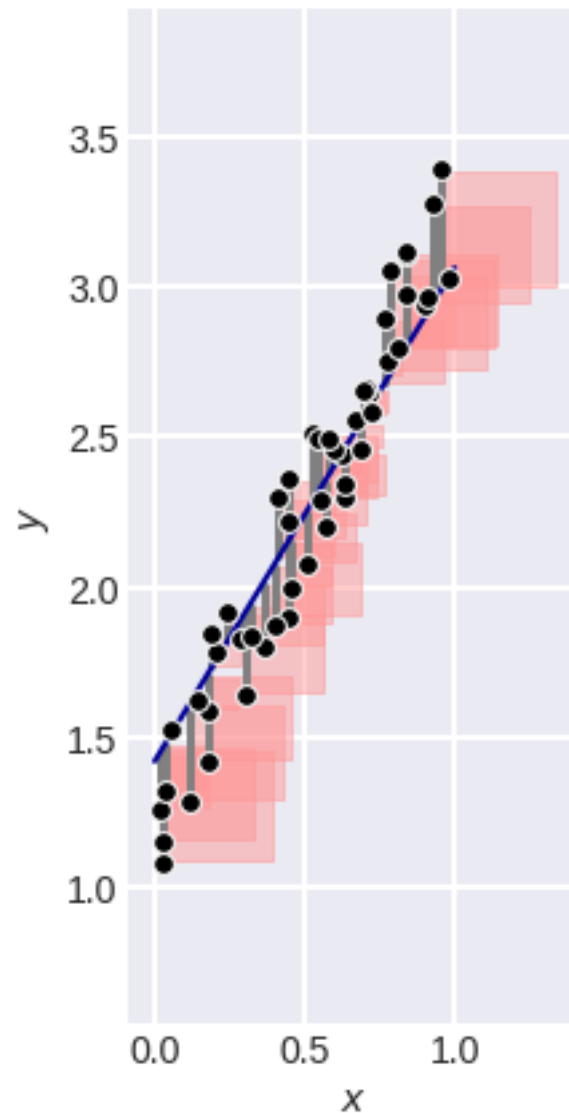
потом вероятностное обоснование, пока... довольно логично мы пытаем
мы пытаемся описать геометрию данных гиперплоскостью

Линейная регрессия от одной переменной

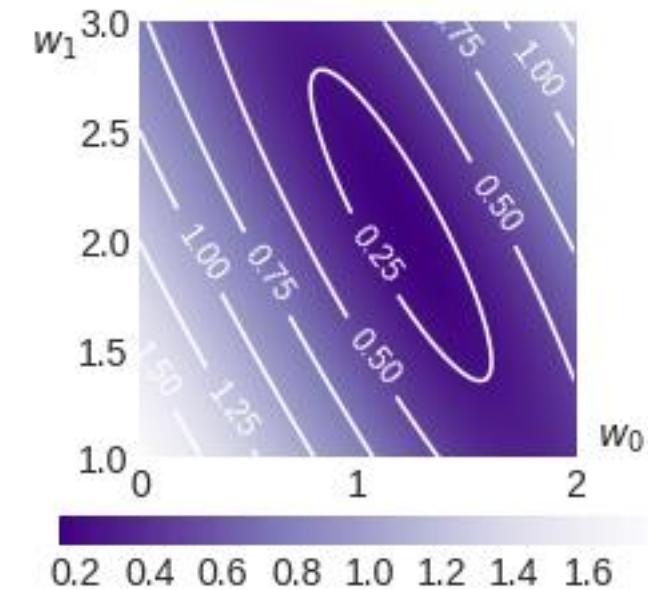


**«Регрессия к посредственности при наследовании роста»
Francis Galton, 1877**

Линейная регрессия от одной переменной: геометрический смысл ошибки



$$a(X_1) = w_0 + w_1 X_1$$



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

Отличается от суммы расстояний до поверхности!

Линейная регрессия от одной переменной

Нетрудно показать (**Д3**):

$$w_1 = \frac{\sum_{i=1}^m (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^m (x_i - \bar{x})^2} = \frac{\text{cov}(\{x_i\}, \{y_i\})}{\text{var}(\{x_i\})},$$

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

где $\bar{x} = \frac{1}{m} \sum_{i=1}^m x_i$, $\bar{y} = \frac{1}{m} \sum_{i=1}^m y_i$.

Полученное уравнение прямой (проходит через «центр масс»):

$$(y - \bar{y}) = \frac{\text{cov}(\{x_i\}, \{y_i\})}{\text{var}(\{x_i\})} (x - \bar{x})$$

Общий случай (многих переменных)

$$a(X_1, \dots, X_n) = w_0 + w_1 X_1 + \dots + w_n X_n = x^T w$$

веса (параметры) – $w = (w_0, w_1, \dots, w_n)^T$

объект – $x = (X_0, X_1, \dots, X_n)^T$

для удобства записи вводим фиктивный признак $X_0 \equiv 1$

обучение: $\{(x_1, y_1), \dots, (x_m, y_m)\}$, $x_i \in \mathbf{R}^{n+1}$,

опять хотим решить $Xw = y$:

$$\begin{cases} x_1^T w = y_1 \\ \dots \\ x_m^T w = y_m \end{cases}$$

как решать?

Общий случай (многих переменных): в матричной форме

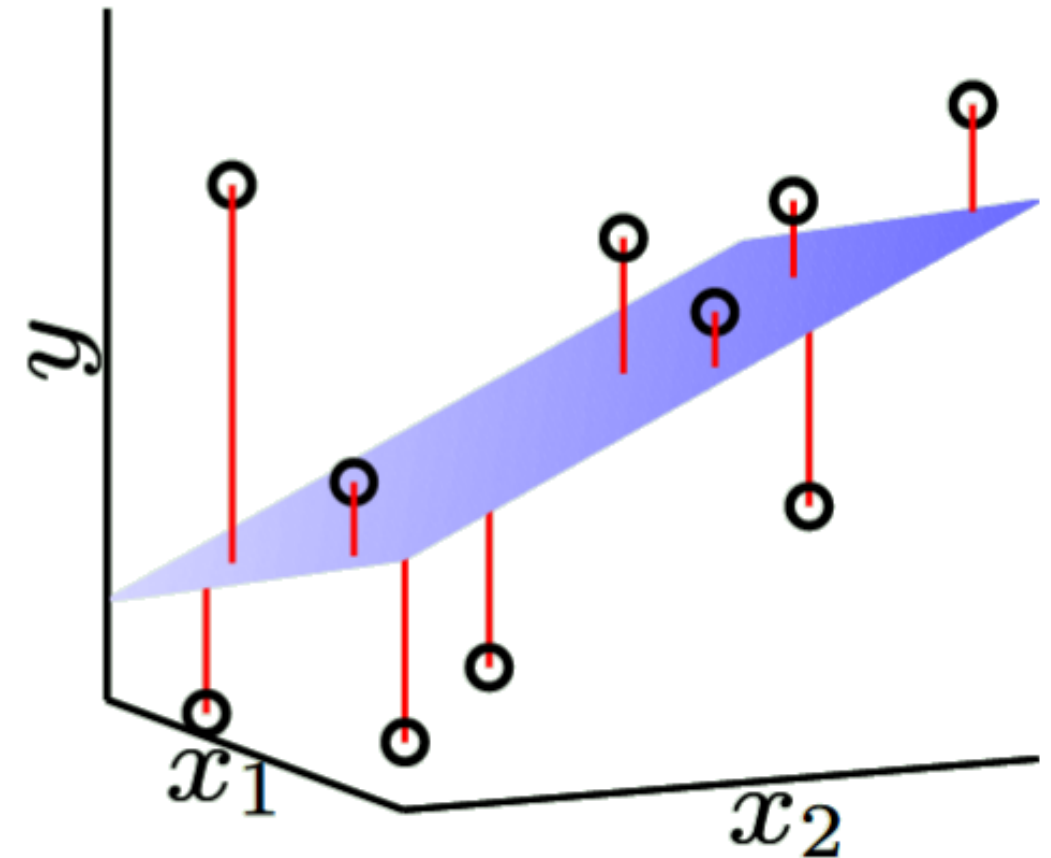
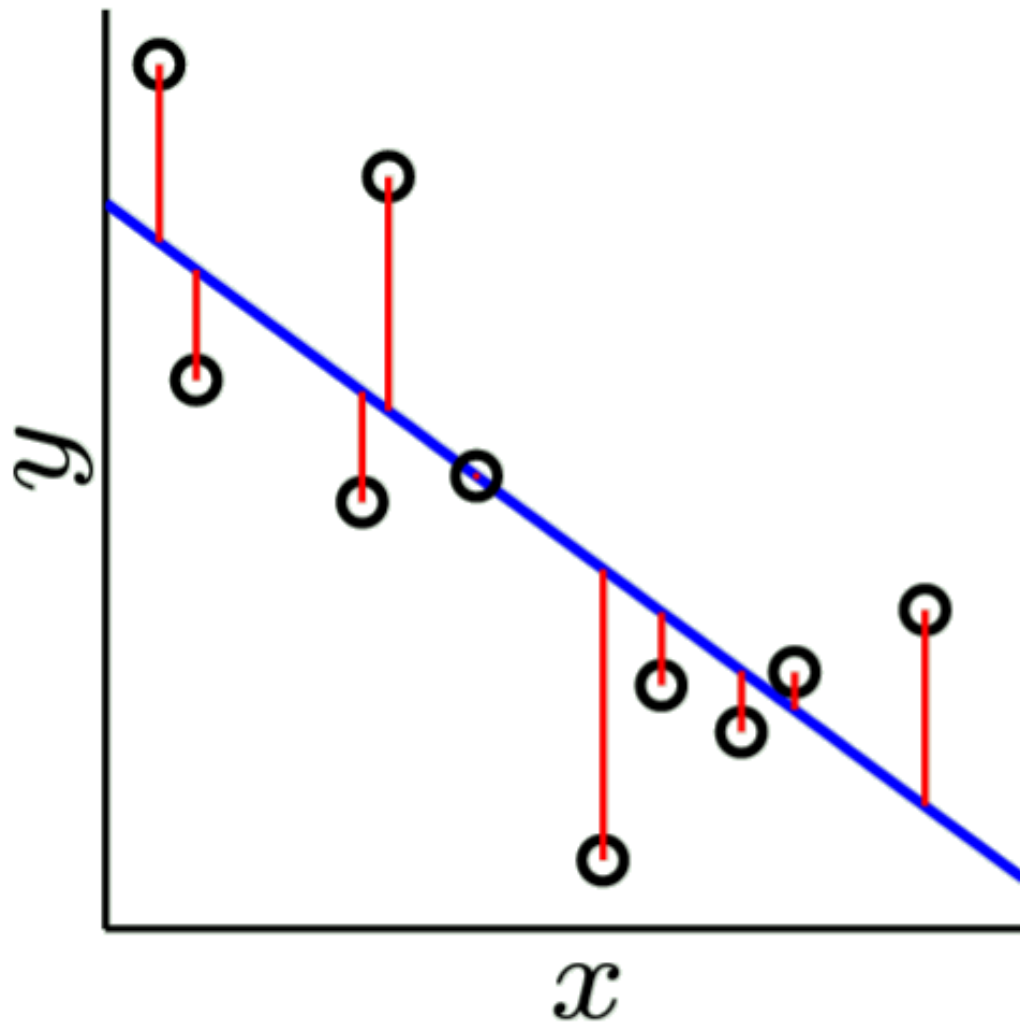
$$Xw = y$$

**в матрице X по строкам записаны описания объектов,
в векторе y значения их целевого признака
(здесь есть коллизия в обозначении y)**

будем решать так:

$$\|Xw - y\|_2^2 \rightarrow \min_w$$

$$\|Xw - y\|_2^2 = \sum_{i=1}^m (x_i^T w - y_i)^2 \rightarrow \min_w$$

Общий случай (многих переменных): геометрический смысл

Кстати, полученная задача оптимизации выпукла, единственный глобальный минимум (кроме вырожденного случая)

Решение задачи минимизации: прямой метод

$$\|Xw - y\|_2^2 \rightarrow \min_w$$

$$\|Xw - y\|_2^2 = (Xw - y)^T (Xw - y) = w^T X^T Xw - w^T X^T y - y^T Xw + y^T y$$

$$\nabla \|Xw - y\|_2^2 = 2X^T Xw - 2X^T y = 0$$

$$X^T Xw = X^T y$$

**решение существует,
если столбцы л/н**

$$w = (X^T X)^{-1} X^T y$$

помним, что $\text{rg}(X^T X) = \text{rg}(X)$

$(X^T X)^{-1} X^T$ – псевдообратная матрица Мура-Пенроуза
обобщение обратной на неквадратные матрицы

Обобщённая линейная регрессия: вместо X – что угодно

выражаем целевое значение через л/к базисных функций
(они фиксированы)

$$a(X_1, \dots, X_n) = w_0 + w_1 \varphi_1(X_1, \dots, X_n) + \dots + w_k \varphi_k(X_1, \dots, X_n)$$

$$w = (w_0, w_1, \dots, w_k)^T$$

$$x = (X_0, X_1, \dots, X_n)^T$$

$$\varphi(x) = (\varphi_0(x), \varphi_1(x), \dots, \varphi_k(x))^T$$

$\equiv 1$

$$a(x) = \sum_{i=0}^k w_i \varphi_i(x) = \varphi(x)^T w$$

$$\| \varphi(X)w - y \|_2^2 \rightarrow \min_w$$

$$\varphi(X) = \begin{bmatrix} \varphi_0(x_1) & \dots & \varphi_k(x_1) \\ \dots & \dots & \dots \\ \varphi_0(x_m) & \dots & \varphi_k(x_m) \end{bmatrix}$$

Подробности в нелинейных методах...

Проблема вырожденности матрицы

$$w = (X^T X)^{-1} X^T y$$

Только ли вырожденность плоха?

Что делать?

Проблема вырожденности матрицы

$$w = (X^T X)^{-1} X^T y$$

Проблемы, когда матрица $X^T X$ плохо обусловлена...

$$\mu(X^T X) = \|X^T X\| \cdot \|(X^T X)^{-1}\| = \frac{\lambda_{\max}(X^T X)}{\lambda_{\min}(X^T X)}$$

Решения:

1. Регуляризация – **здесь и в «сложности»**
2. Селекция (отбор) признаков – **«селекция»**
3. Уменьшение размерности (в том числе, PCA) – **USL**
4. Увеличение выборки

если объектов много – то работать с гигантской матрицей невозможно...
но выдели как это делается в оптимизации онлайн-методами

Регуляризация: упрощённое объяснение смысла

$$a(X_1, \dots, X_n) = w_0 + w_1 X_1 + \dots + w_n X_n$$

**если есть два похожих объекта, то должны быть похожи метки,
пусть отличаются в j -м признаке, тогда ответы модели отличаются на**

$$\varepsilon_j w_j$$

Поэтому не должно быть очень больших по модулю весов
(у признаков, по которым могут отличаться похожие объекты)

**Поэтому вместе с $\|Xw - y\|_2^2 \rightarrow \min$
хотим $\|w\|_2^2 \rightarrow \min$**

Не на все коэффициенты нужна регуляризация! Почему?

Регуляризация: упрощённое объяснение смысла

Пусть есть какая-то зависимость и лишние признаки, например

$$y = X_1 \text{ и } X_2 = X_3$$

Можно получить ответ в таком виде:

$$a = X_1 + w'X_2 - w'X_3$$

Если теперь $X_2 \approx X_3$, тогда $\varepsilon = X_2 - X_3$

$$a = X_1 + w'\varepsilon$$

– может быть сколь угодно большим при больших (по модулю) w'

**аналогично при линейных зависимостях!
автоматически, когда объектов мало (сколько?)**

Регуляризация

Иванова

$$\begin{cases} \|Xw - y\|_2^2 \rightarrow \min \\ \|w\|_2^2 \leq \lambda \end{cases}$$

Тихонова

$$\|Xw - y\|_2^2 + \lambda \|w\|_2^2 \rightarrow \min$$

Удобнее: безусловная оптимизация

$$\|w\|_2^2 = w_1^2 + w_2^2 + \dots + w_n^2 - \text{нет } w_0^2$$

эти две формы эквивалентны: решение одного можно получить как решение другого**Всё это справедливо и для общих задач минимизации!**

$$\begin{cases} L(a) \rightarrow \min \\ \text{complexity}(a) \leq \lambda \end{cases}$$

$$L(a) + \lambda \text{complexity}(a) \rightarrow \min$$

Есть ещё регуляризация Морозова...

Регуляризация и гребневая регрессия

$$\arg \min_w \|Xw - y\|_2^2 + \lambda \|w\|_2^2 = (X^T X + \lambda I)^{-1} X^T y$$
$$\lambda \geq 0$$

ДЗ Доказать!

Такая регрессия называется **гребневой регрессией (Ridge Regression)**

**Виден другой смысл регрессии: складываем две матрицы Грама,
неотрицательно определённая + положительно определённая**

– боремся с вырожденностью матрицы

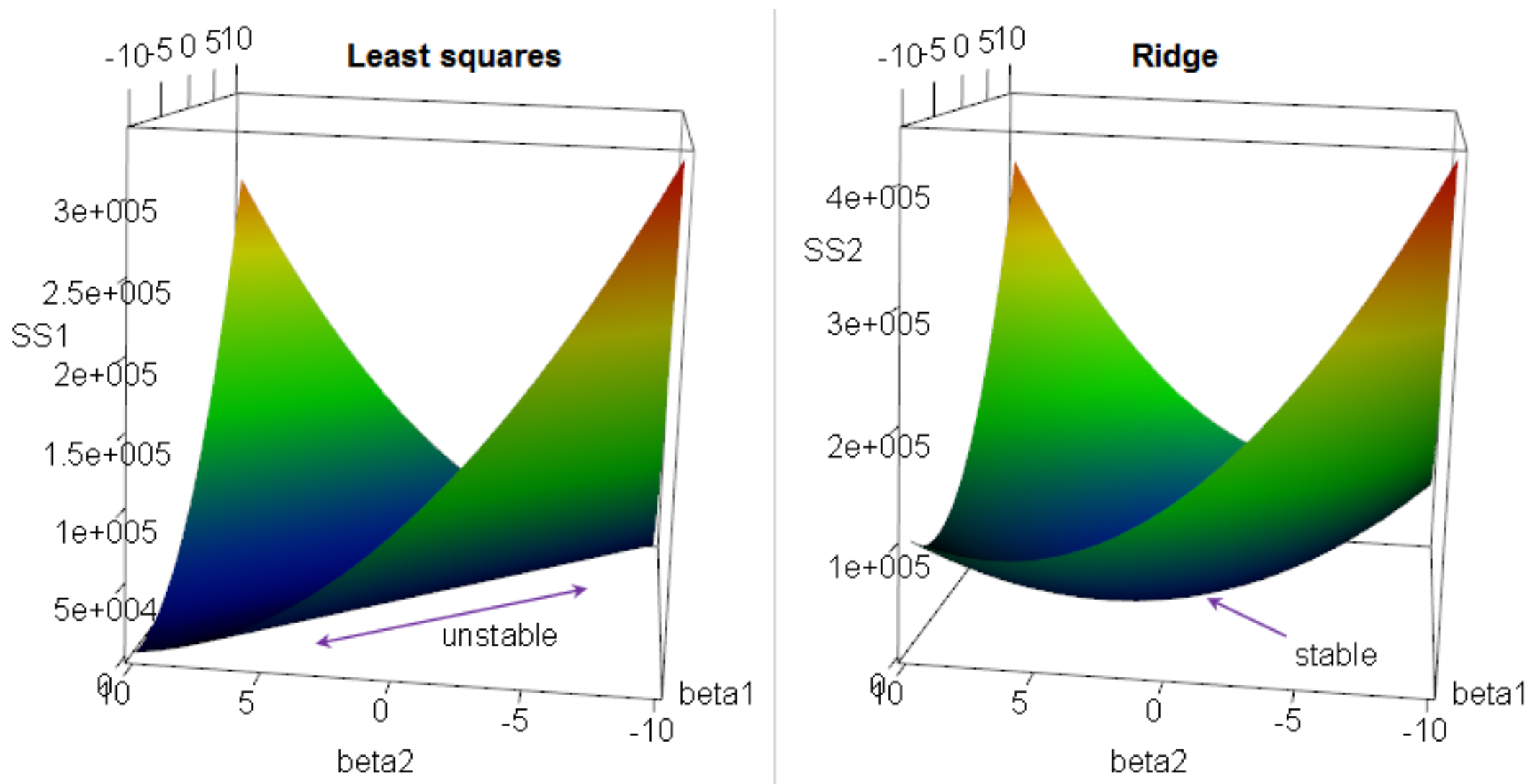
потом вернёмся к этому

коэффициент регуляризации (shrinkage penalty)

$\lambda = 0$ – получаем классическое решение
 $\lambda \rightarrow +\infty$ – меньше «затачиваемся на данные» и больше регуляризуем

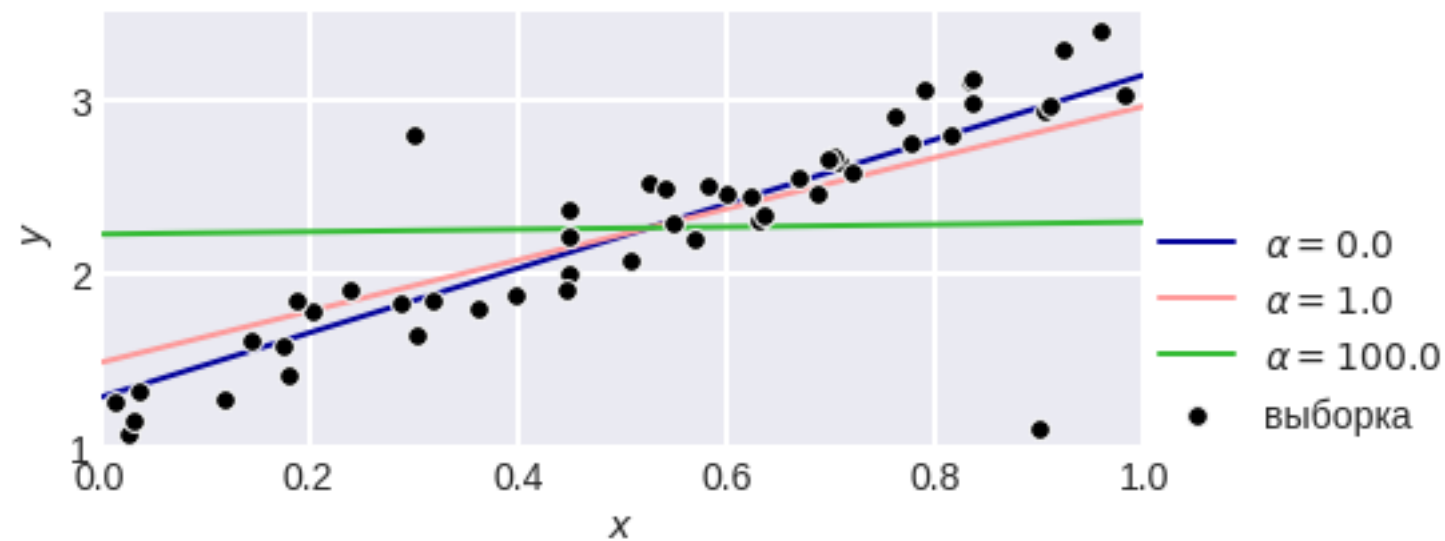
значение параметра регуляризации можно выбрать на скользящем контроле

Регуляризация и гребневая регрессия



<https://stats.stackexchange.com/questions/151304/why-is-ridge-regression-called-ridge-why-is-it-needed-and-what-happens-when>

Минутка кода: регуляризация и гребневая регрессия



```
from sklearn.linear_model import Ridge

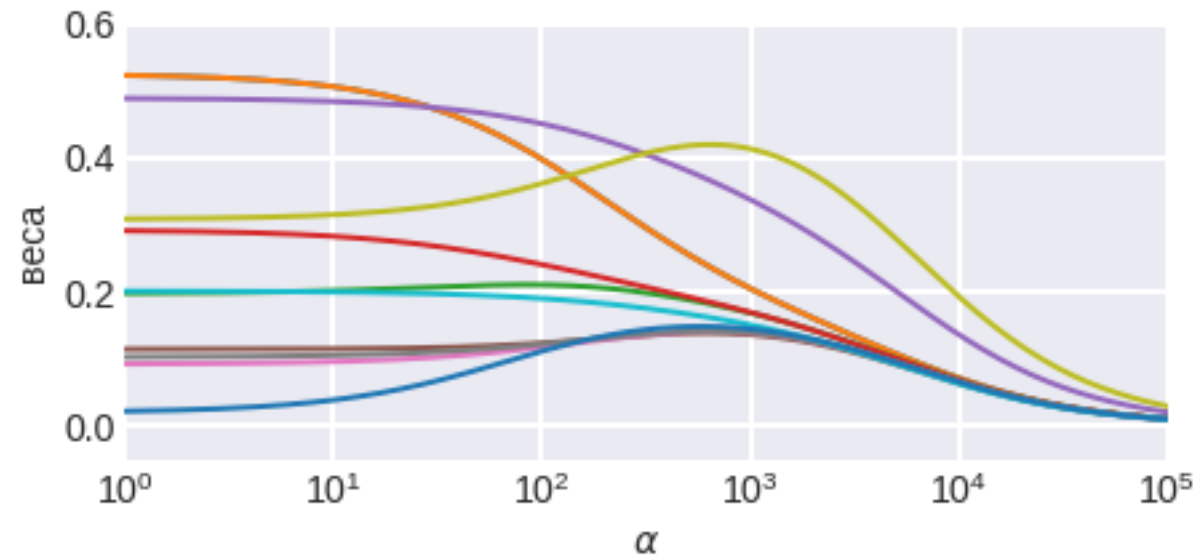
model = Ridge(alpha=0.0) # ридж-регрессия
# обучение
model.fit(x_train[:, np.newaxis], y_train)
# обратите внимание: np.newaxis
# контроль
a_train = model.predict(x_train[:, np.newaxis])
a_test = model.predict(x_test[:, np.newaxis])
```

Кажется, что при регуляризации
отклоняемся к выбросам,
но дело не в этом

Регуляризация и гребневая регрессия

$$\sum_{i=1}^m (y_i - a(x_i))^2 + \lambda \sum_{j=1}^n w_j^2 \rightarrow \min$$
$$\lambda \geq 0$$

добавление shrinkage penalty (регуляризатора)



параметр регуляризации может подбираться с помощью скользящего контроля

Регуляризация и гребневая регрессия

Для ridge-регрессии нужна правильная нормировка признаков!

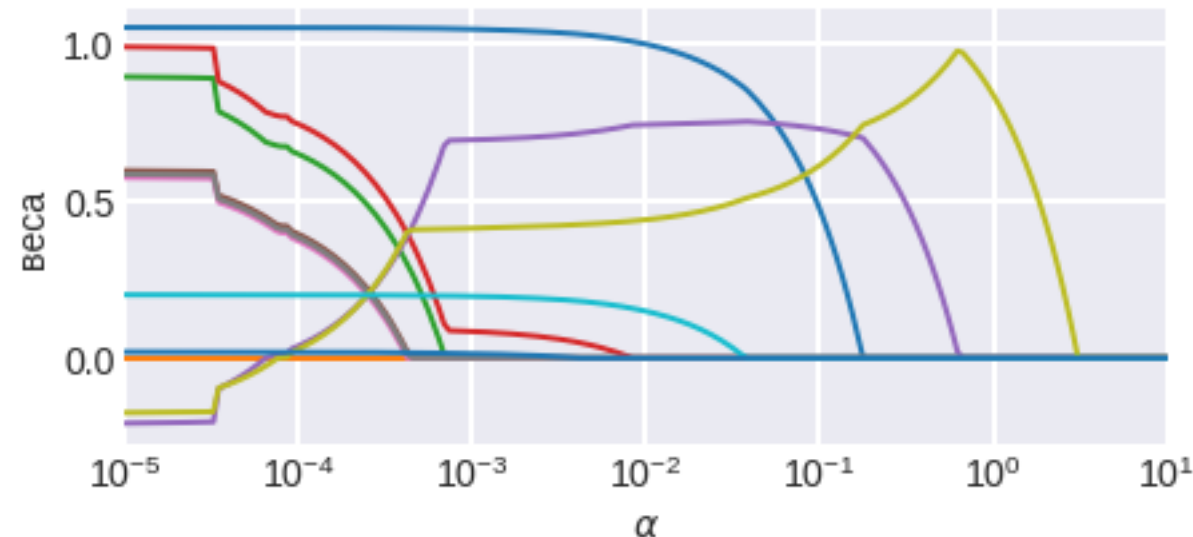
Нет инвариантности (в отличие от линейной) от умножения признаков на скаляры

Перед регуляризацией – стандартизация!!!

LASSO (Least Absolute Shrinkage and Selection Operator)

Попробуем другой «штраф за сложность» (сейчас поймём название)

$$\sum_{i=1}^m (y_i - a(x_i))^2 + \lambda \sum_{j=1}^n |w_j| \rightarrow \min$$
$$\lambda \geq 0$$



Здесь значения коэффициентов существенно меньше (т.к. при $\Sigma|\cdot|$, а не $\Sigma(\cdot)^2$)

Здесь коэффициенты интенсивнее зануляются при увеличении $\lambda \geq 0$

Эксперименты с одинаковыми и зависимыми признаками

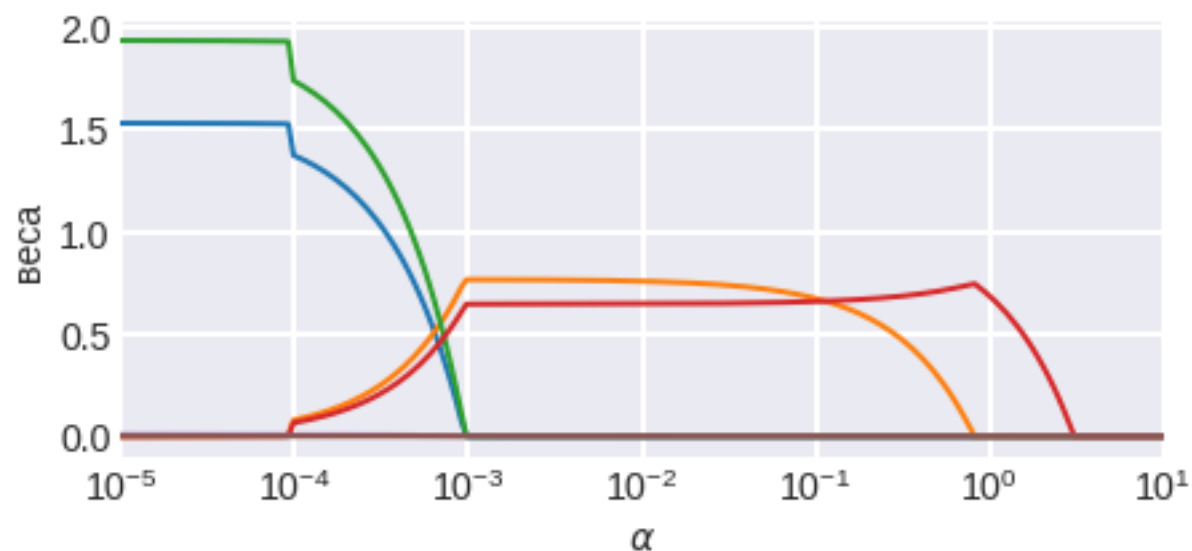
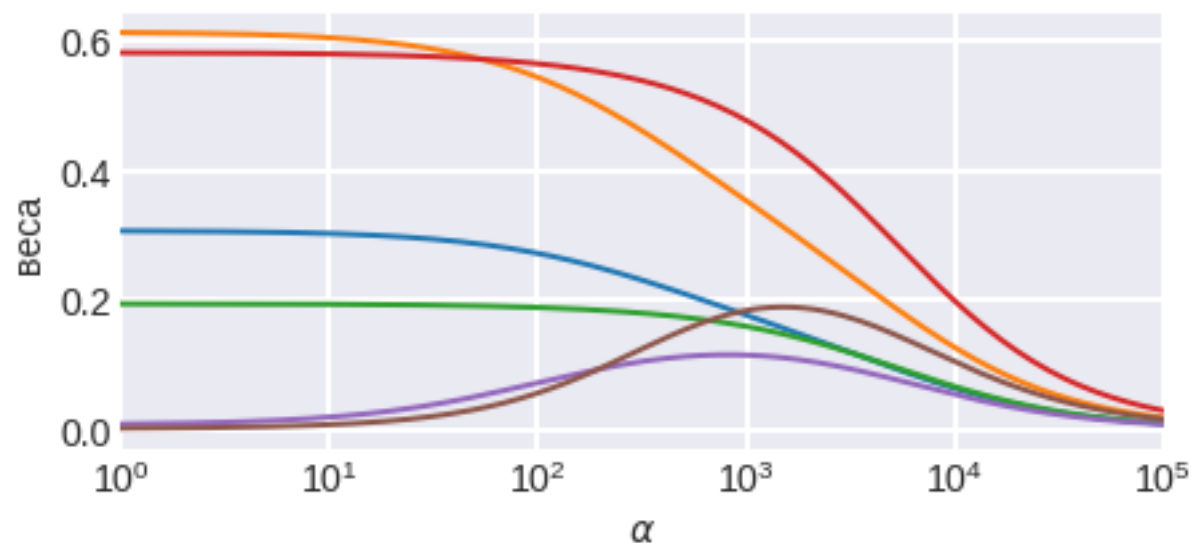
здесь была задача

```
X = np.random.rand(1000, 11)
X[:,1] = X[:,0]
X[:,4] = X[:,2] + X[:,3]
X[:,8] = X[:,5] + X[:,6] + X[:,7]
y = X[:,0] + 0.8*X[:,4] + 0.4*X[:,8] + 0.2*X[:,9] +
    0.5*np.random.randn(1000)
```

зависит от масштаба признаков,
но из-за предварительной нормировки этот эффект не наблюдается

Масштаб очень важен! см. дальше

Эксперименты с одинаковыми и зависимыми признаками



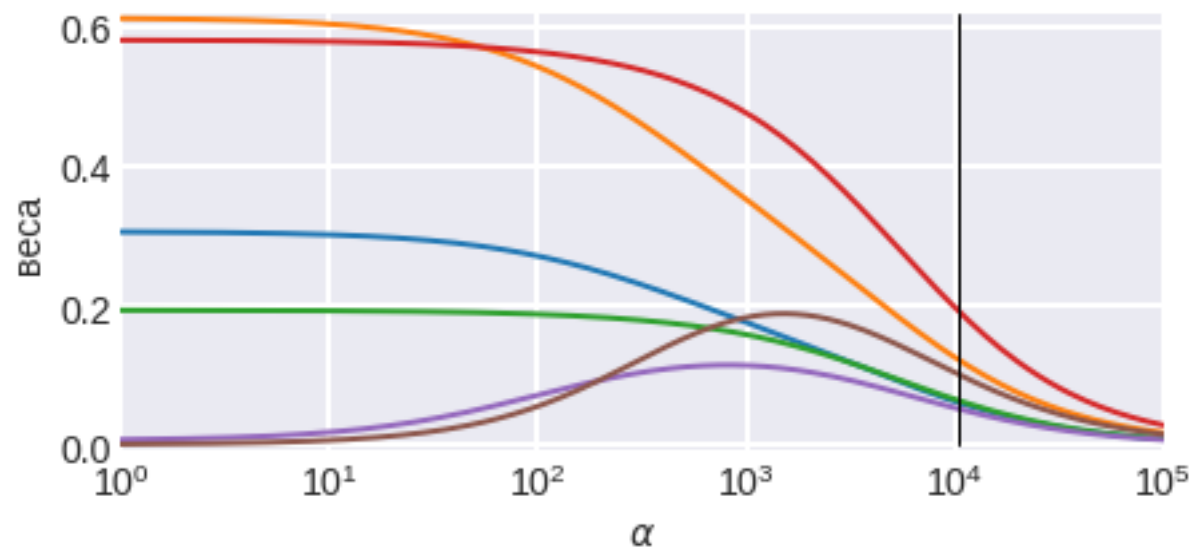
```
np.random.seed(10)
X = np.random.rand(1000, 6)
X[:,1] = X[:,0]
X[:,2] = X[:,3]
```

```
X[:,0] = 1 * X[:,0]
X[:,1] = 2 * X[:,1]
X[:,2] = 1 * X[:,2]
X[:,3] = 3 * X[:,3]
X[:,4] = 1 * X[:,4]
X[:,5] = 2 * X[:,5]
```

```
y = 1.5 * X[:,0] + 2*X[:,2] +
0.5*np.random.randn(1000)
```

Эксперименты с одинаковыми и зависимыми признаками

$$Y = 1.5X_1 + 2X_3 = 0.75X_2 + 0.66X_4$$

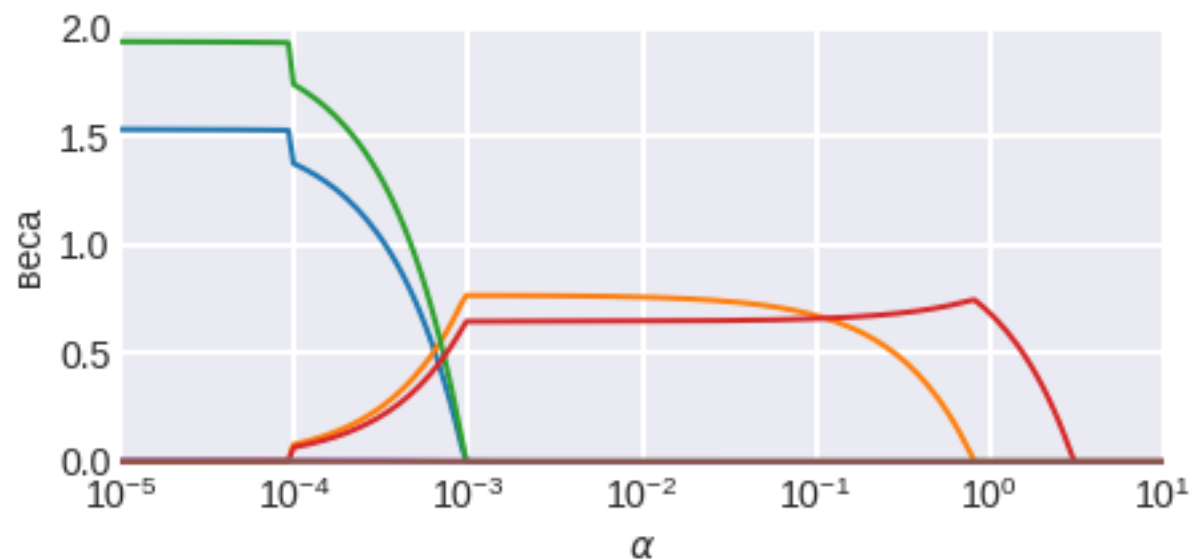


$$\lambda = 1$$

$$Y = 0.31X_1 + 0.61X_2 + 0.19X_3 + 0.58X_4 + 0.01X_5 + 0.0X_6$$

$$\lambda \sim 10500$$

$$Y = 0.06X_1 + 0.12X_2 + 0.06X_3 + 0.19X_4 + 0.05X_5 + 0.1X_6$$



$$\lambda = 10^{-5}$$

$$Y = 1.53X_1 + 1.94X_3$$

$$\lambda \sim 0.01$$

$$Y = 0.76X_2 + 0.65X_4$$

Эксперименты с одинаковыми и зависимыми признаками

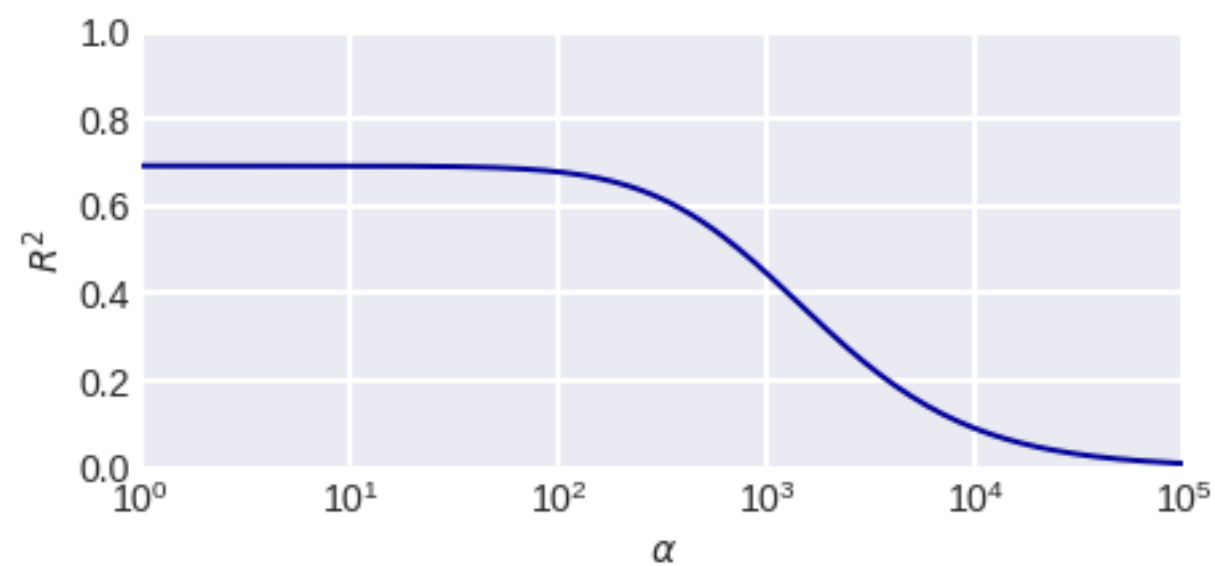
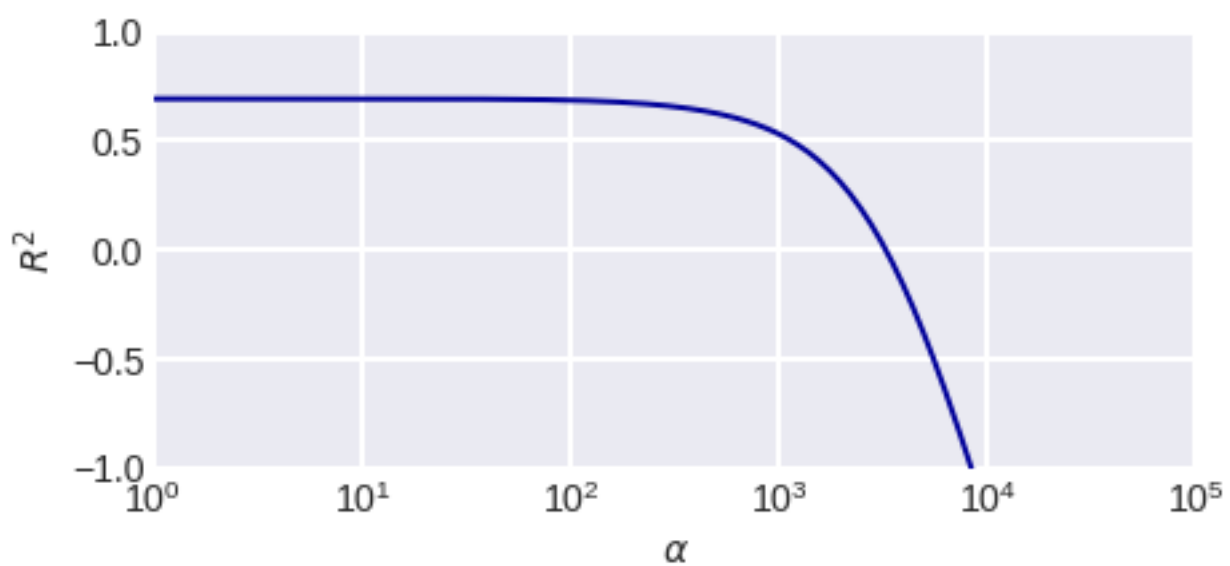
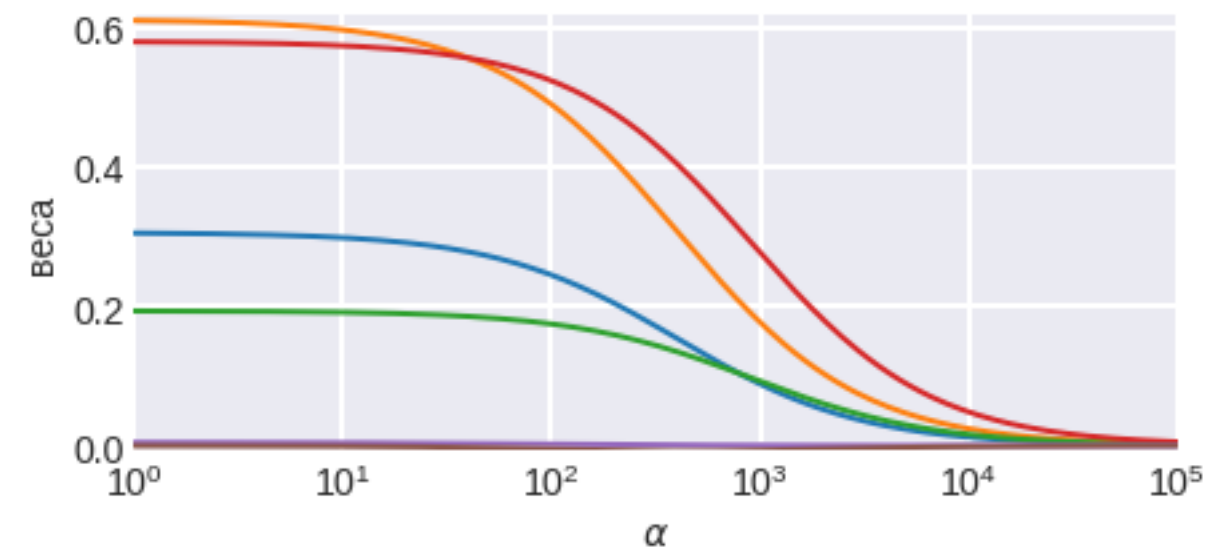
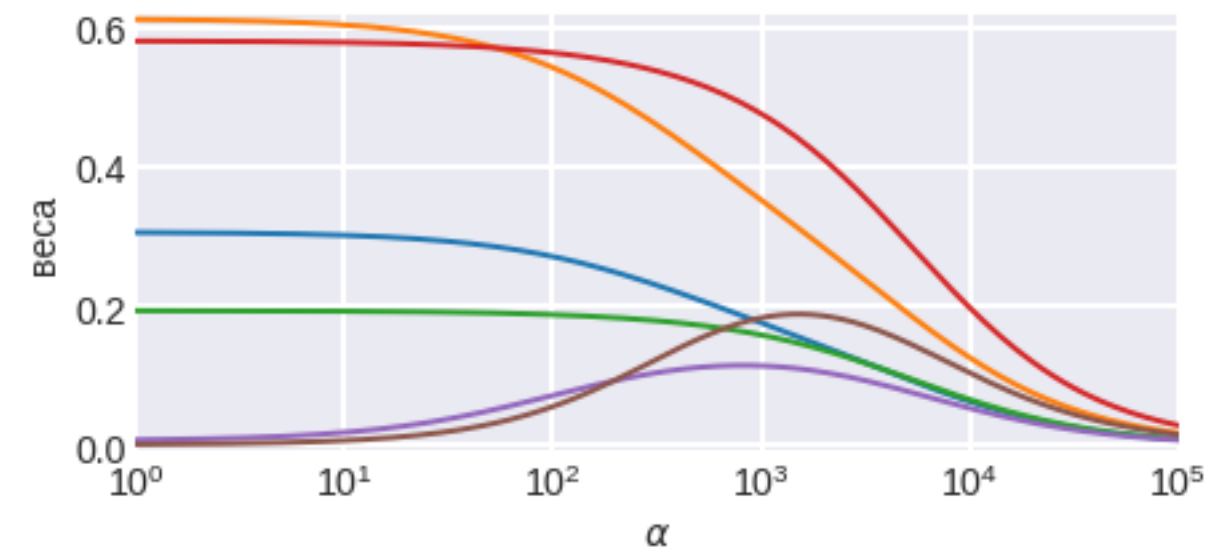
веса зависят от масштаба признаков

при сильной регуляризации меняется распределение весов зависимых признаков

Пусть $Y = 4X_1, X_1=X_2$

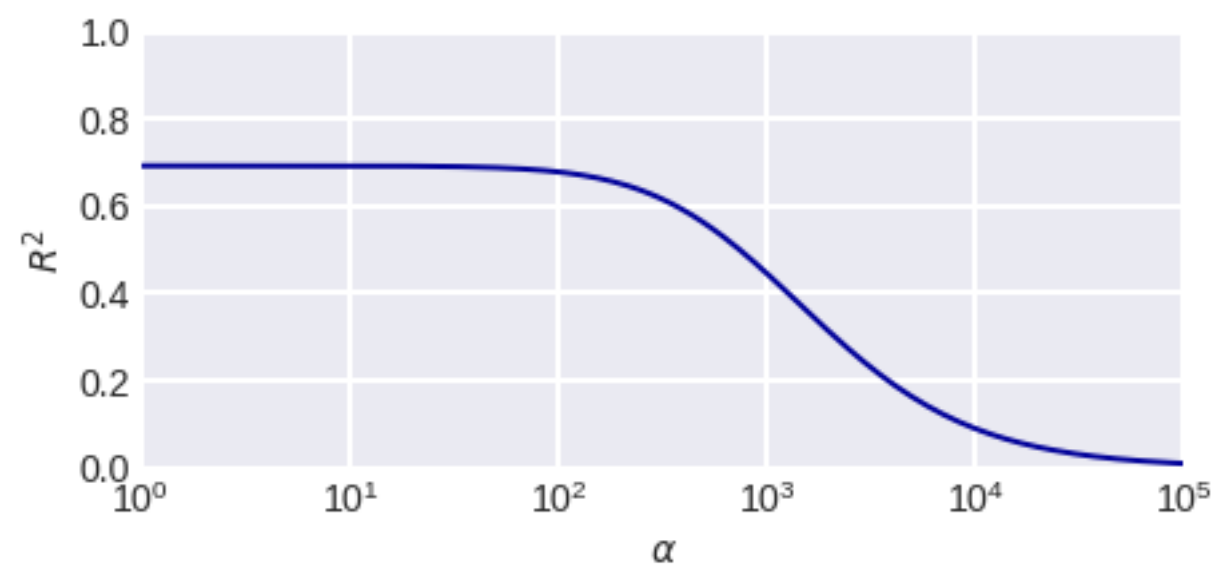
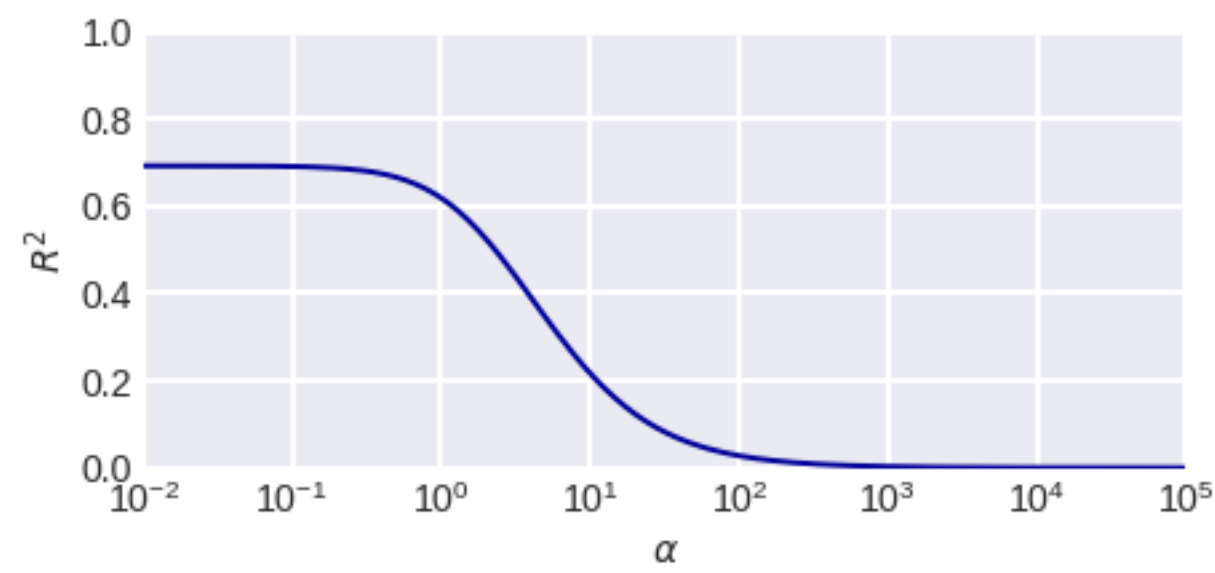
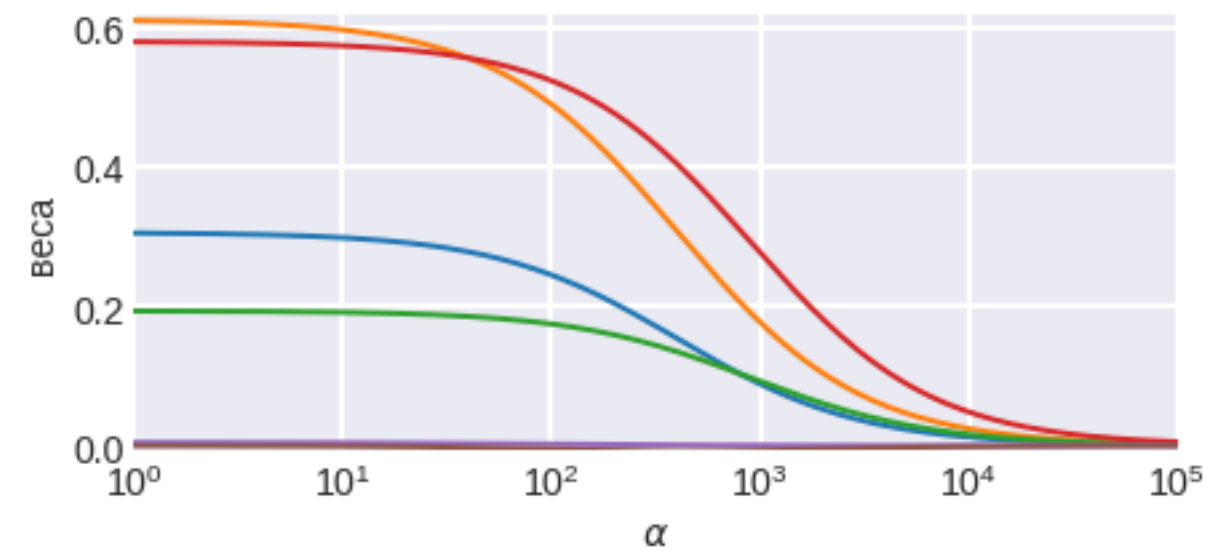
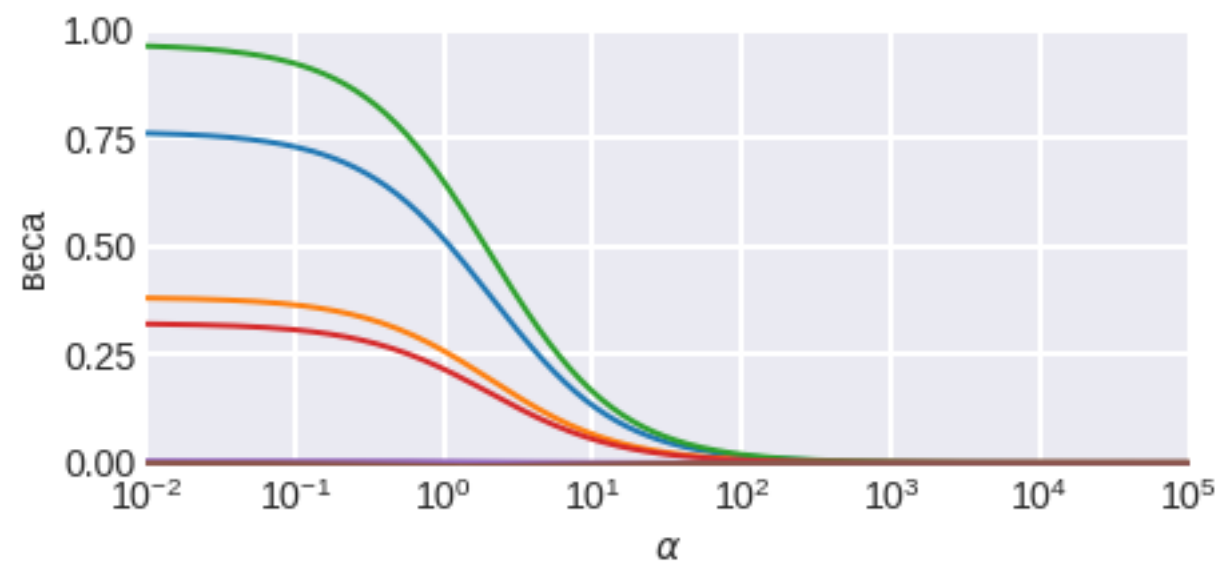
w_1	w_2	$\ w \ _1$	$\ w \ _2^2$
5	- 1	6	26
4	0	4	16
3	1	4	10
2	2	4	8

Эксперименты с одинаковыми и зависимыми признаками: L_2 -регуляризация



`fit_intercept=True`

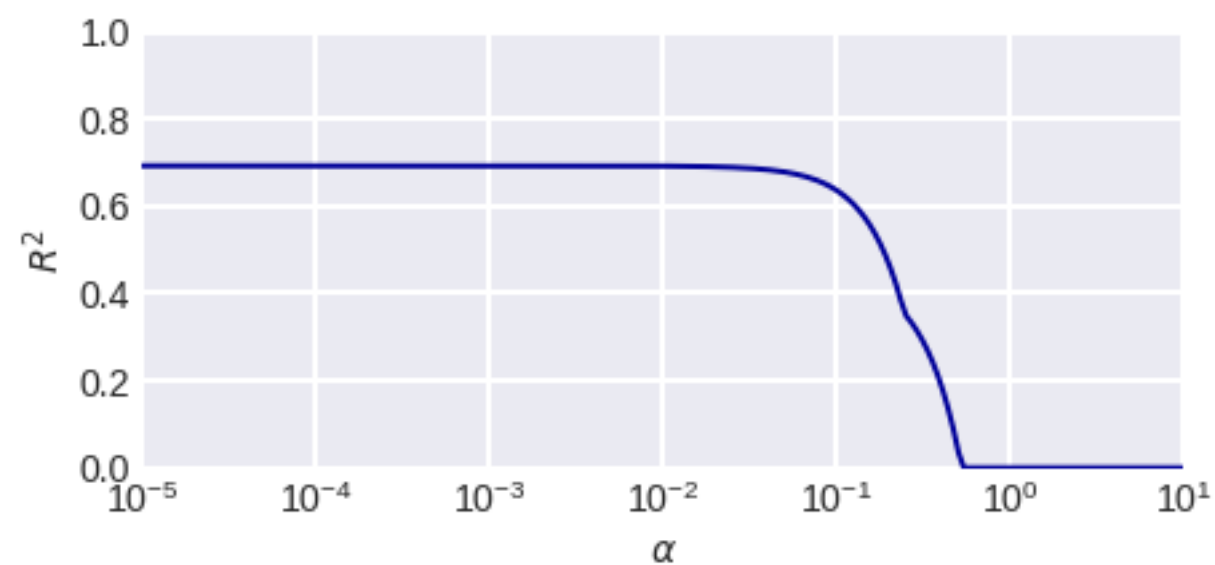
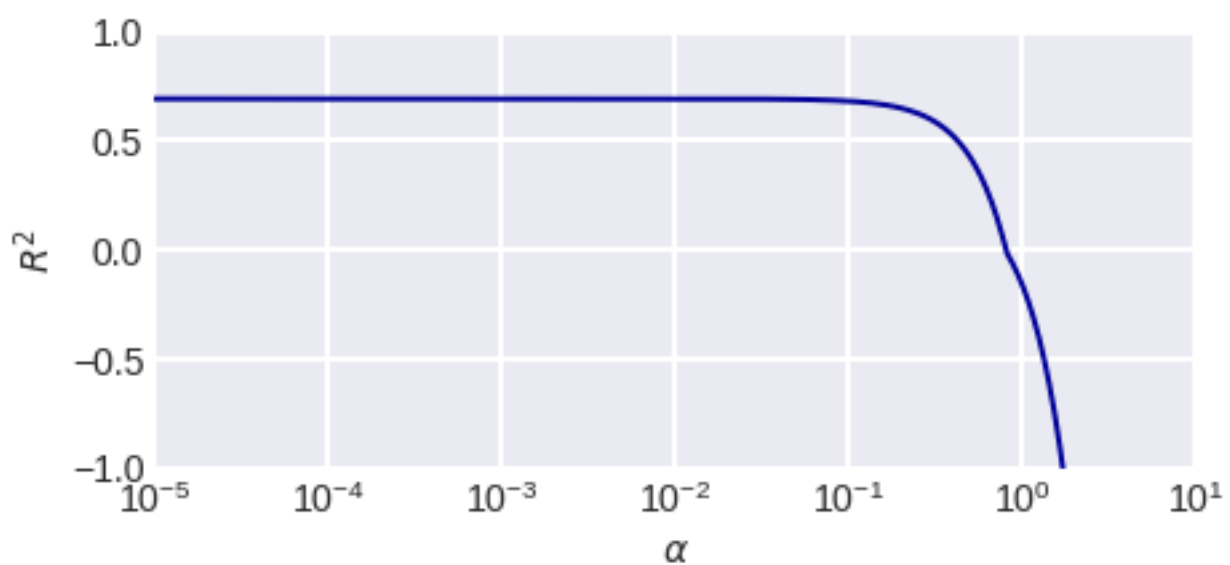
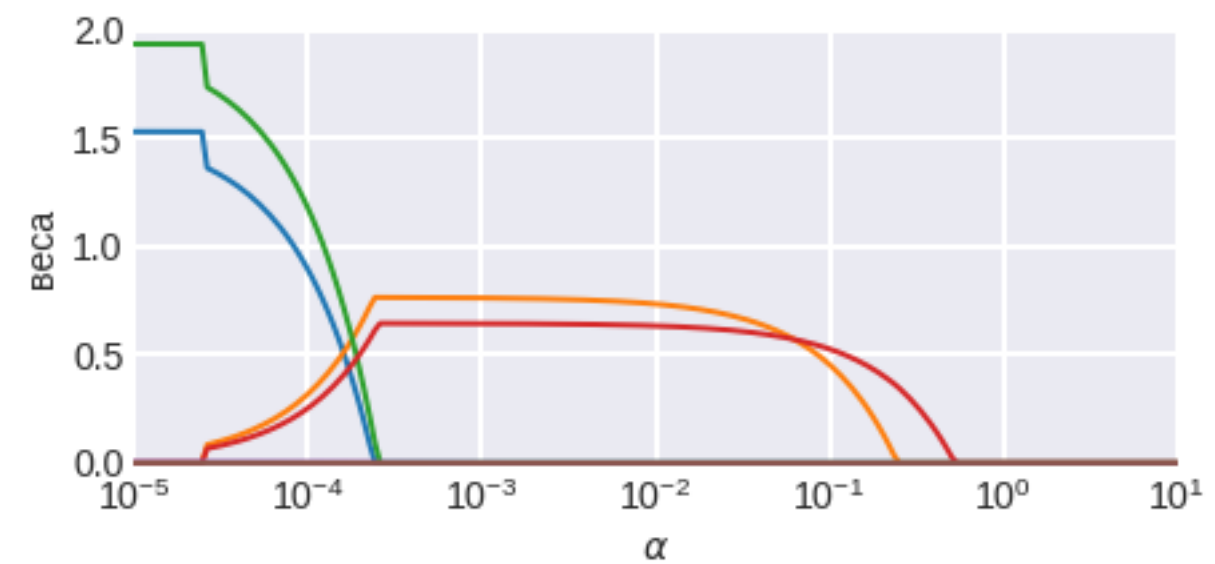
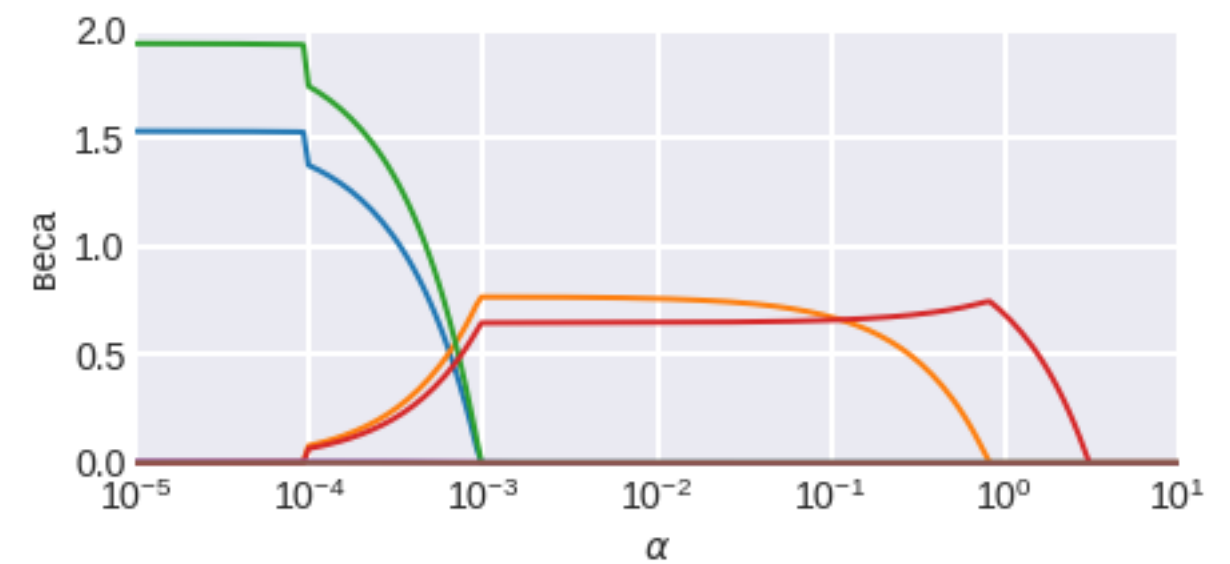
Эксперименты с одинаковыми и зависимыми признаками: L_2 -регуляризация



`fit_intercept=True, normalize=True`

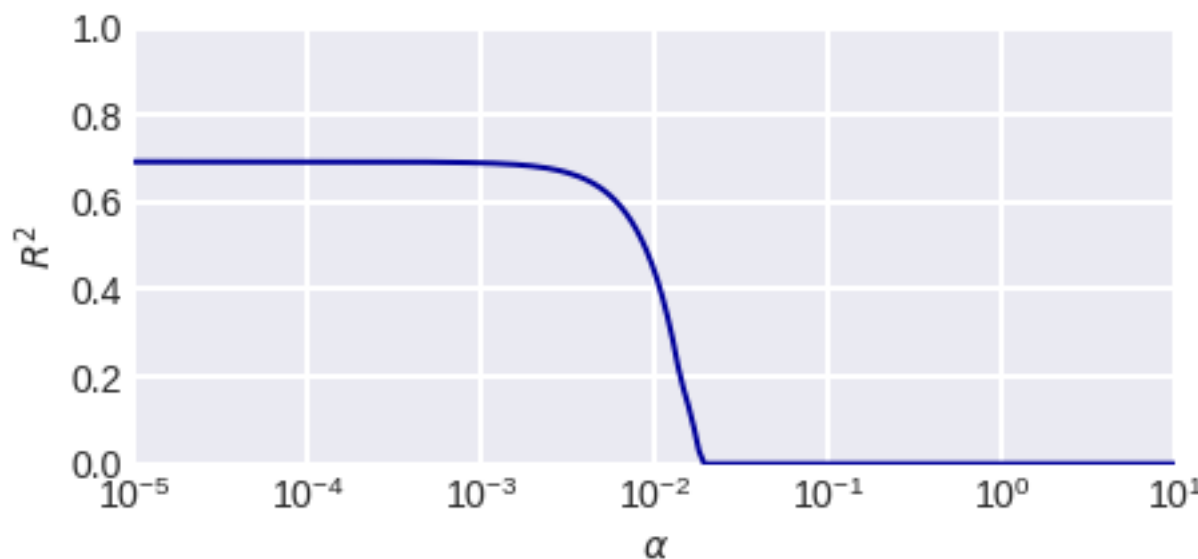
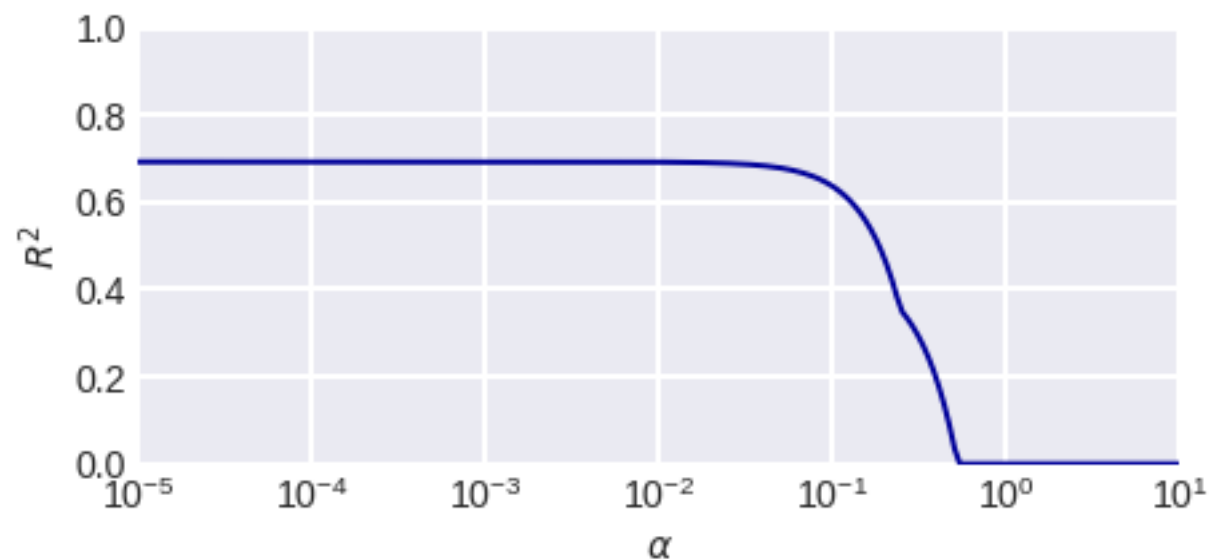
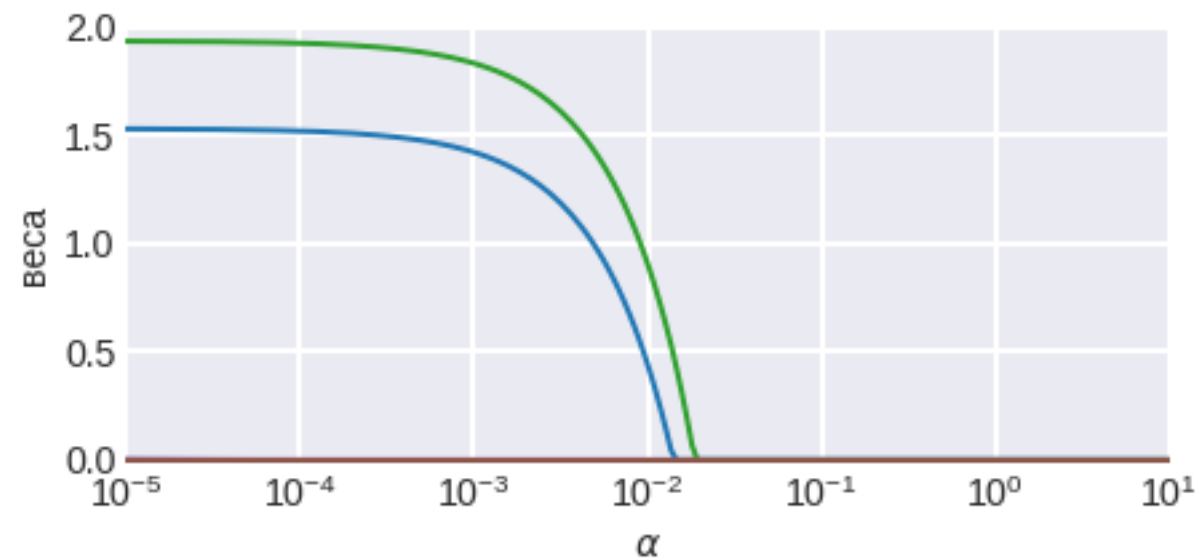
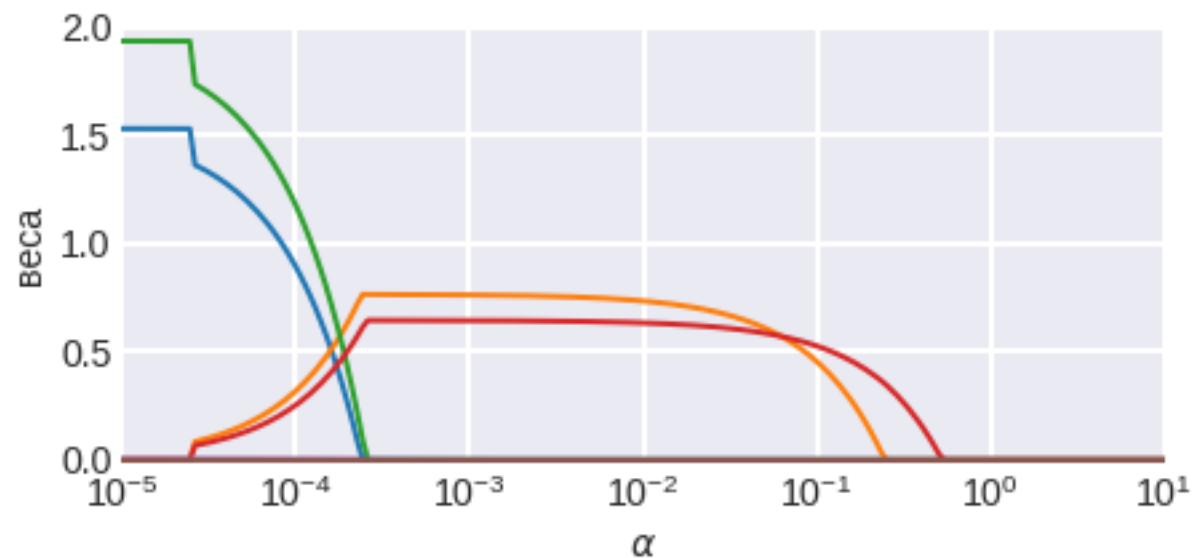
`fit_intercept=True`

Эксперименты с одинаковыми и зависимыми признаками: L_1 -регуляризация



fit_intercept=True

Эксперименты с одинаковыми и зависимыми признаками: L_1 -регуляризация



`fit_intercept=True`

`fit_intercept=True, normalize=True`

Эксперименты с одинаковыми и зависимыми признаками

Часто важно

- **использовать свободный член**
- **предварительно нормировать данные**

Семейство регуляризированных линейных методов**Ridge**

$$\|y - Xw\|_2^2 + \lambda \|w\|_2^2 \rightarrow \min_w$$

LASSO (Least Absolute Shrinkage and Selection Operator)

$$\|y - Xw\|_2^2 + \lambda \|w\|_1 \rightarrow \min_w$$

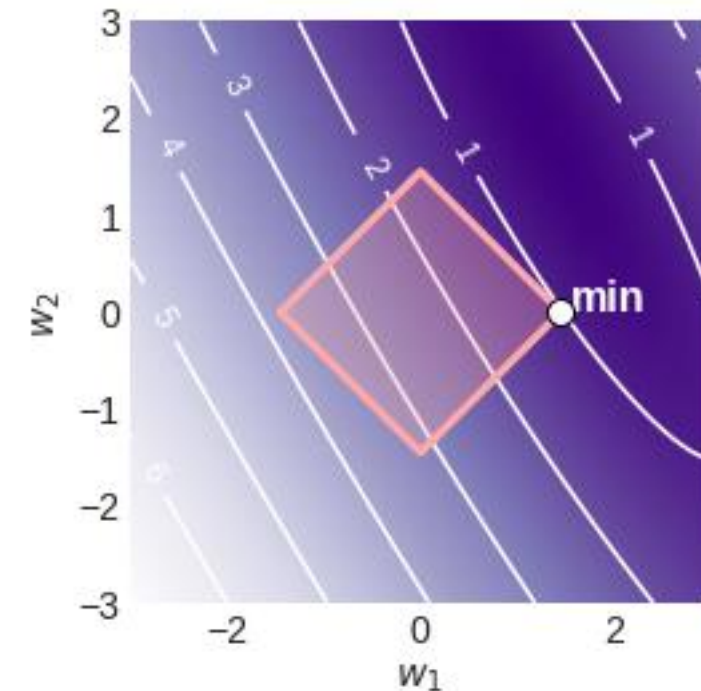
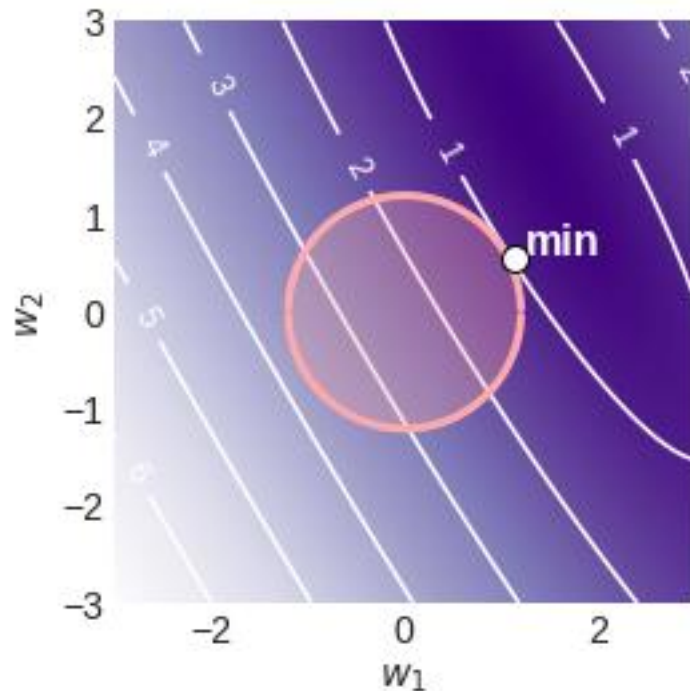
Elastic Net = LASSO + Ridge

$$\|y - Xw\|_2^2 + \lambda_1 \|w\|_1 + \lambda_2 \|w\|_2^2 \rightarrow \min_w$$

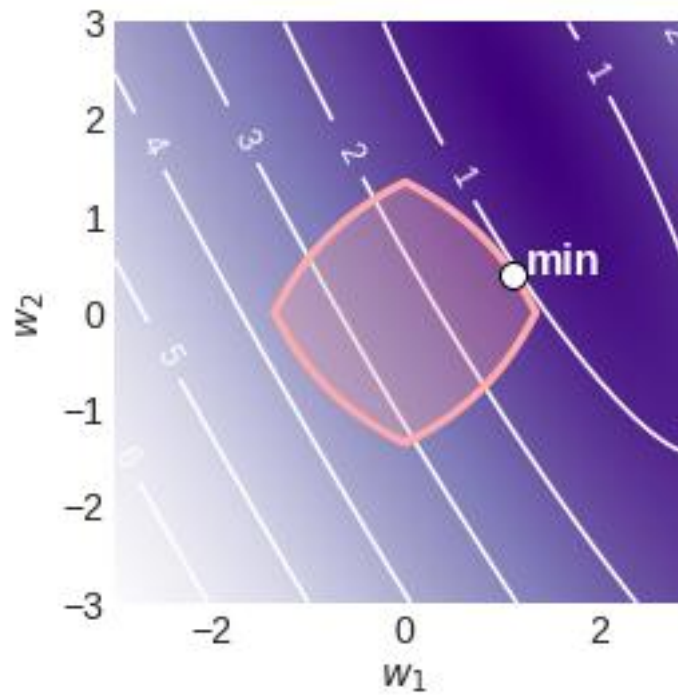
Геометрический смысл Ridge и LASSO

$$\sum_{i=1}^m \left(y_i - w_0 - \sum_{j=1}^n w_j x_{ij} \right)^2 \rightarrow \min_w, \quad \sum_{j=1}^n w_j^2 \leq s$$

$$\sum_{i=1}^m \left(y_i - w_0 - \sum_{j=1}^n w_j x_{ij} \right)^2 \rightarrow \min_w, \quad \sum_{j=1}^n |w_j| \leq s$$



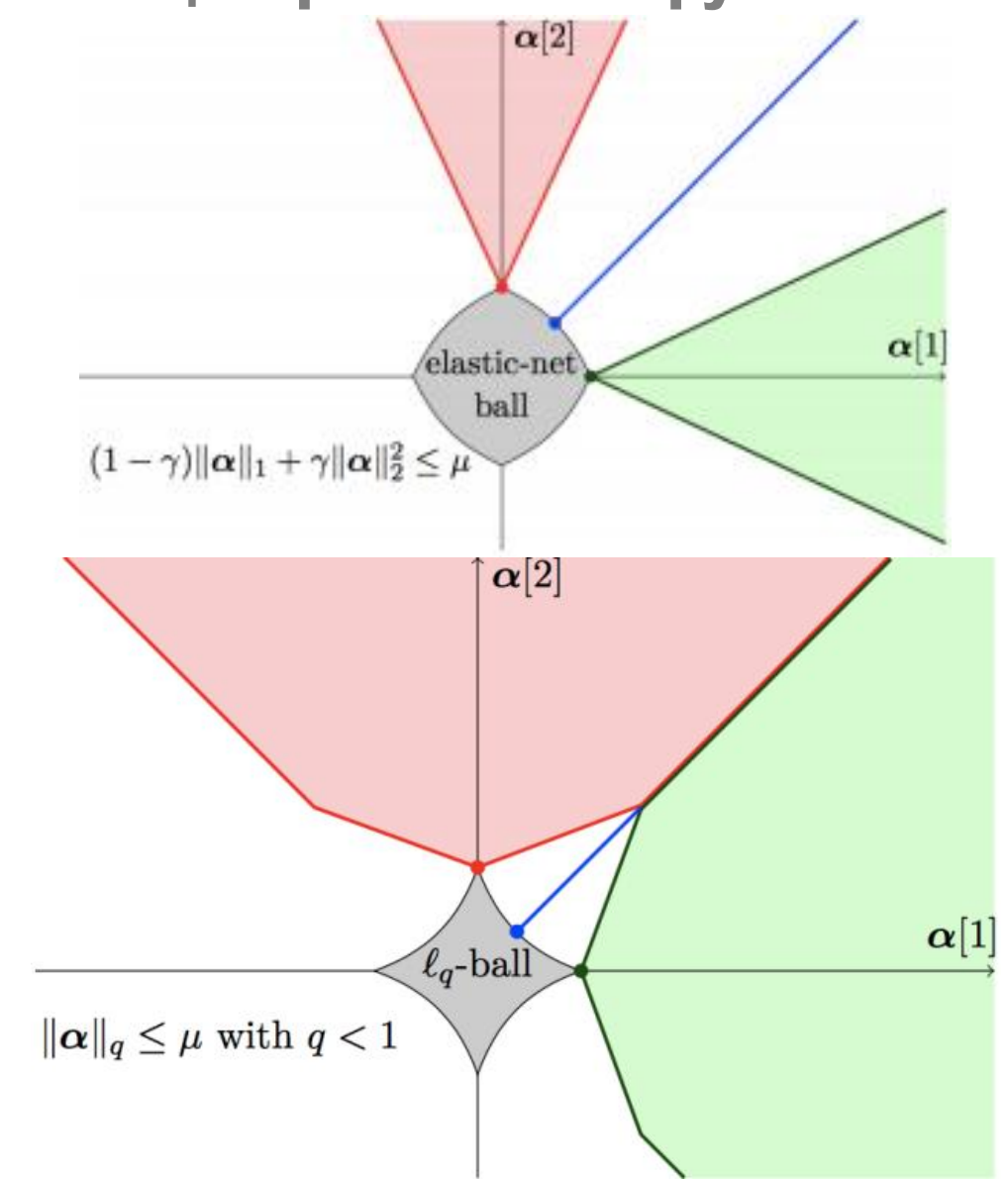
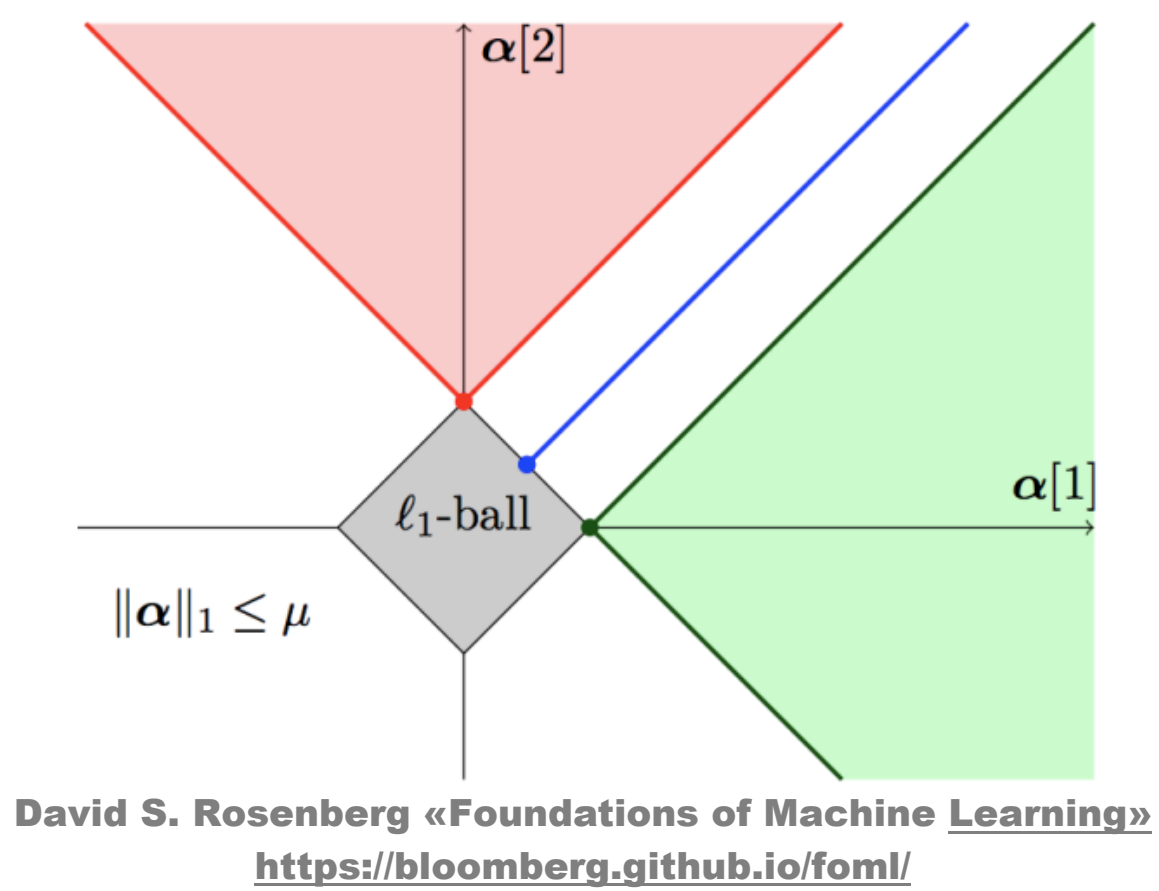
Геометрический смысл Elastic Net



на практике часто модель и не может зависеть от небольшого числа переменных

Эффект разреженности

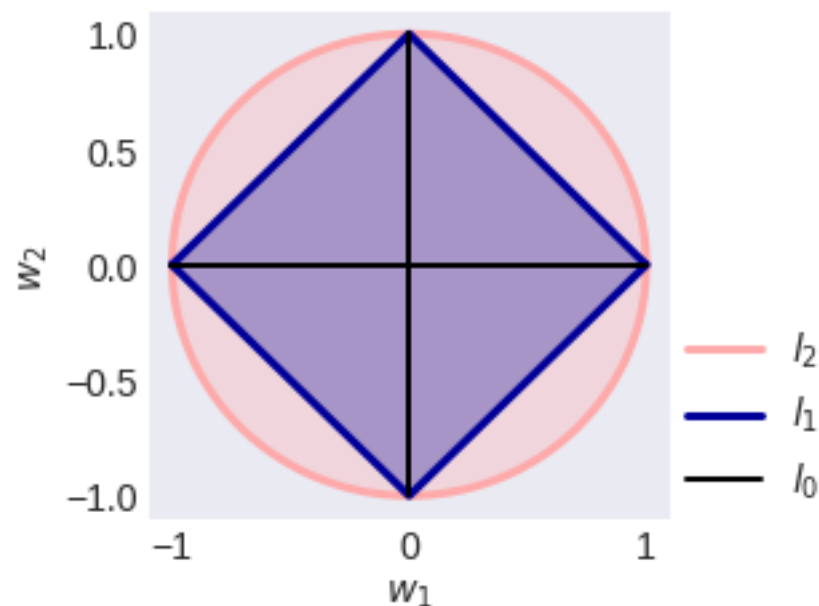
если линии уровня оптимизируемой функции – концентрические окружности...



Почему L1-норма \Rightarrow разреженность

1. Больше вероятность, что линии уровней функции ошибки касаются области ограничений в точках с нулевыми координатами, **см. рис.**

2. L1-норма больше похожа на L0, чем L2



$$\|w\|_0 = |\{t \mid w_t \neq 0\}|$$

**При увеличении коэффициента регуляризации веса стремятся к нулю
Обеспечивается автоматическая селекция признаков!**

Регуляризация \Rightarrow упрощение

Соблюдение принципа Оккама

регуляризация \Rightarrow зануление коэффициентов \Rightarrow упрощение модели

**В целом, неверно, что чем меньше коэффициентов, тем проще модель,
но у нас линейная модель..**

потом будет обоснование регуляризации с помощью вероятностных предположений

Проблема вырожденности / плохой обусловленности матрицы

$$w = (X^T X)^{-1} X^T y$$

Решения:

1. Регуляризация
- 2. Селекция (отбор) признаков**
- 3. Уменьшение размерности (в том числе, PCA)**
- 4. Увеличение выборки**

Селекция признаков в линейной регрессии **отдельная тема**

Какие признаки включить в модель

$$a(X_1, \dots, X_n) = w_0 + w_1 X_1 + \dots + w_n X_n$$

Маленький обзор стратегий:

- 1 стратегия – перебор – умный перебор подмножества признаков**
- 2 стратегий – оценка – оценка качества признаков (фильтры)**
- 3 стратегия – автомат – встроенные методы (ex: LASSO)**

Обоснование необходимости селекции

- **Проблема вырожденности в линейной регрессии**
 - **Проблема «почти дубликатов»**
- **Уменьшение модели и интерпретация**
- **Уменьшение стоимости данных**

Проблема вырожденности матрицы

$$w = (X^T X)^{-1} X^T y$$

Решения:

1. Регуляризация
2. Селекция (отбор) признаков
3. Уменьшение размерности (в том числе, PCA)
4. Увеличение выборки

	x1	x2	x3	y		x1-x2	y
0	0.44	0.62	0.51	-0.25	0	-0.18	-0.25
1	0.03	0.53	0.07	-0.51	1	-0.50	-0.51
2	0.55	0.13	0.43	0.41	2	0.42	0.41
3	0.44	0.51	0.10	0.04	3	-0.07	0.04
4	0.42	0.18	0.13	0.12	4	0.24	0.12
5	0.33	0.79	0.60	-0.45	5	-0.46	-0.45



обоснование необходимости аналогично селекции

Проблема вырожденности матрицы

$$w = (X^T X)^{-1} X^T y$$

Решения:

1. Регуляризация
2. Селекция (отбор) признаков
3. Уменьшение размерности (в том числе, PCA)
4. **Увеличение выборки**

на модельном примере:

$$m \leq n \quad \Rightarrow \quad \text{rg}(X^T X)_{(n+1) \times (n+1)} < n + 1$$

**+ при увеличении выборки могут исчезнуть
линейные зависимости между столбцами**

Линейная регрессия: градиентный метод обучения

недостатки прямого...

работа с большими матрицами (тем более обращение)

В лекции «оптимизация»...

$$\frac{1}{2} \sum_{i=1}^m (a(x_i | w) - y_i)^2 \rightarrow \min$$

$$w^{(t+1)} = w^{(t)} - \eta \sum_{i=1}^m (a(x_i | w^{(t)}) - y_i) \frac{\partial a(x_i | w^{(t)})}{\partial w}$$

Gradient Descent

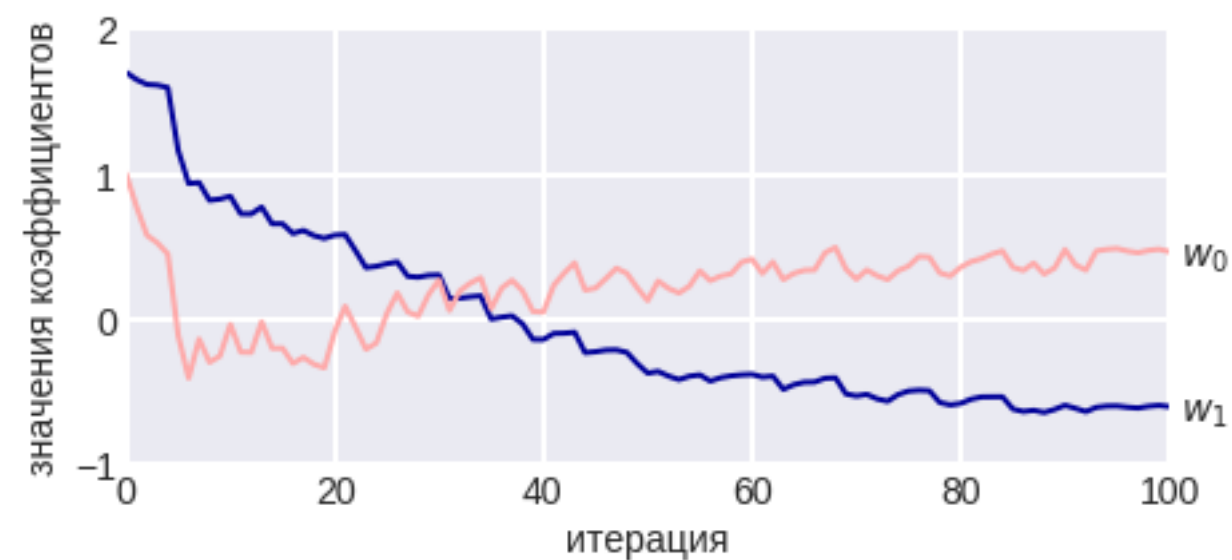
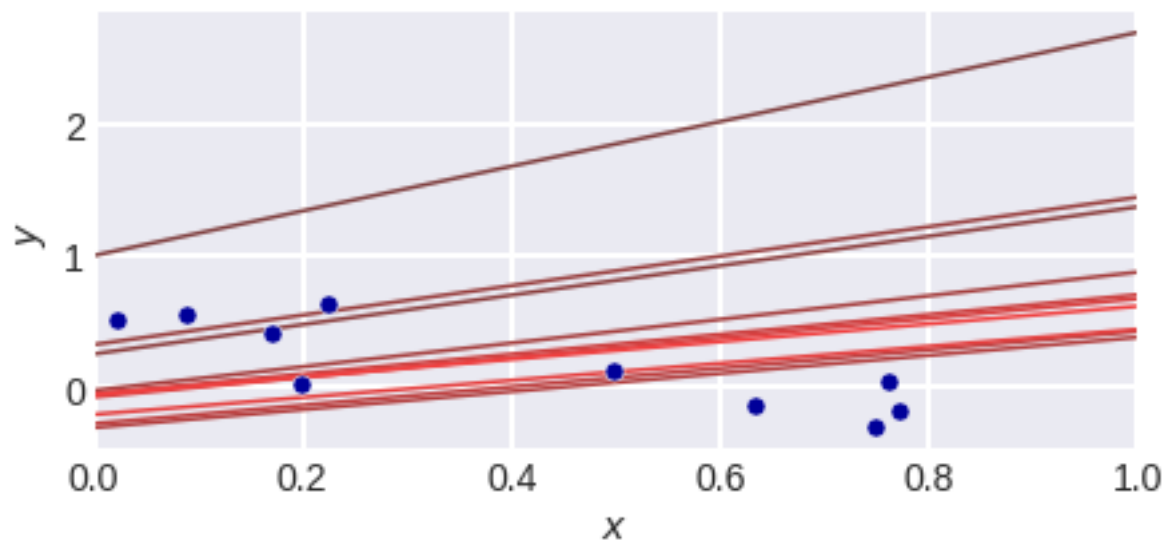
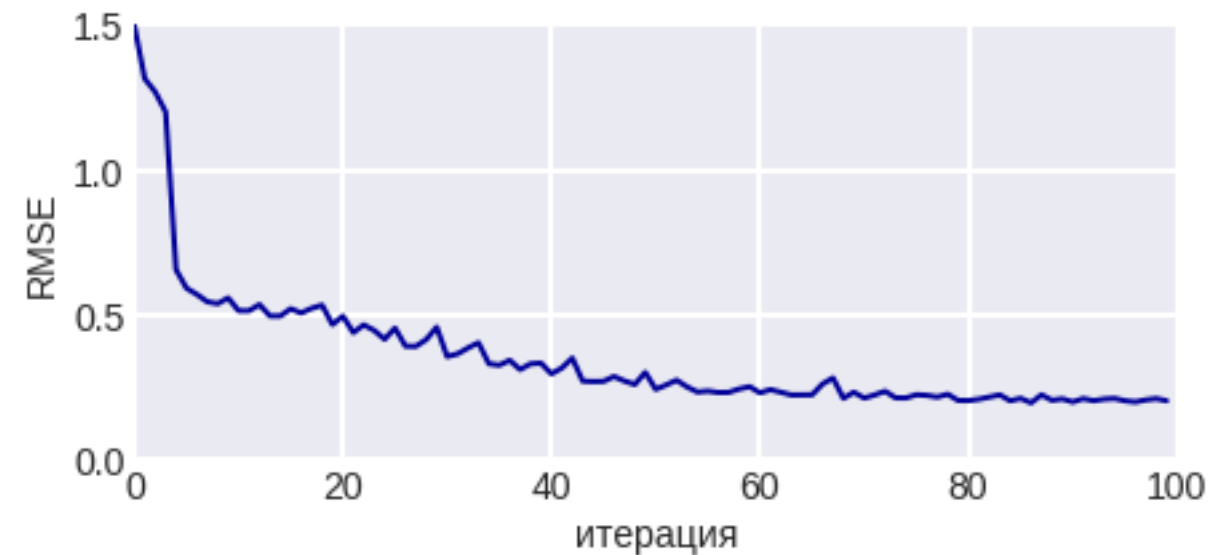
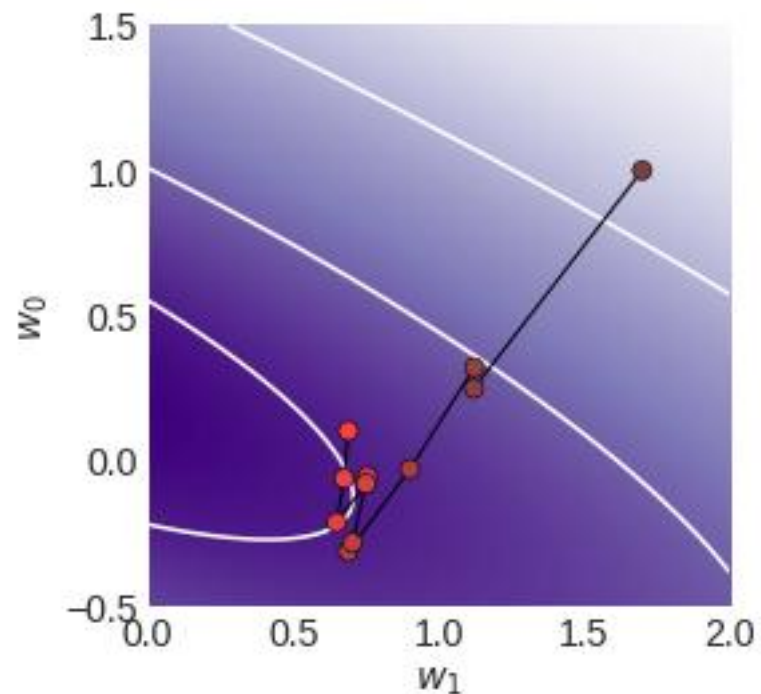
$$a(x | w) = w^T x$$

$$w^{(t+1)} = w^{(t)} - \eta \sum_{i=1}^m (a(x_i | w^{(t)}) - y_i) x_i$$

Stochastic Gradient Descent

$$w^{(t+1)} = w^{(t)} - \eta_t (a(x_i | w^{(t)}) - y_i) x_i$$

Линейная регрессия: градиентный метод обучения



Реализация в `scikit-learn`

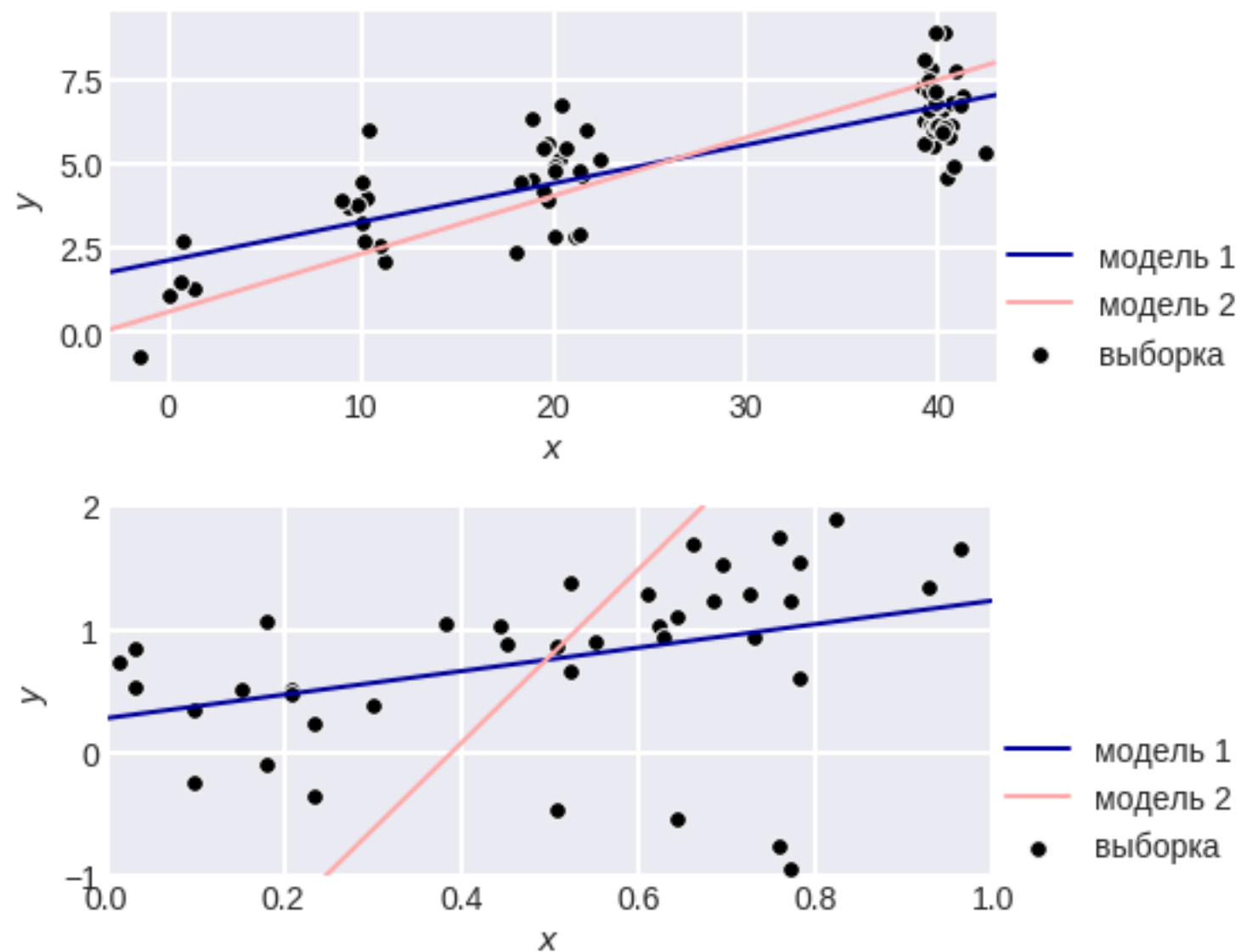
`sklearn.linear_model.Ridge`

<code>alpha=1.0</code>	Коэффициент регуляризации, больше – сильнее (в отличие от других функций)
<code>fit_intercept=True</code>	Использовать ли свободный член
<code>normalize=False</code>	Нормализация данных Игнорируется без свободного члена
<code>solver="auto"</code>	Метод оптимизации "auto", "svd", "cholesky", "lsqr", "sparse_cg", "sag", "saga"
<code>copy_X=True, max_iter=None, tol=0.001, random_state=None</code>	

```
sklearn.linear_model.Lasso(alpha=1.0, fit_intercept=True, normalize=False,
precompute=False, copy_X=True, max_iter=1000, tol=0.0001, warm_start=False,
    positive=False, random_state=None, selection="cyclic")
```

```
sklearn.linear_model.ElasticNet(alpha=1.0, l1_ratio=0.5, fit_intercept=True,
    normalize=False, precompute=False, max_iter=1000, copy_X=True, tol=0.0001,
    warm_start=False, positive=False, random_state=None, selection="cyclic")
```

Две регрессии



Чем отличаются модели 1 и 2?

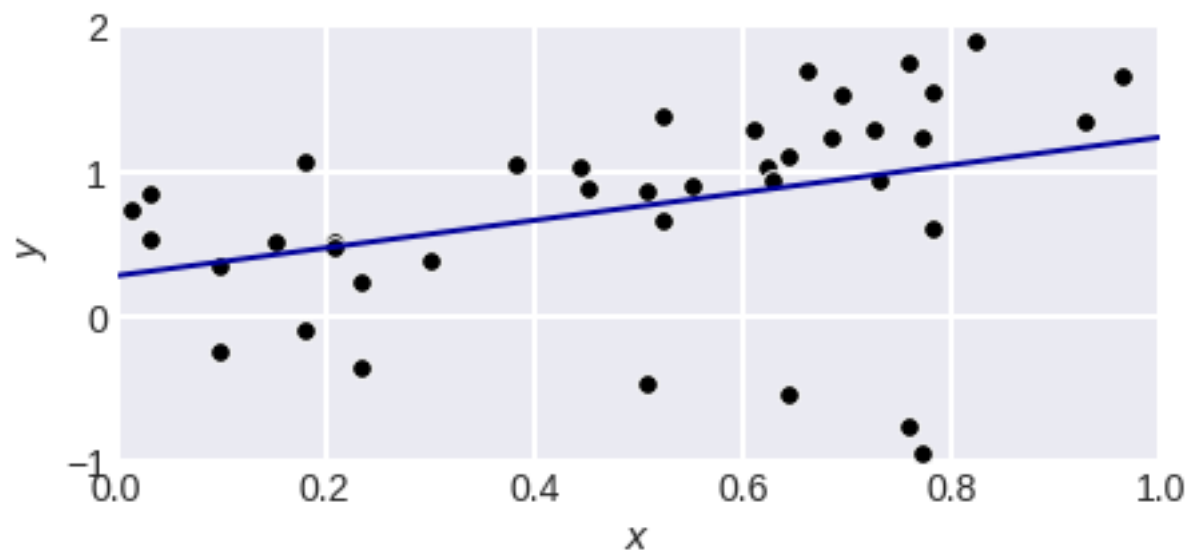
Две регрессии: $y(x)$ vs $x(y)$

разные задачи $y(x)$ и $x(y)$, хотя зависимость линейная

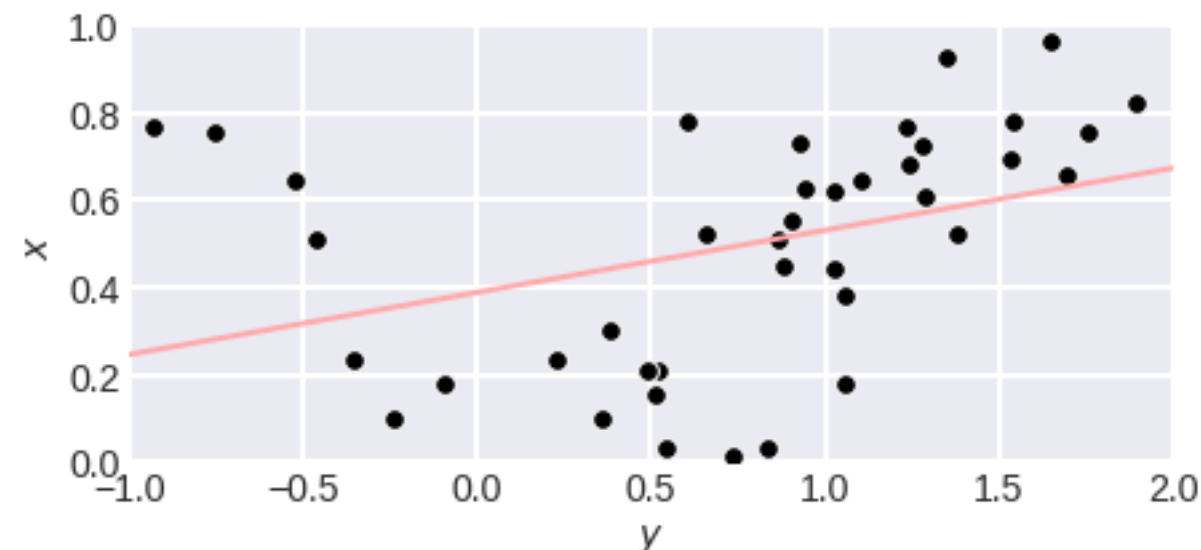
$$Y = w_0 + w_1 X_1$$

$$X_1 = w_0 + w_1 Y$$

$$\left\| \begin{bmatrix} x_1 & 1 \\ \dots & \dots \\ x_m & 1 \end{bmatrix} \begin{pmatrix} w_1 \\ w_0 \end{pmatrix} - \begin{pmatrix} y_1 \\ \dots \\ y_m \end{pmatrix} \right\|_2^2 \rightarrow \min$$

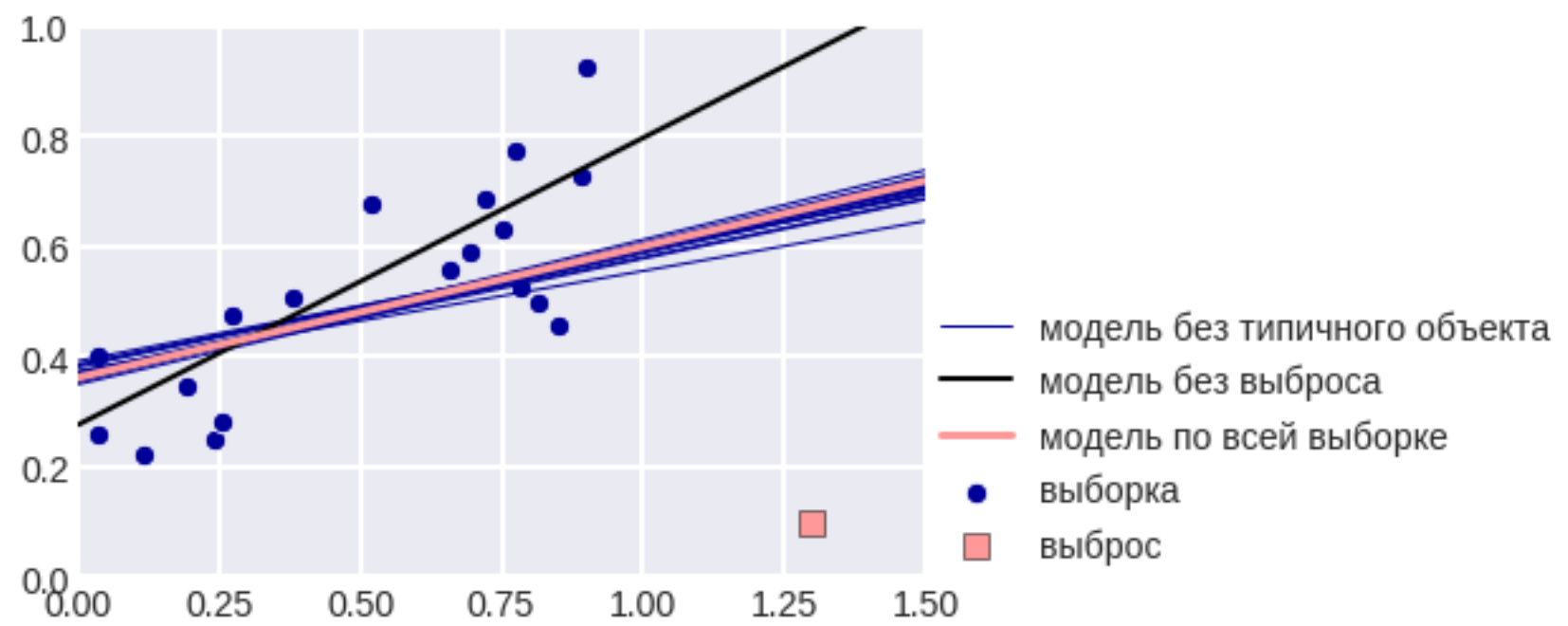


$$\left\| \begin{bmatrix} y_1 & 1 \\ \dots & \dots \\ y_m & 1 \end{bmatrix} \begin{pmatrix} w_1 \\ w_0 \end{pmatrix} - \begin{pmatrix} x_1 \\ \dots \\ x_m \end{pmatrix} \right\|_2^2 \rightarrow \min$$



есть и «промежуточная стратегия» – **дальше РСА**

Линейная регрессия – неустойчивость к выбросам



Ошибка с весами

Если у каждого объекта есть цена ошибки...

$$\sum_{i=1}^m v_i^2 (y_i - w^T x_i)^2 + \dots = \sum_{i=1}^m (v_i y_i - w^T (v_i x_i))^2 + \dots \rightarrow \min$$

небольшая переформулировка задачи:

$$\{(x_1, y_1), \dots, (x_m, y_m)\} \rightarrow \{(v_1 x_1, v_1 y_1), \dots, (v_m x_m, v_m y_m)\}$$

$$(y - Xw)^T V^T V (y - Xw) = \|Vy - VXw\|_2^2 \rightarrow \min_w$$

$$V = \text{diag}(v_1, \dots, v_m)$$

$$w = (X^T V X)^{-1} X^T V y$$

Ошибка с весами

$$\{(x_1, y_1), \dots, (x_m, y_m)\} \rightarrow \{(v_1 x_1, v_1 y_1), \dots, (v_m x_m, v_m y_m)\}$$

Как реализовать:

- 1) перейти к новым данным («испорченными весам»)**
- 2) если веса целые числа – можно продублировать объекты**
- 3) если веса из отрезка $[0, 1]$ – при численном градиентном решении можно выбирать следующий объект с соответствующей вероятностью**

Устойчивая регрессия (Robust Regression)

0. Инициализация весов объектов

$$v = (v_1, \dots, v_m) = (1/m, \dots, 1/m)$$

1. Цикл

1.1. Настроить алгоритм, учитывая веса объектов

$$a = \text{fit}(\{x_i, y_i, v_i\})$$

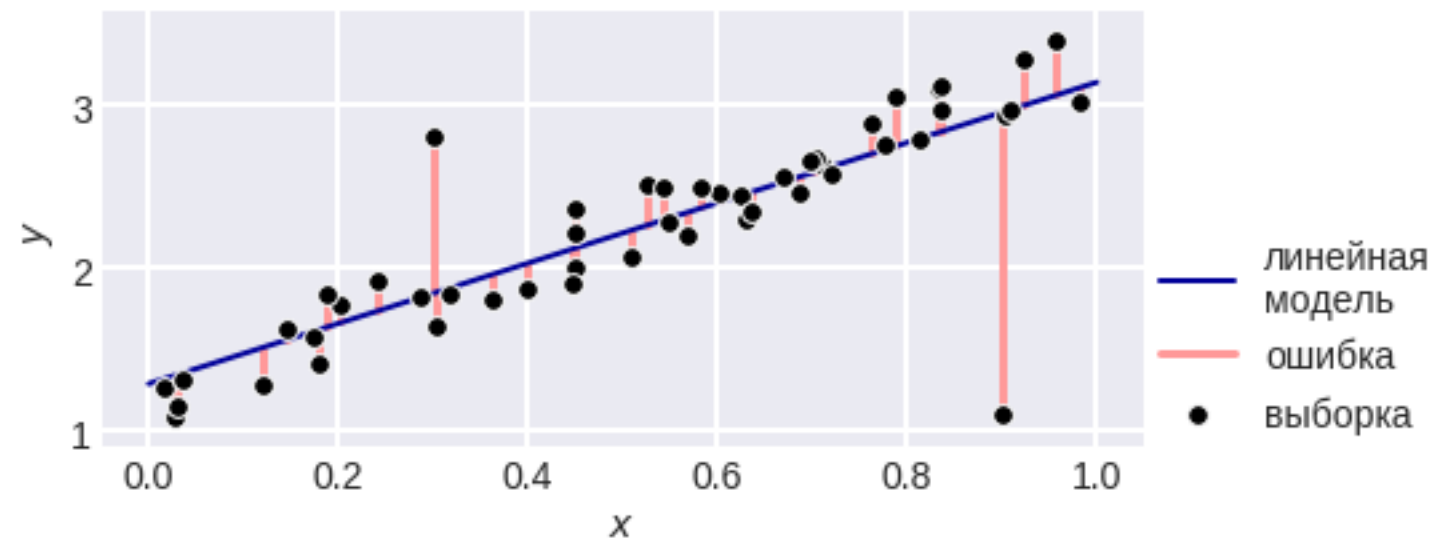
1.2. Вычислить ошибки на обучении

$$\varepsilon_i = a(x_i) - y_i$$

1.3. Пересчитать веса объектов

$$v_i = \exp(-\varepsilon_i^2)$$

нормировать на сумму

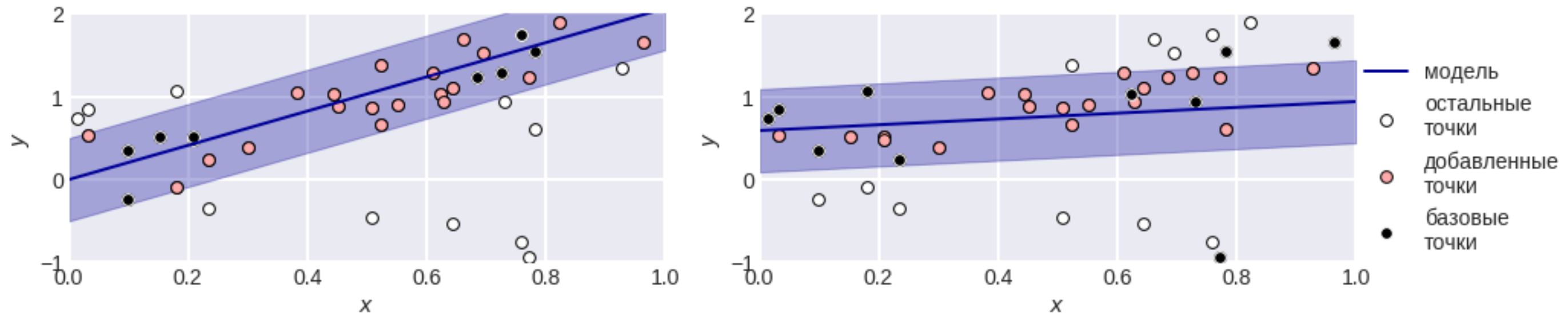


можно использовать любую
регрессионную модель

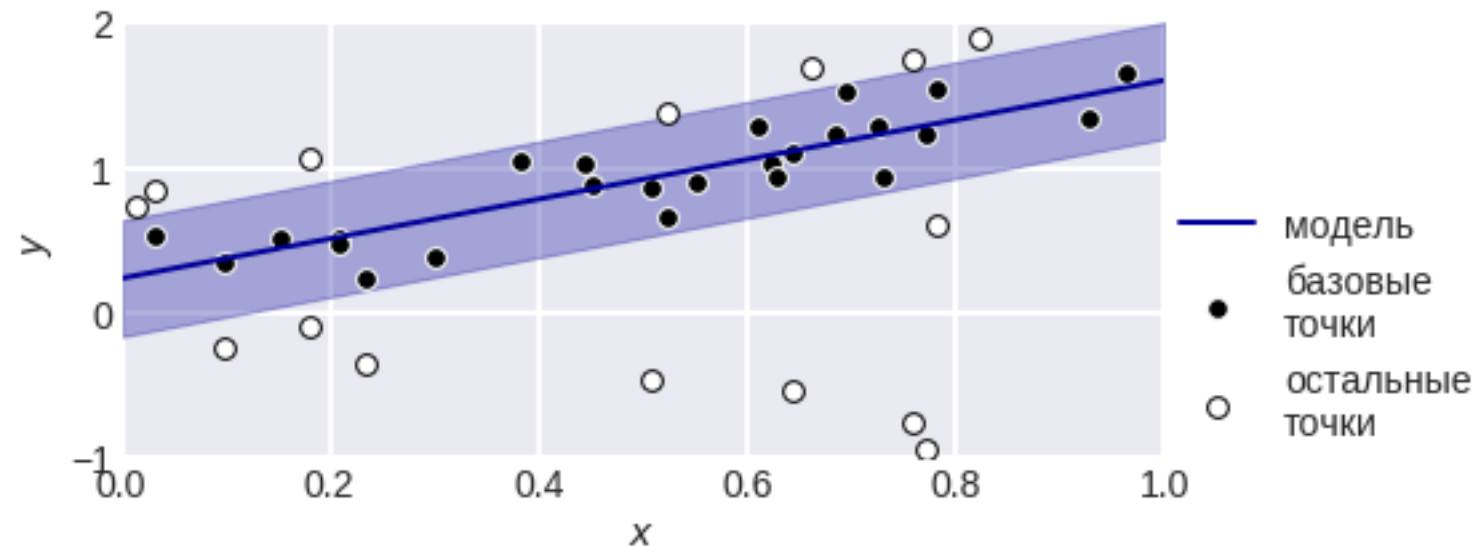
при пересчёте весов можно использовать
другую невозрастающую функцию;
можно (иногда нужно) нормировать

RANdom SAmple Consensus (RANSAC)

- **несколько раз**
 - **выбрать случайное подмножество точек – базовое (inliers)**
 - **обучить модель на базовом подмножестве**
 - **найти все точки, которые хорошо предсказываются моделью**
например, ошибка не больше ε
 - **пополнить ими базовое множество**
 - **(если добавили много) переобучить модель на новом множестве**
- **выбрать модель с наименьшей ошибкой**



Минутка кода: RANSAC в scikit-learn



```
from sklearn.linear_model import RANSACRegressor
# Robustly fit linear model with RANSAC algorithm
ransac = RANSACRegressor()
ransac.fit(x[:, np.newaxis], y)
inlier_mask = ransac.inlier_mask
outlier_mask = np.logical_not(inlier_mask)
```

Минутка кода: RANSAC в scikit-learn

```
sklearn.linear_model import RANSACRegressor
```

base_estimator=None	Базовый алгоритм (по умолчанию – линейная регрессия)
min_samples=None	Число / доля базовых объектов ($n+1$)
residual_threshold=None	Порог для пополнения базового множества (MAD(y))
max_trials=100	Число итераций
stop_n_inliers	Остановить вычисления, если найдено столько базовых точек
loss="absolute_loss"	Как оценивать ошибку

```
is_data_valid=None, is_model_valid=None, max_skips=inf,
stop_score=inf, stop_probability=0.99, lossrandom_state=None
```

Реализация в scikit-learn немного отличается от некоторых описаний:

Качество = число базовых (inliers) объектов;
 лучшая модель выбирается по числу базовых,
 если у нескольких моделей число совпадает,
 тогда сравнивается ошибка на всей выборке.

Ошибка и её оценка (невязка)

вернёмся к линейной регрессии: есть линейная зависимость с точностью до шума

$$y = Xw^* + \varepsilon$$

потом рассмотрим такую **вероятностную постановку**, сами решаем так:

$$y \approx a = Xw$$

с точностью до оценки шума

$$\hat{\varepsilon} = y - a$$

для нашего (оптимального) решения $w = (X^T X)^{-1} X^T y$ оценка ошибки (невязка)

$$\begin{aligned}\hat{\varepsilon} &= y - a = y - Xw = y - \underbrace{X(X^T X)^{-1} X^T}_H y = (I - H)y = \\ &= (I - H)(Xw^* + \varepsilon) = \underbrace{Xw^* - HXw^*}_{Xw^* - Xw^* = 0} + (I - H)\varepsilon = (I - H)\varepsilon\end{aligned}$$

Ошибка и её оценка (невязка)

получили, что

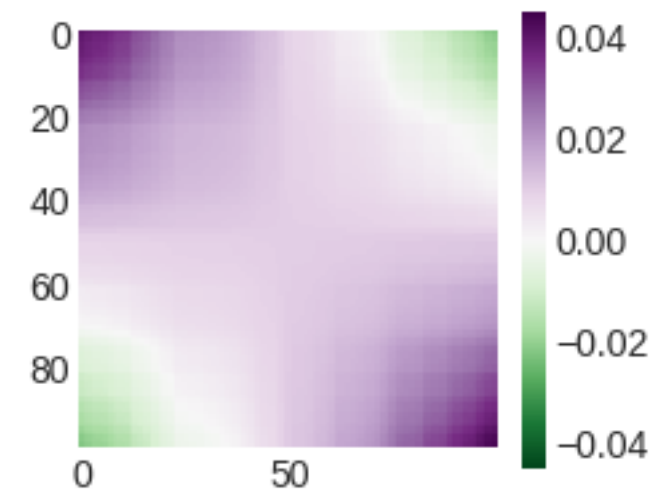
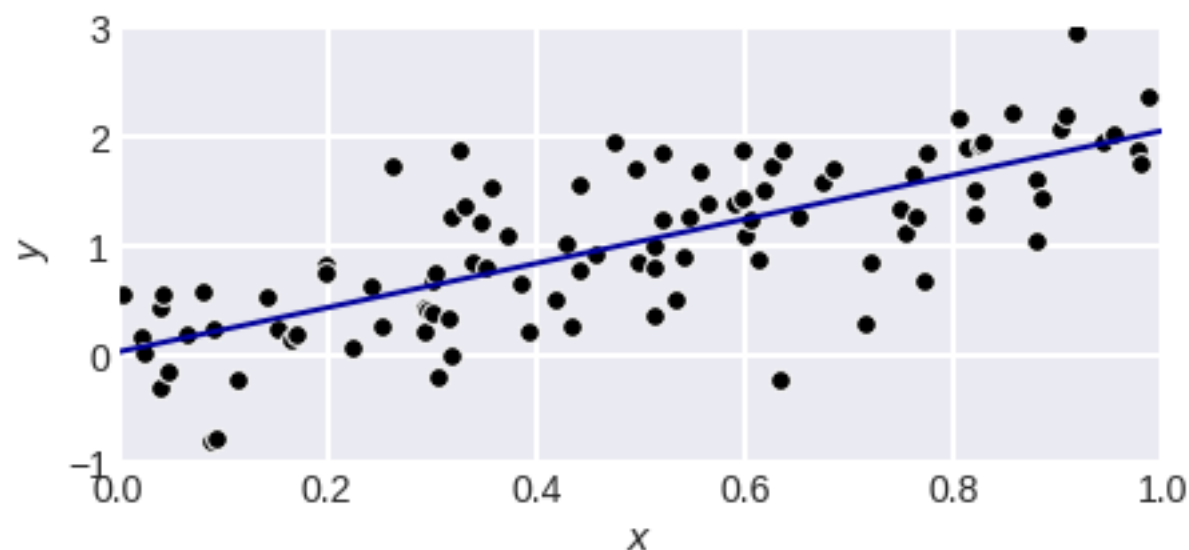
$$\hat{\varepsilon} = (I - H)\varepsilon,$$

где $H = X(X^T X)^{-1} X^T$

– projection (hat) matrix

Кстати, $a = H y$

таким образом,



матрица H

- невязки коррелируют (даже если «истинные ошибки» – нет)
- но нет корреляции с целевым значением! (если ошибка не коррелирует)
- невязки в разном масштабе

иногда стандартизуют (во многих пакетах, но за этим следить!)

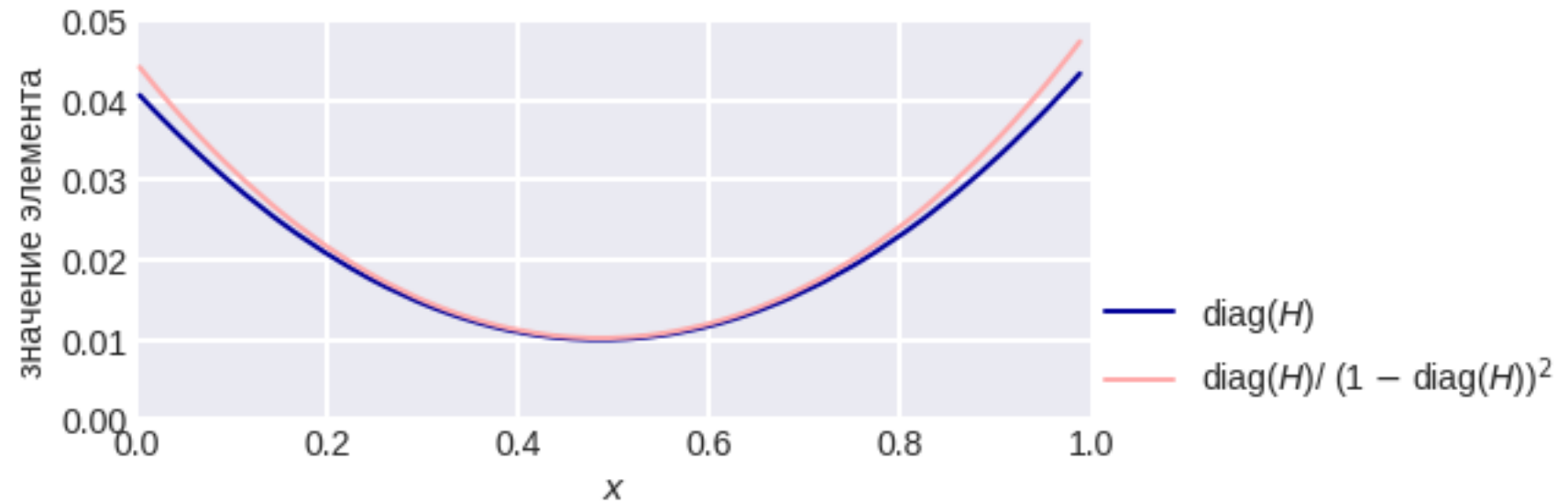
$$\hat{\varepsilon}_{[t]} / \sqrt{1 - h_{tt}}$$

справедливости ради – м.б. не слишком заметный эффект

Ошибка и её оценка (невязка)

Для справки:

$$h_{tt} = \frac{1}{m} + \frac{(x_t - \bar{x})^2}{\sum_{i=1}^m (x_i - \bar{x})^2}$$



тут X распределён равномерно, шум нормальный

Если ввести Cook's distance – как сильно точка влияет на решение (зачем?)

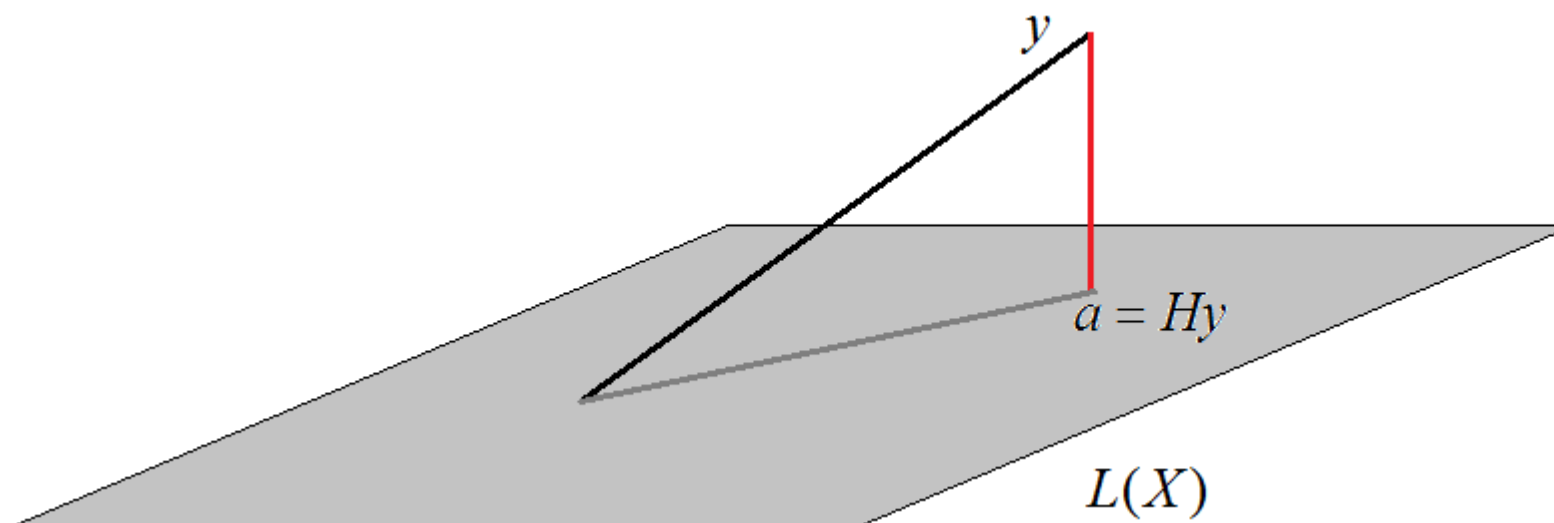
$$D_j = \text{const} \cdot \sum_{i=1}^m (a(x_i | X_{\text{train}}) - a(x_i | X_{\text{train}} \setminus \{x_j\}))^2$$

то можно вывести:
$$D_j = \text{const} \cdot \frac{h_{jj}}{(1 - h_{jj})^2} \hat{\varepsilon}_{[j]}^2$$

Проекционная ($\hat{}$) матрица

$$H = X(X^T X)^{-1} X^T$$
$$L(X)$$

$a = Hy \in L(X)$ – из линейной комбинация столбцов матрицы X



проверьте проекцию на ортогональность

Линейная регрессия: связь с SVD

Рассмотрим сразу гребневую регрессию:

оптимальные веса $w = (X^T X + \lambda I)^{-1} X^T y$

воспользуемся SVD (в полном разложении $(m \times m) \cdot (m \times n) \cdot (n \times n)$):

$X = U \Lambda V^T$, тогда

$$w = (V \Lambda^T U^T U \Lambda V^T + \lambda I)^{-1} V \Lambda^T U^T y = (V (\Lambda^T \Lambda + \lambda I) V^T)^{-1} V \Lambda^T U^T y =$$

$$= V \underbrace{(\Lambda^T \Lambda + \lambda I)^{-1} \Lambda^T}_{\text{diag}(\lambda_1/(\lambda_1^2 + \lambda), \dots, \lambda_k/(\lambda_k^2 + \lambda))_{n \times m}} U^T y = \sum_{j=1}^k \frac{\lambda_j \cdot u_j^T y}{\lambda_j^2 + \lambda} v_j$$

$$k = \min(m, n)$$

Пояснение про матрицу $(\Lambda^T \Lambda + \lambda I)^{-1} \Lambda^T$

$$\left(\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \\ 0 & 0 \end{bmatrix} + \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \end{bmatrix} = \begin{bmatrix} \frac{\lambda_1}{\lambda_1^2 + \lambda} & 0 & 0 \\ 0 & \frac{\lambda_2}{\lambda_1^2 + \lambda} & 0 \end{bmatrix}$$

$$\left(\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \end{bmatrix} + \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \frac{\lambda_1}{\lambda_1^2 + \lambda} & 0 \\ 0 & \frac{\lambda_2}{\lambda_1^2 + \lambda} \\ 0 & 0 \end{bmatrix}$$

Линейная регрессия: связь с SVD

$$w = \sum_{j=1}^k \frac{\lambda_j \cdot u_j^T y}{\lambda_j^2 + \lambda} v_j$$

веса – линейные комбинации столбцов V

коэффициенты – скалярное произведение столбцов U и целевого столбца

виден эффект от регуляризации! при больших λ зануляются коэффициенты без регуляризации, когда $\lambda_j \approx 0$, веса м.б. большими!

Число обусловленности матрицы $(X^T X + \lambda I)$

$$\frac{\lambda_{\max}(X^T X + \lambda I)}{\lambda_{\min}(X^T X + \lambda I)} = \frac{\lambda_{\max}^2 + \lambda}{\lambda_{\min}^2 + \lambda} \xrightarrow{\lambda \rightarrow +\infty} 1$$

Линейная регрессия: связь с SVD

Ещё одно упражнение – посмотрим на проекционную матрицу

$$H = X(X^T X)^{-1} X^T$$

$$\begin{aligned} H &= U \Lambda V^T (V \Lambda^T \mathbf{U}^T \mathbf{U} \Lambda V^T)^{-1} V \Lambda^T U^T = \\ &= U \Lambda V^T (V \Lambda^T \Lambda V^T)^{-1} V \Lambda^T U^T = \\ &= U \Lambda \mathbf{V}^T \mathbf{V} (\Lambda^T \Lambda)^{-1} \mathbf{V}^T \mathbf{V} \Lambda^T U^T = \\ &= U \Lambda (\Lambda^T \Lambda)^{-1} \Lambda^T U^T \end{aligned}$$

и тут тонкость (мы её пользовались на предыдущем слайде),
что зелёная матрица может не быть единичной!

$$\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \\ 0 & 0 \end{bmatrix} \left(\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \\ 0 & 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Поиск ортогонального соответствия (Orthogonal matching pursuit)

Пусть X_J – матрица, образованная столбцами из множества $J \subseteq \{1, 2, \dots, n\}$
Изначально $J = \emptyset$ (т.е. входит лишь нулевой столбец или ничего не входит)

на каждой итерации решаем задачу

$$\|y - X_{J \cup \{j\}} w\|_2^2 \rightarrow \min_{w, j}$$

(тут возможны модификации)

и наращиваем множество используемых признаков

$$J \leftarrow J \cup \{j\}$$

пока $\|y - X_J w\|_2^2 \geq \varepsilon$ или $|J| \leq k$

идею можно применять и для других семейств алгоритмов

Поиск ортогонального соответствия (Orthogonal matching pursuit)

Пусть изначально из матрицы удалены столбцы-дубликаты,
все столбцы пронормированы

вместо $\|y - X_{J \cup \{j\}} w\|_2^2 \rightarrow \min_{w, j}$

решаем такую последовательность задач

$$j = \arg \max_{j \notin J} (X[:, j]^T (y - X_J w))$$

$$J \leftarrow J \cup \{j\}$$

$$w \leftarrow \arg \min_w \|y - X_J w\|_2^2$$

Линейные скоринговые модели в задаче бинарной классификации

Пусть $X = \mathbb{R}^n$, $Y = \{0, 1\}$

Как решать задачи классификации с помощью линейной модели:
будем получать вероятность принадлежности к классу 1

$$a(x) \in [0, 1]$$

Любая линейная функция на \mathbb{R}^n будет получать значения в \mathbb{R} ,
поэтому нужна деформация (transfer function):

$$\sigma : \mathbb{R} \rightarrow [0, 1]$$

Решаем задачу так:

$$\sigma(w^T x)$$

проекция на одномерное пространство и деформация

Функции деформации

В логистической регрессии

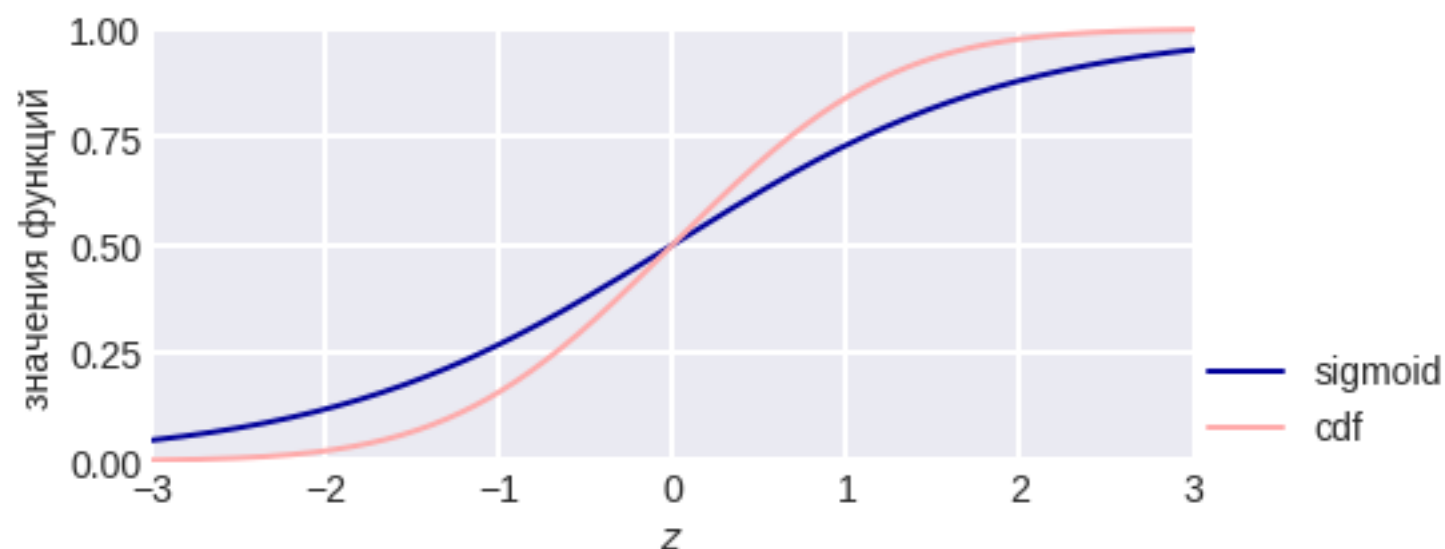
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Логистическая функция (сигмоида)

В Probit-регрессии

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z \exp(-t^2 / 2) \partial t$$

Normal Cumulative distribution function



Логистическая регрессия

$$P(Y = 1 | x) = \sigma(z) = \frac{1}{1 + e^{-z}} \in (0, 1),$$

$$z = w^T x = w_0 + w_1 X_1 + \dots + w_n X_n,$$

$$\log \left(\frac{\sigma(z)}{1 - \sigma(z)} \right) = z$$

– монотонное преобразование, которое называют **logit-transformation**

Решаем задачу классификации, но метод называется логистическая **регрессия**

Логистическая регрессия: интересные свойства сигмоиды

оценка вероятности принадлежности к классу 0 будет «симметрично записываться»

$$\sigma(z) + \sigma(-z) = \frac{1}{1 + e^{-z}} + \frac{1}{1 + e^{+z}} = \frac{e^{+z}}{1 + e^{+z}} + \frac{1}{1 + e^{+z}} = 1$$

поскольку $P(Y = 1 | x) + P(Y = 0 | x) = 1$, то

$$P(Y = 0 | x) = \sigma(-z)$$

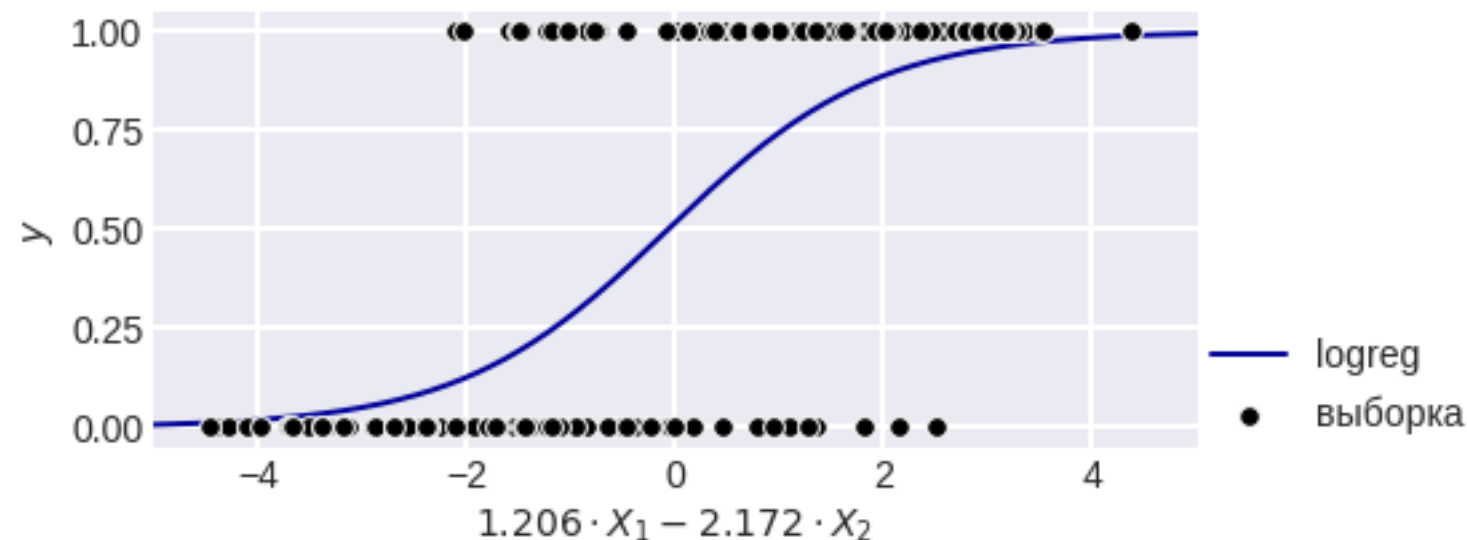
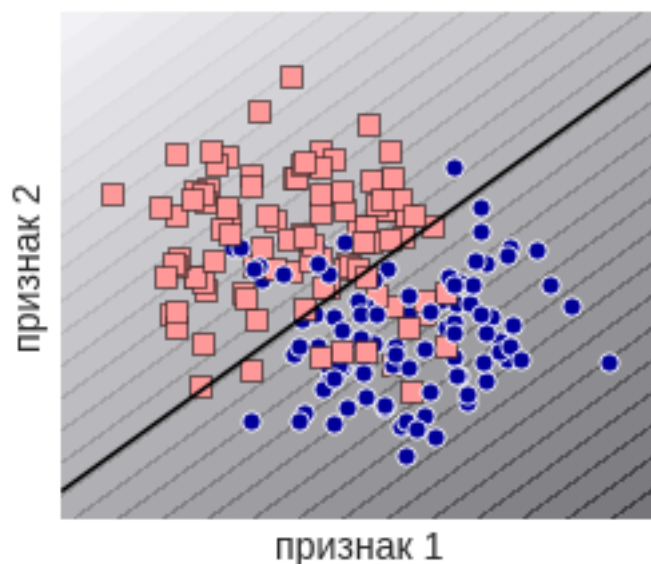
производная выражается через функцию

$$\frac{\partial \sigma(z)}{\partial z} = \left(\frac{1}{1 + e^{-z}} \right)' = \frac{-(-e^{-z})}{(1 + e^{-z})^2} = \frac{1 + e^{-z} - 1}{(1 + e^{-z})^2} = \frac{1}{1 + e^{-z}} \left(1 - \frac{1}{1 + e^{-z}} \right) = \sigma(z)(1 - \sigma(z))$$

нам пока это свойство не понадобится

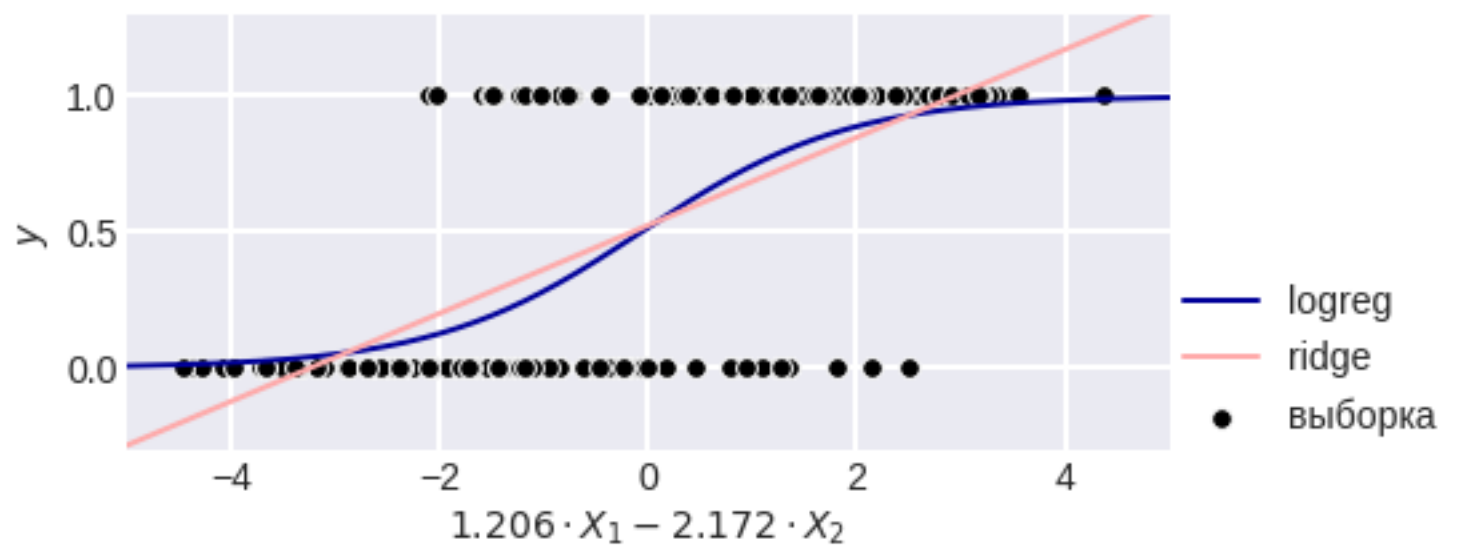
Геометрический смысл логистической регрессии

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(X, y)
a = model.predict_proba(X_test)[:,1]
```

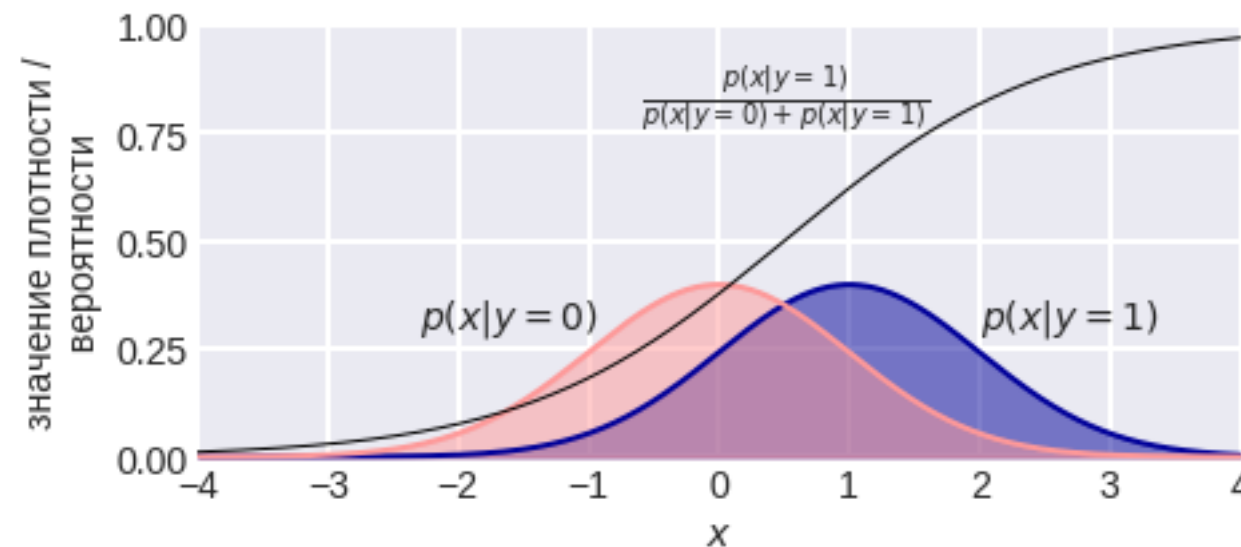


$z = w_0 + w_1 X_1 + \dots + w_n X_n$ – проекция на прямую (один признак)
в однопризнаковом случае надо решить задачу классификации

Чем логистическая регрессия лучше регрессии



Откуда берётся сигмоида



$$p(x | y = t) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_t)^T \Sigma^{-1} (x - \mu_t)\right)$$

нормальное распределение с одинаковыми матрицами ковариации

$$p(y = t | x) = \frac{p(x | y = t)p(y = t)}{p(x | y = 0)p(y = 0) + p(x | y = 1)p(y = 1)}$$

Откуда берётся сигмоида
считаем классы равновероятными

$$\begin{aligned}
 p(y = t \mid x) &= \frac{\frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_t)^\top \Sigma^{-1}(x - \mu_t)\right)}{\sum_t \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_t)^\top \Sigma^{-1}(x - \mu_t)\right)} = \\
 &= \frac{1}{1 + \exp\left(+\frac{1}{2}(x - \mu_t)^\top \Sigma^{-1}(x - \mu_t) - \frac{1}{2}(x - \mu_{1-t})^\top \Sigma^{-1}(x - \mu_{1-t})\right)} = \\
 &= \frac{1}{1 + \exp\left(-\frac{1}{2}\mu_t^\top \Sigma^{-1}x - \frac{1}{2}x^\top \Sigma^{-1}\mu_t + \frac{1}{2}\mu_t^\top \Sigma^{-1}\mu_t + \frac{1}{2}\mu_{1-t}^\top \Sigma^{-1}x + \frac{1}{2}x^\top \Sigma^{-1}\mu_{1-t} - \frac{1}{2}\mu_{1-t}^\top \Sigma^{-1}\mu_{1-t}\right)} = \\
 &= \sigma(w^\top x + w_0)
 \end{aligned}$$

Обучение логистической регрессии

Метод максимального правдоподобия

$$L(w_0, \dots, w_n) = \prod_{i: y_i=1} \sigma(w^T x_i) \prod_{i: y_i=0} (1 - \sigma(w^T x_i)) \rightarrow \max$$

здесь тоже выпуклая задача оптимизации
логарифмируем....

$$\log L(w_0, \dots, w_n) = \sum_{i: y_i=1} \log(\sigma(w^T x_i)) + \sum_{i: y_i=0} \log(\sigma(-w^T x_i)) \rightarrow \max$$

для удобства записи $y'_i = 2y_i - 1$, тогда

$$\log L = \sum_i \log(\sigma(y'_i w^T x_i)) = - \sum_i \log(1 + \exp(-y'_i w^T x_i)) \rightarrow \max$$

Обучение логистической регрессии

$$\log L = -\sum_i \log(1 + \exp(-y_i' w^T x_i)) \rightarrow \max$$

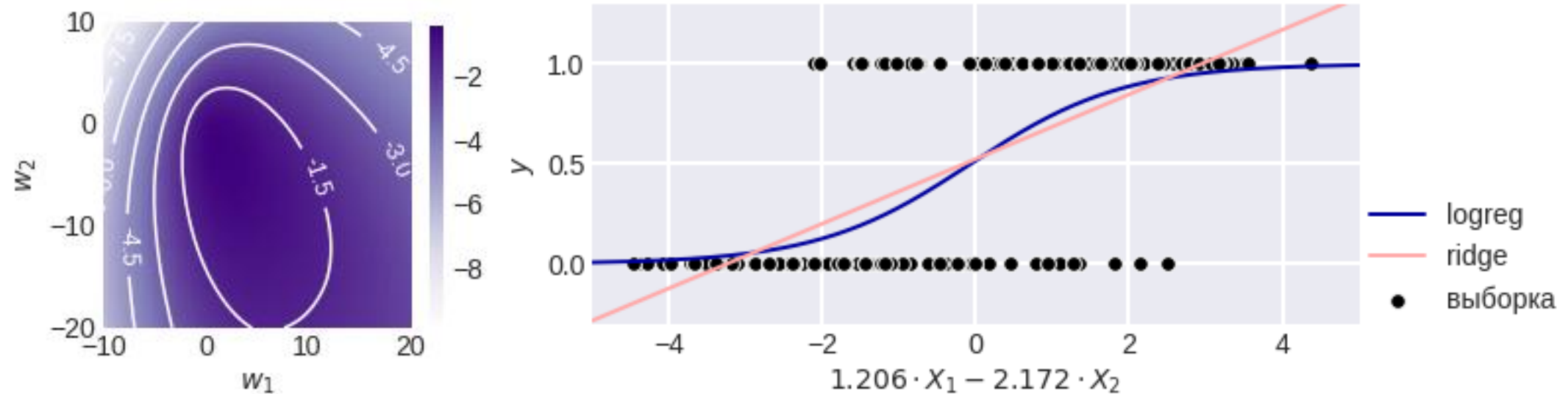
Полученное выражение называют «логистической функцией ошибки» (logistic loss)

Вычислим градиент

$$\begin{aligned} \nabla_w \log L &= -\sum_i \nabla_w \log(1 + \exp(-y_i' w^T x_i)) = \\ &= -\sum_i \frac{\exp(-y_i' w^T x_i)}{1 + \exp(-y_i' w^T x_i)} (-y_i' x_i) = \sum_i \frac{y_i' x_i}{1 + \exp(+y_i' w^T x_i)} = \sum_i \sigma(-y_i' w^T x_i) y_i' x_i \end{aligned}$$

Качество логистической регрессии

– логарифм правдоподобия
(потом будет соответствующая функция ошибки **logloss**)



метод **SGD** (запомним):

$$w \leftarrow w + \eta \sigma(-y'_i w^T x_i) y'_i x_i$$

Многоклассовая логистическая регрессия: Multiclass logreg / multinomial regression

Для j -го класса имеем свою функцию $w_j^T x$

хотим такие значения превращать в распределения

$$(w_1^T x, w_2^T x, \dots, w_l^T x) \in \mathbb{R}^l \rightarrow (p_1(x), p_2(x), \dots, p_l(x)) \in [0, 1]_{\Sigma=1}^l$$

в `glmnet` такой «симметричный вариант»:

$$P(Y = k \mid x) = \frac{\exp(w_{0k} + w_{1k}X_1 + \dots + w_{nk}X_n)}{\sum_{j=1}^l \exp(w_{0j} + w_{1j}X_1 + \dots + w_{nj}X_n)}$$

Такая функция называется **softmax**:

$$\text{softmax}(a_1, \dots, a_l) = \frac{1}{\exp(a_1) + \dots + \exp(a_l)} [\exp(a_1), \dots, \exp(a_l)]$$

Реализация в `scikit-learn`

`sklearn.linear_model.LogisticRegression`

<code>penalty="l2"</code>	Тип регуляризации Не все солверы поддерживают все типы
<code>dual=False</code>	Переход к двойственной задачи
<code>C=1.0</code>	Обратная величина к коэффициенту регуляризации
<code>fit_intercept=True</code>	Свободный член
<code>class_weight=None</code>	Веса классов
<code>solver="warn"</code>	Солвер "newton-cg", "lbfgs", "liblinear", "sag", "saga"
<code>warm_start=False</code>	Использовать ли предыдущие начальные условия
<code>l1_ratio</code>	Формализация штрафов для ElasticNet

`tol=0.0001, intercept_scaling=1, random_state=None, max_iter=100,
multi_class='warn', verbose=0, n_jobs=None`

Приложения

Банковский скоринг

Задачи с текстами

Бенчмарк для дебита нефти

Прогнозирование спроса

Почти любые промышленные задачи!

Банковский скоринг

Name	Description	Type
TCS_CUSTOMER_ID	Идентификатор клиента	ID
BUREAU_CD	Код бюро, из которого получен счет	numeric
BKI_REQUEST_DATE	Дата, в которую был сделан запрос в бюро	date
CURRENCY	Валюта договора (ISO буквенный код валюты)	string
RELATIONSHIP	Тип отношения к договору	string
	1 - Физическое лицо	
	2 - Дополнительная карта/Авторизованный пользователь	
	4 - Совместный	
	5 - Поручитель	
	9 - Юридическое лицо	
OPEN_DATE	Дата открытия договора	date
FINAL_PMT_DATE	Дата финального платежа (плановая)	date
TYPE	Код типа договора	string
	1 – Кредит на автомобиль	
	4 – Лизинг. Срочные платежи за наем/пользование транспортным средством, предприятием или оборудованием и т.п.	
	6 – Ипотека – ссудные счета, имеющие отношение к домам, квартирам и прочей недвижимости. Ссуда выплачивается циклично согласно договоренности до тех пор, пока она не будет полностью выплачена или возобновлена.	
	7 – Кредитная карта	
	9 – Потребительский кредит	
	10 – Кредит на развитие бизнеса	
	11 – Кредит на пополнение оборотных средств	
	12 – Кредит на покупку оборудования	
	13 – Кредит на строительство недвижимости	
	14 – Кредит на покупку акций (например, маржинальное кредитование)	
	99 – Другой	
PMT_STRING_84M	Дисциплина (своевременность) платежей. Строка составляется из кодов состояний счета на моменты передачи банком данных по счету в бюро, первый символ - состояние на дату PMT_STRING_START, далее последовательно в порядке убывания дат.	string
	0 – Новый, оценка невозможна	
	X – Нет информации	
	1 – Оплата без просрочек	
	A – Просрочка от 1 до 29 дней	

Банковский скоринг

По описанию и истории клиента → вероятность (оценка) возврата кредита

Нужна логистическая регрессия

есть возможность получать вещественное число в виде ответа

есть более мощные методы (**на решающих деревьях**),

но здесь полезна интерпретация

Все категориальные признаки – ONE-перекодировка

Банковский скоринг

Если решение сводится к

$$a(x) = 1 / (1 + \exp(-(w_0 + w_1 X_1 + \dots + w_n X_n)))$$

где все признаки бинарные, то мы составляем **скоринговую карту**

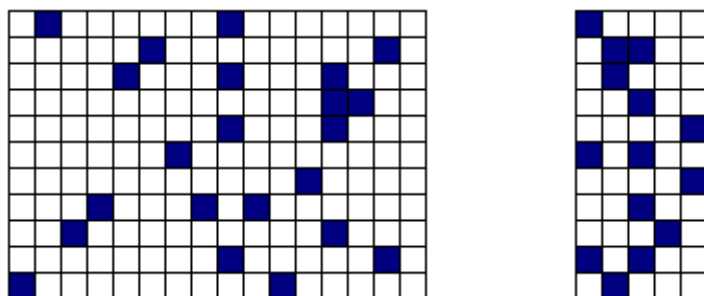
Показатель	Значение показателя	Скоринг-балл
Возраст	До 30 лет	0
	От 30 до 50 лет	35
	Старше 50 лет	28
Образование	Среднее	0
	Среднее специальное	29
	Высшее	35
Состоит ли в браке	Да	25
	Нет	0
Брал ли кредит ранее	Да	41
	Нет	0
Трудовой стаж	Менее 1 года	0
	От 1 до 5 лет	19
	От 5 до 10 лет	24
	Более 10 лет	31

<https://wiki.loginom.ru/articles/scorecard.html>

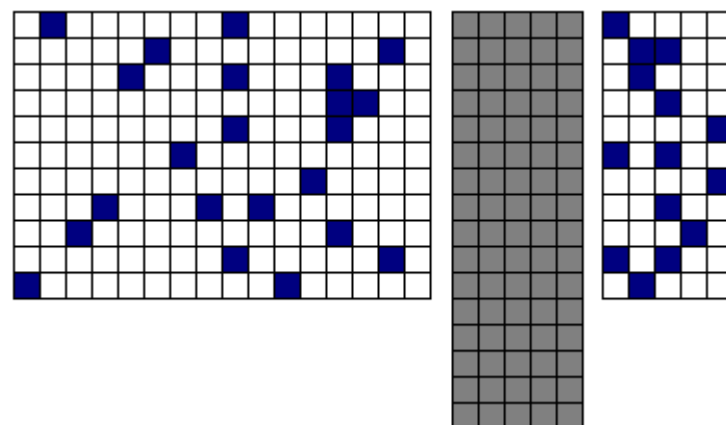
Задачи с текстами

Соревнование «Topical Classification of Biomedical Research Papers»

Данные



Логика решения

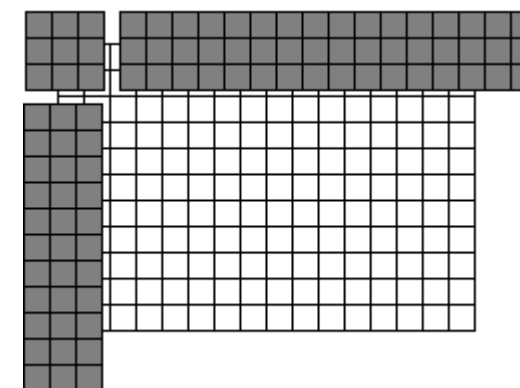


$$X_{q \times n} \cdot W_{n \times l} = Y_{q \times l}$$

$$q = 10000, n = 25000, l = 83$$

нельзя решать напрямую

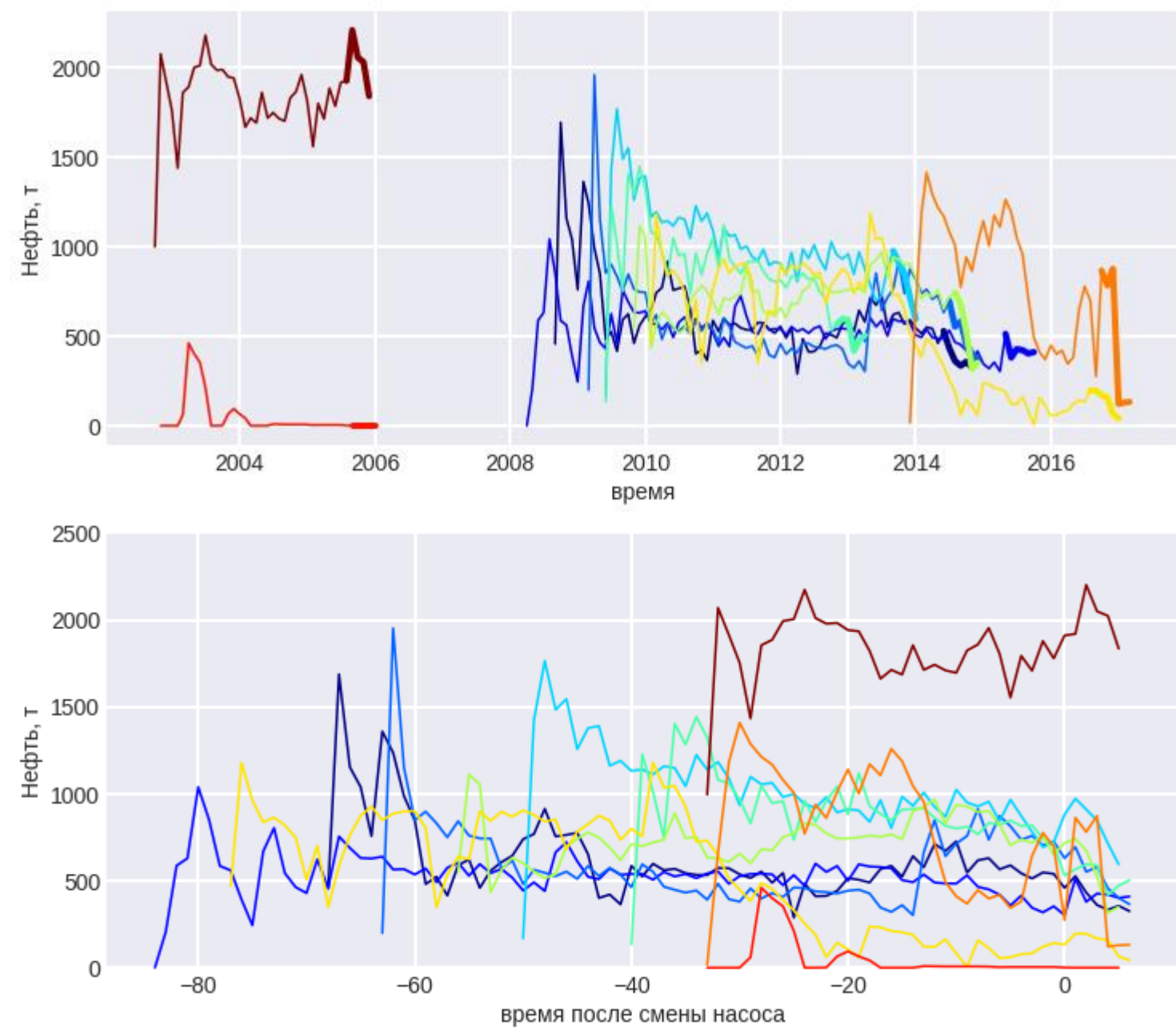
Упрощение: SVD



$$X_{q \times n} \approx U_{q \times k} L_{k \times k} V_{k \times n}$$

$$U_{q \times k} \cdot W_{k \times l} = Y_{q \times l}$$

Бенчмарк прогнозирования дебита нефти



Бенчмарк прогнозирования дебита нефти



$$y_t = \sum_{i=0}^k w_{ti} y_{-i}, \quad w_{t0} \geq w_{t1} \geq \dots$$

соревнование на платформе boosters.pro

<https://dyakonov.org/2018/12/23/>

Прогнозирование спроса

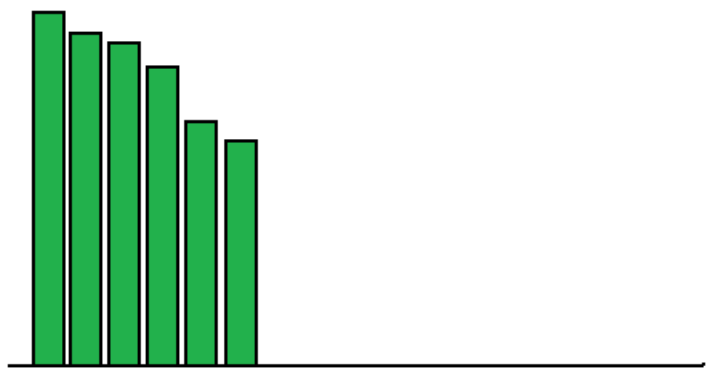
Спрос товара конкретного id (покупок за следующую неделю)

- **# покупок за k дней**
- **# просмотров за k дней**
 - **# корзин за k дней**
 - **# дней без покупок**
- **изменение цены за последние k дней**
- **есть ли маркетинговая акция**
- **...**

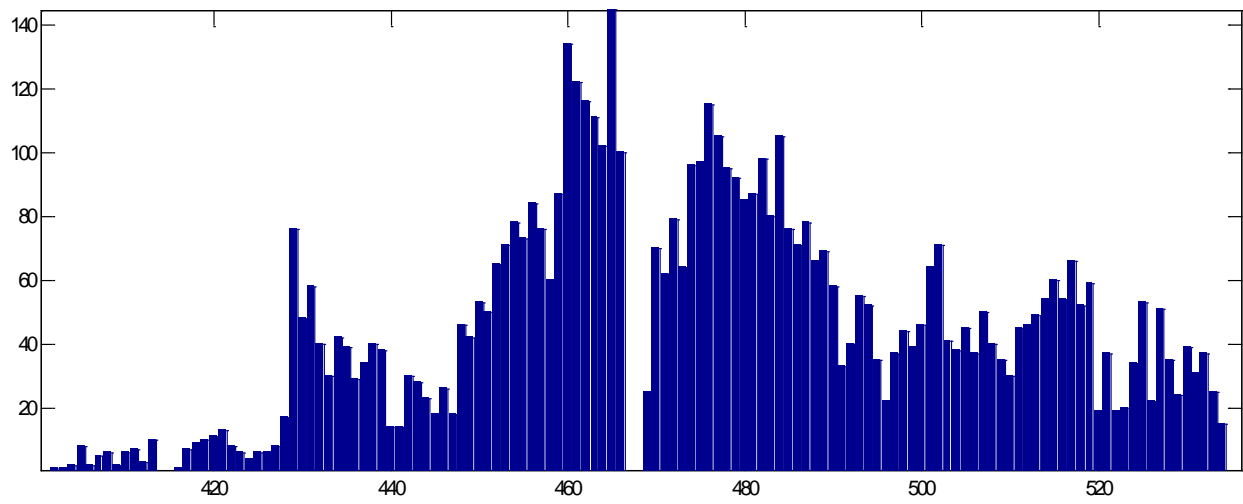
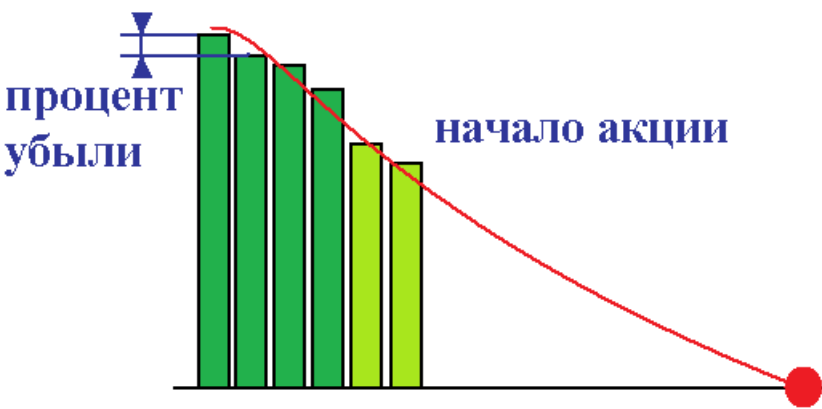
$$Y = \max \left[\sum_t w_t X_t, 0 \right]$$

Прогнозирование раскупаемости

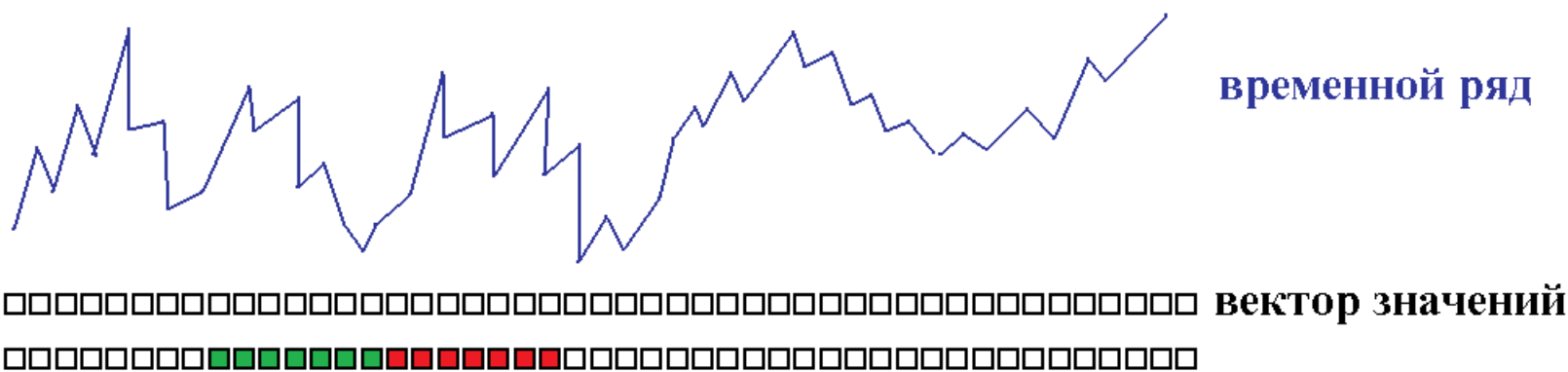
Остатки товара на складе:



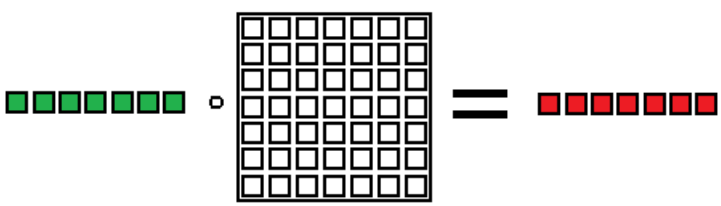
Прогноз точки раскупаемости



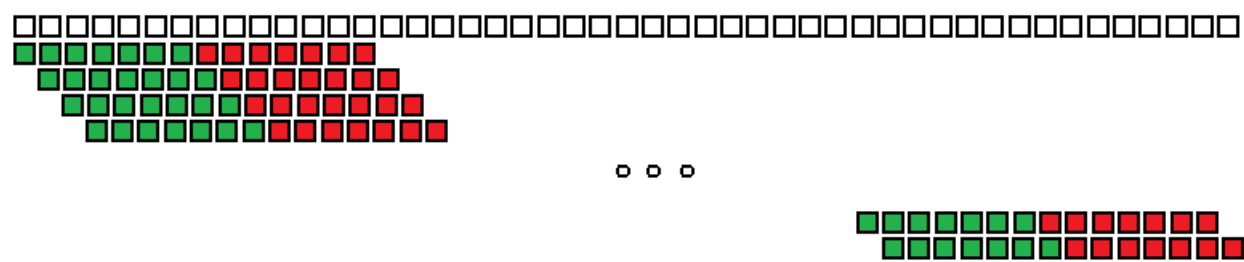
Линейный метод прогнозирования



Пусть существует линейный оператор

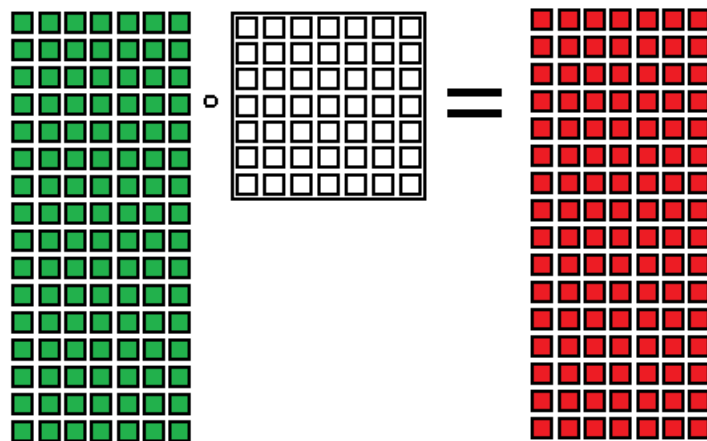


Обучающая выборка



т.е. данных много!

Линейный метод прогнозирования



Это матричное уравнение!

$$A \cdot X = b$$

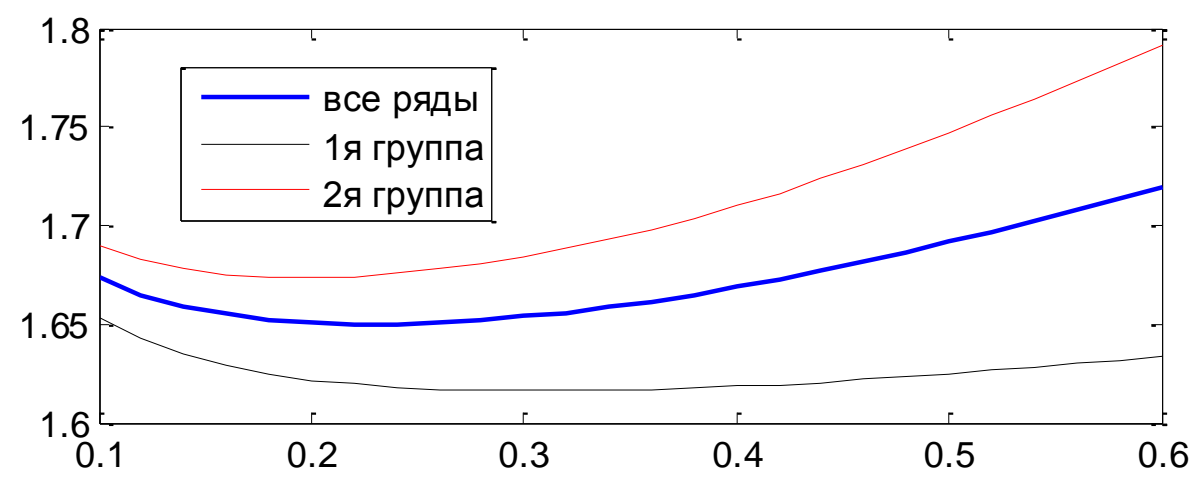
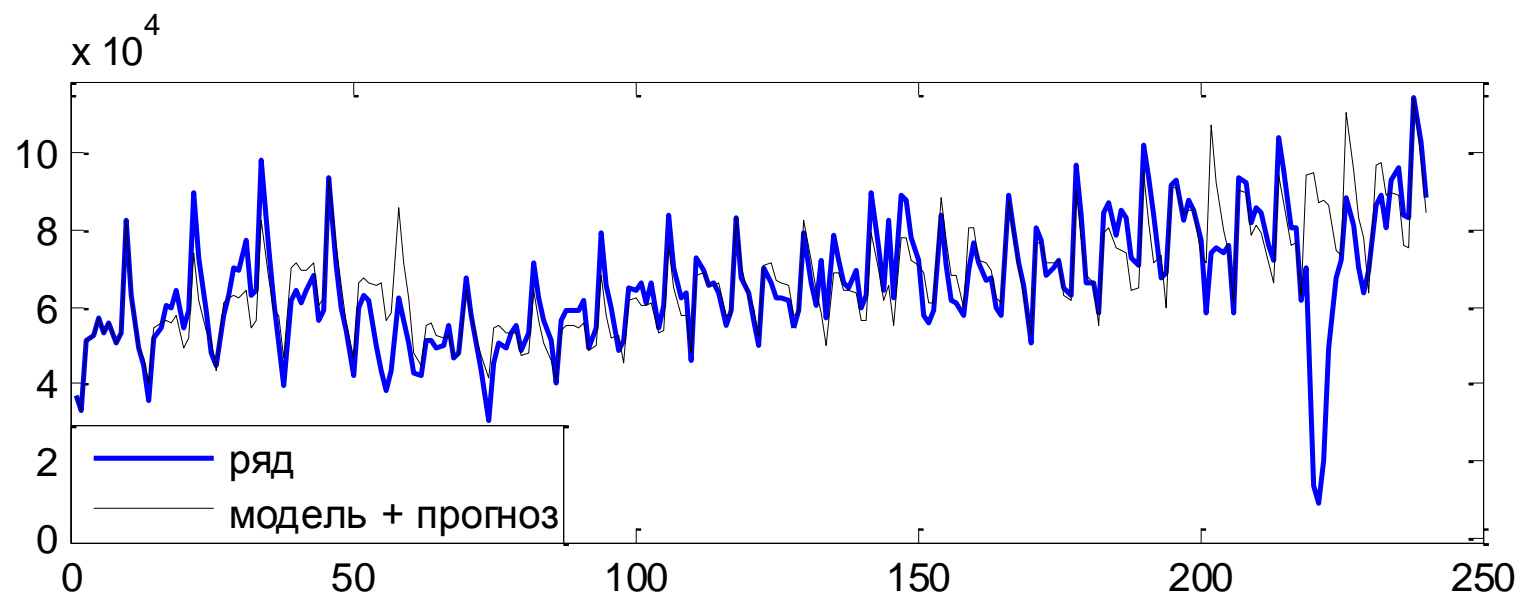
Можно также

применять нелинейные операторы (как?)

Накладывать ограничения стабильности: $A(A(\tilde{x}_t)) \approx A(\tilde{x}_{t+1})$

Делать регуляризацию (и тут её надо правильно сделать – нормировки)

Линейный метод прогнозирования



точность от коэф. регуляризации

Плюсы и минусы линейных алгоритмов

- + простой, надёжный, быстрый, популярный метод**
 - + интерпретируемость (\Rightarrow нахождение закономерностей)**
 - + интерполяция и экстраполяция**
 - + может быть добавлена нелинейность, с помощью генерации новых признаков**
(дальше – это можно автоматизировать)
 - + хороши для теоретических исследований (в Ridge есть явная формула)**
 - + коэффициенты асимптотически нормальны**
(можно тестировать гипотезы о влиянии признаков)
 - + глобальный минимум в оптимизируемом функционале**
 - линейная гипотеза вряд ли верна**
 - в теоретическом обосновании ещё предполагается нормальность ошибок**
(зависит от функции ошибок)
 - «страдает» из-за выбросов**
 - признаки в одной шкале и однородные (см. успешные примеры)**
 - проблема коррелированных признаков**
- \Rightarrow необходимость регуляризации, селекции, PCA, data \uparrow

Проблемы мультиколлинеарности

- **большие коэффициенты**
- **большие изменения коэффициентов при добавлении/удалении признаков**
- **нелогичности**
(чем больше доход, меньше вероятность дать кредит)
- **большое число статистически незначимых оценок коэффициентов**

Зависимость от масштабирования

простая модель	с регуляризацией	пайплайн: нормировка + линейная
нет	есть	нет

Итог

Линейная регрессия ~ матричное уравнение

Но проблема вырожденности

Много методов решают эту проблему с разных сторон

Логистическая регрессия

– деформирование линейной

Есть вероятностная трактовка!

Интересные ссылки

Песня о RANSAC

<https://www.youtube.com/watch?v=1YNjMxxXO-E>

Курс Ramesh Sridharan «Statistics for Research Projects: IAP 2015»

<http://www.mit.edu/~6.s085/>

Про расстояние Кука

https://en.wikipedia.org/wiki/Cook%27s_distance

Про линейную регрессию

<https://dyakonov.org/2019/10/31/линейная-регрессия/>