

БАЗОВЫЕ ПРИНЦИПЫ МАШИННОГО ОБУЧЕНИЯ

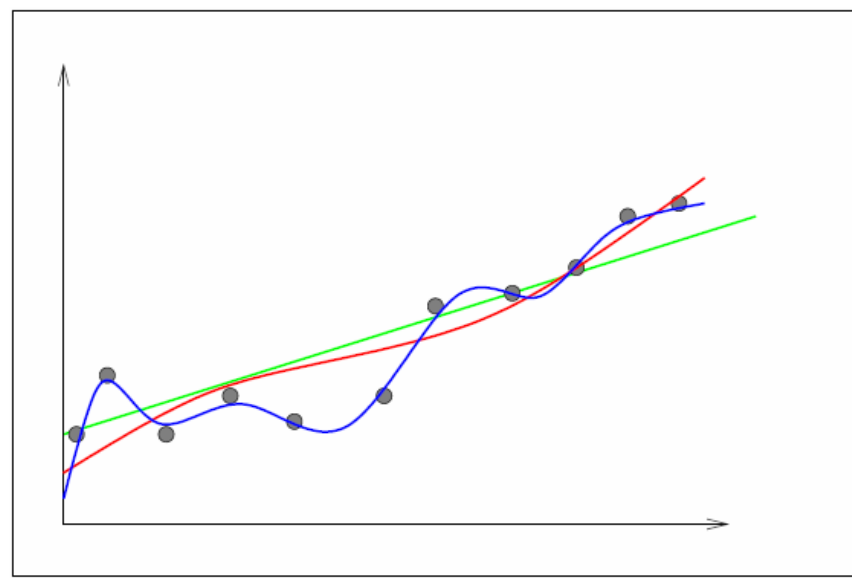
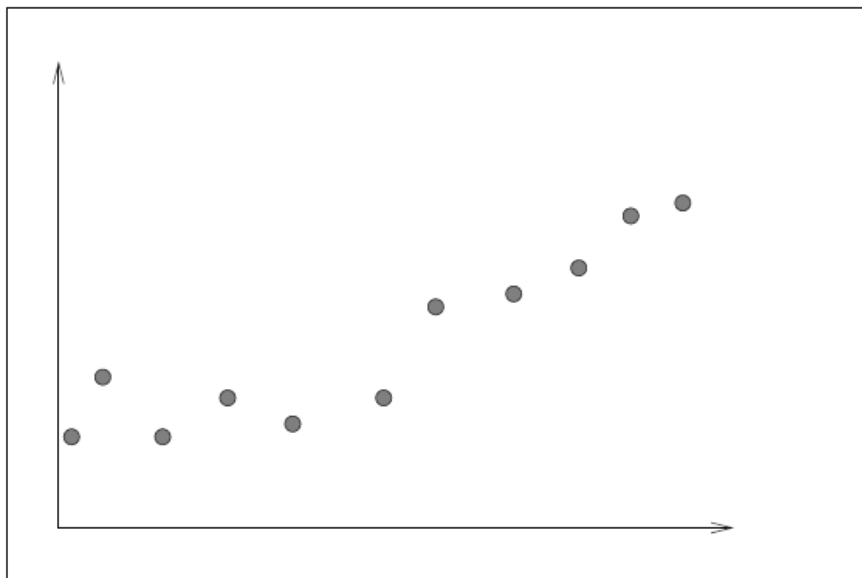
на примере линейной
регрессии

Регрессия

Дана обучающая выборка

$$X_N = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}, \quad (\mathbf{x}_i, y_i) \in \mathbf{R}^P \times \mathbf{R}$$

Цель: для всех новых значений \mathbf{x} оценить значения y



Метод наименьших квадратов

- Линейная модель: рассмотрим линейную функцию

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^p x_j w_j = \mathbf{x}^\top \mathbf{w}, \quad \mathbf{x} = (1, x_1, \dots, x_p).$$

- Таким образом, по вектору входов $\mathbf{x}^\top = (x_1, \dots, x_p)$ мы будем предсказывать выход y как

$$\hat{y}(\mathbf{x}) = \hat{w}_0 + \sum_{j=1}^p x_j \hat{w}_j = \mathbf{x}^\top \hat{\mathbf{w}}.$$

Метод наименьших квадратов

- Как найти оптимальные параметры $\hat{\mathbf{w}}$ по тренировочным данным вида $(\mathbf{x}_i, y_i)_{i=1}^N$?
- Метод наименьших квадратов: будем минимизировать

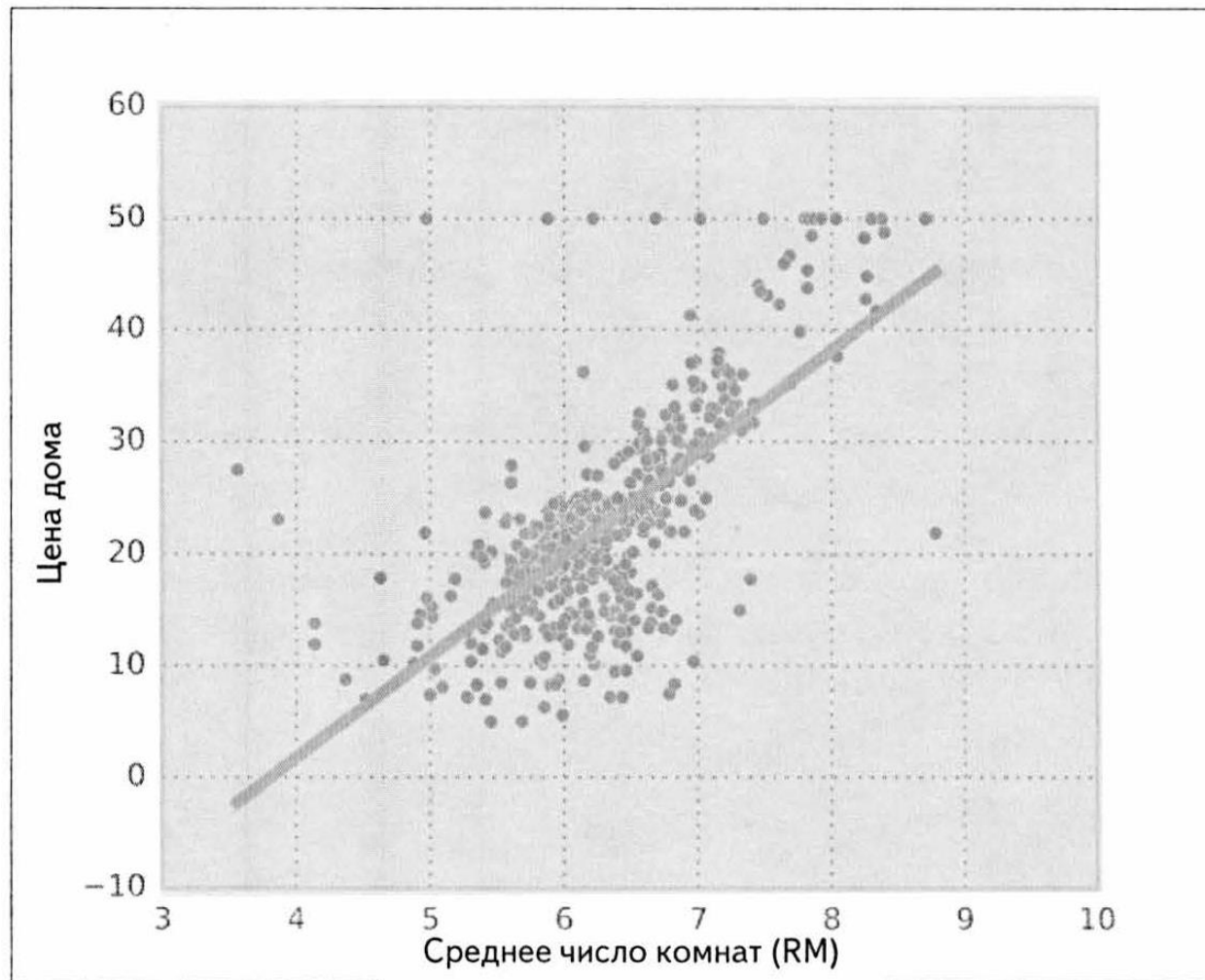
$$\text{RSS}(\mathbf{w}) = \sum_{i=1}^N (y_i - \mathbf{x}_i^\top \mathbf{w})^2.$$

$$\text{RSS}(\mathbf{w}) = (\mathbf{y} - \mathbf{X}\mathbf{w})^\top (\mathbf{y} - \mathbf{X}\mathbf{w}),$$

$$\hat{\mathbf{w}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y},$$

если матрица $\mathbf{X}^\top \mathbf{X}$ невырожденная.

Пример: прогнозирование стоимости домов



RMSE=6.6
 $R^2=0.31$

Измерение ошибки в задачах регрессии

$$L(y, \hat{y}) = (y - \hat{y})^2$$

$$L(y, \hat{y}) = |y - \hat{y}|$$

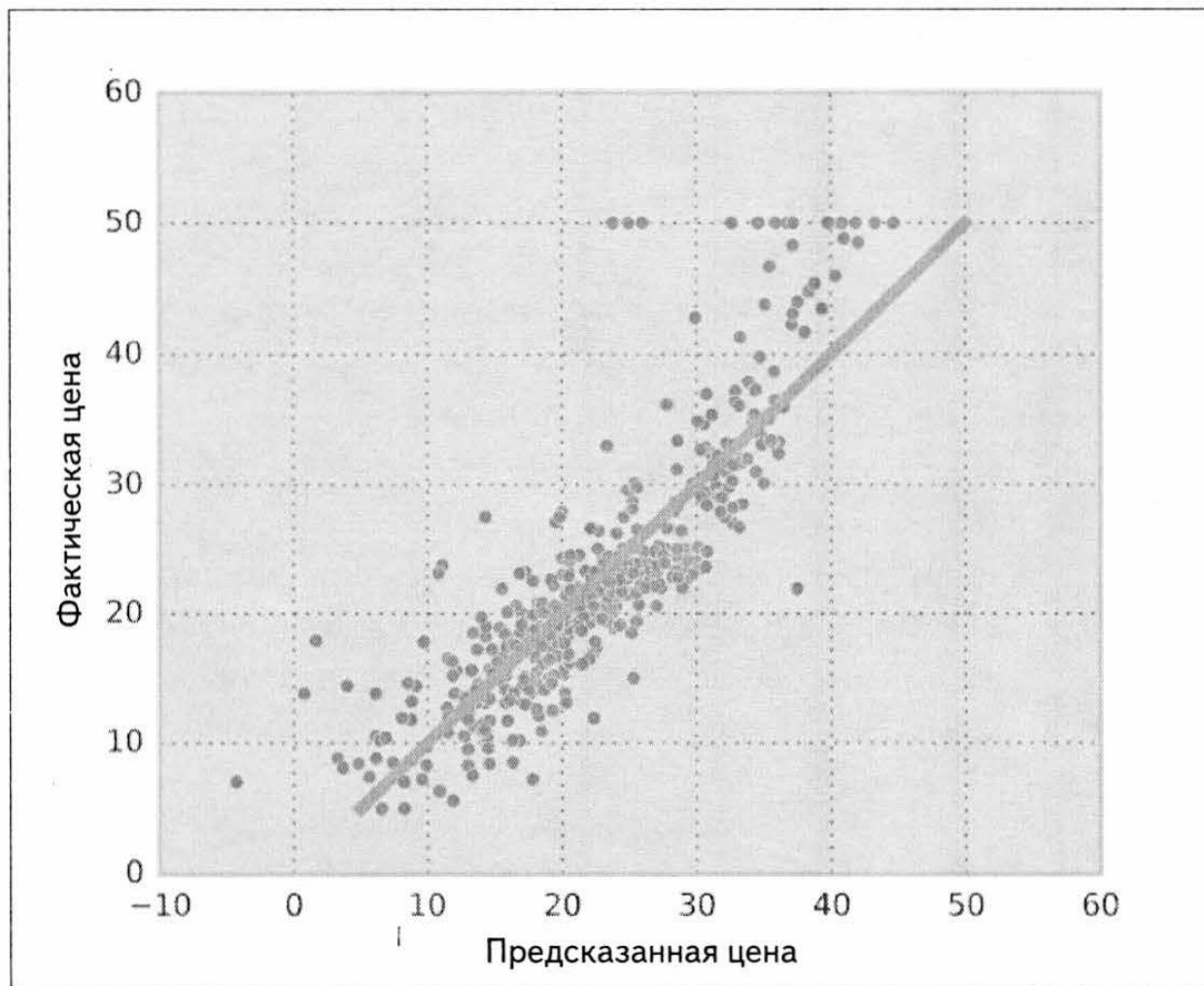
$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

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

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}$$

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2}$$

Многомерная регрессия



RMSE=4.7
 $R^2=0.74$

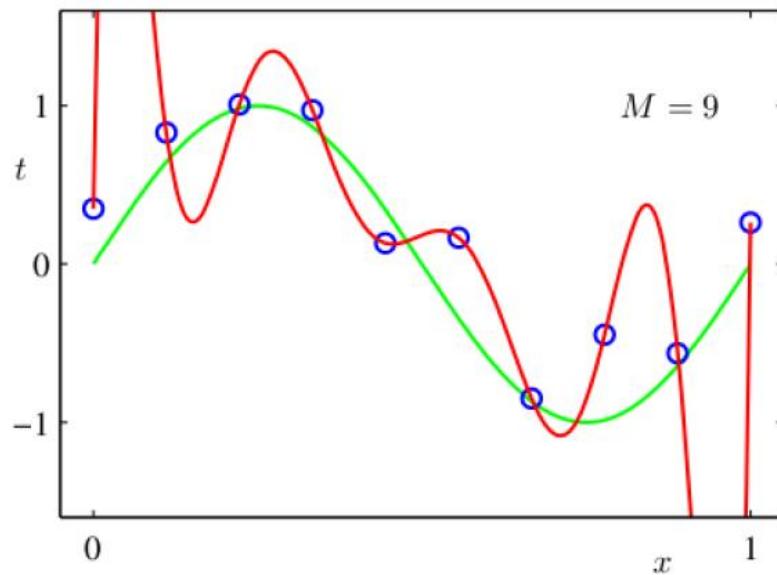
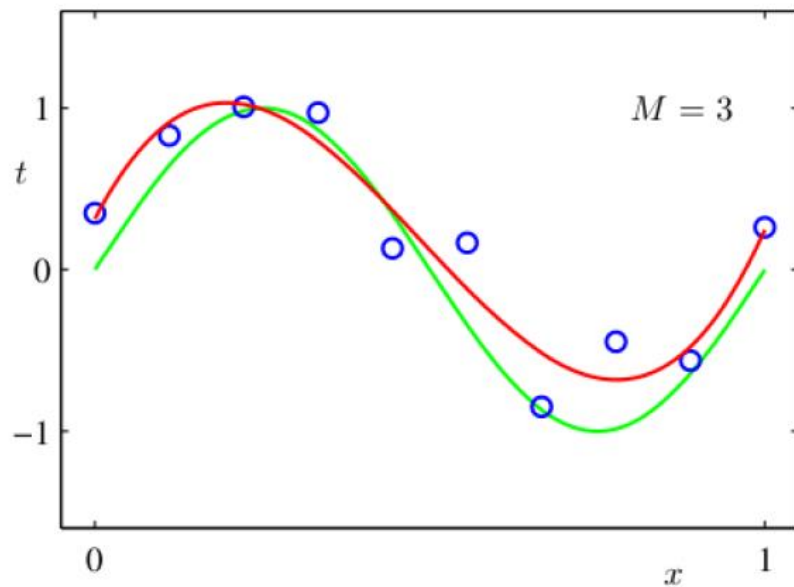
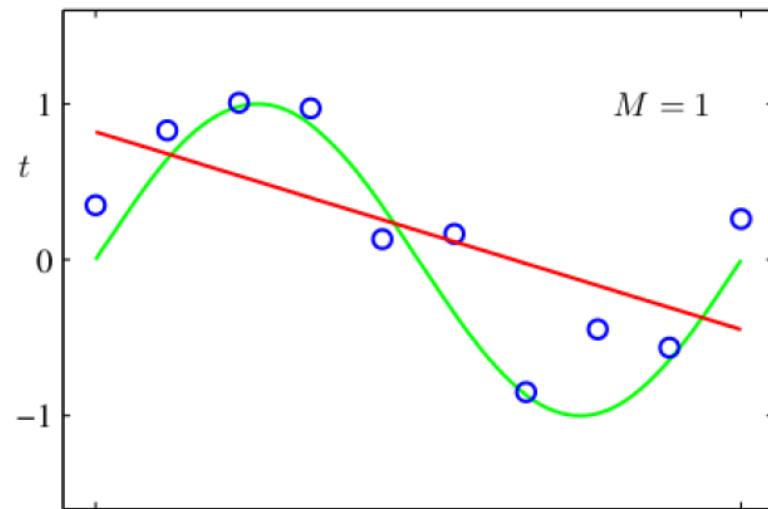
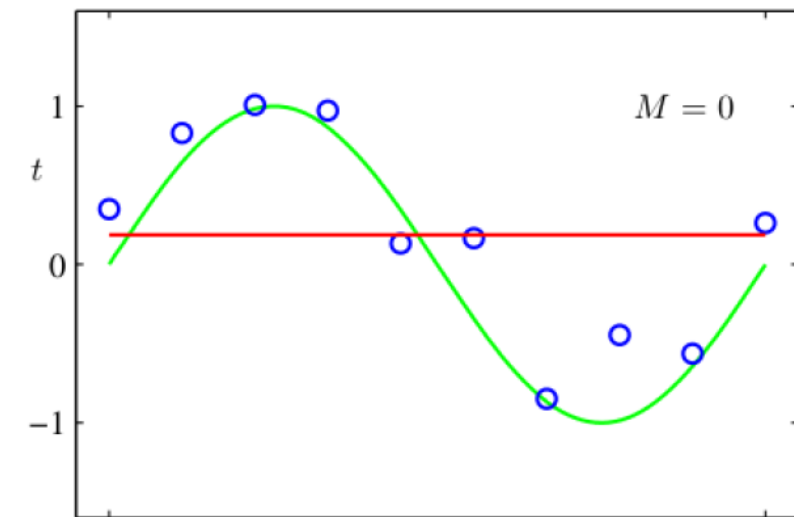
Регрессия, линейная по параметрам

$$f(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^M w_j \phi_j(\mathbf{x}) = \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}).$$

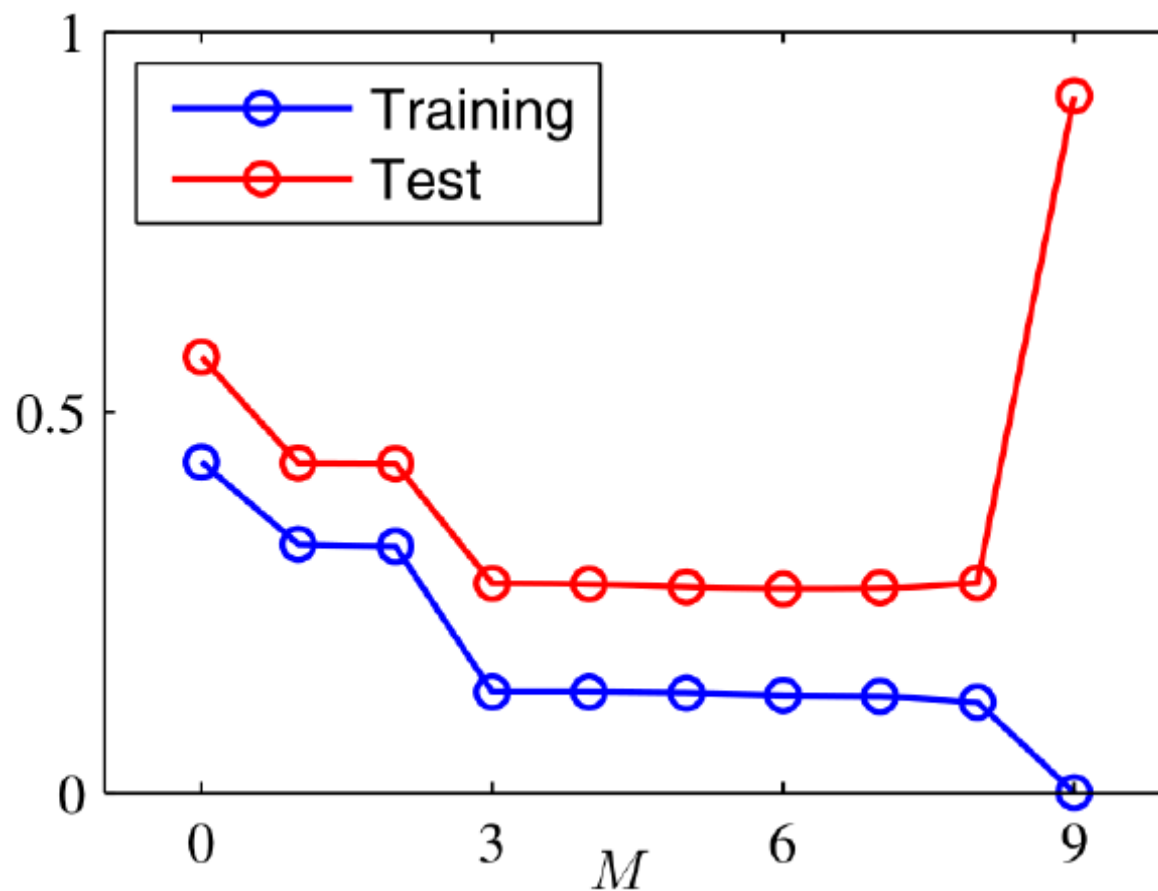
Например:

$$f(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \dots + w_M x^M.$$

Полиномиальная регрессия



Значения $RMSE$



Значения коэффициентов

	$M = 0$	$M = 1$	$M = 6$	$M = 9$
w_0^*	0.19	0.82	0.31	0.35
w_1^*		-1.27	7.99	232.37
w_2^*			-25.43	-5321.83
w_3^*			17.37	48568.31
w_4^*				-231639.30
w_5^*				640042.26
w_6^*				-1061800.52
w_7^*				1042400.18
w_8^*				-557682.99
w_9^*				125201.43

L2- Регуляризация (гребневая регрессия)

- Было (для тестовых примеров $\{(x_i, y_i)\}_{i=1}^N$):

$$\text{RSS}(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^N (f(x_i, \mathbf{w}) - y_i)^2$$

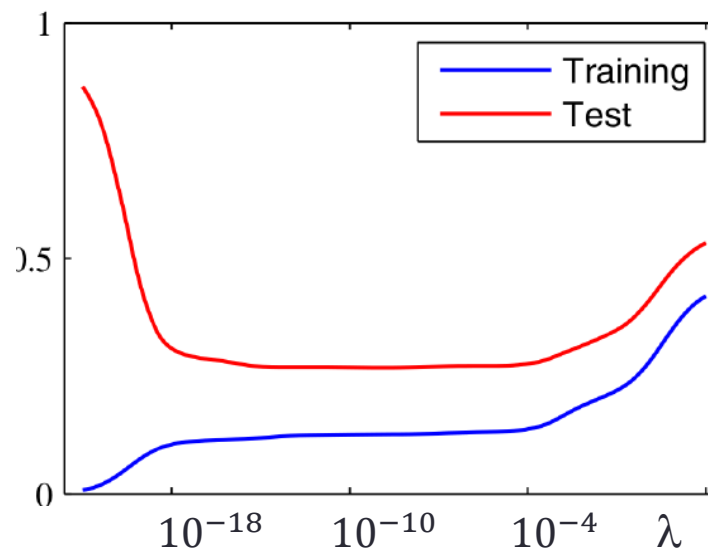
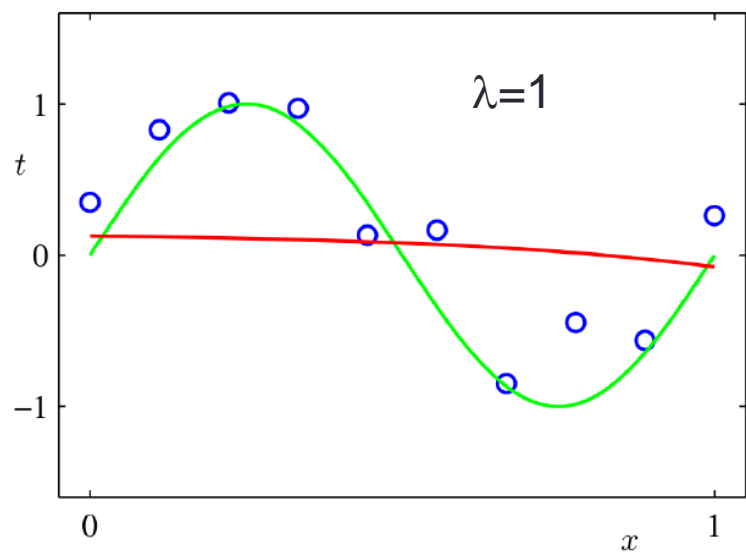
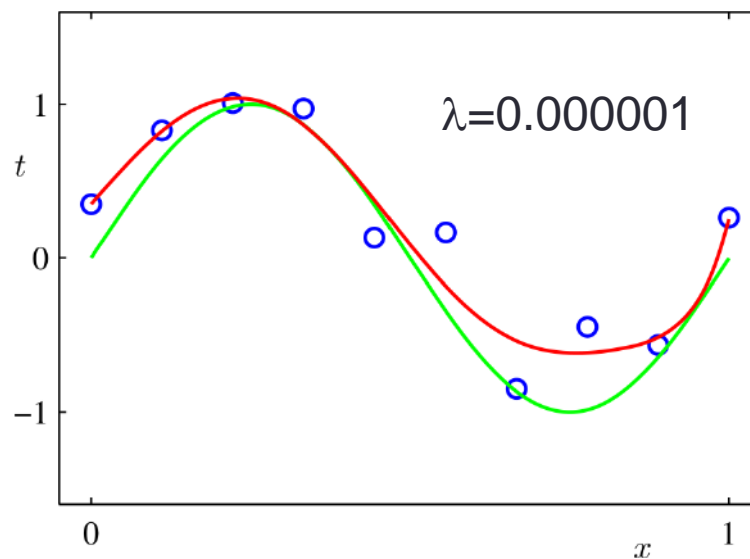
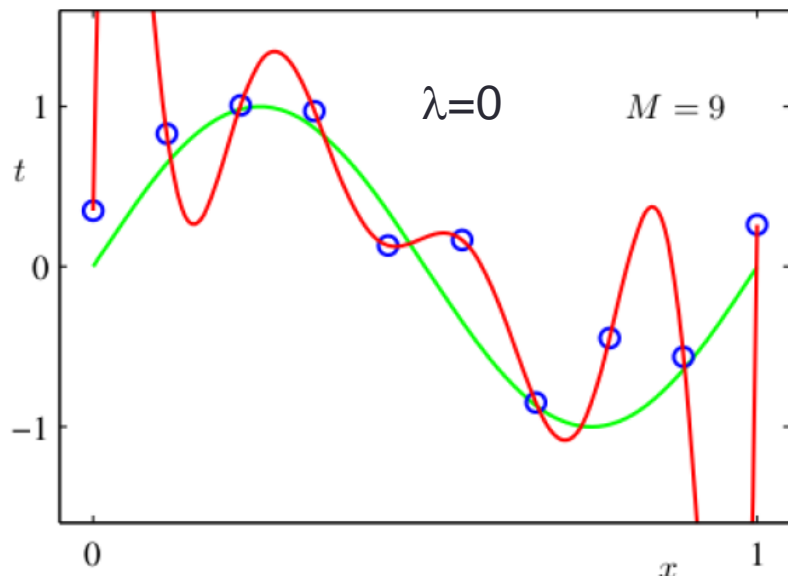
- Стало:

$$\text{RSS}(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^N (f(x_i, \mathbf{w}) - y_i)^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2,$$

где λ – коэффициент регуляризации

В регрессии, линейной по факторам: $\mathbf{w}^* = \left(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I} \right)^{-1} \mathbf{X}^\top \mathbf{y}.$

L2-регуляризация



Коэффициенты гребневой регрессии

	$\lambda=0$	$\lambda= 0.0000001$	$\lambda=1$
w_0^*	0.35	0.35	0.13
w_1^*	232.37	4.74	-0.05
w_2^*	-5321.83	-0.77	-0.06
w_3^*	48568.31	-31.97	-0.05
w_4^*	-231639.30	-3.89	-0.03
w_5^*	640042.26	55.28	-0.02
w_6^*	-1061800.52	41.32	-0.01
w_7^*	1042400.18	-45.95	-0.00
w_8^*	-557682.99	-91.53	0.00
w_9^*	125201.43	72.68	0.01

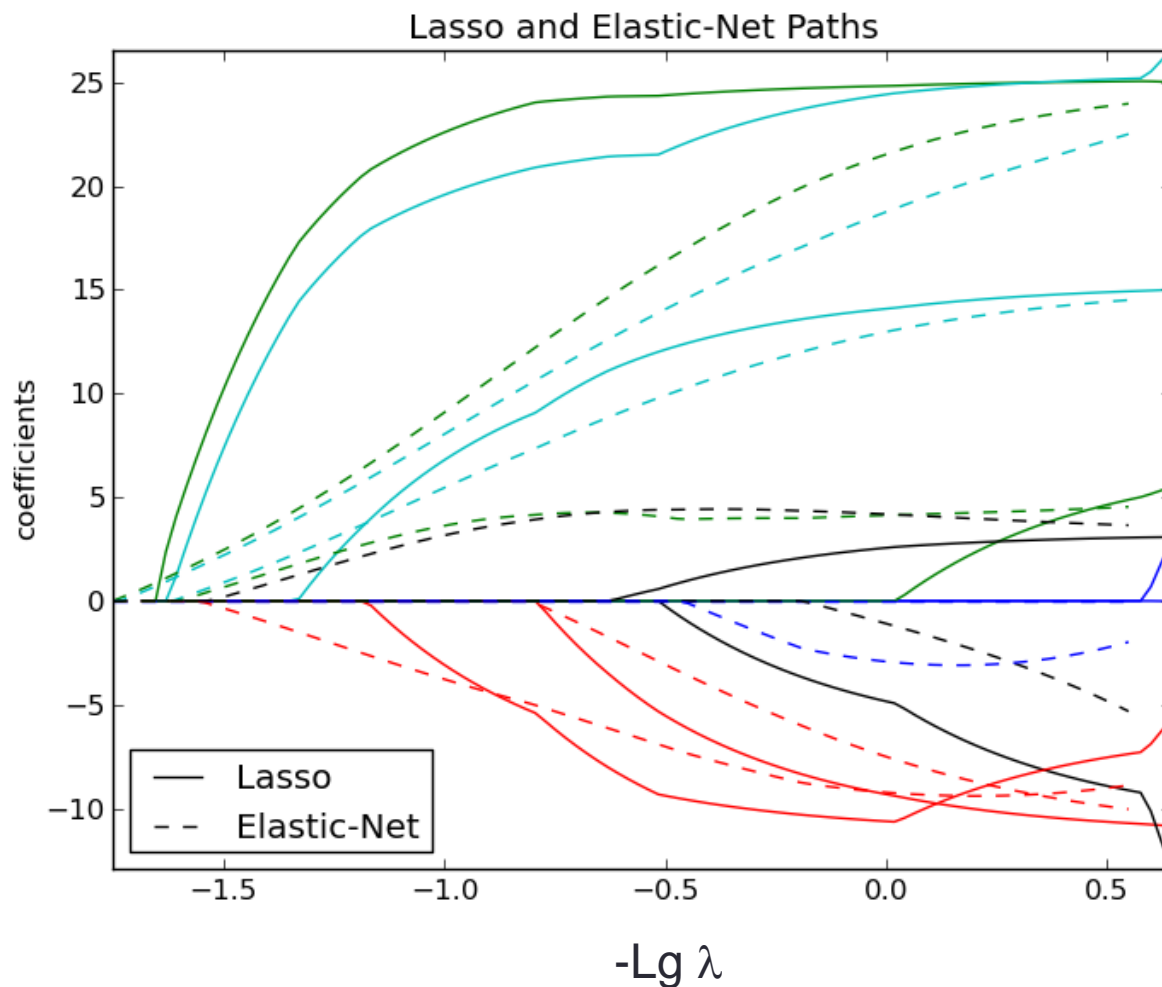
L1-регуляризация (Lasso)

$$\text{RSS}(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^N (f(x_i, \mathbf{w}) - y_i)^2 + \lambda \sum_{j=0}^M |w_j|.$$

Эластичная сеть:

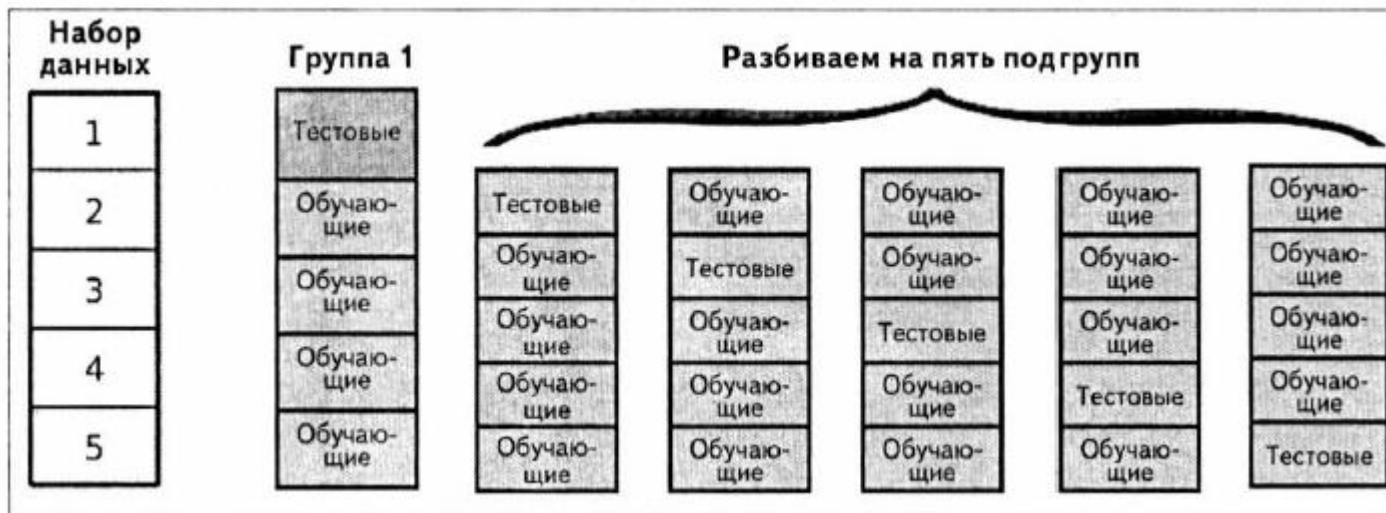
$$\text{RSS}(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^N (f(x_i, \mathbf{w}) - y_i)^2 + \lambda_1 \sum_{j=0}^M |w_j| + \frac{\lambda_2}{2} \|\mathbf{w}\|^2$$

Пример использования Lasso и ElasticNet



Настройка гиперпараметров

- 1) три выборки – обучающая (настраиваются параметры); валидационная (настраиваются гиперпараметры) и тестовая (анализируются результаты обучения)
- 2) кросс-валидация (перекрестная проверка):



ЛИНЕЙНАЯ КЛАССИФИКАЦИЯ

Логистическая регрессия

Бинарный линейный классификатор

Дана обучающая выборка

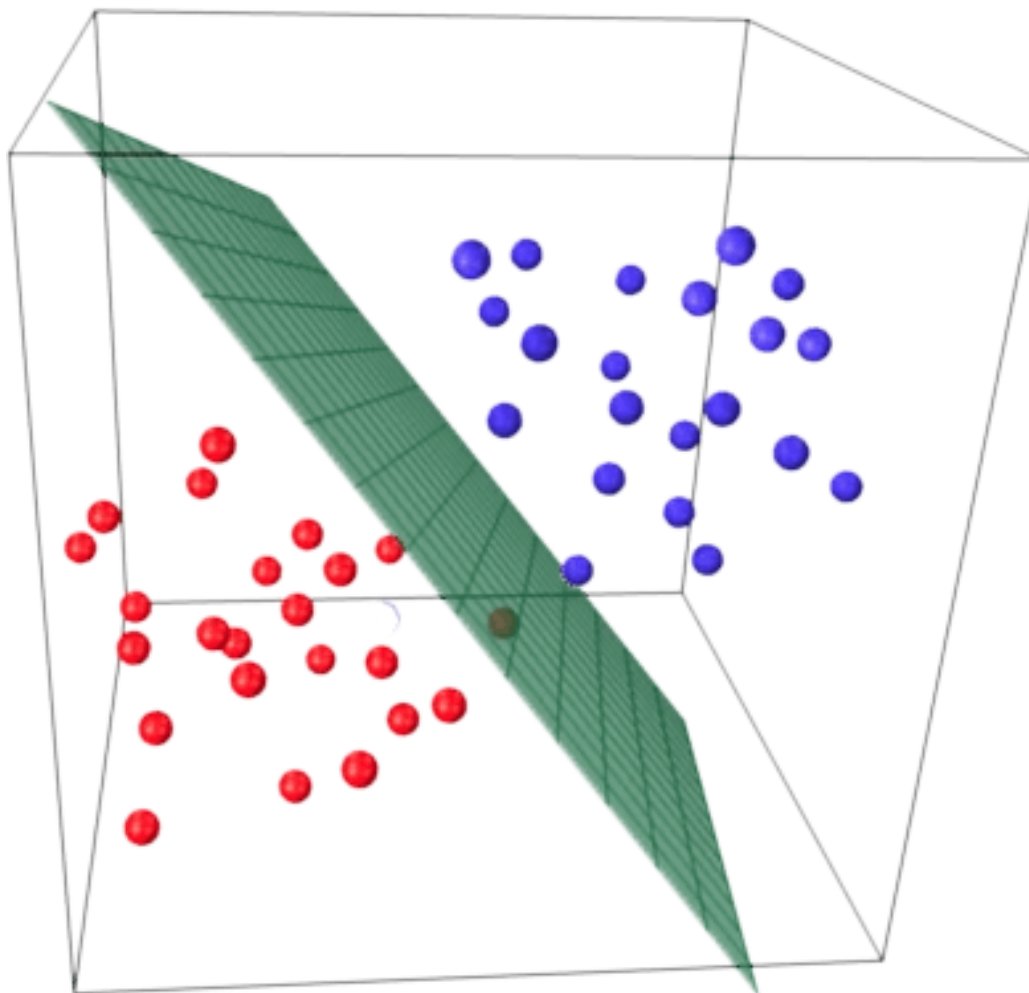
$$X_N = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}, \quad \mathbf{x}_i \in \mathbf{R}^p, y_i \in \{-1, +1\}$$

Цель: каждый новый входной вектор \mathbf{x} отнести к одному из двух классов – положительному «+1» или отрицательному «-1»

$$\hat{y} = \hat{y}(\mathbf{x}, \mathbf{w}) = \text{sign} \left(w_0 + \sum_{j=1}^p w_j x_j \right) = \text{sign}(\mathbf{w}^\top \mathbf{x}),$$

$$\mathbf{x} = (1, x_1, \dots, x_p)$$

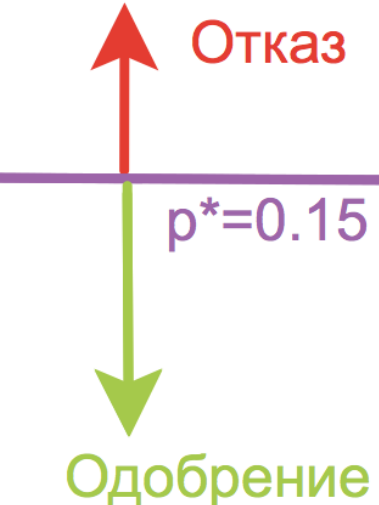
Линейная модель классификации



Логистическая регрессия как линейный классификатор

- Логистическая регрессия прогнозирует вероятность p_+ отнесения примера x к классу "+1".
- Пример: банковский скоринг

Клиент	Вероятность невозврата
Mike	0.78
Jack	0.45
Larry	0.13
Kate	0.06
William	0.03
Jessica	0.02

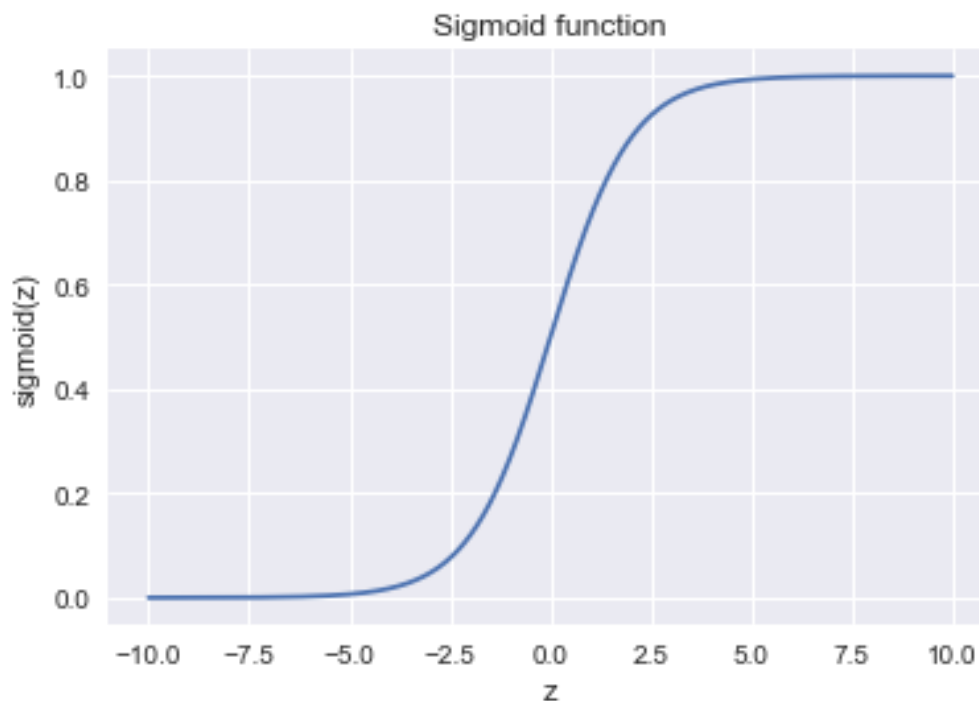


The diagram illustrates the classification process. A horizontal purple line represents the decision threshold at $p^* = 0.15$. A red arrow points upwards from the threshold, labeled "Отказ" (Refusal), indicating that clients with a probability of default greater than 0.15 are refused. A green arrow points downwards from the threshold, labeled "Одобрение" (Approval), indicating that clients with a probability of default less than or equal to 0.15 are approved. In the table, Mike and Jack are above the threshold, while Larry, Kate, William, and Jessica are below it.

Логистическая регрессия

$$p_+ = \sigma(z) = \frac{1}{1 + \exp(-z)}$$

где $z = w^\top \mathbf{x} = w_0 + \sum_{j=1}^p w_j x_j$

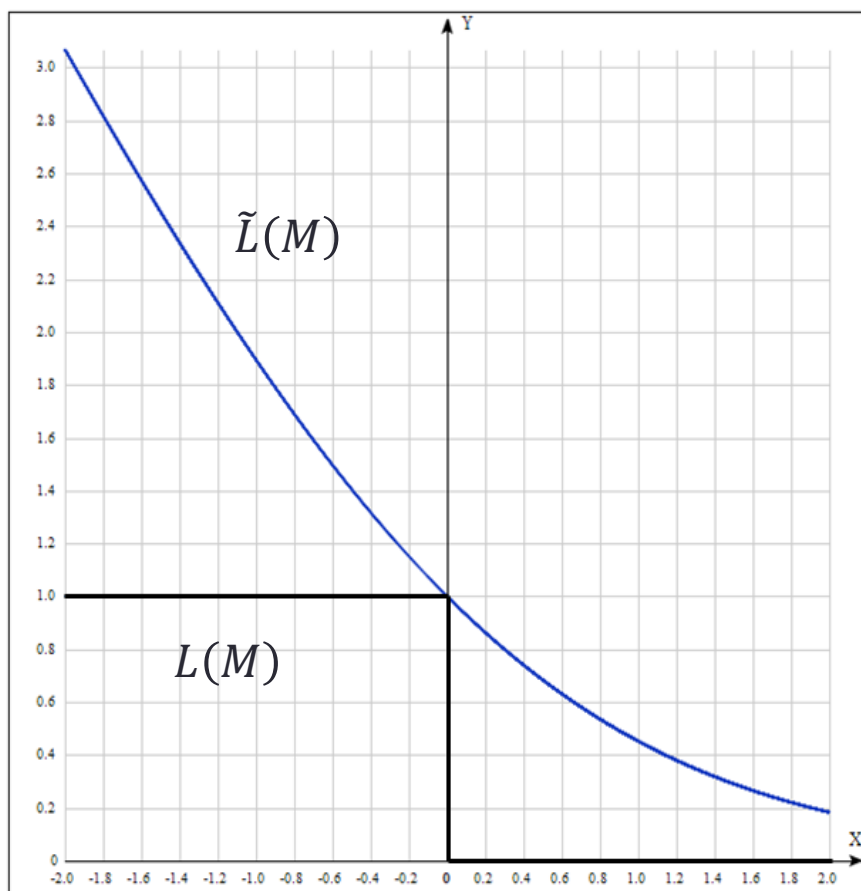


Функция потерь (ошибок классификации)

- Доля неправильных ответов:
- $E(W) = \frac{1}{N} \sum_{i=1}^N [y_i \neq \hat{y}_i] = \frac{1}{N} \sum_{i=1}^N [\text{sign}(w^\top x_i) \neq y_i]$
- $E(W) = \frac{1}{N} \sum_{i=1}^N [y_i (w^\top x_i) < 0]$
- $M_i = y_i (w^\top x_i)$ - отступ
- $L(M) = [M < 0]$ – пороговая функция

Верхняя оценка

$$L(M) \leq \tilde{L}(M) = \log_2(1 + \exp(-M))$$



Логистическая функция потерь

- $ERR(w) = \sum_{i=1}^N \log_2 (1 + \exp(-y_i(w^\top x_i)))$

С учетом L2-регуляризации:

- $ERR(w) = \sum_{i=1}^N \log_2 (1 + \exp(-y_i(w^\top x_i))) + \frac{1}{\lambda} \sum_{j=1}^p w_j^2$

Использование полиномиальных признаков для нелинейного разделения

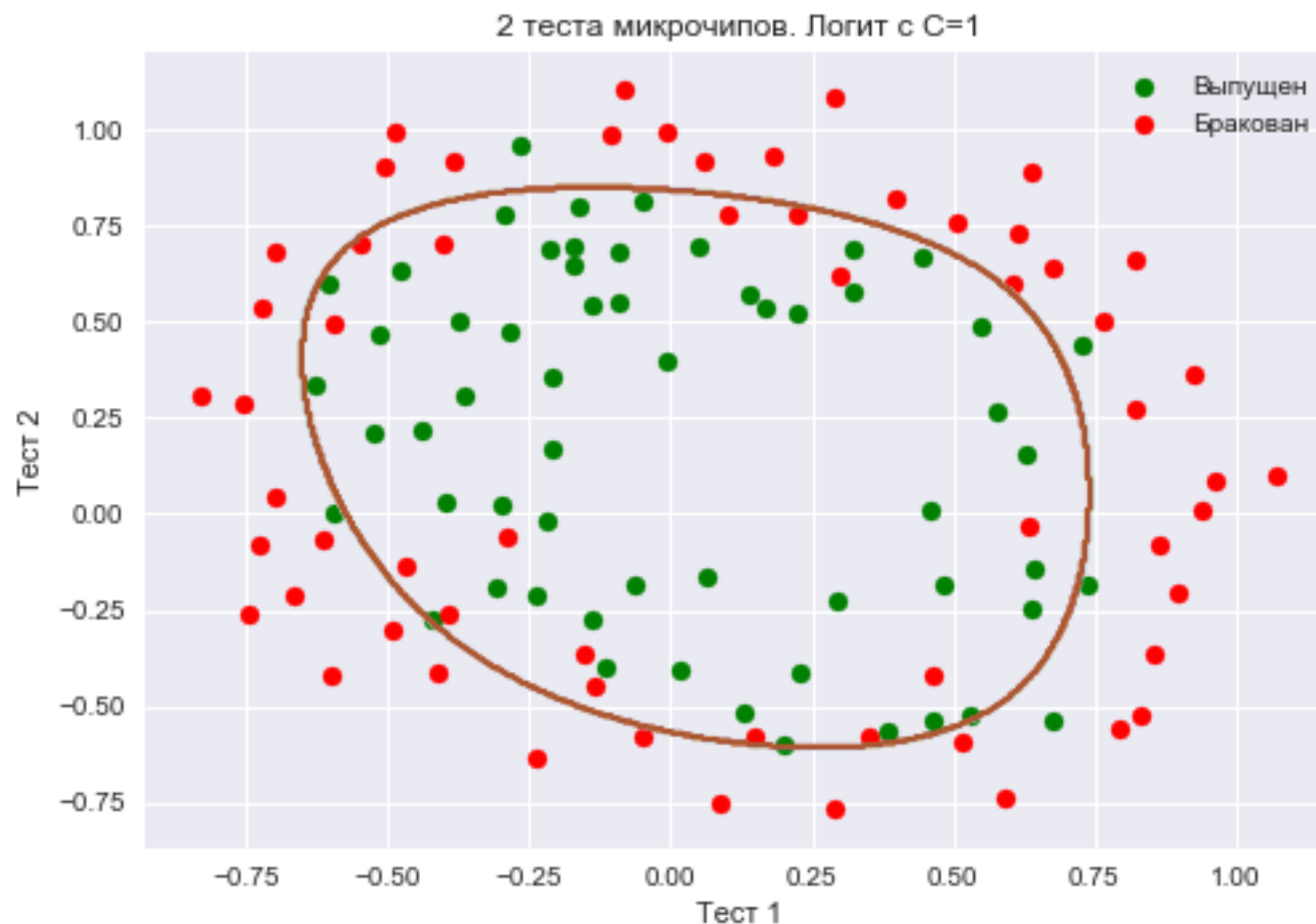
Полиномиальными признаками до степени d для двух переменных x_1 и x_2 мы называем следующие:

$$\{x_1^d, x_1^{d-1}x_2, \dots, x_2^d\} = \{x_1^i x_2^j\}_{i+j=d, i,j \in \mathbb{N}}$$

Например, для $d = 3$ это будут следующие признаки:

$$1, x_1, x_2, x_1^2, x_1 x_2, x_2^2, x_1^3, x_1^2 x_2, x_1 x_2^2, x_2^3$$

Пример нелинейного разделения классов



Confusion matrix (матрица ошибок классификации)

	$y = 1$	$y = 0$
$\hat{y} = 1$	True Positive (TP)	False Positive (FP)
$\hat{y} = 0$	False Negative (FN)	True Negative (TN)

Здесь \hat{y} — это ответ алгоритма на объекте, а y — истинная метка класса на этом объекте.

Таким образом, ошибки классификации бывают двух видов: False Negative (FN) и False Positive (FP).

Метрики качества классификации

- Доля правильных ответов: $accuracy = \frac{TP + TN}{TP + TN + FP + FN}$

Малоинформативна в задачах с неравными классами.

Пример. Допустим, мы хотим оценить работу спам-фильтра почты. У нас есть 100 не-спам писем, 90 из которых наш классификатор определил верно, и 10 спам-писем, 5 из которых классификатор также определил верно. Тогда accuracy:

$$accuracy = \frac{5 + 90}{5 + 90 + 10 + 5} = 0.864$$

Если мы просто будем предсказывать все письма как не-спам, то получим более высокую accuracy

$$accuracy = \frac{0 + 100}{0 + 100 + 0 + 10} = 0.909$$

Метрики качества классификации

- precision (точность) и recall (полнота).

$$precision = \frac{TP}{TP + FP}$$

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

Precision показывает долю объектов, названных классификатором положительными и при этом действительно являющимися положительными, а recall показывает, какую долю объектов положительного класса из всех объектов положительного класса нашел алгоритм.

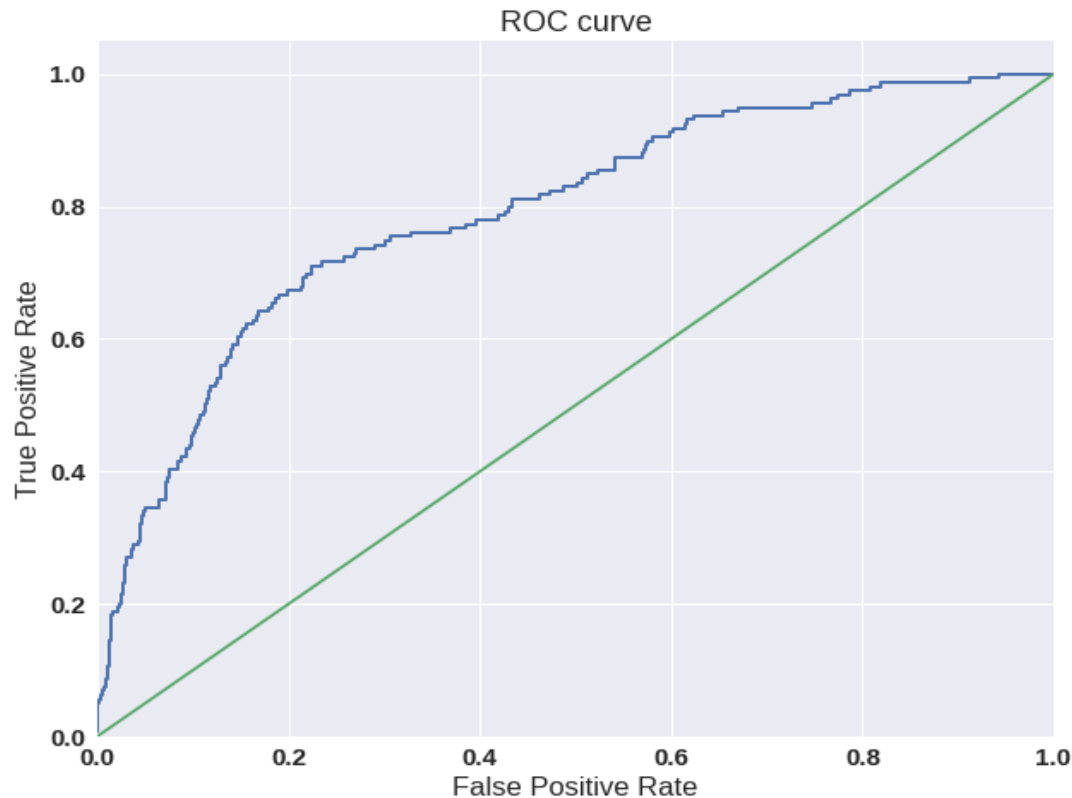
Precision не позволяет записывать все объекты в один класс, так как в этом случае растет значение FP. Recall демонстрирует способность алгоритма обнаруживать данный класс вообще, а precision — способность отличать этот класс от других классов.

AUC-ROC – площадь под кривой ошибок

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

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

TPR - это полнота, а FPR показывает, какую долю из объектов отрицательного класса алгоритм предсказал неверно.



Кривая ошибок или **ROC-кривая** – графическая характеристика качества бинарного классификатора, зависимость доли верных положительных классификаций от доли ложных положительных классификаций при варьировании порога решающего правила.

Спасибо за внимание!

МОЁ ХОББИ: ЭКСТРАПОЛИРОВАТЬ

