ТЕХНОСФЕРА

Лекция №4

Введение в машинное обучение

Спасёнов Алексей

Содержание лекции

1. Задача регрессии
   1. Линейная регрессия
2. Задача классификации
   1. Решающее дерево
   2. K-ближайших соседей
   3. Логистическая регрессия
   4. SVM, Машина опорных векторов
3. Метрики
   1. Классификация
   2. Регрессия
4. Кросс-валидация

Обучение с учителем

Задачи классификации (classification)

- $F_j = \{true, false\}$ – классификация на 2 класса
- $F_j = \{1, \dots, M\}$ – классификация на M непересекающихся классов
- $F_j = \{0,1\}^M$ - классификация на M классов, которые могут пересекаться

Задача восстановления регрессии (regression)

- $F_j = \mathbb{R}$ или $F_j = \mathbb{R}^M$ (ответом является действительное число или числовой вектор)

Задача ранжирования (learning to rank)

- F_j - конечно упорядочено (ответы надо получить сразу на множестве объектов, после чего отсортировать их по значениям ответов)

Обучение с учителем

Этап обучения (train)

Данные $\{X_{1:N}, Y_{1:N}\}$ → **Обучение (Learner)** → Параметры <u>модели</u> Θ

Необходимо учитывать представительность выборки

Этап применения (test)

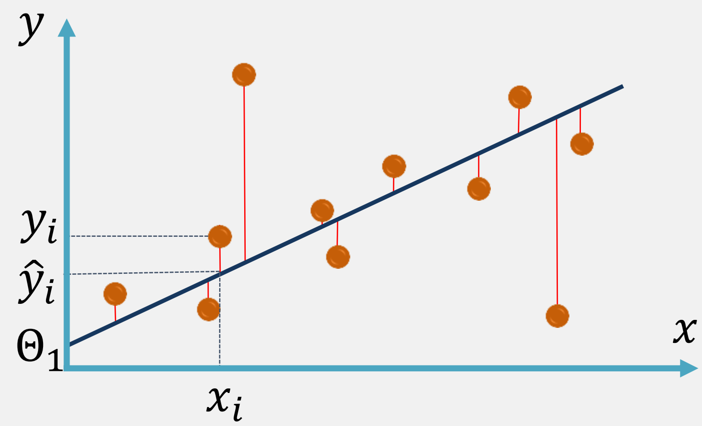
X_{N+1}, Θ → **Предсказание (Predictor)** → $\hat{Y}_{N+1}$

Задача восстановления регрессии

Линейная регрессия

Модель:

$$\hat{y}(X_i) = \Theta_1 + x_i\Theta_2$$



Целевая функция (Objective function, Energy, Loss)

Величина ошибки алгоритма на обучающей выборке

Пример для задачи регрессии:

$$J(\Theta) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n}(y_i - \Theta_1 - x_i\Theta_2)^2$$

Метод наименьших квадратов (Ordinary Least Squares)

Задача восстановления регрессии

Линейная регрессия

Пример задачи регрессии:

Предсказание стоимости жилья

Целевая переменная

Area	Bedroom	Kitchen	HouseStyle	Neighborhood	YearBuilt	Alley	SalePrice
8450	3	1	2Story	CollgCr	2003	NA	208500
9600	3	1	1Story	Veenker	1976	NA	181500
11250	3	1	2Story	CollgCr	2001	NA	223500
9550	3	1	2Story	Crawfor	1915	NA	140000
14260	4	1	2Story	NoRidge	2000	NA	250000
14115	1	1	1.5Fin	Mitchel	1993	Grvl	143000
10084	3	1	1Story	Somerst	2004	NA	307000
10382	3	1	2Story	NWAmes	1973	NA	200000
6120	2	2	1.5Fin	OldTown	1931	NA	129900
7420	2	2	1.5Unf	BrkSide	1939	Grvl	118000

Признаки

https://www.kaggle.com/c/house-prices-advanced-regression-techniques

Задача восстановления регрессии

Линейная регрессия

<u>Модель</u>:

$$\hat{y}_i = \sum_{j=1}^{d} x_{ij}\Theta_j = 1 * \Theta_1 + x_{i2}\Theta_2 + x_{i3}\Theta_3 + \ \dots + x_{id}\Theta_d$$

В матричной форме:

$$\hat{y} = X\Theta$$

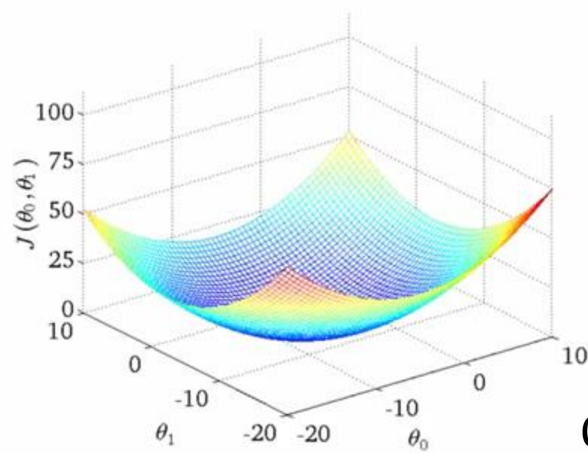
$$\begin{bmatrix} \hat{y}_1 \\ \dots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} x_{11} & \dots & x_{1d} \\ \dots & \dots & \dots \\ x_{n1} & \dots & x_{nd} \end{bmatrix} \begin{bmatrix} \Theta_1 \\ \dots \\ \Theta_d \end{bmatrix}$$

Задача восстановления регрессии

Линейная регрессия

<u>Целевая функция</u>:

$$J(\Theta) = (y - X\Theta)^T(y - X\Theta) = \sum_{i=1}^{n}(y_i - X_i^T\Theta)^2$$



$\Theta \in R^2$

Задача восстановления регрессии

Линейная регрессия

Целевая функция:

$$J(\Theta) = (y - X\Theta)^T(y - X\Theta) = \sum_{i=1}^{n}(y_i - X_i^T\Theta)^2$$

Поиск решения:

$$\frac{\partial J(\Theta)}{\partial \Theta} = \frac{\partial}{\partial \Theta}(y^T y - 2y^T x\Theta + \Theta^T x^T x\Theta) = 0$$

$$\Theta = (x^T x)^{-1} x^T y$$

Задача восстановления регрессии

Оптимизация

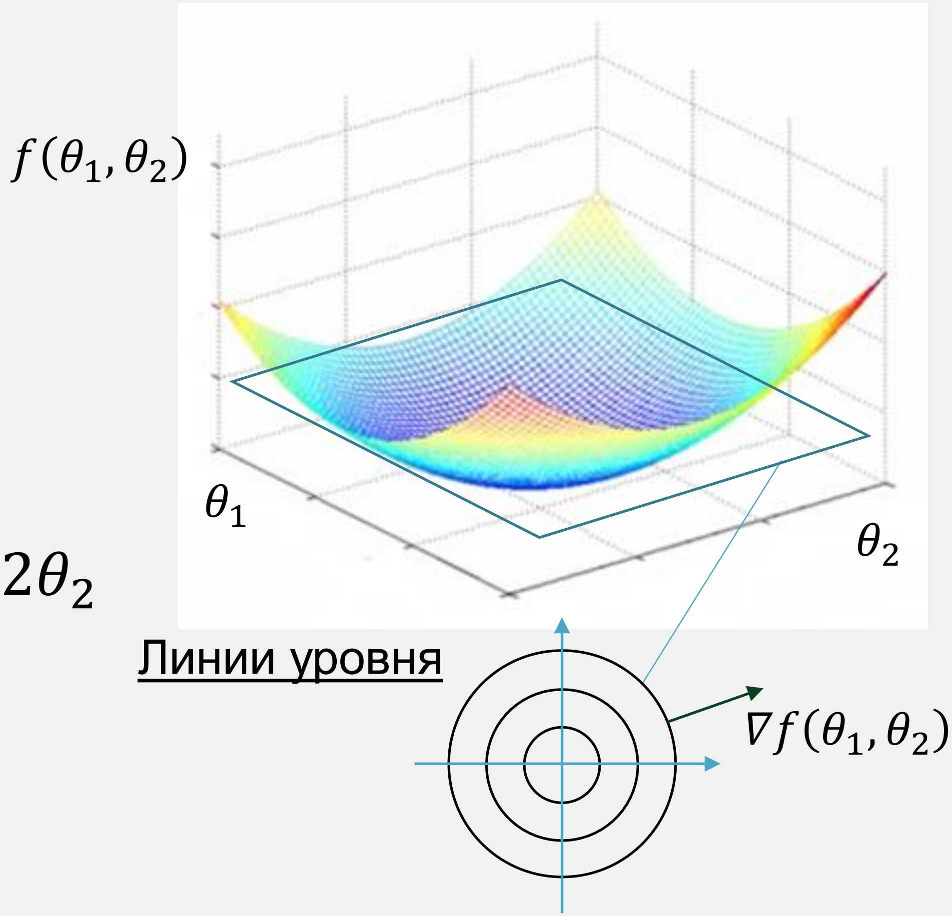
Функция:

$$f(\theta_1, \theta_2) = \theta_1{}^2 + \theta_2{}^2$$

Частные производные:

$$\frac{\delta f(\theta_1,\theta_2)}{\delta \theta_1} = 2\theta_1, \frac{\delta f(\theta_1,\theta_2)}{\delta \theta_2} = 2\theta_2$$

$$\nabla f(\theta_1, \theta_2) = \begin{bmatrix} 2\theta_1 \\ 2\theta_2 \end{bmatrix}$$



Задача восстановления регрессии

Проблема переобучения

1)Обучающая выборка

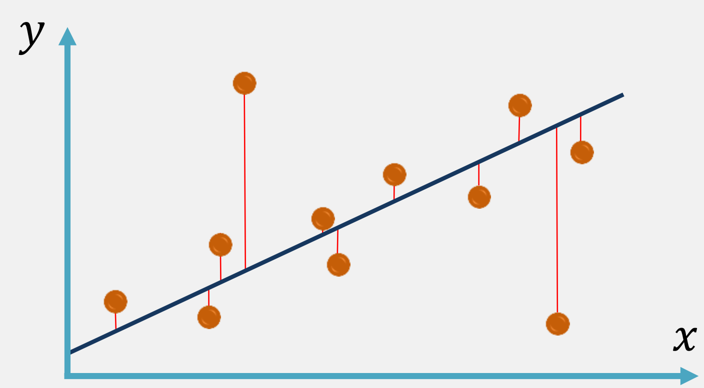
2)Контрольная выборка

Пример

Модель: $h(X, \theta) = \theta_0 + \theta_1 \cdot x + \dots + \theta_n x^n$

Целевая функция: $J(X, \Theta) = \sum_{i=0}^{n} (\theta_0 + \theta_1 \cdot x_i + \dots + \theta_n x_i^n - y_i)^2$

Что будет, если увеличить n?

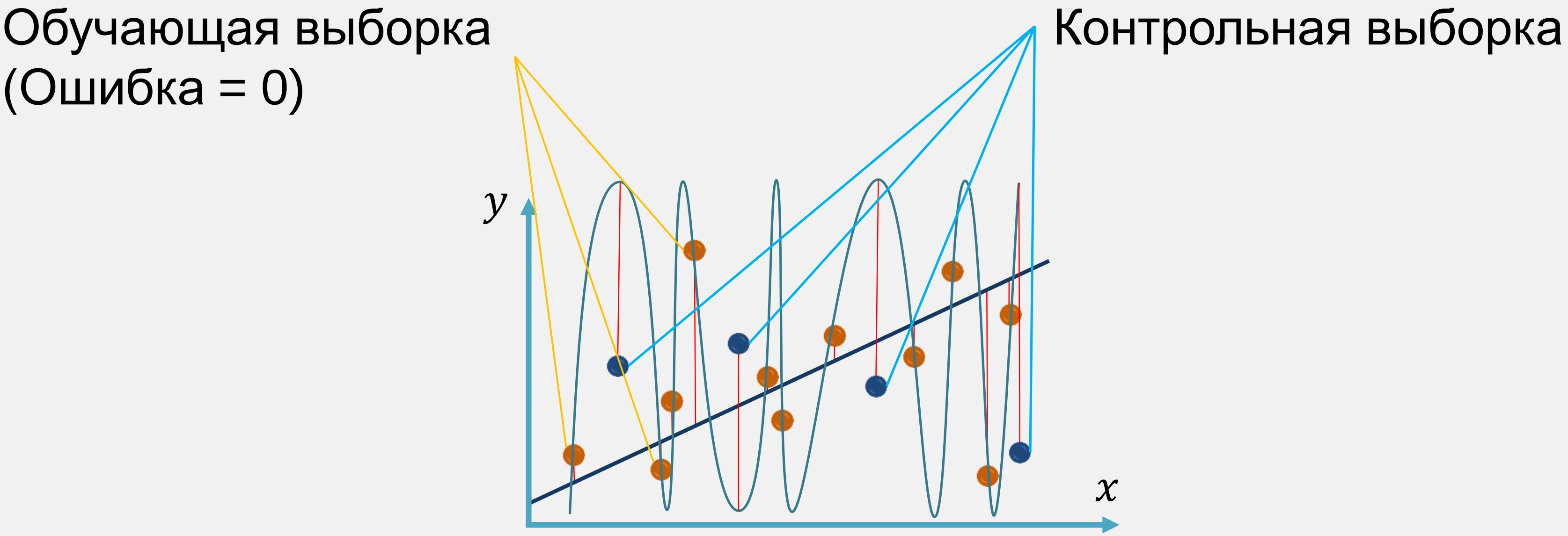


Задача восстановления регрессии

Проблема переобучения

1)Обучающая выборка

2)Контрольная выборка

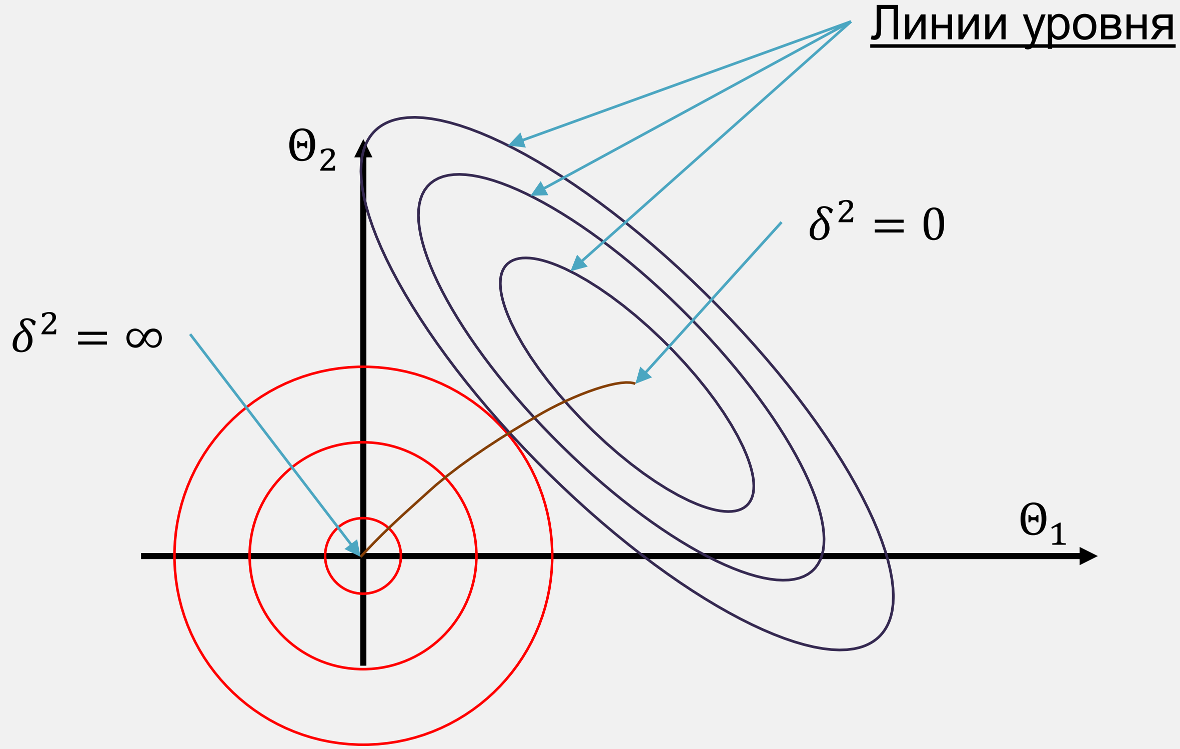


Задача восстановления регрессии

Регуляризация

Целевая функция:

$$J(\Theta) = (y - X\Theta)^T(y - X\Theta) + \delta^2\Theta^T\Theta$$



Задача восстановления регрессии

Регуляризация

L1 регуляризация:

$$J(\Theta) = \sum_j (y_j - \sum_i (x_i^j * \Theta_i))^2 + \delta^2 \sum_i |\Theta_i|$$

L2 регуляризация или регуляризация Тихонова:

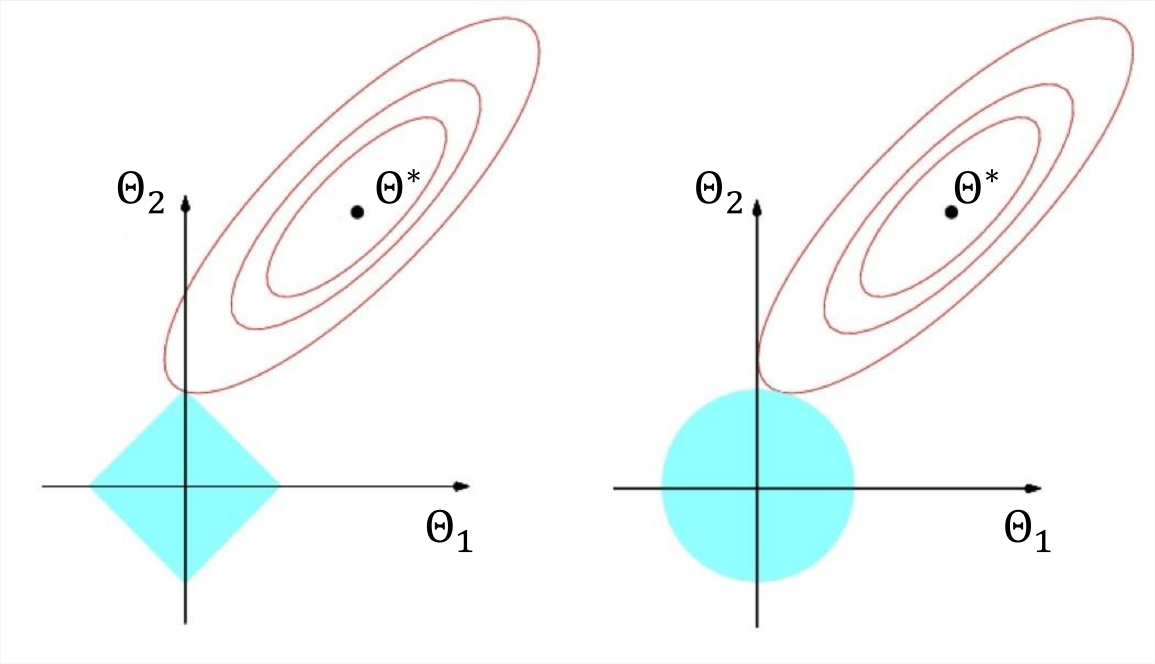
$$J(\Theta) = \sum_j (y_j - \sum_i (x_i^j * \Theta_i))^2 + \delta^2 \sum_i (\Theta_i)^2$$

Elastic Net

$$J(\Theta) = \sum_j (y_j - \sum_i (x_i^j * \Theta_i))^2 + \alpha\rho \sum_i |\Theta_i| + \alpha \frac{(1-\rho)}{2} \sum_i (\Theta_i)^2$$

Задача восстановления регрессии

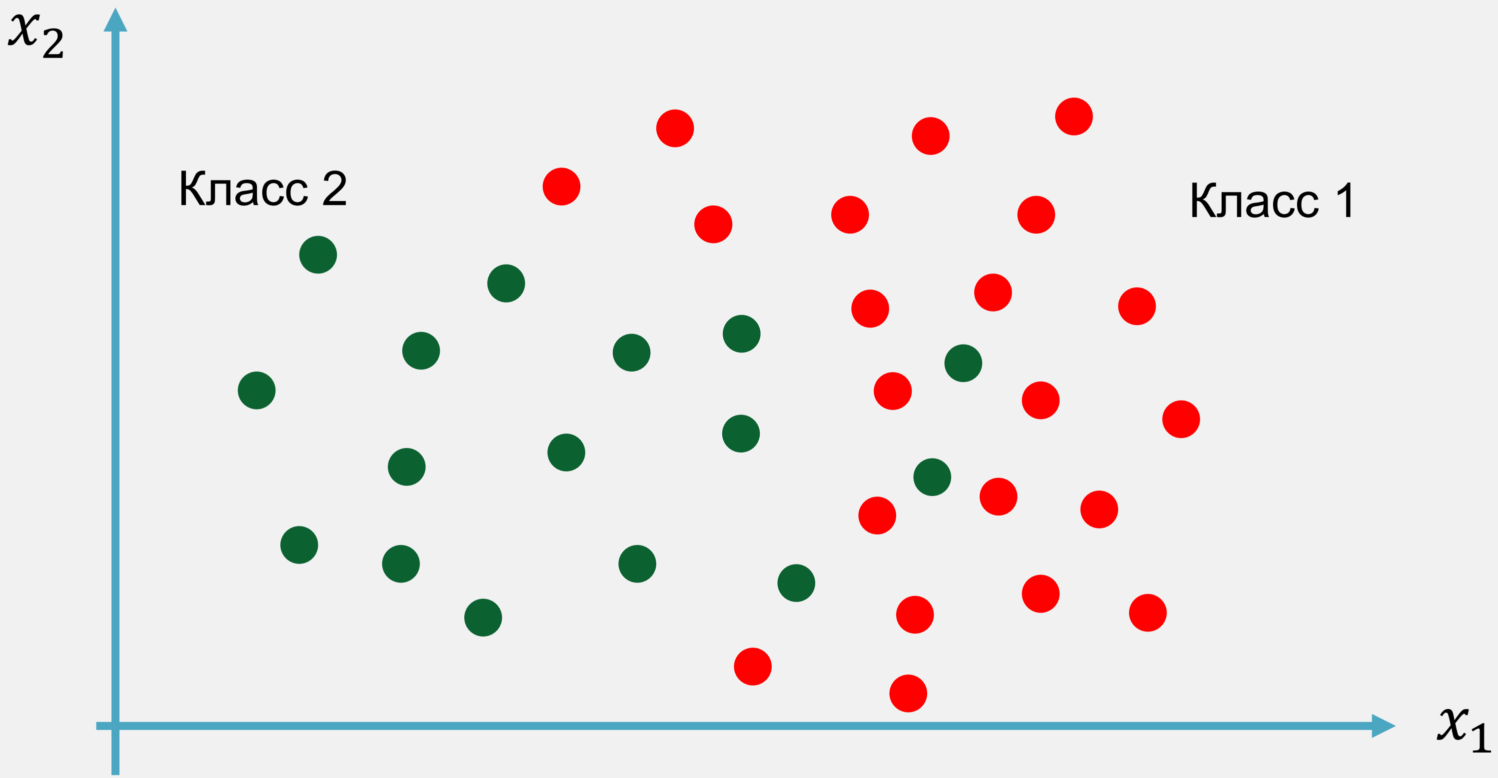
Регуляризация



L1 регуляризация L2 регуляризация

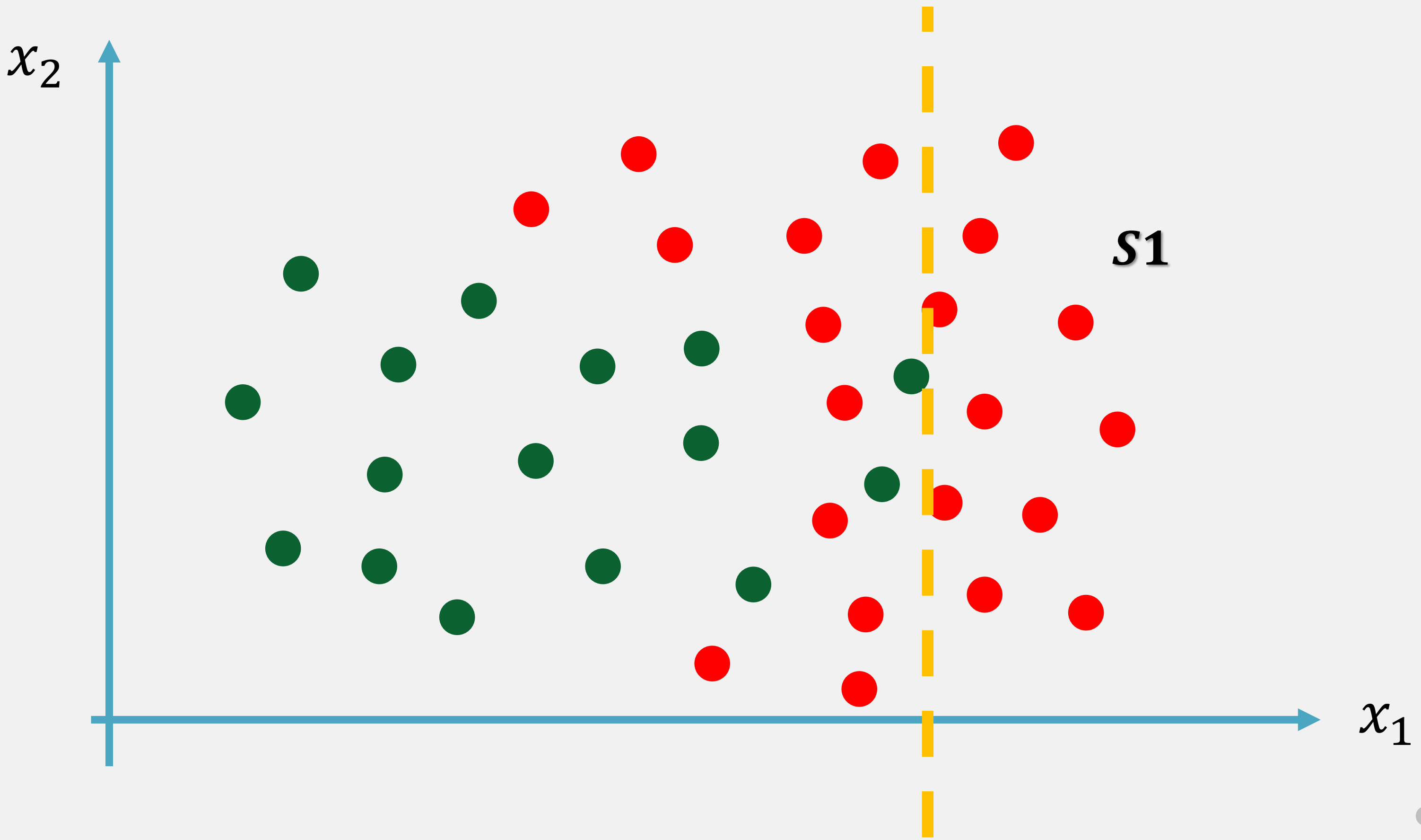
Задача классификации

Бинарное решающее дерево



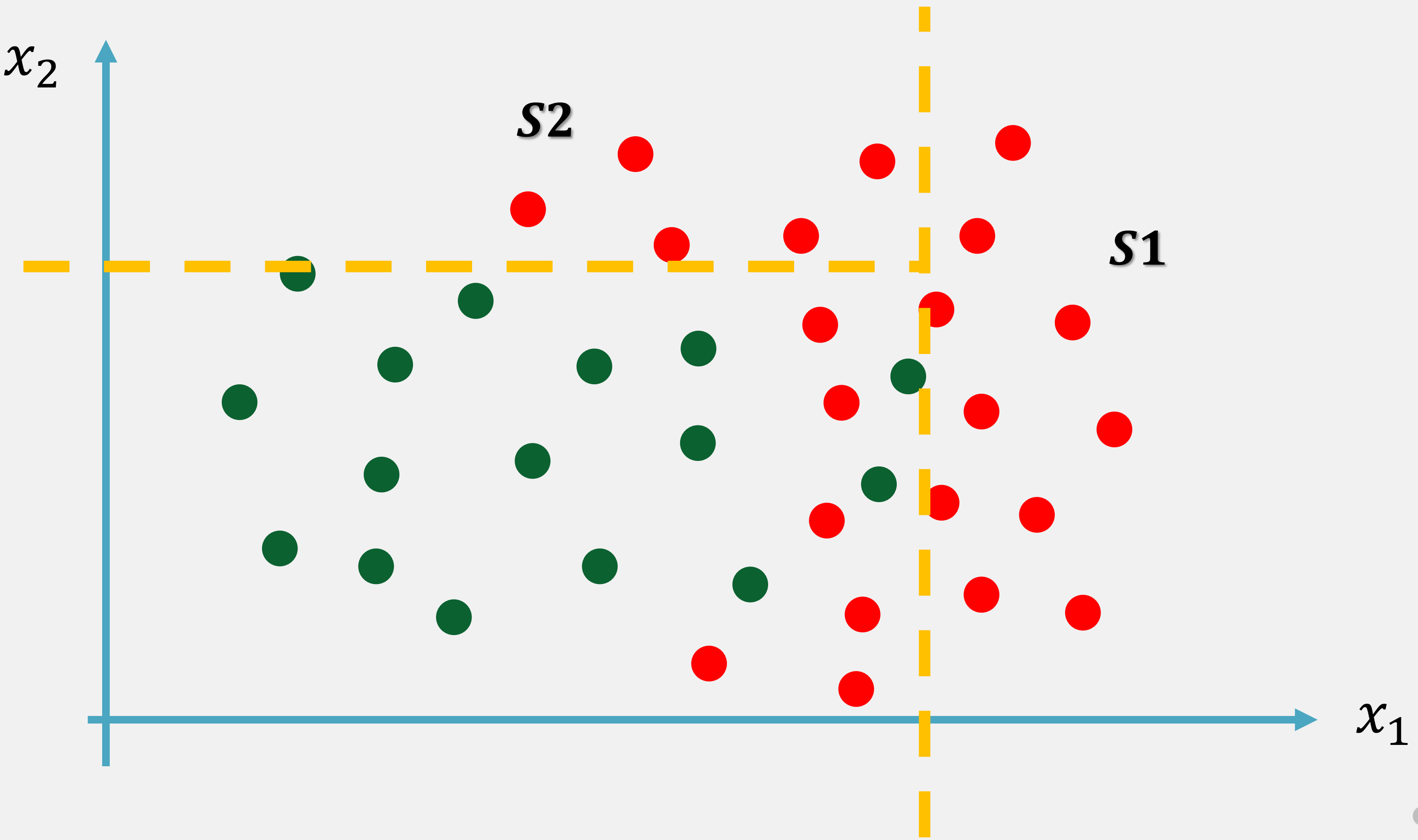
Задача классификации

Бинарное решающее дерево



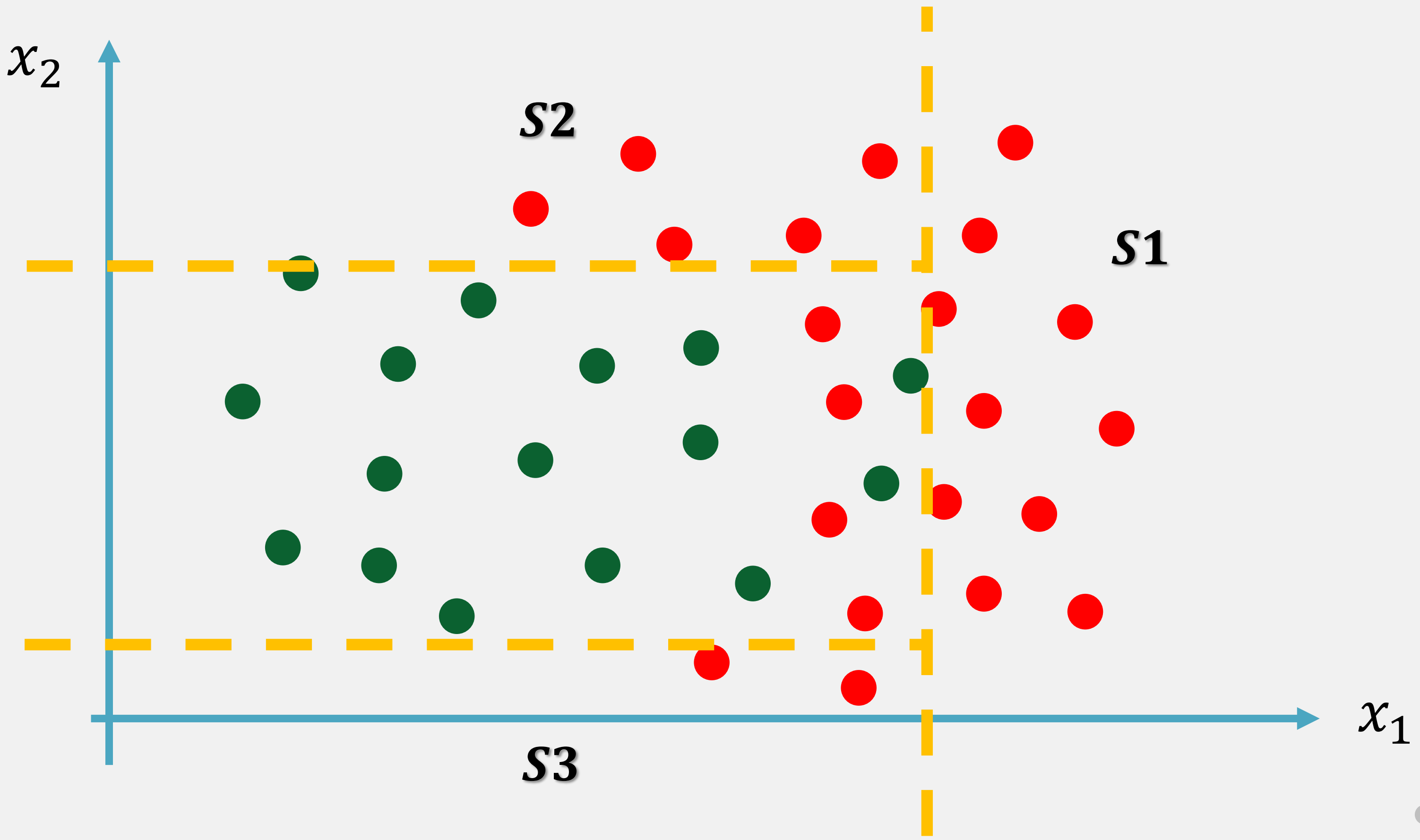
Задача классификации

Бинарное решающее дерево



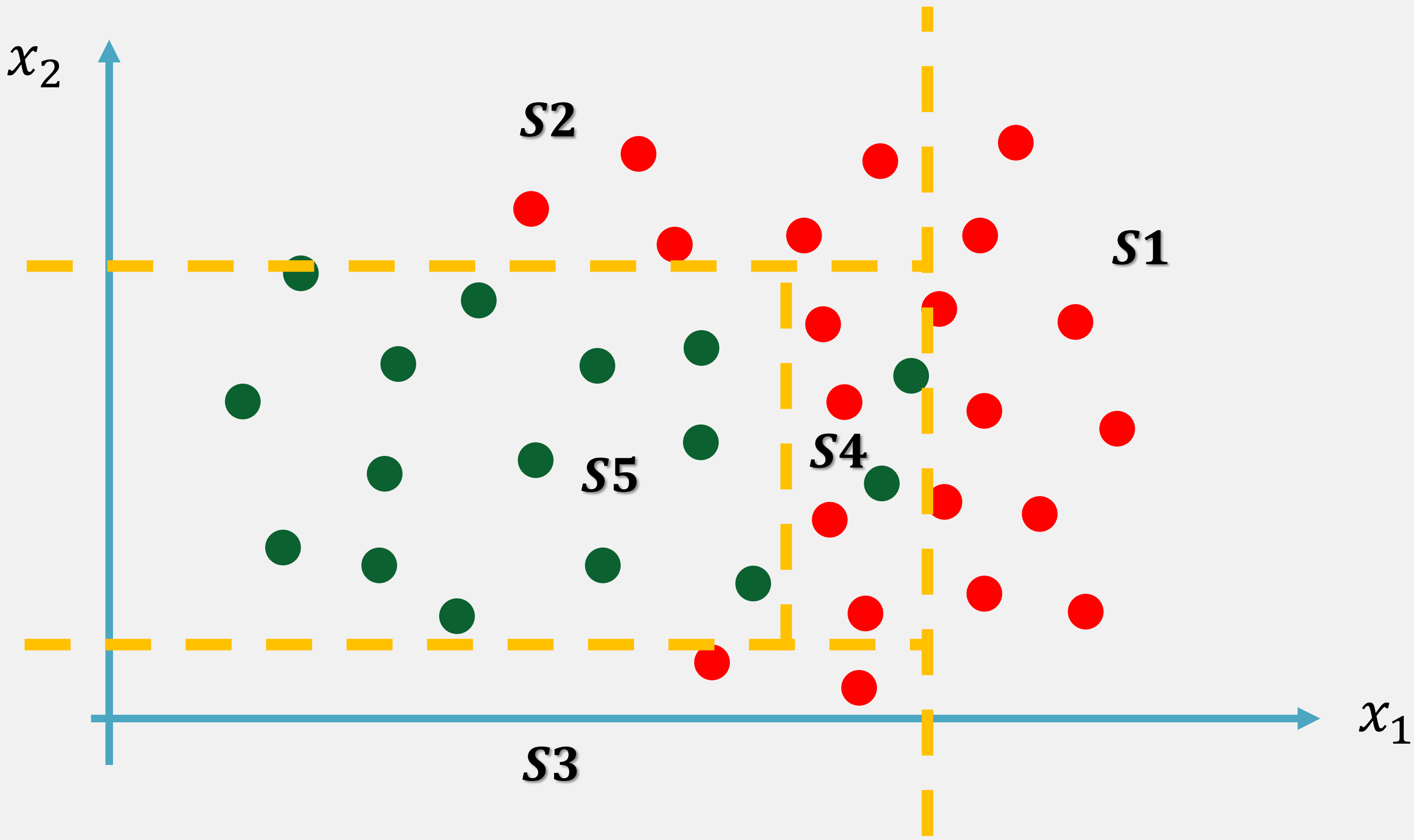
Задача классификации

Бинарное решающее дерево



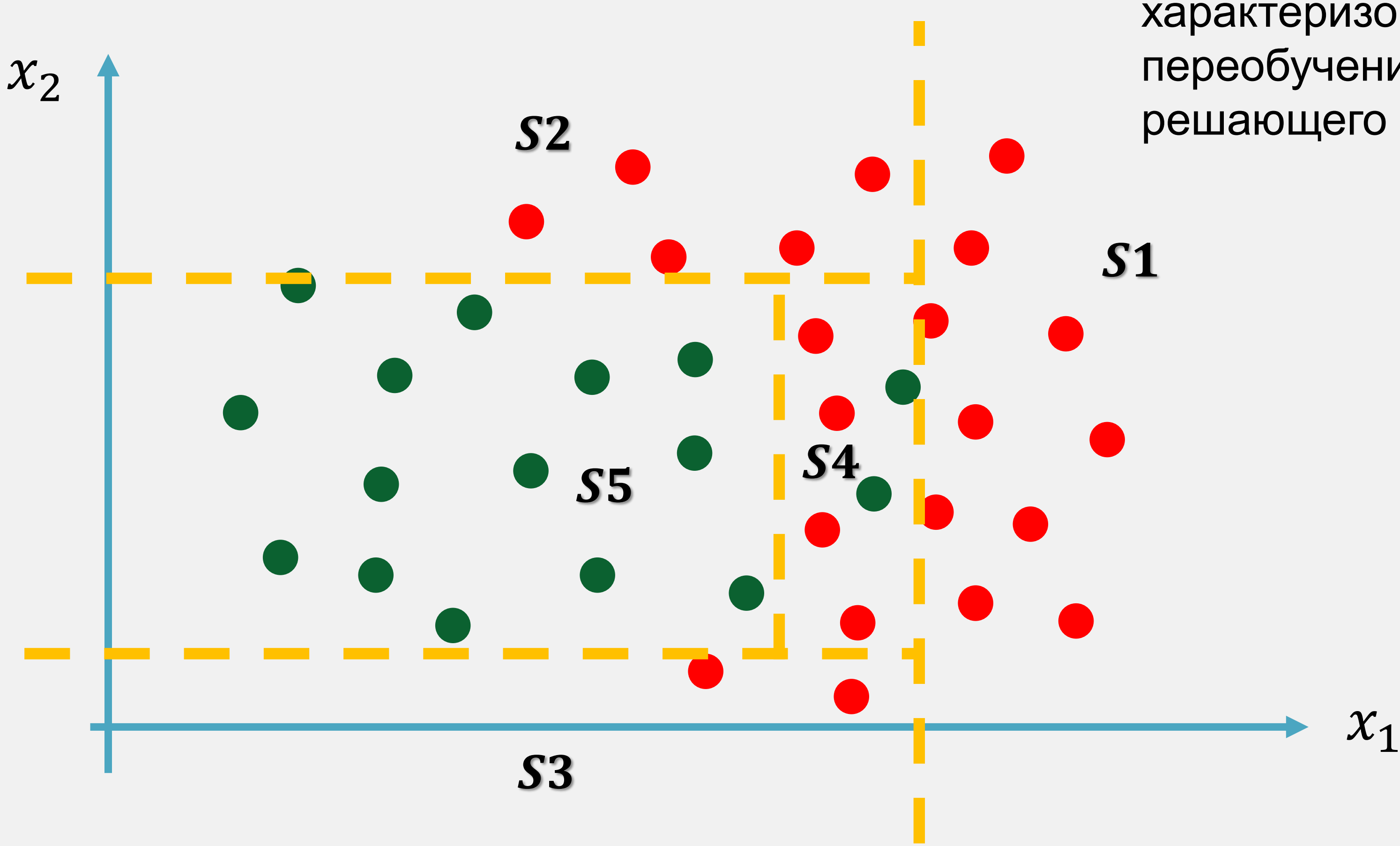
Задача классификации

Бинарное решающее дерево



Задача классификации

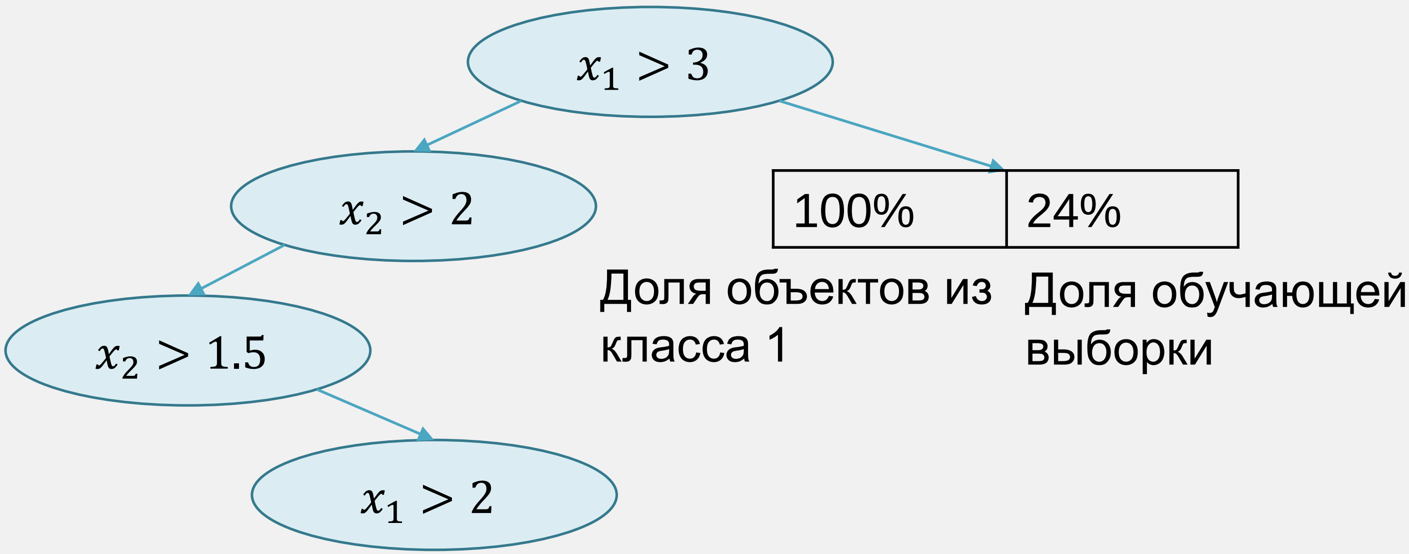
Бинарное решающее дерево



Чем будет характеризоваться переобучения для решающего дерева?

Задача классификации

Бинарное решающее дерево



Задача классификации

Бинарное решающее дерево

Критерий разбиения

Имеется множество: X

Мера неоднородности: $H(X)$

Решаем задачу бинарной классификации

1) Misclassification criteria: $H(X) = 1 - p_{max}$

2) Entropy criteria: $H(X) = -p_0 \ln(p_0) - p_1 \ln(p_1)$

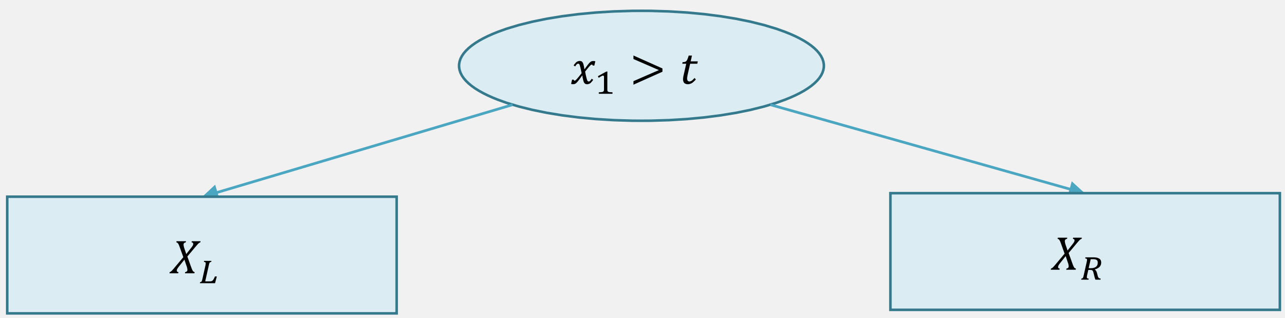
3) Gini criteria: $H(X) = 2p_0 p_1$

где p_0 и p_1 - доли объектов из класса 0 и 1

Задача классификации

Бинарное решающее дерево

Критерий разбиения



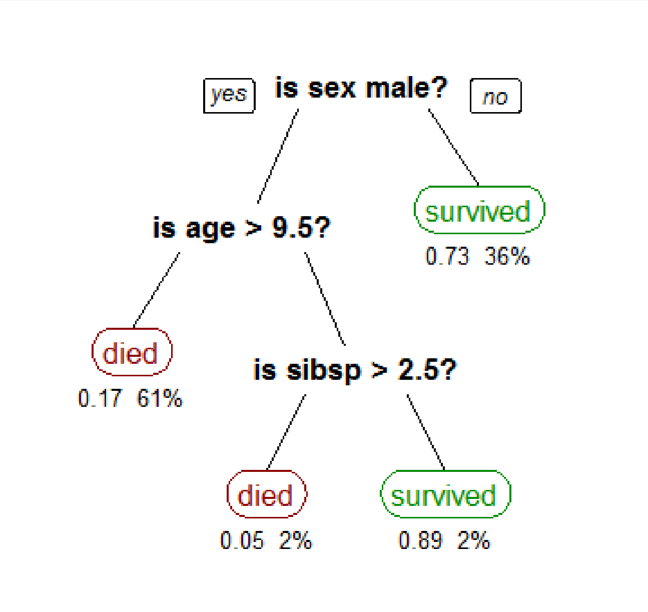
Уменьшения неопределённости (неоднородности) в узле:

$$G(X) = \frac{|X_L|}{|X|} H(X_L) + \frac{|X_R|}{|X|} H(X_R) \to min$$

Задача классификации

Бинарное решающее дерево

Titanic dataset



https://www.kaggle.com/c/titanic

Задача классификации

Алгоритм ближайшего соседа

Имеется обучающая: X^l, y

Метрика схожести объектов: ρ

Для нового объекта u располагаем элементы так, чтобы:

$$\rho\left(u, x_{1,u}\right) \leq \rho\left(u, x_{2,u}\right) \leq \cdots \leq \rho(u, x_{l,u})$$

Тогда объекту u будет относится к тому классу, которому принадлежит объект $x_{1,u}$

Каждый новый объект порождает собственную упорядоченную последовательность элементов $x_{i,u}$

Задача классификации

Алгоритм ближайшего соседа

Достоинства:

Простота реализации

Недостатки:

Неустойчивость к шумам

Параметры настройки алгоритма:

Метрика схожести объектов

Задача классификации

Алгоритм k ближайших соседей

Имеется обучающая: X^l, y

Метрика схожести объектов: ρ

Для нового объекта u располагаем элементы так, чтобы:

$$\rho(u, x_{1,u}) \leq \rho(u, x_{2,u}) \leq \cdots \leq \rho(u, x_{l,u})$$

$$\hat{y}_u = arg \max_{y \in Y} \sum_{i=1}^{k} [y_{i,u} == y]$$

где k – число соседей, голосующих за отнесение объекта u к классу $\hat{y}_u$

Задача классификации

Модификации алгоритма k ближайших соседей

Добавление весов каждому объекту $x_{1,u}, x_{2,u}, \dots, x_{k,u}$

Вес объекта $x_{i,u}$ можно выбирать в зависимости от удалённости от объекта u

Выбор только тех объектов, которые попали в заранее выбранный радиус r

Если выборка достаточно большая, то можно выполнять прореживание путём выбрасывания неинформативных объектов

Задача классификации

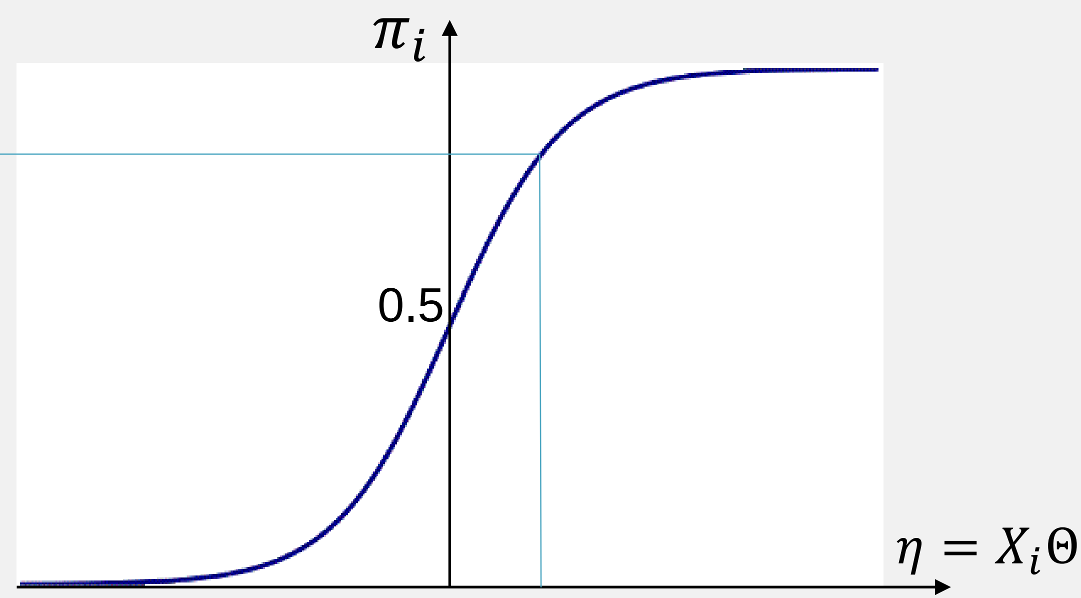
Логистическая регрессия

Рассматривается задача бинарной классификации (0 или 1)

Функция сигмоида

$$\hat{y}_i = sigm(X_i\Theta\ ) = sigm(\eta) = \frac{1}{1+e^{-\eta}} = \frac{e^{\eta}}{e^{\eta}+1}$$

$p(y_i = 1|X_i\Theta) = \pi_i$

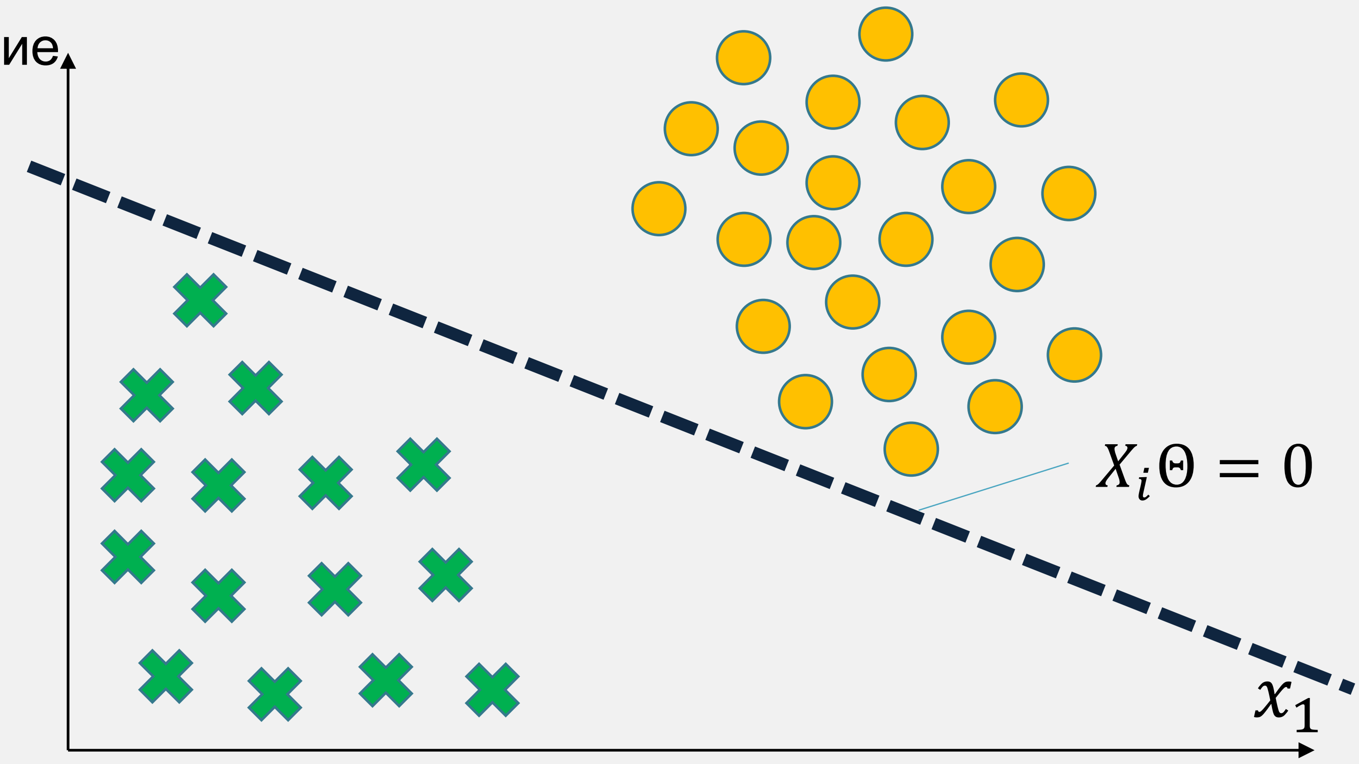


Определяется вероятность принадлежности объекта X_i классу 1

Задача классификации

Логистическая регрессия

Линейное разделение гиперплоскостью



Задача классификации

Логистическая регрессия

Оптимизируемый функционал

Будем максимизировать вероятность попадания объектов из обучающей выборки к соответствующим им классам

$$p(y|X\Theta) = \prod_{i=1}^{n} Ber(y_i|sigm(X_i\Theta)) = \prod_{i=1}^{n} \left[\frac{1}{1+e^{-X_i\Theta}}\right]^{y_i} \left[1 - \frac{1}{1+e^{-X_i\Theta}}\right]^{1-y_i},$$

где $X_i\Theta = \Theta_0 + \sum_{j=1}^{d} \Theta_j x_j$.

Возьмём логарифм от полученного произведения (положение максимума не изменится) и умножим на -1 для решения задачи минимизации функционала

$$C(\Theta) = -\log\big(p(y|X\Theta)\big) = -\sum_{i=1}^{n} (y_i \log(\pi_i) + (1-y_i)\log(1-\pi_i))$$

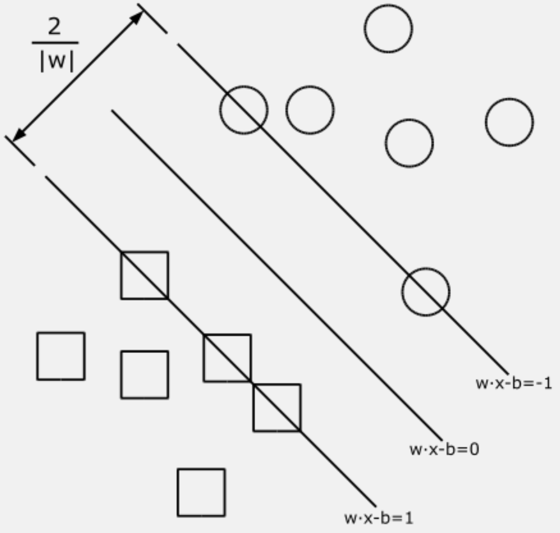
Задача классификации

Метод опорных векторов

$$\hat{y}_i = sigm(X_i \Theta + b)$$

Идея метода:

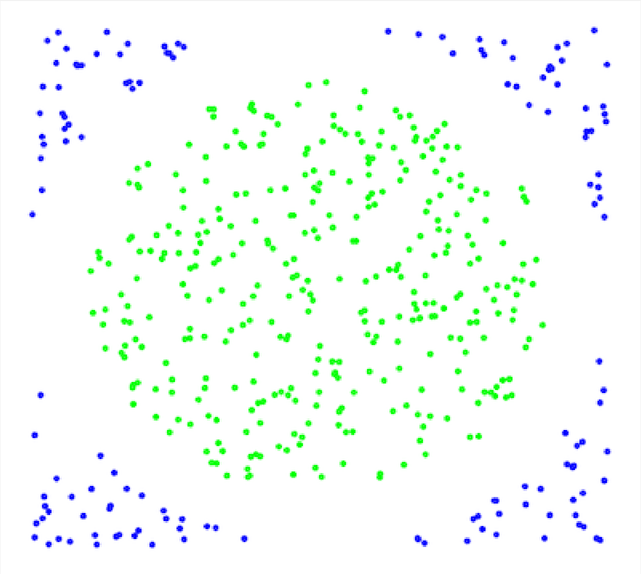
Поиск таких параметров Θ и b, которые максимизируют расстояние до каждого класса.



Задача классификации

Метод опорных векторов

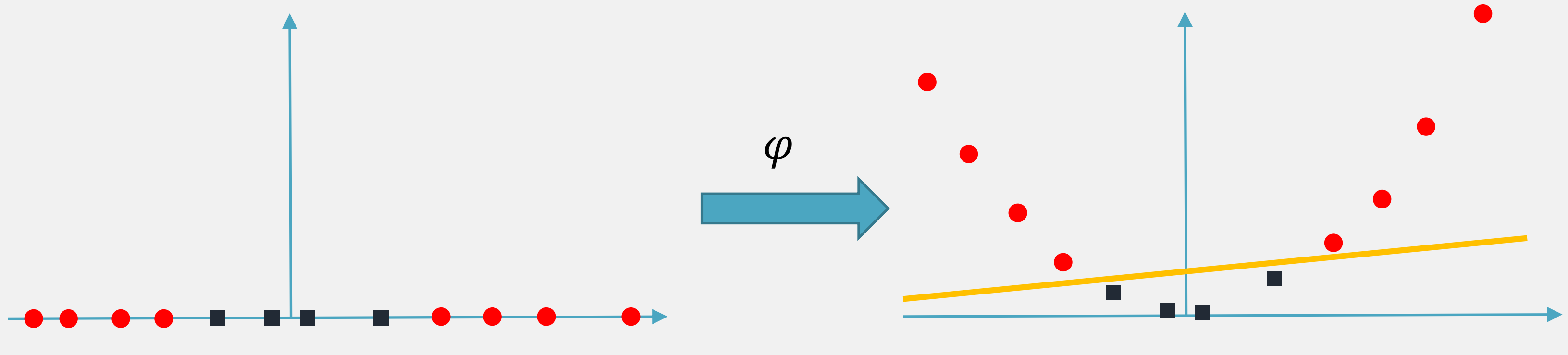
Чаще всего достаточно сложно разделить объекты обучающей выборки гиперплоскостью



Задача классификации

Метод опорных векторов

Для решения проблемы линейной неразделимости использую переход описания объектов из X в пространство H с использованием преобразования $\varphi: X \to H$. Пространство H называют спрямляюшим.



Функция $K : X \times X \to R$ называется ядром (kernel function)

Метрики

Классификация

1. Accuracy
2. Precision
3. Recall
4. ROC-AUC
5. F-метрика
6. Logloss

Метрики

Регрессия

1. MAE
2. MSE
3. RMSE
4. RMSLE
5. R2

Метрики

Классификация

Accuracy

Подсчитываем долю правильно предсказанных объектов

Может быть использована в многоклассовой классификации

1. import numpy as np
2. target = np.array([1, 3, 2, 2, 3, 4, 1, 2])
3. pred = np.array([1, 3, 1, 2, 3, 2, 1, 2])
4. print(np.equal(target, pred).sum())
5. print(np.equal(target, pred).sum()/float(target.shape[0]))

Результат:
6
0.75

Когда могут возникнуть проблемы?

Метрики

Классификация

Accuracy

Подсчитываем долю правильно предсказанных объектов

Может быть использована в многоклассовой классификации

```
1. import numpy as np
2. target = np.array([1, 1, 1, 2, 1, 1, 1, 2])
3. pred = np.array([1, 1, 1, 2, 1, 1, 1, 1])
4. print(np.equal(target, pred).sum())
5. print(np.equal(target, pred).sum()/float(target.shape[0]))
```

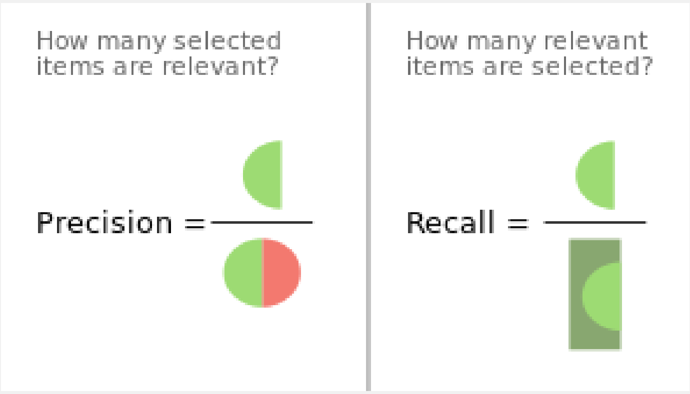
Результат:
7
0.875

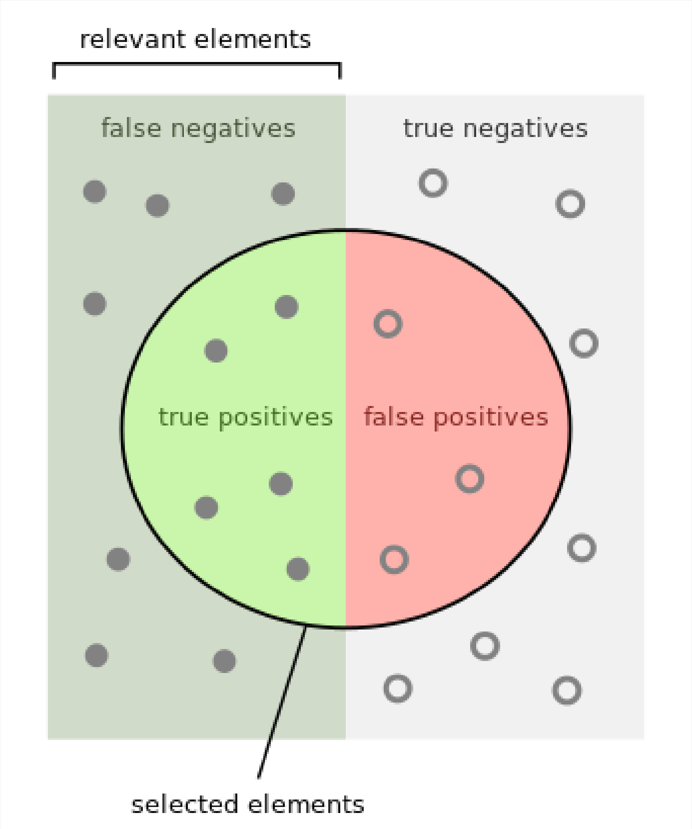
Метрики

Классификация

Precision, Recall

Задачи бинарной классификации





Метрики

Классификация

Precision, Recall

	Предсказали True	Предсказали False
Ожидали True	True Positive (*tp*)	False Negative (*fn*)
Ожидали False	False Positive (*fp*)	True Negative (*tn*)

Полнота $$Recall = \frac{tp}{tp + fn}$$

Recall: Какую часть из объектов класса 1 мы нашли?

Точность $$Precision = \frac{tp}{tp + fp}$$

Precision: Какая часть из тех объектов класса 1, которую мы нашли, действительно принадлежат этому классу?

Метрики

Классификация

Precision, Recall

```
1. target = np.array([0, 1, 1, 0, 1, 1])
2. pred = np.array([1, 0, 1, 0, 1, 0])
3. print(recall(target, pred))
4. print(precision(target, pred))
```

Результат:

0.5	2/(2+2)	Recall: Какую часть из объектов класса 1 мы нашли?
0.67	2/(2+1)	Precision: Какая часть из тех объектов класса 1, которую мы нашли, действительно принадлежат этому классу?

Метрики

Классификация

Precision, Recall

1. target = np.array([0, 1, 1, 0, 1, 1])
2. pred = np.array([1, 1, 1, 1, 1, 1])
3. print(recall(target, pred))
4. print(precision(target, pred))

Результат:

0.1	4/(4+0)	Recall: Какую часть из объектов класса 1 мы нашли?
0.67	4/(4+2)	Precision: Какая часть из тех объектов класса 1, которую мы нашли, действительно принадлежат этому классу?

Метрики

Классификация

$F1$-score

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

```
1. target = np.array([0, 1, 1, 0, 1, 1])
2. pred = np.array([1, 0, 1, 0, 1, 0])
3. print(f1_score(target, pred))
```

Результат:
0.57

```
1. target = np.array([0, 1, 1, 0, 1, 1])
2. pred = np.array([1, 1, 1, 1, 1, 1])
3. print(f1_score(target, pred))
```

Результат:
0.8

Метрики

Классификация

$F1$-score

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

```
1. target = np.array([0, 1, 0, 0, 0, 0])
2. pred = np.array([1, 1, 1, 0, 1, 0])
3. print(f1_score(target, pred))
```

Результат:
0.4

```
1. target = np.array([0, 1, 0, 0, 0, 0])
2. pred = np.array([1, 1, 1, 1, 1, 1])
3. print(f1_score(target, pred))
```

Результат:
0.29

Skewness…

Метрики

Классификация

F_β-score

$$F_\beta = (1+\beta^2) * \frac{Precision*Recall}{(\beta^2)Precision+Recall}$$

$0 < \beta < 1$ – предпочтение точности ($Precision$)

$1 < \beta$ – предпочтение полноте ($Recall$)

Метрики

Классификация

$ROC - AUC$

ROC - Receiver Operating Characteristic

AUC - Area Under the Curve

Определяет долю правильно отранжированных пар

$$TPR(Recall) = \frac{tp}{tp + fn}$$

The Relationship Between Precision-Recall and ROC Curves
Jesse Davis, Mark Goadrich

$$FPR = \frac{fp}{fp + tn}$$

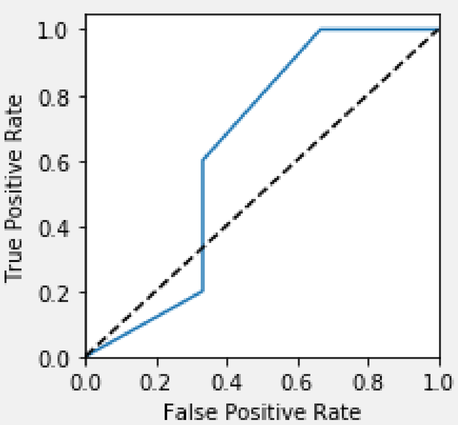
https://alexanderdyakonov.wordpress.com/2015/10/09/задачки-про-auc-roc/

Метрики

Классификация

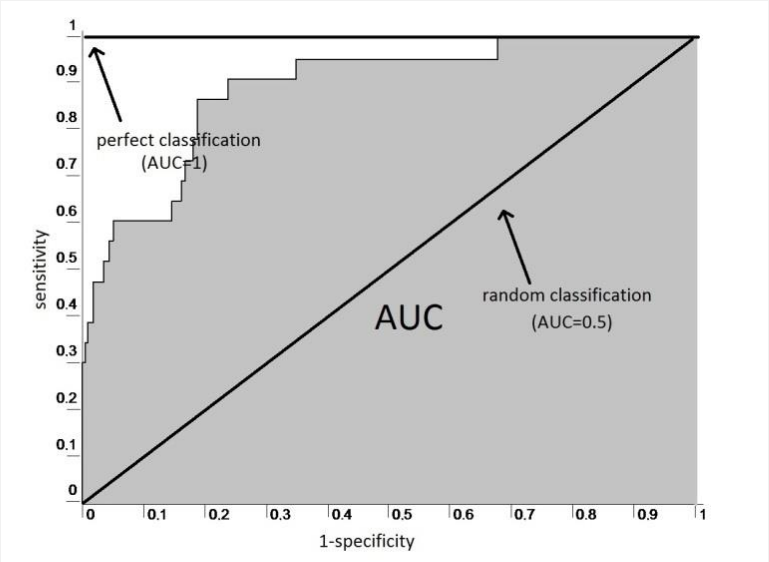
$ROC - AUC$

1. target = [0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0]
2. pred = [0.1, 0.3, 0.2, 0.4, 0.7, 0.8, 0.7, 0.8, 0.3, 0.4, 0.8]



$$TPR(Recall) = \frac{tp}{tp + fn}$$

$$FPR = \frac{fp}{fp + tn}$$



Метрики

Классификация

$Logloss$

Чем сильнее ошибаемся, тем больше ошибка

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{i,j}\log(p_{i,j})$$

N —размер выборки, M — количество классов

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(p_i) + (1 - y_i)\log(p_i))$$

https://www.kaggle.com/wiki/LogarithmicLoss

Метрики

Регрессия

MAE

<u>M</u>ean <u>A</u>bsolute <u>E</u>rror

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\widehat{y_i} - y_i|$$

Лучшее константное предсказание - медиана

$\hat{Y}$	Y
0.1	0
0.5	1
0.6	1
0.5	1
0.3	0

$$MAE = \frac{1}{5}(|0.1 - 0| + |0.5 - 1| + |0.6 - 1| + |0.5 - 1| + |0.3 - 0|)$$

Метрики

Регрессия

MSE (Mean Squared Error)

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

$RMSE$ (Root Mean Squared Error)

$$RMSE = \sqrt[2]{\frac{1}{n}\sum_{i=1}^{n}(\widehat{y}_i - y_i)^2}$$

Лучшее константное предсказание - среднее

Метрики

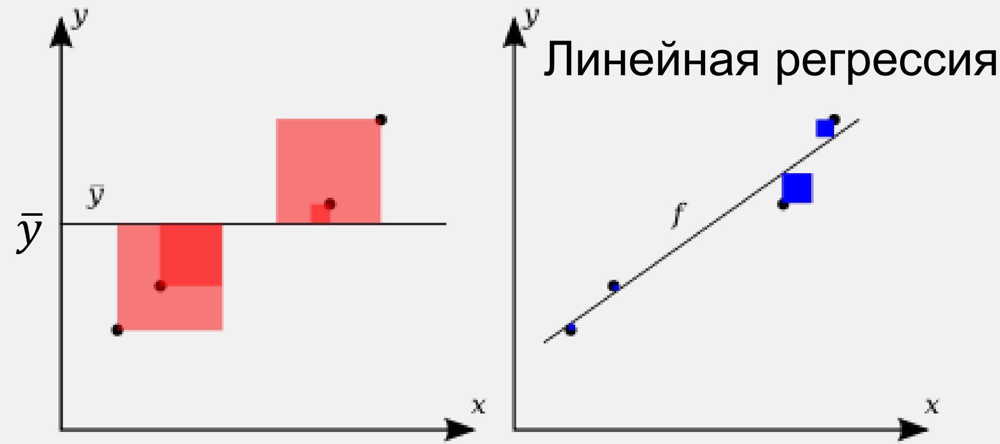
Регрессия

$$R2 - score$$

$$\bar{y} = \sum_{i=1}^{N} y_i$$

$$SS_{tot} = \sum_{i=1}^{N} (y_i - \bar{y})^2 \ - variance$$

$$SS_{res} = \sum_{i=1}^{N} (y_i \ - f_i)^2$$



SS_{tot} SS_{res}

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

Метрики

Кросс-валидация

Процедура эмпирического оценивания обобщающей способности алгоритмов, обучаемых по прецедентам.

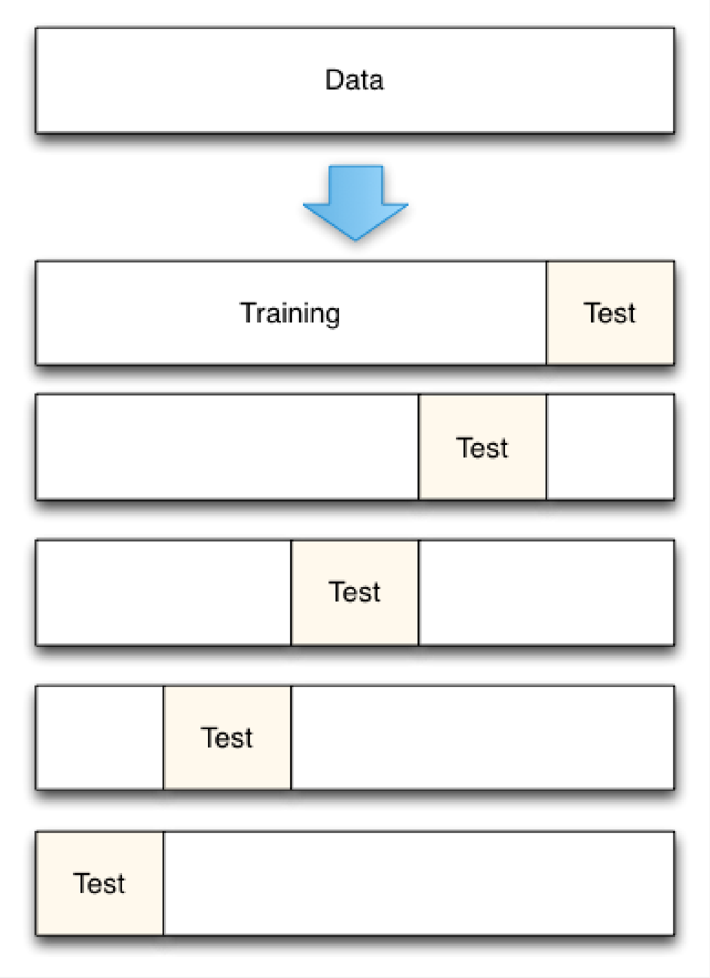
Фиксируется некоторое множество разбиений исходной выборки на две подвыборки:

1. обучающую
2. контрольную

Для каждого разбиения выполняется настройка алгоритма по обучающей подвыборке, затем оценивается его средняя ошибка на объектах контрольной подвыборки. Оценкой скользящего контроля называется средняя по всем разбиениям величина ошибки на контрольных подвыборках.

Метрики

Кросс-валидация



ТЕХНОСФЕРА

Спасибо за внимание!

Спасёнов Алексей

a.spasenov@corp.mail.ru