ТЕХНОСФЕРА

Лекция №2

Часть 2

Введение в машинное обучение

Спасёнов Алексей

Введение в машинное обучение. Лекция 2

Содержание лекции

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

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

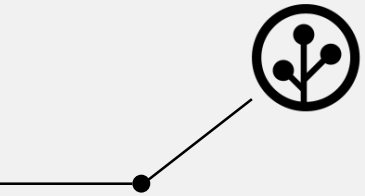
Задача восстановления регрессии (regression)

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

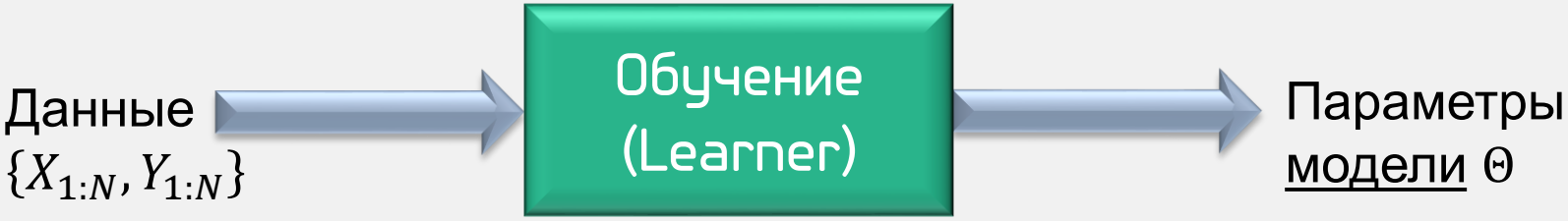
Задача ранжирования (learning to rank)

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

Обучение с учителем

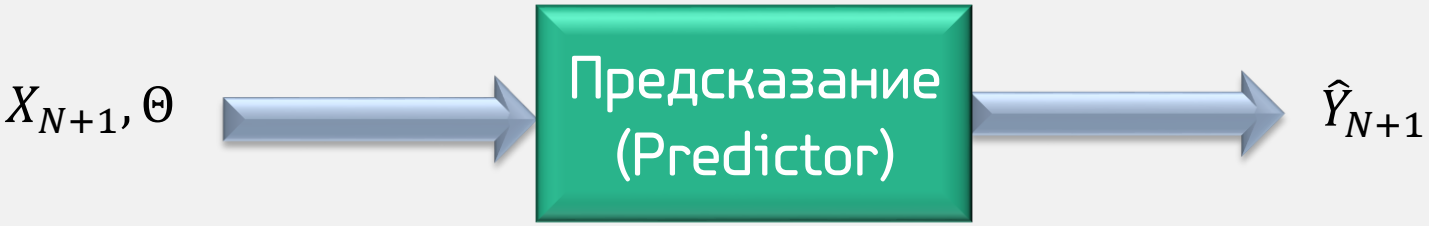


Этап обучения (train)



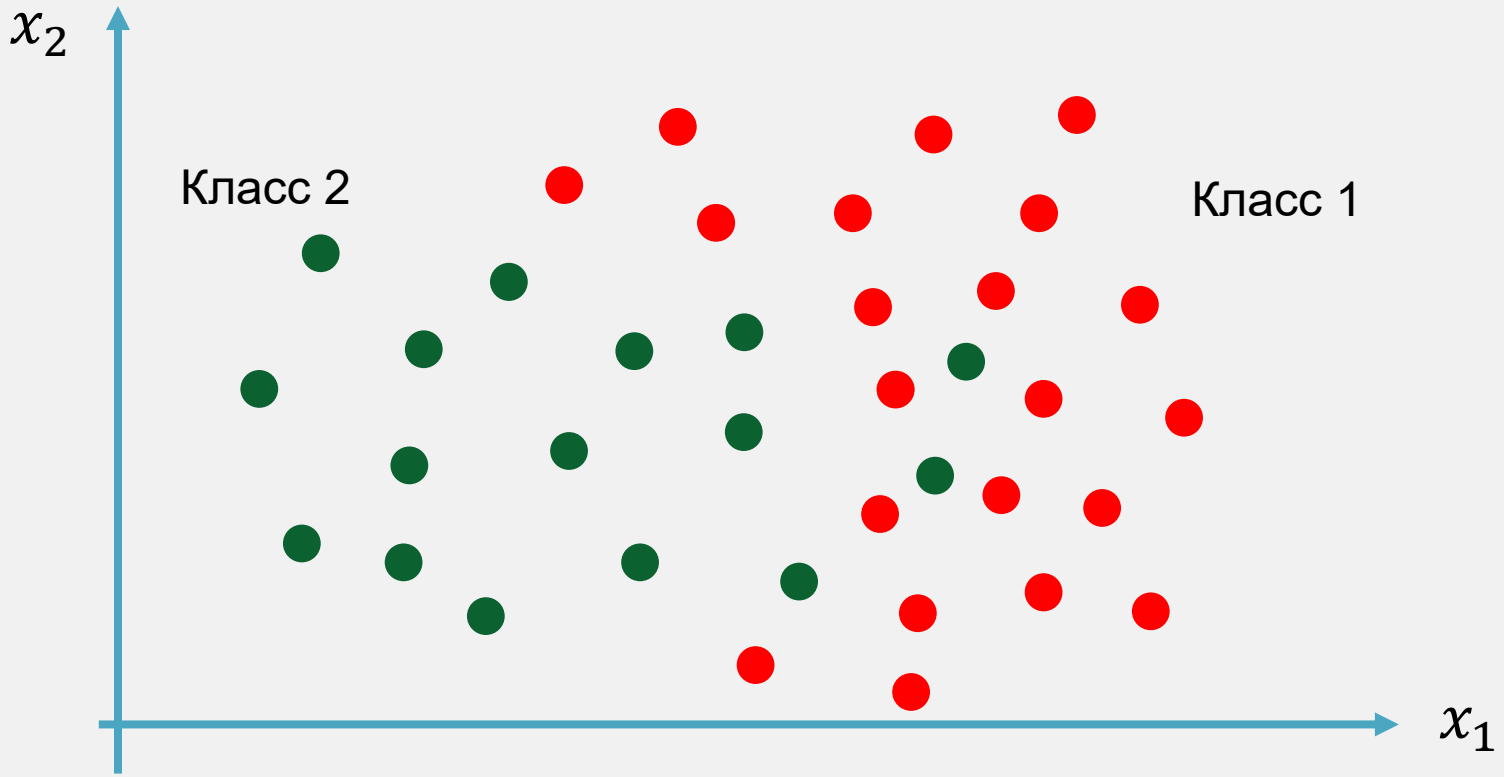
Необходимо учитывать представительность выборки

Этап применения (test)



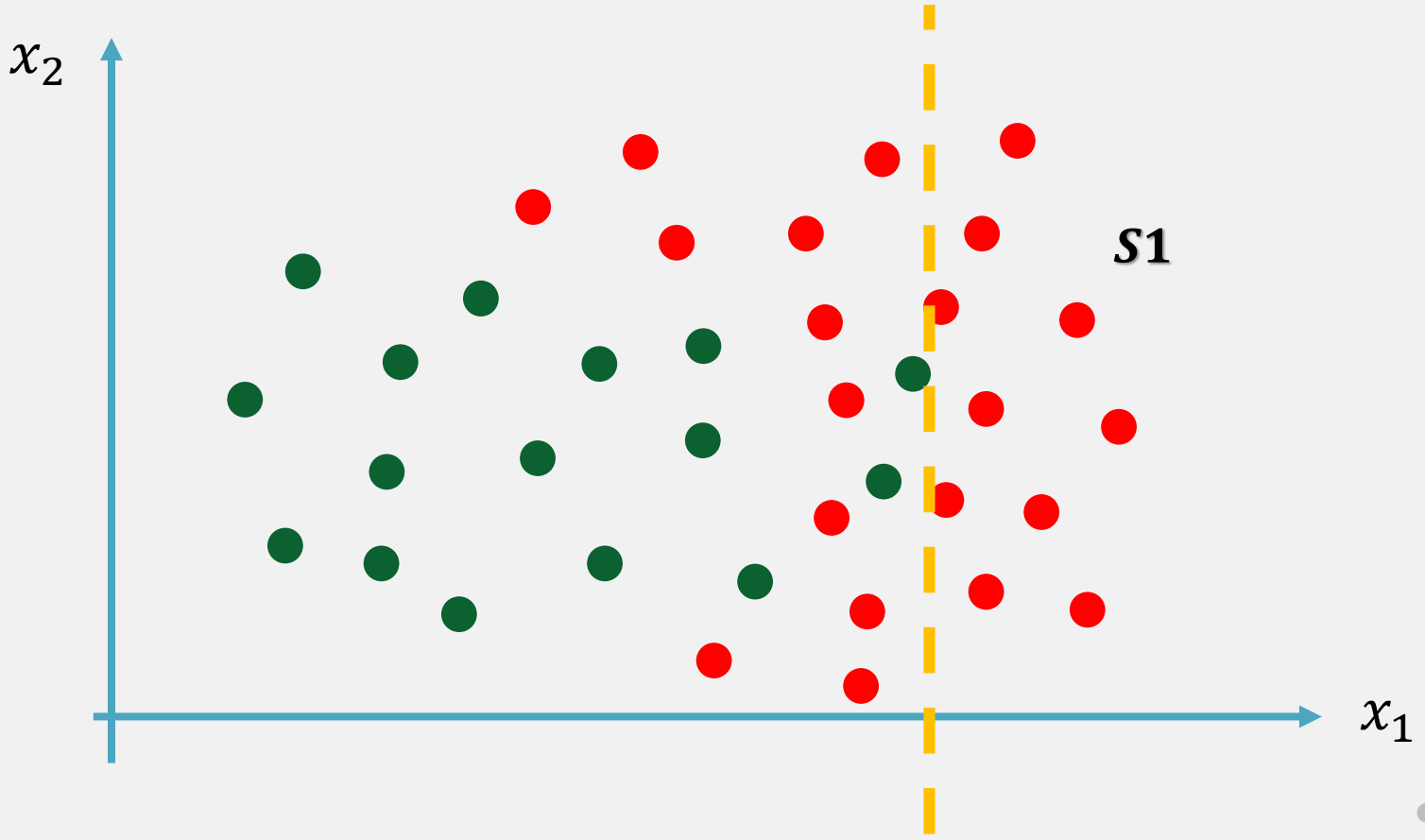
Задача классификации

Бинарное решающее дерево



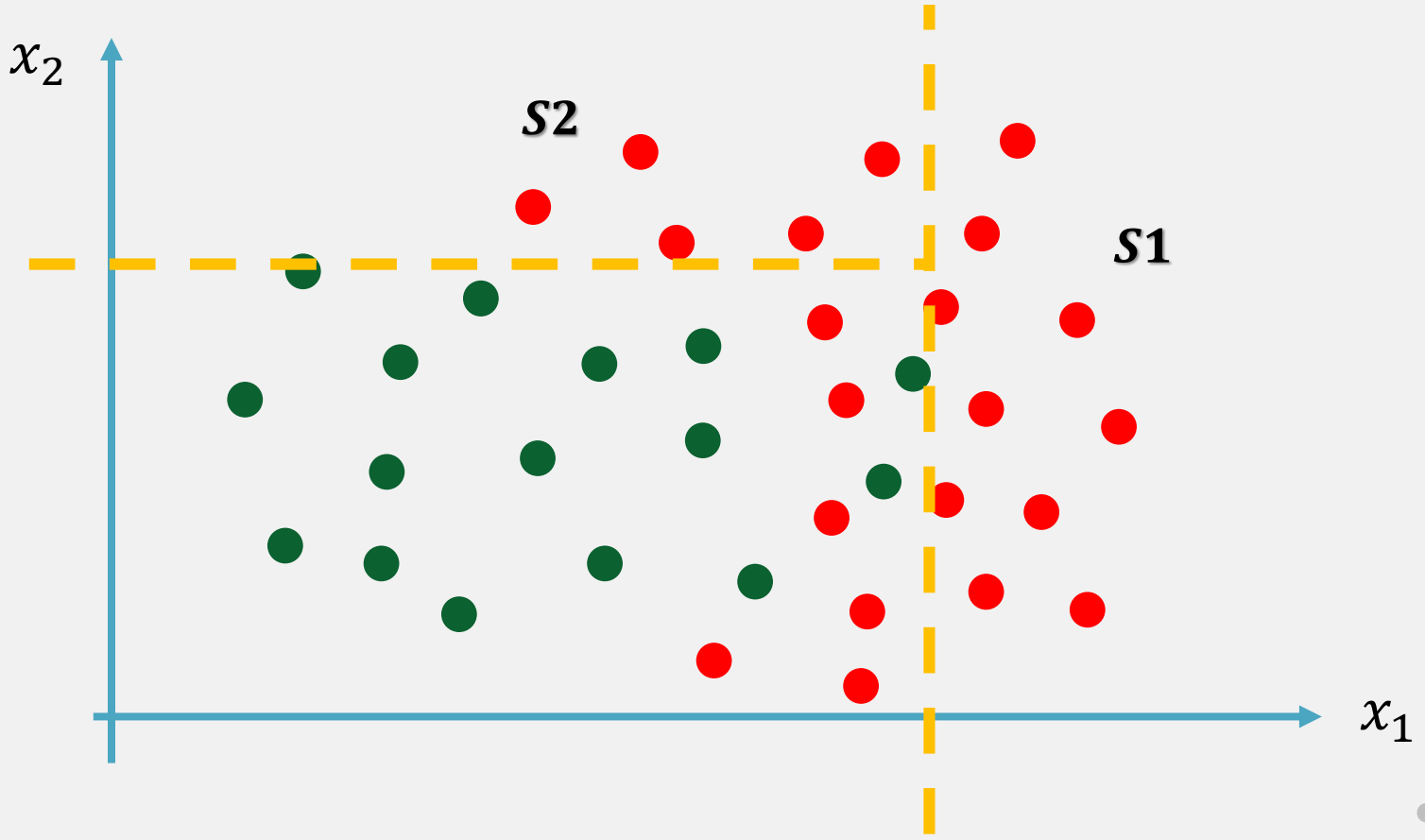
Задача классификации

Бинарное решающее дерево



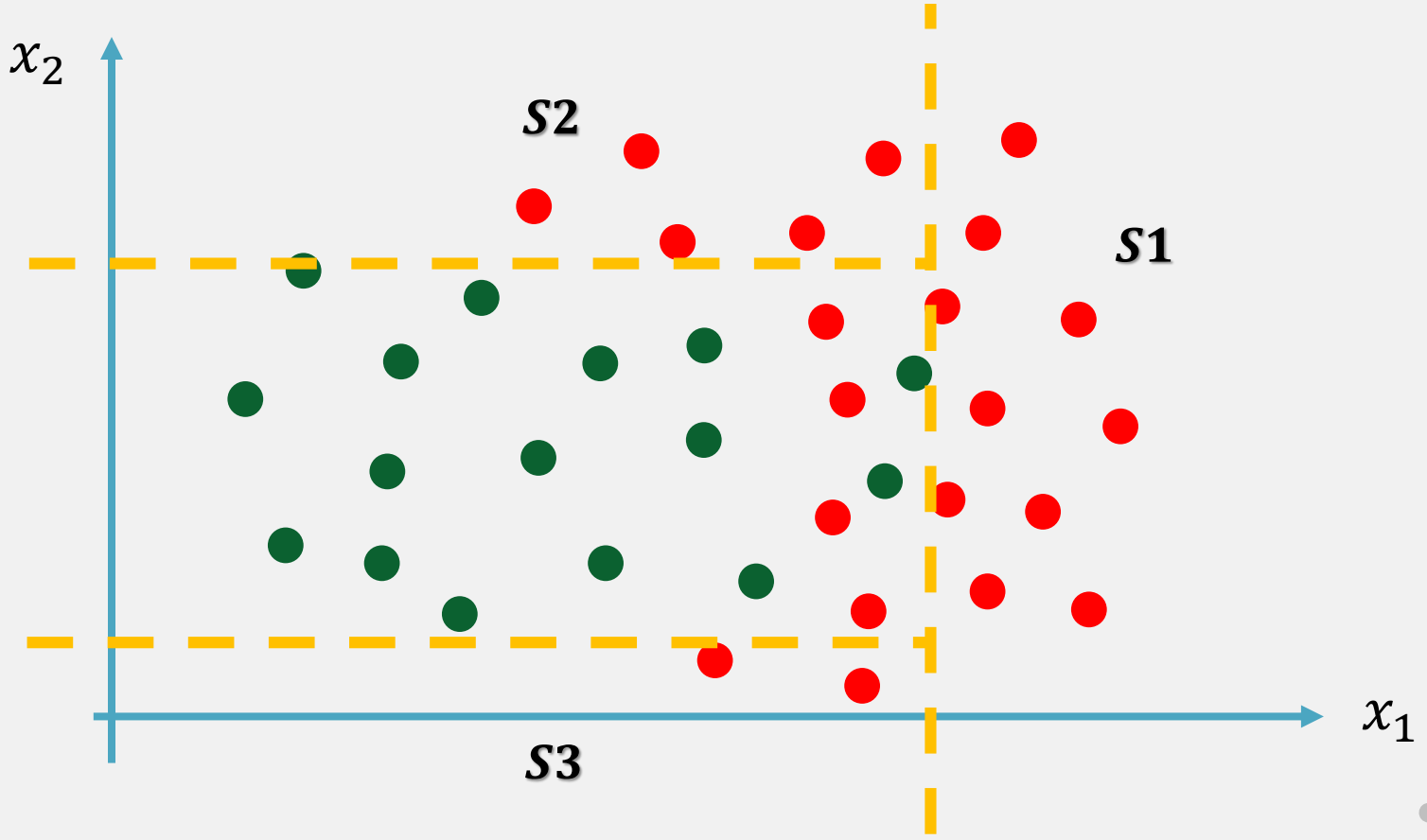
Задача классификации

Бинарное решающее дерево



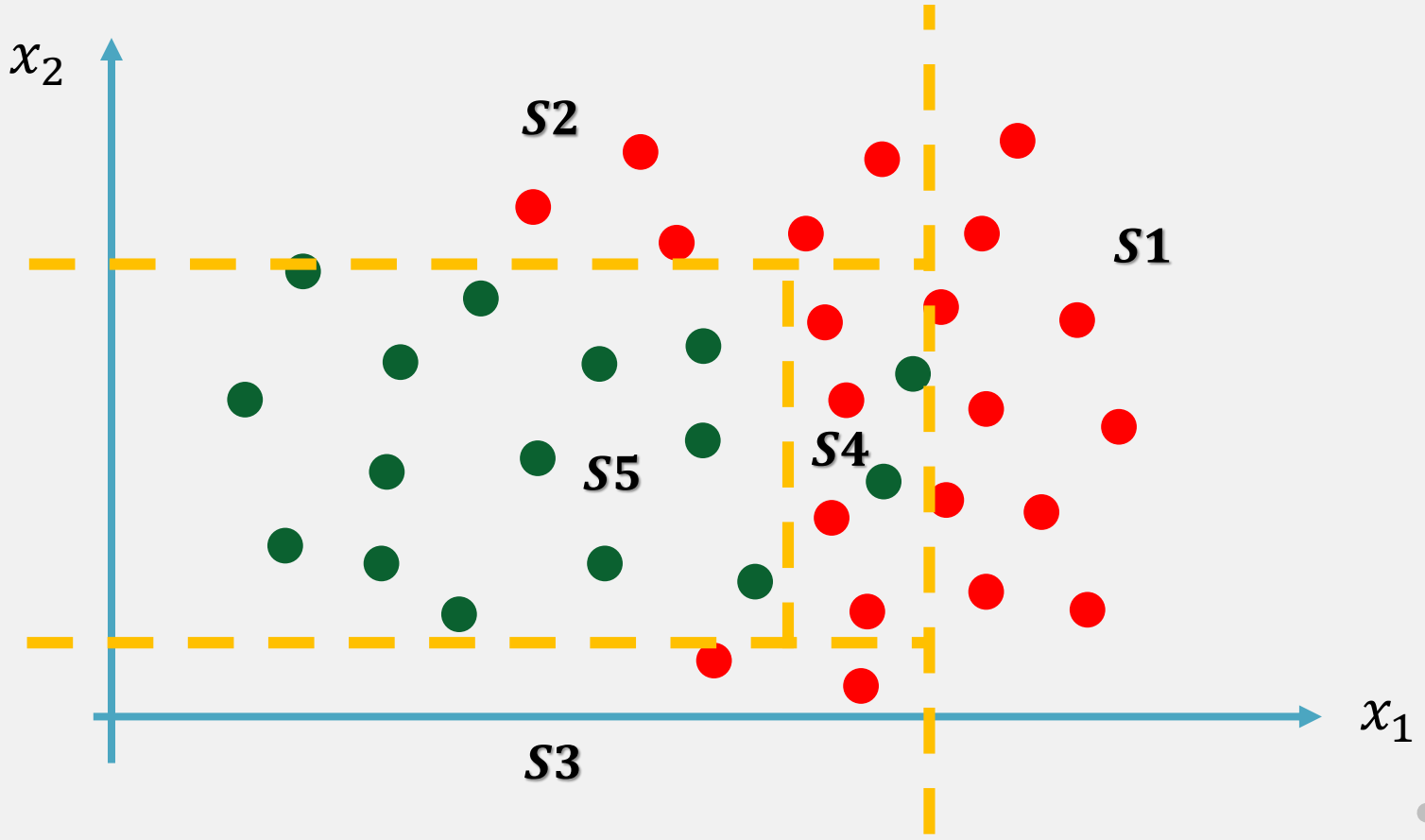
Задача классификации

Бинарное решающее дерево



Задача классификации

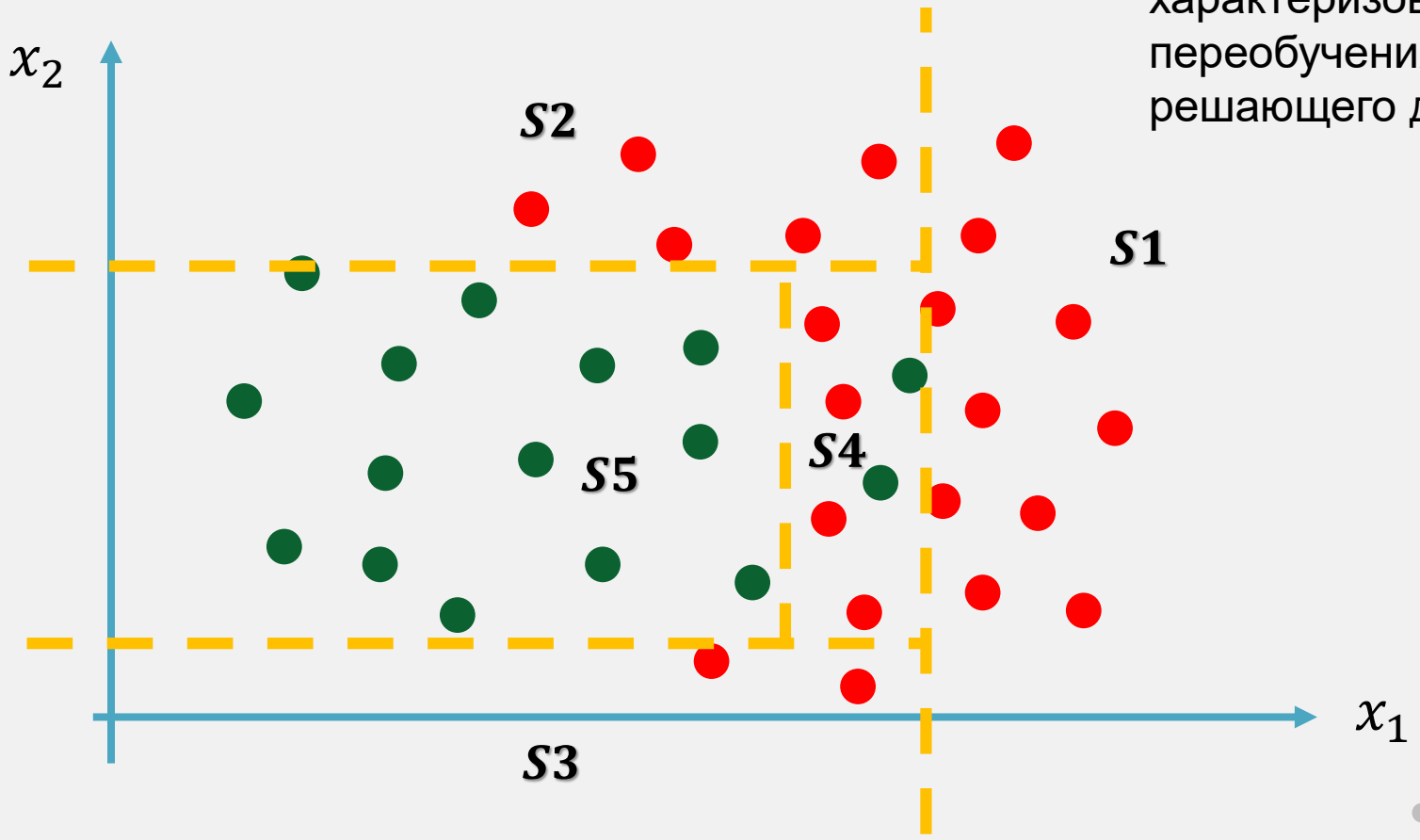
Бинарное решающее дерево



Задача классификации

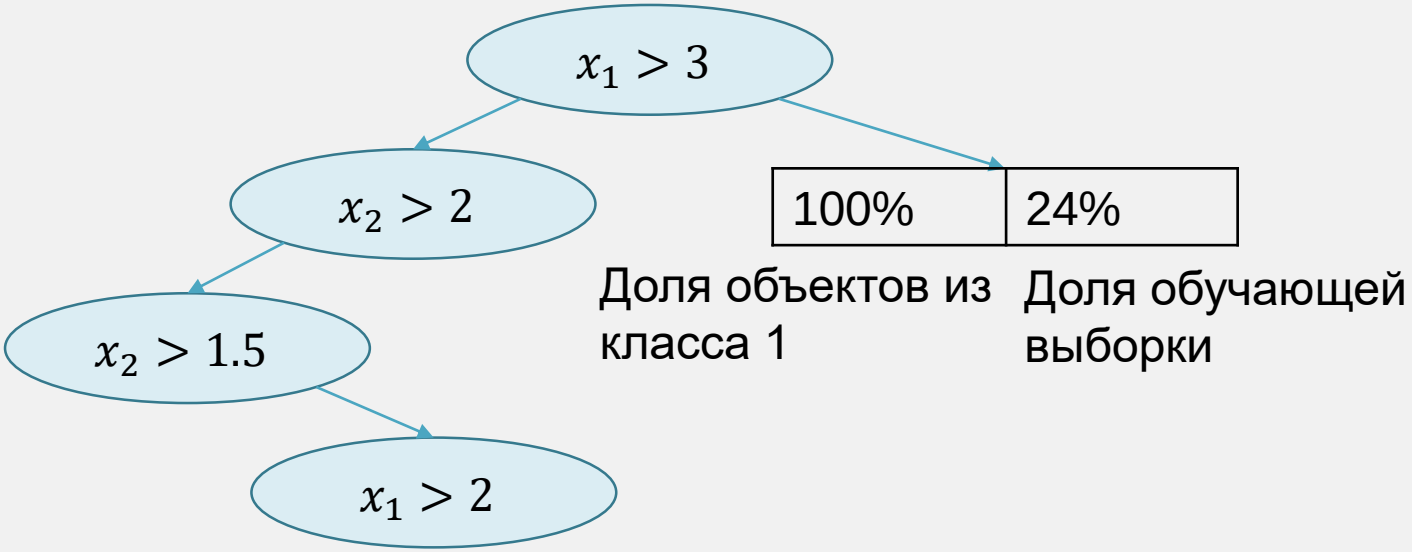
Бинарное решающее дерево

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

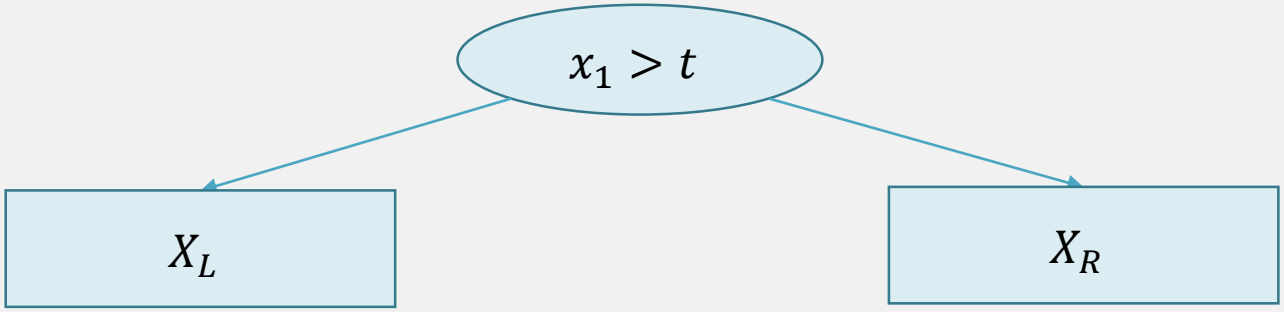
3) Gini criteria: $H(X) = 2p_0 p_1$

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

Задача классификации

Бинарное решающее дерево

Критерий разбиения



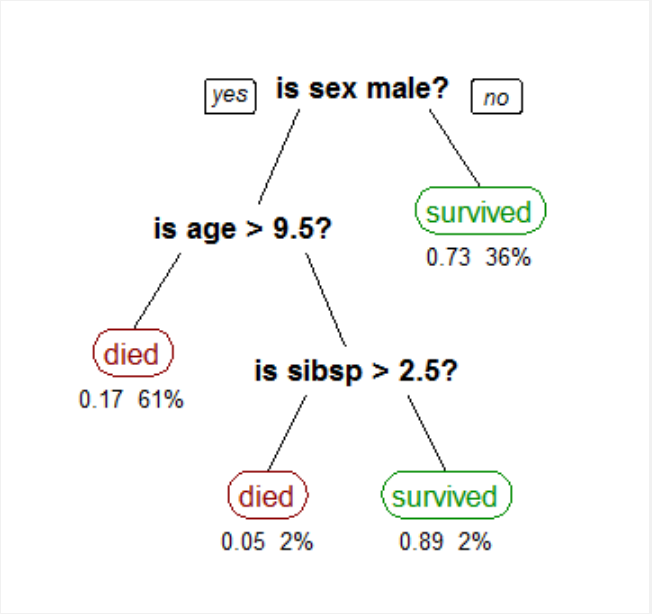
Уменьшения неопределённости (неоднородности) в узле:

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

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:

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

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

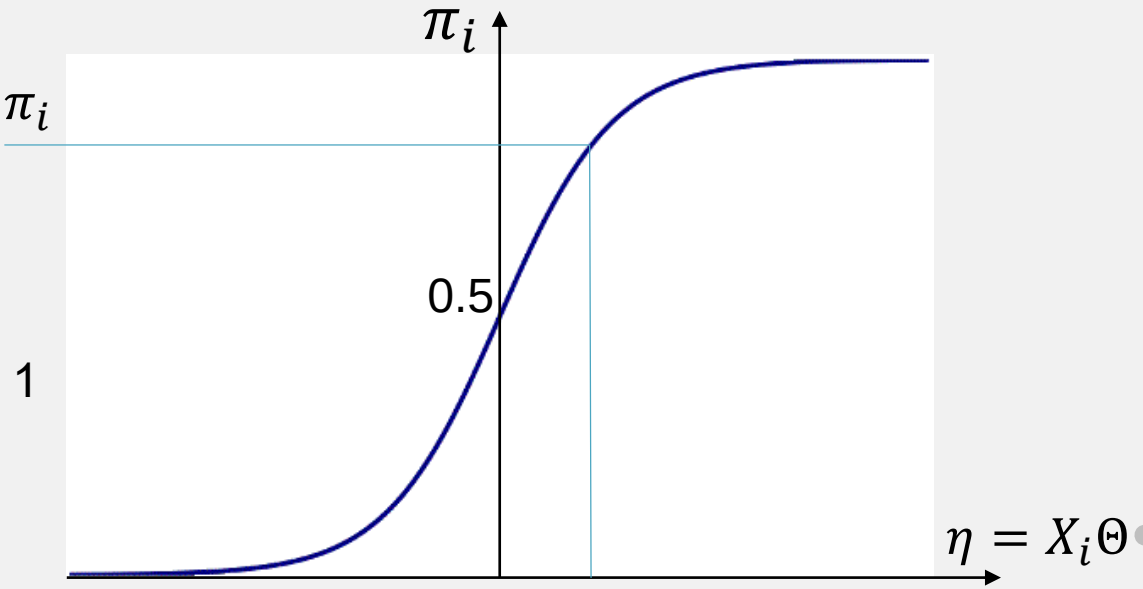
Рассматривается задача бинарной классификации (0 или 1)

Функция сигмоида

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

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

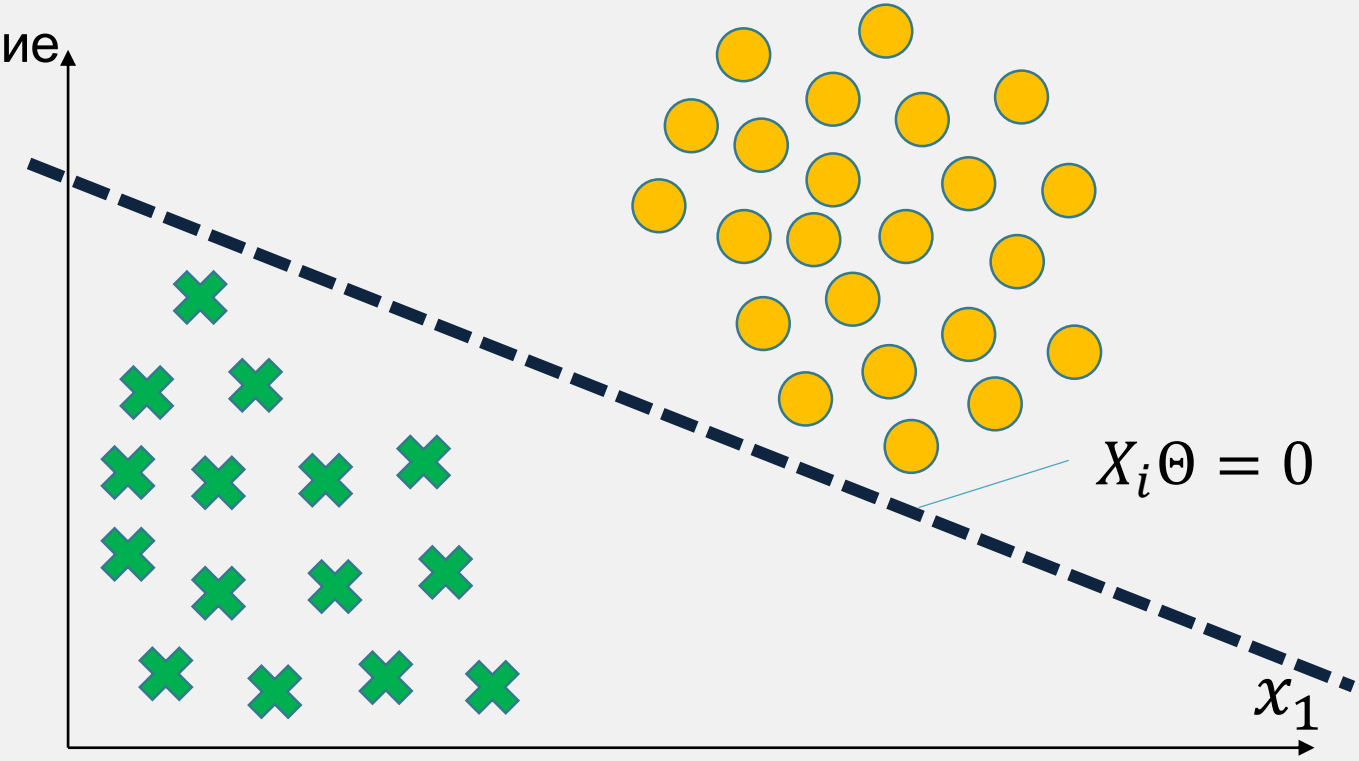
Определяется вероятность принадлежности объекта X_i классу 1



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

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

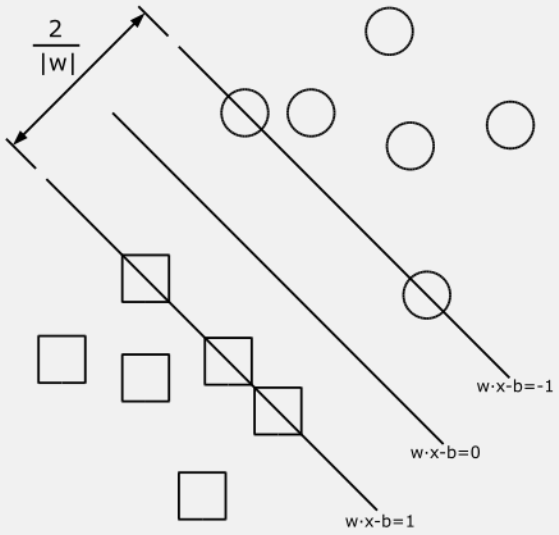
Задача классификации

Метод опорных векторов

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

Идея метода:

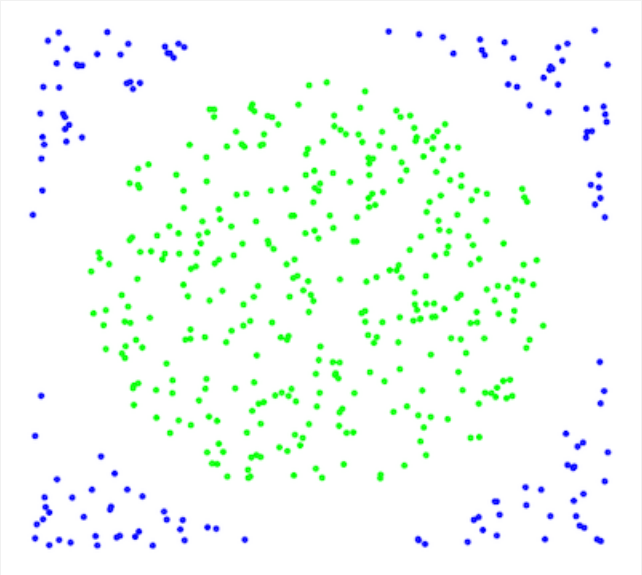
Поиск таких параметров Θ и b, которые максимизируют расстояние до каждого класса.



Задача классификации

Метод опорных векторов

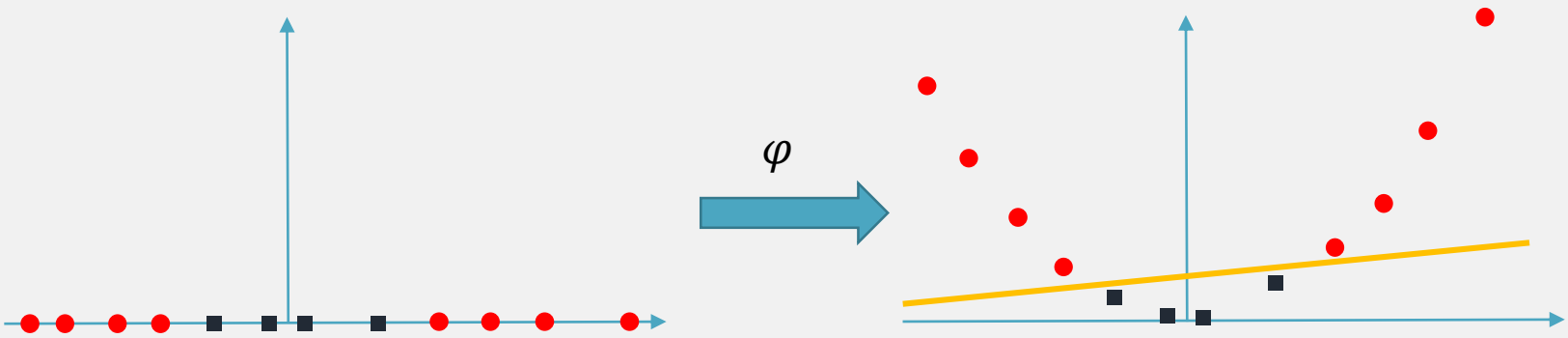
Чаще всего достаточно сложно разделить объекты обучающей выборки гиперплоскостью



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

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]))
```

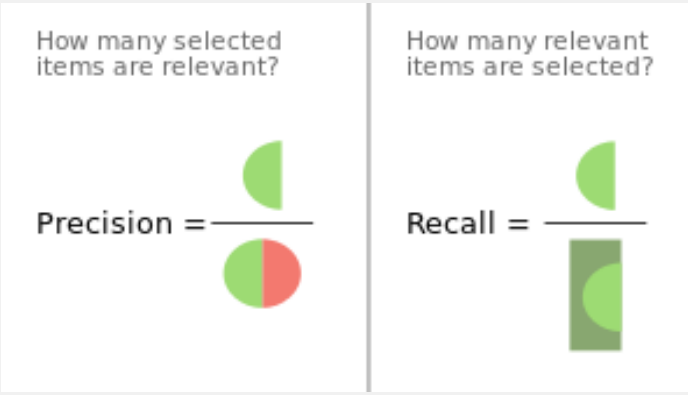
Результат:
7
0.875

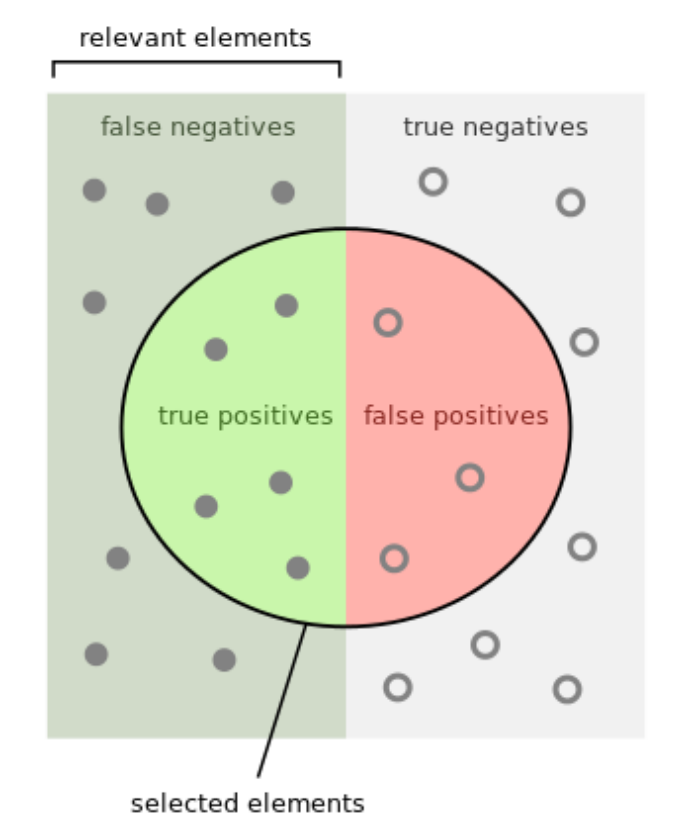
Метрики

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

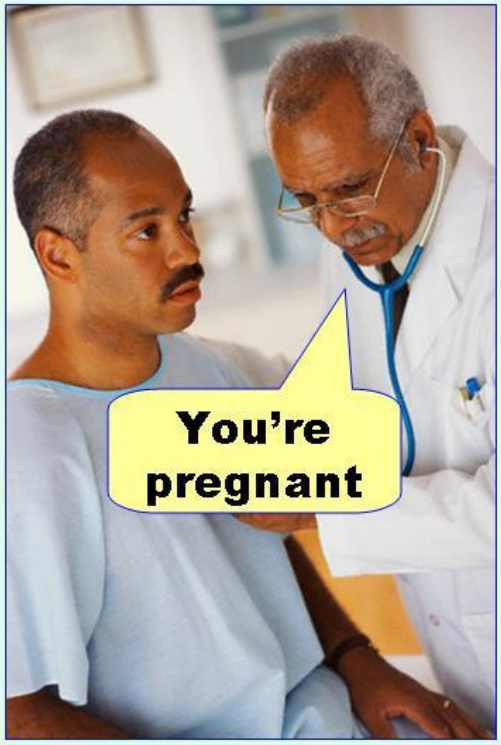
Recall: Какую часть из объектов класса 1 мы нашли?

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

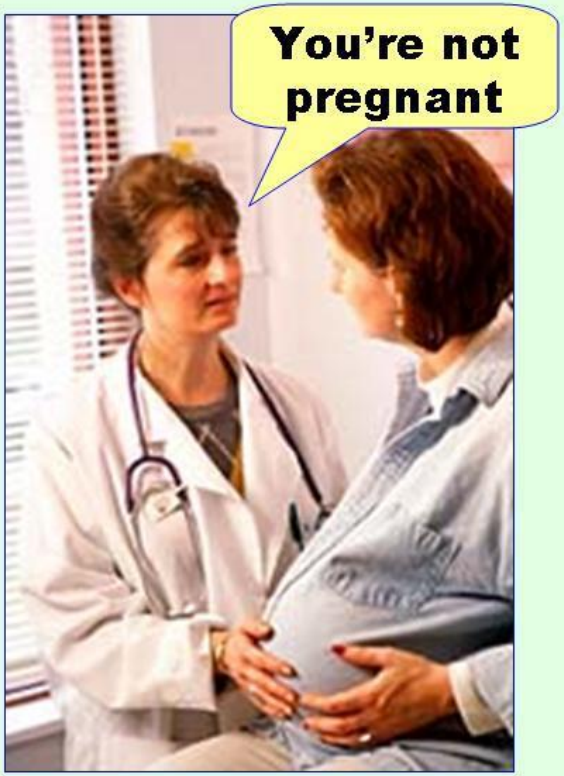
Precision: Какая часть из тех объектов класса 1, которую мы нашли, действительно принадлежат этому классу?

Метрики

Type I error
(false positive)



Type II error
(false negative)



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:

1.0	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/

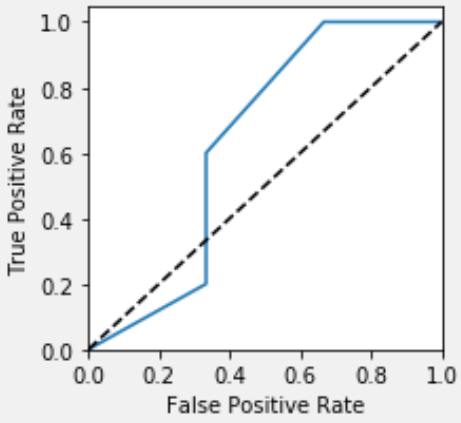
Метрики

Классификация

$ROC - AUC$

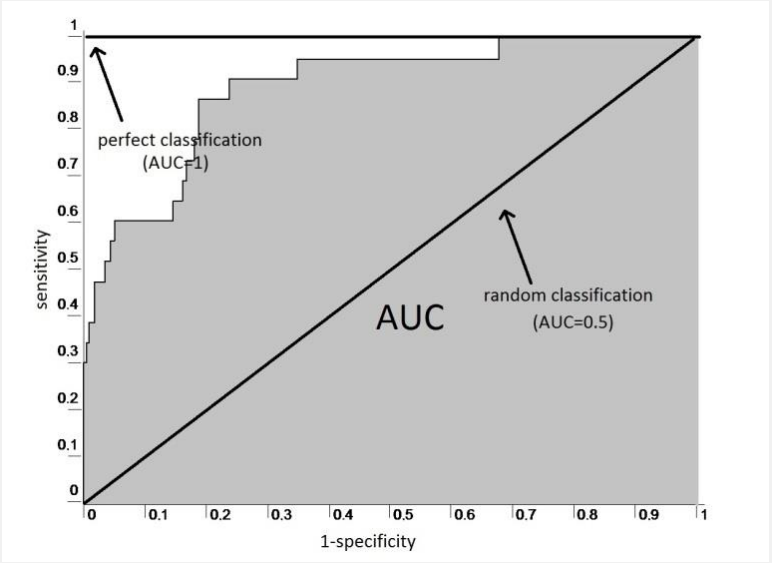
1. target = [0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0]

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

Кросс-валидация

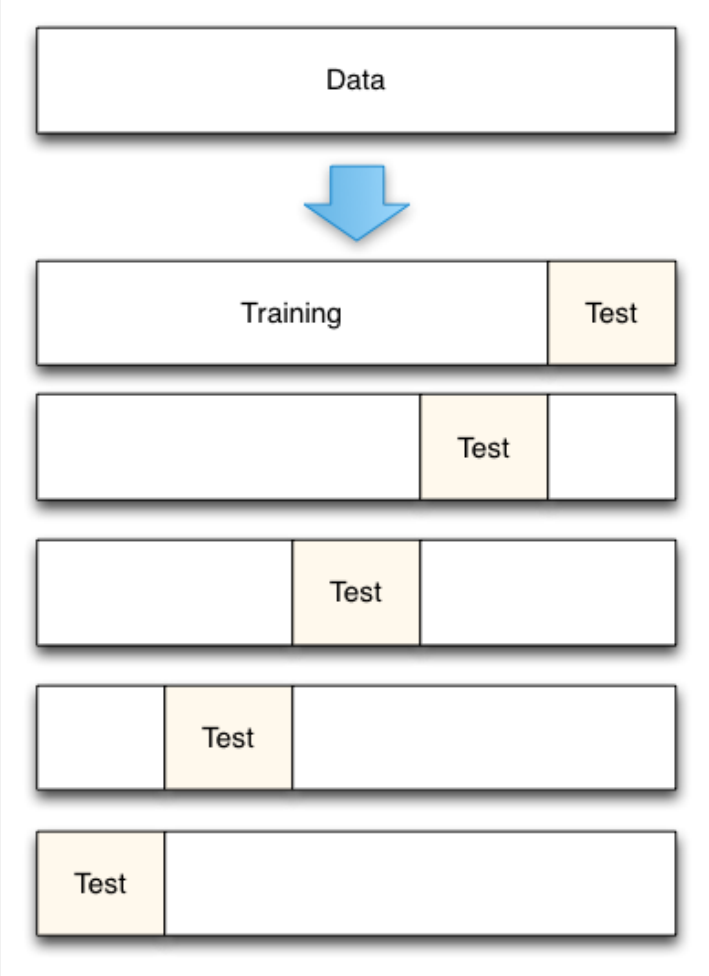
Процедура эмпирического оценивания обобщающей способности алгоритмов, обучаемых по прецедентам.

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.

Кросс-валидация



ТЕХНОСФЕРА

Спасибо за внимание!

Спасёнов Алексей

a.spasenov@corp.mail.ru