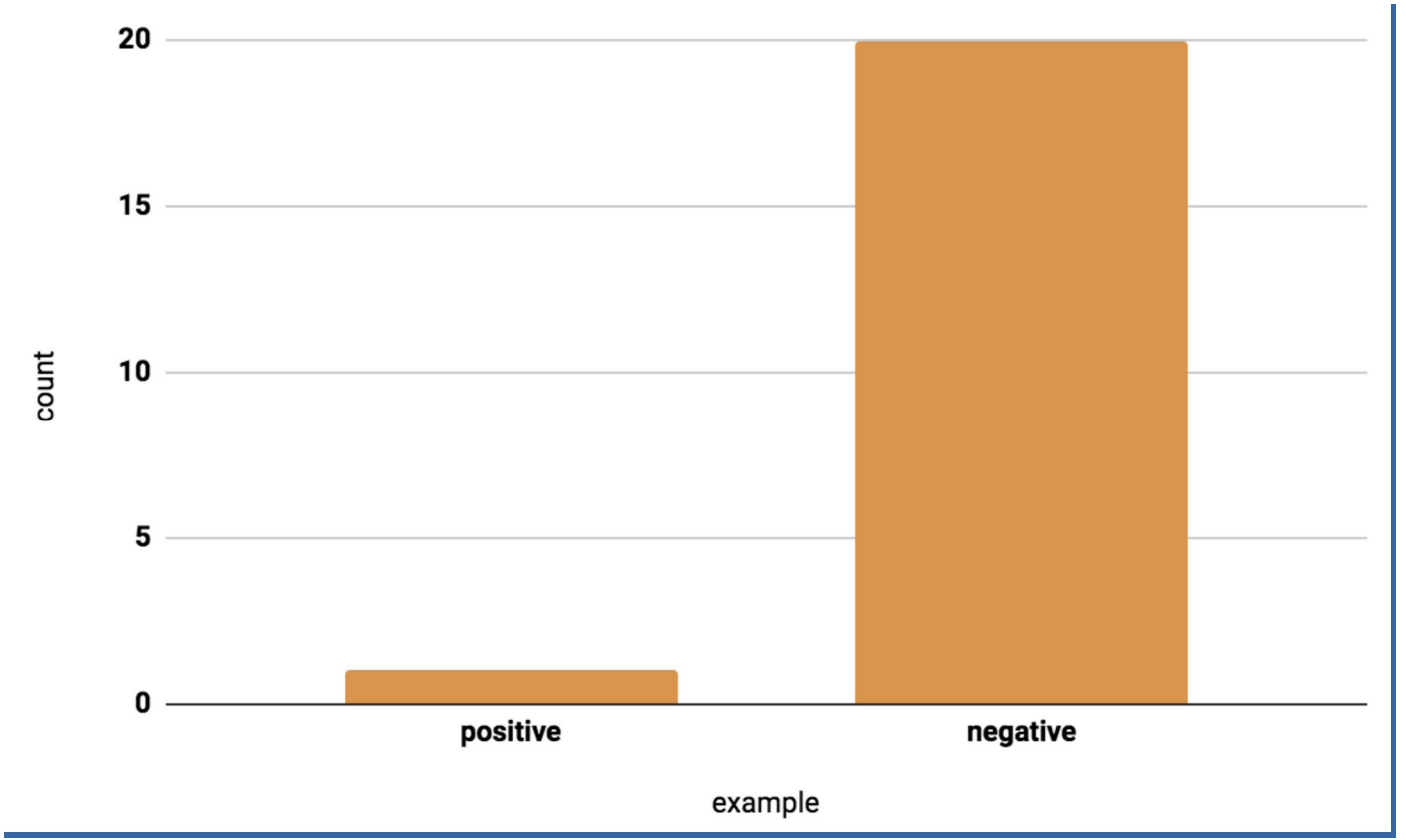
MACHINE LEARNING AND AI

ЛЕКЦИЯ 4

ПЛАН

- Несбалансированная классификация
- Precision, Recall, ROC AUC
- Collaborating Filtering (Совменстная фильтрация)
- Ансамблиевые методы
- Random Forest
- Gradient Boosting

НЕСБАЛАНСИРОВАННЫЕ КЛАССЫ



Классификация для диагностики рака

- Тренируем логистическую регрессию (*y=1* if cancer else *y=0*)
- Находим, что мы имеем 1% погрешности на тестовой выборке (99% верных диагн озов)

- Только 0.5% пациэнтов имеют рак

```
Function y = predictCancer(x)
        y = 0; # ignore x
return
```

Precision/Recall

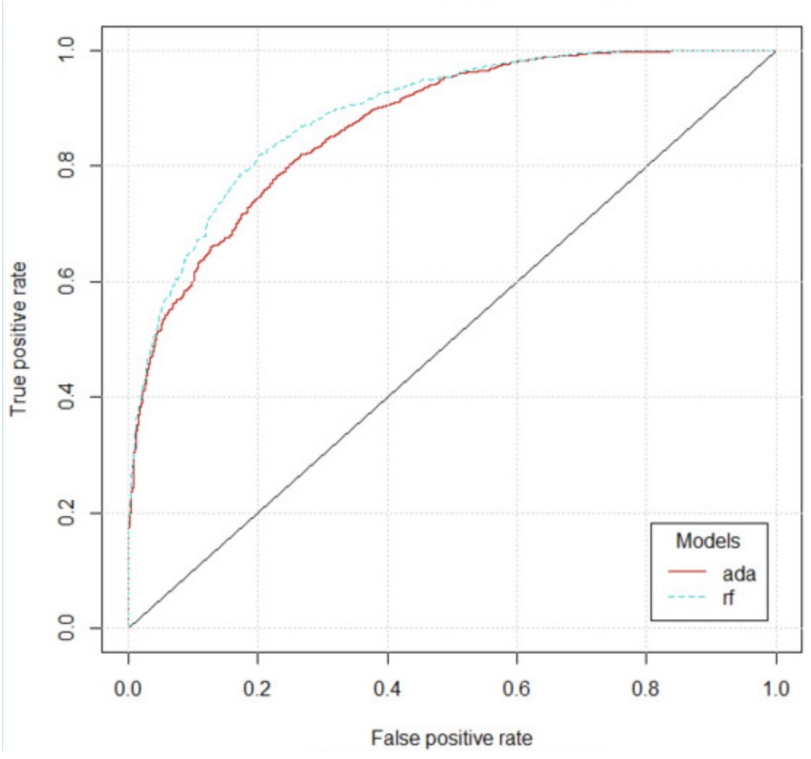
- Precision (из всех пациентов, где мы предсказали y = 1, какая доля действительно имеет рак?)
- Recall (из всех пациентов, в действительности имеющих рак, сколько процентов из них мы определили корректно?)

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

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

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{\text{tp}}{\text{tp} + \frac{1}{2}(\text{fp} + \text{fn})}$$

Receiver operating characteristic (ROC)



$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = 1 - \mathrm{FNR}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{N}} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} = 1 - \mathrm{TNR}$$

РЕКОМЕНДАТЕЛЬНЫЕ СИСТЕМЫ

Фильм	Вася	Петя	Маша	Ира
Love at last	5	5	0	0
Romance forever	5			0
Cute puppies of love		4	0	
Nonstop car chases	0	0	5	4
Swords vs. karate	0	0	5	

- n_u – no. users
- n_m = no. movies
- *r(i,j)* = 1 if user *j* has rated movie *i*
- *y(i,j)* = rating given by user *j* to movie *i*

Collaborating Filtering

Фильм	Вася	Петя	Маша	Ира	x1 (romance)	x_2 (action)
Love at last	5	5	0	0	0.9	0.01
Romance forever	5	?	?	0	?	?
Cute puppies of love	?	4	0	?	?	?
Nonstop car chases	0	0	5	4	?	?
Swords vs. karate	0	0	5	?	0.2	0.85

Оптимизационный алгоритм

- Дано $a^{(1)}, \ldots, a^{(nu)}$, обучить $x^{(1)},\ldots,x^{(nm)}$

$$min \frac{1}{2}\sum_{i=1}^{n_m}\sum_{j:r(i,j)=1}\left((a^{(j)})^T x^{(i)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2}\sum_{i=1}^{n_m}\sum_{k=1}^{n}\left(x_k^{(i)}\right)^2$$

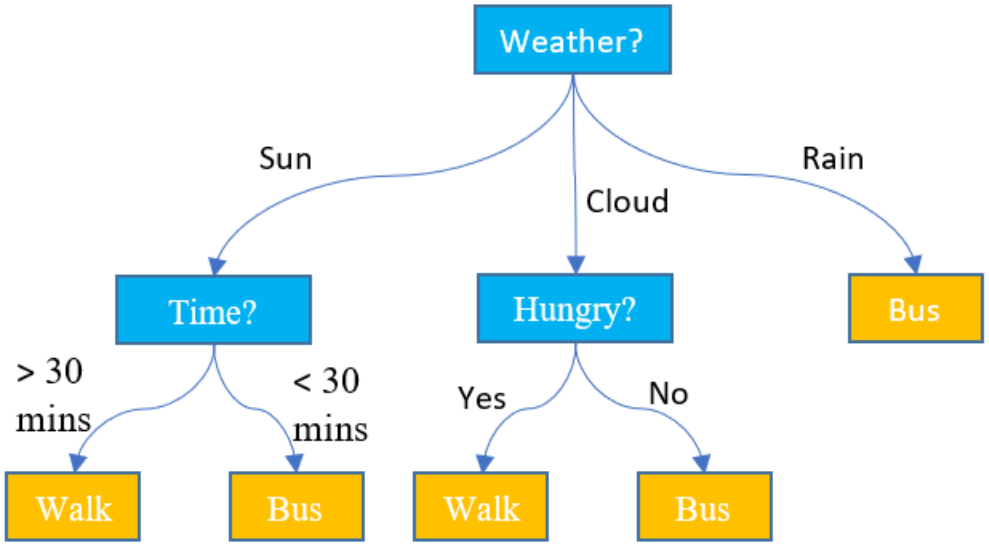
- Дано $x^{(1)}, \ldots, x^{(nm)}$, обучить $a^{(1)},\ldots,a^{(nu)}$

$$min \frac{1}{2}\sum_{i=1}^{n_m}\sum_{j:r(i,j)=1}\left((a^{(j)})^T x^{(i)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}\left(a_k^{(j)}\right)^2$$

- Всё вместе

$$J\left(x^{(1)},\ldots,x^{(n_m)},a^{(1)},\ldots,a^{(n_u)}\right) = \frac{1}{2}\sum_{i=1}^{n_m}\sum_{j:r(i,j)=1}\left((a^{(j)})^T x^{(i)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}\left(a_k^{(j)}\right)^2 + \frac{\lambda}{2}\sum_{i=1}^{n_m}\sum_{k=1}^{n}\left(x_k^{(i)}\right)^2$$

Дерево решений (Decision tree)



Пример задачи

Таблица: Как играет «Зенит».

Соперник	Играем	Лидеры	Дождь	Победа
Выше	Дома	На месте	Да	Нет
Выше	Дома	На месте	Нет	Да
Выше	Дома	Пропускают	Нет	Да
Ниже	Дома	Пропускают	Нет	Да
Ниже	В гостях	Пропускают	Нет	Нет
Ниже	Дома	Пропускают	Да	Да
Выше	В гостях	На месте	Да	Нет
Ниже	В гостях	На месте	Нет	*???*

Энтропия

Определение

Предположим, что имеется множество A из n элементов, m из которых обладают некоторым свойством S. Тогда энтропия множества A по отношению к свойству S — это

$$H(A,S) = -\frac{m}{n}\log_2\frac{m}{n} - \frac{n-m}{n}\log_2\frac{n-m}{n}.$$

Энтропия зависит от пропорции, в которой разделяется множество. Чем «ровнее» поделили, тем больше энтропия.

Энтропия

Если свойство S не бинарное, а может принимать s различных значений, каждое из которых реализуется в m_i случаях, то

$$H(A, S) = -\sum_{i=1}^{s} \frac{m_i}{n} \log \frac{m_i}{n}.$$

Энтропия — это среднее количество битов, которые требуются, чтобы закодировать атрибут S у элемента множества A. Если вероятность появления S равна $1/2$, то энтропия равна 1, и нужен полноценный бит; а если S появляется не равновероятно, то можно закодировать последовательность элементов A более эффективно.

Энтропия

В нашем примере из 7 матчей «Зенит» три проиграл и четыре выиграл. Поэтому исходная энтропия

$$H(A, \text{Победа}) = -\frac{4}{7}\log_2\frac{4}{7} - \frac{3}{7}\log_2\frac{3}{7} \approx 0.9852.$$

Прирост информации

Атрибут для классификации нужно выбирать так, чтобы после классификации энтропия (относительно целевой функции) стала как можно меньше.

Определение

Предположим, что множество A элементов, характеризующихся свойством S, классифицировано посредством атрибута Q, имеющего q возможных значений. Тогда прирост информации (information gain) определяется как

$$\mathrm{Gain}(A, Q) = H(A, S) - \sum_{i=1}^{q} \frac{|A_i|}{|A|} H(A_i, S),$$

где A_i — множество элементов A, на которых атрибут Q имеет значение i.

Прирост информации

Теперь вычислим приросты информации для различных атрибутов:

$$\begin{aligned}\text{Gain}(A, \text{Соперник}) = H(A, \text{Победа}) - \frac{4}{7} H(A_{\text{выше}}, \text{Победа}) - \\ - \frac{3}{7} H(A_{\text{ниже}}, \text{Победа}) \approx \\ \approx 0.9852 - \frac{4}{7}\left(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}\right) - \\ - \frac{3}{7}\left(-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}\right) \approx 0.0202.\end{aligned}$$

Прирост информации

$$\text{Gain}(A, \text{Играем}) \approx 0.4696.$$

$$\text{Gain}(A, \text{Лидеры}) \approx 0.1281.$$

$$\text{Gain}(A, \text{Дождь}) \approx 0.1281.$$

Прирост информации советует сначала классифицировать по тому, домашний ли матч или гостевой.

Индекс Джини

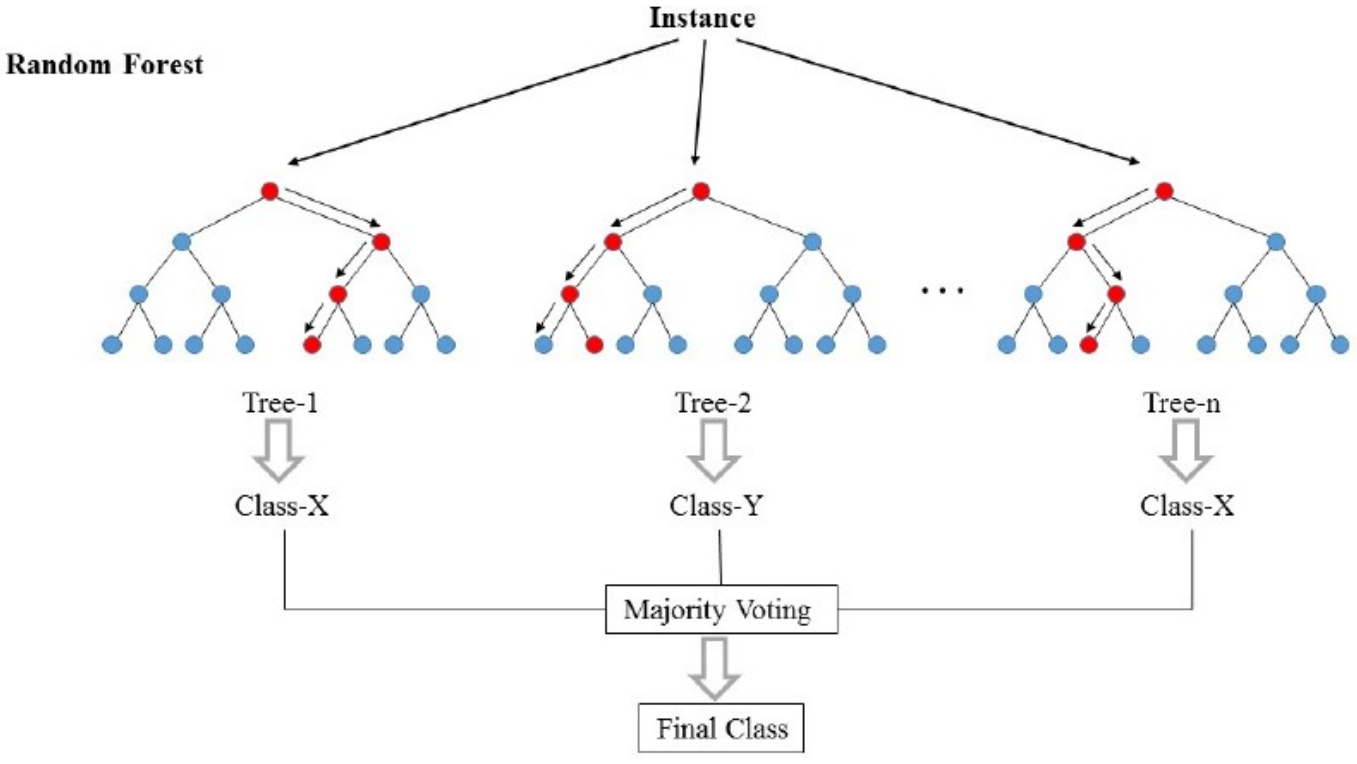
Для набора тестов A и свойства S, имеющего s значений, этот индекс вычисляется как

$$\mathrm{Gini}(A, S) = 1 - \sum_{i=1}^{s} \left(\frac{|A_i|}{|A|} \right)^2.$$

Соответственно, для набора тестов A, атрибута Q, имеющего q значений, и целевого свойства S, имеющего s значений, индекс вычисляется следующим образом:

$$\mathrm{Gini}(A, Q, S) = \mathrm{Gini}(A, S) - \sum_{j=1}^{q} \frac{|A_j|}{|A|} \mathrm{Gini}(A_j, S).$$

Алгоритм Random Forest



Алгоритм Gradient Boosting