

Data Mining in Action

Лекция 4

Признаки и метрики

ML: ожидание

```
clf = XGBClassifier()  
clf.fit(X_train, y_train)  
labels = clf.predict(X_test)
```

ML: реальность

Выгрузка таблиц

Соединение по ключам

Расчёт агрегатов

Заполнение пропусков

```
clf = XGBClassifier()  
clf.fit(X_train, y_train)  
labels = clf.predict(X_test)
```

Калибровка

Усреднение моделей

Расчёт целевой функции

Оптимизация

- Мы подали в модель всю релевантную информацию?
- Не подали ли мы туда ничего лишнего?
- Сможет ли модель корректно обработать признаки?

- Что должна выдавать модель?
- Насколько отличается её прогноз от желаемого?
- За счёт чего прогноз можно было бы улучшить?

На этой лекции

I. Работа с признаками

1. Предобработка данных
2. Категориальные признаки
3. Разреженные признаки

II. Метрики качества моделей

1. Применение алгоритмов и целевые функции
2. Метрики для регрессии
3. Метрики для классификации

Признаки

Почему работа с признаками важна?

- Хорошие алгоритмы для обучения у вас и так уже есть
- Garbage in – garbage out
- Признаки – это тоже модели

Одномерные преобразования

- Масштабирование признаков

- Перевод в $[0,1]$

$$\tilde{x} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

- Стандартизация

$$\tilde{x} = \frac{x - \bar{x}}{\sigma_x}$$

- Робастная стандартизация

$$\tilde{x} = \frac{x - q_\alpha(x)}{q_{1-\alpha}(x) - q_\alpha(x)}$$

Регуляризация
Меры близости
Начальные веса
Интерпретация

- Обработка выбросов

- Клипование

$$\tilde{x} = \min(100500, \max(x, 0))$$

- Нормализация

- Логарифмирование

$$\tilde{x} = \log(1 + x)$$


- Квантильное отображение

$$\tilde{x} = \text{rank}(x)$$

Может сделать
распределение
более симметричным

Заполнение пропусков (imputation)

- Откладывание
 - Заполняем необычным значением (-999999)
 - Заполняем нулём, оставляем бинарную метку
- Одномерное заполнение
 - Среднее, медиана, мода
 - Предыдущее значение ряда
- Моделирование
 - Заполнить по ближайшим соседям
 - Предсказать какой-нибудь регрессией
 - Надёргать из условного совместного распределения



Отсутствие информации –
это тоже информация!

Feature engineering

- Многомерные преобразования (любые функции N переменных)
 - Отношения, min/max
- «Признаки признаков»
 - Part-of-speech для слов
 - Классификация высокого уровня (большой город/маленький город/село)
- Учёт динамики
 - Лаги
 - Скользящие средние
- Играйтесь!

Иерархические признаки

- Данные бывают вложенными
- Агрегация:
 - Количество, сумма
 - Среднее и т.п.
 - Минимум, максимум
- Сложные признаки
 - Тренды
 - Подмодели по одному объекту
 - Эмбединги нейросетями

```
credit_history = [  
    loan1: {name,  
            kind,  
            amount,  
            interest,  
            status,  
            open_date, close_date  
            [pay_1, pay_2, ..., pay_n]  
    },  
    loan2: {...},  
    ...]
```

Шкалы

| | | | |
|-------------------|------------------|---|-------------------------|
| • Номинальная | $(=, \neq)$ | } | Категориальные признаки |
| • Порядковая | $(>, <)$ | | По ситуации |
| • Интервальная | $(+, -)$ | } | Числовые признаки |
| • Шкала отношений | (\times, \div) | | |

Категориальные признаки

- Кодирование номером
- Двоичное кодирование (one-hot-encoding)
- Кодирование средним значением целевой переменной
 - Чтобы не переобучиться, нужно брать среднее с другого фолда
 - Если есть ось времени, можно брать среднее из прошлого
 - Именно так работает CatBoost

Разреженные признаки

- Разреженное хранение
- Hashing trick
 - быстрое и неаккуратное сокращение размерности
 - Удивительно, на часто спасает
- Латентные признаки
 - Разложение матриц
 - Эмбединги из других моделей (например word2vec)
- Вспомогательные признаки
 - Кодирование средним всем (не только целевой переменной)

Метрики

Целевая функция

- На основе моделей принимаем решение
- Решение приносит пользу
 - Например, увеличение прибыли
- Неправильное решение приводит к потерям
- Хотим прогноз, который минимизирует потери!



Вероятностный подход

- Мы пытаемся прогнозировать Y , но он случаен
- Будем прогнозировать распределение $p(Y|X)$!
- Правдоподобие: $Likelihood = p(y_1, \dots y_n | x_1, \dots x_n)$
- Если наблюдения независимы: $Likelihood = \prod_{i=1}^n p(y_i | x_i)$
- На практике почти всегда работают с суммой логарифмов
- Правдоподобие порождает привычные функции потерь
 - Например, если $p(y|\hat{y}) \sim \mathcal{N}(0, \sigma^2)$, то логарифм правдоподобия равен

$$\log L = -n \log \sqrt{2\pi}\sigma - \sum_{i=1}^n \frac{(y - \hat{y})^2}{2\sigma^2} = a - b \times MSE$$

Принятие решений

- Могут быть важны вероятности *и* потери

Денежные результаты

| Заемщик | Хороший | Плохой |
|----------|---------|----------|
| Одобрили | 0 | -100 000 |
| Отказали | -5 000 | 0 |

Статистика по заявкам

| Заемщик | Хороший | Плохой |
|----------|---------|--------|
| Одобрили | 48% | 2% |
| Отказали | 50% | |

Алгоритм машинного обучения

Модель ML можно разложить на три компоненты

1. Функциональная форма

- Линейная, дерево, композиция...

2. Целевая функция

- Разница прогноза и факта, доля ошибок, отступ, правдоподобие...

3. Метод оптимизации

- Линейное программирование, градиентный спуск, генетические алгоритмы....

Иерархия метрик

- Бизнес-метрики (онлайн)
 - То, ради чего вы работаете (прибыль, счастье пользователя, ...)
- Прокси-метрики (онлайн, но быстрее)
 - То, что можно измерить быстро (конверсия, средний чек, ...)
- Меры качества (оффлайн, для выбора гиперпараметров)
 - Доля верных/неверных прогнозов, ...
- Функции потерь (оффлайн, для подгона основных параметров)
 - Правдоподобие, MSE, ...
 - Должны быть гладкими и быстро вычисляться

Хотелось бы жить ближе к бизнес-метрикам, но это не всегда удаётся

Офлайн-метрики для регрессии

Mean Absolute Error

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

Лучшее константное предсказание - медиана

Данные:

| X | Y |
|----|---|
| -1 | 0 |
| -1 | 1 |
| -1 | 1 |

Минимизация ошибки:

Input interpretation:

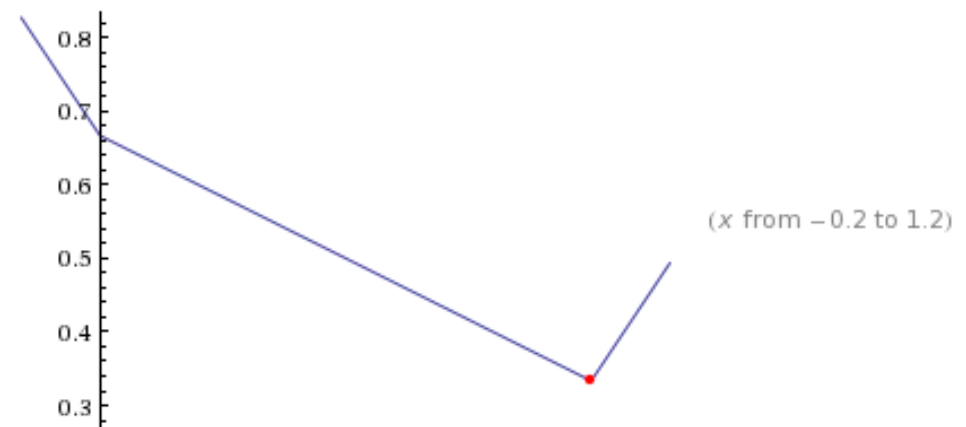
| | | |
|----------|----------|---|
| minimize | function | $\frac{1}{3} (0 - x + 1 - x + 1 - x)$ |
| | domain | $0 \leq x \leq 1$ |

[2]

Global minimum:

$$\min \left\{ \frac{1}{3} (|0 - x| + |1 - x| + |1 - x|) \mid 0 \leq x \leq 1 \right\} = \frac{1}{3} \text{ at } x = 1$$

Plot:



Root Mean Squared Error

Минимизация ошибки:

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

Лучшее константное предсказание - среднее

Данные:

| X | Y |
|----|---|
| -1 | 0 |
| -1 | 1 |
| -1 | 1 |

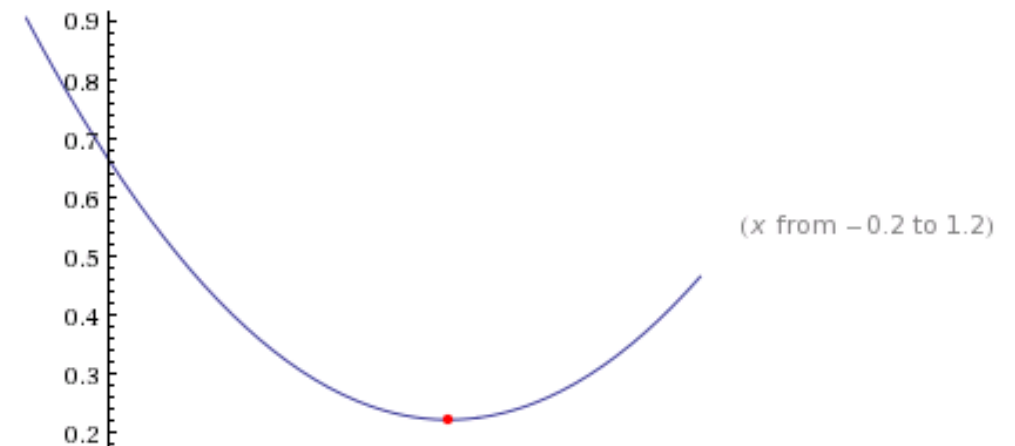
Input interpretation:

| | | |
|----------|----------|---|
| minimize | function | $\frac{1}{3} ((0 - x)^2 + (1 - x)^2 + (1 - x)^2)$ |
| | domain | $0 \leq x \leq 1$ |

Global minimum:

$$\min \left\{ \frac{1}{3} ((0 - x)^2 + (1 - x)^2 + (1 - x)^2) \mid 0 \leq x \leq 1 \right\} = \frac{2}{9} \text{ at } x = \frac{2}{3}$$

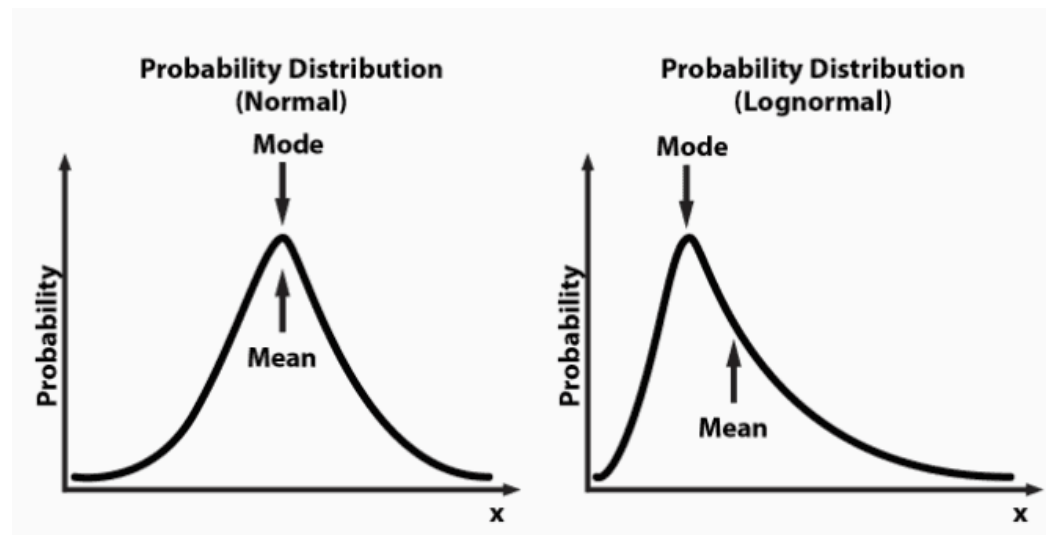
Plot:



Root Mean Squared Logarithmic Error

$$RMSLE_{\delta} = \frac{1}{n} \sum_{i=1}^n (\log(\hat{y} + \delta) - \log(y + \delta))^2$$

Лучшее константное предсказание - *среднее геометрическое* для $y + \delta$



Mean average percentage error

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

Минимизирует непонятно что

- Интуитивная
- Несимметрично штрафует положительные и отрицательные выбросы
- Может быть $[0, +\infty)$
- Очень чувствительна к маленьким y

Symmetric mean average percentage error

$$SMAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{(y_i + \hat{y}_i)/2}$$

Минимизирует непонятно что

- Более «справедливая», чем MAPE
- Может быть $[0, 200\%]$
- Тоже несимметричная

Офлайн-метрики для классификации

Accuracy

`Accuracy = np.mean(ytrue == ypred)`

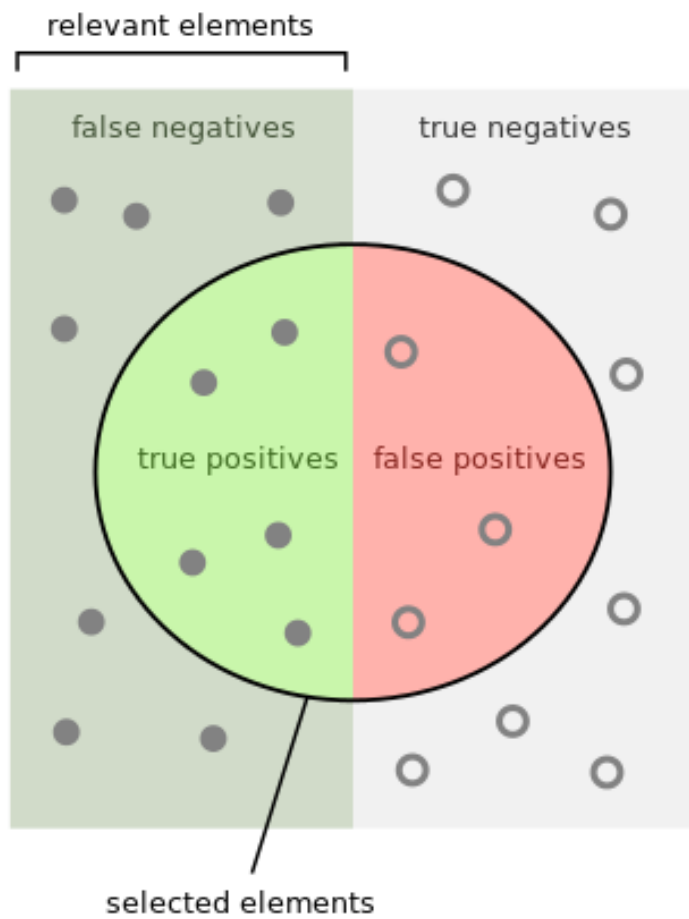
Лучшее константное решение - самый часто встречающийся класс

Пример с несбалансированными классами:

`y={0,1}; np.mean(y) == 0.99`

`Accuracy = np.mean(ytrue == 1) = 0.99`

Precision and recall

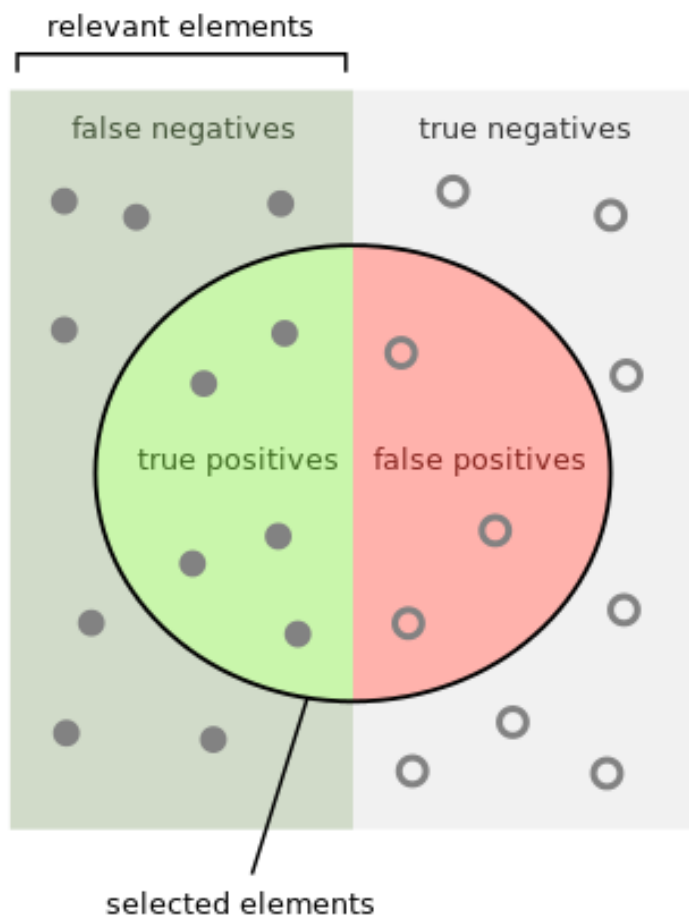


False Positive - ошибка I рода (ложное срабатывание)

False Negative - ошибка II рода (объект пропущен)



Precision and recall



How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

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

Сколько хороших клиентов среди одобренных?

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

Какой доле всех хороших мы дали кредит?

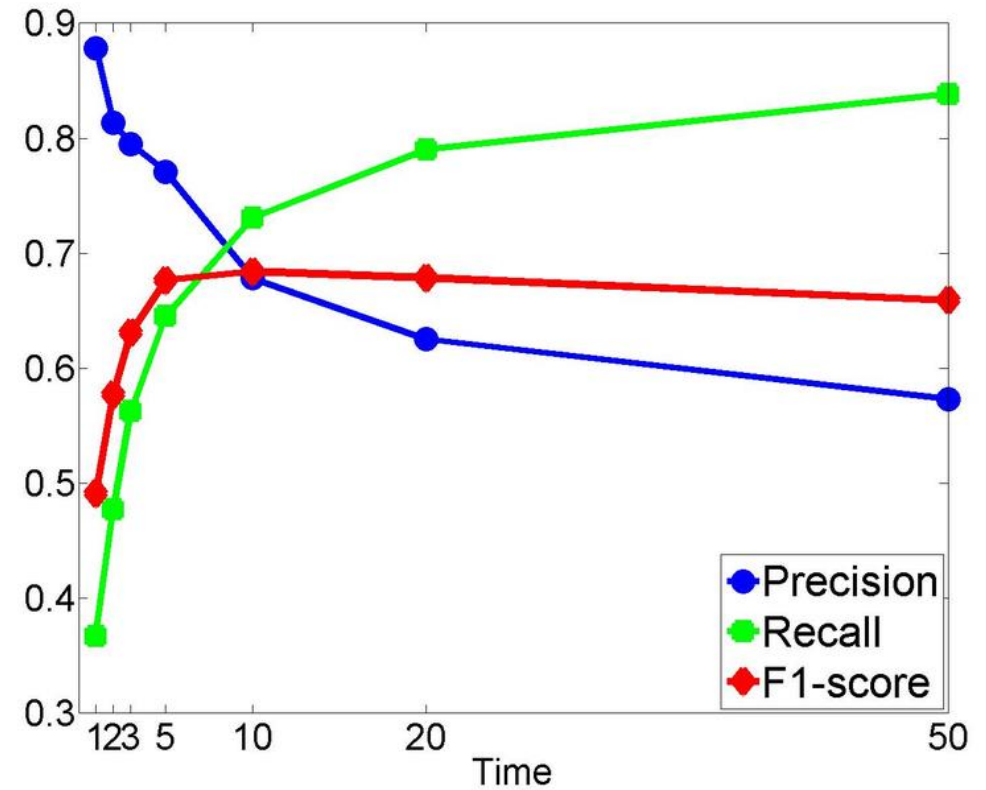
Disambiguation

- Accuracy = точность
- Recall = полнота
- Precision = **тоже** точность

Будьте бдительны!

F-score

$$F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$



F-score

$$F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

```
y_true = [[1, 2],  
          [3, 4, 5],  
          [6],  
          [7]]
```

```
y_pred = [[1, 2, 3, 9],  
          [3, 4],  
          [6, 12],  
          [1]]
```

```
mean_f1(y_true, y_pred)  
# 0.53333333
```

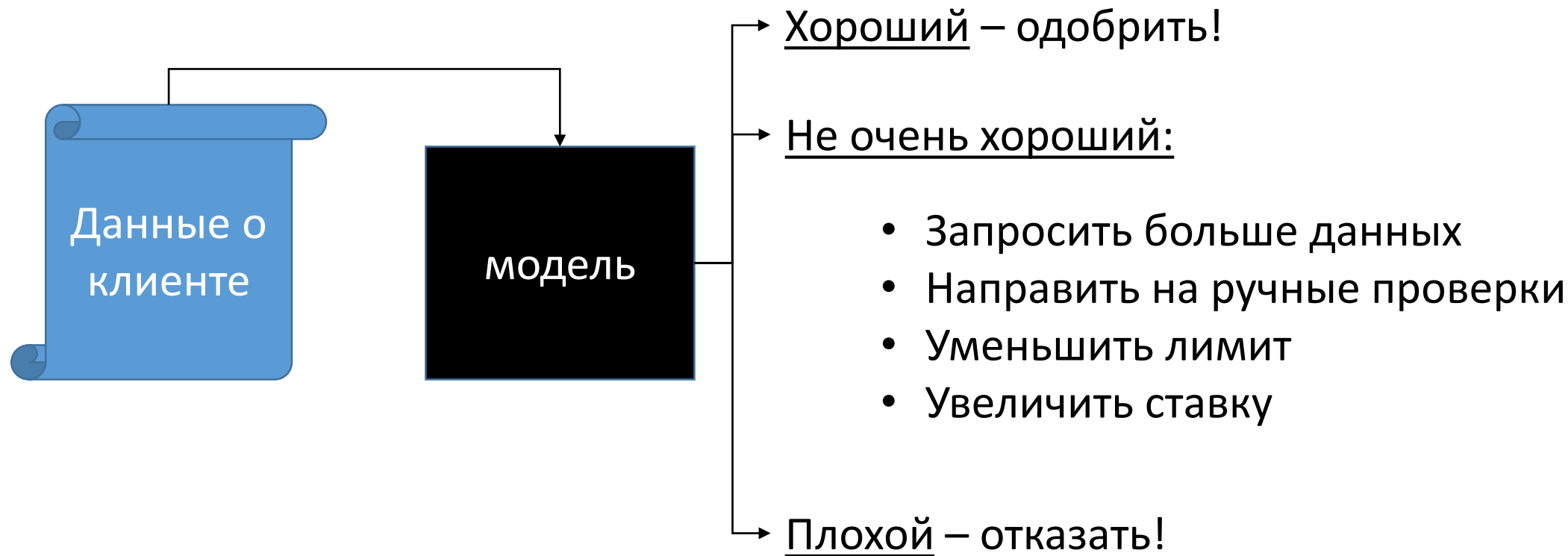

F-score

$$F_{\beta} = (1 + \beta^2) * \frac{\text{precision} * \text{recall}}{(\beta^2 * \text{precision}) + \text{recall}}$$

при $0 < \beta < 1$ предпочтение отдаётся точности
при $\beta > 1$ больший вес приобретает полнота



Кредитный скоринг: бинарной оценки мало

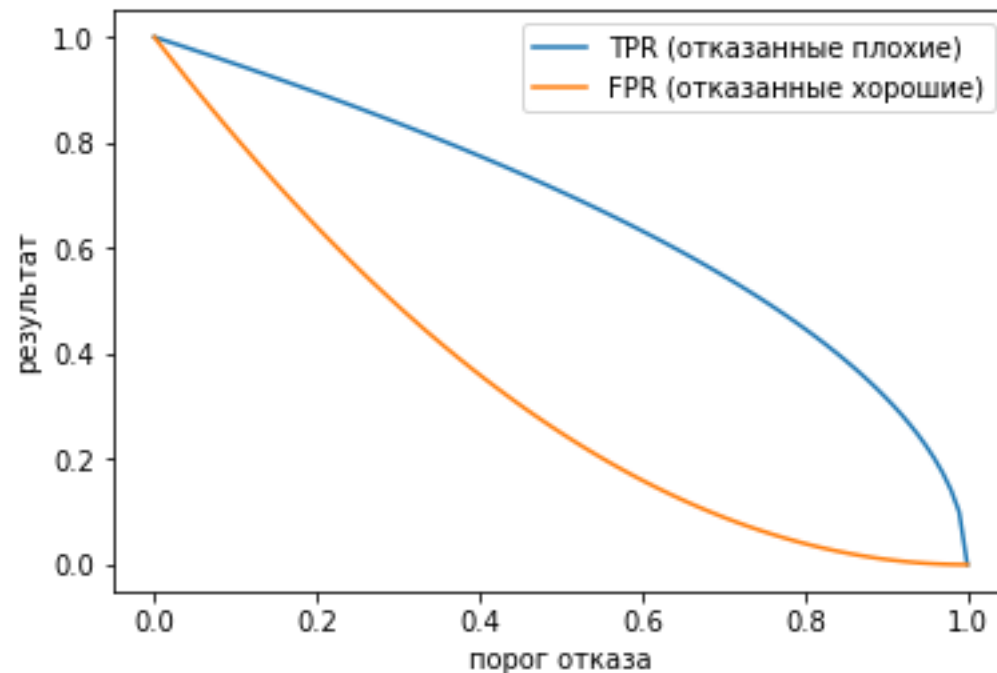


Нужно уметь *ранжировать* клиентов

Хорошее ранжирование + калибровка = точная оценка вероятности

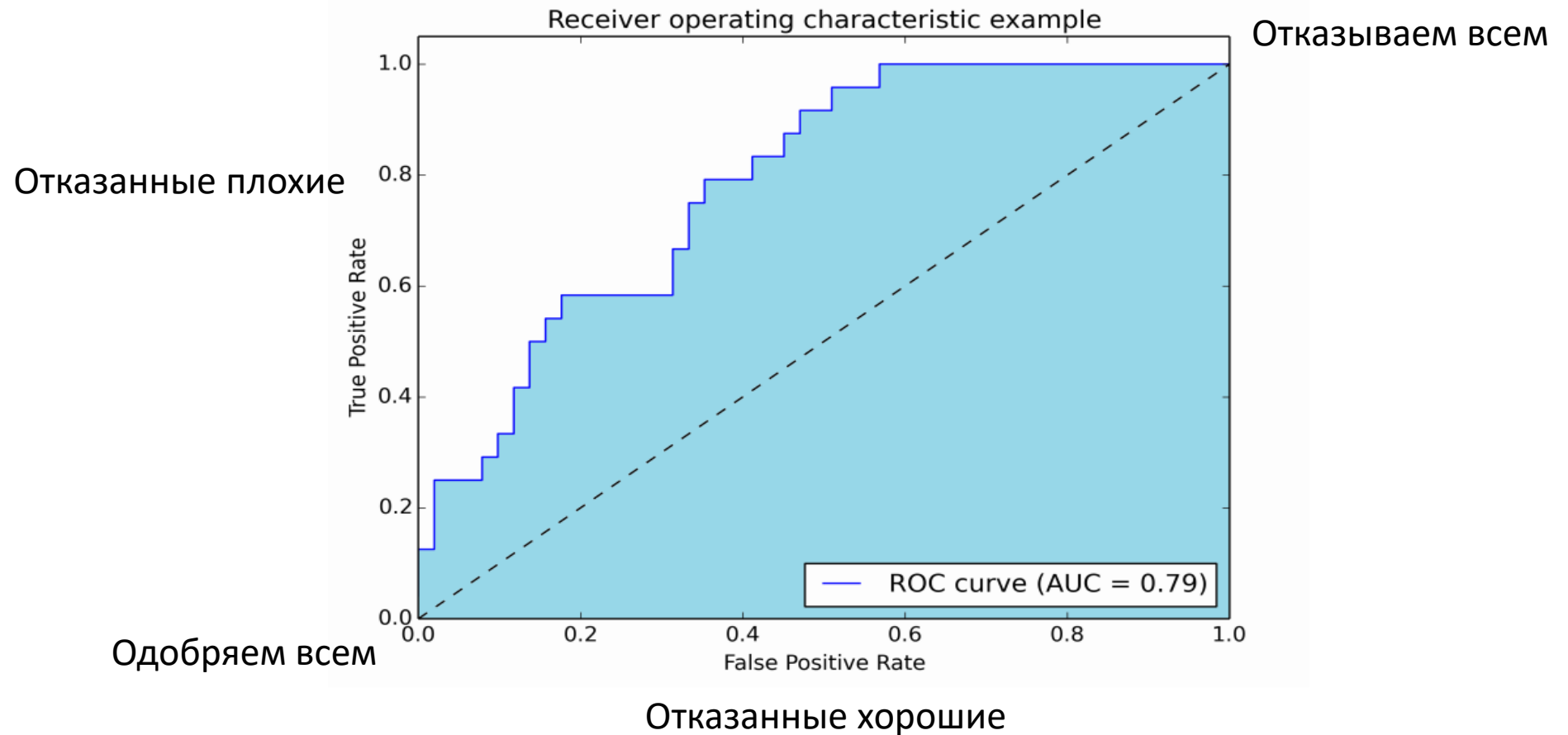
Какой порог выставить?

- \tilde{p} - вероятность дефолта
- Отказывать, если:
- $\tilde{p} > 50\%$?
- $\tilde{p} > 10\%$?
- $\tilde{p} \times \mathbb{E}(\textit{profit}|\textit{bad}) + (1 - \tilde{p})\mathbb{E}(\textit{profit}|\textit{good}) < 0$
- Как оценить качество, ещё не зная порога отказа?



ROC-кривая

Как будет меняться TPR в зависимости от FPR при смене порога?



AUC (ROC)

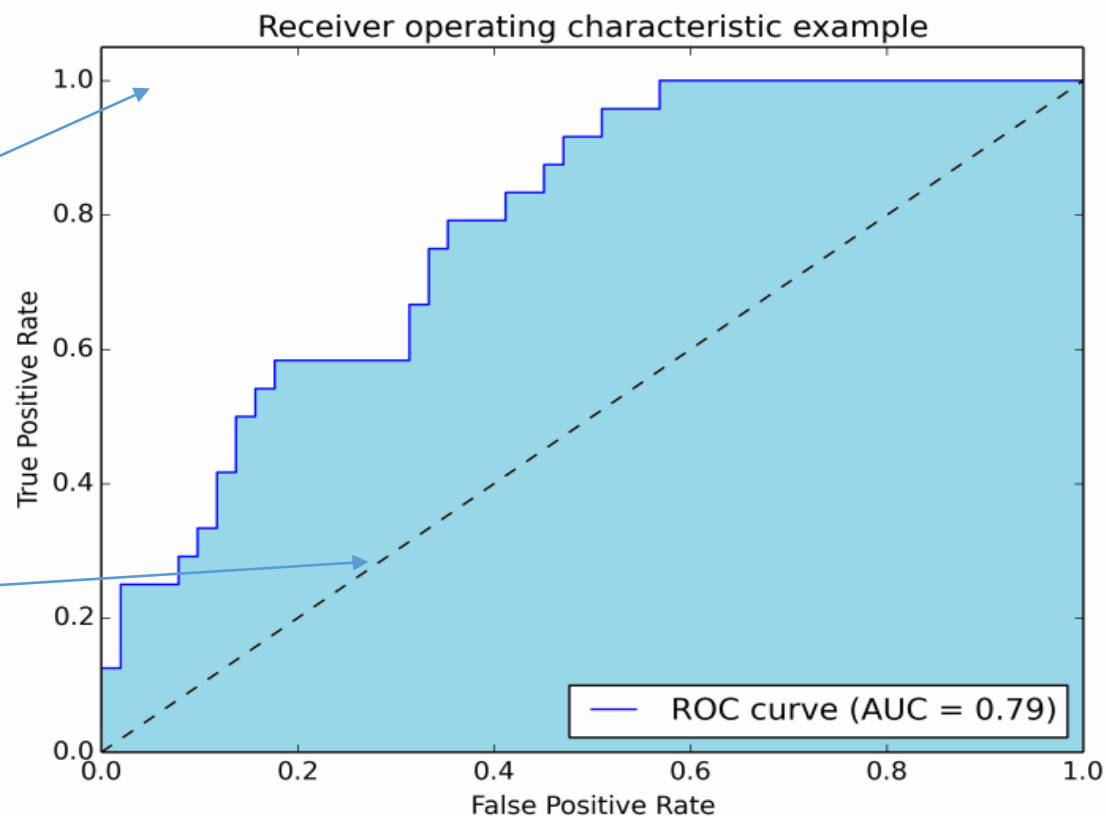
Доля правильно отранжированных пар:

$y_{pred_i} > y_{pred_j}$ если $y_{true_i} > y_{true_j}$

Или площадь под кривой:

Идеальный алгоритм

Бесполезный алгоритм



AUC (ROC)

Доля правильно отранжированных пар:

$y_{\text{pred}_i} > y_{\text{pred}_j}$ IF $y_{\text{true}_i} > y_{\text{true}_j}$

То же самое:

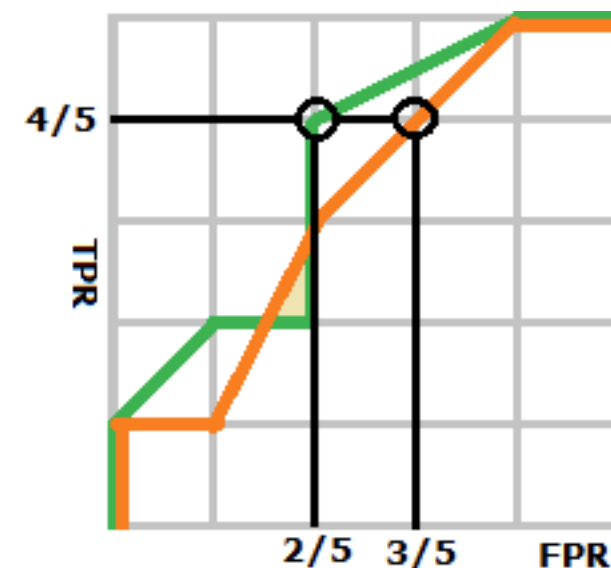
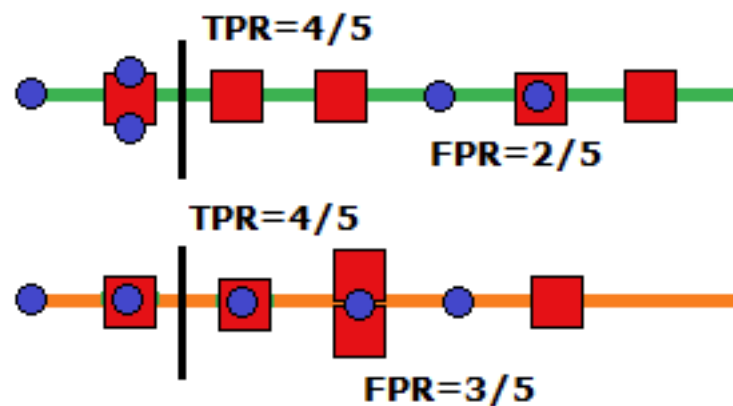
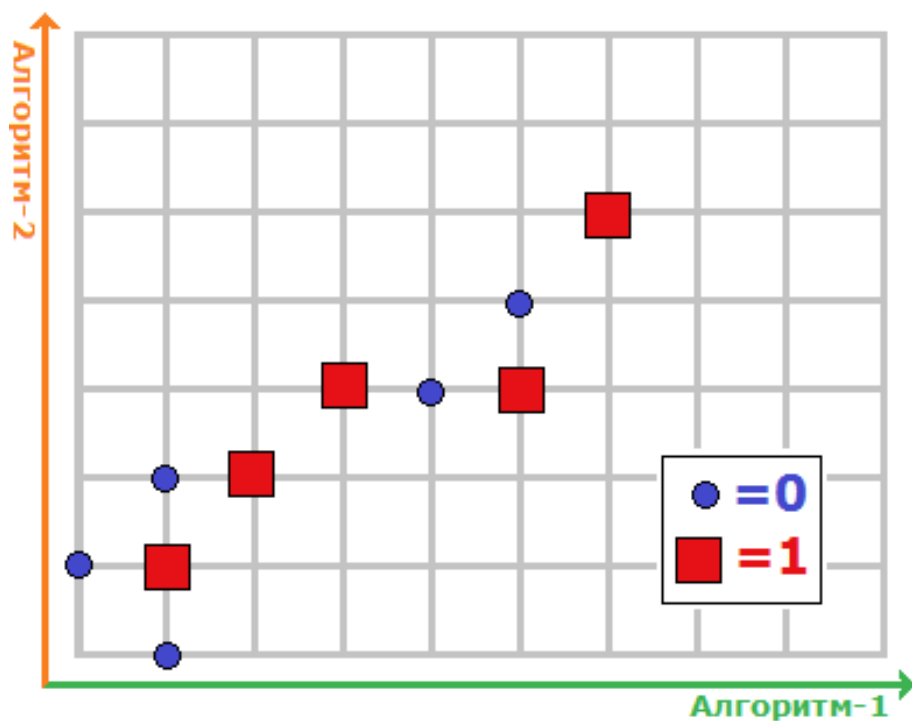
$y_{\text{pred}_i} > y_{\text{pred}_j}$ IF $y_{\text{true}_i} == 1$ and $y_{\text{true}_j} == 0$

То же самое (с точностью до линейного преобразования):

Средняя по всем порогам доля хороших среди одобренных

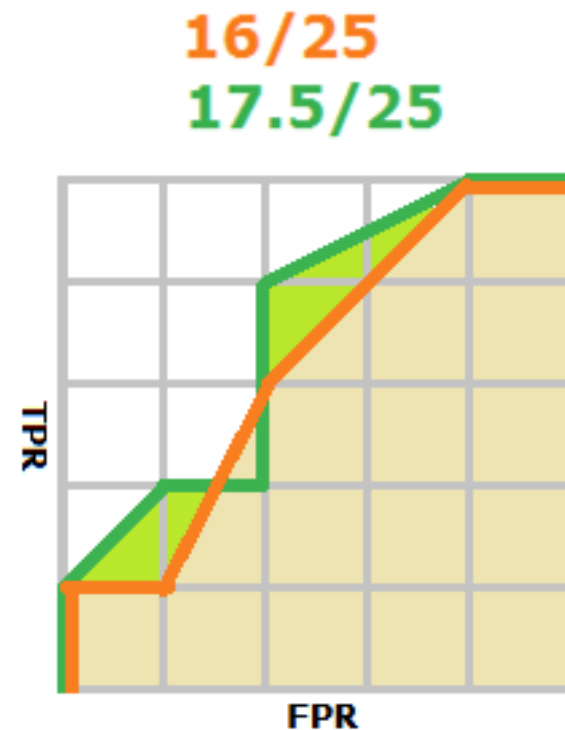
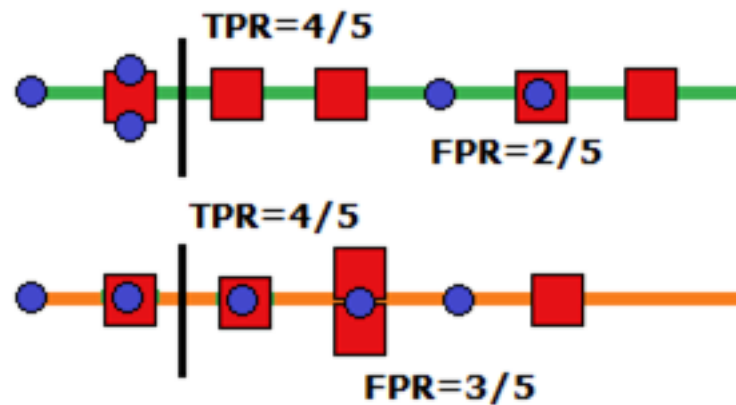
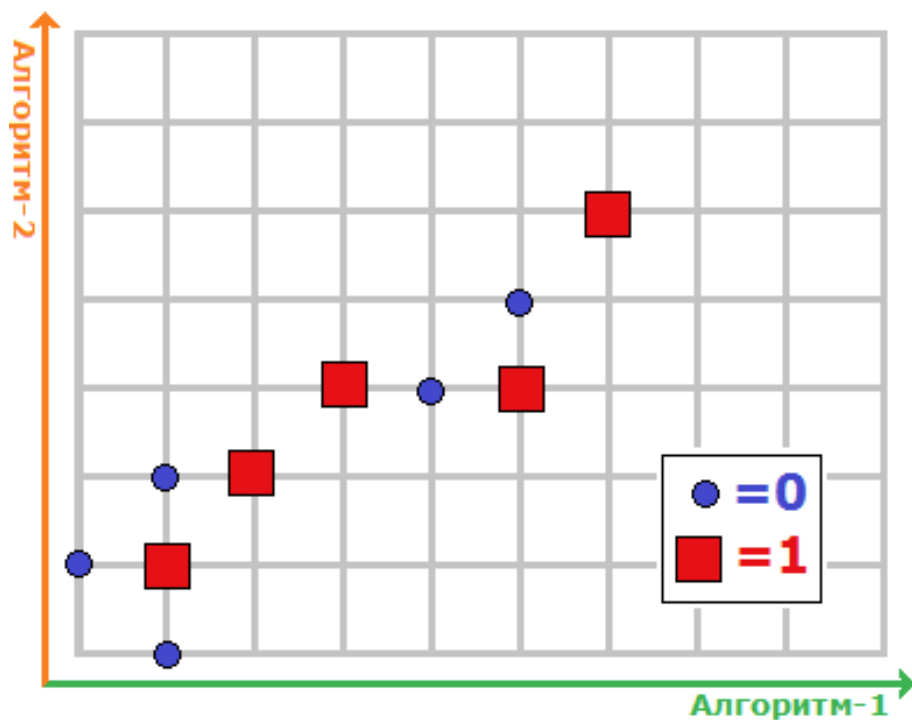
AUC (ROC)

Пример: построить ROC-кривую для предсказаний двух алгоритмов



AUC (ROC)

Пример: построить ROC-кривую для предсказаний двух алгоритмов



Logloss

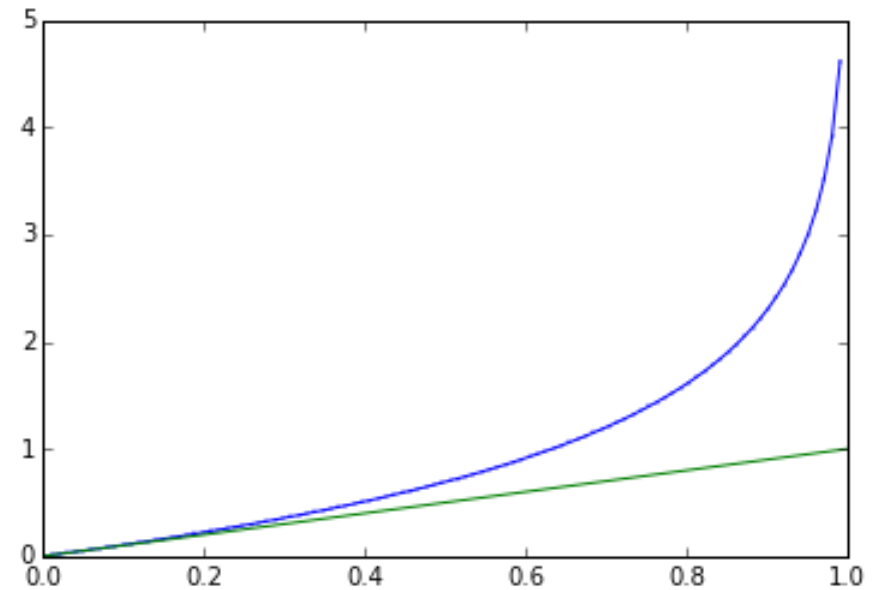
$$\text{LogLoss} = -\frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

Прогноз - действительное число от 0 до 1

Лучшее константное предсказание - среднее,
то есть частота класса 1

Выгодней сделать много незначительно отличающихся
от истины предсказаний, чем мало, отличающихся
значительно

Это средний логарифм бинарного правдоподобия ☺



По X: abs(ytrue - ypred)

По Y: LogLoss

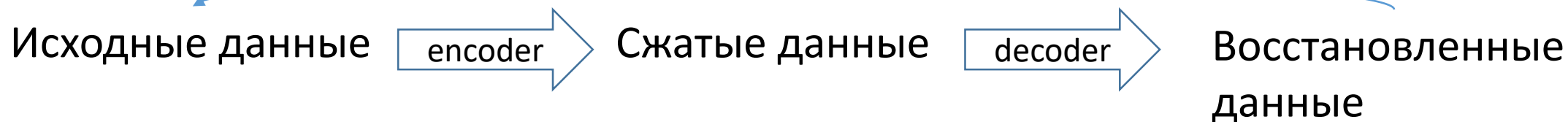
Многоклассовая классификация

- Accuracy, Logloss – обобщаются
- Precision, Recall, F1, ROC AUC – не обобщаются
 - Можно усреднить по классам
 - Усреднять можно с весами
 - Можно смотреть на confusion matrix глазами
 - Можно наложить свои потери за каждый вид ошибки

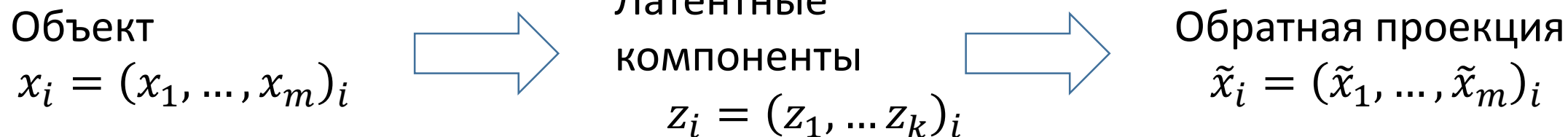
Офлайн-метрики
обучения без учителя

Кодер-декодер

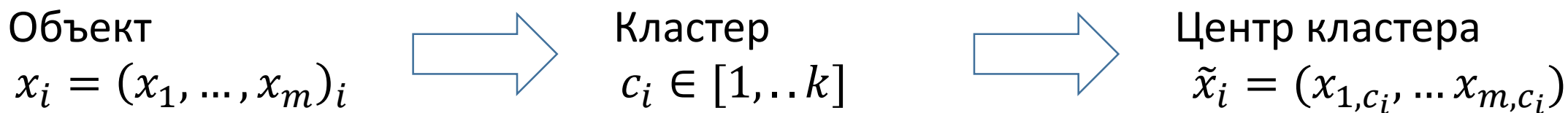
Можно сравнить любой метрикой для регрессии



Сокращение размерности



Кластеризация



Измерение плотности распределения

- Кластеризация – поиск областей высокой плотности
- Сокращение размерности – поиск подпространства высокой плотности
- Поиск аномалий – поиск объектов в области низкой плоскости
- Всё зависит от качества оценки плотности!
- Это качество измеряется правдоподобием наблюдений

Итог

- Качество модели часто определяется качеством признаков
- Для регрессии меряем «расстояние» от цели до прогноза
- Для классификации меряем долю верных ответов и качество ранжирования
- Полезно следить за метриками разной природы