Как проверить качество алгоритма?

- По обучающей выборке
  - алгоритм подогнан под эту выборку
  - при переобучении качество будет хорошим при отсутствии обобщающей способности
- По отложенной выборке
- С помощью кросс-валидации

Отложенная выборка (hold-out)

Выборка разбивается на две части: обучающую и валидационную.

В каком соотношении разбивать?

- Если обучающая выборка маленькая, то оценка качества будет слишком пессимистичной
- Если валидационная выборка маленькая, то оценка будет неточной
- Типичный выбор: 70/30

Проблемы:

- Результат сильно зависит от разбиения — каждый объект участвует или только в обучении, или только в валидации
- Если сравнивать много моделей, то есть риск подгонки под конкретную валидационную выборку

Кросс-валидация

Выборка разбивается на k частей
Каждая по очереди выступает как валидационная

Какое k выбирать?

- Небольшие k — пессимистичные, но точные оценки
- Большие k — несмещенные оценки с большой дисперсия
- Типичный выбор: $k = 5$

Проблема:

- Нужно обучать k алгоритмов

Как разбивать выборку?

Оба метода предполагают, что все объекты **принадлежат одному распределению** и независимы

- Задача: предсказание увольнений сотрудников компании в следующем полугодии
- Число увольнений сильно зависит от экономической ситуации — распределение зависит от времени
- Если разбивать сотрудников случайно, то алгоритм «подглядит» в распределение
- Оценка качества будет завышенной — алгоритм знает информацию, которая недоступна в реальных условиях
- Нужно разбивать по времени

[картинка с распределениями увольнений в разные моменты времени]

Как разбивать выборку?

Оба метода предполагают, что все объекты принадлежат одному распределению и **независимы**

- Задача: предсказание скорости транспортного потока на участках дорог в Москве
- Скорость на смежных участках практически одинаковая — объекты зависимые
- Если разбивать участки случайно, то алгоритм «подглядит» скорость на соседних участках в этот же момент времени
- Оценка качества будет завышенной — алгоритм знает информацию, которая недоступна в реальных условиях
- Нужно разбивать участки с учетом близости или по времени

[картинка со скоростью потока]

Отбор признаков

Как не надо отбирать признаки и понижать размерность:

- Находим оптимальные признаки по всей выборке
- Оцениваем качество алгоритма на новых признаках с помощью кросс-валидации

Признаки отобраны так, чтобы оптимизировать качество алгоритмов на всех объектах из выборки!

Как надо отбирать признаки и понижать размерность:

- Выбираем очередные обучающую и тестовую выборки в кросс-валидации
- Находим оптимальные признаки по обучающей выборке
- Проверяем качество на тестовой выборке

Заключение

- Проверять качество алгоритма нужно на выборке, отличной от обучающей
- Можно оценивать качество по отложенной выборке или по кросс-валидации
- Кросс-валидация лучше, но при этом дольше работает
- Если объекты зависят друг от друга или относятся к разным распределениям, то это надо учитывать при разбиении
- Нельзя отбирать признаки до кросс-валидации