



Кросс-валидация и проект по линейной регрессии



Кросс-валидация и проект по линейной регрессии

- Ранее мы выполнили очистку данных для Ames, и мы готовы к проекту по линейной регрессии, где мы проверим Ваши новые знания и построим модель предсказания цены продажи домов.
- Но перед этим мы должны рассмотреть ещё две важные темы.



Кросс-валидация и проект по линейной регрессии

- Обзор раздела
 - Кросс-валидация
 - Разбиение Train | Test
 - Разбиение Train | Validation | Test
 - Scikit-learn `cross_val_score`
 - Scikit-learn `cross_validate`
 - Поиск по сетке – Grid Search
 - Проект по линейной регрессии



Кросс-валидация и проект по линейной регрессии

- Ранее мы видели модели со встроенной кросс-валидацией (например, RidgeCV).
- Здесь мы продолжим эту дискуссию и рассмотрим общие инструменты Scikit-Learn для любых моделей
- Позднее это позволит нам применять поиск по сетке (grid search) для поиска оптимальных значений гиперпараметров.



Кросс-валидация и проект по линейной регрессии

- Мы начнём с самого простого процесса кросс-валидации для разбиения Train | Test, и далее постепенно перейдём к разбиению данных на K частей и продвинутой кросс-валидации.
- Давайте начнём!



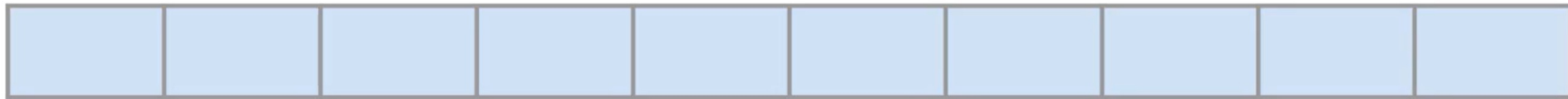
Кросс-валидация

Разбиение Train | Test



Кросс-валидация (перекрёстная проверка)

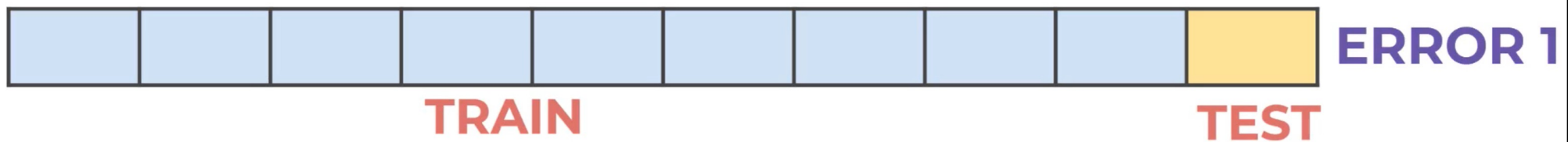
- Начнём с полного набора всех данных:





Кросс-валидация (перекрёстная проверка)

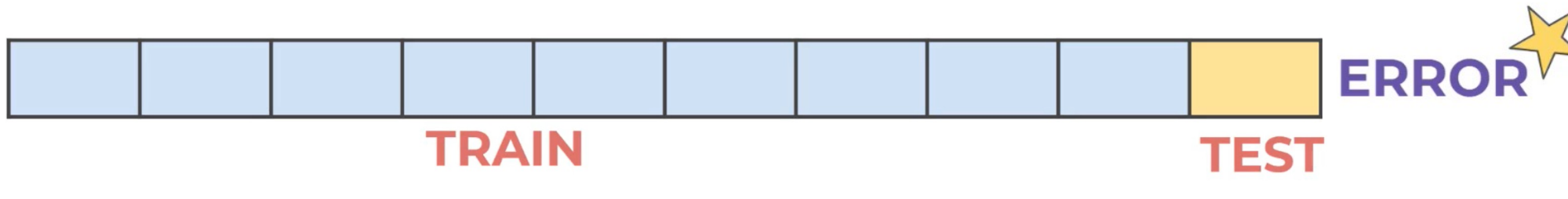
- Разбиение на обучающий и тестовый наборы:





Кросс-валидация (перекрёстная проверка)

- Можем уточнить параметры модели:





Кросс-валидация (перекрёстная проверка)

- Посмотрим, как выполнить эти шаги в Python!



Кросс-валидация

Разбиение Train | Validation | Test



Кросс-валидация (перекрёстная проверка)

- Разбиение Train | Test не позволяет нам отложить часть данных для оценки модели – такие данные, которых модели ещё не видела.
- Оптимизация гиперпараметров на тестовых данных оправданна, и обычно не считается “утечкой данных”. Но потенциально это может влиять на корректность оценки модели.



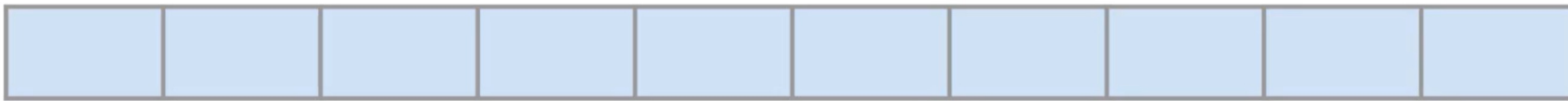
Кросс-валидация (перекрёстная проверка)

- Если же мы хотим получить чистый результат, то оценку нужно проводить на тех данных, которые были заранее отложены в сторону.
- Давайте быстро вспомним теорию и далее применим это на практике!



Кросс-валидация (перекрёстная проверка)

- Начнём с полного набора всех данных:





Кросс-валидация (перекрёстная проверка)

- Разбиение “Train – Validation – Test”



TRAIN

VALIDATION

TEST



Кросс-валидация (перекрёстная проверка)

- Разбиение “Train – Validation – Test”



TRAIN

VALIDATION

TEST

- Обучение на данных Train



Кросс-валидация (перекрёстная проверка)

- Разбиение “Train – Validation – Test”



TRAIN

VALIDATION

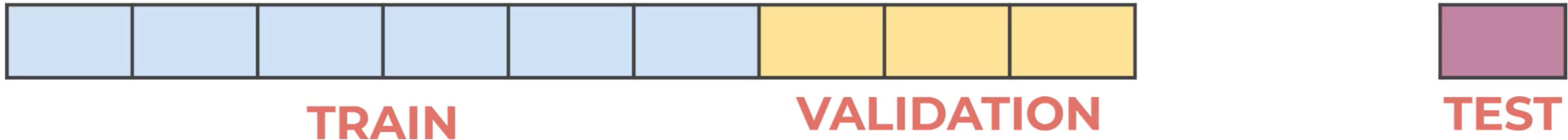
TEST

- Обучение на данных Train
- Проверка и выбор гиперпараметров на Validation



Кросс-валидация (перекрёстная проверка)

- Разбиение “Train – Validation – Test”



- Обучение на данных Train
- Проверка и выбор гиперпараметров на Validation
- Финальная проверка модели на данных Test



Кросс-валидация (перекрёстная проверка)

- После финальной проверки модель не отлаживаем.
- Финальные тестовые данные не использовались ни для обучения, ни для подбора параметров.
- То есть, модель действительно ещё не видела эти данные.



Кросс-валидация (перекрёстная проверка)

- Чтобы сделать это в Scikit-Learn, мы выполним `train_test_split()` дважды:
 - Первый раз, чтобы отделить обучающий набор данных
 - Второй раз, чтобы разделить оставшиеся данные на Validation и Test



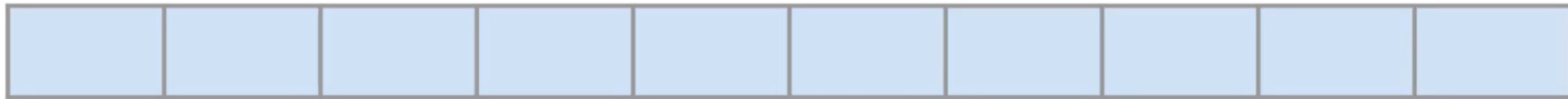
Кросс-валидация

Функция `cross_val_score`



Кросс-валидация (перекрёстная проверка)

- Начнём с полного набора всех данных:





Кросс-валидация (перекрёстная проверка)

- Берем $1/K$ часть данных для Validation



TRAIN

VALIDATION

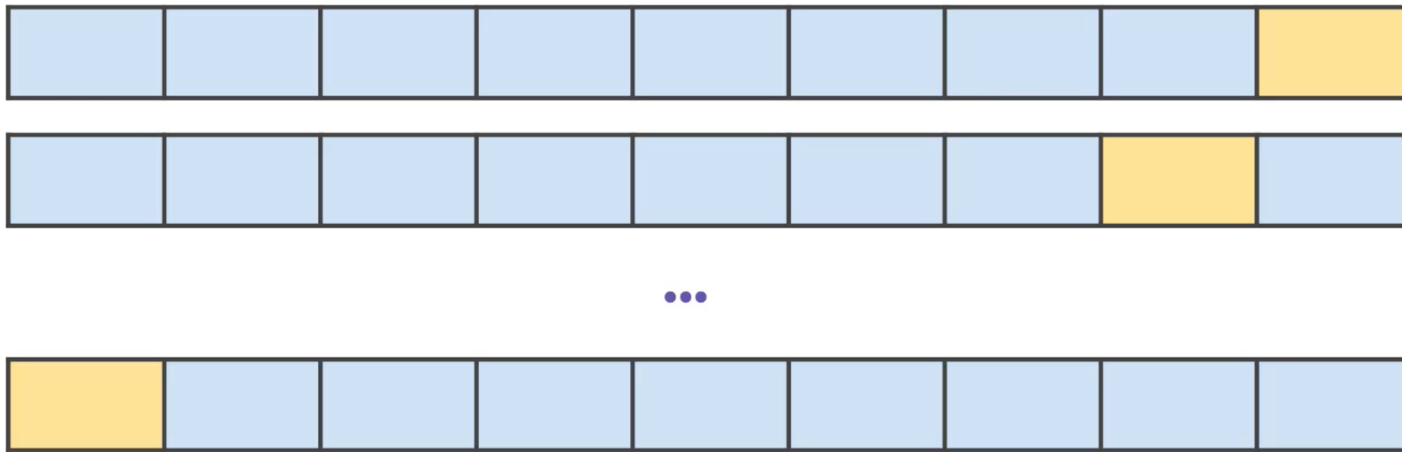


TEST



Кросс-валидация (перекрёстная проверка)

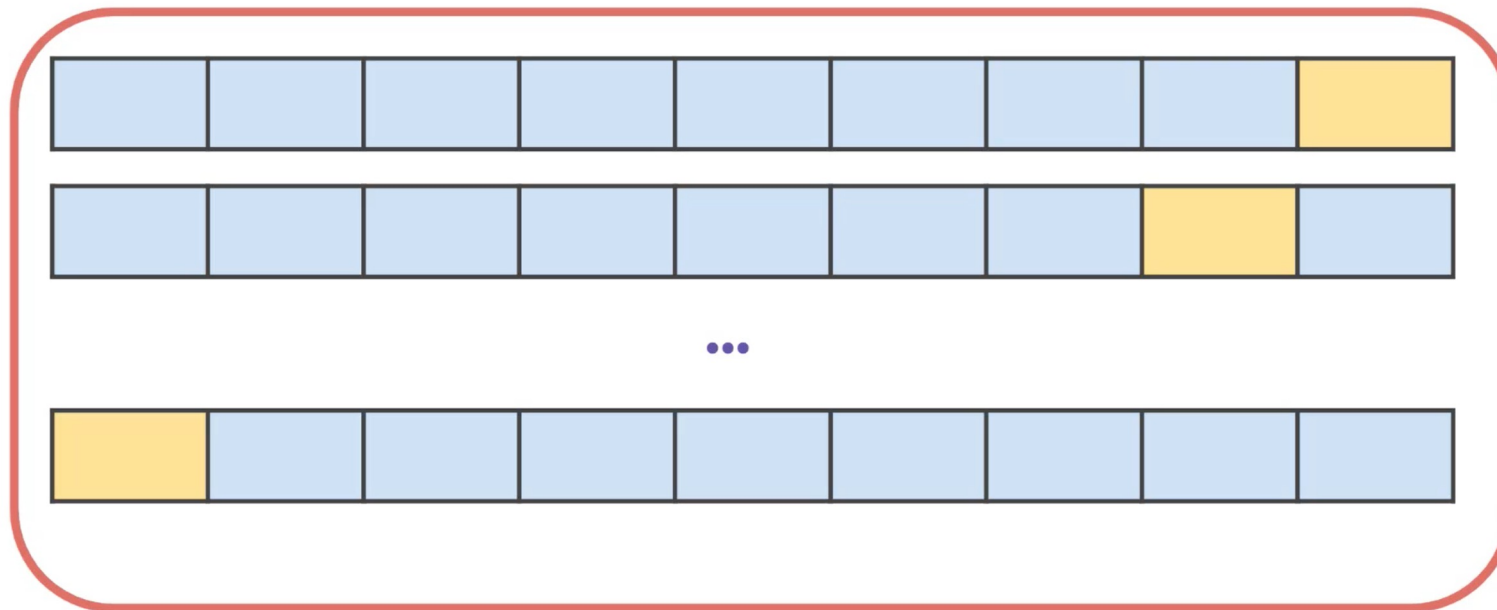
- Выполняем такое разбиение K раз.





Кросс-валидация (перекрёстная проверка)

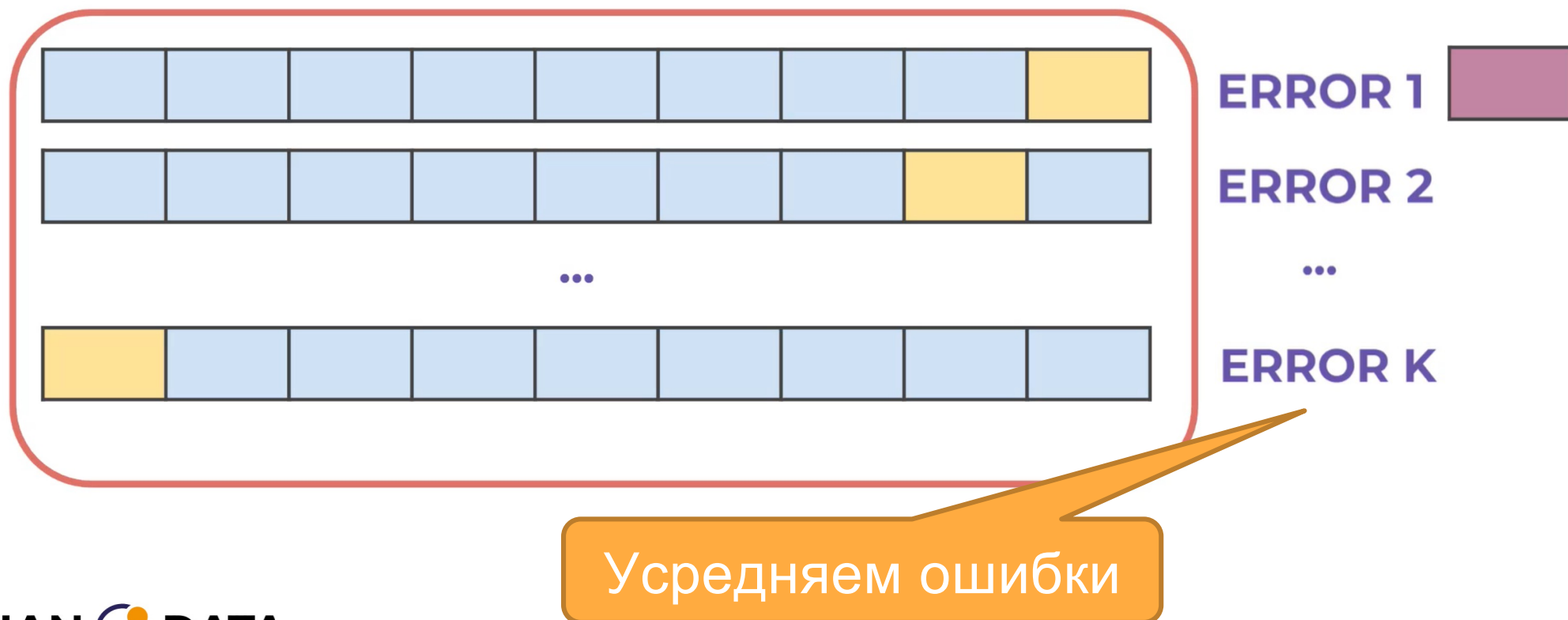
- Обучение/тюнинг только на этих данных





Кросс-валидация (перекрёстная проверка)

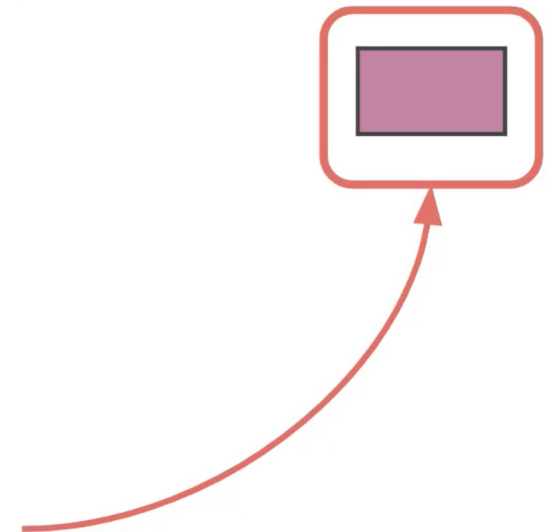
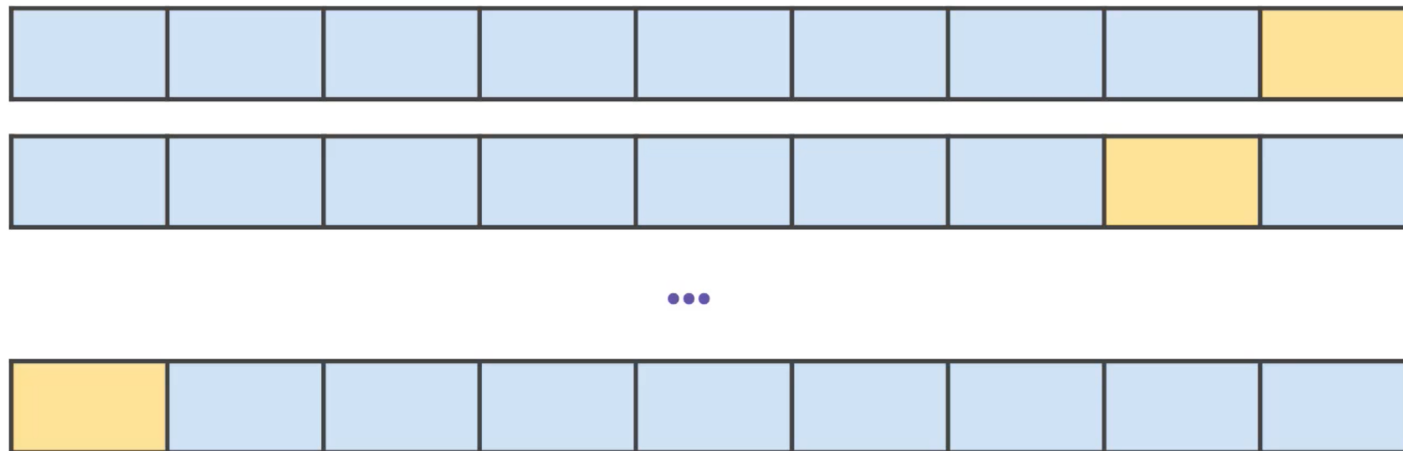
- Обучение/тюнинг только на этих данных





Кросс-валидация (перекрёстная проверка)

- После обучения/тюнинга выполняем финальную проверку на данных “Test”





Кросс-валидация (перекрёстная проверка)

- Метод `cross_val_score()` позволяет сделать это всё автоматически. На вход подаётся модель и обучающий набор данных.
- Это позволяет применить кросс-валидацию в K шагов для любой модели.
- Посмотрим, как это делается!



Кросс-валидация (перекрёстная проверка)

- Во многих методах регуляризации есть параметры для настройки с помощью методов кросс-валидации
- Во время курса, для простоты изложения, мы часто будем использовать простое разбиение на обучающий и тестовый наборы данных.



Кросс-валидация

Функция `cross_validate`



Кросс-валидация (перекрёстная проверка)

- Функция `cross_validate()` позволяет посмотреть различные метрики кросс-валидации, а также понять, сколько времени заняли процессы обучения и проверки.



Поиск по сетке (grid search)



Поиск по сетке (grid search)

- Сложные модели как правило имеют несколько гиперпараметров.
- Поиск по сетке – это способ обучения и оценки работы модели для различных комбинаций значений гиперпараметров.



Поиск по сетке (grid search)

- В Scikit-Learn есть класс GridSearchCV, позволяющий перебирать значения по словарю с применением кросс-валидации.
- Это позволяет выполнять кросс-валидацию и поиск по сетке универсальным способом для любой модели.



Проект по линейной регрессии

Обзор



Проект по линейной регрессии

Решения