Машинное обучение

Д.Ю. Хартьян

Масштабирование

Для каждого признака Xj мы вычисляем среднее значение μ_j и стандартное отклонение σ_j на обучающей выборке. Затем мы применяем следующее преобразование для каждого значения $x_{i,j}$ знака *j* в обучающей выборке:

$$x'_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j}$$

Таким образом, после масштабирования среднее значение каждого признака будет равно 0, а стандартное отклонение будет равно 1

Регуляризация

$$y = \mathbf{w_0} + \mathbf{w_1} x_1 + \mathbf{w_2} x_2 + \mathbf{w_3} x_3 + \mathbf{w_4} x_4$$

$$y = \mathbf{0.32} + \mathbf{0.87}\, x_1 + \mathbf{0.15}\, x_2 + \mathbf{0.46}\, x_3 + \mathbf{0.31}\, x_4$$

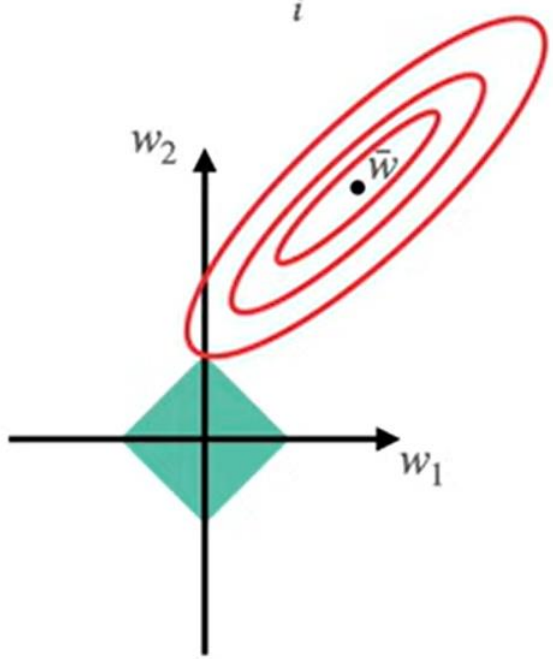
$$y = \mathbf{0.12} + \mathbf{184894168}\, x_1 + \mathbf{0.26}\, x_2 + \mathbf{0.13}\, x_3 + \mathbf{0.21}\, x_4$$

Регуляризация - штраф за большие веса, которые, как правило, свидетельствуют о переобучении

Регуляризация

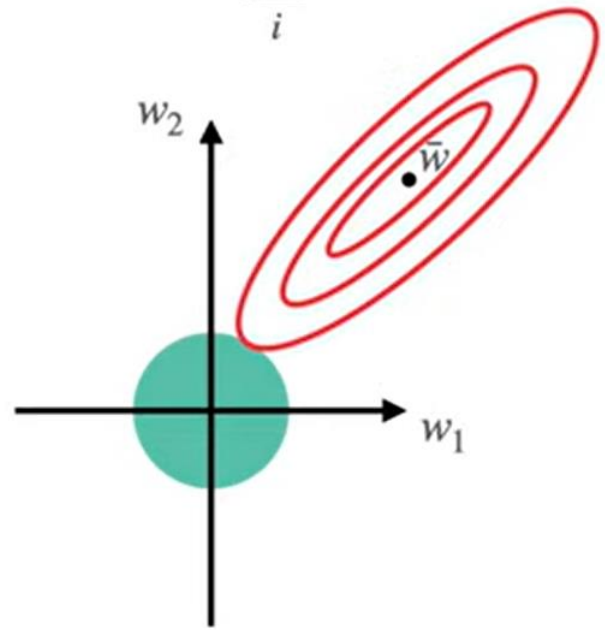
L1, LASSO

$$(y - Xw)^2 + \lambda \sum_i |w_i| \to min$$



L2, Ridge

$$(y - Xw)^2 + \lambda \sum_i w_i^2 \to min$$



Elastic Net = L1+L2

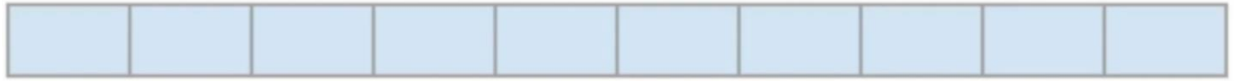
$$(y - Xw)^2 + \lambda_1 \sum_i |w_i| + \\ + \lambda_2 \sum_i w_i^2 \to min$$

Кросс-валидация

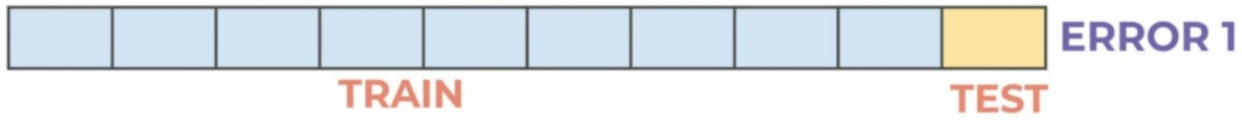
Кросс-валидация (cross-validation) - это метод оценки производительности модели машинного обучения, который позволяет оценить, насколько хорошо модель будет работать на новых данных, которые не были использованы при ее обучении.

Кросс-валидация

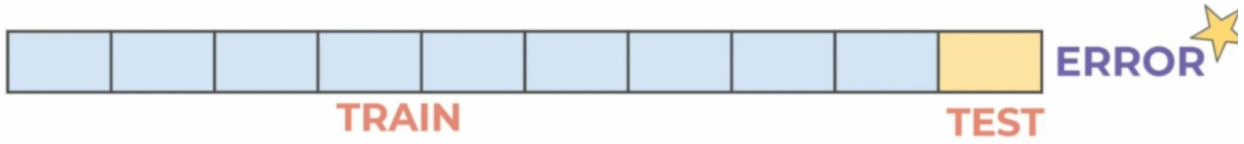
Начнём с полного набора всех данных:



Разбиение на обучающий и тестовый наборы:



Можем уточнить параметры модели:

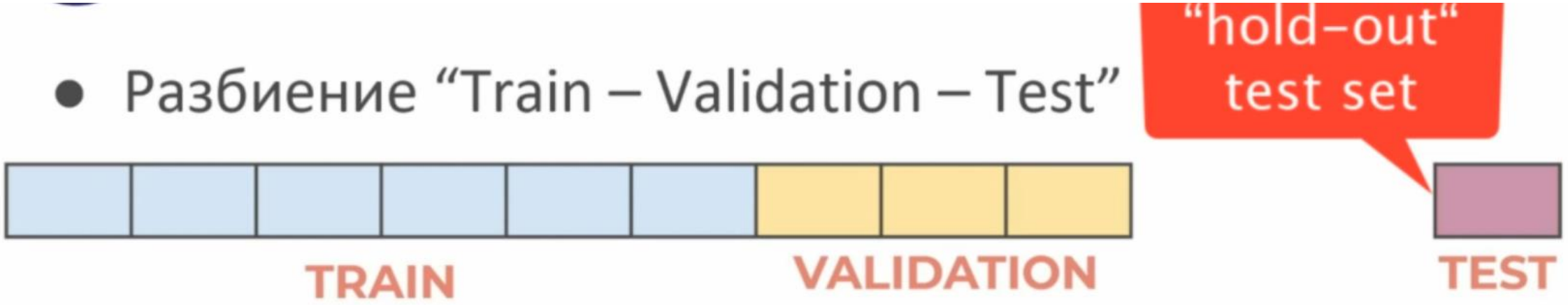


Кросс-валидация (перекрестная проверка)

- 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 (перекрестная проверка)

- Разбиение “Train – Validation – Test”



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

Кросс-валидация (перекрестная 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.

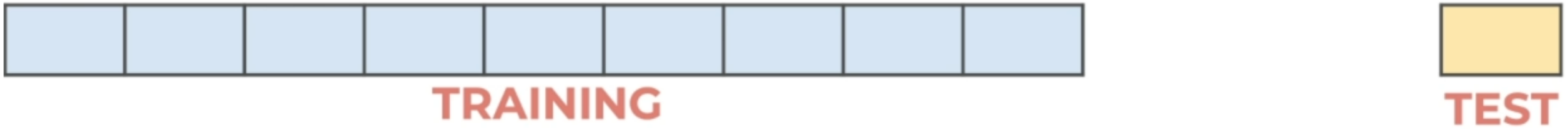
Чтобы сделать это в Scikit-Learn, мы выполним train_test_split() дважды:

- Первый раз, чтобы отделить обучающий набор данных
- Второй раз, чтобы разделить оставшиеся данные на Validation и Test

Кросс-валидация с помощью cross_val_score

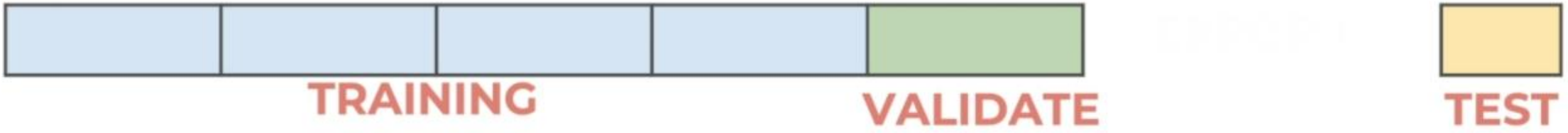
Кросс-валидация (перекрестная проверка)

- Выбираем число K для разбиения K-Fold Split



Кросс-валидация (перекрестная проверка)

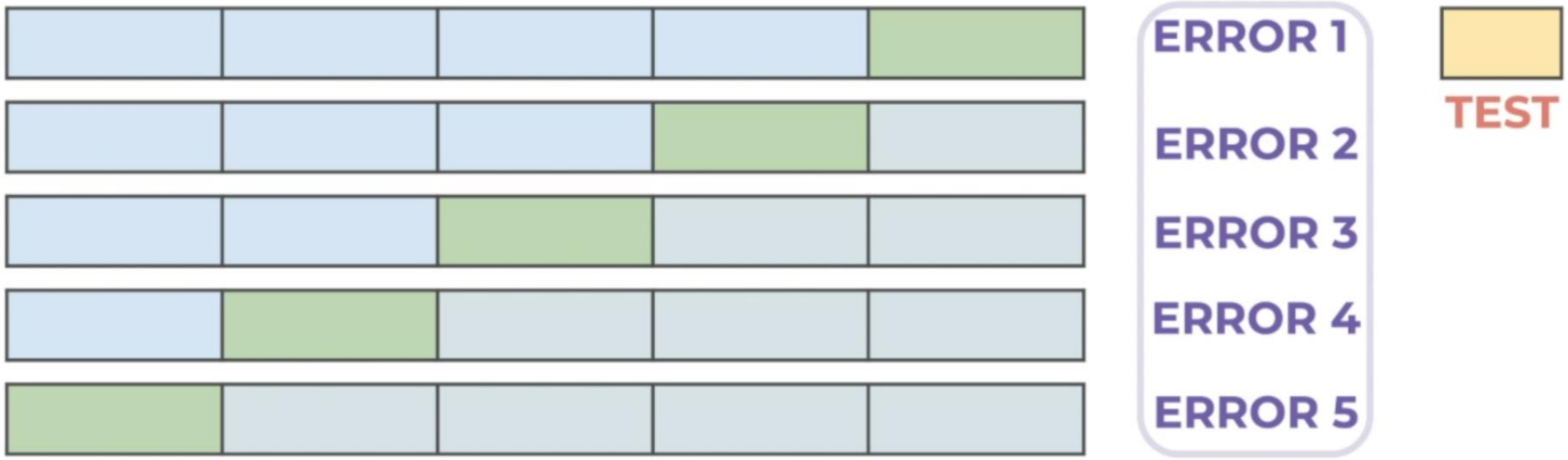
- Обучаем на K-1 частях и проверяем на 1 части



- Получаем метрику ошибки для этого разбиения:

Кросс-валидация с помощью cross_val_score

- Усредняем ошибки; настраиваем гиперпараметры



Находим среднее значение всех ошибок AVG Error.

Настраиваем гиперпараметры и продолжаем до тех пор пока AVG Error нас не начнет устраивать

Кросс-валидация с помощью cross_validate

Функция cross_validate отличается от cross_val_score двумя аспектами:
эта функция позволяет использовать для оценки несколько метрик;
она возвращает не только оценку на тестовом наборе (test score), но и словарь с замерами времени обучения и скоринга, а также - опционально - оценки на обучающем наборе и объекты estimator.
В случае одной метрики для оценки, когда параметр scoring является строкой string, вызываемым объектом callable или значением None, ключи словаря будут следующими:
- `['test_score', 'fit_time', 'score_time']`
А в случае нескольких метрик для оценки, возвращаемый словарь будет содержать следующие ключи:
`['test_<scorer1_name>', 'test_<scorer2_name>', 'test_<scorer...>', 'fit_time', 'score_time']`
return_train_score по умолчанию принимает значение False, чтобы сэкономить вычислительные ресурсы. Чтобы посчитать оценки на обучающем наборе, достаточно установить этот параметр в значение True.