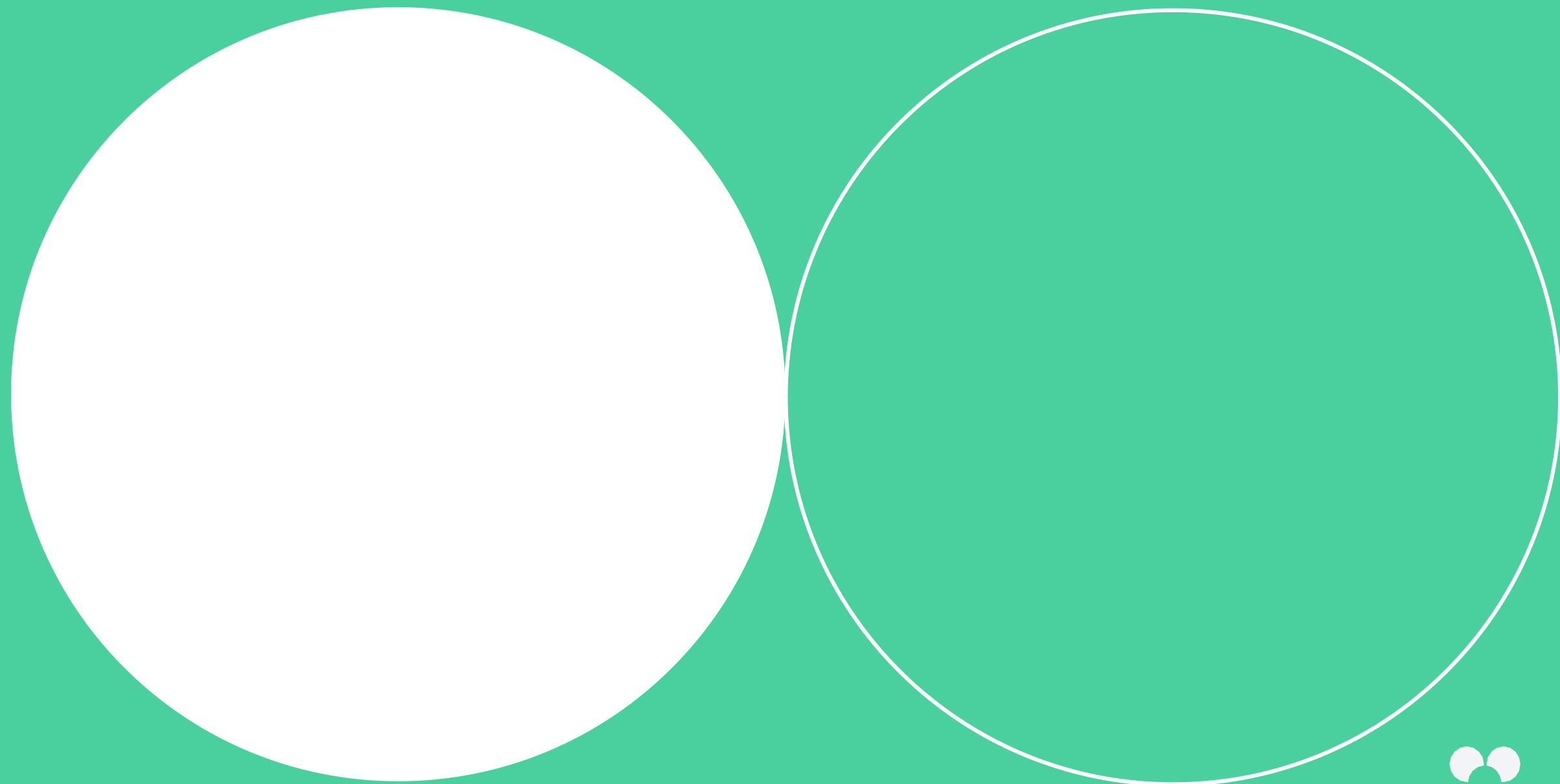


Метрики качества модели и переобучение



Цели занятия

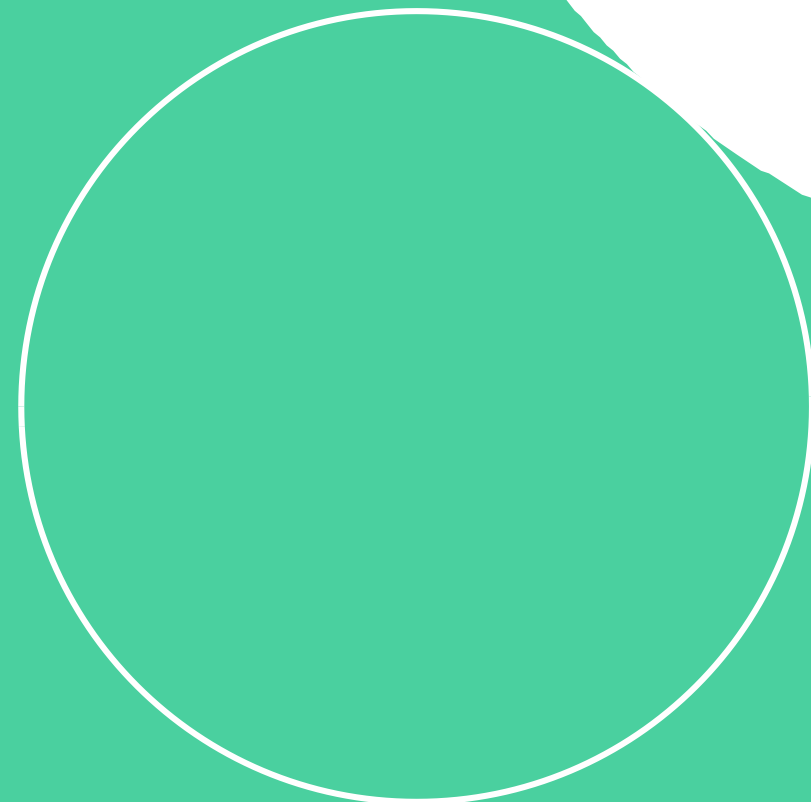


В конце занятия вы:

- 1 Как проводить кросс-валидацию для проверки качества обучения модели
- 2 Будете знать основные метрики качества обучения моделей регрессии и классификации
- 3 Будете знать что такое переобучение и как с ним бороться



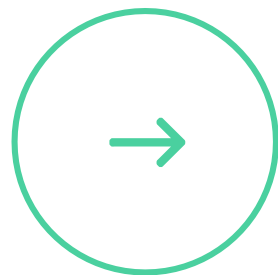
Обучающая, тестовая выборка



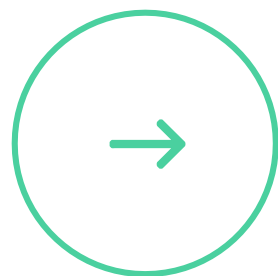
Проблема при обучении моделей

Модель может хорошо работать при обучении, однако сильно терять в качестве на новых данных (большая ошибка обобщения).

Обучающая выборка

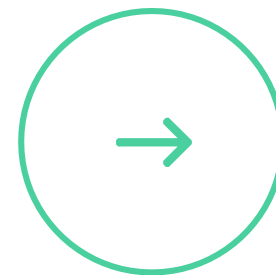


Содержит значения признаков и целевой переменной.

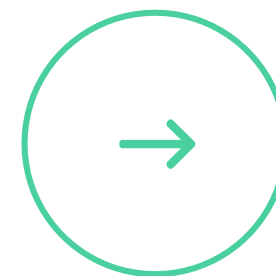


На обучающей выборке строим модель.

Тестовая выборка



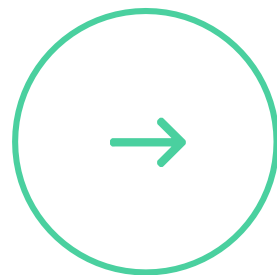
Содержит значения признаков, по которым необходимо предсказать известные значение целевой переменной.



Оцениваем качество различных вариантов модели.



Разбиваем обучающую выборку



Разбиваем обучающую выборку на 2 части.
На одной будем тренировать модель (минимизировать ошибку обучения), на другой – проверять (минимизировать ошибку обобщения) (т.е. использовать в качестве тестовой, только с известной целевой переменной)

```
from sklearn.model_selection import train_test_split  
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size = 0.3, random_state = 0 )
```

Обучающая выборка



Training

TEST

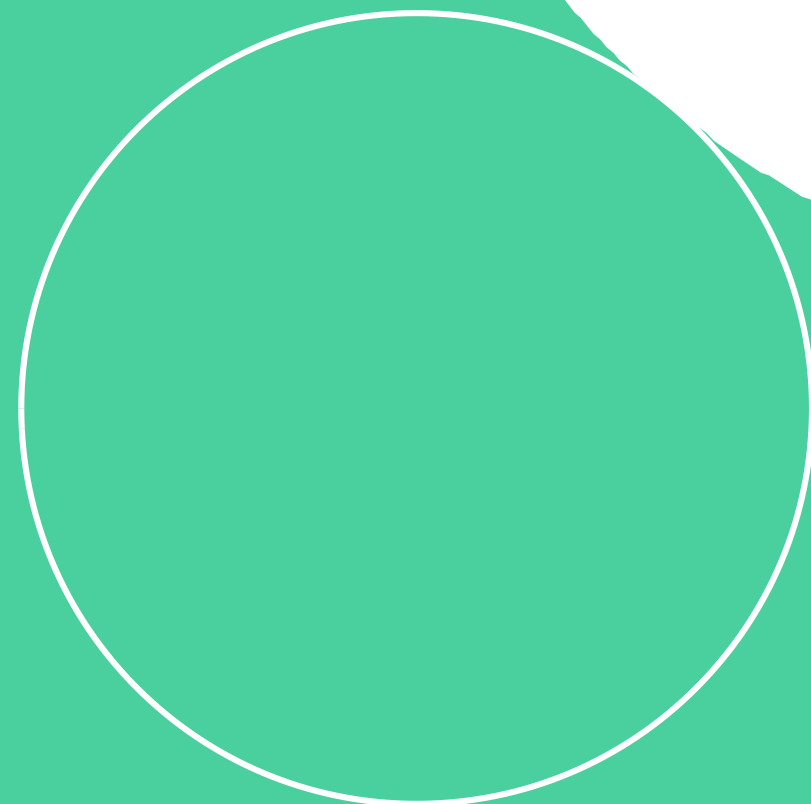


Практика

LOGRES_AFFAIR.IPYNB

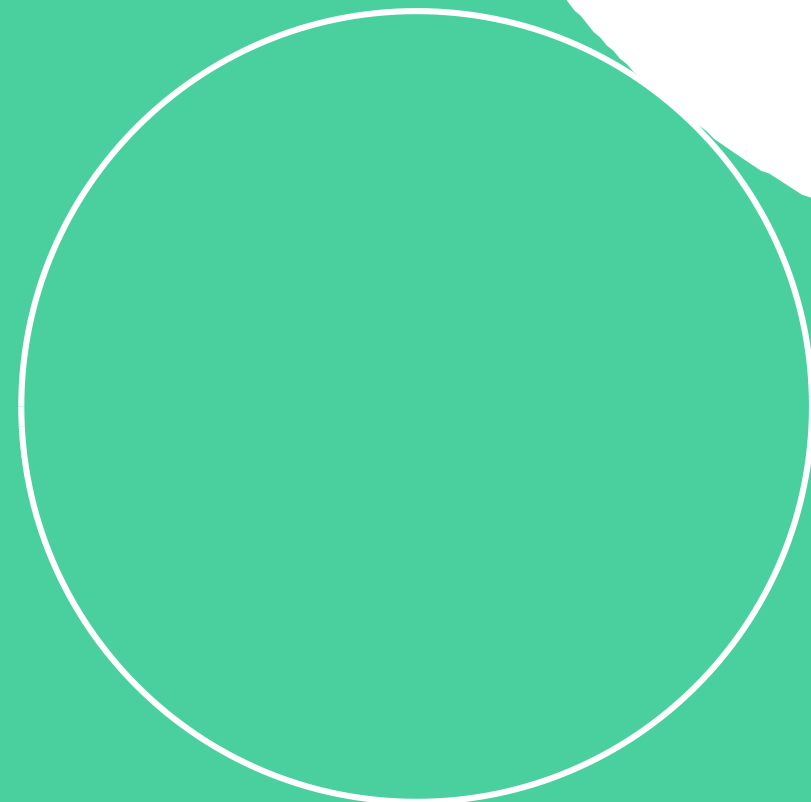


Оценка качества модели



Регрессия

MSE, MAE и R2



Метрики регрессии

Средняя абсолютная ошибка

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

Средняя квадратичная ошибка

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2$$

Квадратный корень средней квадратичной ошибки

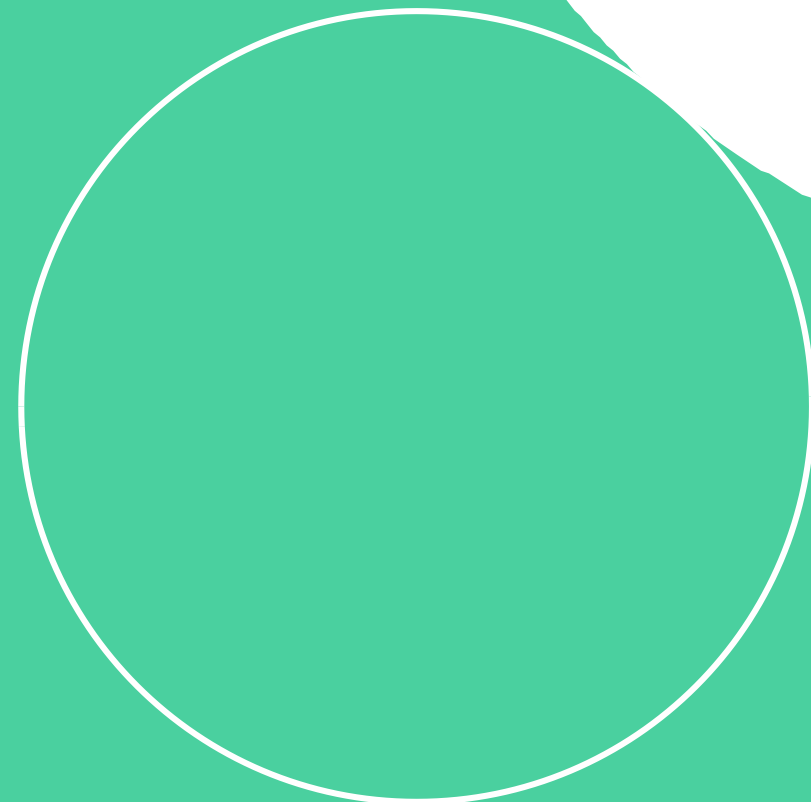
$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2}$$

Коэффициент детерминации - это доля дисперсии зависимой переменной, объясняемая рассматриваемой моделью.

$$R^2 = 1 - \frac{\sum (y_i - \hat{y})^2}{\sum (y_i - \bar{y})^2}$$



Классификация Precision и Recall Точность и Полнота



Порог для тестовой выборки

```
model = LogisticRegression()

model.fit(X_train, y_train)
predictions = model.predict_proba(X_test)

zip(predictions[:, 1], y_test)
```

```
[(0.64583193796528038, 0),
 (0.075906148028446599, 0),
 (0.2704606033743272, 0),
 (0.26938542699540474, 0),
 (0.26433391263337475, 1),
 (0.1443590034736055, 0),
 (0.17840859560894495, 0),
 (0.21871761029690232, 0),
 (0.75293068528621931, 1),
 (0.2694630112685994, 0),
 (0.11209927315788928, 0),
 (0.18717054508217956, 0),
 (0.081787486664569364, 0)]
```

Выберем порог, выше которого будем считать полученное значение принадлежащим первому классу, а ниже – второму.

Это определит долю угаданных моделью значений.



Матрица ошибок для порога

	Actual positive	Actual negative
Predicted positive	True positive	False positive
Predicted negative	False negative	True negative

False positive – ошибка I рода
(ложная тревога)

False negative – ошибка II рода
(пропуск цели)



Точность

	Actual positive	Actual negative
Predicted positive	True positive	False positive
Predicted negative	False negative	True negative

Accuracy – доля правильно предсказанных от всех вариантов

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$



Почему точности недостаточно

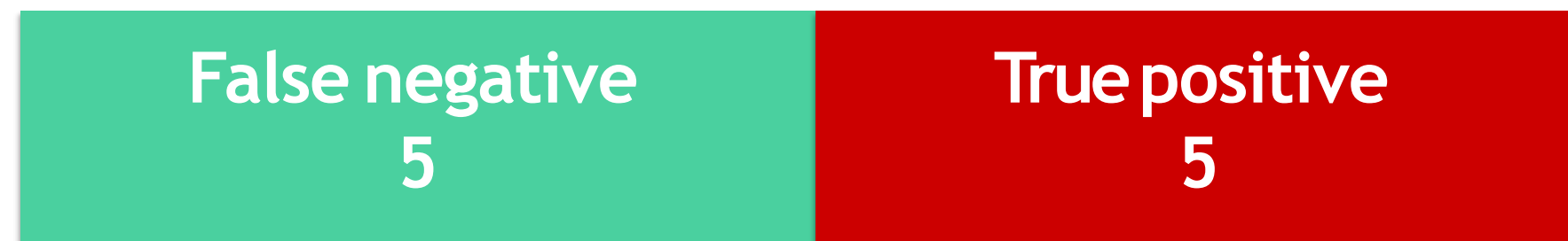
100 обычных писем



На почту пришло 100 обычных писем и из них 10 писем спама.

Наша модель из 100 обычных 10 классифицировала как спам. Из 10 спам-писем – 5 как спам

10 спам-писем



Почему точности не достаточно

	Actual positive	Actual negative
Predicted positive	5	5
Predicted negative	10	90

Ассурасу – доля правильно предсказанных от всех вариантов

$$Accuracy = \frac{5 + 90}{5 + 90 + 10 + 5} = 86\%$$



Почему точности недостаточно

100 обычных писем

True negative
100

Возьмем модель, которая
считает все письма обычными

10 спам-писем

False negative
10



Почему точности не достаточно

	Actual positive	Actual negative
Predicted positive	0	0
Predicted negative	10	100

Возьмем модель, которая считает все письма обычными

$$Accuracy = \frac{0 + 100}{0 + 100 + 0 + 10} = 91\%$$



Precision

	Actual positive	Actual negative
Predicted positive	True positive	False positive
Predicted negative	False negative	True negative

Precision – доля правильно предсказанных среди причисленных моделью к категории 1

$$Precision = \frac{TP}{TP + FP}$$

Способность алгоритма отличать данный класс от других классов



Recall

	Actual positive	Actual negative
Predicted positive	True positive	False positive
Predicted negative	False negative	True negative

Recall – доля правильно предсказанные среди категории 1

$$Recall = \frac{TP}{TP + FN}$$

Синоним - True Positive Rate (sensitivity)

Способность алгоритма обнаруживать данный класс вообще



Precision и Recall для спама

100 обычных писем

True negative
100

10 спам-писем

False negative
10

	Actual positive	Actual negative
Predicted positive	0	0
Predicted negative	10	100



True positive rate

	Actual positive	Actual negative
Predicted positive	True positive	False positive
Predicted negative	False negative	True negative

True Positive Rate – доля
правильно предсказанных
среди категории 1

$$TPR = \frac{TP}{TP + FN}$$



False positive rate

	Actual positive	Actual negative
Predicted positive	True positive	False positive
Predicted negative	False negative	True negative

False Positive Rate – доля неправильно предсказанных среди относящихся к категории 0

$$FPR = \frac{FP}{FP + TN}$$

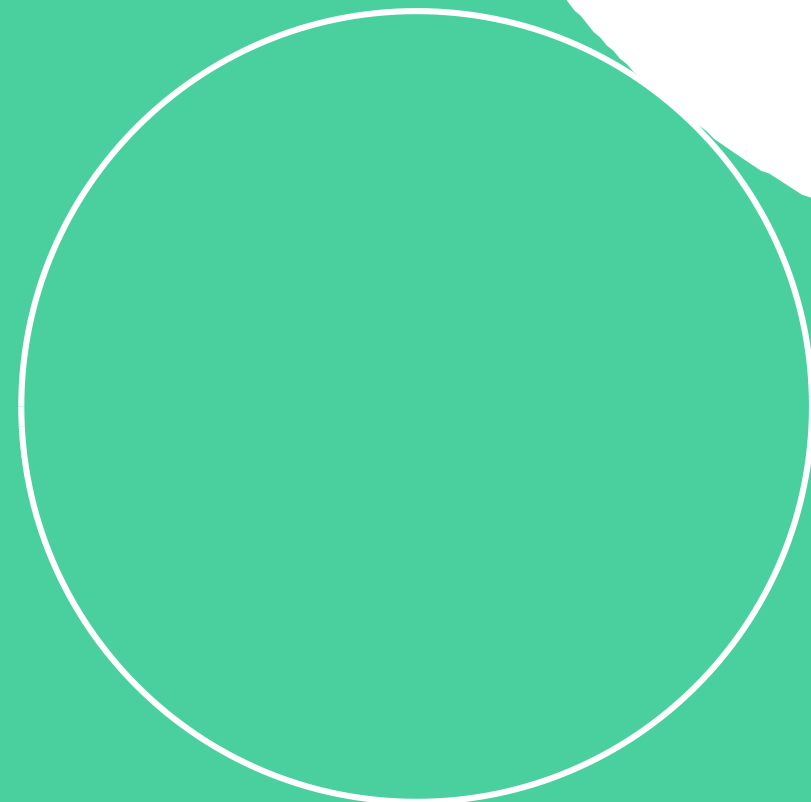


Практика

LOGRES_AFFAIR.IPYNB



Area under curve



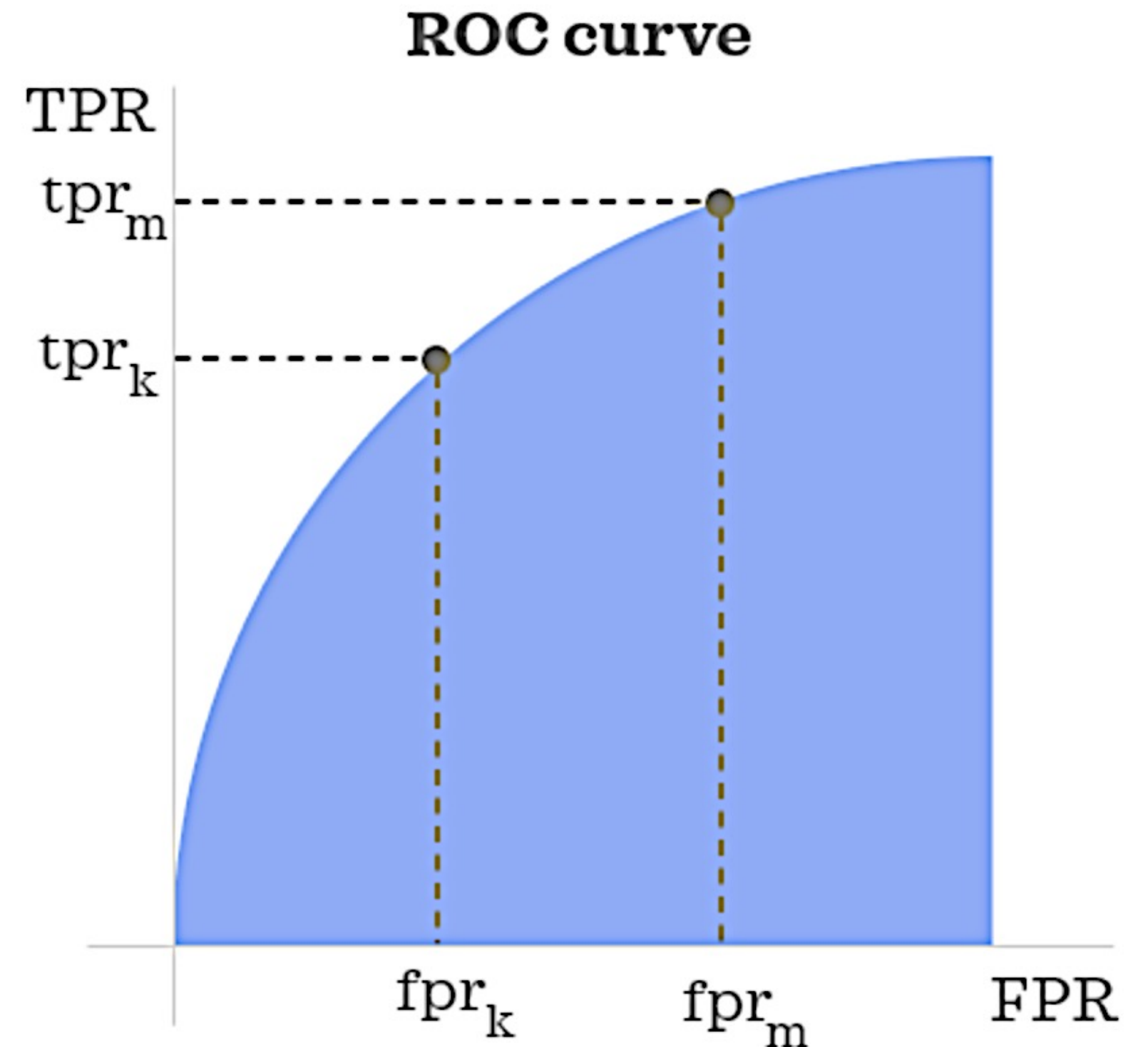
Receiver Operating Characteristic ROC AUC

Area under the curve

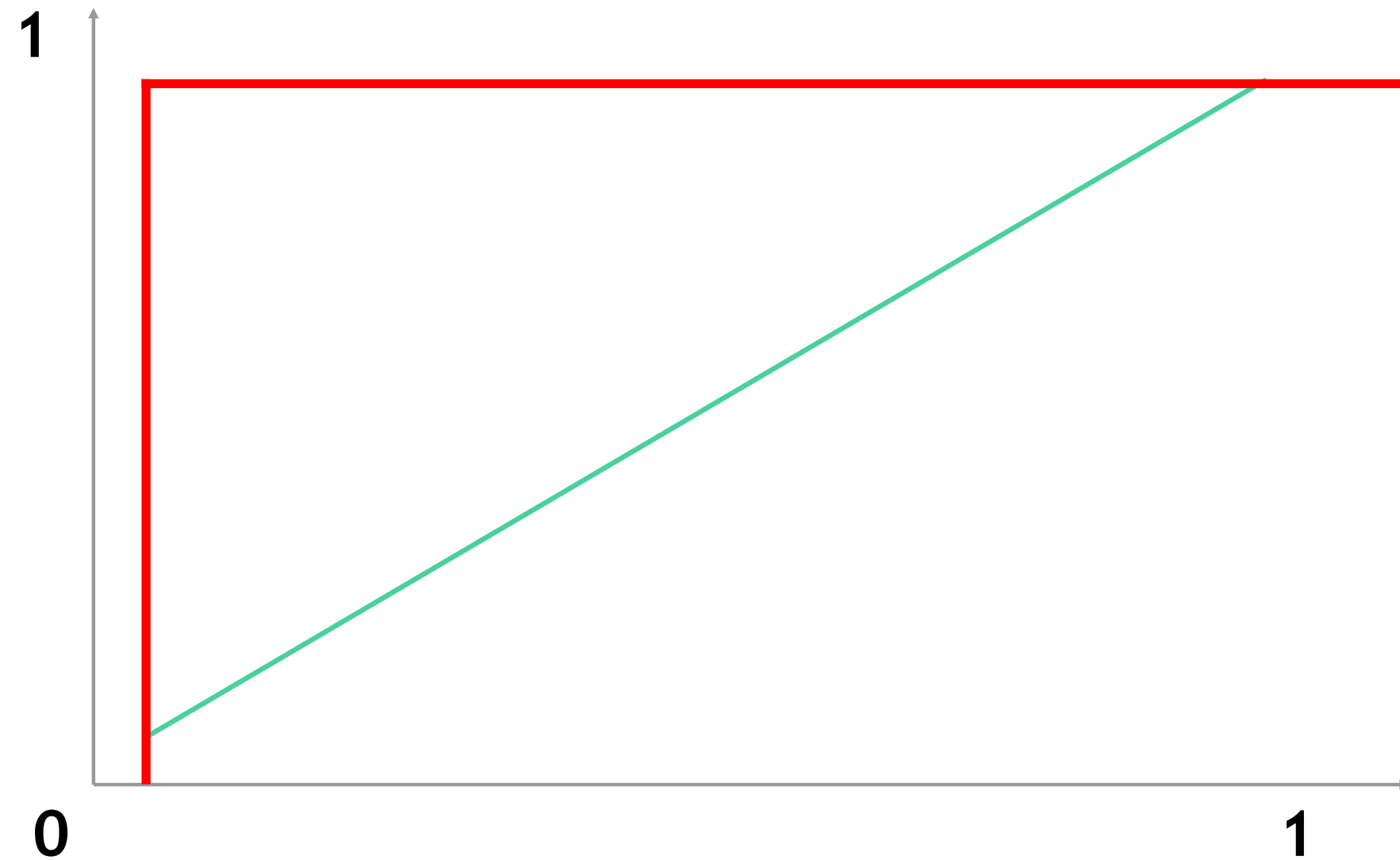
Кривая ROC показывает нам взаимосвязь между показателем ложных положительных результатов (*FPR*) и истинно положительным показателем (*TPR*) для разных пороговых значений.

$$FPR = \frac{FP}{FP + TN}$$

$$TPR = \frac{TP}{TP + FN}$$



Идеальный случай



Модель предсказывает
абсолютно верно

$TPR = 1$

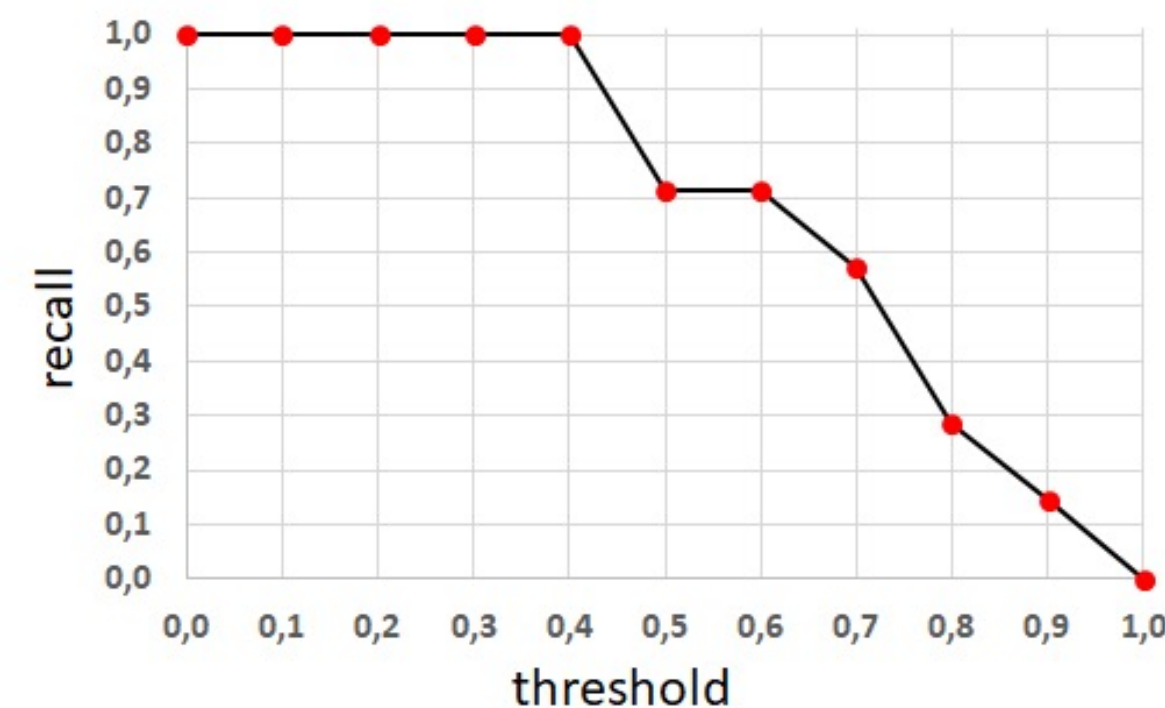
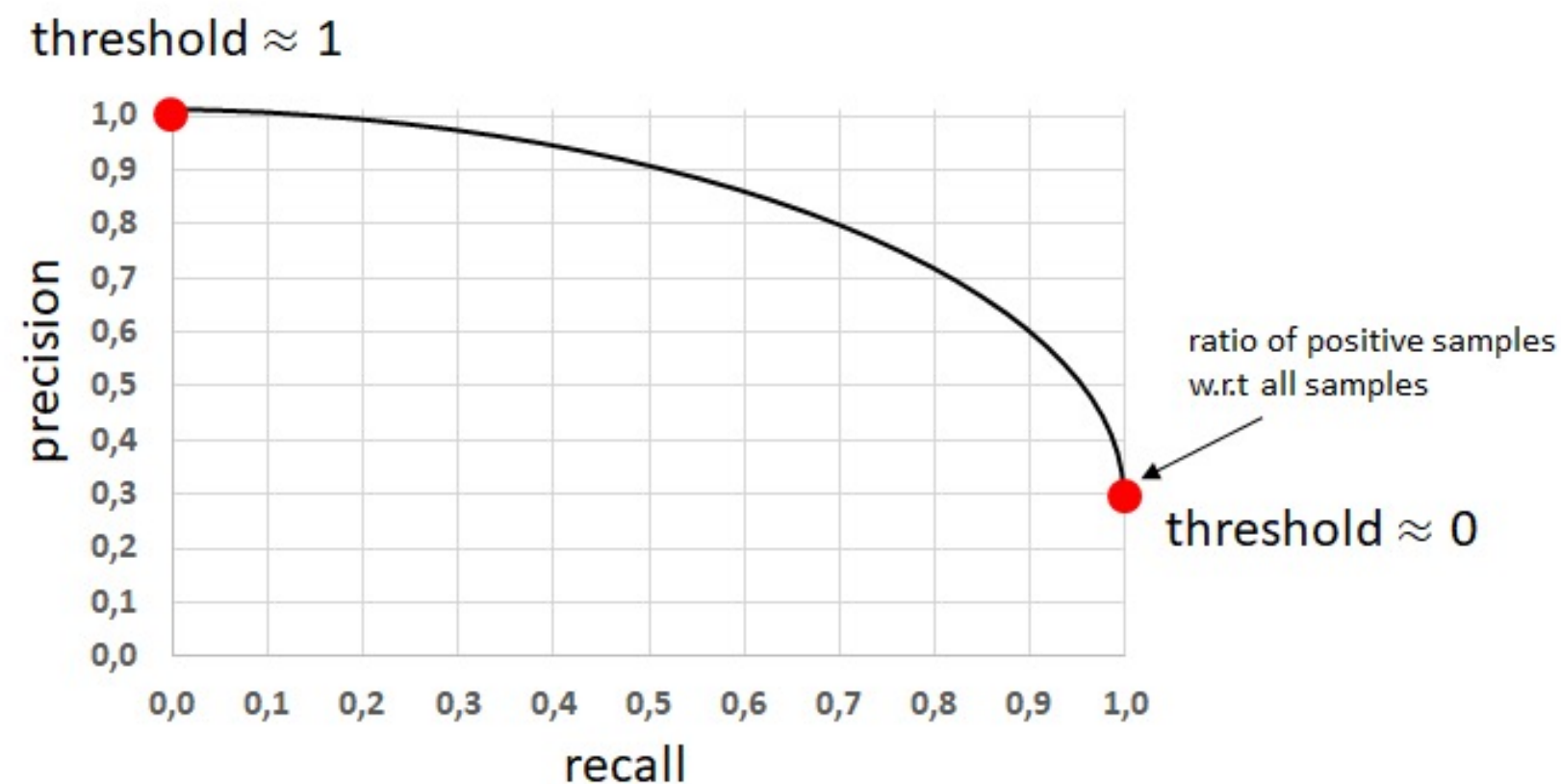
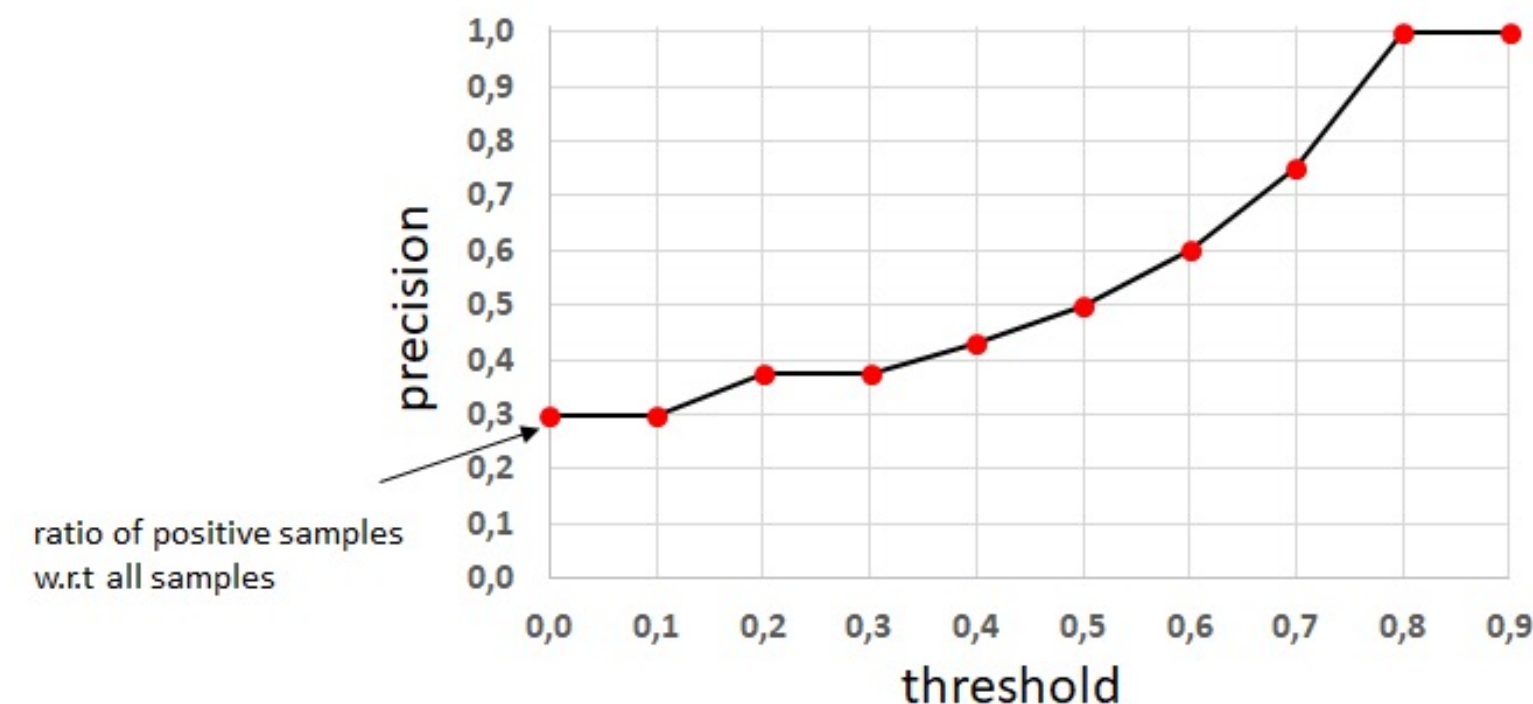
$FPR = 0$

случайные
предсказания

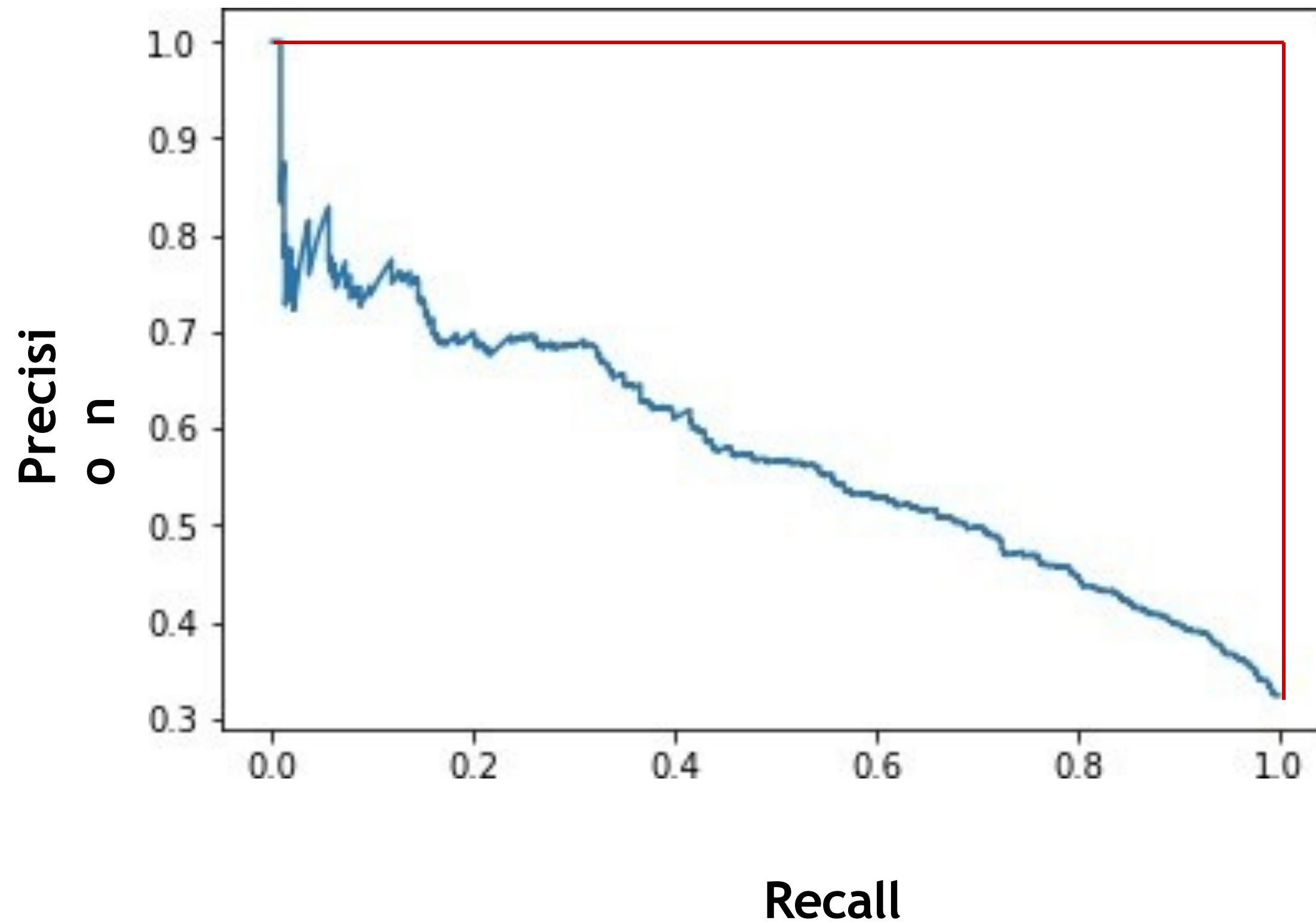


Precision Recall Area under the curve PR AUC

Кривая PR показывает нам взаимосвязь между показателем *Precision* (точность) и Recall (полнота) для разных *пороговых значений*.



Кривая Precision – Recall



Модель тем лучше, чем
выше площадь подкривой

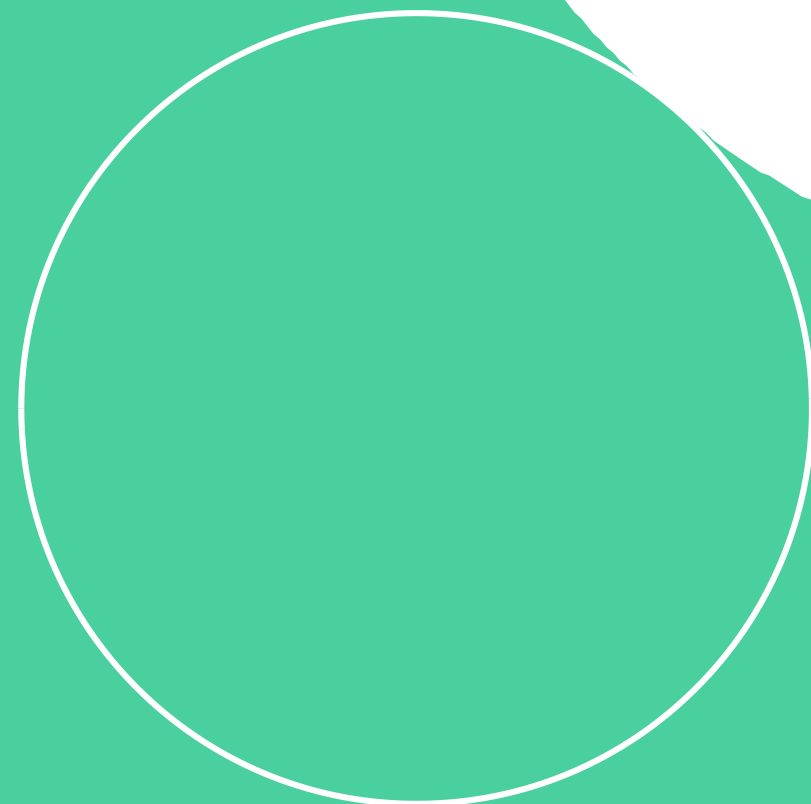


Практика

LOGRES_AFFAIR.IPYNB

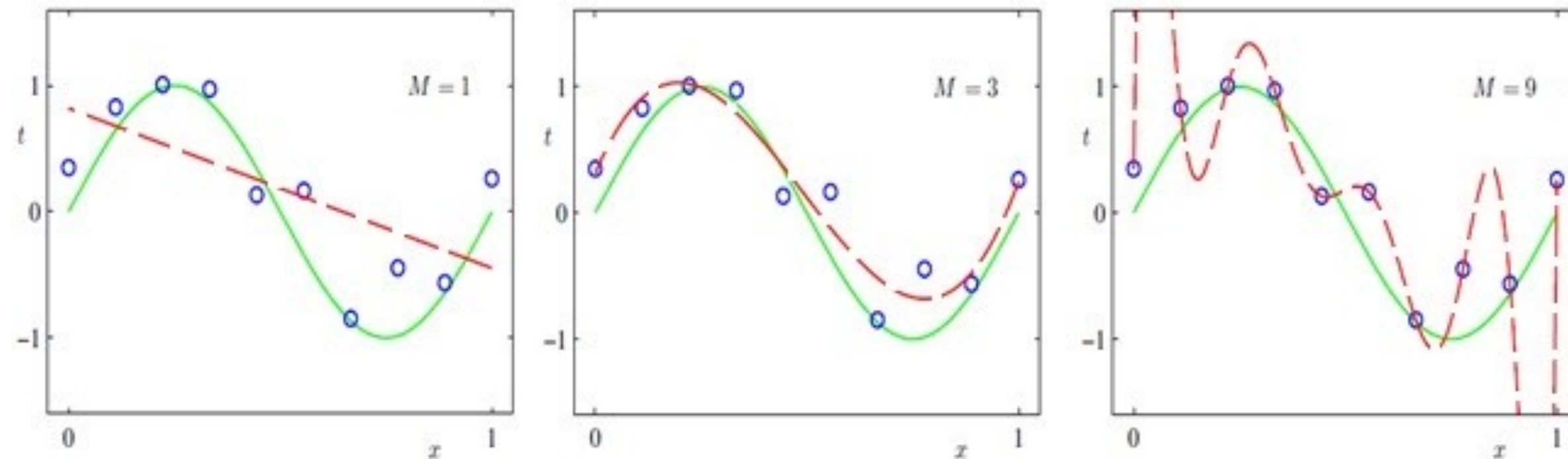


Борьба с переобучением

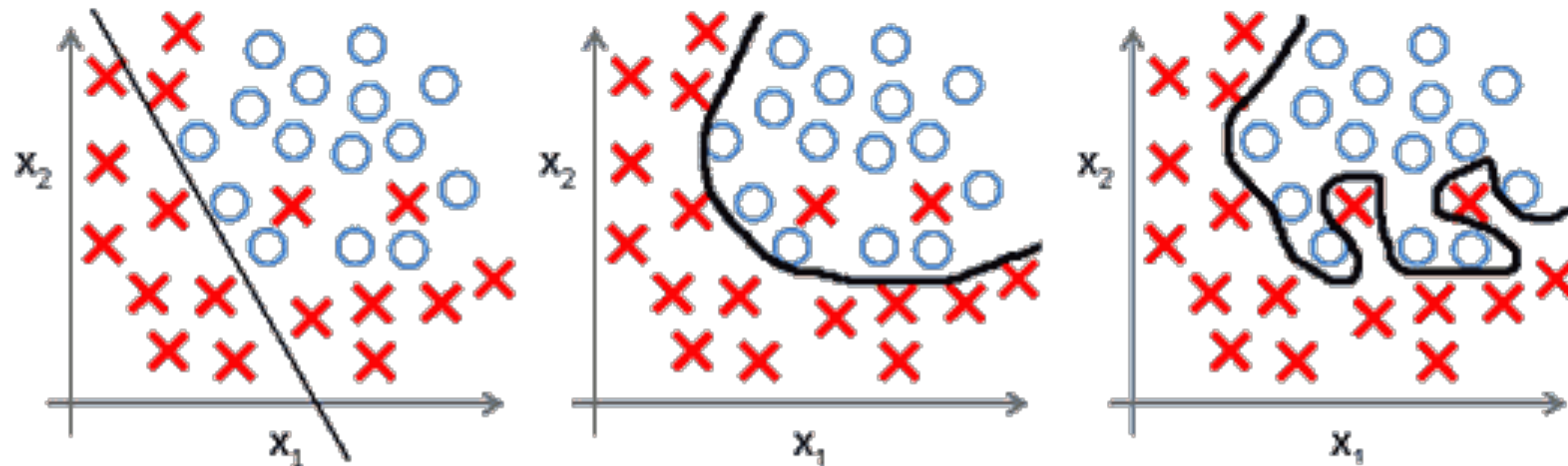


Переобучение и недообучение

Регрессия

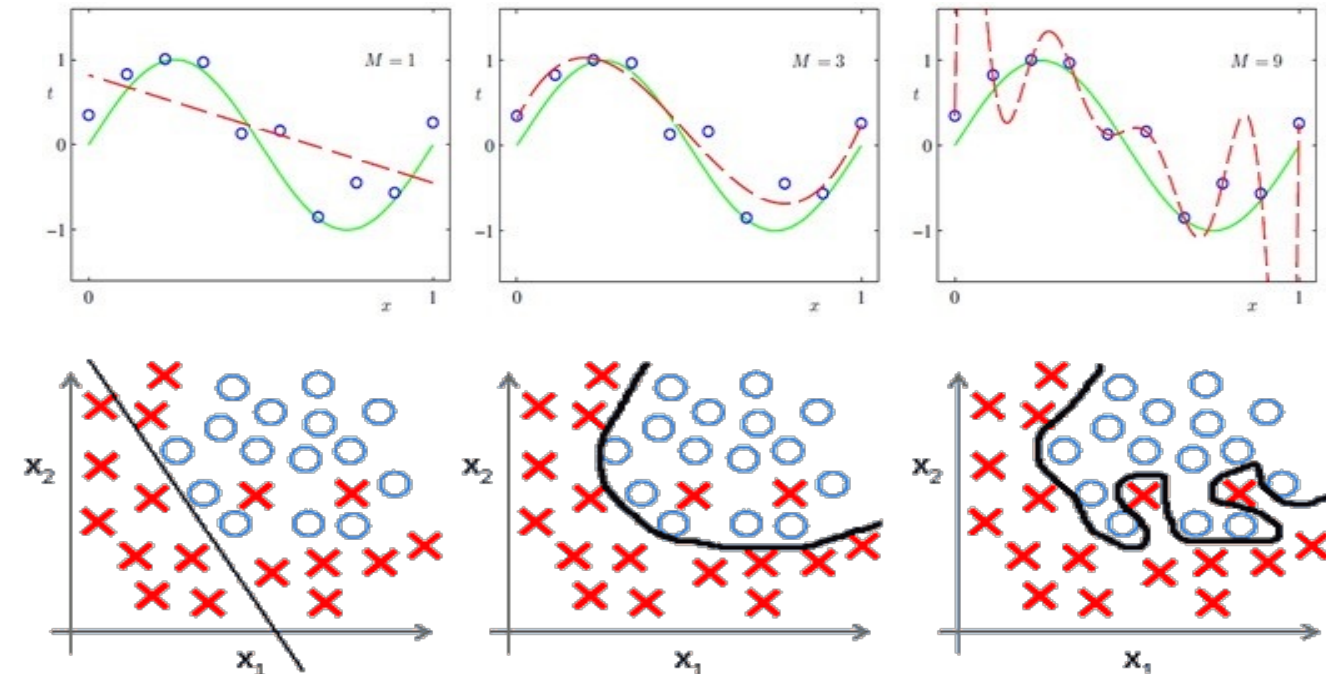


Классификация



Переобучение и недообучение

Переобучение (overfitting) – явление, когда ошибка на тестовой выборке заметно больше ошибки на обучающей.



Недообучение (underfitting) – явление, когда ошибка на обучающей выборке достаточно большая, часто говорят «не удаётся настроиться на выборку».

Сложность (complexity) модели алгоритмов – оценивает, насколько разнообразно семейство алгоритмов в модели с точки зрения их функциональных свойств (например, способности настраиваться на выборки). Повышение сложности (т.е. использование более сложных моделей) решает проблему недообучения и вызывает переобучение.

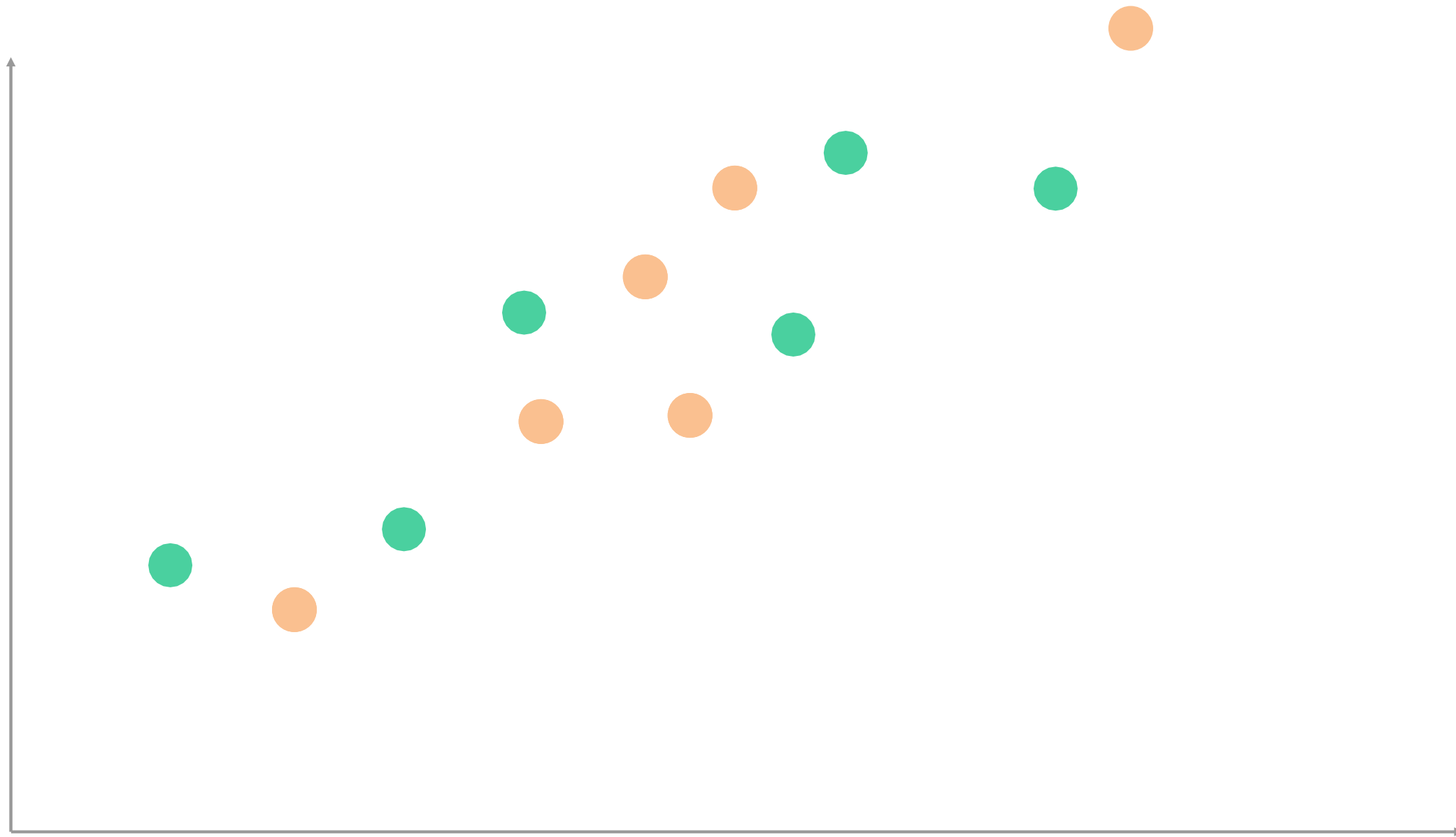


Переобучение и недообучение



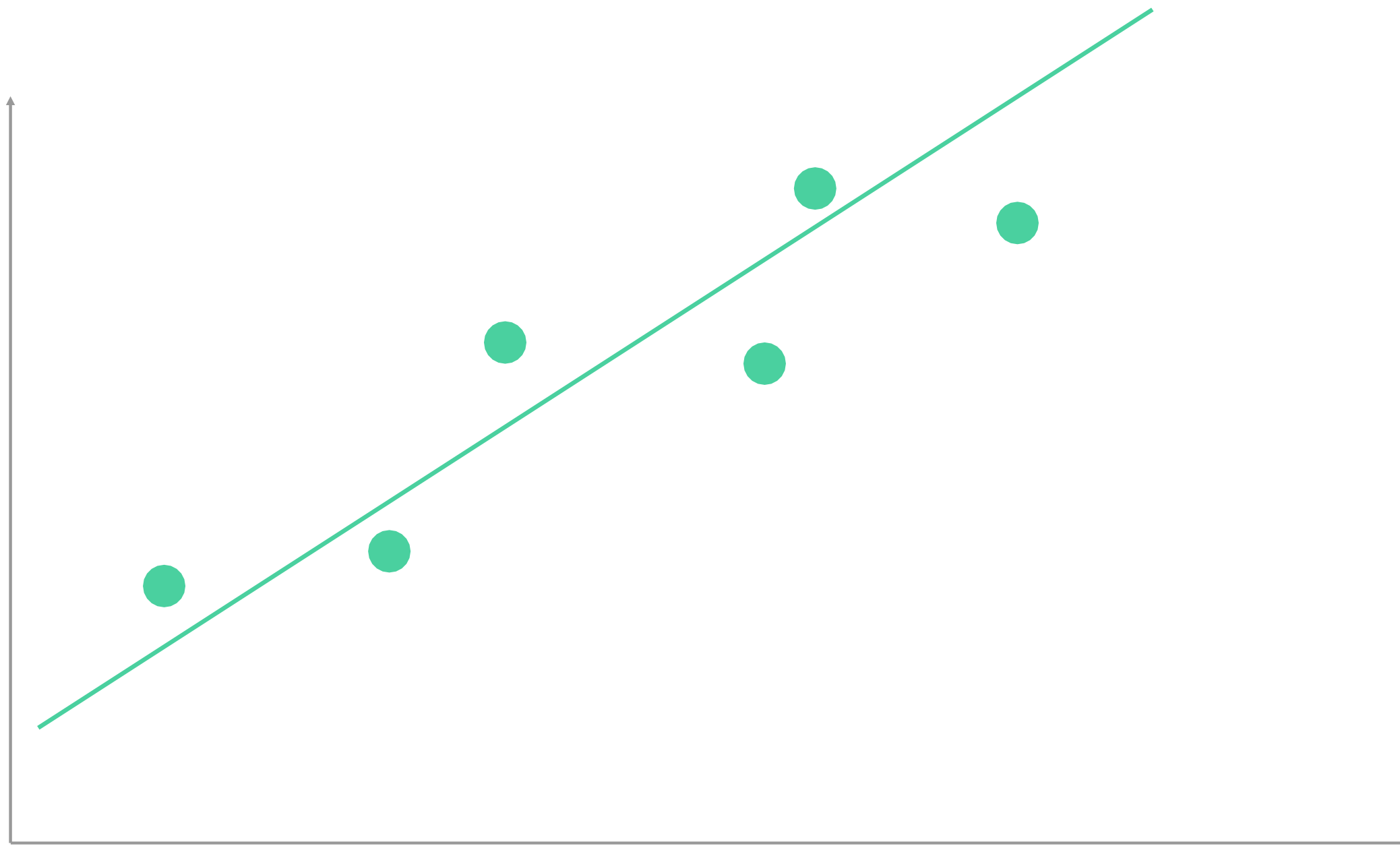
Пример переобучения

Имеются данные из 6 точек и 6 точек новых данных



Пример переобучения

Строим простую модель



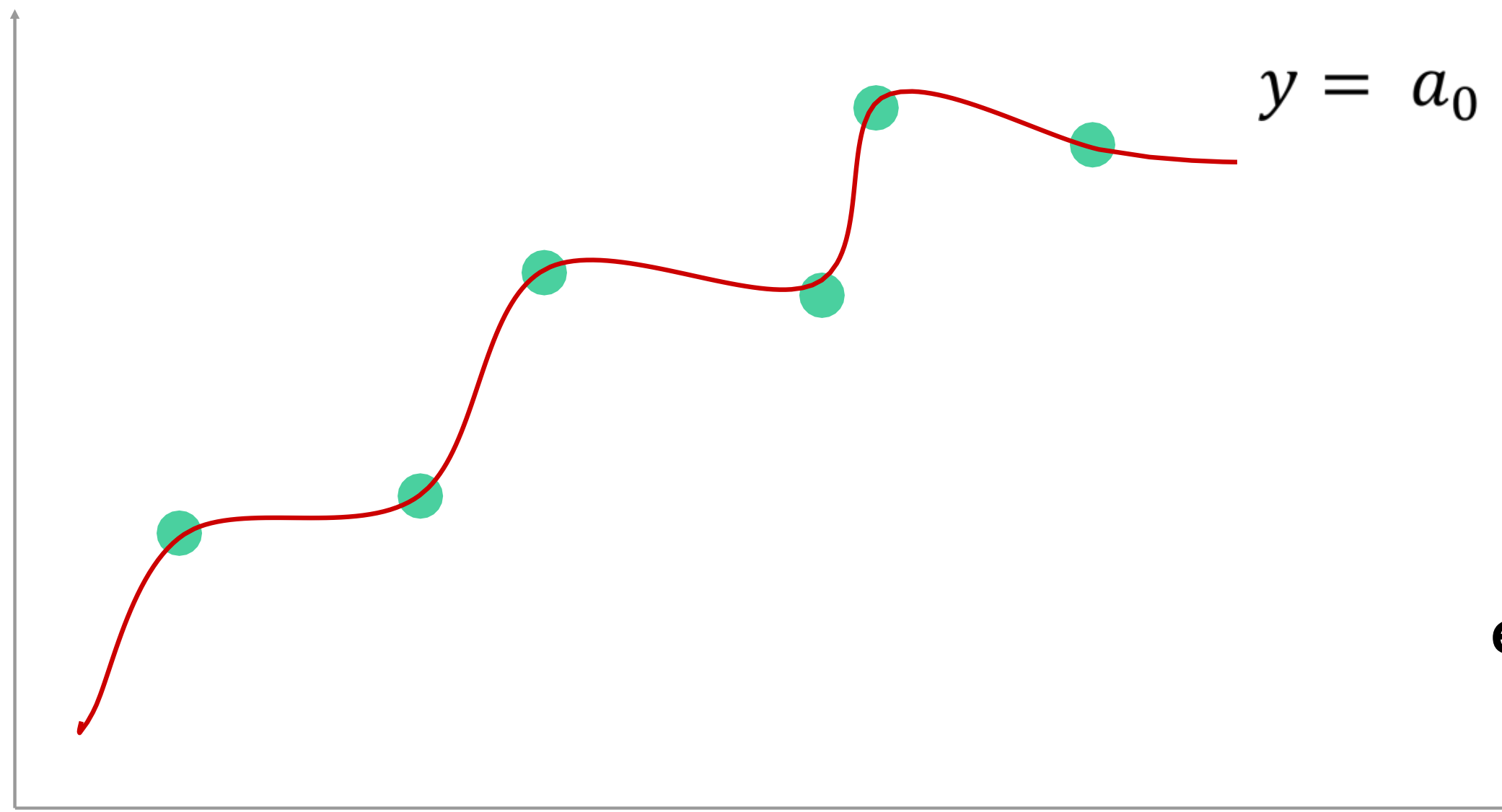
— $y = kx + b;$
ошибка > 0

e_1 - ошибка на новых данных > 0



Пример переобучения

Строим сложную модель



$$y = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5$$

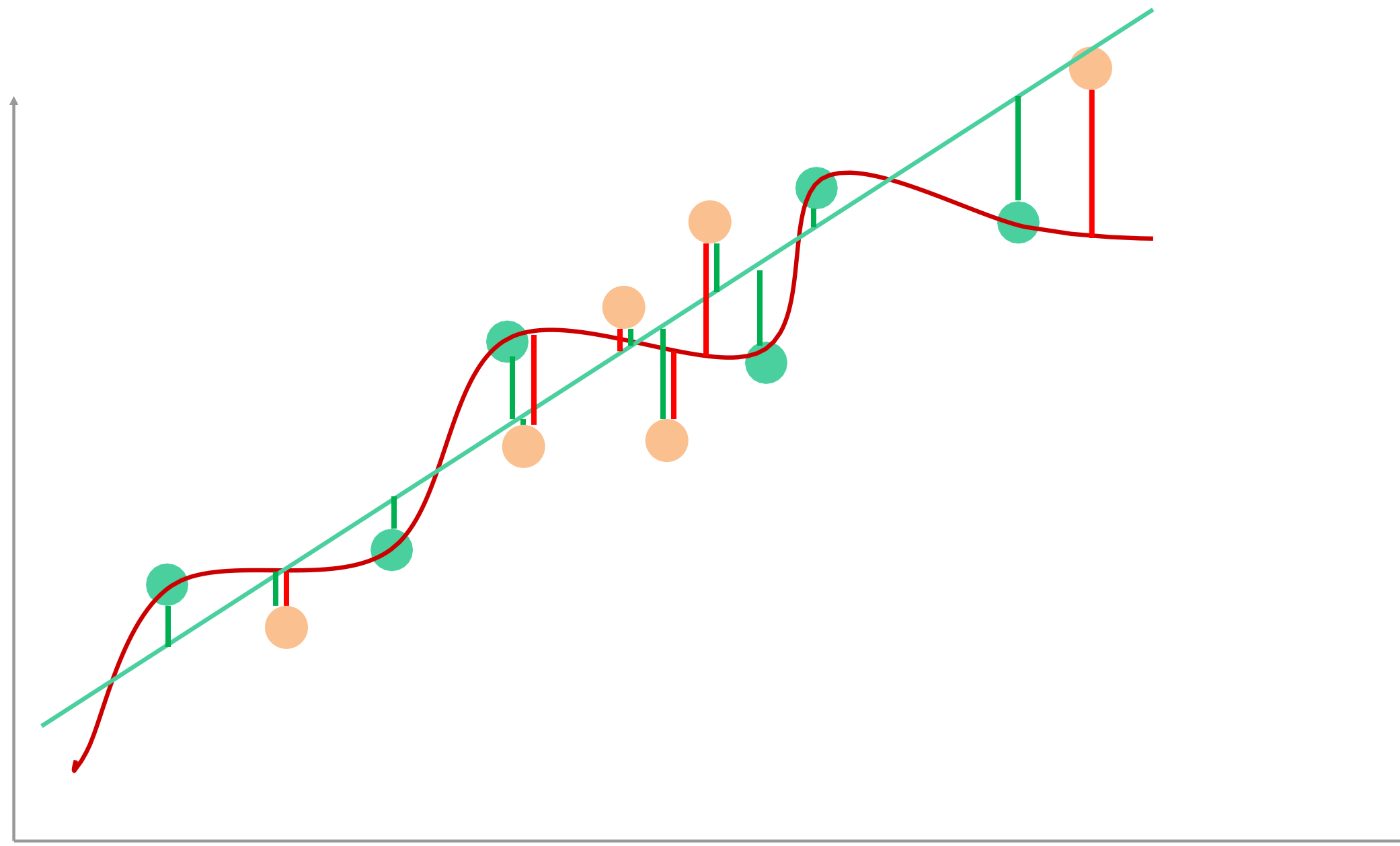
ошибка = 0. Хорошо?

e_2 - ошибка на новых данных > 0



Пример переобучения

На тестовых данных получаем большую ошибку



e_1 - ошибка на новых данных > 0

e_2 - ошибка на новых данных > 0

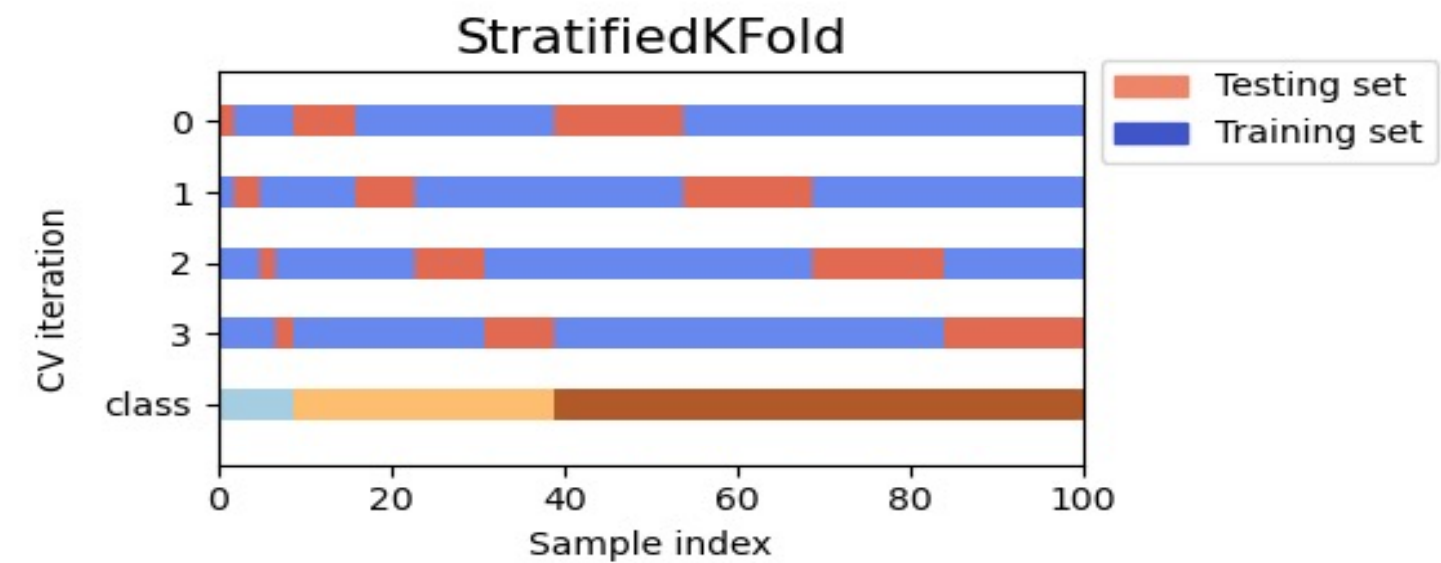
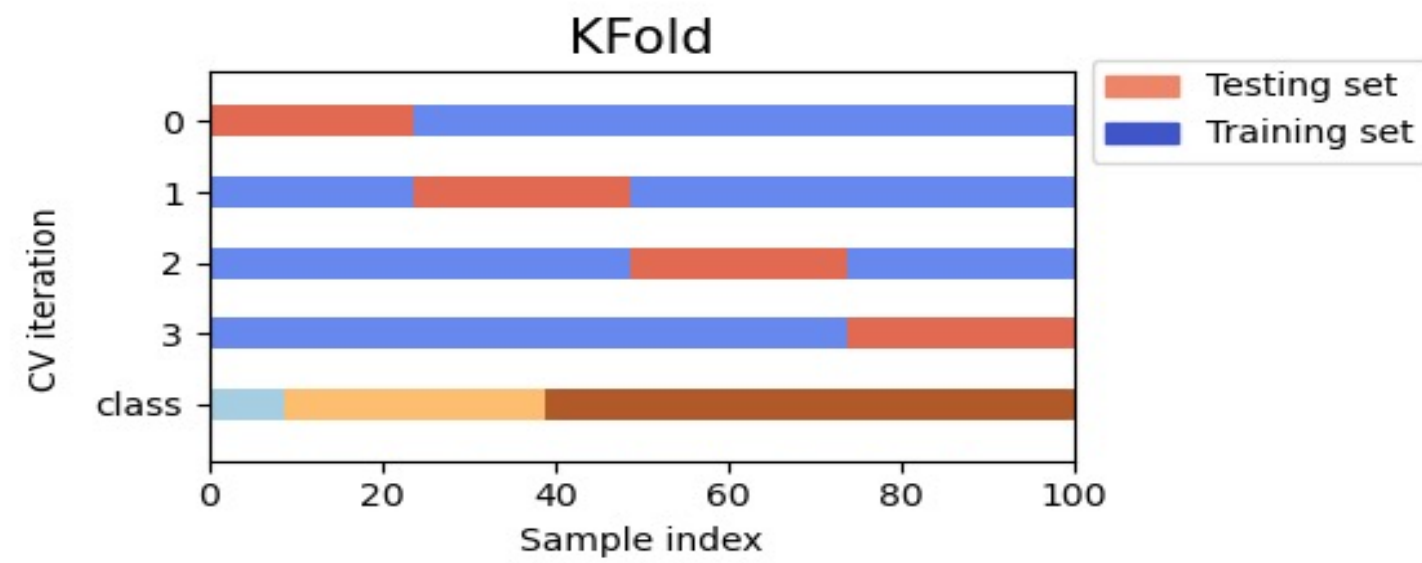
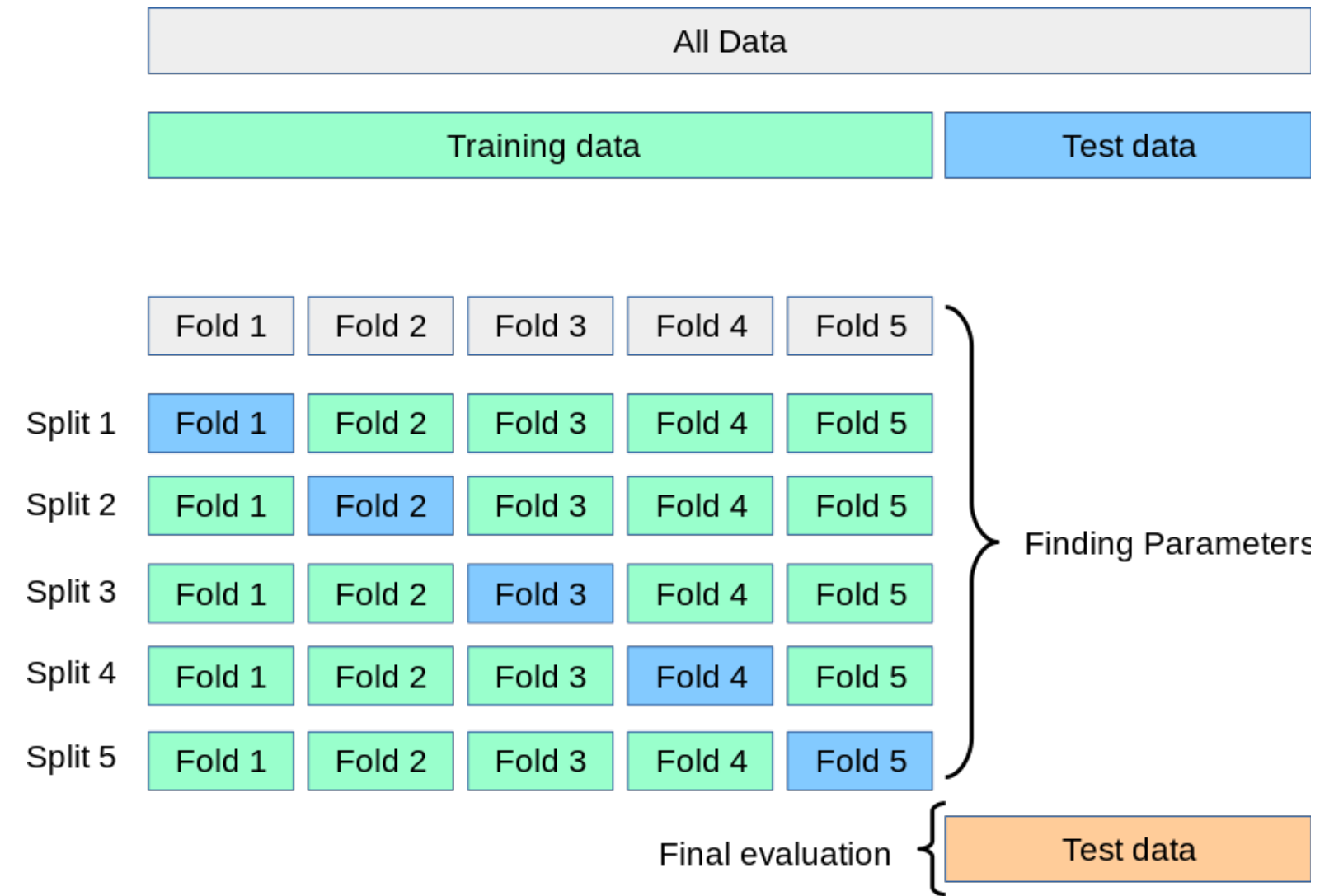
$e_2 > e_1$



Кросс-валидация

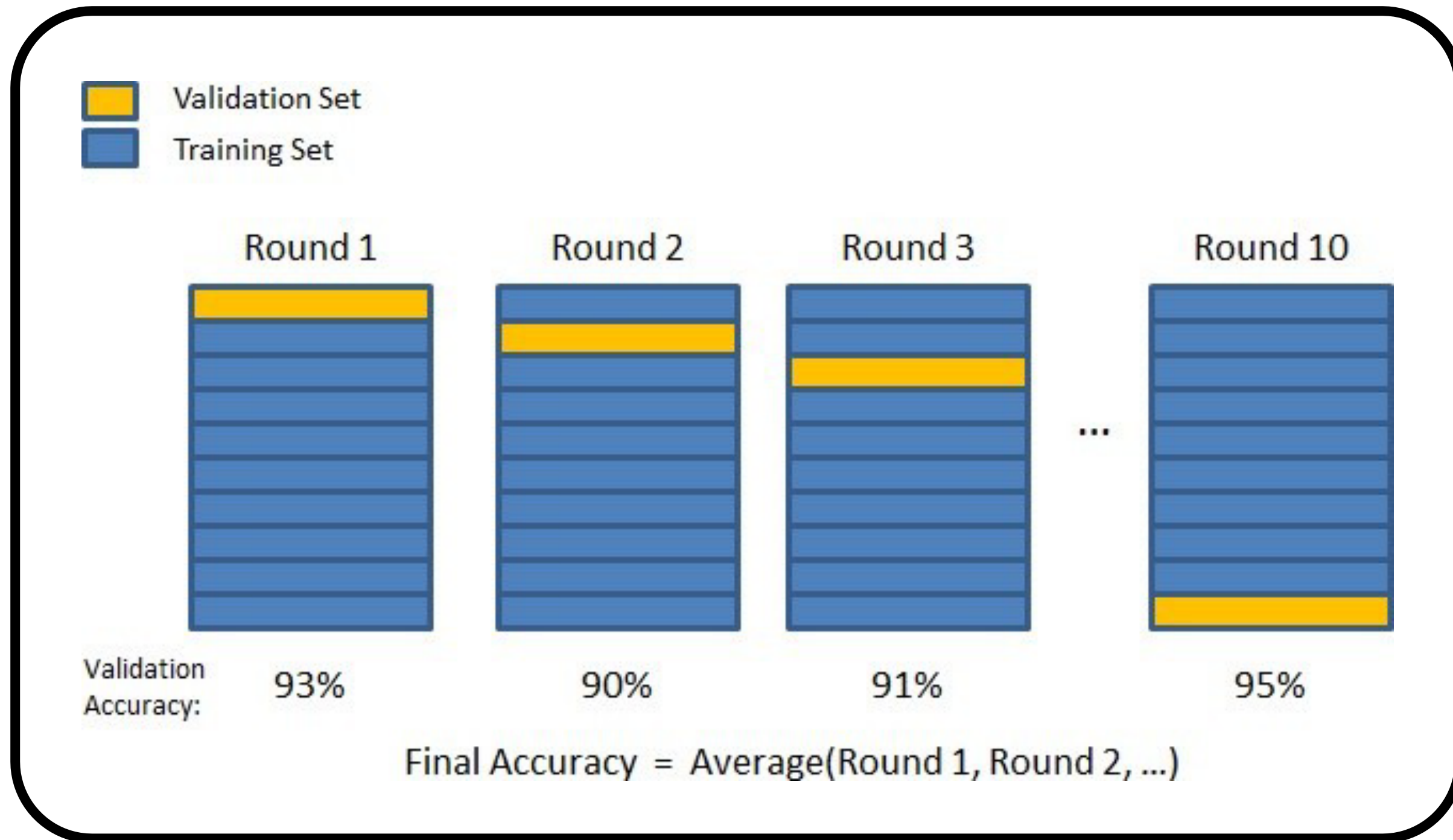
k-fold cross validation

Кросс-валидация или скользящий контроль — процедура эмпирического оценивания обобщающей способности алгоритма.



Кросс-валидация

k-fold cross validation

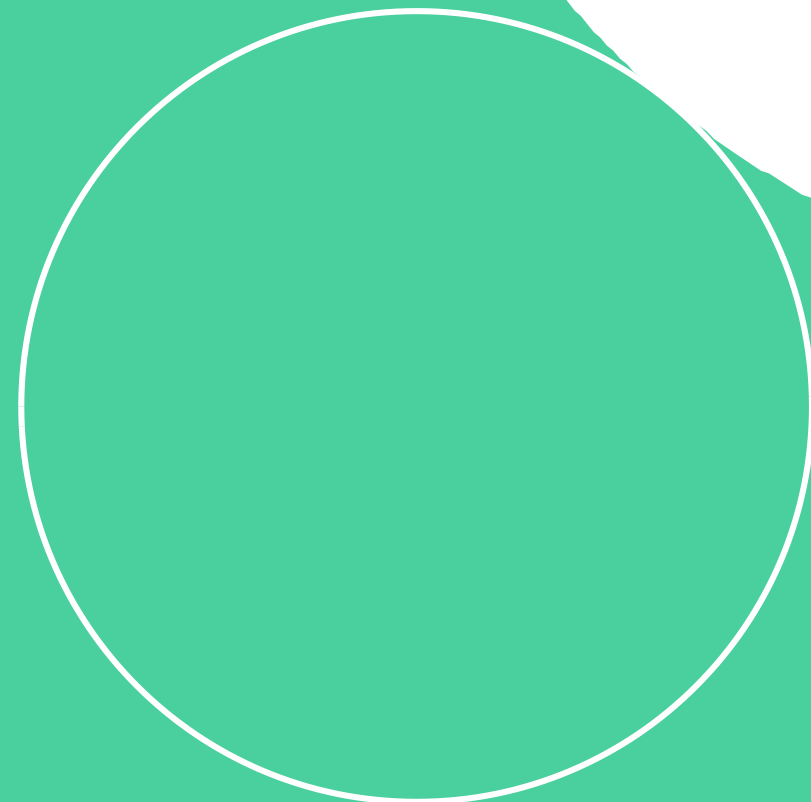


Практика

LOGRES_AFFAIR.IPYNB



Смещение и разброс



Ошибка

прогноза

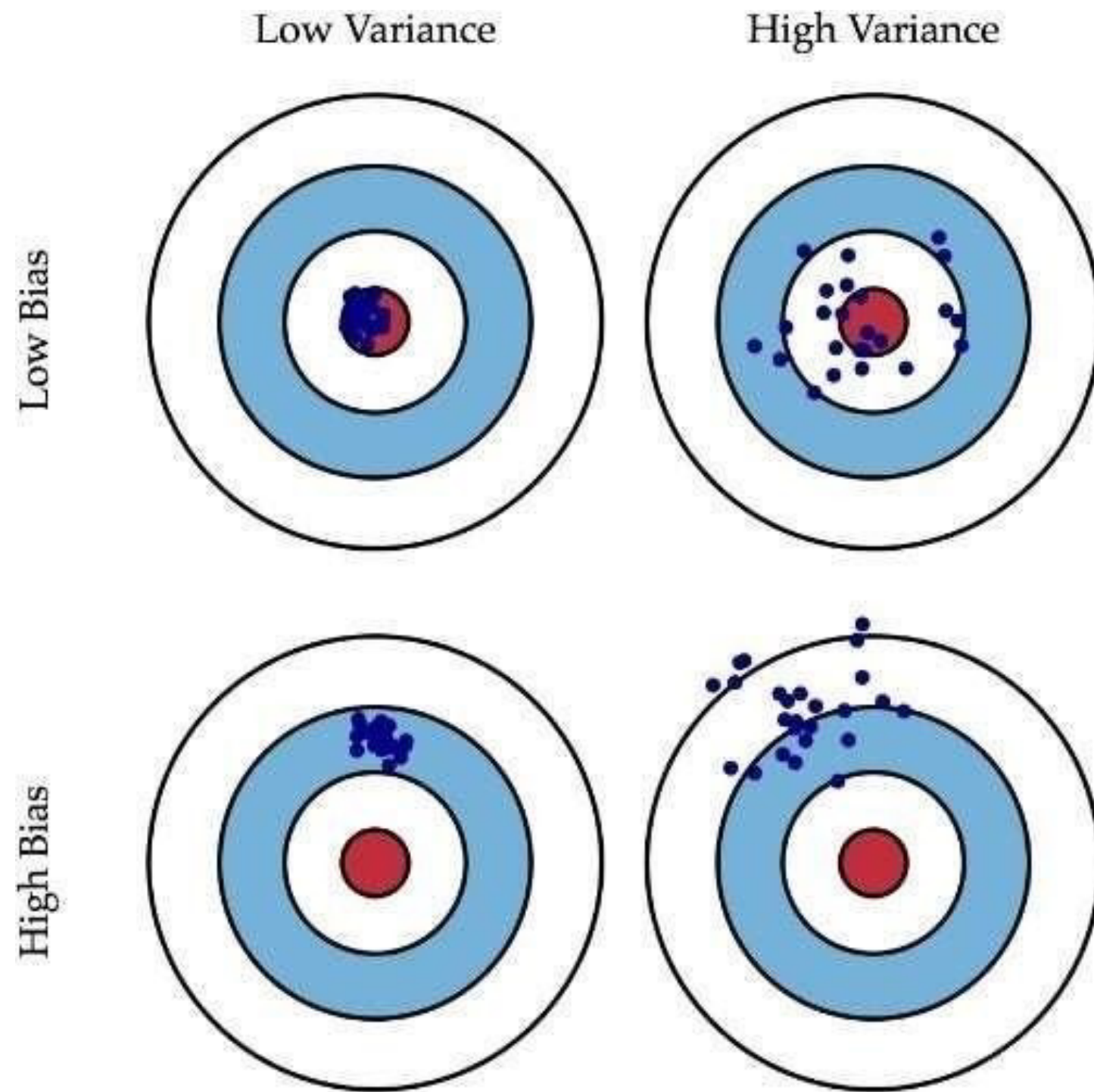
Ошибка можем разложить на слагаемые:

- Bias – средняя ошибка прогноза. Характеризует способность модели алгоритмов настраиваться на целевую зависимость.
- Variance – изменение ошибки при обучении на разных наборах данных. Характеризует разнообразие алгоритмов, которые могут быть реализованы моделью данного типа.
- Неустраняемая ошибка

<https://habrahabr.ru/company/ods/blog/323890/#razlozhenie-oshibki-na-smeschenie-i-razbros-bias-variance-decomposition>



Ошибка прогноза

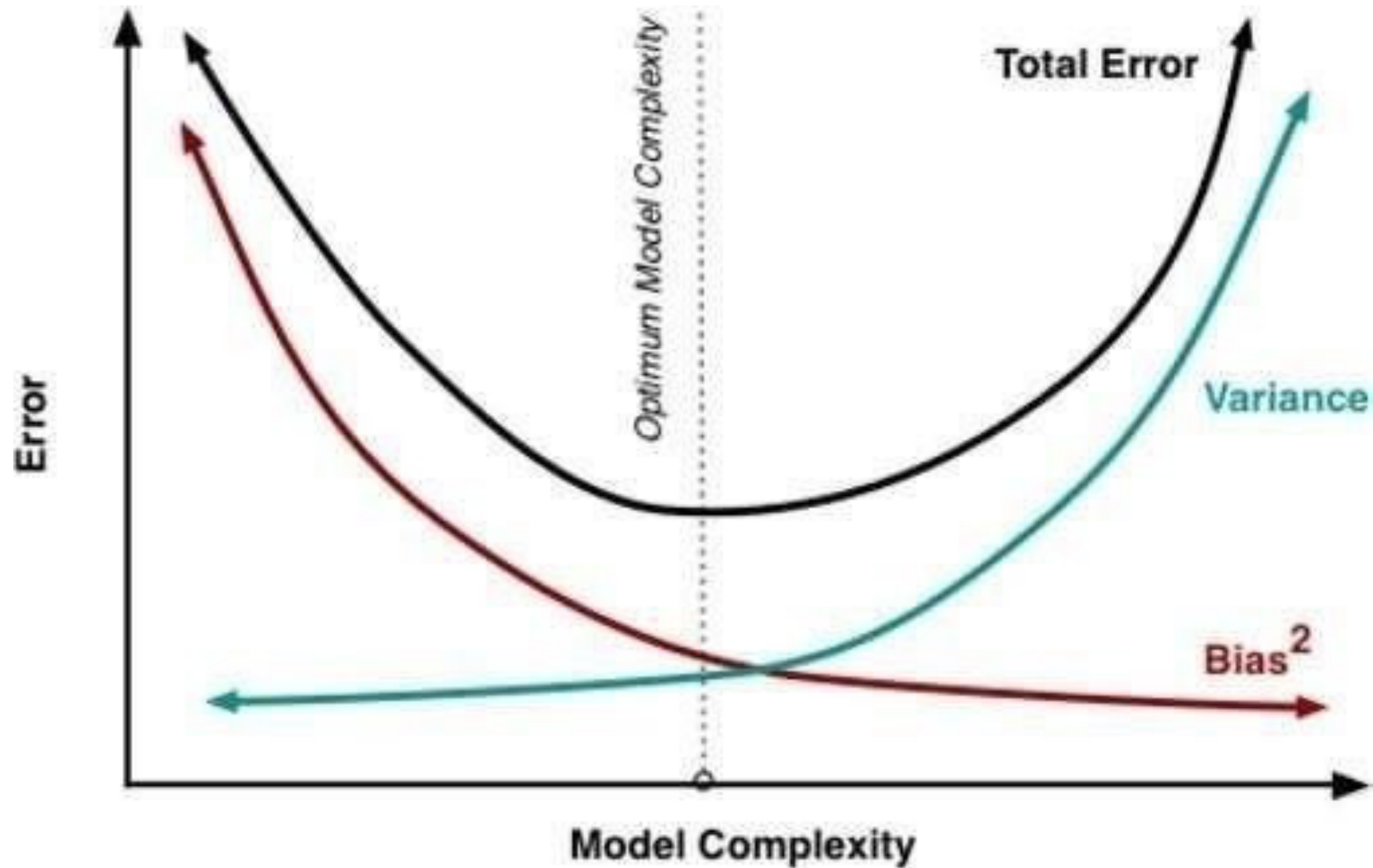


Сложная модель (учитывает много признаков) – увеличивает разброс ошибки

Слишком простая модель (мало признаков) – вызывает смещение в пользу одного признака



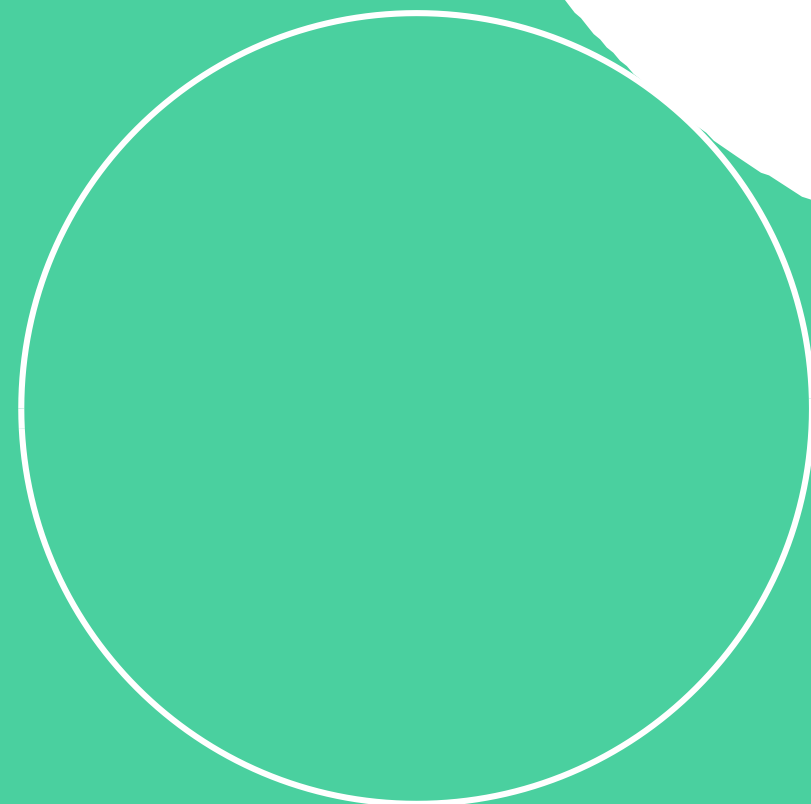
Оптимальный вариант



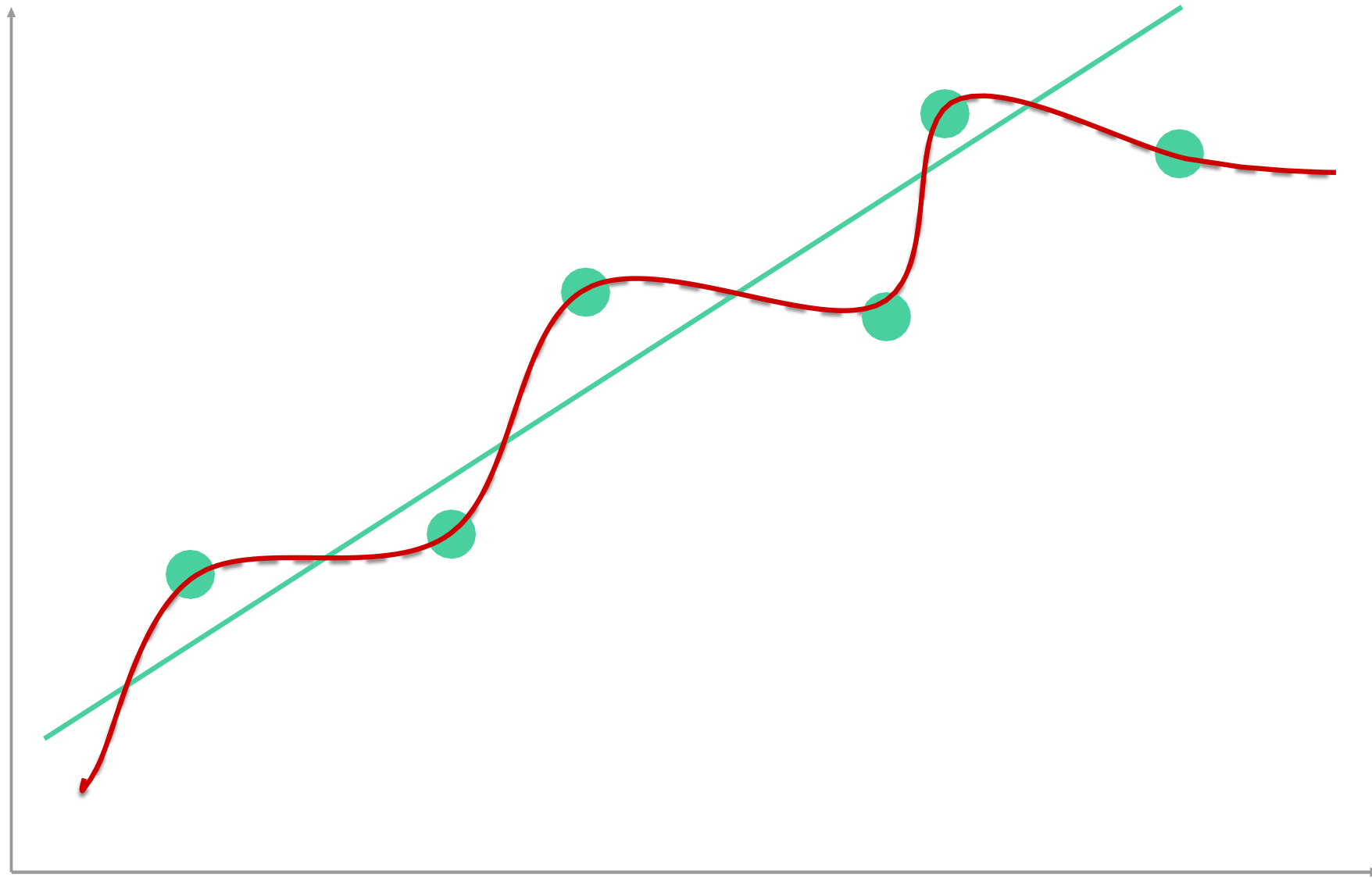
Можно ли повлиять на стабильность модели, т.е. уменьшить Variance?



L1 и L2 регуляризация



Прошлый пример переобучения



Переберем модели,
увеличивая степень
функции

$$y = a_0 + a_1x$$

$$y = a_0 + a_1x + a_2x^2$$

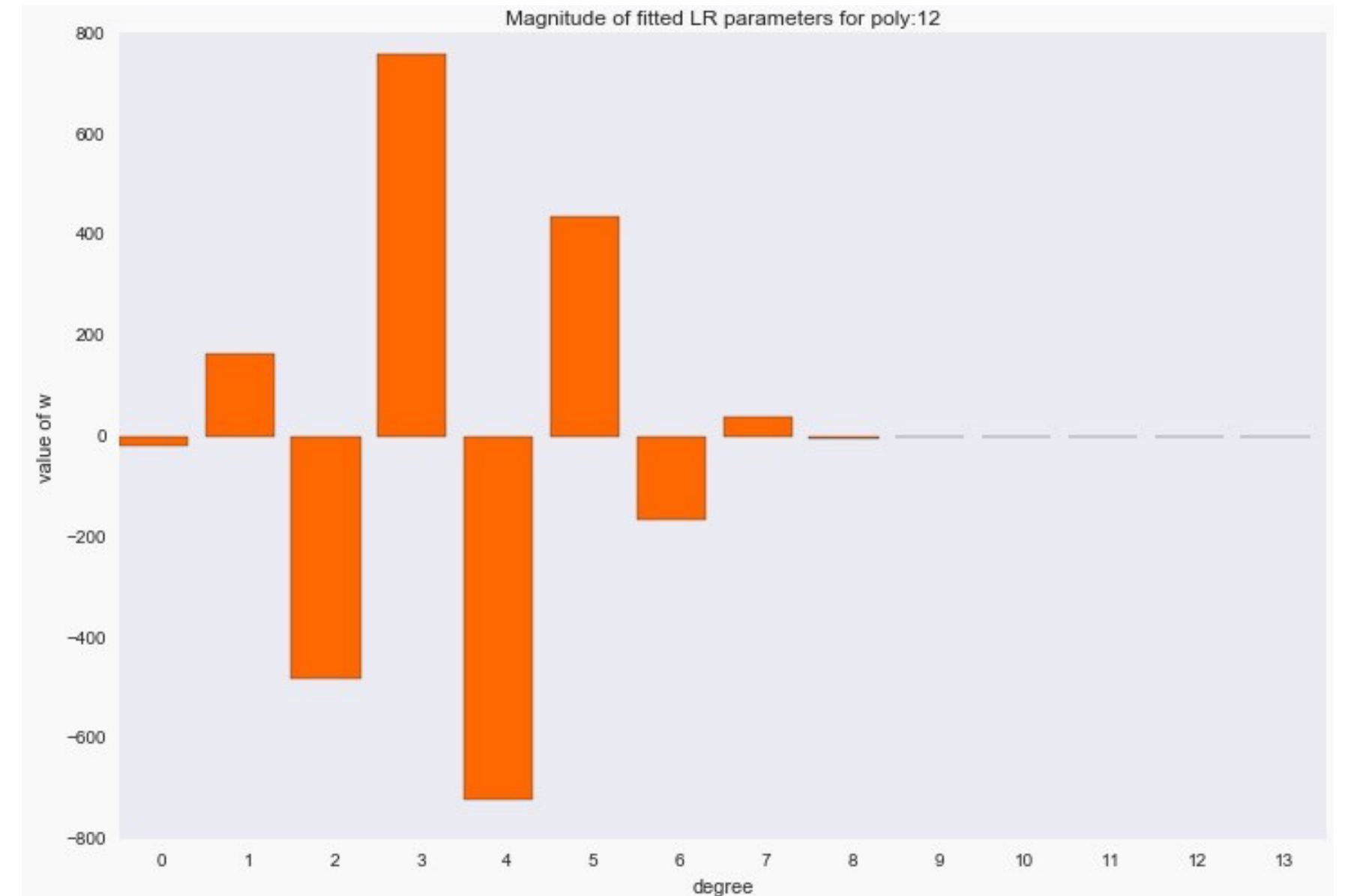
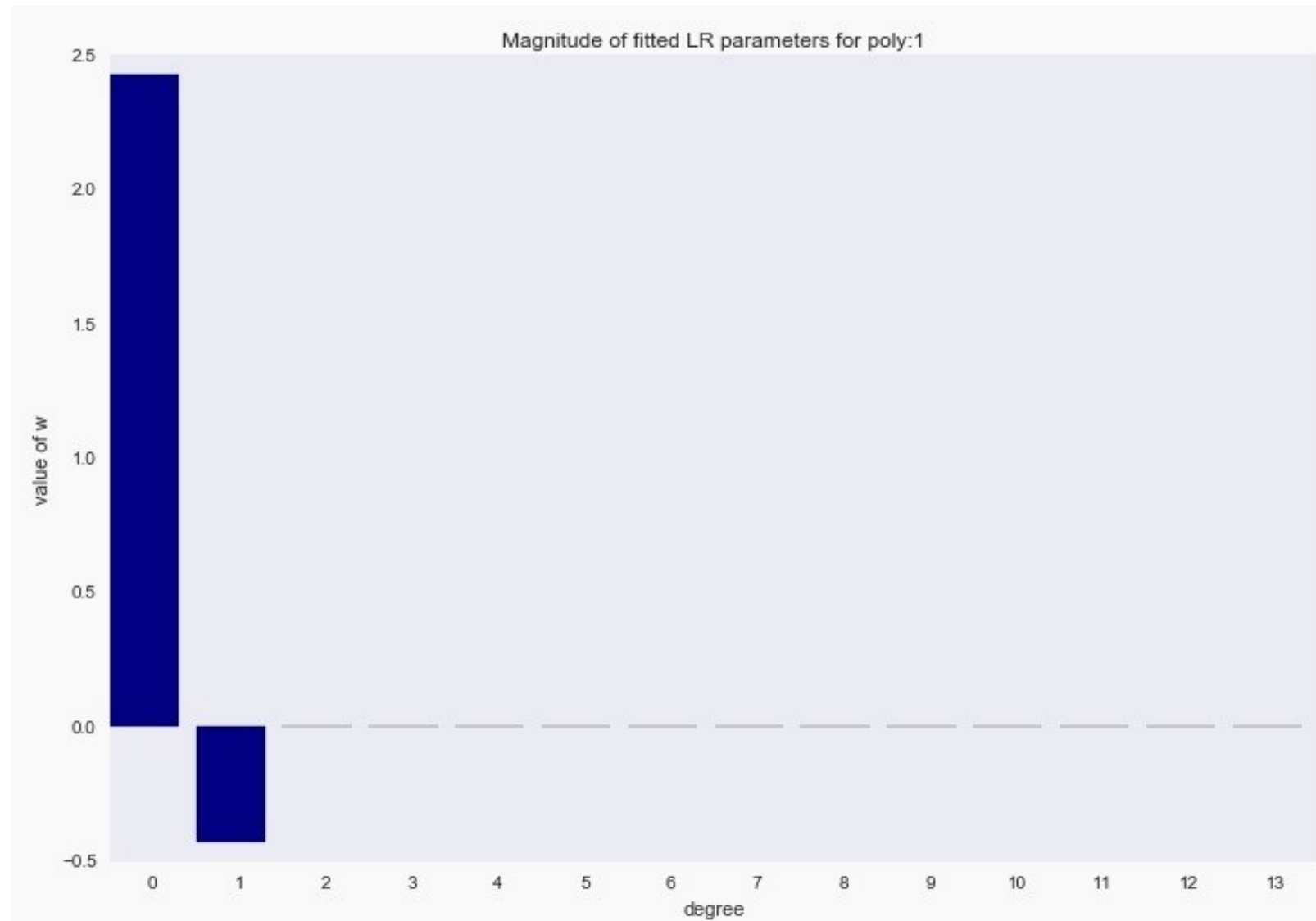
$$y = a_0 + a_1x + a_2x^2 + a_3x^3$$

$$y = a_0 + a_1x + a_2x^2 + \dots + a_5x^5$$



Как будут варьироваться?

При увеличении степени полинома
вариация коэффициентов быстро растёт



Корреляция признаков

Рост коэффициентов от корреляции между признаками

Имеем линейную модель в которой есть коррелированные переменные x_1 x_2

$$\dots + w_1 \cdot x_1 + w_2 \cdot x_2 + \dots$$

$$k \cdot x_1 = x_2$$

Тогда одну переменную можно выразить через другую и коэффициент c может быть любым

$$\begin{aligned} & \dots + w_1 \cdot x_1 + w_2 \cdot x_2 + \dots = \\ & = \dots + (w_1 + c \cdot k) \cdot x_1 + (w_2 - c) \cdot x_2 + \dots \end{aligned}$$



Надо уменьшить разброс коэффициентов

Имеем модель целевой переменной y и коэффициентами

$$\text{Целевая функция} = \sum_i (y_{\text{факт}} - Xa)^2$$



Штраф за сложность

Основные варианты регуляризации

$$L_1 = \sum_i (y_{\text{факт}} - Xa)^2 + \lambda \sum_i |a_i|$$

$$L_2 = \sum_i (y_{\text{факт}} - Xa)^2 + \lambda \sum_i a_i^2$$



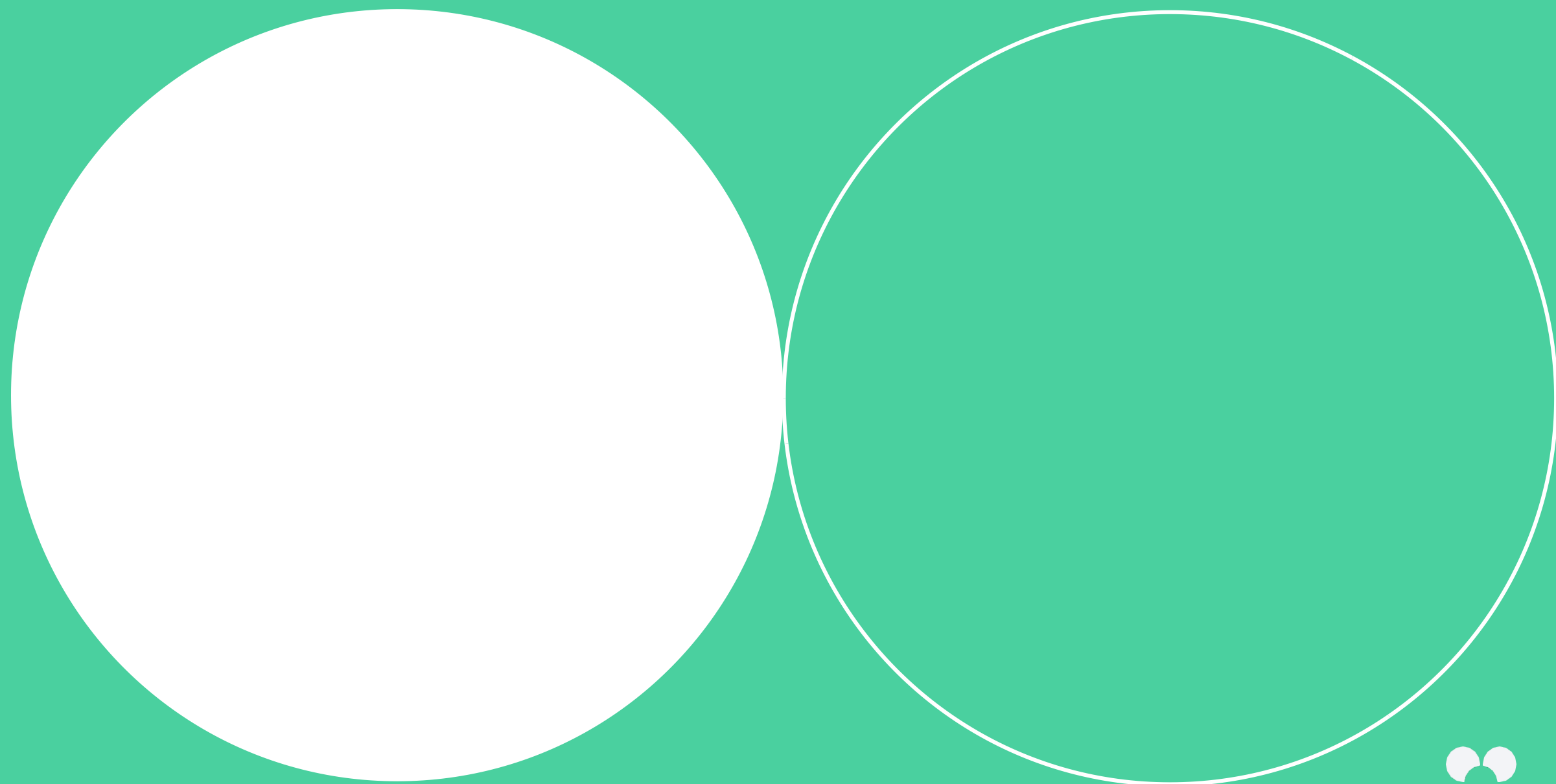
Практика

LOGRES_AFFAIR.IPYNB

регулізация.ipynb



Что мы сегодня узнали

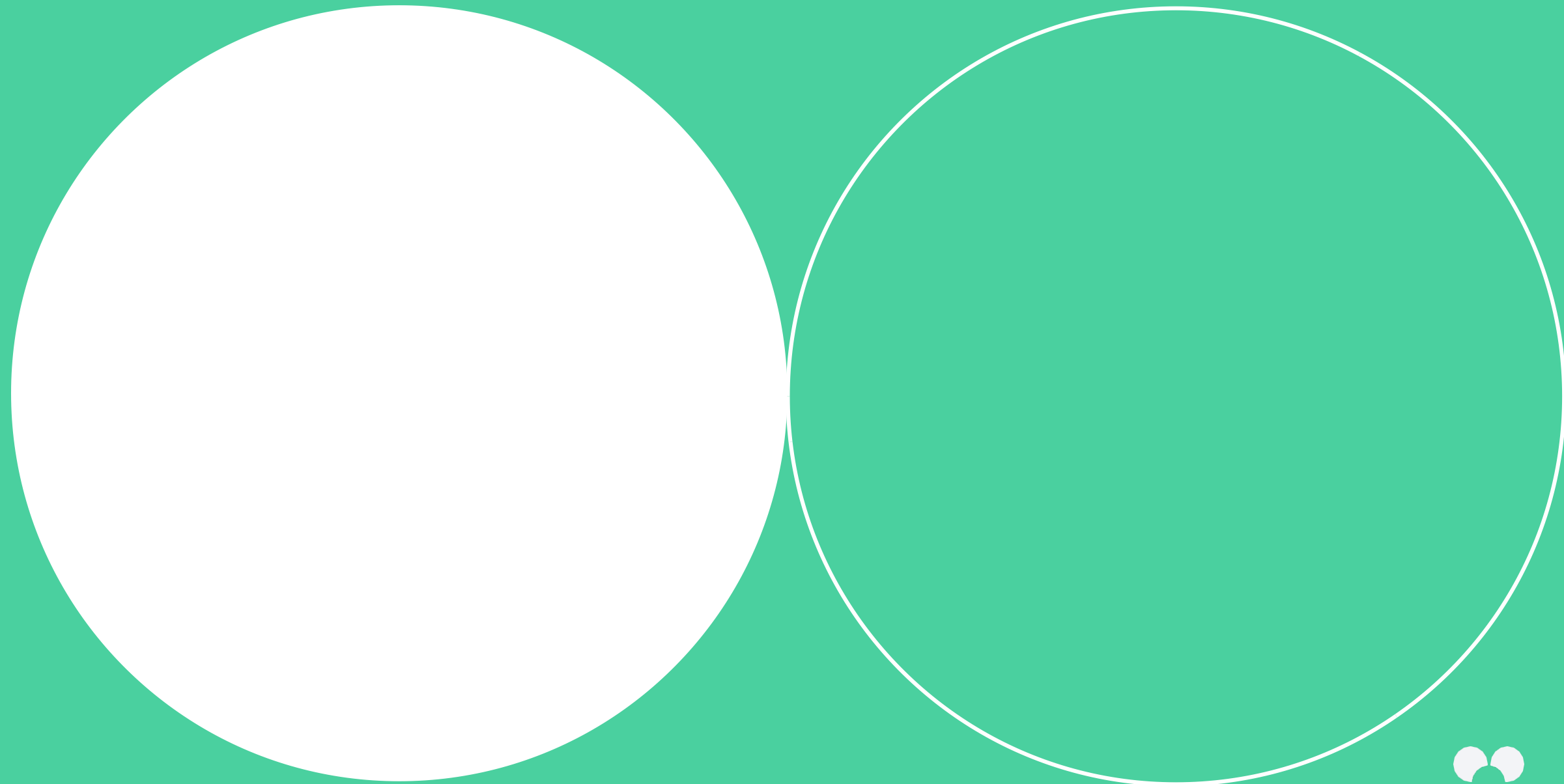


Что мы сегодня узнали

1. Изучили метрики оценки качества моделей.
2. На практике потренировались в проведении кросс-валидации моделей.
3. Изучили признаки и способы борьбы с переобучением на примере L1 и L2 регуляризации.



Полезные материалы



Полезные материалы

1. Наглядные примеры переобучения модели и теоретические выкладки регуляризации

<https://habrahabr.ru/company/ods/blog/322076/>

1. О разнице между L1 и L2 регуляризацией

<http://www.chioka.in/differences-between-l1-and-l2-as-loss-function-and-regularization/>

2. Более сложный пример регуляризации

<https://habrahabr.ru/company/ods/blog/323890/#3-naglyadnyy-primer-regulyarizacii-logisticheskoy-regressii>



Спасибо за внимание!

