

# Машинное обучение

Лекция 2

Метрики, валидация. Простейшие методы

*Власов Кирилл Вячеславович*



2018

# Формальная постановка задачи

Дана обучающая выборка (объекты независимы):

$$X_m = \{ (x_1, y_1), \dots, (x_m, y_m) \}$$

Для задачи регрессии - Целевая переменная задана вещественным числом

$$(x_1, y_1) \in \mathbb{R}^m \times \mathbb{Y}, \mathbb{Y} = \mathbb{R}$$

Для задачи классификации - Целевая переменная задана конечным числом меток

$$(x_1, y_1) \in \mathbb{R}^m \times \mathbb{Y}, \mathbb{Y} = \{-1; 1\}$$

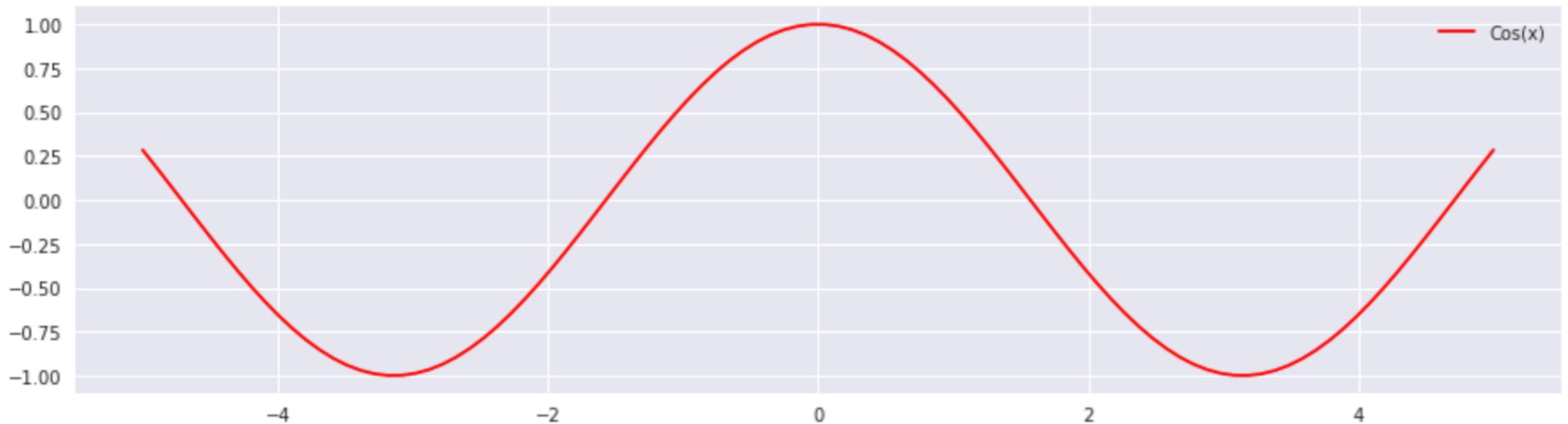
Задать такую функцию  $f(x)$  от вектора признаков  $x$ , которое выдает ответ для любого возможного наблюдения  $x$

$$f(x): \mathbb{X} \rightarrow \mathbb{Y}$$

**Основная гипотеза МО:** Схожим объектам соответствуют схожие объекты

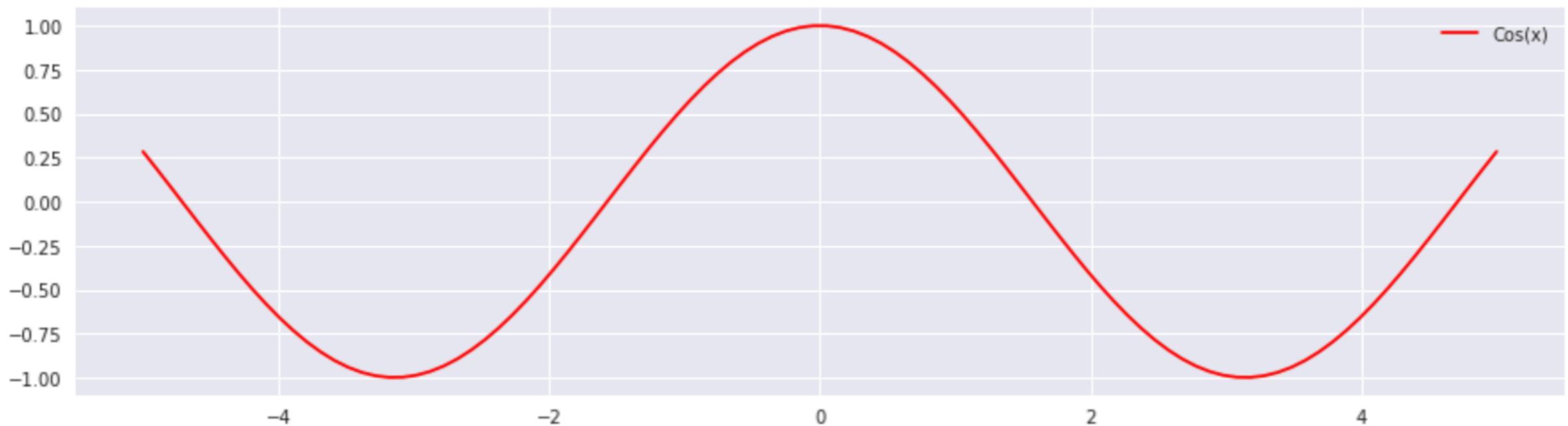
# Формальная постановка задачи

$$y = \cos(x), x \in [-5, 5]$$

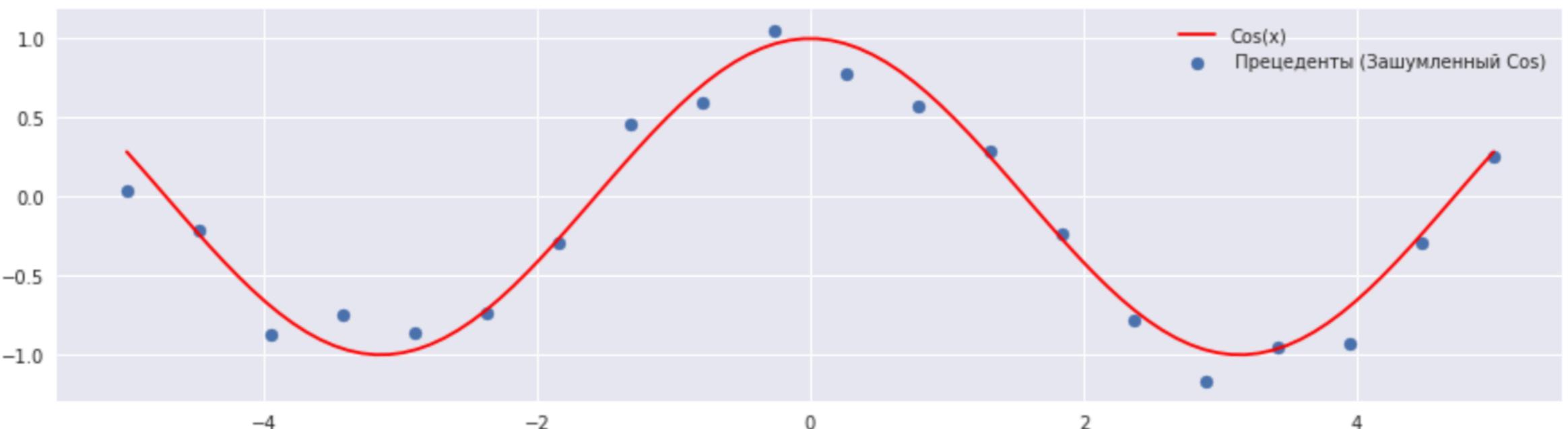


# Метрики качества

$$y = \cos(x), x \in [-5, 5]$$

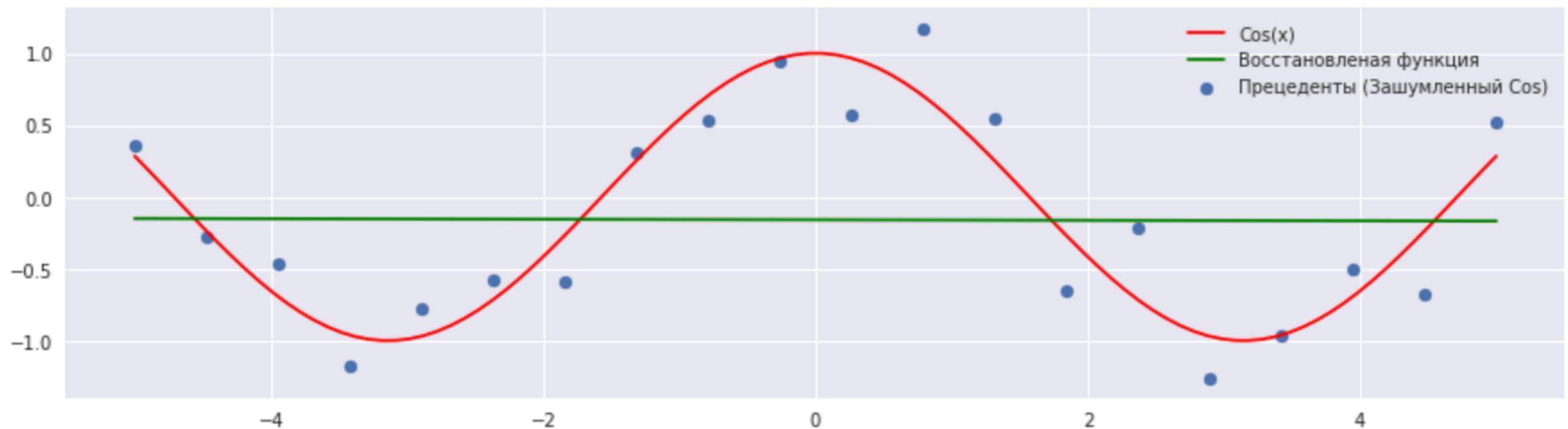


$$y = \cos(x) + \varepsilon, \text{ где } \varepsilon = \mathcal{N}\left(0, \frac{1}{2}\right), x \in [-5, 5]$$



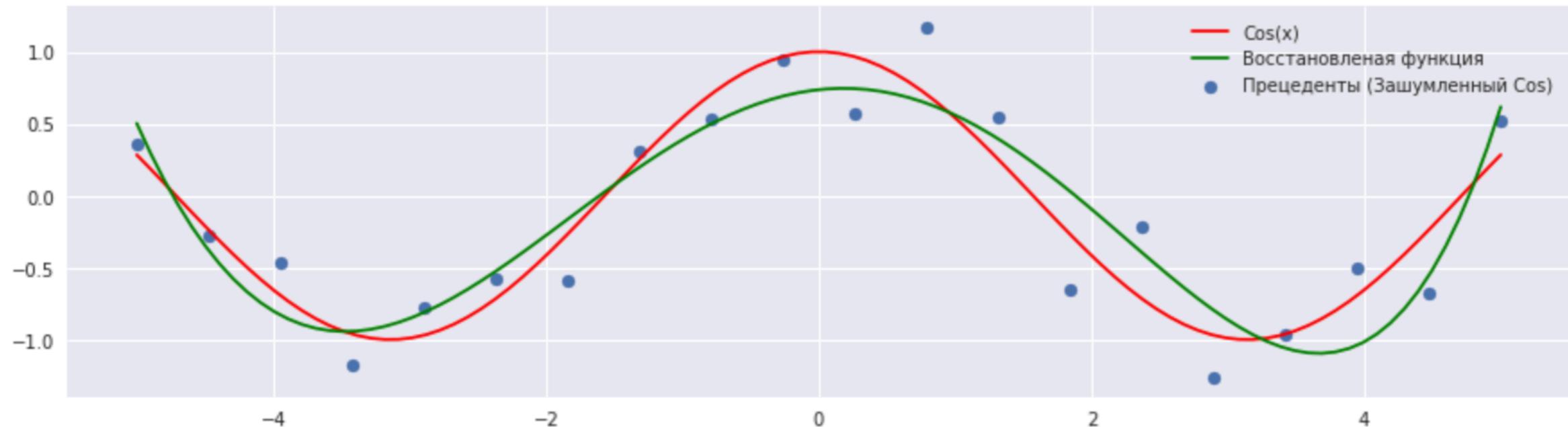
# Метрики качества

**Восстановим зависимость линейной функцией**



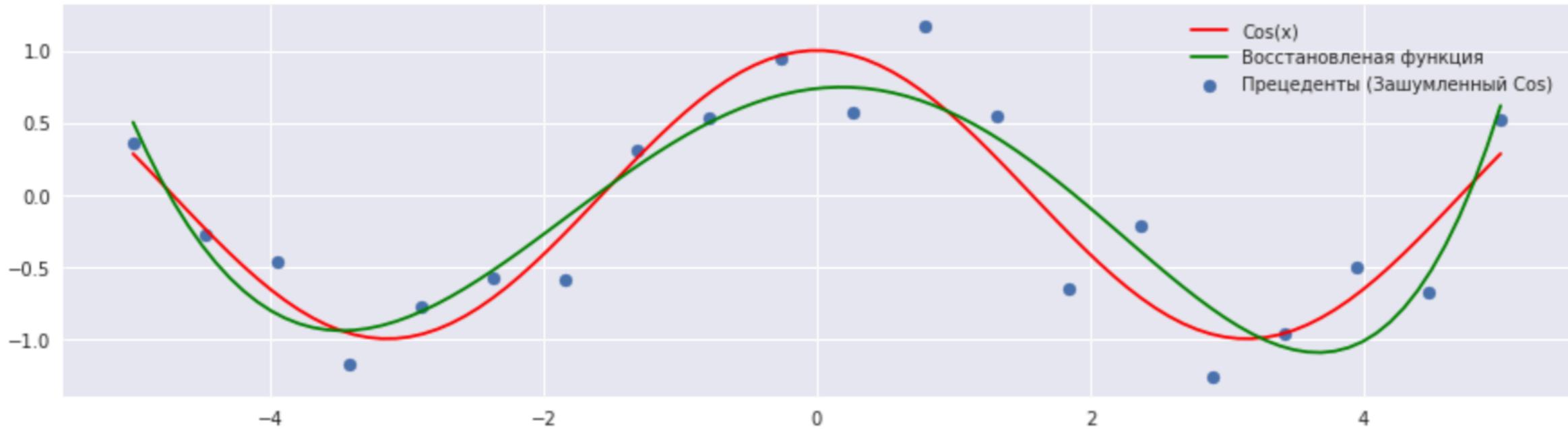
# Метрики качества

**Восстановим зависимость с помощью полинома 5-ого порядка**

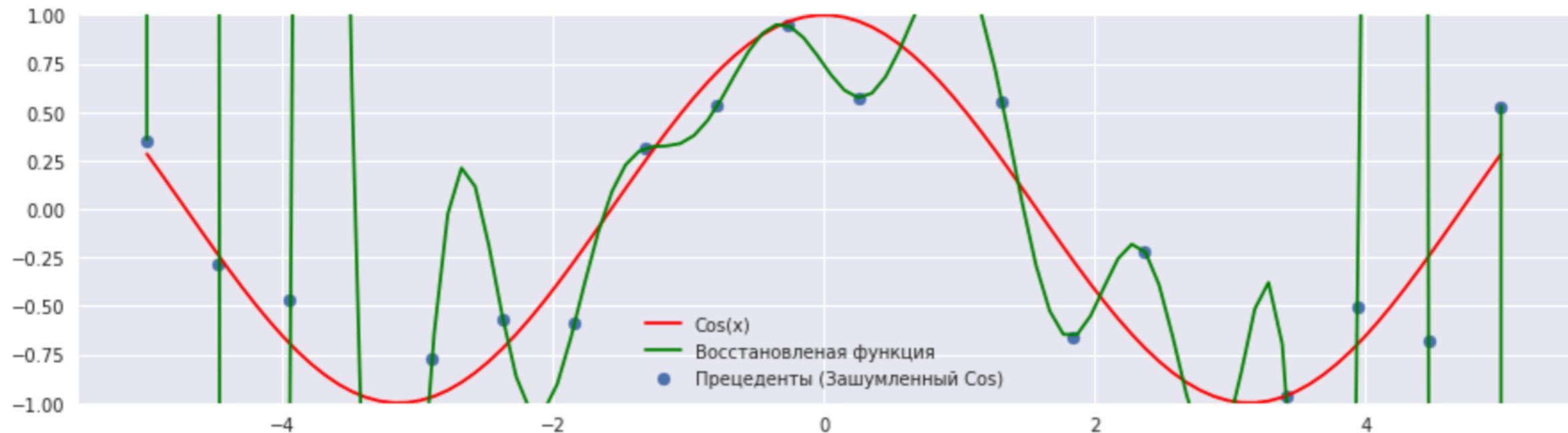


# Метрики качества

**Восстановим зависимость с помощью полинома 5-ого порядка**



**Восстановим зависимость с помощью полинома 11-ого порядка**



# Метрики качества в задачах регрессии

Средняя квадратичная (*Mean Squared Error, MSE*) ошибка:

$$MSE = \frac{1}{l} \sum_{i=1}^l (f(x_i) - y_i)^2$$

Средняя абсолютная (*Mean Absolute Error, MAE*) ошибка:

$$MAE = \frac{1}{l} \sum_{i=1}^l |f(x_i) - y_i|$$

Среднеквадратичный функционал сильнее штрафует за большие отклонения по сравнению со среднеабсолютным, и поэтому более чувствителен к выбросам.

Идеальны для сравнения моделей, но не всегда понятно как их оценивать относительно целевой переменной. Например: MSE - 10 это хорошо, если переменная лежит в пределах интервала (10000, 100000), но плохо, если целевая переменная принимает значения от 0 до 1

# Метрики качества в задачах регрессии

Коэффициент детерминации ( $R^2$ ):

$$R^2 = 1 - \frac{\sum_{i=1}^l (f(x_i) - y_i)^2}{\sum_{i=1}^l (y_i - \bar{y})^2}, \text{ где } \bar{y} = \sum_{i=1}^l y_i$$

Коэффициент детерминации измеряет долю дисперсии, объясненную моделью, в общей дисперсии целевой переменной

Если она близка к единице, то модель хорошо объясняет данные, если же она близка к нулю, то прогнозы сопоставимы по качеству с константным предсказанием.

**Другие полезные метрики:**

Квантильная регрессия

Mean Absolute Percentage Error

# Метрики качества в задачах классификации

Матрица ошибок (*confusion matrix*):

		Actual class	
		Yes	No
Predicted class	Yes	90	20
	No	10	50

Выборка: Всего 170

Положительного класса 100

Отрицательного класса 70

Прогноз:

Положительного класса 110

Отрицательного класса 60

# Метрики качества в задачах классификации

Матрица ошибок (*confusion matrix*):

		Actual class	
		Yes	No
Predicted class	Yes	True Positive (TP)	False Positive (FP)
	No	False Negative (FN)	True Negative (TN)

# Метрики качества в задачах классификации

Матрица ошибок (*confusion matrix*):

		Actual class	
		Yes	No
Predicted class	Yes	True Positive (TP)	False Positive (FP)
	No	False Negative (FN)	True Negative (TN)

Доля правильных ответов (*accuracy*):

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

# Метрики качества в задачах классификации

Матрица ошибок (*confusion matrix*):

		Actual class	
		Yes	No
Predicted class	Yes	True Positive (TP)	False Positive (FP)
	No	False Negative (FN)	True Negative (TN)

Доля правильных ответов (*accuracy*):

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

Сколько в нашем примере?  
О чём говорит эта цифра?

		Actual class	
		Yes	No
Predicted class	Yes	90	20
	No	10	50

# Метрики качества в задачах классификации

Еще один пример:

		Actual class	
		Yes	No
Predicted class	Yes	90	5
	No	10	5

Доля правильных ответов (accuracy):

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

# Метрики качества в задачах классификации

Еще один пример:

		Actual class	
		Yes	No
Predicted class	Yes	90	5
	No	10	5

Доля правильных ответов (accuracy):

$$\text{accuracy} = \frac{90 + 5}{90 + 5 + 10 + 5} = 86,4$$

# Метрики качества в задачах классификации

Матрица ошибок (*confusion matrix*):

		Actual class	
		Yes	No
Predicted class	Yes	90	5
	No	10	5

Доля правильных ответов (*accuracy*):

$$\text{accuracy} = \frac{90 + 5}{90 + 5 + 10 + 5} = 86,4$$

Давайте всегда будем предсказывать константным значением (110 объектов и все положительные). Посчитайте accuracy

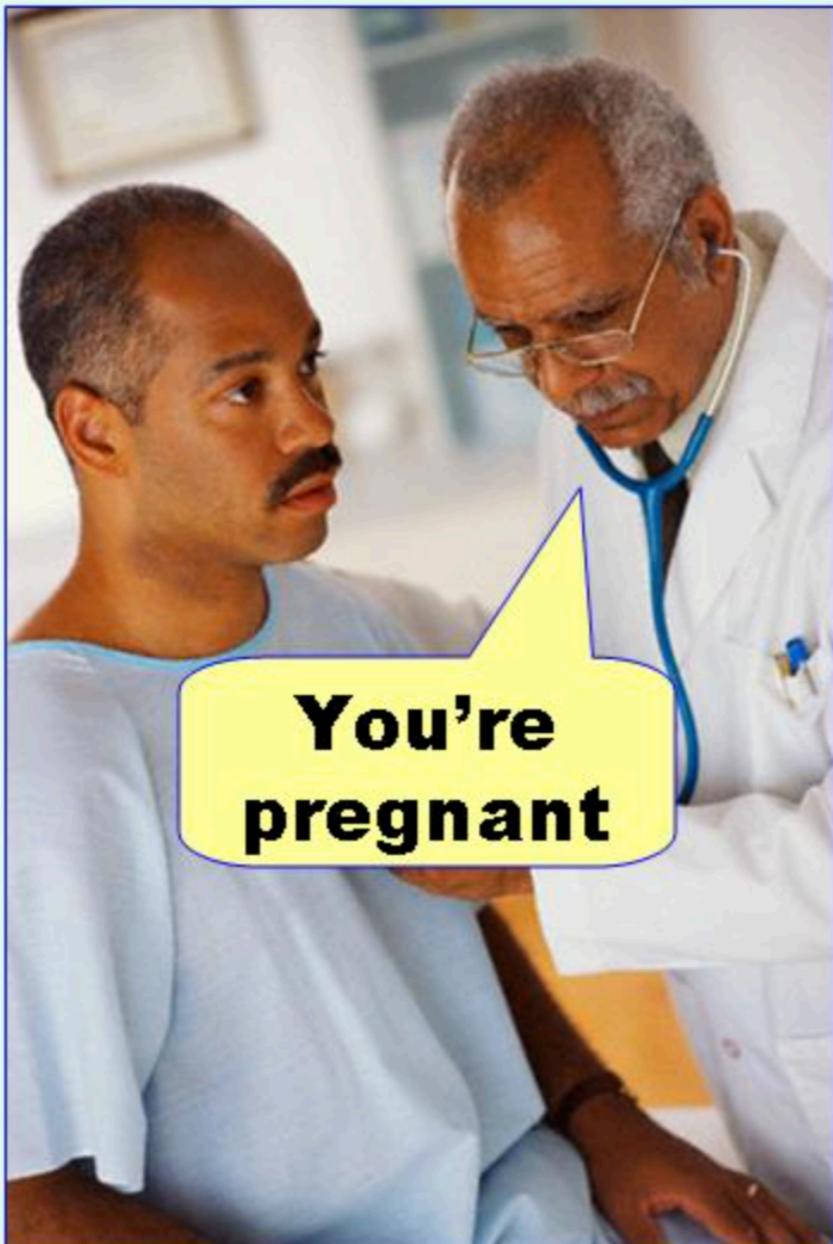
# Метрики качества в задачах классификации

Матрица ошибок (*confusion matrix*):

		Actual class		
		Yes	No	
Predicted class	Yes	True Positive (TP)	False Positive (FP)	Ошибка I-ого рода
	No	False Negative (FN)	True Negative (TN)	Ошибка II-ого рода

# Метрики качества в задачах классификации

**Type I error**  
(false positive)



**Type II error**  
(false negative)



# Метрики качества в задачах классификации

Матрица ошибок (*confusion matrix*):

		Actual class	
		Yes	No
Predicted class	Yes	True Positive (TP)	False Positive (FP)
	No	False Negative (FN)	True Negative (TN)

Точность (*precision*):

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

доля объектов, предсказанных как положительные, действительно является положительными.

Полнота (*recall*):

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

Доля положительных объектов, которую выделил классификатор

# Метрики качества в задачах классификации

Матрица ошибок (*confusion matrix*):

		Actual class	
		Yes	No
Predicted class	Yes	True Positive (TP)	False Positive (FP)
	No	False Negative (FN)	True Negative (TN)

Точность (*precision*):

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

Полнота (*recall*):

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

Какая ошибку важнее оптимизировать?

Например:

1. Решаем задачу, уйдет ли от нас клиент. Какая цена ошибки?
2. Выявление фрода. Заблокировать хорошего клиента или пропустить злоумышленника?
3. Кредитный скоринг. Выдать кредит злостному неплательщику или не выдать положительному?

# Метрики качества в задачах классификации

Матрица ошибок (*confusion matrix*):

		Actual class	
		Yes	No
Predicted class	Yes	True Positive (TP)	False Positive (FP)
	No	False Negative (FN)	True Negative (TN)

F-мера:

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

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

$\beta$  в данном случае определяет вес точности в метрике, а при  $\beta=1$  это среднее гармоническое

F-мера достигает максимума при полноте и точности, равными единице, и близка к нулю, если один из аргументов близок к нулю.

# Метрики качества в задачах классификации

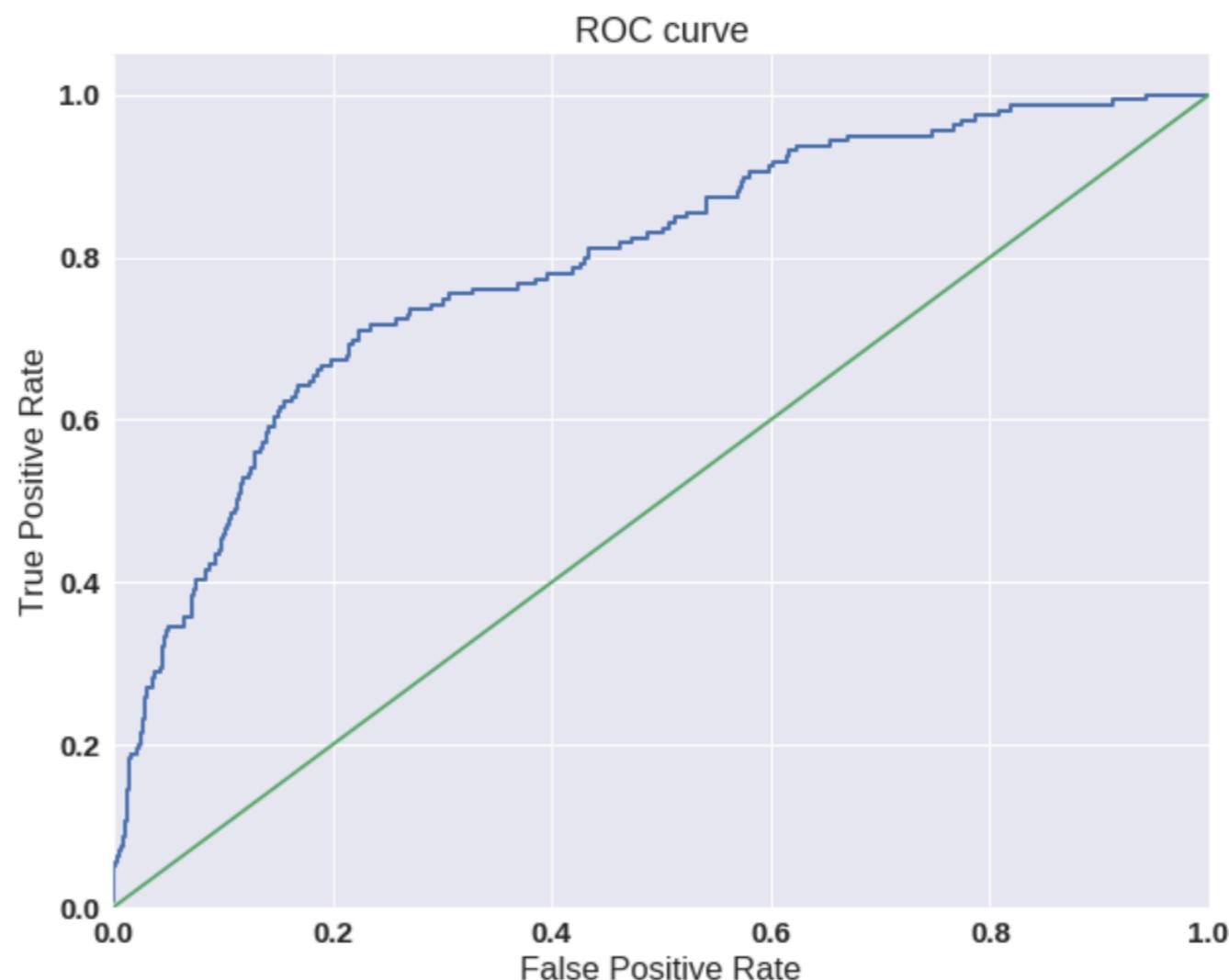
*ROC AUC*

или площадь (*Area Under Curve*) под кривой ошибок (*Receiver Operating Characteristic curve* ).

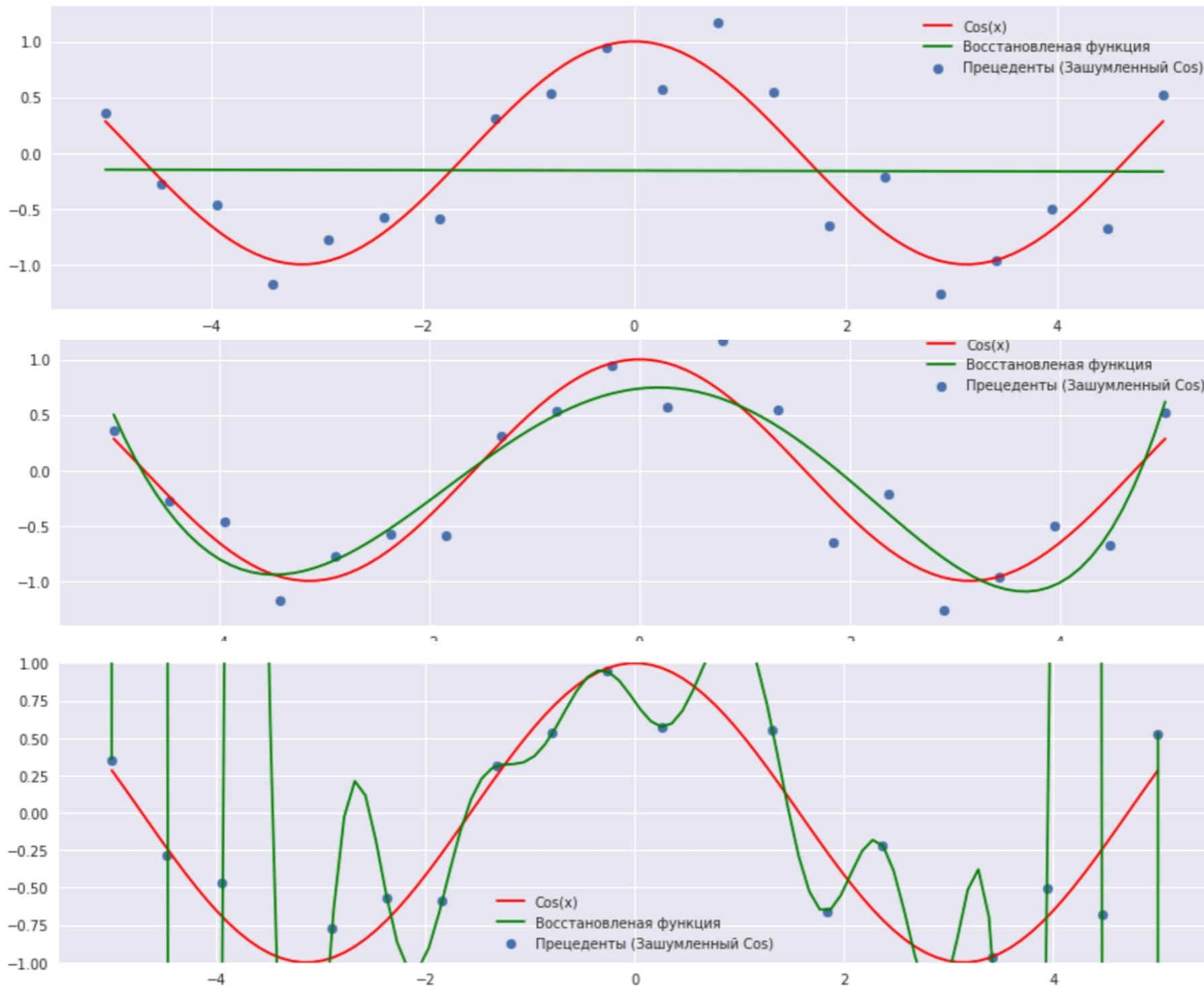
Кривая ошибок (*Receiver Operating Characteristic curve* ) представляет из себя линию от (0,0) до (1,1) в координатах True Positive Rate (TPR) и False Positive Rate (FPR):

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

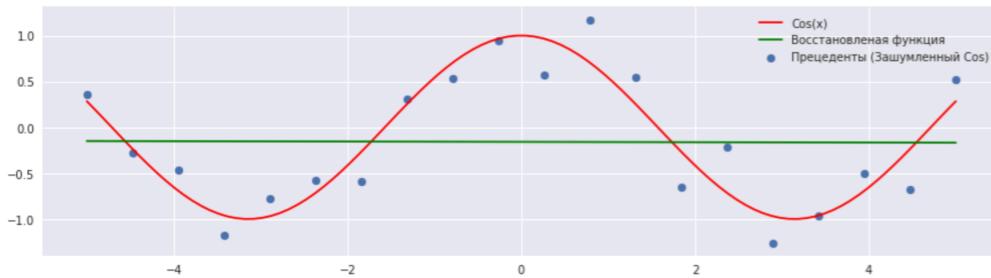
$$FPR = \frac{FP}{FP + TN}$$



# Мотивация валидации

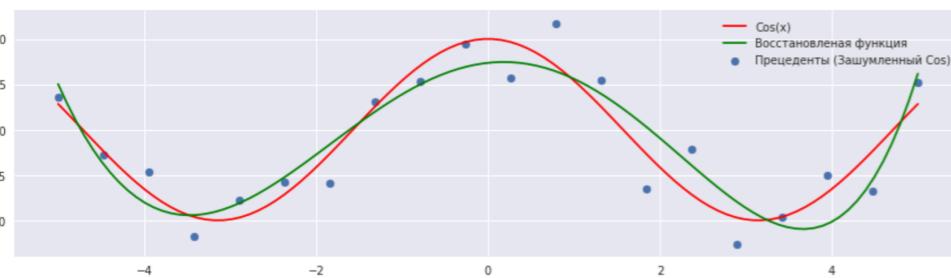


# Мотивация валидации



MSE	MAE	R2
-----	-----	----

Линейная модель	0.472	0.586	0.0004
-----------------	-------	-------	--------

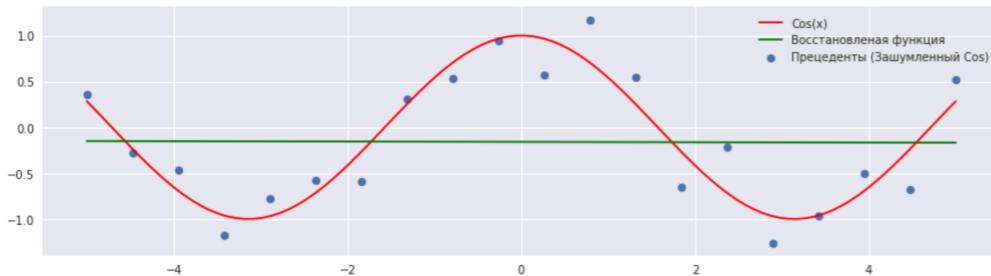


Полином 5-ой степени	0.047	0.179	0.9000
----------------------	-------	-------	--------



Полином 11-ой степени	0.000	0.000	1.0000
-----------------------	-------	-------	--------

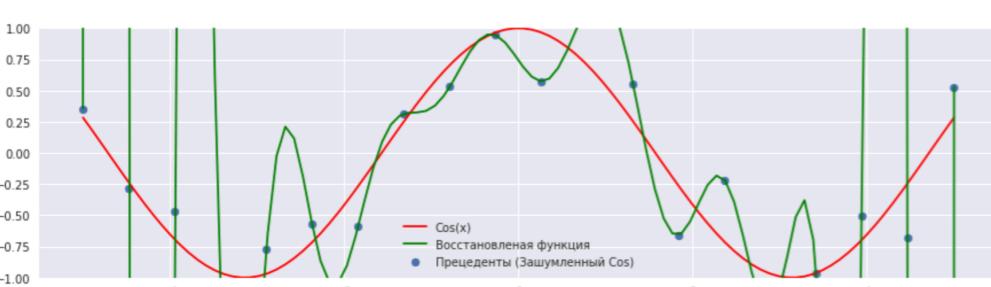
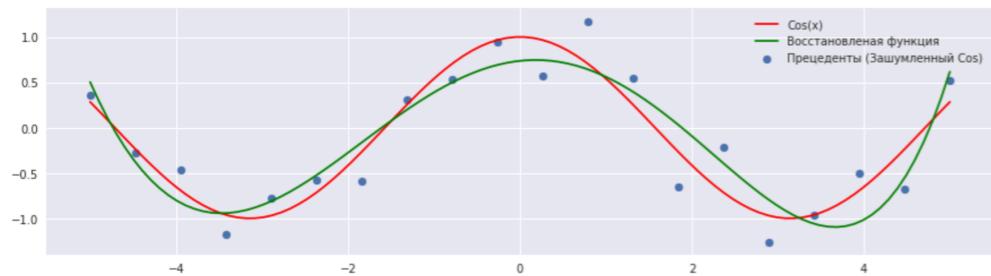
# Мотивация валидации



MSE	MAE	R2
-----	-----	----

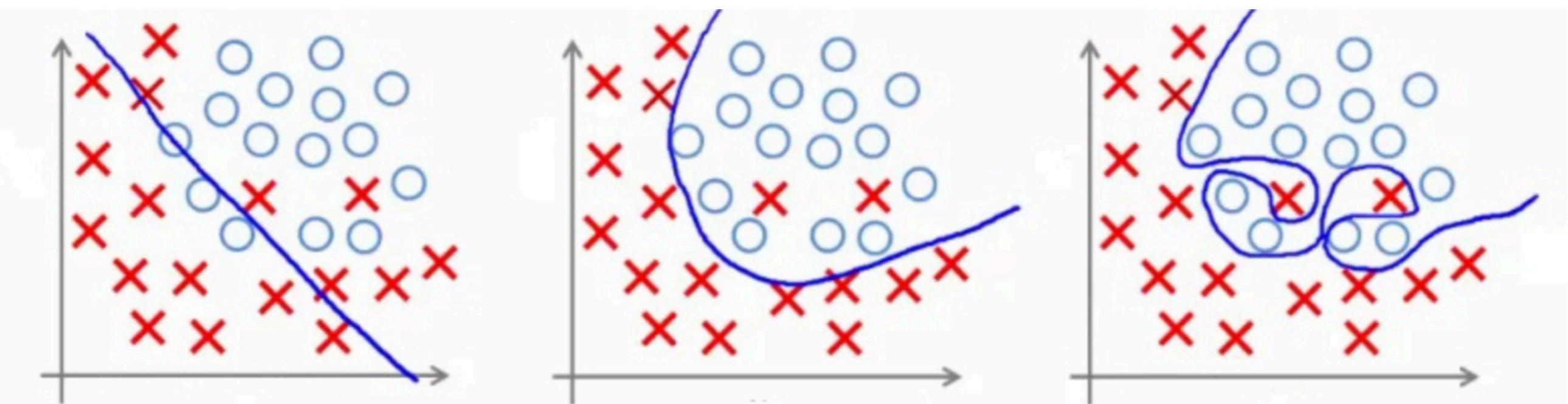
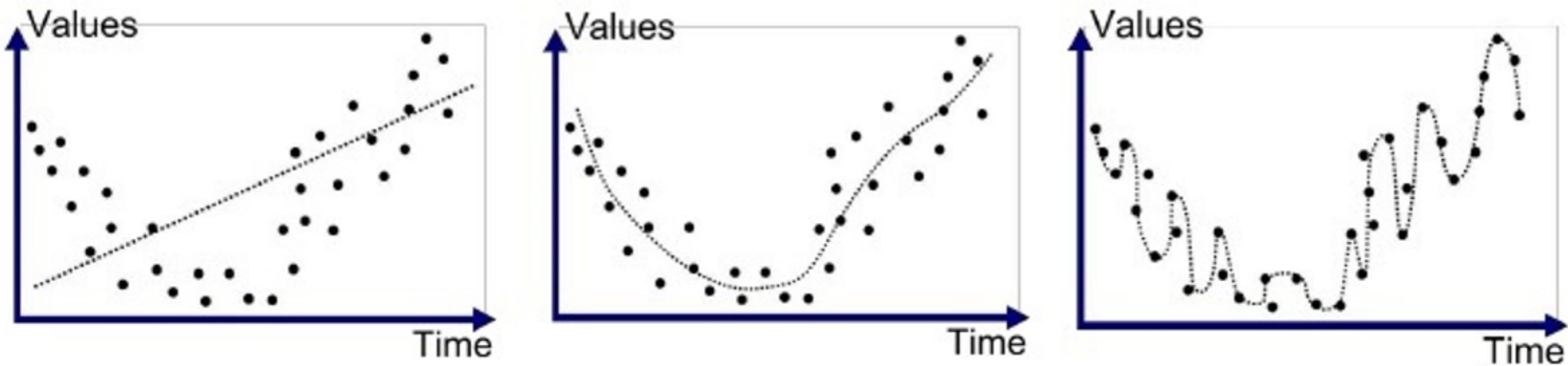
Линейная модель	0.472	0.586	0.0004
-----------------	-------	-------	--------

Полином 5-ой степени	0.047	0.179	0.9000
----------------------	-------	-------	--------

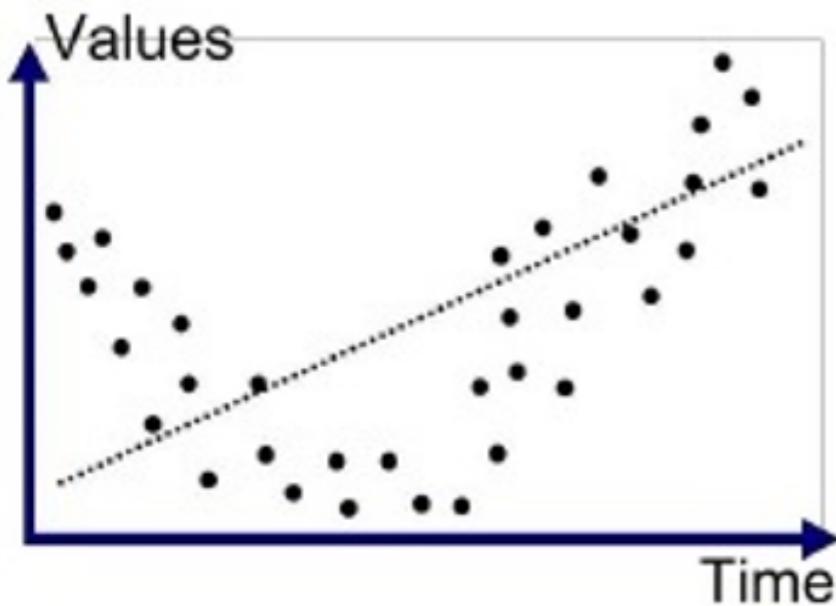


**Обобщающая способность  
Полинома 11-ой степени  
отсутствует.**

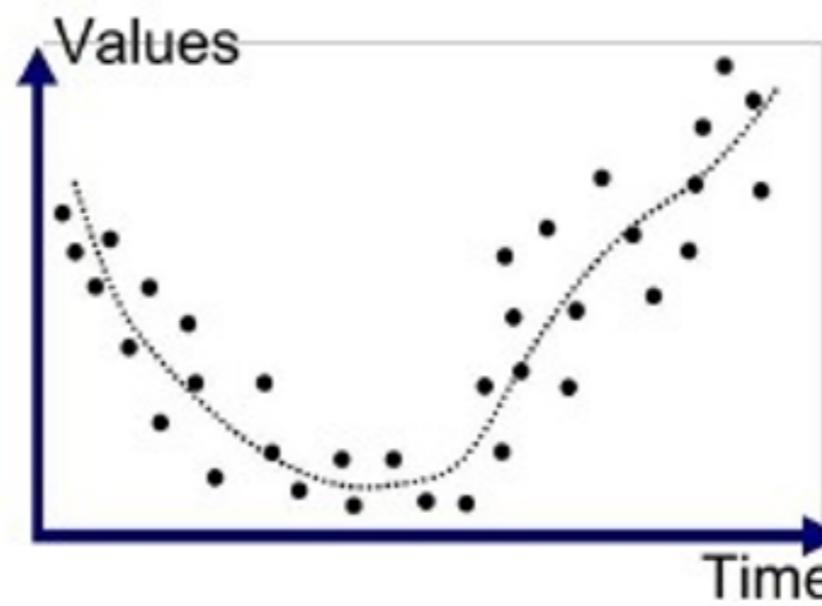
# Мотивация валидации



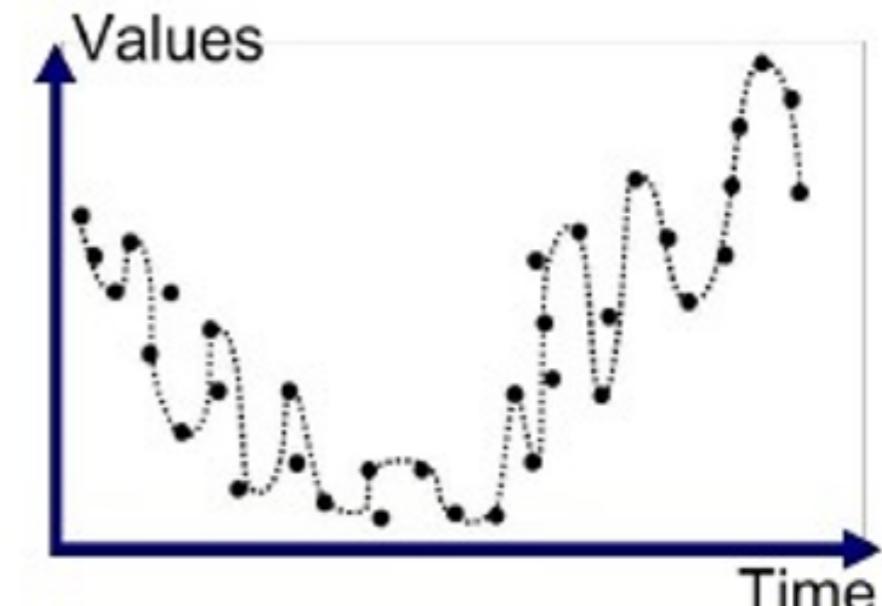
# Мотивация валидации



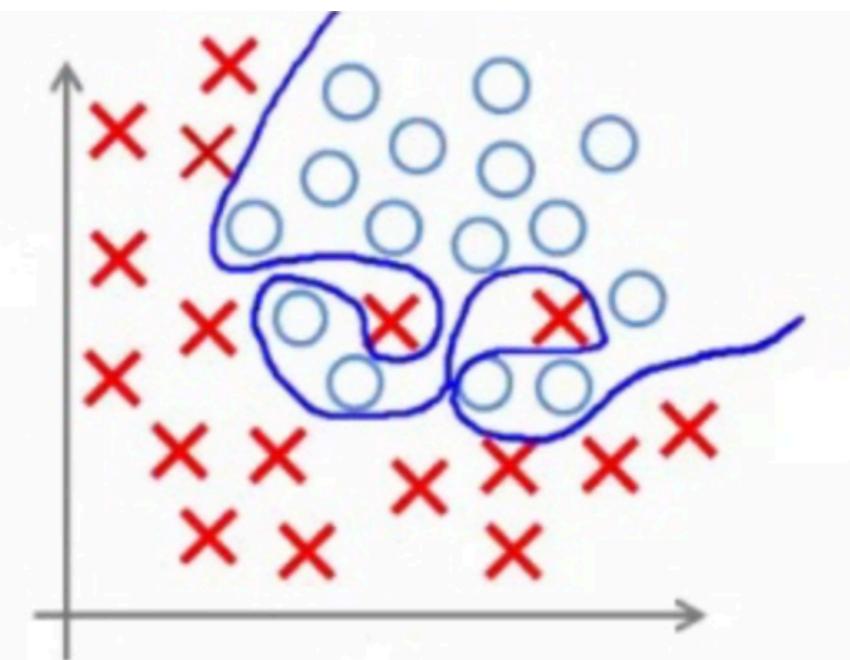
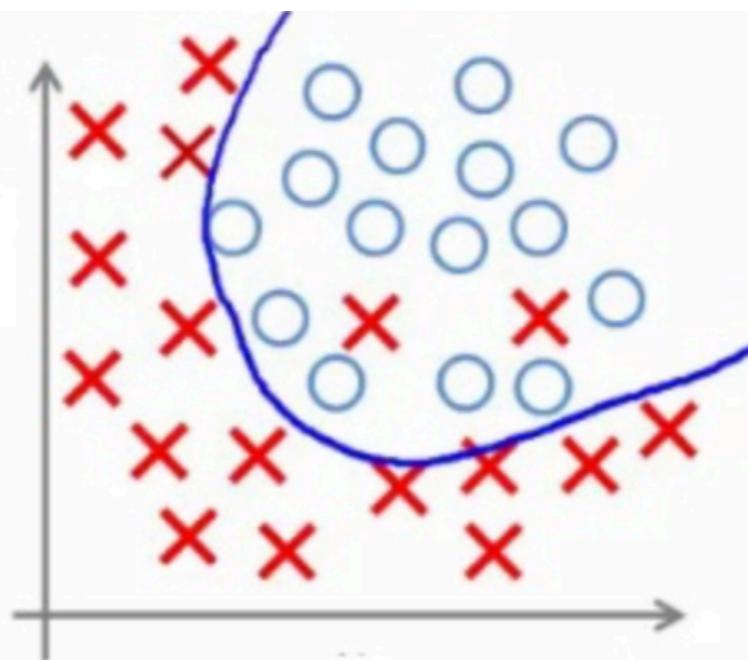
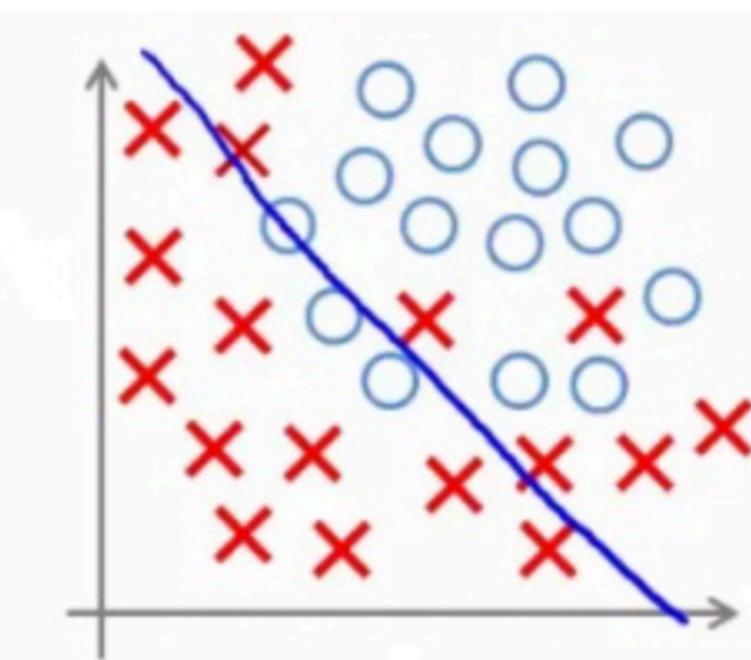
Underfitted



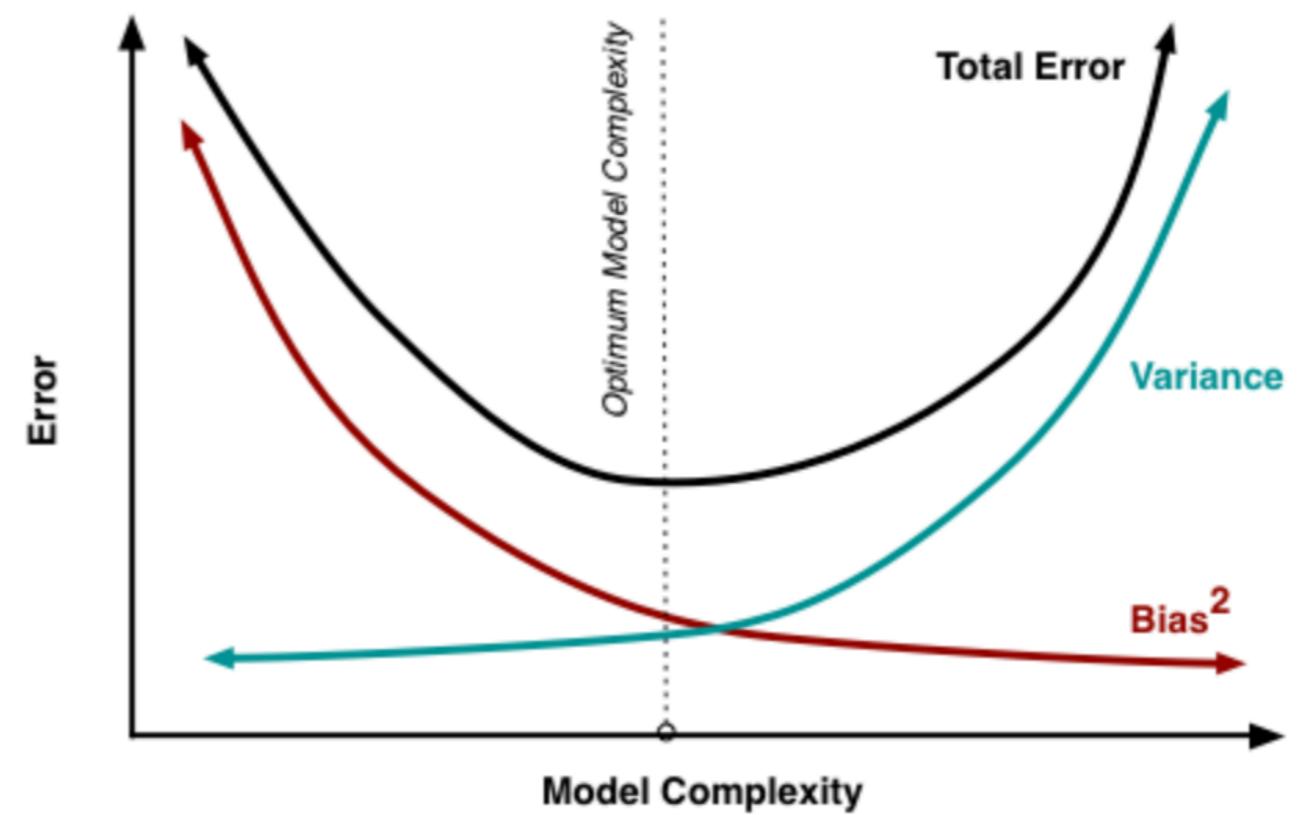
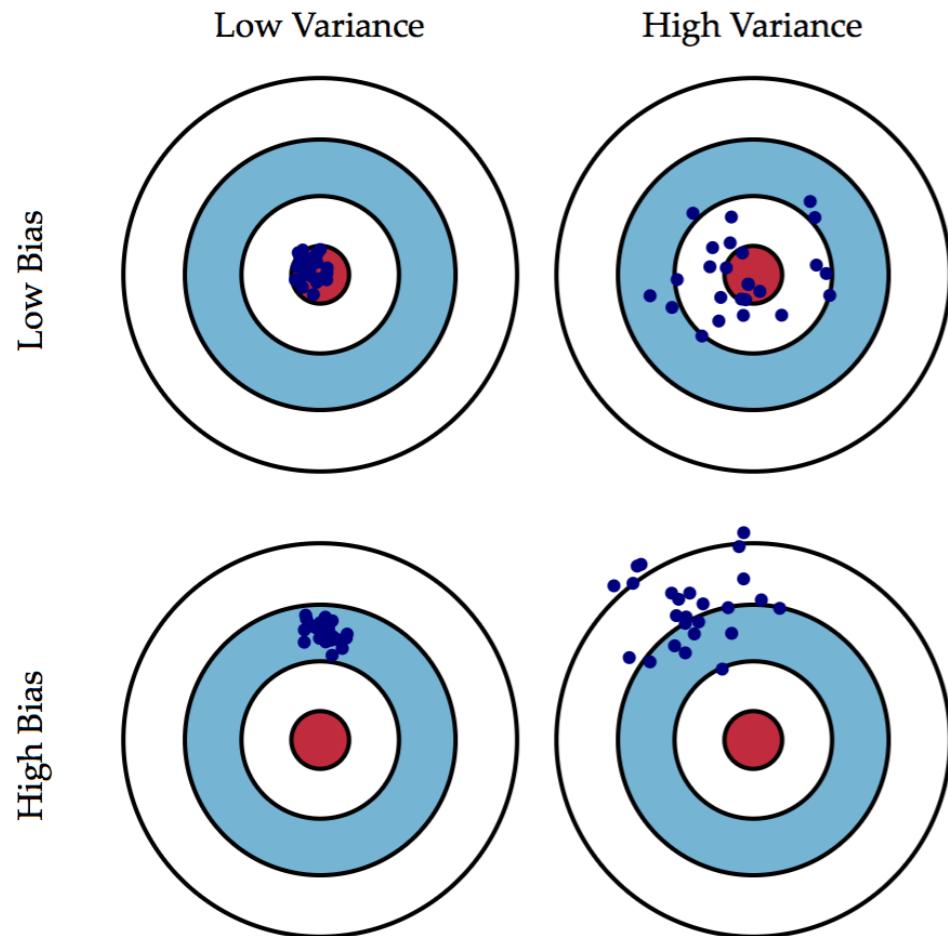
Good Fit/R robust



Overfitted



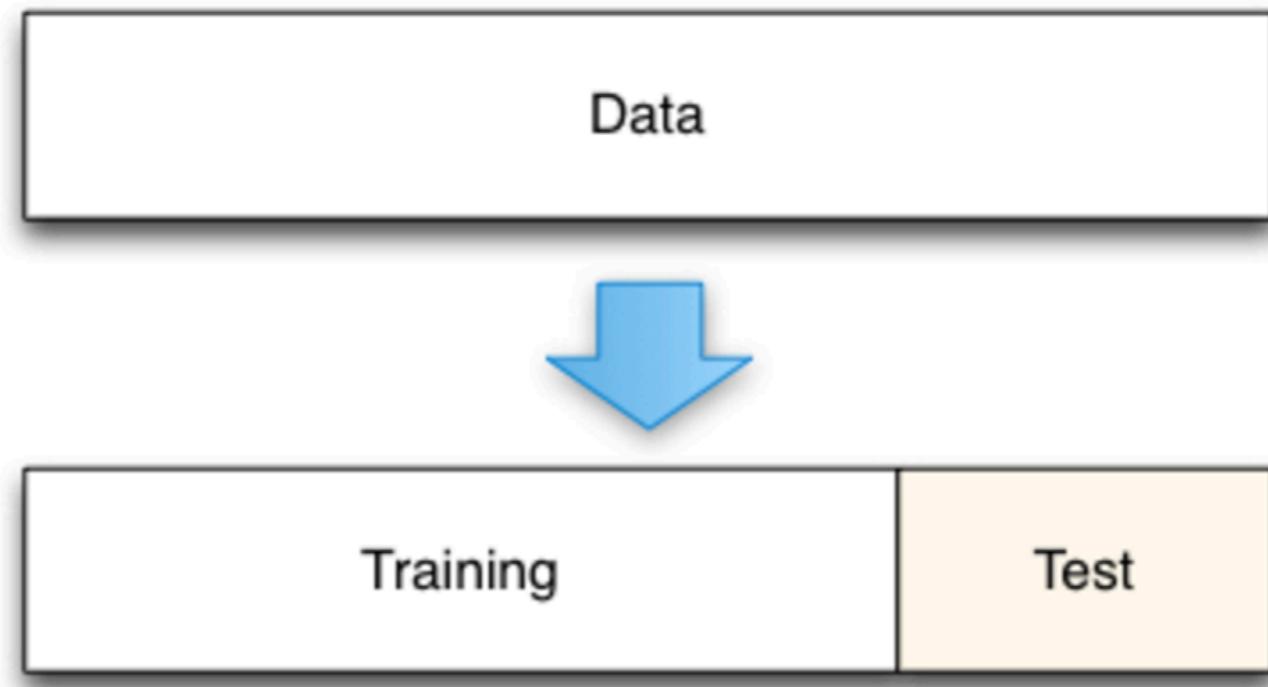
# Bias and Variance tradeoff



$$Err(x) = E[(Y - \hat{f}(x))^2]$$

$$Err(x) = Bias^2 + Variance + IrreducibleError$$

# Стратегии валидации



# Стратегии валидации



# Формальная постановка задачи

Дана обучающая выборка (объекты независимы):

$$X_m = \{ (x_1, y_1), \dots, (x_m, y_m) \}$$

Для задачи регрессии - Целевая переменная задана вещественным числом

$$(x_1, y_1) \in \mathbb{R}^m \times \mathbb{Y}, \mathbb{Y} = \mathbb{R}$$

Для задачи классификации - Целевая переменная задана конечным числом меток

$$(x_1, y_1) \in \mathbb{R}^m \times \mathbb{Y}, \mathbb{Y} = \{-1; 1\}$$

Задать такую функцию  $f(x)$  от вектора признаков  $x$ , которое выдает ответ для любого возможного наблюдения  $x$

$$f(x): \mathbb{X} \rightarrow \mathbb{Y}$$

**Основная гипотеза МО:** Схожим объектам соответствуют схожие объекты

# Метод k ближайших соседей (kNN, *k Nearest Neighbours*)

**Гипотеза компактности:** если мера сходства объектов введена достаточно удачно, то схожие объекты гораздо чаще лежат в одном классе, чем в разных.

- Вычислить расстояние до каждого из объектов обучающей выборки
- Отобрать k объектов обучающей выборки, расстояние до которых минимально
- Класс классифицируемого объекта — это класс, наиболее часто встречающийся среди k ближайших соседей



## `sklearn.neighbors.KNeighborsRegressor`

`(n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2,  
metric='minkowski', metric_params=None, n_jobs=None, **kwargs)`

## `sklearn.neighbors.KNeighborsClassifier`

`(n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2,  
metric='minkowski', metric_params=None, n_jobs=None, **kwargs)`

# Метод k ближайших соседей (kNN, *k Nearest Neighbours*)

**Гипотеза компактности:** если мера сходства объектов введена достаточно удачно, то схожие объекты гораздо чаще лежат в одном классе, чем в разных.

- Вычислить расстояние до каждого из объектов обучающей выборки
- Отобрать k объектов обучающей выборки, расстояние до которых минимально
- Класс классифицируемого объекта — это класс, наиболее часто встречающийся среди k ближайших соседей



## `sklearn.neighbors.KNeighborsRegressor`

`(n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2,  
metric='minkowski', metric_params=None, n_jobs=None, **kwargs)`

## `sklearn.neighbors.KNeighborsClassifier`

`(n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2,  
metric='minkowski', metric_params=None, n_jobs=None, **kwargs)`

### Выбор числа соседей k

При  $k=1$  алгоритм ближайшего соседа неустойчив к шумовым выбросам: он даёт ошибочные классификации не только на самих объектах-выбросах, но и на ближайших к ним объектах других классов.

При  $k=l$ , наоборот, алгоритм чрезмерно устойчив и вырождается в константу. Таким образом, крайние значения нежелательны.

# Метод k ближайших соседей (kNN, *k Nearest Neighbours*)

## Выбор метрики

Евклидово расстояние (“euclidean”)

$$\sqrt{\sum (x - y)^2}$$

Расстояние городских кварталов «манхэттенское расстояние» (“manhattan”)

$$\sum |x - y|$$

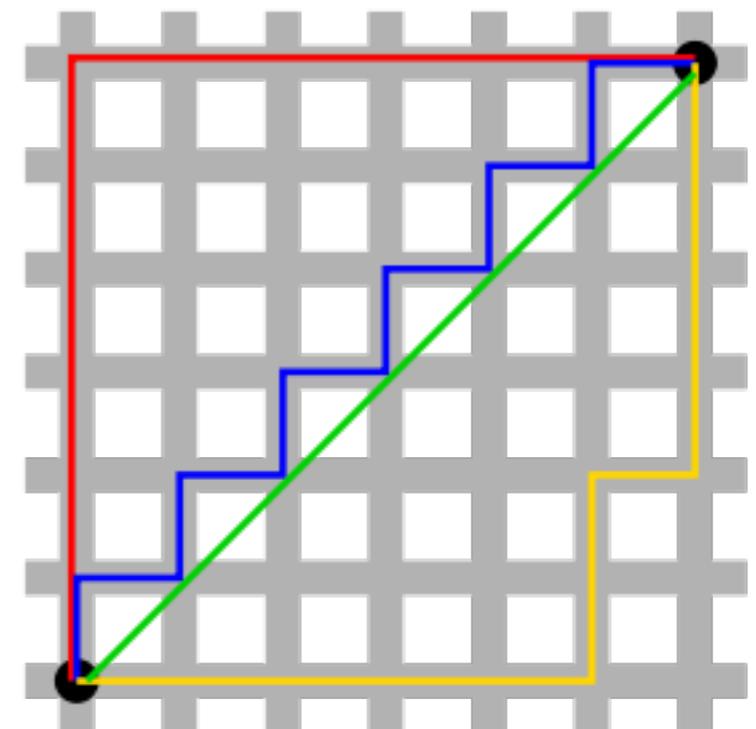
Расстояние Чебышева “chebyshev”

$$\max(x - y)$$

Расстояние Минковского “minkowski”

$$\left( \sum |x - y|^p \right)^{\frac{1}{p}}$$

расстояния с параметром  $p$  равным 1 (расстояние городских кварталов) или 2 (евклидова метрика).  
 $p = \infty$  метрика обращается в расстояние Чебышёва.



# Метод k ближайших соседей (kNN, *k Nearest Neighbours*)

## Нормирование признаков

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

$$z \in [0,1]$$

## Стандартизация признаков

$$z = \frac{x - \mu}{\sigma}$$

где

$\mu$  - Среднее

$\sigma$  - Стандартное отклонение

## Проклятие размерности

В пространстве высокой размерности все объекты примерно одинаково далеки друг от друга; выбор ближайших соседей становится практически произвольным.

# Ссылки

1. Статья на habr: [Метрики в задачах машинного обучения](#)
2. [Семинар](#) из курса Евгения Соколова