# Титульный лист материалов по дисциплине

*(заполняется по каждому виду учебного материала)*

| | |
|---|---|
| ДИСЦИПЛИНА | **Технологии извлечения знаний из больших данных**<br>(полное наименование дисциплины без сокращений) |
| ИНСТИТУТ | ИКБ |
| КАФЕДРА | **КБ-4 «Интеллектуальные системы информационной безопасности»**<br>полное наименование кафедры) |
| ВИД УЧЕБНОГО МАТЕРИАЛА | **Лекция**<br>(в соответствии с пп.1-11) |
| ПРЕПОДАВАТЕЛЬ | **Никонов В.В.**<br>(фамилия, имя, отчество) |
| СЕМЕСТР | **3 семестр 2023/2024 уч. года**<br>(указать семестр обучения, учебный год) |

# Базовая работа с табличными данными.

## Задача регрессии. Постановка задач регрессии и примеры таких задач

Обычная задача машинного обучения — построить модель зависимости «выхода» от входных данных. Мы будем рассматривать **задачу регрессии.** В такой задаче на выходе модели получается непрерывная величина в заданном диапазоне.

Для начала приведем формальную постановку нашей задачи:

- Задана выборка значений признаков: $x_n$

$$\{x, \ldots, x_n \mid x \in R^d\}.$$

Здесь ***d*** — размерность признакового пространства и ***n*** — количество элементов в нашей выборке входных данных.

- Задана выборка соответствующих значений целевой переменной:

$$\{y, \ldots, y_n \mid y \in R\}.$$

Получаем множество исходных данных:

$$\mathrm{D} = \{(x,\ y)_i\}_{i=1}^{n}$$

- Задано параметрическое семейство функций $f(\omega,\ x),$ зависящее от параметров $\omega \in R$ и от входных признаков *x*.

Здесь $f(\omega,\ x)$ — функция регрессионной зависимости. Формально это параметрическое семейство функций вида $f\colon W \times X \to Y$, где $W$ — пространство параметров (весов модели) $\omega$, $X$ — пространство значений признаков *x* и Y — пространство целевых переменных y.

- Нужно построить модель, предсказывающую по $x_i$ значение $\widehat{y_i}$, наиболее близкое к $y_i$: $\widehat{y_i} = f(\omega, x_i)$

- Для оценки близости предсказания к истинному значению используются функции потерь, например**MSE** — **M**ean**s**quared**e**rror, или среднеквадратичная ошибка:

$$\mathcal{L}(x, y, \omega) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\omega, x_i))^2$$

Чем меньше значение MSE, тем ближе к истинным значениям будет выход модели.

Таким образом, задача регрессии состоит в том, чтобы решить оптимизационную задачу $\mathcal{L}(x, y, \omega) \to \min_{\omega}$ и найти подходящий набор параметров $\omega$.

### *Примеры задач регрессии*

В целом задача регрессии — это прогнозирование вещественного числа на основе выборки объектов с различными признаками. Так, можно смоделировать задачу нахождения стоимости недвижимости или количества кликов пользователей на баннеры рекламы.

Давайте рассмотрим задачу прогнозирования трафика автомагистрали:

| | holiday | temp | rain_1h | snow_1h | clouds_all | weather_main | weather_description | date_time | traffic_volume |
|---|---|---|---|---|---|---|---|---|---|
| 0 | None | 288.28 | 0.0 | 0.0 | 40.0 | Clouds | scattered clouds | 2012-10-02 09:00:00 | 5545.0 |
| 1 | None | 289.36 | 0.0 | 0.0 | 75.0 | Clouds | broken clouds | 2012-10-02 10:00:00 | 4516.0 |
| 2 | None | 289.58 | 0.0 | 0.0 | 90.0 | Clouds | overcast clouds | 2012-10-02 11:00:00 | 4767.0 |

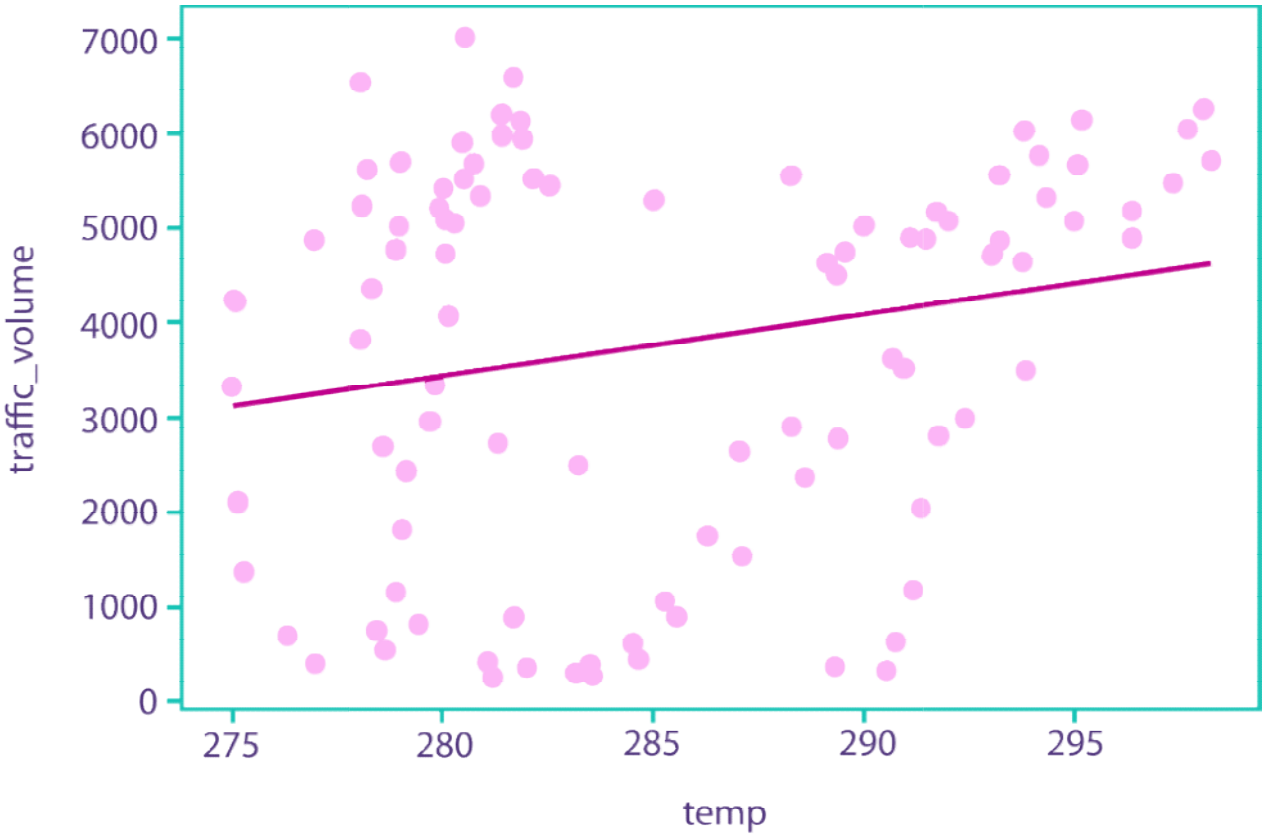
Здесь, первый столбец — это индексы строк датасета, а последний — целевая переменная ***y***, значение трафика, зависящее от вектора признаков $\boldsymbol{x} = (x^{holiday}, x^{temp}, \ldots, x^{date\_time})$. Чтобы спрогнозировать значение трафика, мы можем взять линейную модель, например*линейную регрессию* (подробнее методы регрессии рассмотрим далее в курсе, здесь же приведем только функцию регрессионной зависимости):

$$\hat{y}_i = f(\omega, x_i) = \sum_{j \in F} \omega_j \cdot x_i^j = \omega^T x_i$$

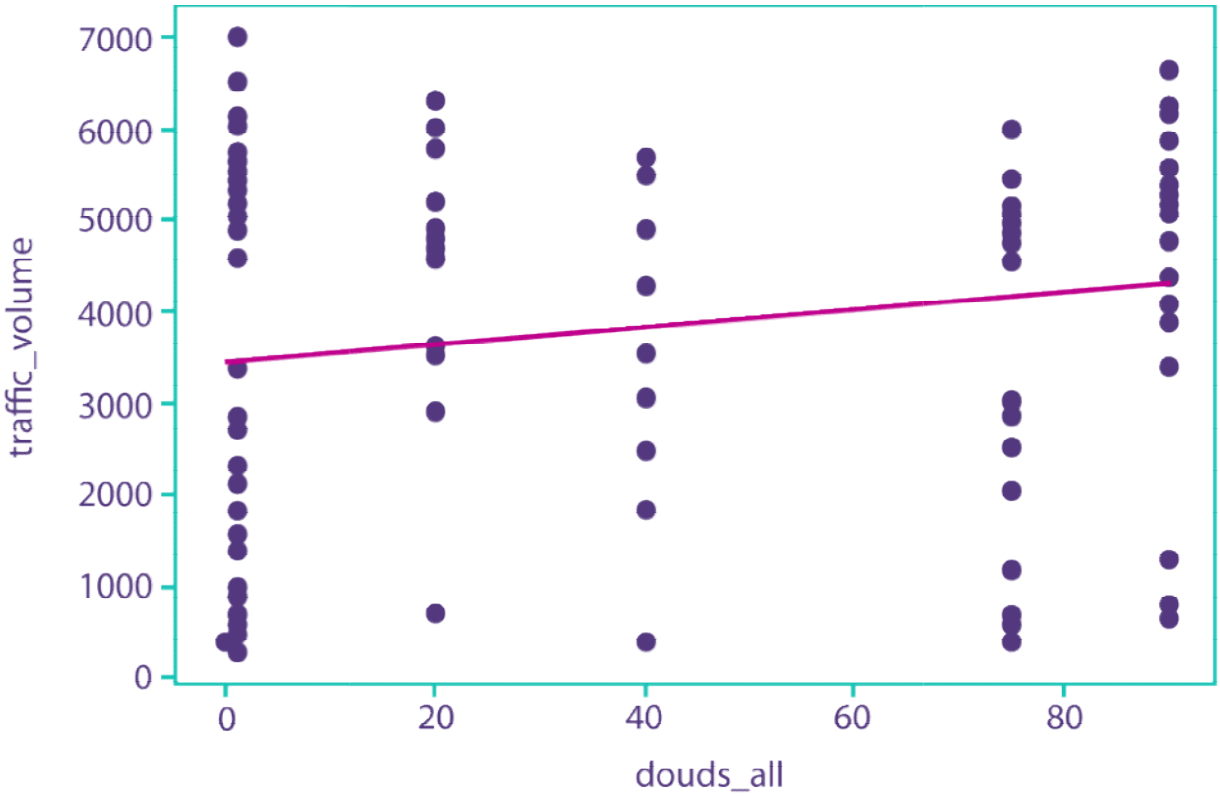
Линейная регрессия является скалярным произведением вектора весов $\omega$ и вектора признаков $x_i$, то есть их поэлементным произведением по всем признакам $j$ из пространства признаков $\boldsymbol{F}$.

Затем с помощью минимизации среднеквадратичной ошибки **MSE** найдем подходящий набор параметров $\hat{\omega}$ .

Для примера, если взять из признаков только температуру (столбец „temp”) и построить зависимость трафика от температуры, то получим следующий график:




Как видим, данные сильно разбросаны, так что только по показаниям температуры вряд ли получится сделать предсказание, близкое к истинному. Если же учитывать только погодный показатель (столбец „clouds_all”), то получим еще менее надежные результаты:




Но если объединить все признаки в линейную комбинацию (что и представляет собой линейная регрессия), то предсказания модели будут близки к истинным.

**Метрики, критерии качества для задач регрессии**

**Метрика качества** — это оценка качества нашей модели. Причем для конкретной задачи может возникать целая иерархия метрик.

Рассмотрим на примере. Пусть перед нами стоит задача создать предсказательную модель для портала поиска недвижимости для ее последующей аренды или покупки. В таком случае иерархия метрик может иметь следующий вид:

- Самый верхний уровень: будущий доход сервиса — практически невозможно измерить в моменте, сложным образом зависит от совокупности наших усилий как в выборе и обучении модели, так и в продакшене
- Доля удовлетворенных качеством подбора жилья экспертов, на которых мы протестируем модель, прежде чем представлять ее пользователям

- Функция потерь, на которой мы будем обучать нашу модель

Как видно, можно выделить следующую закономерность: во-первых, бизнес-метрики, интересующие нас в первую очередь, практически неимплементируемы в процесс обучения модели; во-вторых, функция потерь ничего не говорит нам о том, насколько хорошо мы решили поставленную задачу.

Поэтому в общем случае метрика качества не совпадает с функцией потерь, и нужно уметь различать эти понятия:

**Метрика** — внешний критерий качества, зависящий от предсказанных значений, но не от параметров модели.

**Функция потерь** — критерий «обучаемости» нашей модели. Возникает, когда мы переходим от построения модели к задаче оптимизации, то есть поиску оптимальных параметров модели.

Теперь вернемся к задаче регрессии. Мы уже рассматривали такую функцию потерь, как среднеквадратичная ошибка MSE. Такая функция может быть одновременно и метрикой в силу постановки задачи. Мы стараемся не точно предсказать значение, а найти наиболее близкое в интервале по всей совокупности объектов в выборке.

В таком случае появляются другие вопросы о характеристиках модели. Важно понимать, подходит ли выбранная метрика под решаемую задачу. Например, MSE, с которой мы познакомились в прошлом лонгриде, накладывает большой штраф на сильные отклонения в выборке за счет возведения в квадрат, но в то же время ничего не говорит об «абсолютном» качестве модели. Если мы получим значение MSE в 10 000, то отклонение составит $\sqrt{10\,000} = 100$. Тогда при предсказанном значении в интервале [0; 200] смысла от такой модели будет мало. Если же предсказываются значения из интервала [0; 20 000], то отклонение в 100 получается не таким уж и большим.

Далее мы рассмотрим различные метрики регрессии и приведем характеристики каждой из них.

### *Метрики качества для задачи регрессии*

Итак, давайте повторим обозначения:

$\{x, \ldots, x_n \mid x \in R^d\}$ — выборка значений признаков;

$\{y, \ldots, y_n \mid y \in R\}$ — выборка соответствующих значений целевой переменной;

$f(\omega, x) = \omega^T x$ — функция линейной регрессии.

В прошлый раз мы уже познакомились с одной из самых распространенных метрик задачи регрессии — **MSE** (**M**ean**s**quare**e**rror):

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - w^T x_i)^2$$

Как мы уже рассмотрели выше, MSE-функционал сильно чувствителен к выбросам за счет возведения в квадрат; однако о том, насколько хорошо наша модель решает поставленную задачу, используя MSE, вывод мы сделать не можем.

Также есть и другие метрики:

- **MAE** (**M**ean**a**bsolute**e**rror):

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - w^T x_i|$$

- **MAPE** (**M**ean **a**bsolute **p**ercentage **e**rror):

$$\frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - w^T x_i}{y_i}\right|$$

Используется, чтобы учитывать масштаб целевой переменной. То есть мы минимизируем *процент*,на который наша модель ошибается. MAPE — метрика с очень простой интерпретацией, измеряемая в долях или процентах. Если у вас получилось, например, что MAPE = **n**%, то это говорит о том, что ошибка составила **n**% от реальных значений. Однако такая метрика нестабильна: любое сильное отклонение может значительно повлиять на размер ошибки.

- **MASE** (**M**ean **a**bsolute **s**caled **e**rror):

$$\frac{\sum_{i=1}^{n} |y_i - w^T x_i|}{\frac{n}{n-1} \sum_{i=2}^{n} |y_i - y_{i-1}|}$$

Функционал MASE является отличным показателем точности модели: не зависит от масштаба данных, является симметричным и дает несмещенную оценку. Хотя на практике данную метрику почти невозможно интерпретировать.

- **R2** (коэффициент детерминации):

$$1 - \frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - w^T x_i)^2}{\frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{y_i})^2}$$

Здесь $\overline{y}$ — среднее истинного значения *y*.

Заметим, что дробь в формуле можно понимать как отношение дисперсии, *объясняемой моделью*, к дисперсии выборки (дисперсии константной модели, которая всегда предсказывает выборочное среднее). Тогда формулу можно переписать следующим образом:

$$R^2 = 1 - \frac{MSE}{\sigma_y^2}$$

В отличие от других метрик, эту мы стремимся максимизировать при обучении модели, т. к. она максимальна при MSE = 0, например. Значение $R^2$ меньше 0 означает, что рассматриваемая модель *хуже* константой. Для других задач можно использовать иные метрики качества, в своей задачи вы должны отталкиваться от того, что модель должна делать хорошо на выходе.

Итак, давайте подведем итоги по полученным метрикам в виде таблицы:

| Метрика | Достоинства | Недостатки |
|---|---|---|
| MSE | Позволяет накладывать большие штрафы на сильные отклонения, практически обнуляя соответствующие веса признаков | Нестабильна при сильных отклонениях в выборке |

| | | |
|---|---|---|
| MAE | Легко интерпретировать | В отличие от MSE не штрафует за сильные отклонения |
| MAPE | Легко интерпретировать | Нельзя использовать, если есть нулевые или близкие к нулю значения, также нет верхнего предела процентной ошибки для слишком больших предсказаний — из-за отклонений в выборке можно получить ошибку в сотни процентов |
| MASE | Лишена недостатков MAE и инвариантна к масштабу данных, поэтому ее можно использовать для сравнения прогнозов по разным наборам данных | Не интерпретируется |
| $R^2$ | Легко интерпретировать — определяет долю изменений, обусловленных влиянием признаков на целевую переменную | Не уменьшается при добавлении новых признаков, независимо от их связи с целевой переменной; поэтому модели с разным количеством признаков с помощью $R^2$ сравнить не получится |

Как видим, нет универсальной метрики, отвечающей всем требованиям, поэтому стоит не зацикливаться на одной оценке качества, а проводить комплексный анализ модели.

### *Связь квадратичной ошибки и метода максимума правдоподобия*

Давайте разберемся в этом вопросе на примере **линейной регрессии.** Для начала введем несколько определений:

**- Функция правдоподобия**

Пусть дано параметрическое семейство распределений вероятности $\{P_\omega\}_{\omega \in W}$ и выборка $X_1, X_2, \ldots, X_d \sim P_\omega$.

Тогда для фиксированной реализации выборки x (при работе с табличными данными $x$ — строчка датасета, а $X_i$ — i-й столбец датасета) совместное распределение этой выборки $f_X(x \mid \omega), x \in R^d$ называется *функцией правдоподобия*.

**- Метод максимизации правдоподобия**

Это метод оценивания неизвестного параметра путем максимизации функции правдоподобия.

Неформально правдоподобие позволяет оценивать неизвестные параметры (параметры модели $\omega$), основанные на известных результатах (множество исходных данных). Функция правдоподобия — это функция вероятности, поэтому и работать изначально мы будем с вероятностью, а не функцией потерь.

С математической точки зрения регрессия — зависимость математического ожидания случайной величины от других случайных величин:

$$E(y|x) = f(x)$$

Мы можем представить регрессию как сумму неслучайной ($\boldsymbol{f}$) и случайной частей:

$$y = f(w, x) + \epsilon$$

Здесь, $\omega \in W$ — пространство параметров. $x \in X$ — пространство признаков и $Y$ — пространство зависимых (целевых переменных). Также стоит отметить, что $\epsilon$ — аддитивная случайная величина, имеющая нормальное (гауссово) распределение с нулевым средним и фиксированной дисперсией. Такое предположение позволит нам в дальнейшем использовать, например, метод наименьших квадратов для линейной регрессии.

$f(\omega, x)$ - функция регрессионной зависимости. Формально это параметрическое семейство функций вида $f: W \times X \to Y$.

Теперь приведем формулировку **задачи регрессии** с прошлого раза, но в несколько другой форме:

- Задана выборка значений признаков: $x_n$

$\left\{x, \ldots, x_n \mid x \in R^d\right\}$.

Здесь ***d*** — размерность признакового пространства и ***n*** — количество элементов в нашей выборке входных данных.

- Задана выборка соответствующих значений целевой переменной:

$\{y, \ldots, y_n \mid y \in R\}$.

Получаем множество исходных данных:

$$D = \{(x, y)_i\}_{i=1}^{n}$$

- Задано параметрическое семейство функций $f(\omega, x)$, зависящее от параметров $\omega \in R$ и от входных признаков *x*.
- Нужно найти наиболее вероятные параметры (или **максимум правдоподобия**)

$$\hat{w} = \underset{w \in R^W}{\operatorname{argmax}}\, p(D|w, f)$$

Однако на деле же работают не с вероятностью, а с функцией ошибки $\mathcal{L}(x, y, \omega)$, например MSE: $\mathcal{L}(x, y, w) = \frac{1}{n}\sum_{i=1}^{n}(y_i - w^T x_i)^2$

Давайте разберемся на примере линейной регрессии, почему такой переход возможен.

Итак, модель **линейной регрессии**:

$$y = f(w, x) + \epsilon, \;\; f(w, x) = w^T x, \;\; \epsilon \sim \mathcal{N}(0, \sigma^2)$$

Запишем модель в новом виде:

$$p(D|w, f) = p(y_i|x_i, w) =$$

$$\sum_{j=1}^{n} w_j x_{ij} + \mathcal{N}(0, \sigma^2) = \mathcal{N}(\sum_{j=1}^{n} w_j x_{ij}, \sigma^2)$$

**Максимизация правдоподобия**, то есть нахождение максимума ***p***, — это то же самое, что и максимизация его логарифма:

$$\log p(y_i|x_i, w) = \log \prod_{i=1}^{N} \mathcal{N}(\sum_{j=1}^{n} w_j x_{ij}, \sigma^2) =$$

$$\sum_{i=1}^{N} \log \mathcal{N}(\sum_{j=1}^{n} w_j x_{ij}, \sigma^2) = -\frac{N}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - w^T x_i)^2 \Leftrightarrow$$

$$\hat{w} = \underset{w \in R^W}{\operatorname{argmax}}\, p(D|w, f) = \underset{w \in R^W}{\operatorname{argmax}} -\mathcal{L}(x, y, w) = \underset{w \in R^W}{\operatorname{argmin}}\, \mathcal{L}(x, y, w)$$

Таким образом, мы свели **метод максимального правдоподобия** к минимизации среднеквадратической ошибки. Здесь мы пользовались предположением, что случайная величина распределена нормально.

**Обучающая и тестовая выборка. Скользящий контроль**

Мы уже вводили понятия признакового пространства и целевой переменной:

- Задана выборка значений признаков: $x_n$

$$\left\{x, \ldots, x_n \mid x \in R^d\right\}.$$

Здесь ***d*** — размерность признакового пространства и ***n*** — количество элементов в нашей выборке входных данных.

- Задана выборка соответствующих значений целевой переменной:

$$\left\{y, \ldots, y_n \mid y \in R\right\}.$$

Получаем множество исходных данных:

$$\mathrm{D} = \{(x, y)_i\}_{i=1}^{n}.$$

Таким образом, ***D*** — это вся выборка наших данных. Почему бы нам просто не обучить модель на всей этой выборке? В прошлом модуле мы уже говорили о важности данных для решения бизнес-задач с помощью машинного обучения. Чем больше данных, тем выше может быть получена точность модели. Здесь мы имеем в виду, что проверка модели происходит

на реальных данных, отличных от исходной выборки. В прошлый раз мы уже говорили о том, что для оценки работоспособности модели в реальной бизнес-задаче используется не одна метрика, являющаяся функцией потерь, а целая иерархия метрик, в которой наиболее значимые метрики основываются на внешних статистиках (например, на отзывах асессоров), а не на доступной выборке.

Однако в реальности $\boldsymbol{D}$ — это все доступные нам данные, и процесс оценки качества модели — ее *тестирование (или валидация)* — происходит только на выборке $\boldsymbol{D}$, как и процесс обучения модели. Поэтому перед нами встает задача разделения всей выборки на *обучающую и тестовую* части.

Но не все так просто. Ошибка на обучающей выборке является смещенной. Не вдаваясь в математику, модель отражает зависимости только на тех данных, на которых она обучалась. А вот на тестовых данных предположение о близости выхода модели к истинному значению уже не будет выполняться. На обучающей выборке мы получаем оптимистически заниженную функцию потерь и, соответственно, высокую точность предсказаний. В то же время на данных вне обучающей выборки (на тестовой) точность низкая. Такая проблема возникает при переобучении модели, о которой мы уже говорили в видеоролике.

Поэтому появляется необходимость разбиения всей выборки на обучающую и тестовую (валидационную) части определенным образом.

Первым соображением мы можем разбить нашу выборку на части случайно. Но в таком случае качество метрики не улучшится, и проблема переобучения не решится. Можно брать среднюю ошибку по такому разбиению. Метрика при этом также усредняется, но оценка останется смещенной. Чтобы получить *несмещенную среднюю ошибку*, будем использовать **скользящий контроль**:

1. Разбиваем множество исходных данных на две непересекающиеся подвыборки$\boldsymbol{J}$ различными способами:

$$D = \{D_i^k \cup D_i^m\}_{i=1}^J$$

Здесь $k$ — длина обучающей подвыборки и $m = N - k$ — длина тестовой подвыборки.

2. Для каждого разбиения из ***J*** обучаем модель и получаем метрику. Затем считаем среднее по всем полученным ***J*** значениям метрики.

Мы рассмотрим подробно один из вариантов скользящего контроля, а именно **кросс-валидацию**:

1. Делим выборку на **k блоков (folds)**, непересекающихся и равных по объему

2. Производим **k** итераций:

a. обучаем заранее заданную модель на **k – 1**фолде

b. тестируем модель на **1**фолде, не участвующем в обучении, и считаем метрику $q_i$ на ***i***-йитерации

c. смещаем инициализацию фолдов для обучения на **1**

3. Считаем среднее по полученным метрикам:

$$q = \frac{1}{k}\sum_{i=1}^{k} q_i$$

На рисунке ниже показан пример инициализации фолдов для кросс-валидации.

Первый фолд**F[0]** будет тестовым, остальные k – 1 фолдов пойдут в обучающую выборку **F[1 : k – 1]**. После итерации обучения и тестирования назначаем следующий фолд в качестве тестового — **F[1]**, а для обучающей выборки — фолды**F[2 : k – 1] + F[0]**. Затем снова итерация обучения + тестирования и так далее.