



Тверской  
государственный  
технический  
университет

# Интеллектуальные информационные системы

## Вводная лекция

Шпигарь Андрей Николаевич

Материалы курса доступны по ссылке:  
<https://github.com/AndreyShpigar/ML-course>

2025 г.

# Чего не будет в курсе?

- × Сложной математики
- × Написания кода
- × Exploratory data analysis
- × Домашних заданий



# Что будет в курсе?

- ✓ Не очень сложная математика
- ✓ Обзор основных методов ML и DL
- ✓ Примеры прикладных задач
- ✓ Реализация алгоритмов



# Содержание курса

- L00 – Math refresher
- L01 – Вводная лекция
- L02 – Линейные модели в задачах регрессии
- L03 – Линейные модели классификации. Байесовские методы классификации
- L04 – Оценка качества моделей. Выбор модели
- L05 – Метод опорных векторов
- L06 – Решающие деревья
- L07 – Ансамблевые методы
- L08 – Развитие ансамблевых методов
- L09 – Отбор признаков и понижение размерности
- L10 – Искусственные нейронные сети
- L11 – Сверточные нейронные сети
- L12 – Рекуррентные нейронные сети
- L13 – Модели внимания и трансформеры
- L14 – Генеративные модели
- L15 – Интерпретируемость и объяснимость в машинном обучении

# Определение интеллектуальных информационных систем.

## Основные понятия и определения.

- **Интеллект** – мыслительная способность человека.
- **Мышление** – способность человека с помощью размышлений и последовательных мыслительных действий получать желаемые результаты.
- **Искусственный интеллект** – создание вычислительной системы, имитирующей человеческие навыки обработки информации.
- **Данные** – совокупность объективных сведений.
- **Информация** – сведения, неизвестные ранее получателю информации, пополняющие его знания, подтверждающие или опровергающие положения и соответствующие убеждения.
- **Знания** – совокупность факторов, закономерностей и эвристических правил, с помощью которых решается поставленная задача.
- **Интеллектуальная информационная система** – модель интеллектуальных возможностей человека в целенаправленном поиске, анализе и синтезе текущей информации об окружающей действительности для получения о ней новых знаний и решения на этой основе различных задач.

# Классификация интеллектуальных систем

## Интеллектуальные информационные системы

- Интеллектуальный интерфейс
- Экспертные системы
- Самообучающиеся системы
- Адаптивные системы

***Системы с интеллектуальным интерфейсом*** включают в себя:

1. Естественно языковой интерфейс
2. Интеллектуальные базы данных
3. Гипертекстовые системы
4. Системы контекстной помощи
5. Когнитивная графика

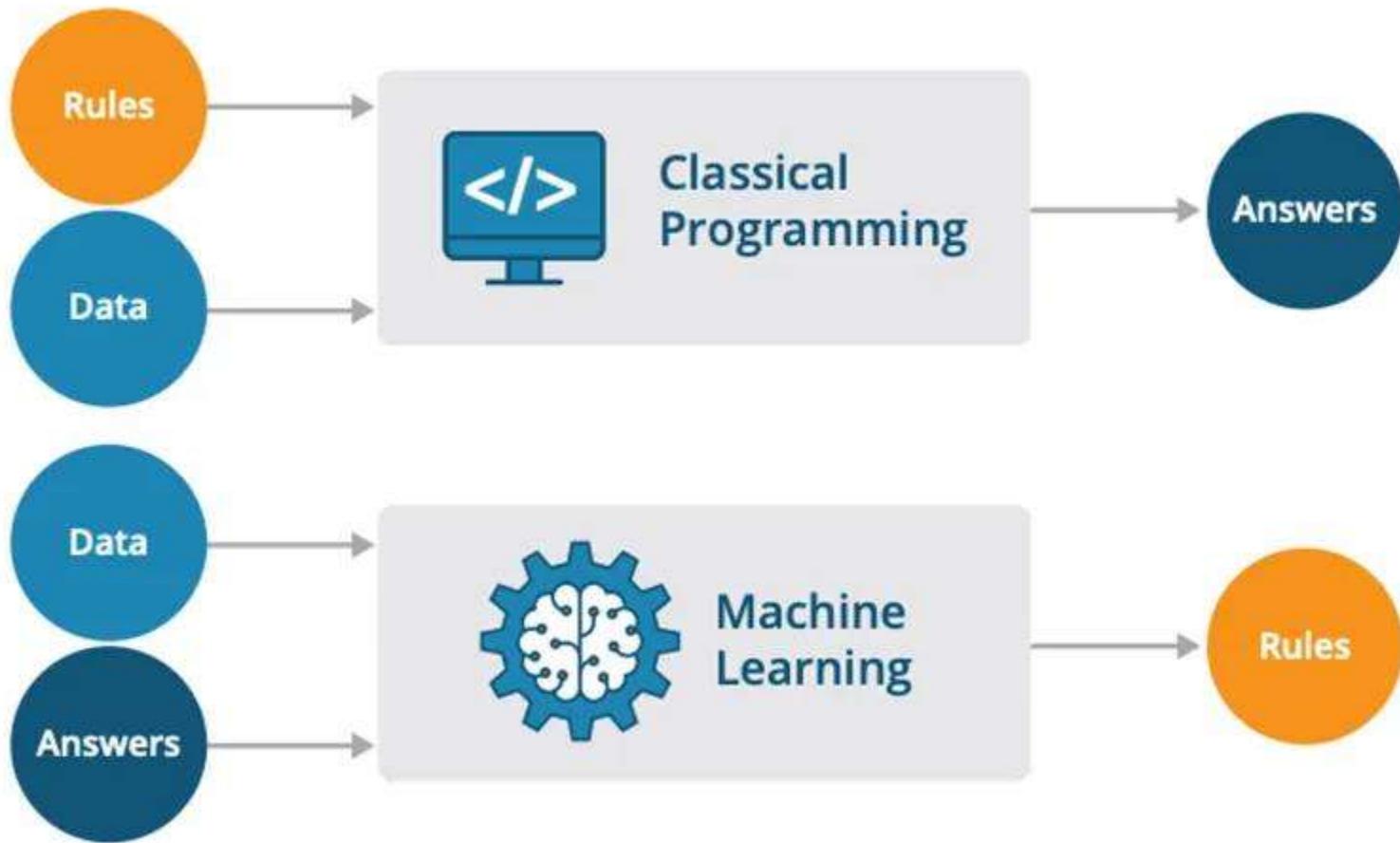
***Экспертные системы*** – решают задачи на основе накапливаемой базы знаний, отражающей опыт работы экспертов в некоторой проблемной области. Включают в себя:

1. Классифицирующие системы
2. Доопределяющие системы
3. Трансформирующие системы
4. Многоагентные системы

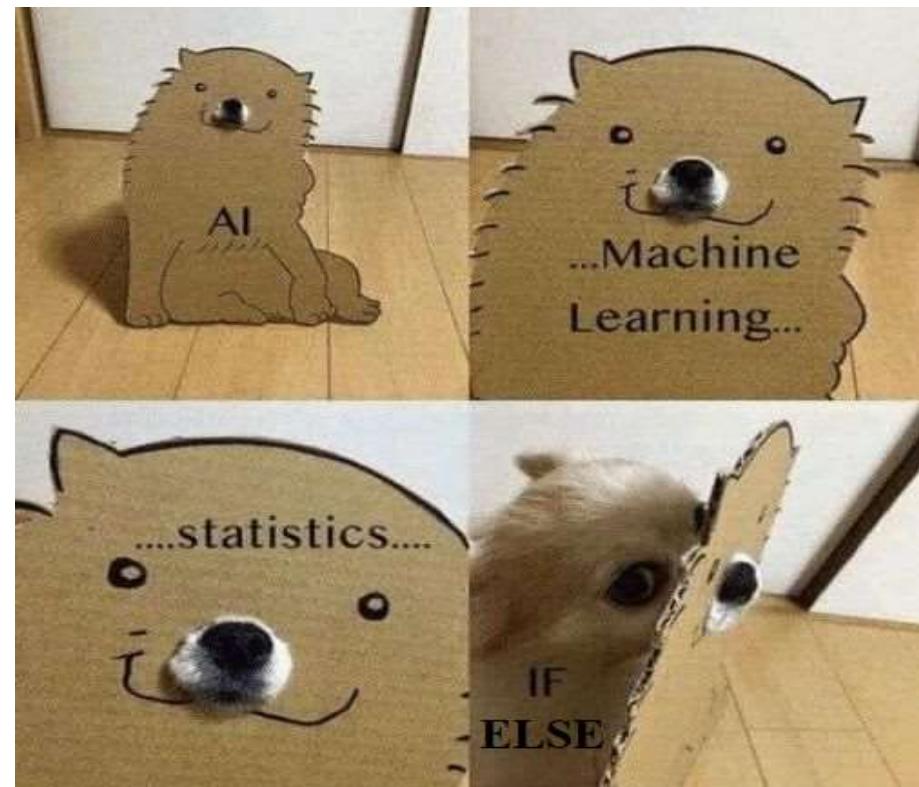
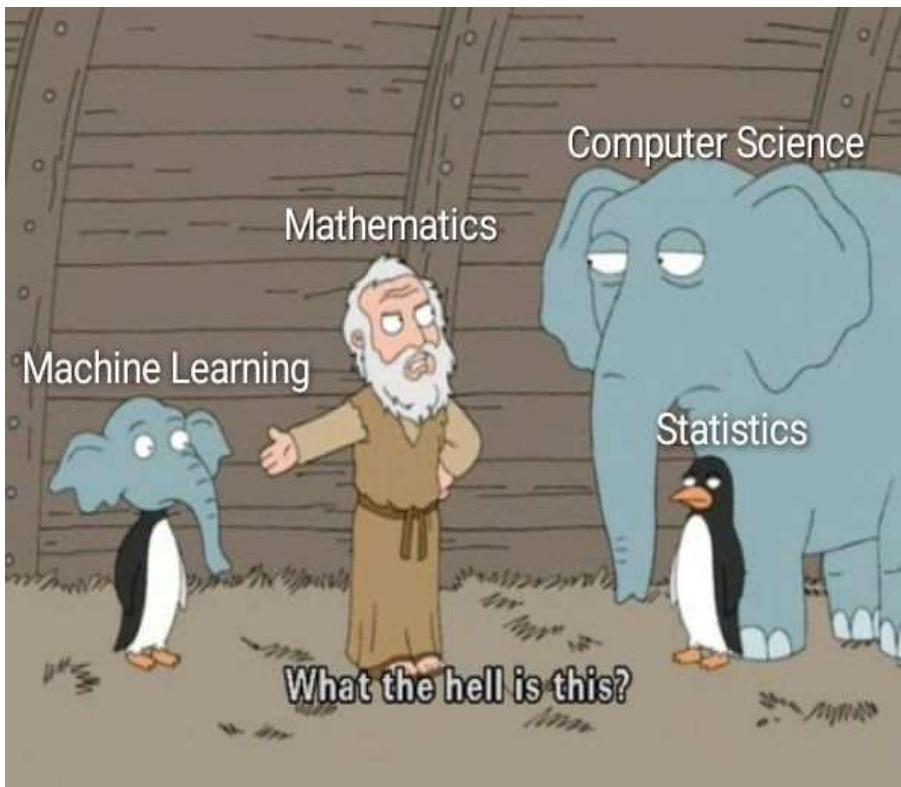
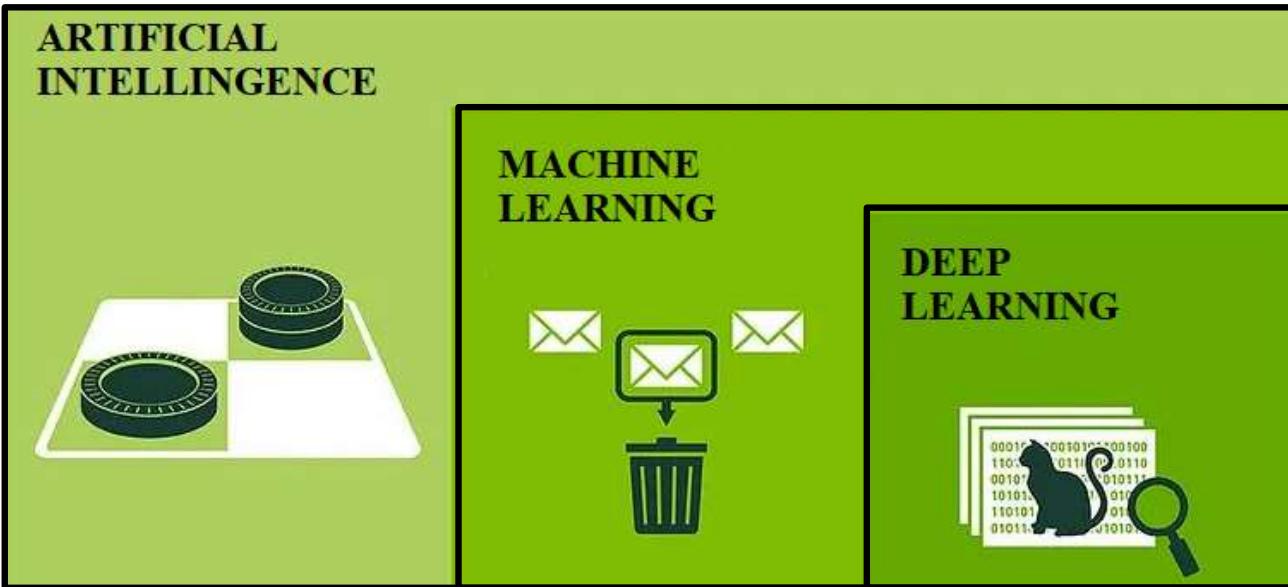
***Самообучающиеся системы*** – основаны на методах автоматической классификации примеров ситуаций реальной практики (обучение на примерах)

***Адаптивные системы*** – основаны на постоянно развивающейся модели проблемной области, поддерживаемой в базе знаний, на основе которой осуществляется генерация или конфигурация программного обеспечения

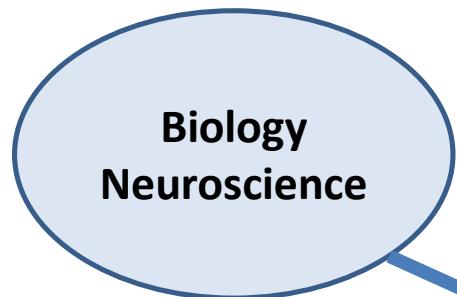
# Концепция машинного обучения



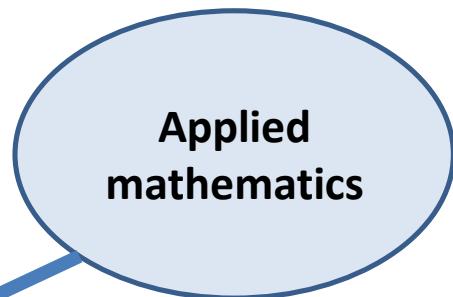
## ARTIFICIAL INTELLIGENCE



Парадигма обучения  
Принцип работы нейрона



Линейная алгебра  
Методы оптимизации  
Математический анализ



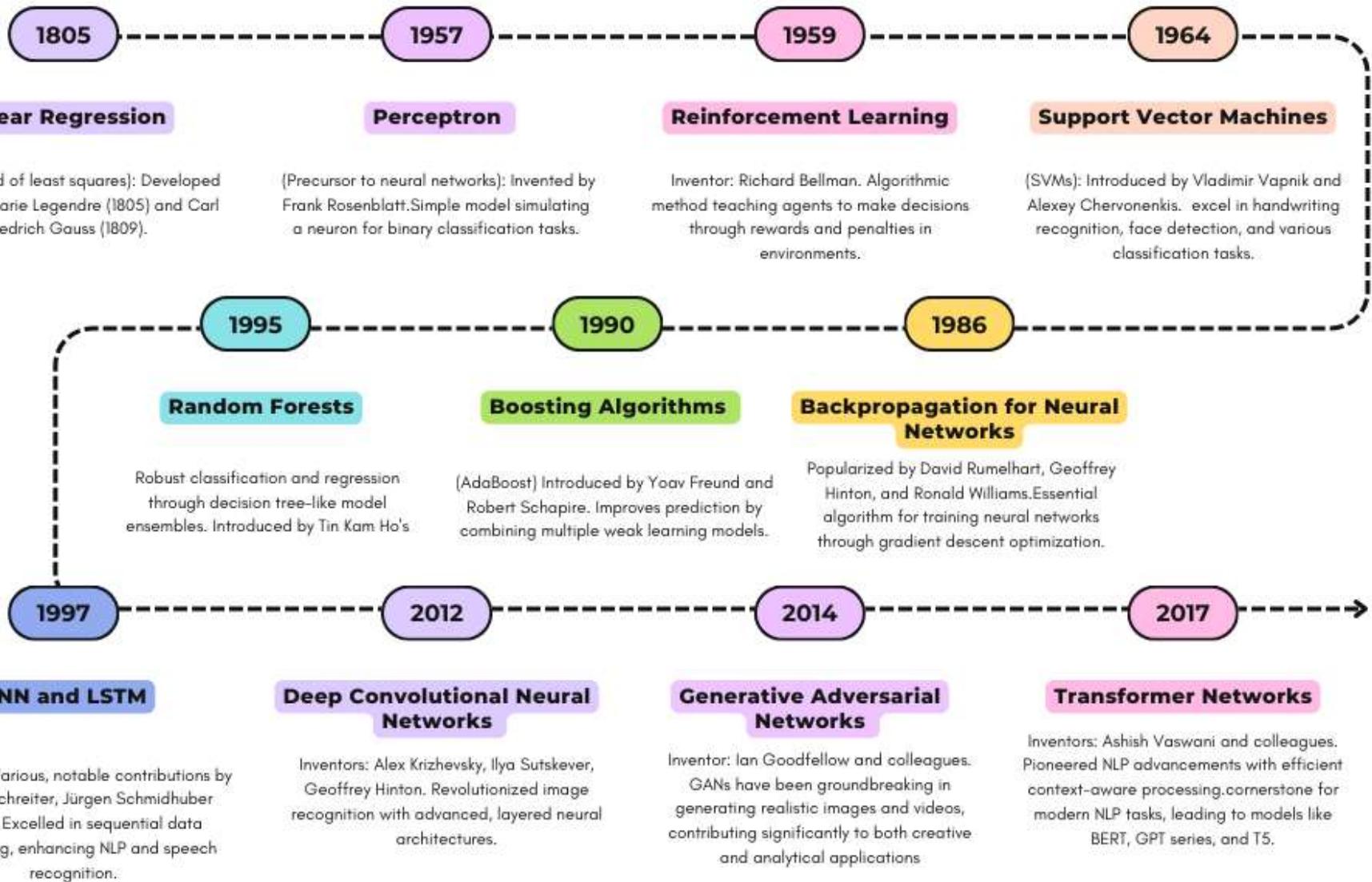
Алгоритмы  
Структура данных  
Языки программирования



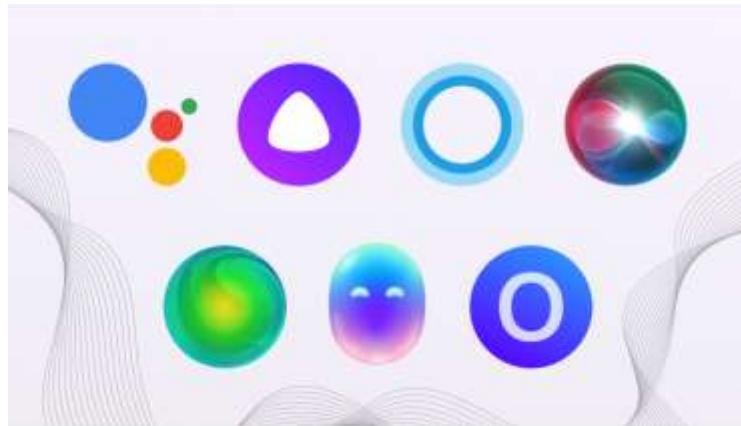
Теория вероятностей  
Статистический анализ  
Теория оценивания  
Проверка статистических гипотез



# Развитие машинного обучения



# Машинное обучение сегодня



# Лауреаты Нобелевской премии-2024

## ФИЗИКА

🇺🇸 Джон Хопфилд  
🇬🇧 🇨🇦 Джеффри Хинтон

за исследования в области машинного обучения нейросетей



## ХИМИЯ

🇺🇸 Дэвид Бейкер  
за разработку вычислительных инструментов, позволяющих создавать новые белки

🇬🇧 Демис Хассabis  
🇺🇸 🇬🇧 Джон Джампер  
за методы предсказания сложных структур белков по их аминокислотным последовательностям



## МЕДИЦИНА

🇺🇸 Виктор Эмбрюс  
🇺🇸 Гэри Равкан

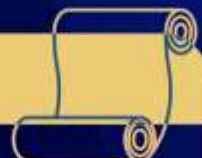
за открытие микроРНК и ее роли в посттранскрипционной регуляции генов



## ЛИТЕРАТУРА

🇰🇷 Хан Ган

за «насыщенную поэтическую прозу, которая смотрит в лицо исторической травме и обнажает хрупкость человеческой жизни»



## ЭКОНОМИКА

🇹🇷 🇺🇸 Дарон Аджемоглу

🇬🇧 🇬🇧 Саймон Джонсон

🇬🇧 🇺🇸 Джейсон Робинсон

за изучение причин различий в уровне благосостояния между странами



## ПРЕМИЯ МИРА

🇯🇵 Нихон Хиданкё

за усилия по достижению мира, свободного от ядерного оружия



# Несколько примеров прикладных задач

## Регрессия:

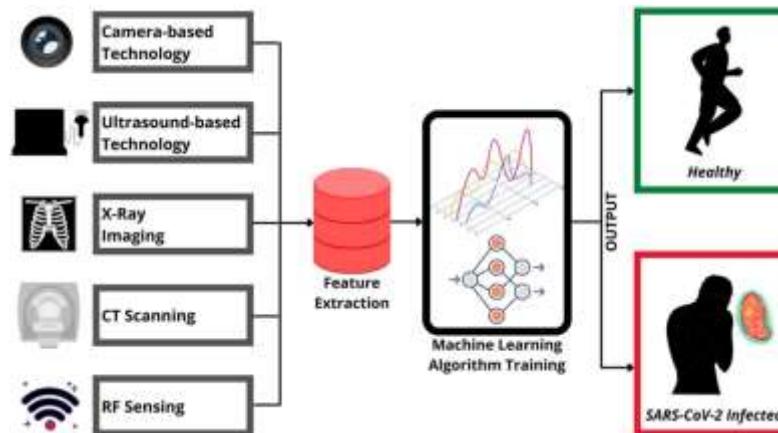
- Экономика и финансы – прогнозирование потребительского спроса. Необходимо оценить объемы продаж для каждого товара на заданный интервал времени. На основе этих прогнозов осуществляется планирование закупок и формирование ценовой политики
- Рекомендательные системы – задача предсказания рейтинга, товара или услуги. Приобретая товар или услугу, клиент может оценить ее, например, от 1 до 5. Система использует информацию о всех выставленных рейтингах для персонализации предложений. Основная задача – прогнозировать рейтинг товаров, которые клиент еще не приобрел

Задача ранжирования – ранжирование документов при поиске по запросу пользователя

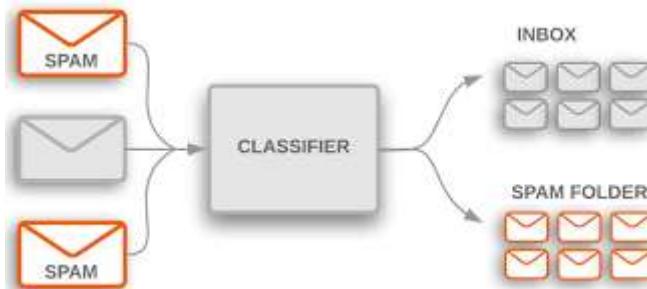
# Несколько примеров прикладных задач

## Классификация:

- Медицина – определить, болен пациент или нет. Признаками могут быть результаты обследований, симптомы заболевания и прочие (анамнез)



- Информационная безопасность – классификация спама, обнаружение мошеннических транзакций



# Несколько примеров прикладных задач

## Классификация:

- Экономика и финансы – задача оценивания заемщика банками.  
«Хороший-плохой» заемщик, подсчет количества кредитных баллов – credit scoring. Задача минимизации риска невозврата кредита



- Задача предсказания оттока клиентов (churn prediction) – выделение сегмента клиентов, склонных к уходу в ближайшее время

# Несколько примеров прикладных задач

## Классификация:

- Классификация изображений



Car



Tiger



Dog



Car



Tiger



Tiger



Dog

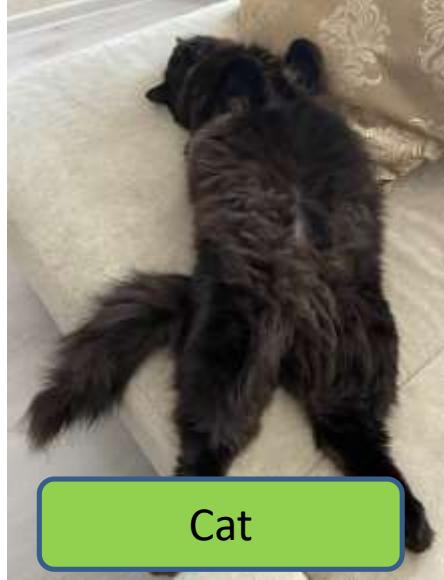
# Несколько примеров прикладных задач

## Классификация:

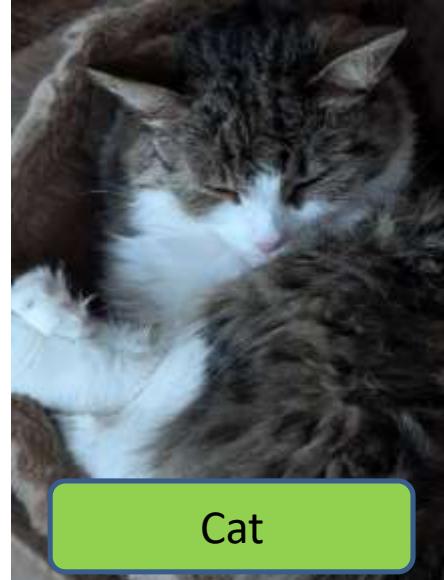
- Классификация изображений



Person



Cat



Cat



Cat



Person



Person

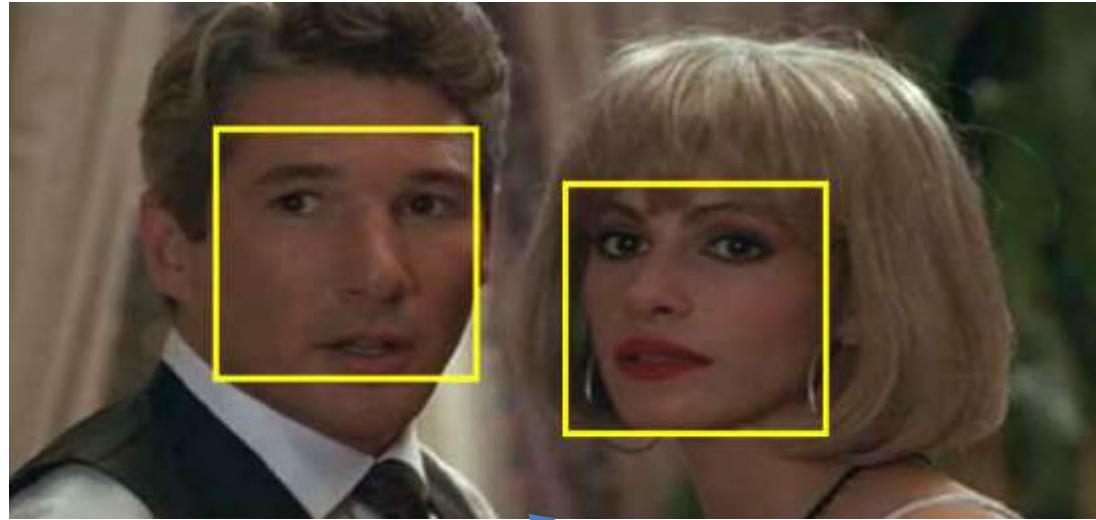


Cat



Person

# Несколько примеров прикладных задач



**Classification**



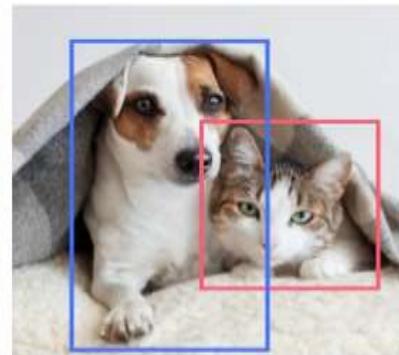
Cat

**Classification + Localization**



Cat

**Object Detection**



Cat, Dog

**Instance Segmentation**



Cat, Dog

Single object

Multiple objects

# Виды машинного обучения

- *Обучение с учителем (supervised learning)*
- *Обучение без учителя (unsupervised learning)*
- *Обучение с подкреплением (reinforcement learning)*

# Виды машинного обучения

- ***Обучение с учителем (supervised learning)***

$X$  – множество объектов, описанные в пространстве признаков ( $f$  – признак объекта, feature)

$Y$  – множество ответов (оценок, предсказаний, прогнозов)

$y^*: X \rightarrow Y$  – неизвестная целевая функция (target function), значения которой известны только на конечном подмножестве объектов

$$\{x_1, \dots, x_l\} \subset X$$

Пары «объект – ответ»  $\{x_i, y_i\}$  называются прецедентами (обучающими примерами)

Совокупность обучающих примеров  $X^l = (x_i, y_i)_{i=1}^l$  называется обучающей выборкой (dataset)

Задача обучения с учителем: по выборке  $X^l$  восстановить зависимость  $y^*$ , то есть построить решающую функцию  $a: X \rightarrow Y$ , которая бы приближала целевую функцию  $y^*(x)$ , причем не только на объектах обучающей выборке, но и на всем множестве  $X$

# Признаковое описание объектов

Признак  $f$  объекта  $x$  – результат измерения некоторой характеристики объекта. Признак это отображение  $f: X \rightarrow D_f$ , где  $D_f$  - множество допустимых значений признака

Вектор  $(f_1(x), \dots, f_n(x))$  – признаковое описание объекта

Типы признаков:

- Если  $D_f = \{0, 1\}$ , то  $f$  – *бинарный* признак
- Если  $D_f$  - конечное множество, то  $f$  – *категориальный* признак
- Если  $D_f$  - конечное упорядоченное множество, то  $f$  – *порядковый* признак
- Если  $D_f = \mathbb{R}$ , то  $f$  – *количественный* признак

*Матрица объектов-признаков (design matrix, матрица плана):*

$$F = \|f_i(x_i)\|_{l \times n} = \begin{pmatrix} f_1(x_1) & \dots & f_n(x_1) \\ \dots & \dots & \dots \\ f_1(x_l) & \dots & f_n(x_l) \end{pmatrix}$$

# Типы задач

Задача классификации (*classification*):

- $Y = \{-1, +1\}$  – классификация на 2 класса (binary classification)
- $Y = \{1, \dots, M\}$  – на М непересекающихся классов (multi-class classification)
- $Y = \{0,1\}^M$  - на М классов, которые могут пересекаться (multi-label classification)

Задача регрессии (*regression*):

- $Y = \mathbb{R}$  или  $Y = \mathbb{R}^m$

Задача ранжирования (*ranking, learning to rank*):

- $Y$  – конечное упорядоченное множество

Частичное обучение (*semi-supervised learning*) – задача, в которой для одной части объектов обучающей выборки известны и признаки, и ответы, а для другой только признаки

# Пример датасета – табличные данные

Имя	Возраст	Пол	Оценка за экзамен «Статистика»	Оценка за экзамен «Математический анализ»	Оценка за экзамен «Программирование»	Количество пропущенных лекций по предмету «ИИС»	Оценка за экзамен «ИИС»
Андрей	20	М	3	3	3	2	3
Антон	20	М	5	5	5	0	5
Анастасия	21	Ж	4	4	3	0	4
Ирина	20	Ж	3	3	3	15	2
Илья	22	М	4	4	4	7	3
Глеб	20	М	3	5	5	4	

признаки

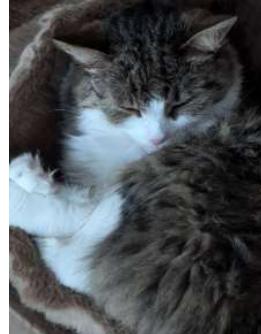
целевая переменная  
(таргет)

- **Обучение без учителя (*unsupervised learning*)**

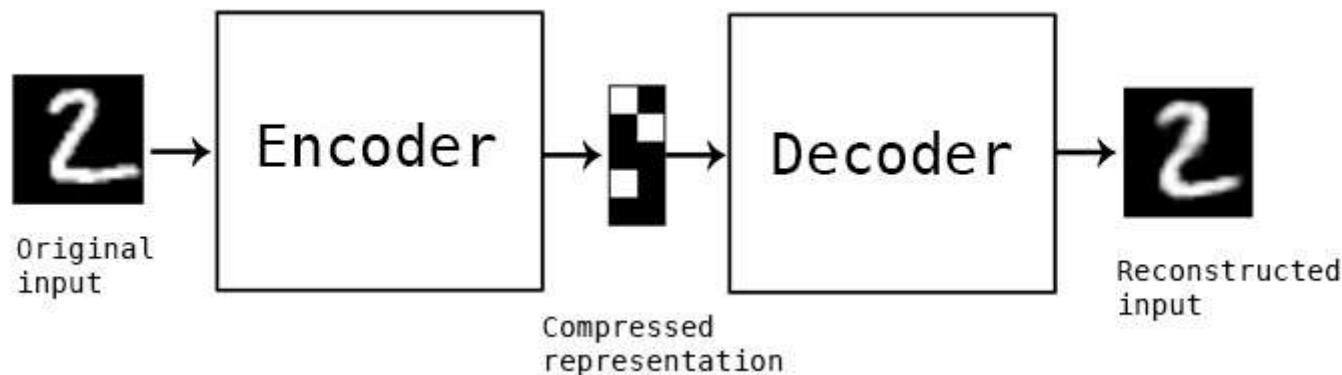
Класс задач, где ответы неизвестны или вообще не существуют, требуется найти некоторые закономерности на основе признаковых описаний



1. Кластеризация – задача разделения выборки на подмножества (кластеры) так, чтобы каждый кластер состоял из похожих объектов, а объекты разных кластеров существенно отличались



2. Оценивание плотности – задача приближения распределения объектов. Пример – обнаружение аномалий, в которой на этапе обучения известны лишь примеры корректной работы оборудования, в дальнейшем требуется обнаруживать случаи некорректной работы
3. Понижение размерности – задача генерации новых признаковых описаний меньшей размерности без потери качества модели (либо с незначительными потерями)



- **Обучение с подкреплением (reinforcement learning)**

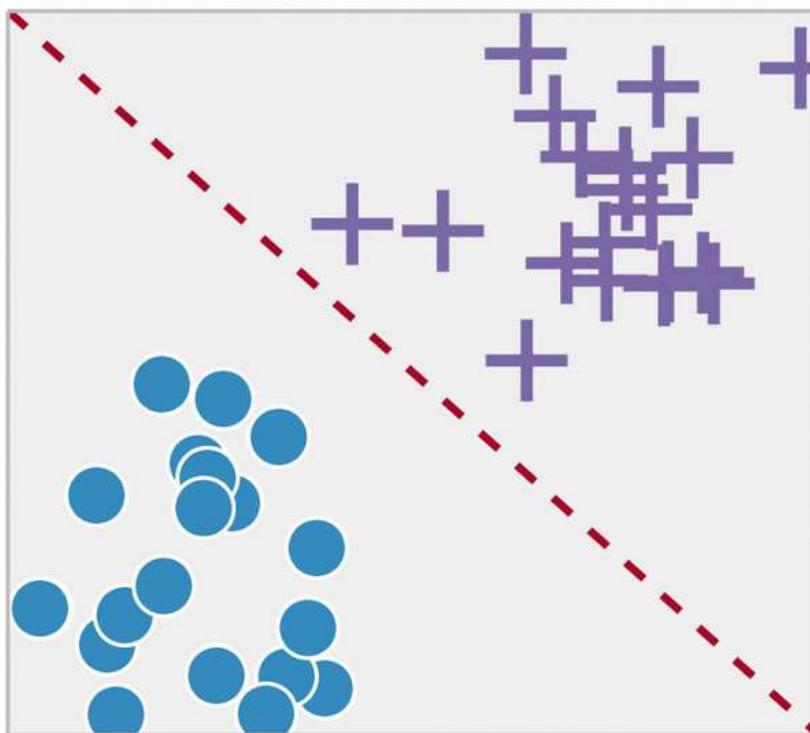
Алгоритм на каждом шаге наблюдает какую-то ситуацию, выбирает одно из доступных ему действий, получает некоторую награду и корректирует свою стратегию. Задачей алгоритма является максимизация некоторой функции награды

Области применения – робототехника, игры, управление транспортом и другие

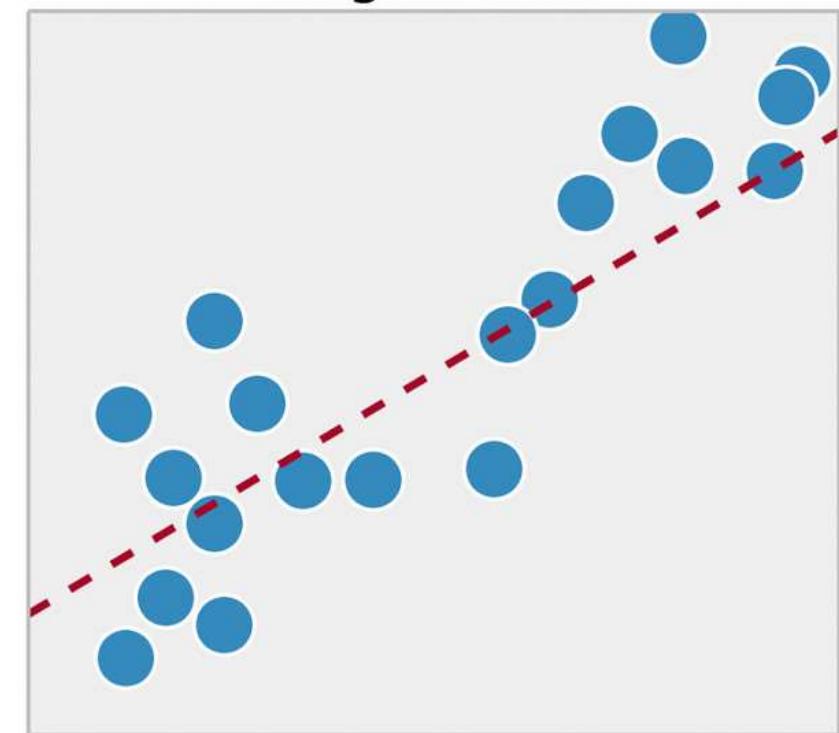


# Визуализация двух основных задач обучения с учителем – классификации и регрессии

Classification



Regression



# Модель алгоритмов и метод обучения

*Модель алгоритмов* – параметрическое семейство отображений  $A = \{g(x, \theta) \mid \theta \in \Theta\}$ , где  $g: X \times \Theta \rightarrow Y$  некоторая фиксированная функция,  $\Theta$  – множество допустимых значений параметра  $\theta$ , называемое пространством параметров или пространством поиска (search space)

*Метод обучения (learning algorithm)* – отображение  $\mu: (X \times Y)^l \rightarrow A$ , которое произвольной конечной выборке  $X^l = (x_i, y_i)_{i=1}^l$  ставит в соответствие некоторый алгоритм  $\alpha \in A$ . Метод  $\mu$  строит алгоритм  $\alpha$  по выборке  $X^l$

# Два основных этапа машинного обучения

- Этап *обучения (train)* – на этапе обучения метод  $\mu$  по выборке  $X^l$  строит алгоритм  $\alpha = \mu(X^l)$ . Этап обучения сводится к поиску параметров модели, обеспечивающих оптимальное значение заданному функционалу качества. На этапе обучения метод выдает элемент параметрического семейства функций. *Необходимо оптимизировать вектор параметров модели*
- Этап *применения (test)* – алгоритм  $\alpha$  для новых объектов  $x$  выдает ответы  $y = \alpha(x)$

# Функционал качества

Функция потерь (*loss function*) –  $L(a, x)$  – величина ошибки алгоритма  $a \in A$  на объекте  $x \in X$

Функционал качества алгоритма  $a$  на выборке  $X^l$ :

$$Q(a, X^l) = \frac{1}{l} \sum_{i=1}^l L(a, x_i)$$

Функционал  $Q$  так же называют функционалом средних потерь или эмпирическим риском, так как он вычисляется по эмпирическим данным  $(x_i, y_i)_{i=1}^l$

Классический метод обучения, называемый минимизацией эмпирического риска (empirical risk minimization, ERM), заключается в том, чтобы найти в заданной модели  $A$  алгоритм  $a$ , обеспечивающий минимальное значение функционалу качества  $Q$  на заданной обучающей выборке  $X^l$ :

$$\mu(X^l) = \arg \min Q(a, X^l)$$

- Функции потерь для задач классификации:

$$L(a, x) = [a(x) \neq y(x)] - \text{индикатор ошибки}$$

- Функции потерь для задач регрессии:

$$L(a, x) = |a(x) - y(x)| - \text{абсолютное значение ошибки}$$

$$L(a, x) = (a(x) - y(x))^2 - \text{квадратичная ошибка}$$

#### Примечание:

Функция потерь (*loss*) оценивает, как часто модель ошибается. Функция потерь оказывает существенное влияние на метод машинного обучения. Важно, чтобы ее было легко оптимизировать, например, гладкая функция потерь – это хорошо, а кусочно-постоянная – плохо. Существует большое количество функций потерь, выбор конкретной функции зависит от многих факторов: тип задачи, тип модели, специфика данных и другие. Различные функции потерь и их особенности будут рассмотрены в следующих лекциях

# Вероятностная постановка задачи обучения

Неизвестное вероятностное распределение на множестве  $X \times Y$  с плотностью  $p(x, y)$  из которого случайно и независимо выбираются  $l$  наблюдений

$$X^l = (x_i, y_i)_{i=1}^l$$

Свойство i.i.d. (independent and identically-distributed) -  
**независимые одинаково распределенные**

**Функция правдоподобия (likelihood):**

$$L(\theta, X^l) = \prod_{i=1}^l \varphi(x_i, y_i, \theta)$$

Минимизация **логарифма** функции правдоподобия:

$$-\ln L(\theta, X^l) = -\sum_{i=1}^l \ln \varphi(x_i, y_i, \theta) \rightarrow \min$$

# Overfitting vs Underfitting

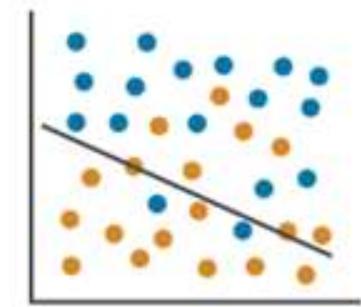
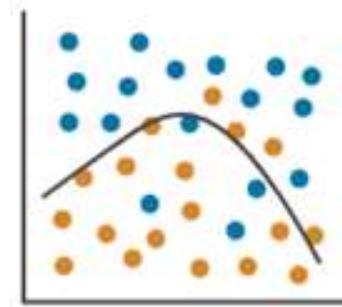
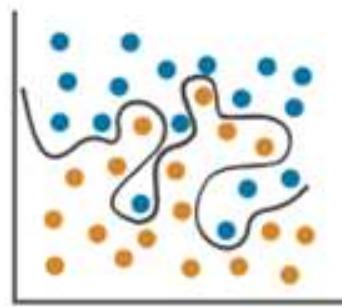
Минимизация эмпирического риска функционала  $Q(\alpha, X^l)$  не гарантирует, что  $\alpha$  будет хорошо приближать целевую зависимость на произвольном наборе объектов  $X^k = (x'_i, y'_i)_{i=1}^k$  - контрольной (тестовой) выборке

Предположение: выборки  $X^l$  и  $X^k$  - простые, полученные из одного и того же неизвестного вероятностного распределения на множестве  $X$

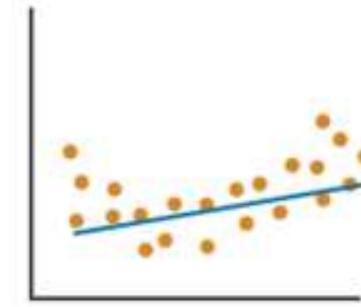
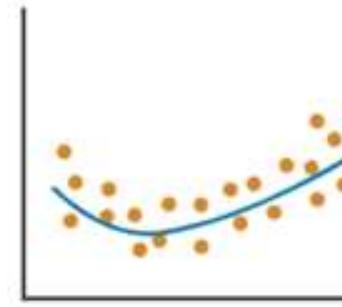
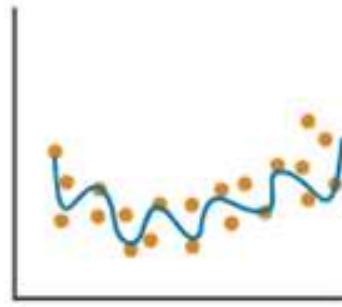
Объекты независимые одинаково распределенные (свойство i.i.d – independent and identically-distributed)

- Переобучение модели (*overfitting*) – эффект, когда оценка качества работы алгоритма на тестовой выборке  $X^k$  существенно хуже, чем на обучающей выборке  $X^l$

Classification



Regression

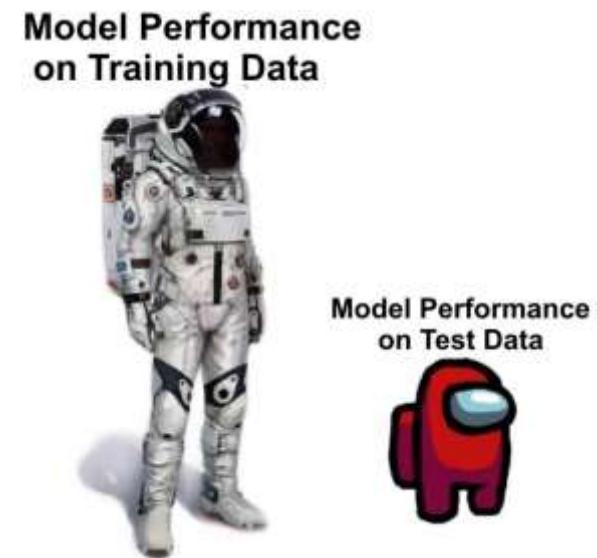
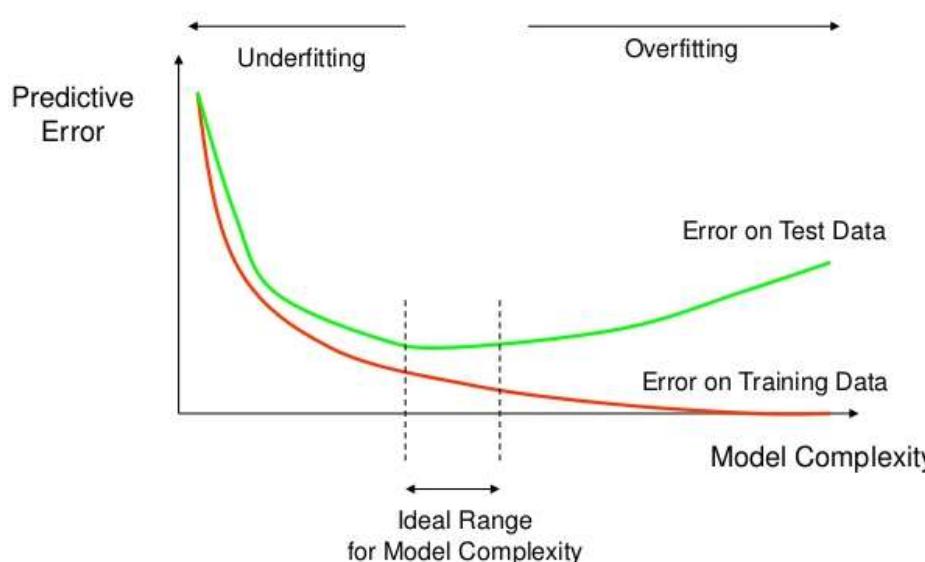


# Из-за чего возникает переобучение?

- Избыточная параметризация модели, модель слишком «сложная»
- Малая обучающая выборка, низкое качество данных в обучающей выборке (большое количество пропусков, шумов, выбросов и т.д.)

## Как понять, что модель переобучилась?

- Эмпирически – разбивать выборку на *train* и *test*, оценивать качество работы алгоритма них



# Как минимизировать переобучение?

- Увеличить размер и улучшить качество обучающей выборки
- Наложить ограничение на значения параметра  $\theta$  - регуляризация
- Выбор модели (model selection) по оценкам обобщающей способности (generalization performance)

# Эмпирические оценки обобщающей способности

- Метод hold-out: простое разделение на train и test
  - эмпирический риск на тестовых данных

$$HO(\mu, X^l, X^k) = Q(\mu(X^l), X^K) \rightarrow \min$$

- Метод leave-one-out: каждый объект выборки выбирается как тестовый (скользящий контроль)

$$LOO(\mu, X^L) = \frac{1}{L} \sum_{i=1}^L L(\mu(X^L \setminus \{x_i\}), x_i) \rightarrow \min$$

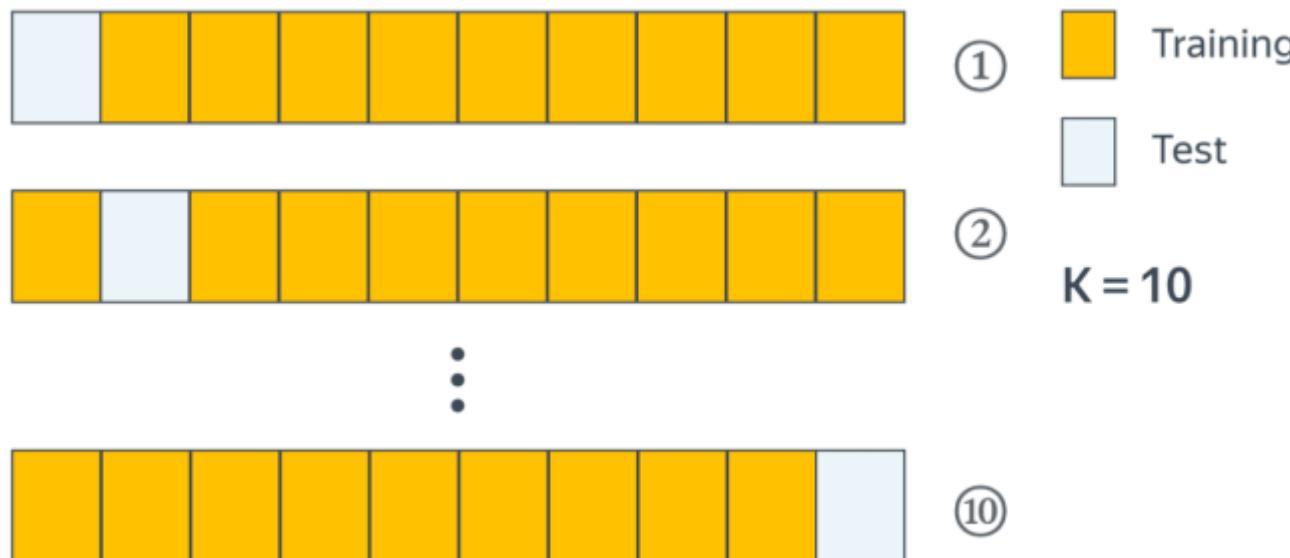
# Эмпирические оценки обобщающей способности

- Кросс-проверка (*cross-validation*): разбиение обучающей выборки на  $k$  одинаковых частей (фолдов), каждая из которых по очереди выступает в роли тестовой выборки

$$CV(\mu, X^L) = \frac{1}{|P|} \sum_{p \in P} Q(\mu(X_p^l), X_p^k) \rightarrow \min$$

$P$  – множество разбиений  $X^L = X_p^l \sqcup X_p^k$

1. Фиксируется некоторое целое число  $k$  (обычно от 5 до 10), меньшее числа объектов в обучающей выборки
2. Выборка разбивается на  $k$  одинаковых частей (фолдов) – отсюда название *k-Fold cross-validation*
3. Выполнение  $k$  итераций, во время каждой из которых один фолд выступает в роли тестовой выборки, а остальные в роли обучающей



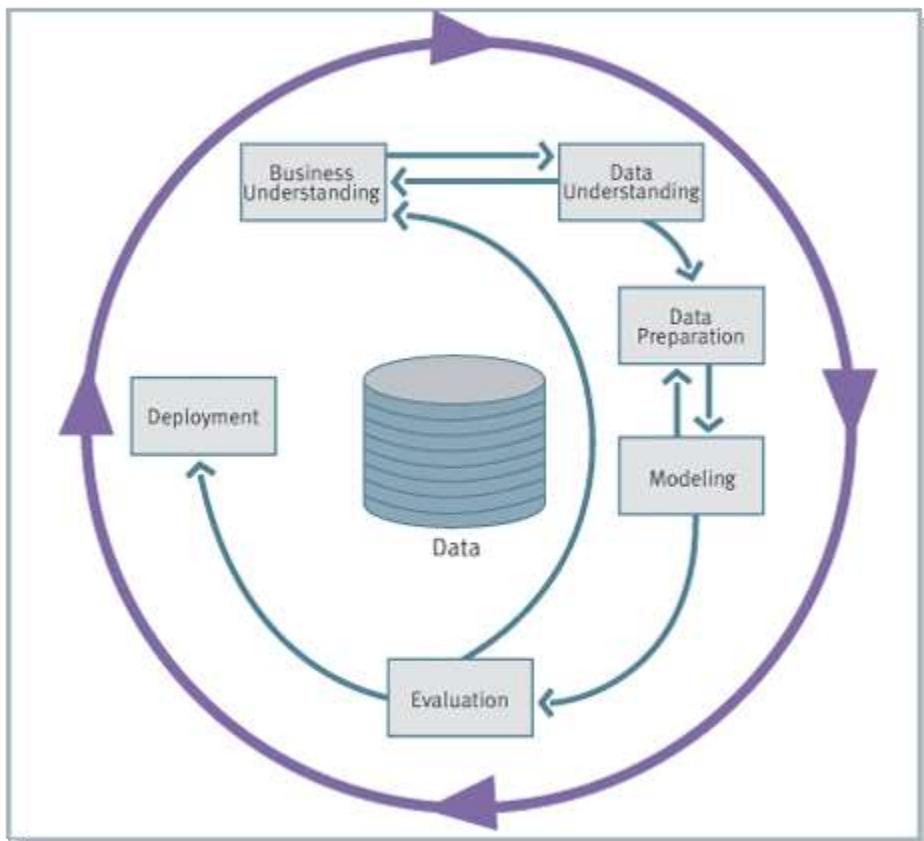
# Особенности данных и постановок прикладных задач

Данные могут быть:

- Разнородные (признаки измерены в разных шкалах)
- Неполные (измерены не все, имеются пропуски)
- Неточные (погрешности измерений, шум)
- Противоречивые (объекты одинаковые, ответы разные)
- Избыточные (огромное количество данных, не понятно, необходимо ли использовать все, тяжело обрабатывать)
- Недостаточные (объектов меньше, чем признаков)
- Неструктурированные (нет признаковых описаний, признаковые описания сильно различаются)

# Межотраслевой стандарт интеллектуального анализа данных

CRISP-DM: CRoss Industry Standard  
Process for Data Mining (1999)

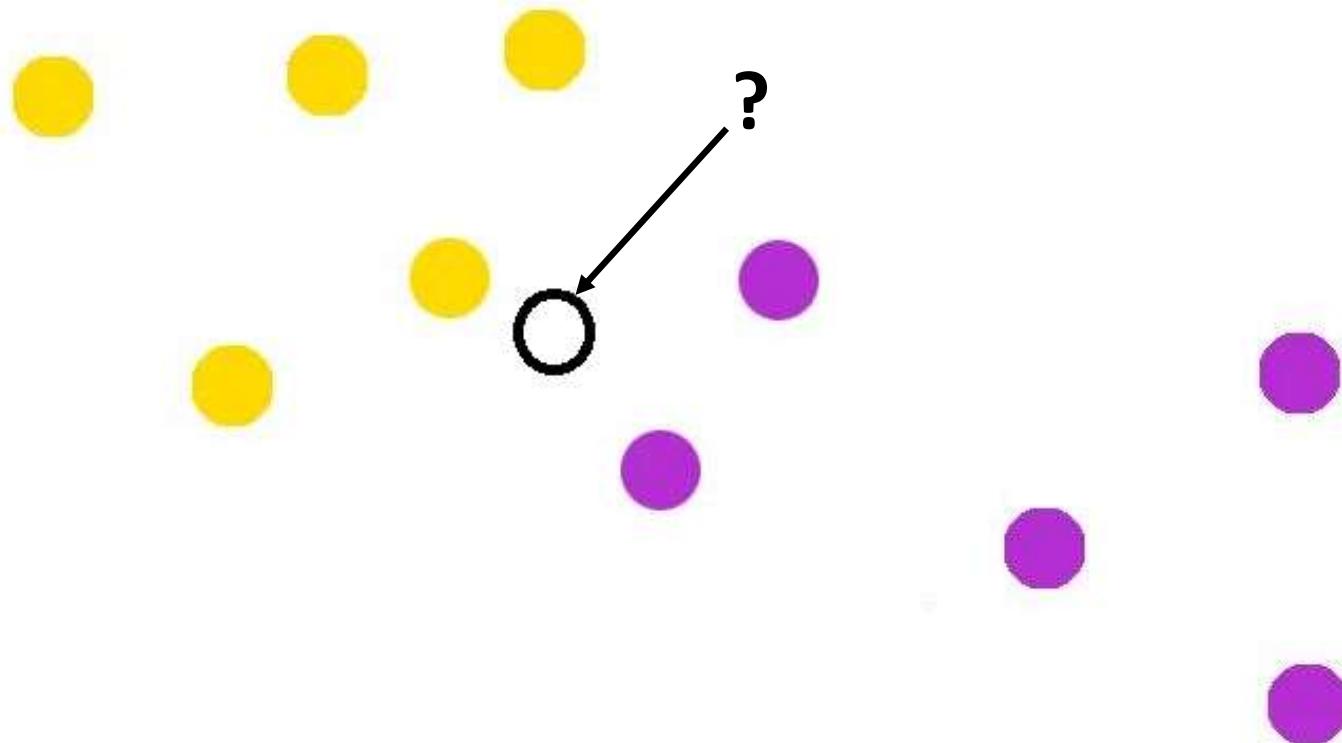


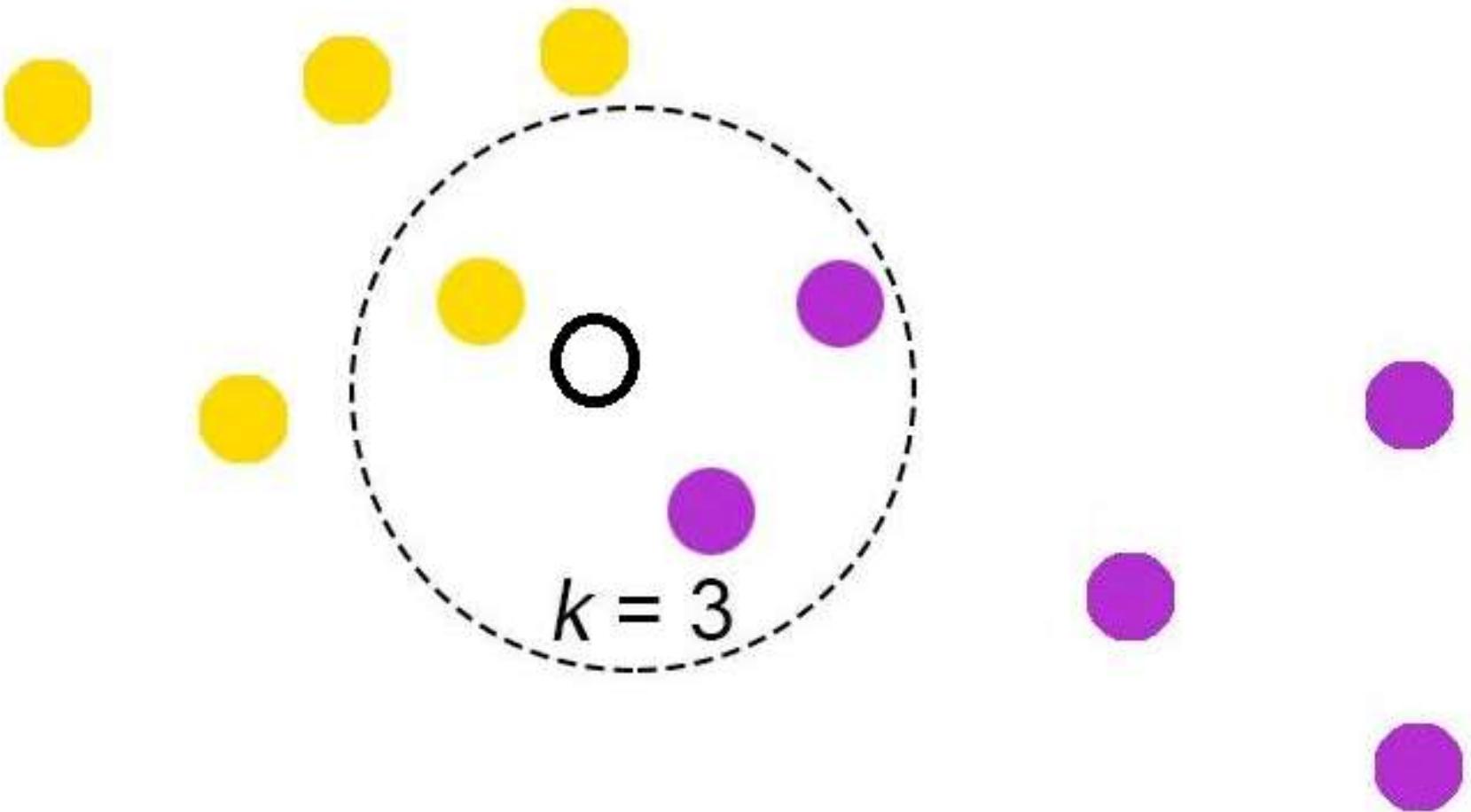
- Понимание бизнеса
- Понимание данных
- Предобработка данных и инженерия признаков
- Разработка моделей и настройка параметров
- Оценивание качества
- Внедрение

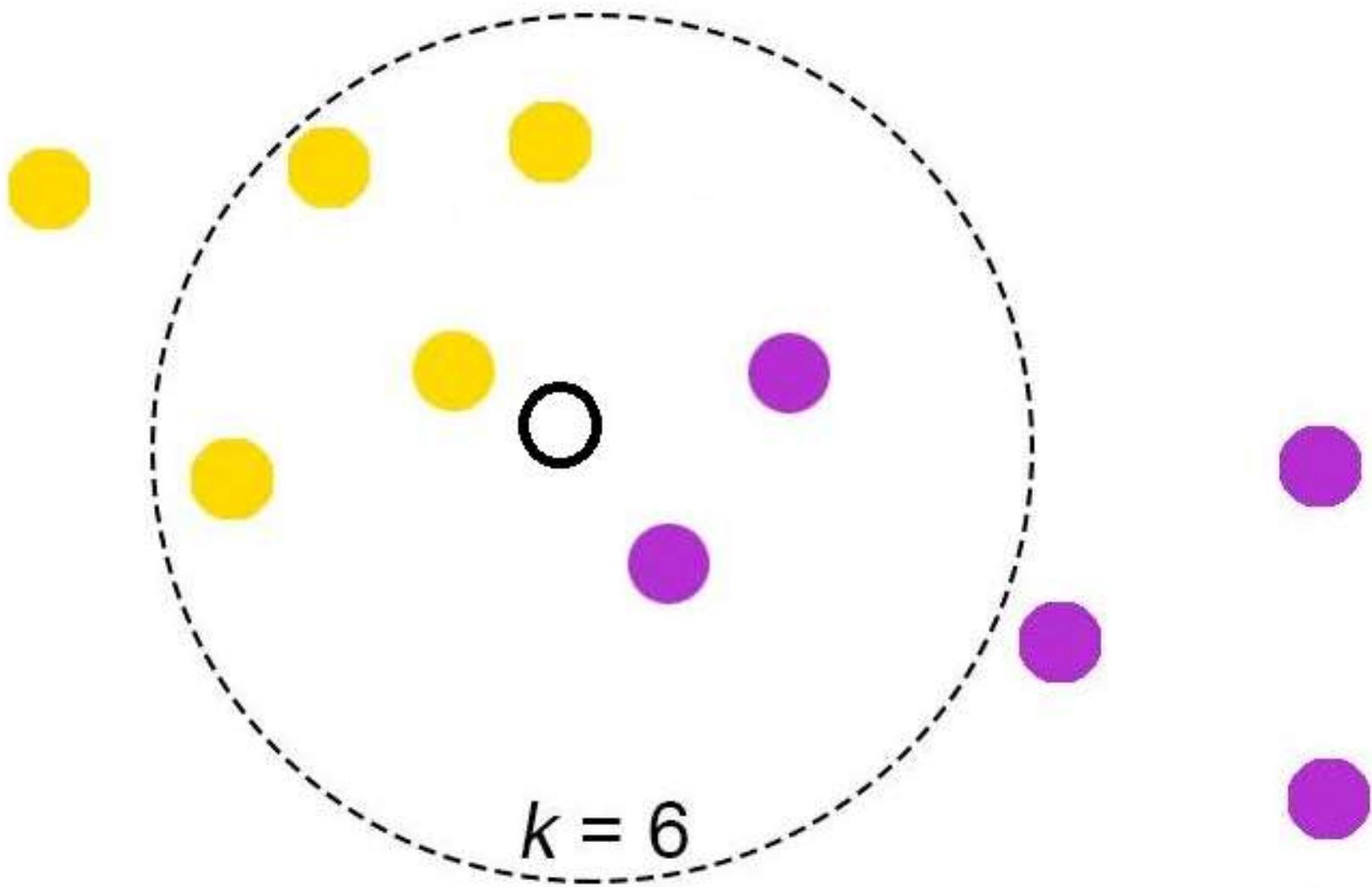
# Метрические методы

## Метод k-ближайших соседей (KNN)

Задача – какого цвета объект?

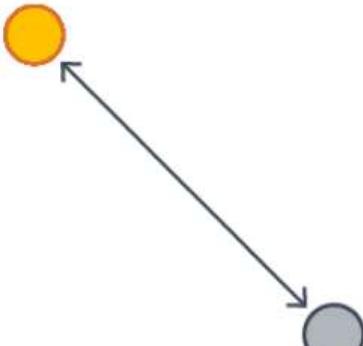




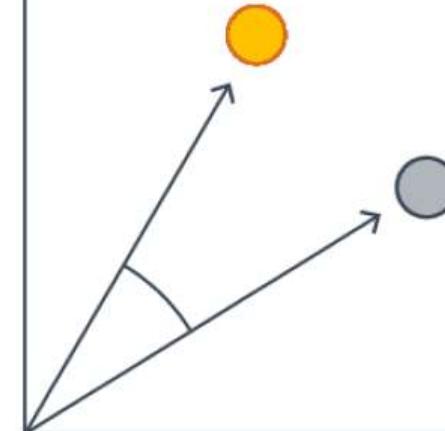


# Выбор метрики расстояния

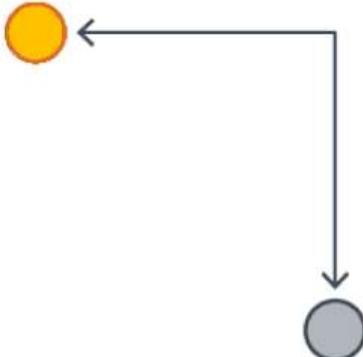
Euclidean



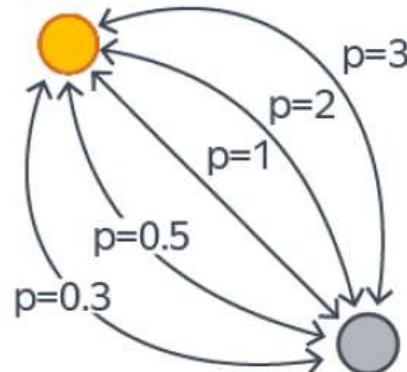
Cosine



Manhattan



Minkowski



- Евклидово расстояние:

$$\rho(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$$

- Манхэттенская метрика:

$$\rho(x, y) = \sum_i |x_i - y_i|$$

- Метрика Минковского:

$$\rho(x, y) = \left( \sum_i |x_i - y_i|^p \right)^{1/p}$$

- Косинусное расстояние:

$$\rho(x, y) = 1 - \cos \theta = 1 - \frac{x \times y}{|x||y|}$$

# Взвешенный KNN (weighted KNN)

Метрический алгоритм классификации:

$$a(x; X^l) = \operatorname{argmax} \sum_{i=1}^l [y^{(i)} = y] w(i, x)$$



Оценка близости объекта  $x$  к классу  $y$

$w(i, x)$  – вес, степень близости к объекту  $x$  его  $i$ -го соседа

# Итоги лекции

- Основные понятия машинного обучения: объект (sample), ответ (target), обучающая выборка (dataset), признак (feature), алгоритм, модель алгоритмов, метод обучения, эмпирический риск, функция потерь (loss), переобучение (overfitting)
- Виды машинного обучения и типы решаемых задач
- Этапы решения задач машинного обучения
  - Понимание задачи и данных
  - Предобработка данных и feature engineering
  - Построение модели
  - Сведение к задаче оптимизации
  - Решение проблемы переобучения
  - Оценивание качества
  - Внедрение модели
- Рассмотрены некоторые прикладные задачи машинного обучения