Тверской государственный технический университет

Интеллектуальные информационные системы
Вводная лекция

Шпигарь Андрей Николаевич

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

2024 г.

Чего не будет в курсе?

× Сложной математики

× Написания кода

× Exploratory data analysis

× Домашних заданий

Идея: https://youtube.com/playlist?list=PLpIXia0UeJFV3MIGparopqRK5sGvaeZmc&si=KCRZG2mLXNJD2N5U
Лекции Виктора Немченко «Нейронные сети и их применение в научных исследованиях»

Что будет в курсе?

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

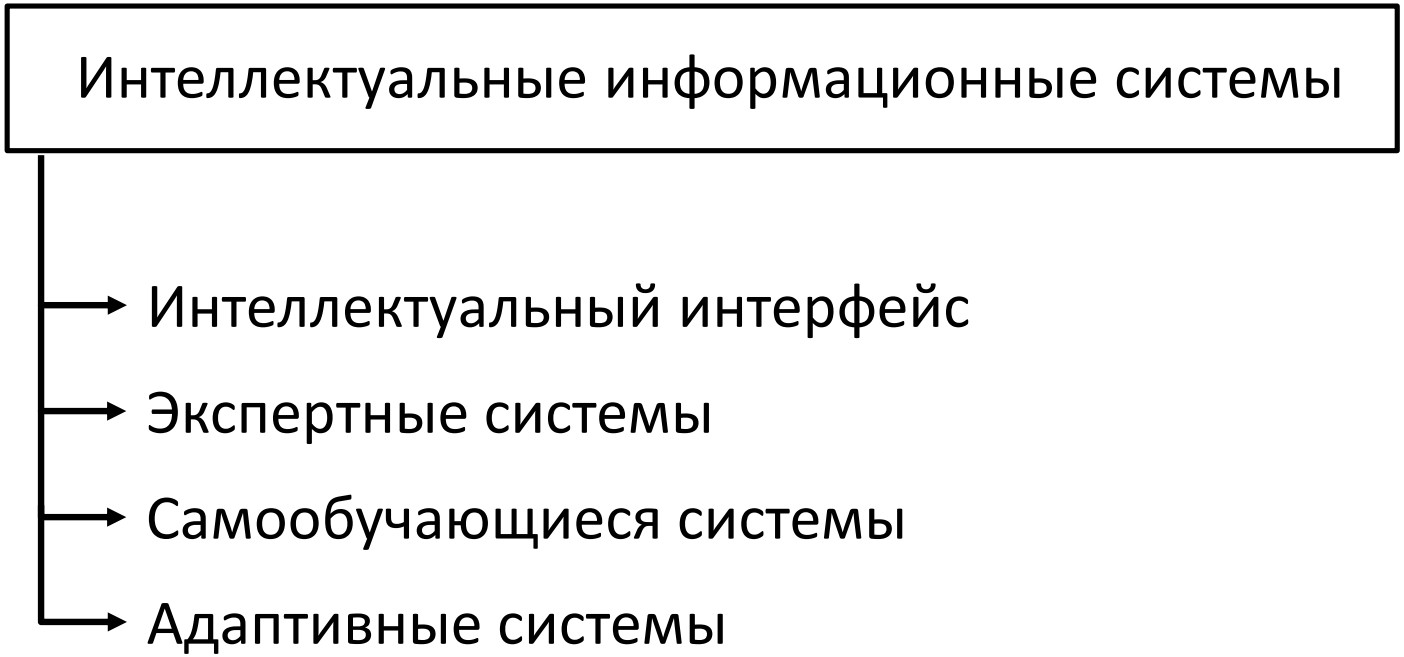
Содержание курса

- L00 – Math refresher
- L01 – Введение в интеллектуальные информационные системы
- 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



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

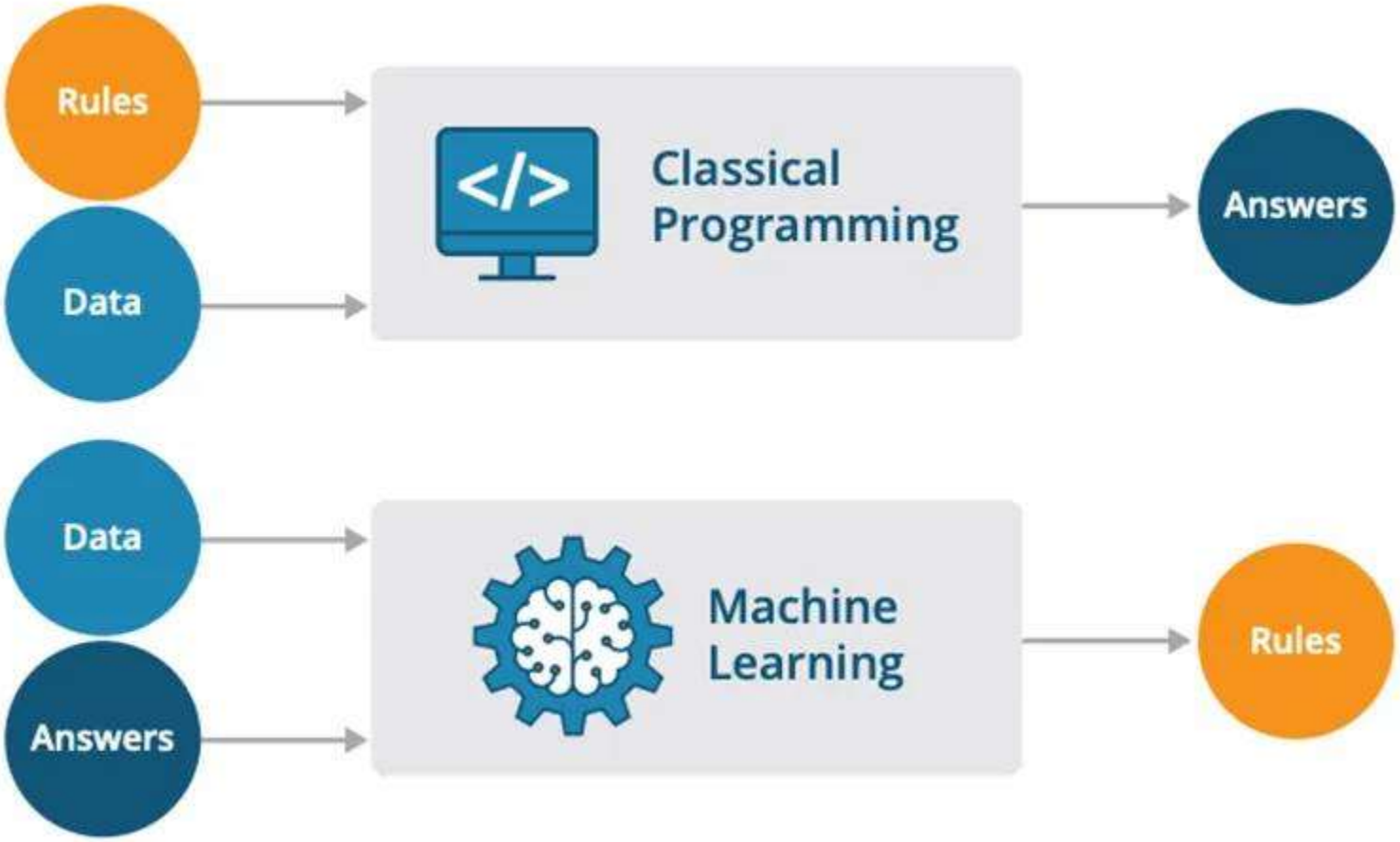
Экспертные системы – решают задачи на основе накапливаемой базы знаний, отражающей опыт работы экспертов в некоторой проблемной области. Включают в себя:

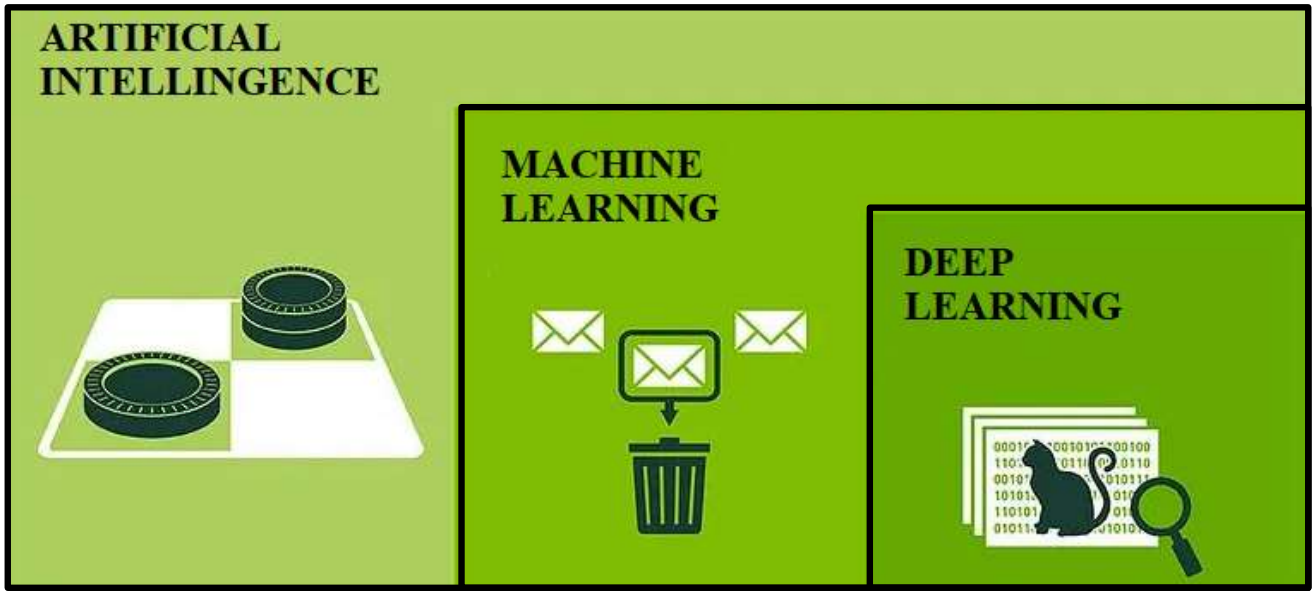
1. Классифицирующие системы
2. Доопределяющие системы
3. Трансформирующие системы
4. Многоагентные системы

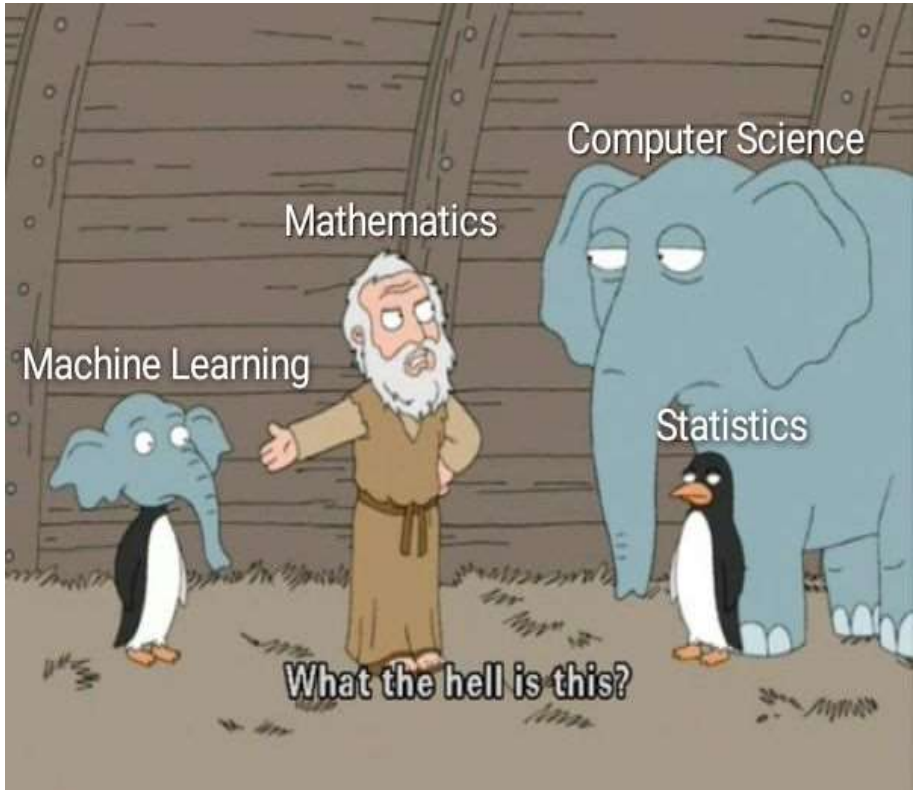
Самообучающиеся системы – основаны на методах автоматической классификации примеров ситуаций реальной практики (обучение на примерах)

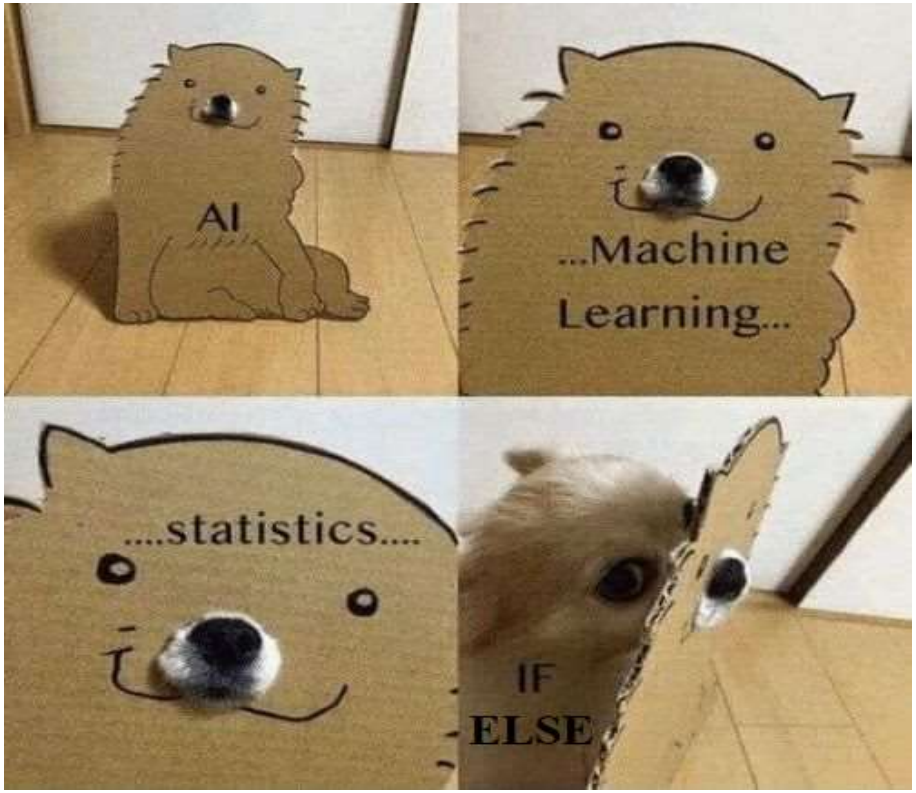
Адаптивные системы – основаны на постоянно развиваемой модели проблемной области, поддерживаемой в базе знаний, на основе которой осуществляется генерация или конфигурация программного обеспечения

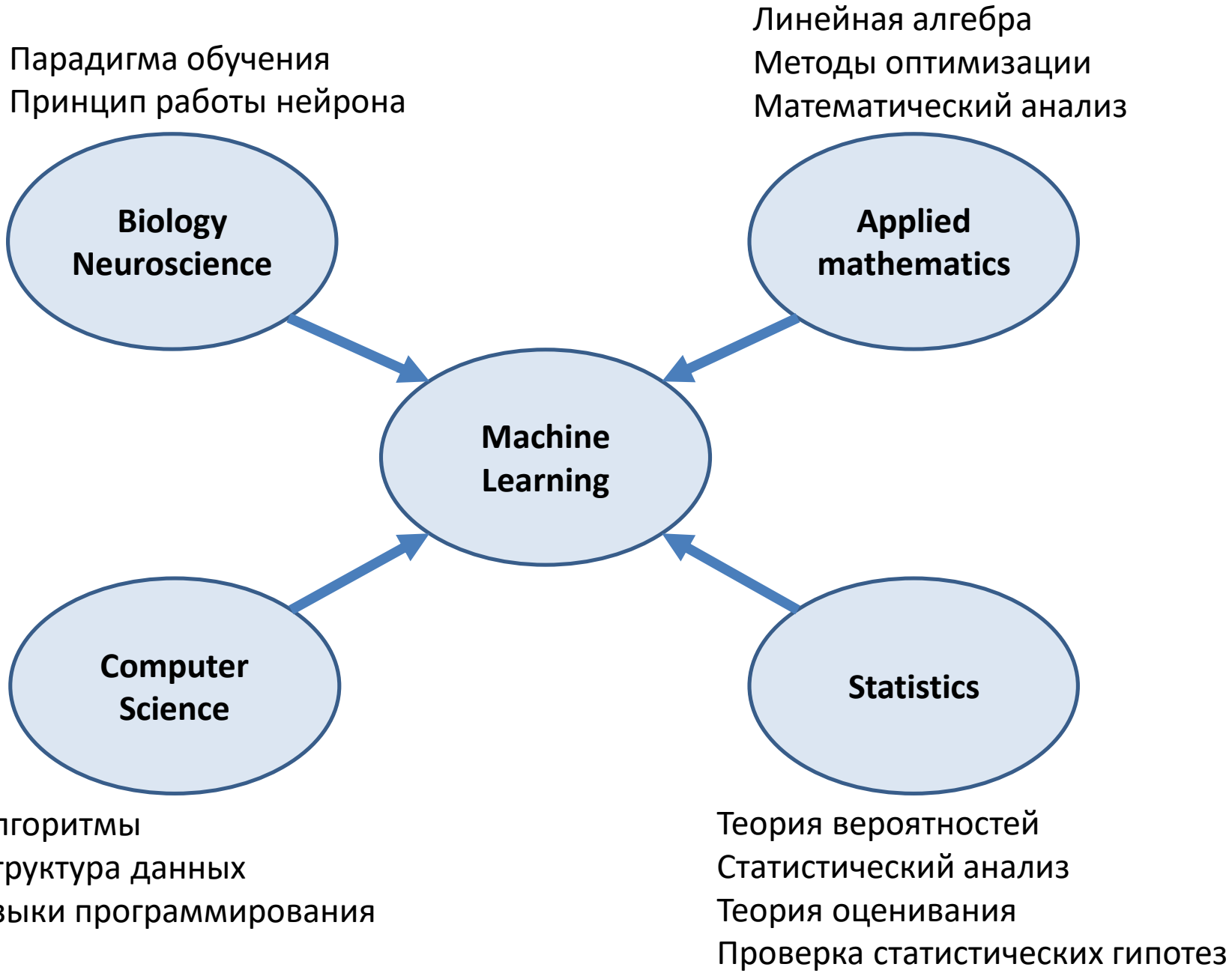
Концепция машинного обучения



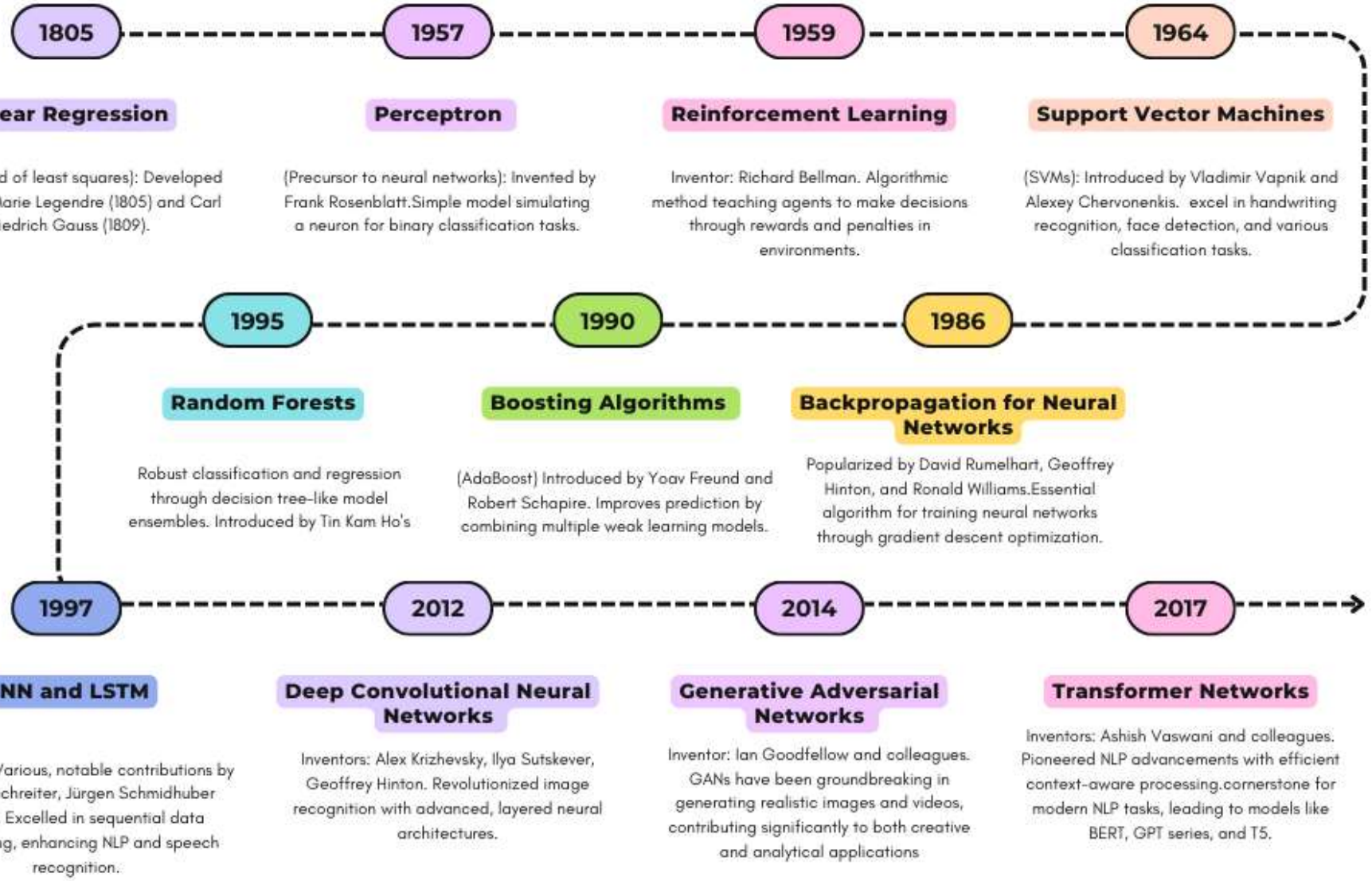

ARTIFICIAL INTELLINGENCE
MACHINE LEARNING
DEEP LEARNING


Computer Science
Mathematics
Machine Learning
Statistics
What the hell is this?


AI
...Machine Learning...
....statistics....
IF
ELSE


Парадигма обучения
Принцип работы нейрона
Линейная алгебра
Методы оптимизации
Математический анализ
Biology
Neuroscience
Applied
mathematics
Machine
Learning
Computer
Science
Statistics
Алгоритмы
Структура данных
Языки программирования
Теория вероятностей
Статистический анализ
Теория оценивания
Проверка статистических гипотез

Развитие машинного обучения



Источник: https://www.marketcalls.in/machine-learning/infographic-evolution-of-machine-learning.html

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)***

$\boldsymbol{X}$ – множество *объектов*, описанные в пространстве *признаков* (f – признак объекта, feature)

$\boldsymbol{Y}$ – множество ответов (оценок, предсказаний, прогнозов)

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

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

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

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

Задача обучения с учителем: по выборке $\boldsymbol{X^l}$ восстановить зависимость y^*, то есть построить решающую функцию $\boldsymbol{a: X \to Y}$, которая бы приближала целевую функцию $\boldsymbol{y^*(x)}$, 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 $\boldsymbol{f}: \boldsymbol{X} \rightarrow \boldsymbol{D_f}$, где $\boldsymbol{D_f}$ - множество допустимых значений признака

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

Типы признаков:

- Если $\boldsymbol{D_f} = \{\mathbf{0}, \mathbf{1}\}$, то *f – бинарный* признак
- Если $\boldsymbol{D_f}$ - конечное множество, то *f – категориальный* признак
- Если $\boldsymbol{D_f}$ - конечное упорядоченное множество, то *f – порядковый* признак
- Если $\boldsymbol{D_f} = \mathbb{R}$, 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 M непересекающихся классов (multi-class classification)
- $Y = \{0,1\}^M$ - на M 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 обучающей выборки (датасета) – задача Titanic на Kaggle

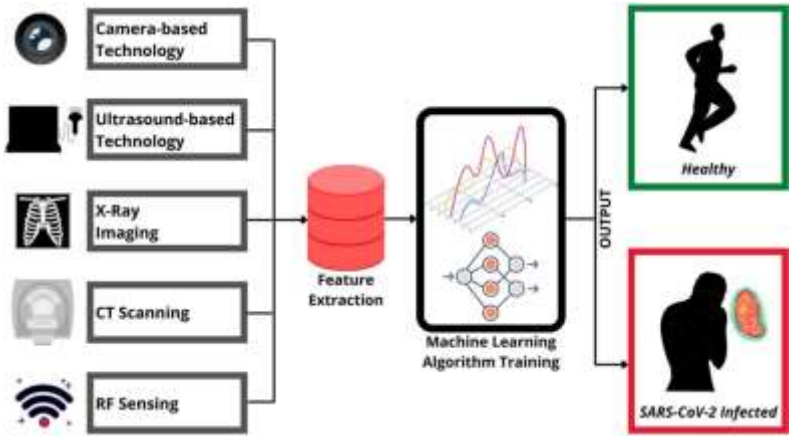
PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Survived
1	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25	nan	S	0
2	1	Cumings, Mrs. John Bradley…	female	38	1	0	PC 17599	71.2833	C85	C	1
3	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. …	7.925	nan	S	1
4	1	Futrelle, Mrs. Jacques Hea…	female	35	1	0	113803	53.1	C123	S	1
5	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05	nan	S	0
6	3	Moran, Mr. James	male	nan	0	0	330877	8.4583	nan	Q	0
7	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S	0
8	3	Palsson, Master. Gosta Leo…	male	2	3	1	349909	21.075	nan	S	0
9	3	Johnson, Mrs. Oscar W (Eli…	female	27	0	2	347742	11.1333	nan	S	1
10	2	Nasser, Mrs. Nicholas (Ade…	female	14	1	0	237736	30.0708	nan	C	1
11	3	Sandstrom, Miss. Marguerit…	female	4	1	1	PP 9549	16.7	G6	S	1
12	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.55	C103	S	1
13	3	Saundercock, Mr. William H…	male	20	0	0	A/5. 2151	8.05	nan	S	0
14	3	Andersson, Mr. Anders Johan	male	39	1	5	347082	31.275	nan	S	0
15	3	Vestrom, Miss. Hulda Amand…	female	14	0	0	350406	7.8542	nan	S	0
16	2	Hewlett, Mrs. (Mary D King…	female	55	0	0	248706	16	nan	S	1

Источник:https://www.kaggle.com/competitions/titanic

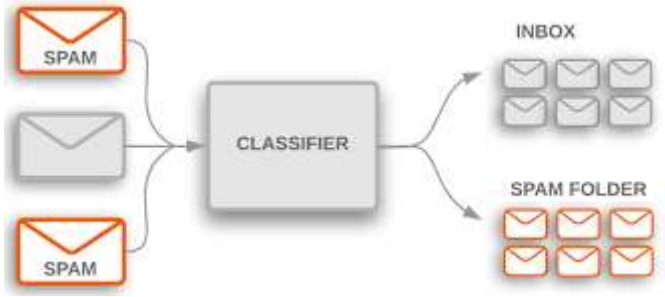
Несколько примеров прикладных задач

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

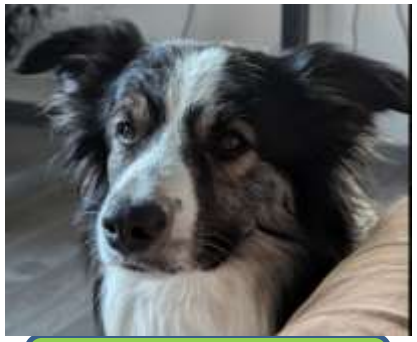
Tiger

Dog

Car

Tiger

Tiger



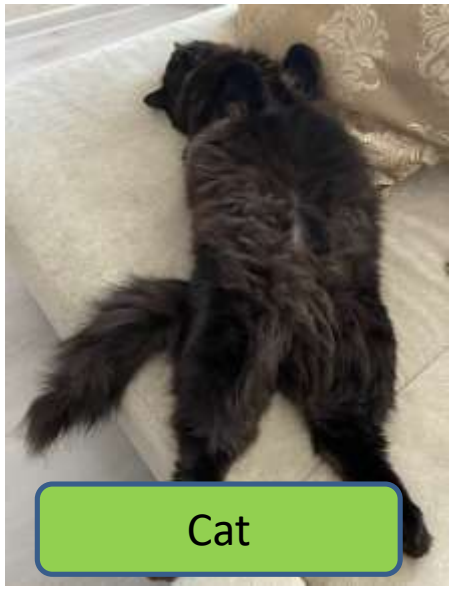
Dog

Несколько примеров прикладных задач

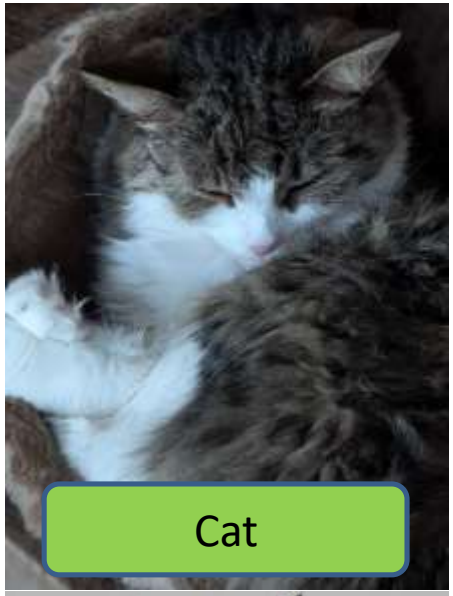
Классификация:

- Классификация изображений

Person



Cat

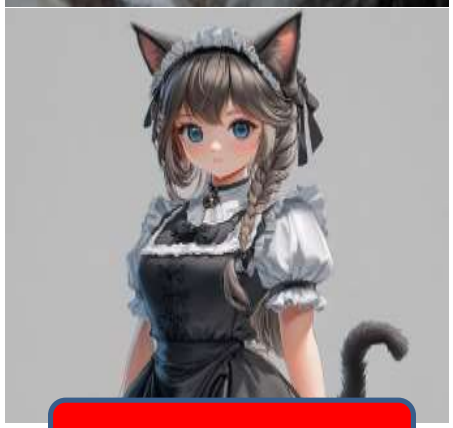


Cat

Cat

Person

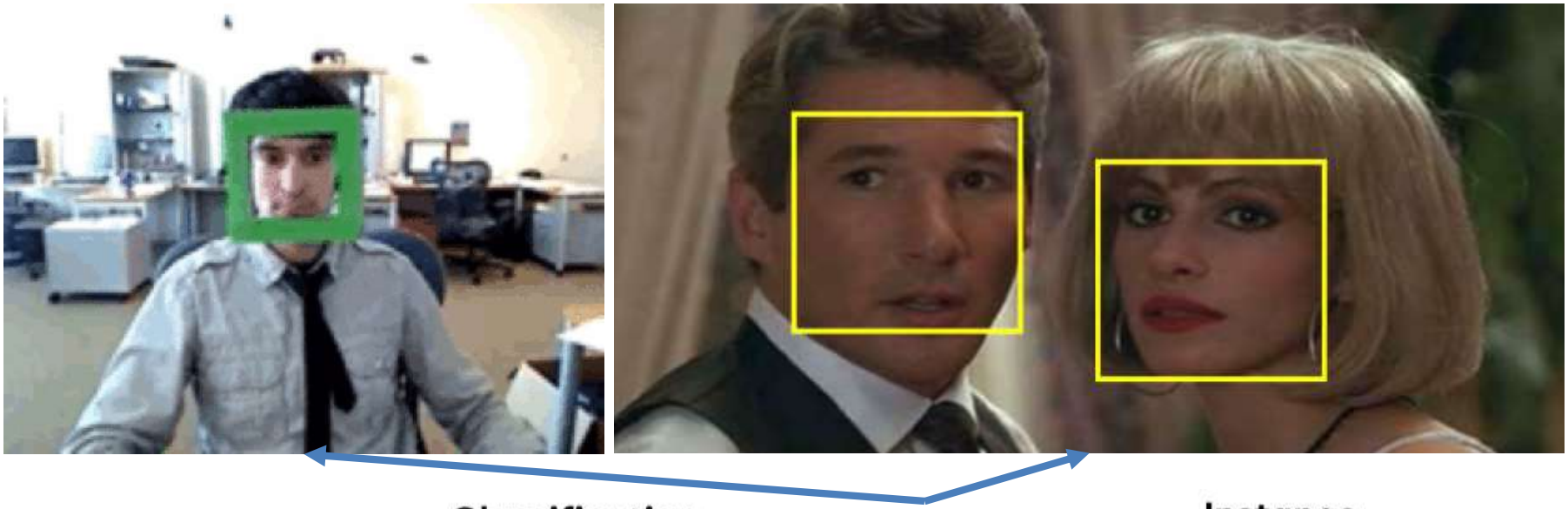
Person



Cat

Person

Несколько примеров прикладных задач



Источник:https://www.superannotate.com/blog/image-segmentation-for-machine-learning

Несколько примеров прикладных задач

Регрессия:

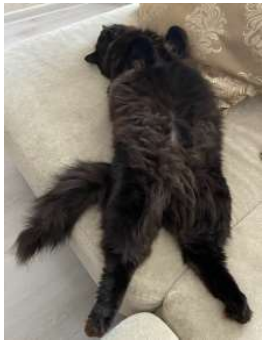
- Экономика и финансы – прогнозирование потребительского спроса. Необходимо оценить объемы продаж для каждого товара на заданный интервал времени. На основе этих прогнозов осуществляется планирование закупок и формирование ценовой политики
- Рекомендательные системы – задача предсказания рейтинга, товара 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

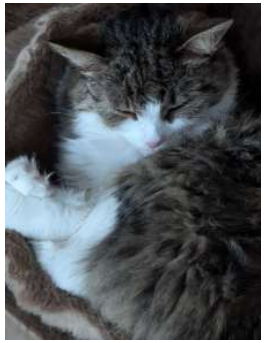
Задача ранжирования – ранжирование документов при поиске по запросу пользователя

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

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

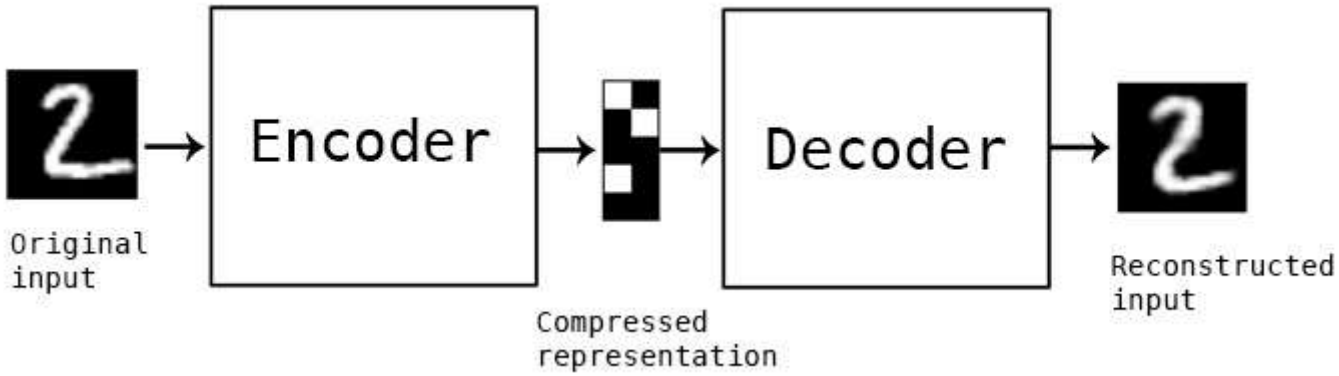
1. *Кластеризация* – задача разделения выборки на подмножества (кластеры) так, чтобы каждый кластер состоял из похожих объектов, а объекты разных кластеров существенно отличались





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)



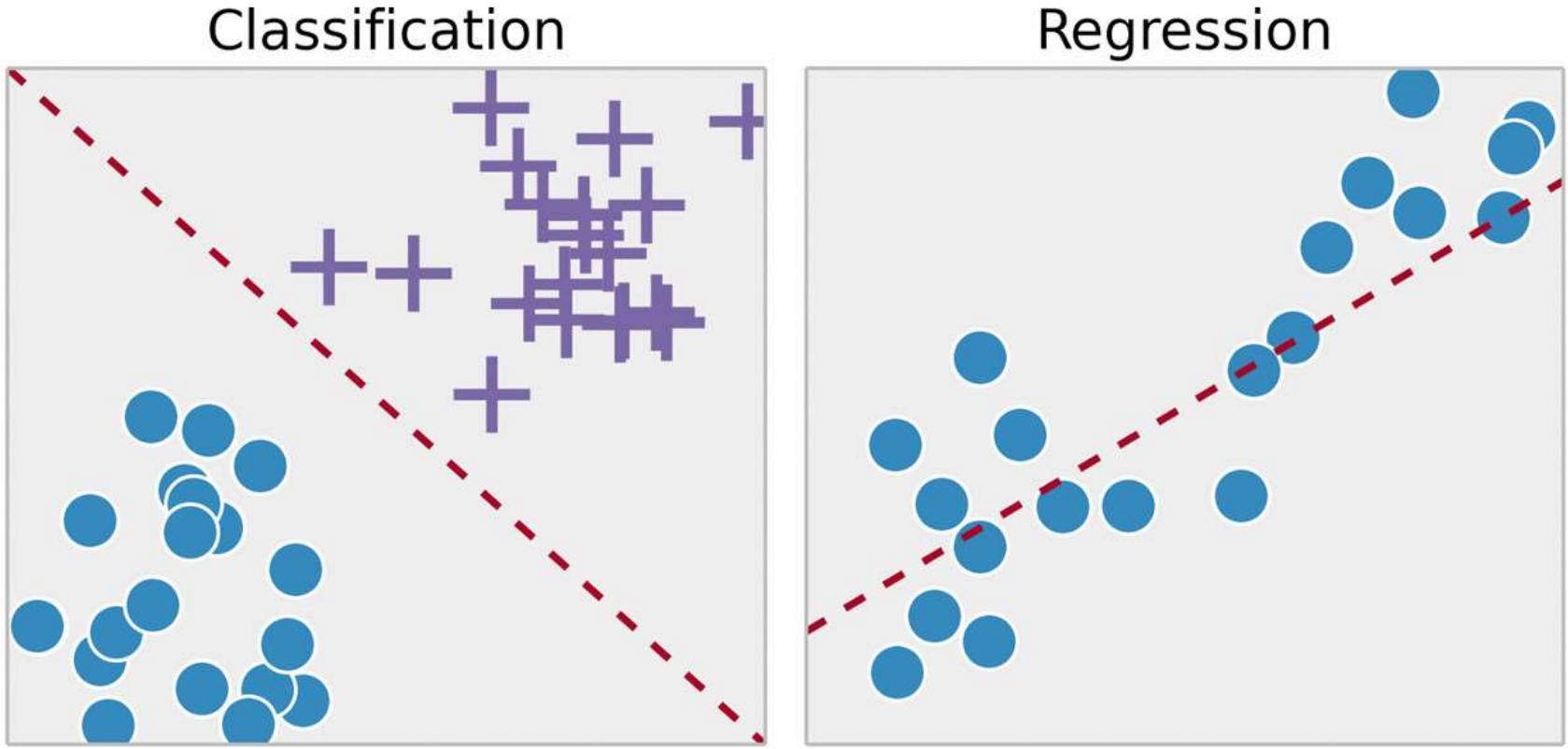
Источник:https://bytepawn.com/building-a-pytorch-autoencoder-for-mnist-digits.html

- ***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*



Модель алгоритмов и метод обучения

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

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

Два основных этапа машинного обучения

- Этап *обучения (train)* – на этапе обучения метод $\boldsymbol{\mu}$ по выборке $\boldsymbol{X^l}$ строит алгоритм $\boldsymbol{\alpha = \mu(X^l)}$. 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 $\boldsymbol{\alpha}$ для новых объектов $\boldsymbol{x}$ выдает ответы **y=** $\boldsymbol{\alpha(x)}$

Функционал качества

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

Функционал качества алгоритма α на выборке X^l:

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

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:

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

- Функции потерь для задач регрессии:

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

$\boldsymbol{L(a, x) = (a(x) - y(x))^2}$ - 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 $\boldsymbol{X} \times \boldsymbol{Y}$ с плотностью $\boldsymbol{p}(\boldsymbol{x}, \boldsymbol{y})$ из которого случайно и независимо выбираются $\boldsymbol{l}$ наблюдений

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

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

Функция правдоподобия (likelihood):

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

Минимизация *логарифма* функции правдоподобия:

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

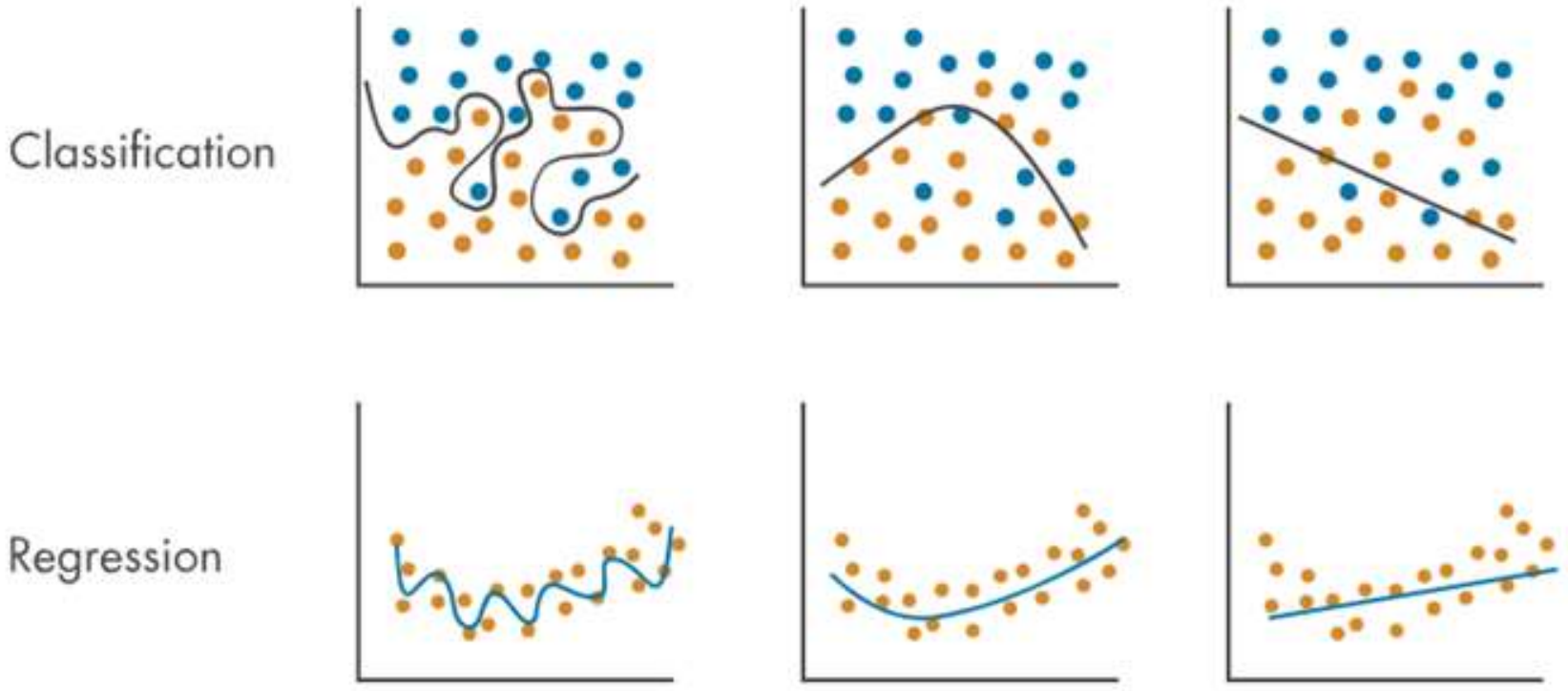
Overfitting vs Underfitting

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

Предположение: выборки $\boldsymbol{X^l}$ и $\boldsymbol{X^k}$ - 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 $\boldsymbol{X^k}$ существенно хуже, чем на обучающей выборке $\boldsymbol{X^l}$

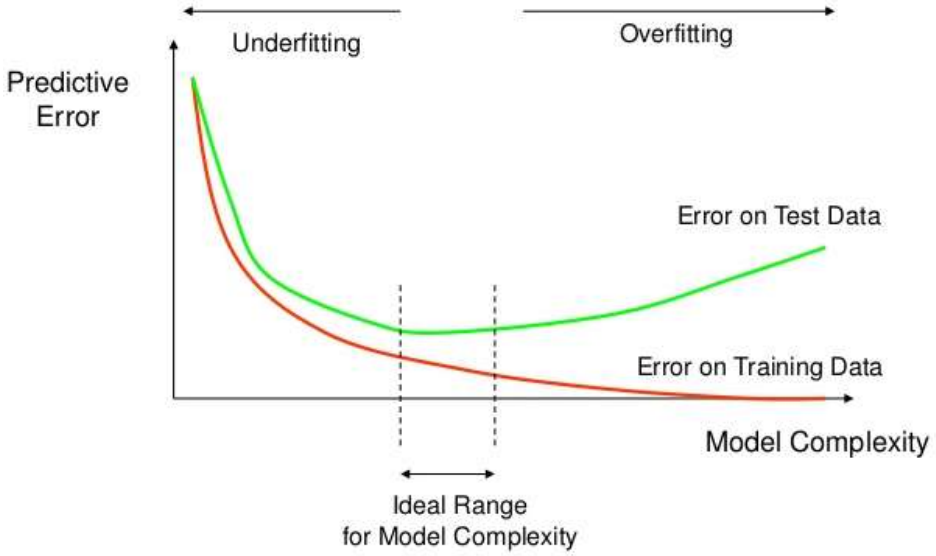


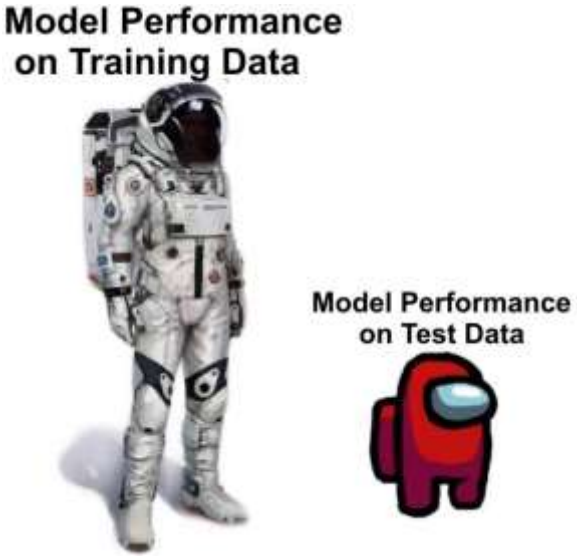
Из-за чего возникает переобучение?

- 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 θ - *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

$$\boldsymbol{HO}\left(\boldsymbol{\mu}, \boldsymbol{X}^{\boldsymbol{l}}, \boldsymbol{X}^{\boldsymbol{k}}\right) = \boldsymbol{Q}\left(\boldsymbol{\mu}\left(\boldsymbol{X}^{\boldsymbol{l}}\right), \boldsymbol{X}^{\boldsymbol{K}}\right) \rightarrow \boldsymbol{min}$$

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

$$\boldsymbol{LOO}\left(\boldsymbol{\mu}, \boldsymbol{X}^{\boldsymbol{L}}\right) = \frac{\boldsymbol{1}}{\boldsymbol{L}} \sum_{\boldsymbol{i}=\boldsymbol{1}}^{\boldsymbol{L}} \boldsymbol{L}\left(\boldsymbol{\mu}\left(\boldsymbol{X}^{\boldsymbol{L}} \backslash \{\boldsymbol{x}_{\boldsymbol{i}}\}\right), \boldsymbol{x}_{\boldsymbol{i}}\right) \rightarrow \boldsymbol{min}$$

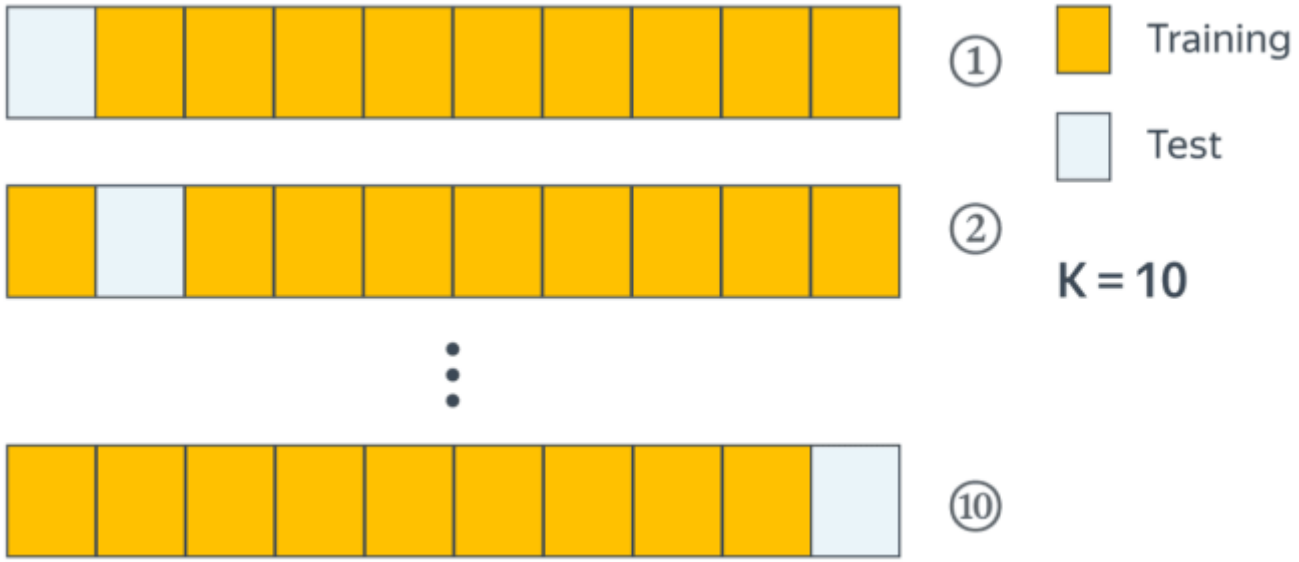
Эмпирические оценки обобщающей способности

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

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

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



Источник:https://neptune.ai/blog/cross-validation-in-machine-learning-how-to-do-it-right

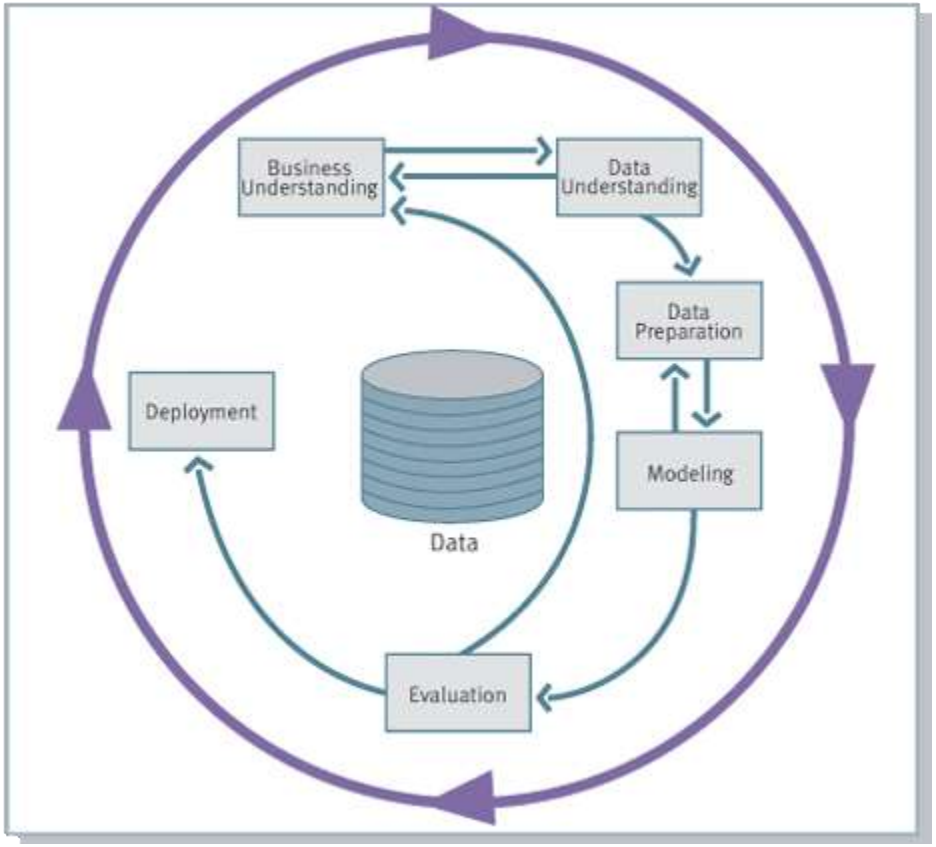
Особенности данных и постановок прикладных задач

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

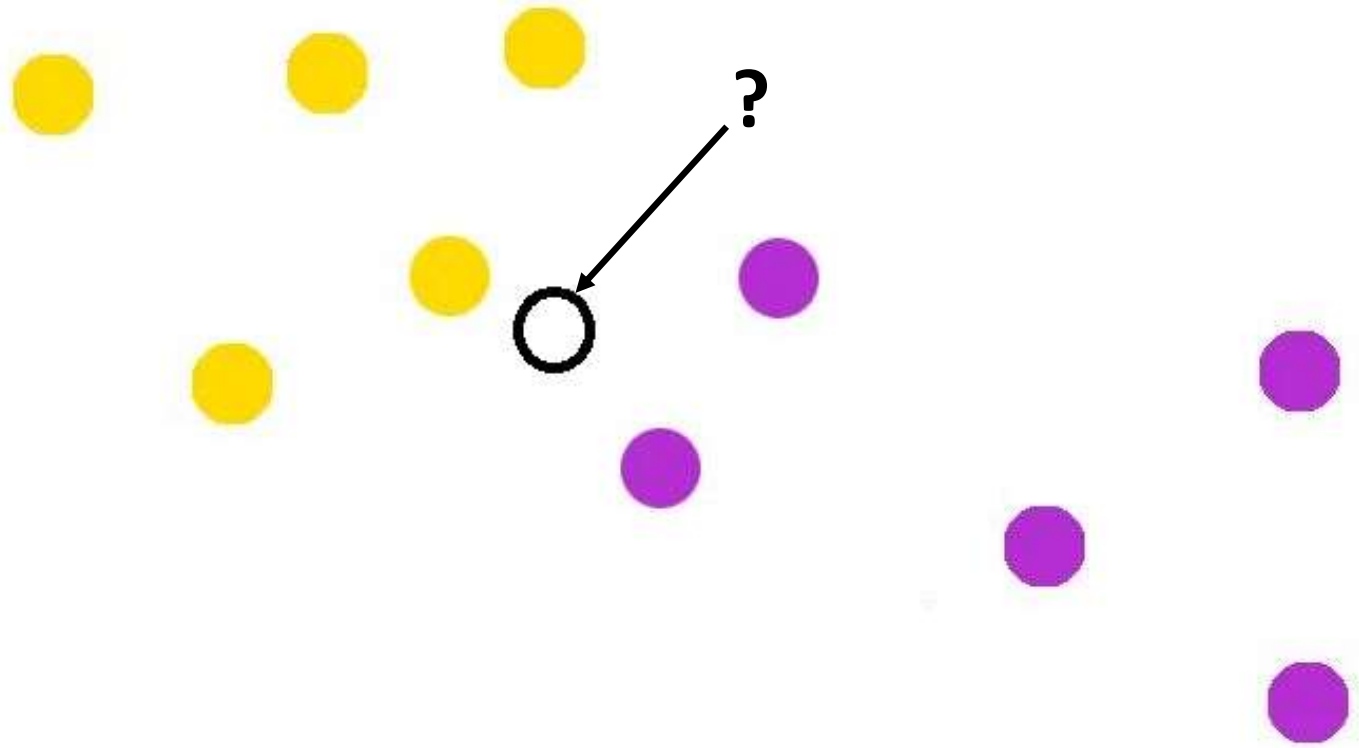
CRISP-DM: CRoss Industry Standard Process for Data Mining (1999)

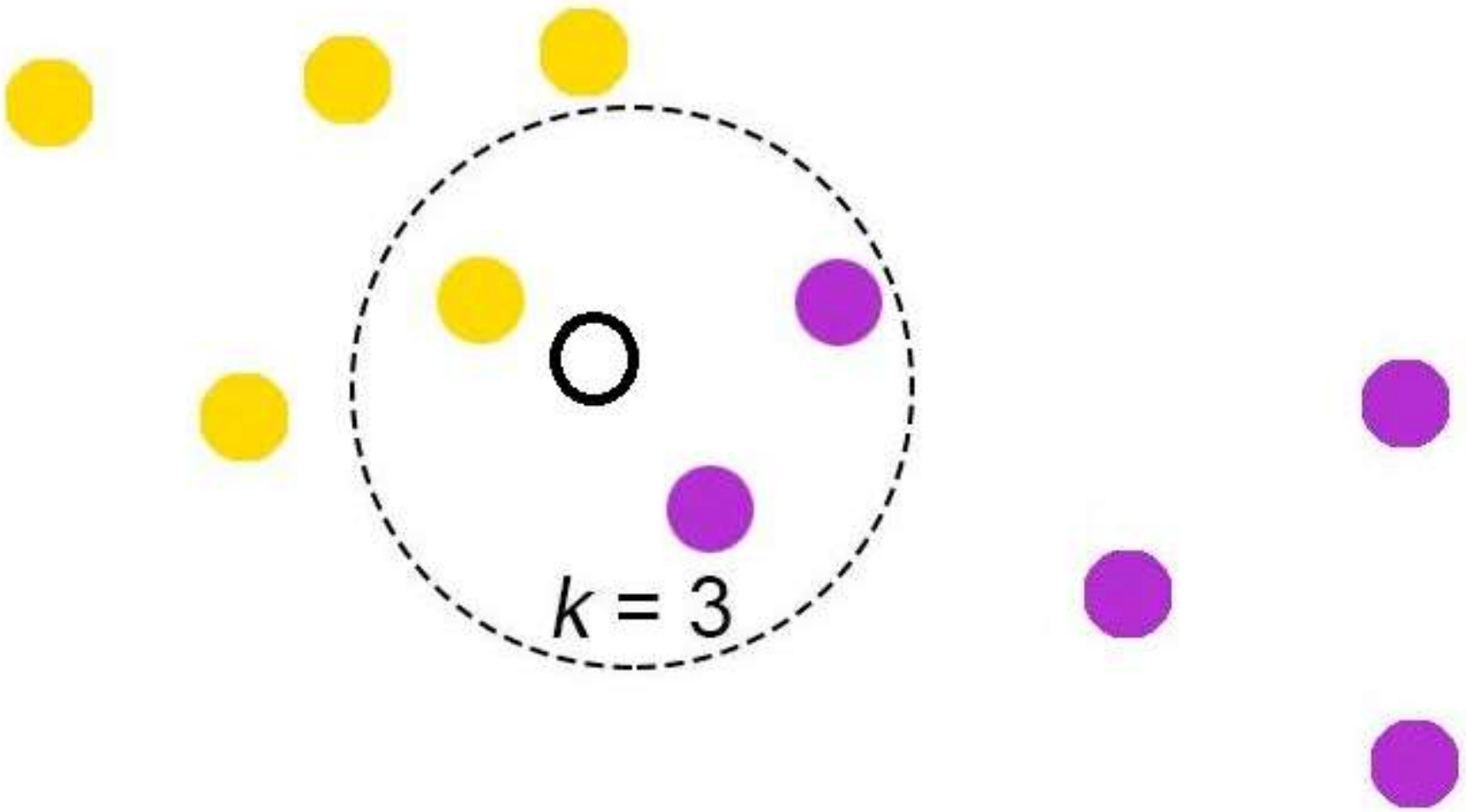


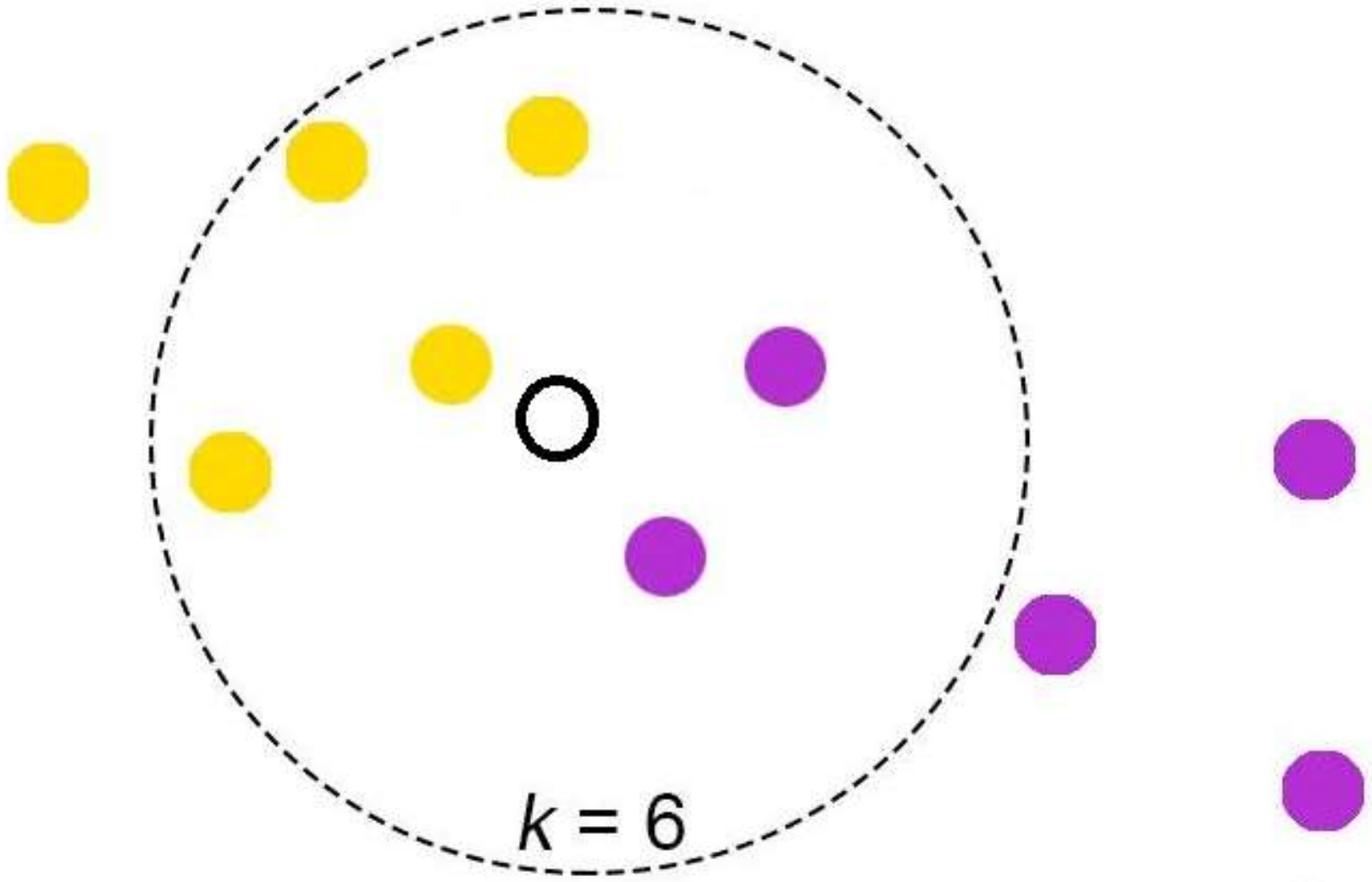
- Понимание бизнеса
- Понимание данных
- Предобработка данных и инженерия признаков
- Разработка моделей и настройка параметров
- Оценивание качества
- Внедрение

Метрические методы
Метод k-ближайших соседей (KNN)

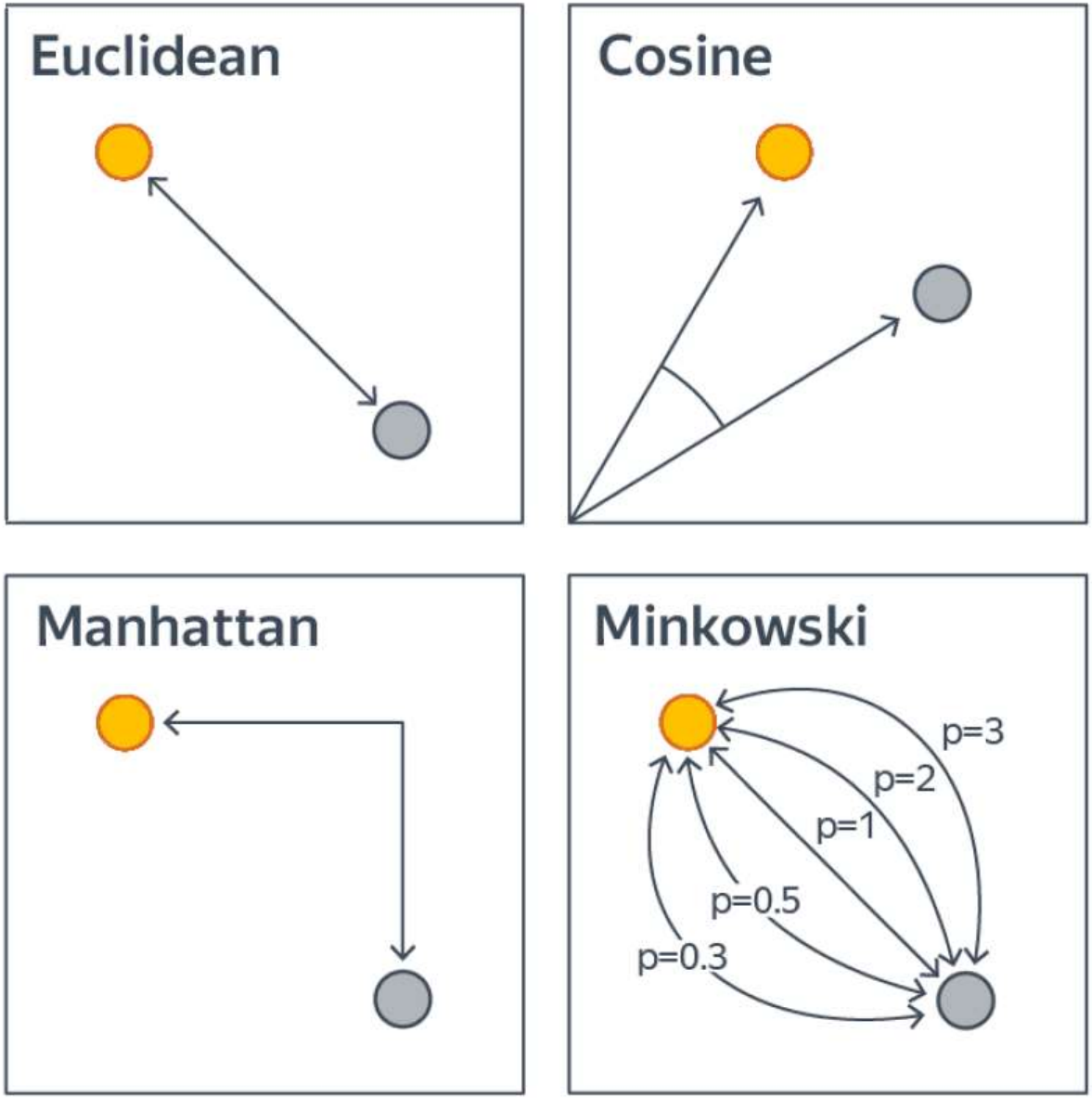
Задача – какого цвета объект?




O
k = 3


k = 6

Выбор метрики расстояния



- Евклидово расстояние:

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

- Манхэттенская метрика:

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

- Метрика Минковского:

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

- Косинусное расстояние:

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

Взвешенный KNN (weighted KNN)

Метрический алгоритм классификации:

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

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