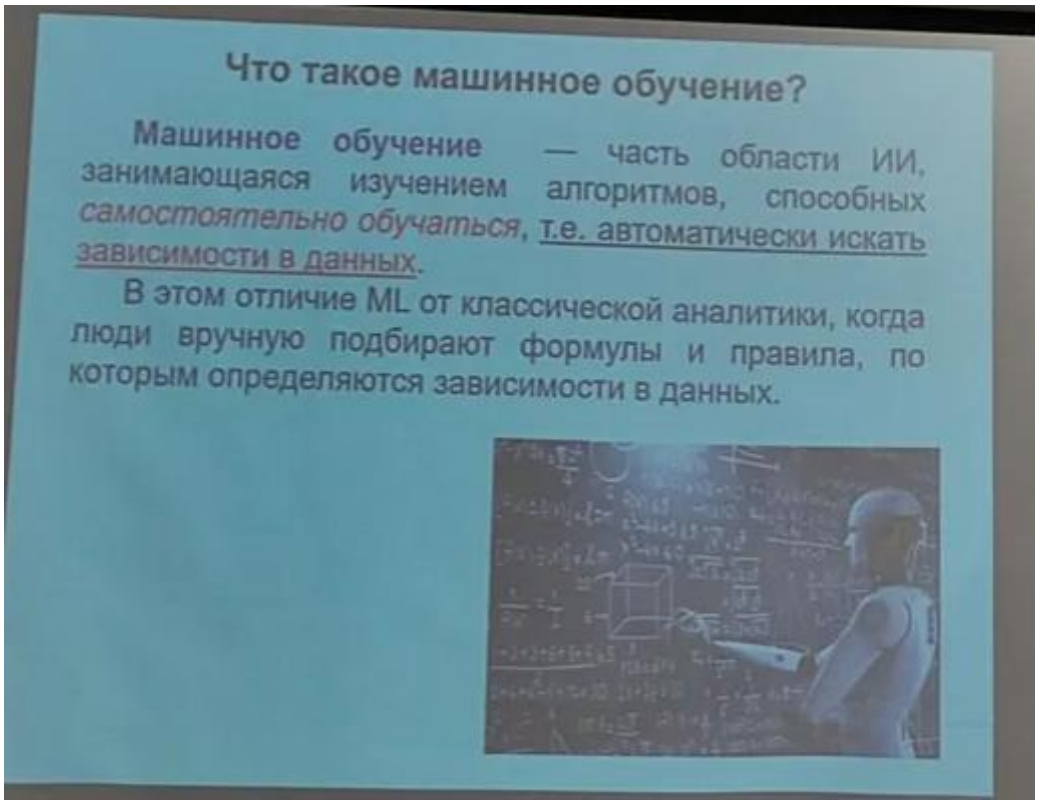
1. Искусственный интеллект . Машинное обучение (ML). Виды ML. Типы задач в ML. Примеры задач.

Искусственный интеллект - комплекс технологических решений, позволяющий имитировать когнитивные функции человека (включая самообучение и поиск решений без заранее заданного алгоритма) и получать при выполнении конкретных задач результаты, сопоставимые, как минимум, с результатами интеллектуальной деятельности человека. Комплекс технологических решений включает в себя информационно-коммуникационную инфраструктуру, программное обеспечение (в том числе в котором используются методы машинного обучения), процессы и сервисы по обработке данных и поиску решений;



Основные виды машинного обучения

ML можно условно разделить на три категории:

Контролируемое обучение (Supervised learning)

Программа ML получает как входные данные, так и соответствующую маркировку. Это означает, что данные обучения должны быть предварительно помечены человеком.

Неконтролируемое обучение (Unsupervised learning)

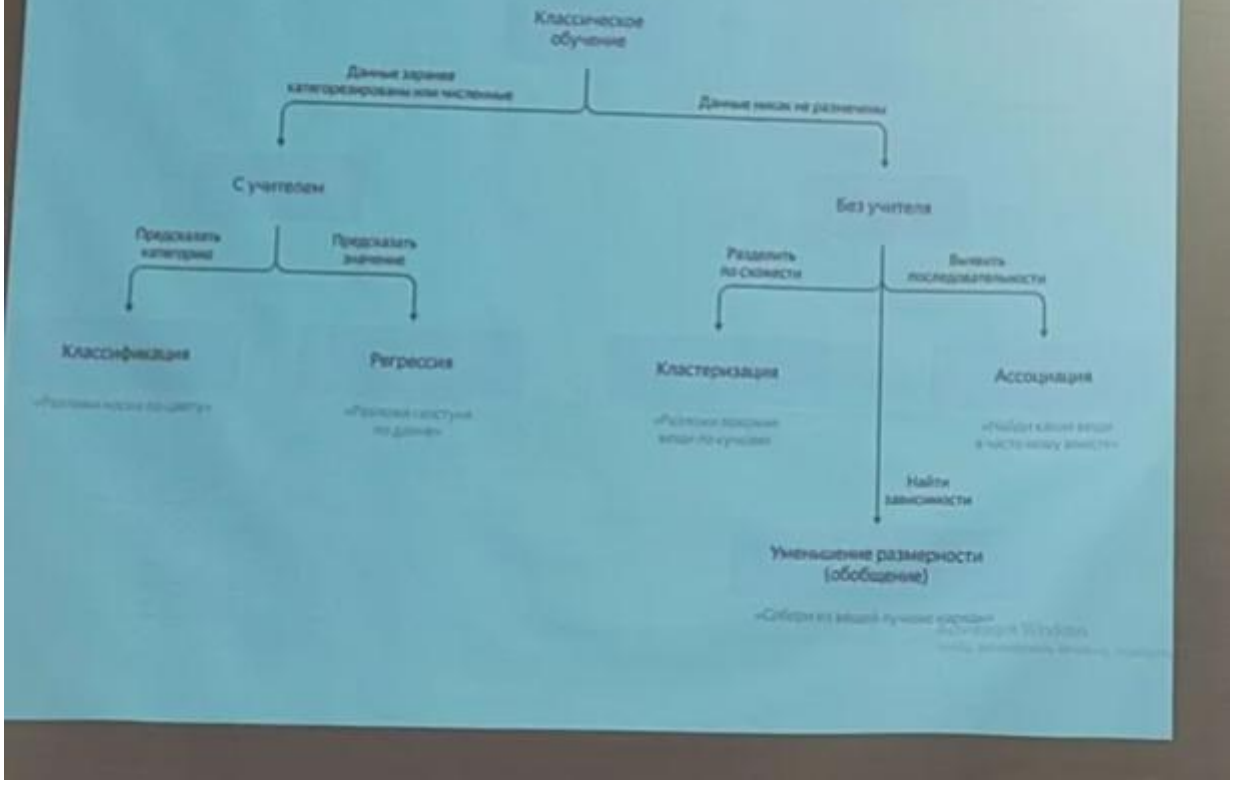
Алгоритму обучения не присваиваются метки. Алгоритм должен определить кластеризацию входных данных.

Обучение с подкреплением (Reinforcement learning)

Компьютерная программа динамически взаимодействует со своим окружением. Это означает, что программа получает положительные и/или отрицательные отзывы для улучшения ее производительности.

Обучение с подкреплением больше относится к методам глубинного обучения (Deep Learning).

Типы задач в машинном обучении



Типы задач в машинном обучении

Самые популярные типы задач в ML — это **классификация** и **регрессия**.

В задачах **классификации** *целевая переменная* — класс объекта, т.е. в задачах классификации ответ может быть одним из конечного числа классов.

Например, в задаче *бинарной (или двухклассовой) классификации* можно предсказывать:

- пол клиента (мужчина или женщина)
- уйдет клиент из компании или нет
- вернет человек кредит или нет
- болен пациент или здоров

и т. д.

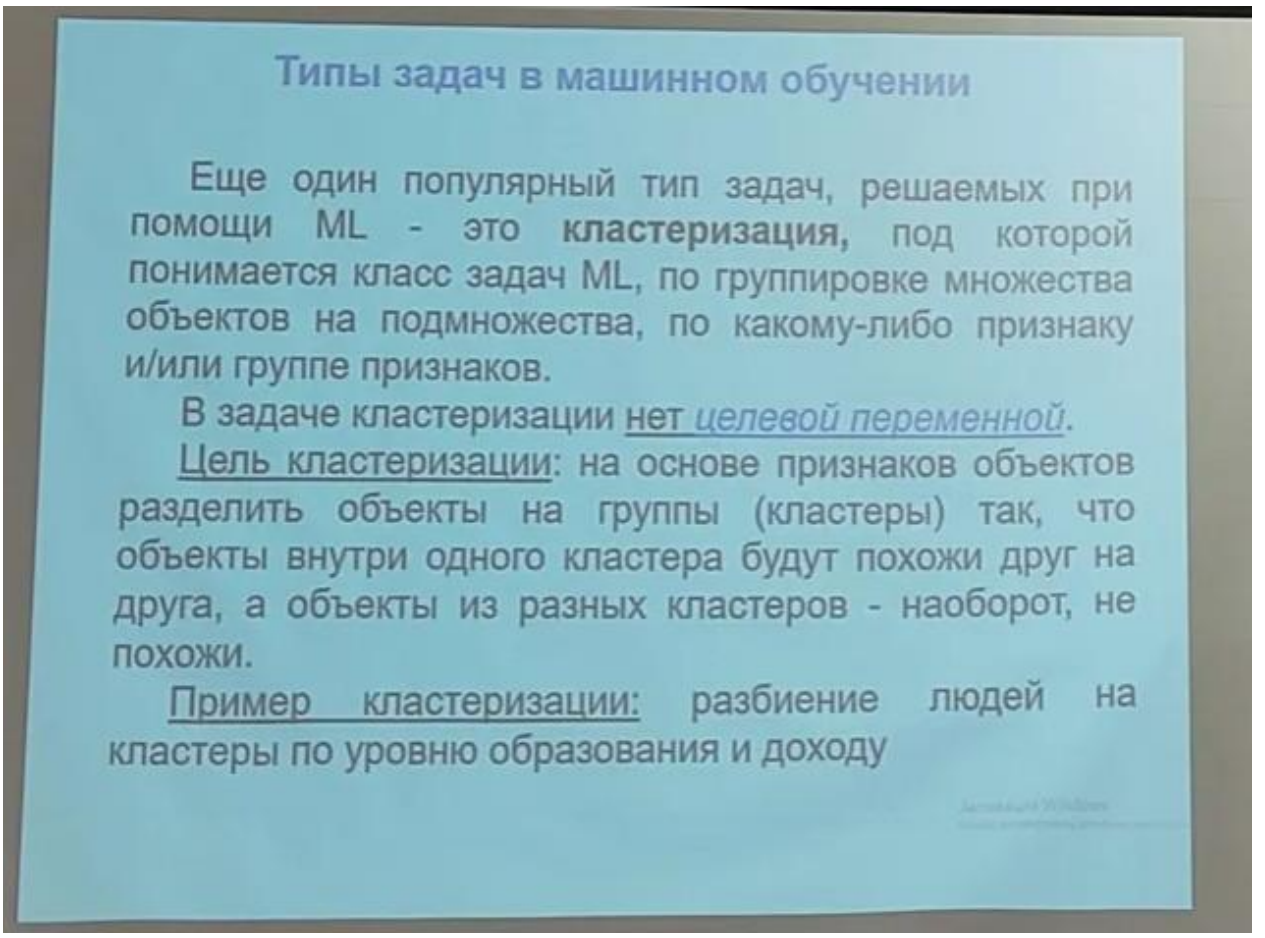
Типы задач в машинном обучении

В задачах **регрессии** *целевая переменная* может принимать бесконечно много значений.

Например, прибыль фирмы может быть любым числом (как очень большим, так и очень маленьким) - даже отрицательным или нецелым

Примеры задач регрессии:

- Предсказание стоимости квартиры
- Предсказание прибыли фирмы (она может быть и отрицательной, и нецелой!)
- Предсказание курса валют



2. Основные понятия ML (данные, знания, признаки, набор данных, алгоритм ML, модель алгоритма ML). Жизненный цикл модели ML. Схема проекта по ML. Проблемы ML.

Данные и Знания

Данные – совокупность зафиксированных фактов

Информация – сведения, уменьшающие неопределённость

Знания – сведения, позволяющие действовать с прогнозируемым результатом

Типичная проблема:

Мы располагаем данными, они хранятся в цифровом виде, но мы не знаем, что в них.

Машинное обучение заключается в извлечении знаний из больших объемов данных.

Главные компоненты ML — данные, признаки и алгоритмы.

Данные. Для предсказания курса акций нужна история цен, а для определения интересов пользователя — лайки и репосты. Качество результата напрямую зависит от качества и количества данных: чем они разнообразнее, тем проще машине найти закономерности и тем точнее выходит результат.

Признаки (или фичи) — переменные, которые описывают отдельные характеристики объекта, важные для обучения.

Алгоритмы, или модели — методы решения задачи. От выбора метода зависит точность, скорость работы и размер готовой модели.

Алгоритмы ML требуют для своей работы данные!

В Python при использовании библиотеки Pandas основной прямоугольной структурой данных является объект DataFrame

Набор данных — совокупность данных, прошедших предварительную подготовку (обработку) в соответствии с требованиями законодательства РФ об информации, информационных технологиях и о защите информации и необходимых для разработки программного обеспечения на основе ИИ.

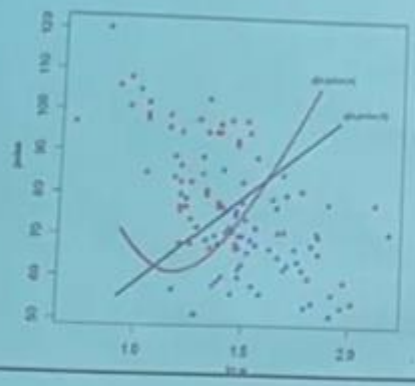
Национальная стратегия развития искусственного интеллекта до 2030 г. Утверждена Указом Президента РФ от 10 октября 2019 г. № 490. – URL: https://www.garant.ru/products/ipo/prime/doc/72738946/

Модель алгоритма

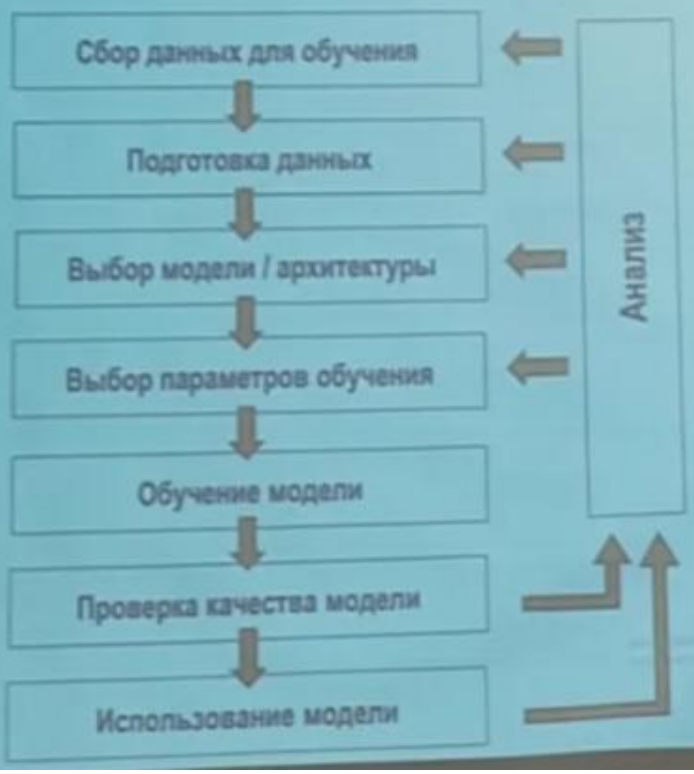
Решить задачу машинного обучения *означает разработать алгоритм или модель алгоритма, зависящего от параметров и позволяющих определить значение метки класса (Y) для* ***нового*** *объекта (**x**).*

Модель алгоритма

- **Моделью алгоритма** a называется параметрическое семейство функций $g : X \to Y$ или $g(x, \theta)$, где $\theta \in \Theta$ параметры в пространстве параметров.
- Процесс подбора оптимальной функции g и оптимальных параметров θ по обучающей выборке называют **настройкой** (fitting, tuning) или **обучением** (training) алгоритма a.



Жизненный цикл модели машинного обучения



В проблемах можно сказать переобучение, недообучение, нехватка данных, некачественные данные.

3. Разведочный анализ данных (EDA). Этапы EDA. Основы описательной статистики в EDA. Инструменты визуализации.

РАЗВЕДОЧНЫЙ АНАЛИЗ ДАННЫХ (Exploratory Data Analysis, EDA) – предварительное исследование датасета с целью определения его основных характеристик, взаимосвязей между признаками, а также сужения набора методов, используемых для создания модели ML.

РАЗВЕДОЧНЫЙ АНАЛИЗ ДАННЫХ

1. Нахождение пропущенного значения
2. Нахождение выбросов.
3. Понимание атрибутов с использованием описательной статистики.
4. Визуализация распределения атрибутов с использованием однофакторной и многофакторной визуализации данных.
5. Нахождение атрибутной корреляции и анализ того, какой атрибут важнее.
6. Изучение целевого признака (target).
7. Изучение признаков, по которым строится прогноз.
8. Изучение влияния признаков на таргет.

ОПИСАТЕЛЬНАЯ СТАТИСТИКА

Описательная статистика (Descriptive Statistics) – раздел статистической науки, в рамках которого изучаются методы систематизации, описания и представления основных свойств данных.

Такую статистику можно разделить на 2 категории: *меры центральной тенденции* (или *меры центра*) и *меры разброса*.

http://statistica.ru/theory/opisatelnye-statistiki/

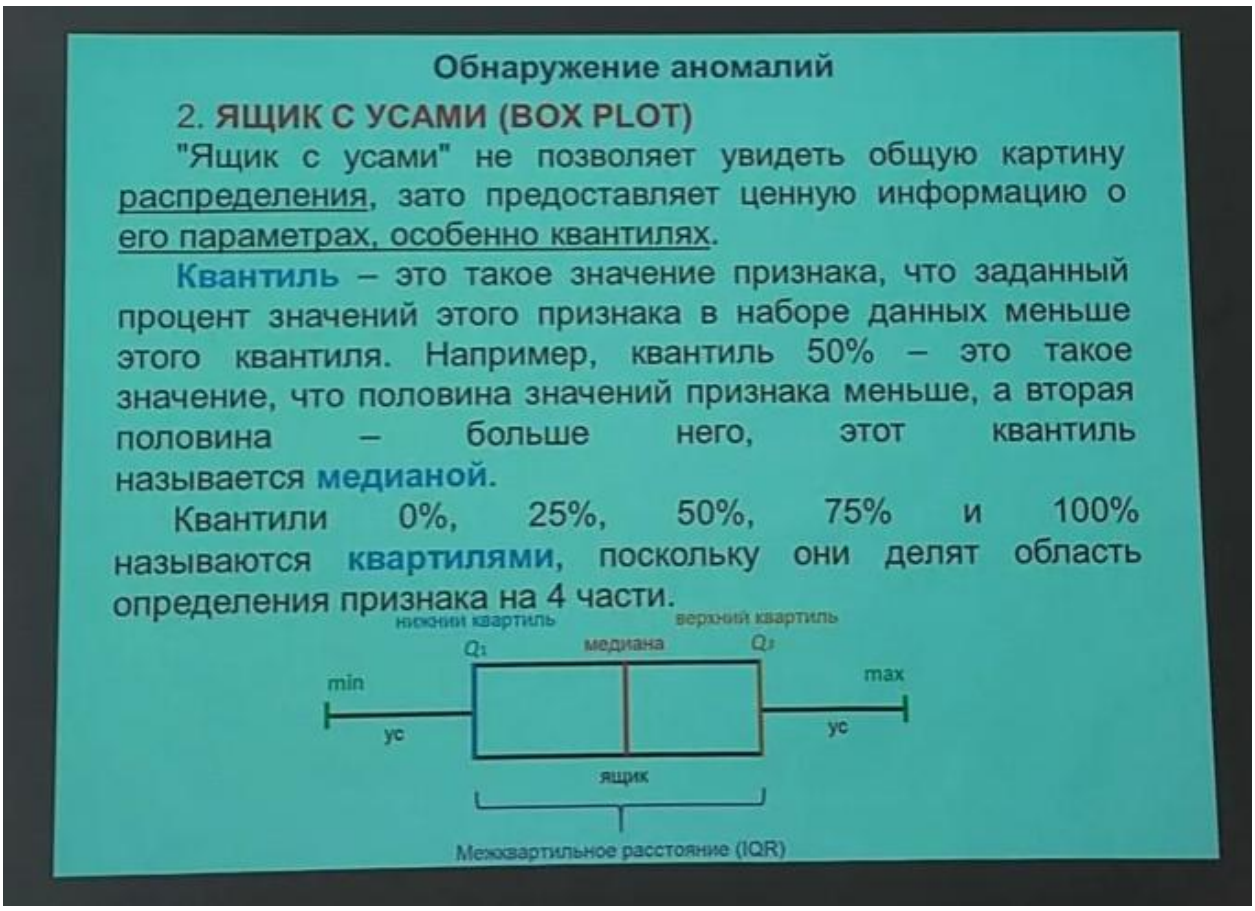
А. Визуализация данных

Визуализация данных позволяет понять паттерны, тренды и взаимосвязи в данных через графику и диаграммы.

1. ГИСТОГРАММЫ и ДИАГРАММЫ РАССЕЯНИЯ

Гистограмма – графическое представление распределения данных по различным интервалам. Она позволяет оценить, как часто значения попадают в определенные диапазоны и какие имеются пики или провалы в данных.

Диаграмма рассеяния – график, в котором каждая точка представляет собой отдельное наблюдение и показывает взаимосвязь между двумя переменными. Это может помочь нам определить, есть ли какая-либо зависимость или корреляция между ними.



4. Задача регрессии. Метод наименьших квадратов. Метрики качества модели регрессии (MSE, RMSE , MAE, R2).

target y continious

Прогноз непрерывного (continious) действительного числа с плавающей точкой (floating-point number)

Метод наименьших квадратов — метод нахождения оптимальных параметров линейной регрессии, таких, что сумма квадратов ошибок (регрессионных остатков) минимальна.

Метод заключается в минимизации евклидова расстояния между двумя векторами — вектором восстановленных значений зависимой переменной и вектором фактических значений зависимой переменной.

$$\hat{y} = w^T \mathbf{x} + b = \sum_{i=1}^{p} w_i x_i + b$$

$$\min_{w \in \mathbb{R}^p, b \in \mathbb{R}} \sum_{i=1}^{n} (w^T \mathbf{x}_i + b - y_i)^2$$

Метрика RMSE(среднеквадратичная ошибка — Mean Squared Error) вычисляется как квадратный корень из средней квадратичной ошибки

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

*Как и в случае **MAE (средней абсолютной ошибки)**, метрика **RMSE** не может указывать, насколько плоха модель, но может использоваться для сравнения двух моделей.*

Метрика MAE (средняя абсолютная ошибка — Mean Absolute Error) — результат деления суммы абсолютных значений ошибок прогноза на количество точек тестовой выборки:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - y_i'|$$

(n – объем тестовой выборки, y_i – истинные y_i' – предсказанные значения).

MAE=1/2(|110-100|+|50-70|)=15.

Test	Рост	Вес	Пол	IQ	IQ предсказанный
Маша	160	50	0	110	100
Петя	200	70	1	50	70

Общая метрика регрессии — коэффициент детерминации (Coefficient of Determination) R^2 || r^2. Обычно значение находится в **[0;1]**.

R^2 показывает количество объясненной дисперсии.

R^2 численно показывает, сколько процентов разброса данных объяснила модель. Чем ближе к 1, тем лучше (значит, вся дисперсия учтена моделью).

$$R^2 = 1 - \frac{MSE}{\frac{1}{m}\sum_{i=1}^{m}(y_i - \bar{y})^2} \cdot \bar{y}$$

5. Многомерная регрессия, проблема мультиколлинеарности. Полиномиальная регрессия. Решение проблемы переобучения: L1-регуляризация (Lasso), L2- Регуляризация (гребневая регрессия), эластичная сеть.

Многомерная регрессия: Многомерная линейная регрессия – это метод моделирования, который предсказывает зависимую переменную на основе нескольких независимых переменных. Это расширение простой линейной регрессии на случай с несколькими предикторами (факторами). Многомерная регрессия моделирует зависимость путем

построения линейной комбинации различных предикторов.

Проблема мультиколлинеарности: Мультиколлинеарность возникает, когда в многомерной регрессии независимые переменные сильно коррелируют между собой. Это может создать проблемы при оценке коэффициентов регрессии и их интерпретации. Мультиколлинеарность может быть обнаружена при помощи анализа коэффициентов корреляции и вариансных инфляторов. Для устранения проблемы мультиколлинеарности можно использовать методы, такие как понижение размерности, исключение одного из коррелирующих предикторов или применение регуляризации.

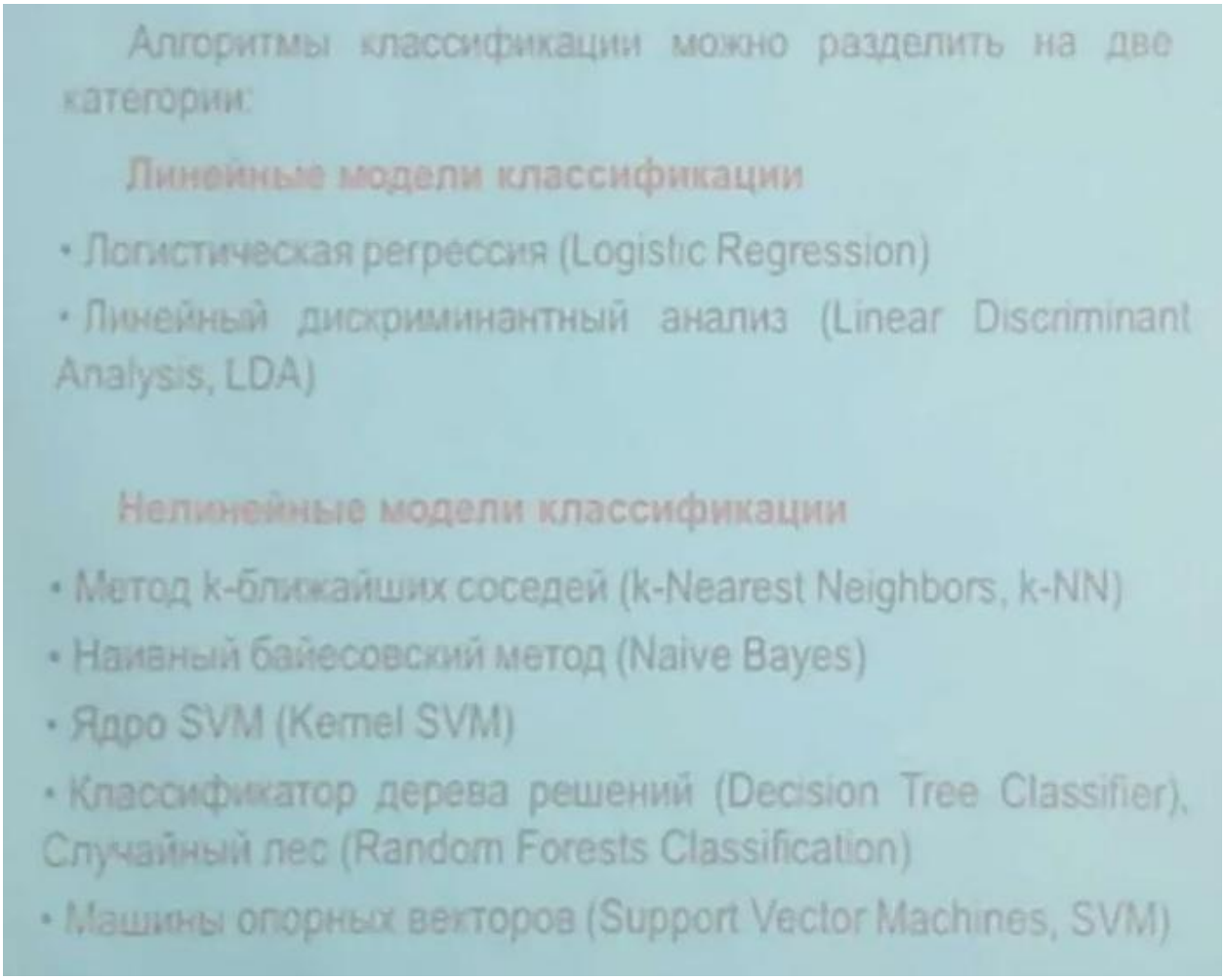
Полиномиальная регрессия: Полиномиальная регрессия – это метод моделирования, который использует полиномиальные функции для предсказания зависимой переменной на основе независимой переменной. Вместо использования простой линейной функции, полиномиальная регрессия включает степени переменных, позволяя выразить более сложные зависимости между переменными.

Регуляризация:
- L1-регуляризация (Lasso): L1-регуляризация добавляет штраф к сумме абсолютных значений коэффициентов регрессии. Она позволяет уменьшить вес незначимых переменных, приводя к отбору признаков и упрощению модели.
- L2-регуляризация (гребневая регрессия): L2-регуляризация добавляет штраф к сумме квадратов коэффициентов регрессии. Она позволяет уменьшить влияние мультиколлинеарности и сделать модель более устойчивой.
- Эластичная сеть (Elastic Net): Эластичная сеть комбинирует L1- и L2-регуляризацию. Данный метод применяется для снижения размерности модели и учета мультиколлинеарности.

6. Задача классификации. Алгоритмы классификации ML. Проблема дисбаланса классов. Метрики качества классификации: accuracy, матрица ошибок, precision, recall, F1- мера. AUC-ROC (площадь под кривой ошибок), ROC-кривая.

Задача классификации: Задача классификации в машинном обучении заключается в прогнозировании меток классов для новых, непрогнозируемых данных на основе обучающих данных с известными

метками классов. Она включает разделение данных на предопределенные классы на основе их характеристик и свойств, которые изучаются из обучающего набора данных.



Проблема дисбаланса классов: Проблема дисбаланса классов возникает, когда один класс представлен значительно большим количеством объявлений, чем другой класс. В таком случае модель может быть смещена в сторону доминирующего класса и иметь сложности в обнаружении редкого класса. Для решения этой проблемы можно использовать методы перевзвешивания классов, аугментацию данных, определение пороговых значений и другие подходы.

Метрики качества классификации:
1. Accuracy (точность) - это самая простая метрика, которая показывает, как часто классификатор предсказывает правильные метки классов.
2. Матрица ошибок - это таблица, которая показывает количество верно и неверно предсказанных меток классов.
3. Precision (точность) - это доля верно предсказанных положительных случаев относительно всех положительных предсказаний.
4. Recall (полнота) - это доля верно предсказанных положительных

случаев относительно всех истинных положительных случаев.

5. F1-мера - это гармоническое среднее между precision и recall, предоставляющее баланс между точностью и полнотой.

6. AUC-ROC - это площадь под кривой ошибок, которая представляет собой графическую метрику, измеряющую производительность классификатора. ROC-кривая показывает отношение между True Positive Rate и False Positive Rate при изменении порогового значения для классификации.

7. Линейная модель классификации. Логистическая регрессия как линейный классификатор. Функция потерь. Различия между линейной регрессией и логистической регрессией.

Линейная модель классификации: Линейная модель классификации - это метод, который пытается разделить данные на два класса путем использования линейной функции. Такая модель строит гиперплоскость в многомерном пространстве, чтобы разделить данные на два класса. Линейная модель предназначена для бинарной классификации, где входные данные представляются в виде вектора признаков, и каждый признак имеет свой вес.

Логистическая регрессия как линейный классификатор: Логистическая регрессия - это линейный классификатор, который использует логистическую функцию для предсказания вероятности принадлежности к классу. Она относится к семейству обобщенных линейных моделей. Логистическая регрессия используется для решения задач бинарной классификации, где классификация происходит на основе вероятности принадлежности к одному из двух классов.

Функция потерь в логистической регрессии: В логистической регрессии используется функция потерь, называемая логистической функцией потерь (log loss) или перекрестной энтропией (cross-entropy loss). Эта функция минимизирует разницу между предсказанной вероятностью классов и фактическими метками классов. Она штрафует модель за неправильные предсказания и стремится уменьшить различия между предсказаниями и истинными значениями.

Модели, возвращающие *вероятности классов*, использует **перекрестную энтропийную потерю (cross-entropy).**

$$\mathcal{L}_{log}(\mathbf{w}) = \sum_{n=1}^{N} H(p_n, q_n) = -\sum_{n=1}^{N}\sum_{c=1}^{C} p_{n,c} log(q_{n,c})$$

На основе истинных вероятностей (0 или 1) и прогнозируемых вероятностей для экземпляров и классов.

Бинарная классификация:

(C=2): $\mathcal{L}_{log}(\mathbf{w}) = -\sum_{n=1}^{N} \left[y_n log(\hat{y}_n) + (1-y_n) log(1-\hat{y}_n) \right]$

Штраф растет экспоненциально по мере увеличения разницы между p и q.

Часто используется вместе с потерей L2 (или L1)

$$\mathcal{L}_{log}'(\mathbf{w}) = \mathcal{L}_{log}(\mathbf{w}) + \alpha \sum_i w_i^2$$

Различия

Критерий сравнения	Линейная регрессия	Логистическая регрессия
Ввод – вывод	Ввод – независимая переменная (или несколько), вывод – прогноз зависимой переменной (задача регрессии)	Ввод – независимая переменная (или несколько), вывод – вероятность принадлежности к группе
Линия наилучшего соответствия	Прямая	Кривая
Способ минимизации ошибки	Метод наименьших квадратов	Метод максимального правдоподобия.
Результат на выходе	**Вещественное число** (прогноз изменений)	**Число в интервале от 0 до 1.** Можно осуществить бинарную классификацию: если число ниже порогового значения — то объект относится к 0 («нет»), а если выше — то к 1 («да»)
Цель	**Прогнозирование** линейных трендов	**Классификация** (при добавлении соответствующего правила), но классификатором при этом не является

8. Метрический классификатор. k-ближайшие соседи (k-Nearest Neighbor – kNN). Метрики расстояний. Алгоритм k-ближайших соседей. Наивный байесовский классификатор.

Метрический классификатор — это алгоритм, основанный на оценивании сходства объектов и классификации исходного объекта на основе ближайшего к нему объекта из обучающего набора данных.

Один из известных метрических классификаторов — k-ближайшие соседи (kNN). В этом алгоритме для классификации нового объекта определяется k ближайших к нему объектов из обучающего набора.

Затем новый объект классифицируется путем применения большинства голосов среди его k ближайших соседей.

Основными компонентами алгоритма k-ближайших соседей являются:

1. Метрики расстояний: Для определения близости между объектами требуется метрика расстояния. Некоторые из распространенных метрик включают евклидово расстояние, манхэттенское расстояние и расстояние Минковского.

2. Алгоритм k-ближайших соседей: Для классификации нового объекта сначала определяются его k ближайших соседей из обучающего набора данных, используя выбранную метрику расстояния. Затем новый объект классифицируется на основе большинства голосов среди его k ближайших соседей.

Однако алгоритм k-ближайших соседей имеет некоторые ограничения, включая необходимость правильного выбора значений k и метрики расстояния, а также проблемы с вычислительной сложностью для больших наборов данных.

Еще одним классификатором является наивный байесовский классификатор, основанный на байесовской вероятности. В этом алгоритме предполагается, что значения признаков независимы, и классификация основана на оценке вероятности принадлежности объекта к определенному классу. Для обучения наивного байесовского классификатора используются данные обучающего набора, из которых вычисляются априорные и условные вероятности для каждого класса. Затем для классификации нового объекта вычисляются вероятности его принадлежности каждому классу и выбирается класс с наивысшей вероятностью.

9. Машина опорных векторов SVM. Линейный SVM. Ядерные функции (kernel function) и спрямляющие пространства.

Машина опорных векторов (SVM) - это алгоритм для задач классификации и регрессии, который основан на поиске оптимальной разделяющей гиперплоскости между двумя классами данных. Основная идея SVM состоит в том, чтобы найти гиперплоскость в n-мерном пространстве (где n - число признаков), которая максимально разделяет точки двух классов.

Линейный SVM строит разделяющую гиперплоскость в виде линейной комбинации входных признаков. Цель состоит в том, чтобы найти такую гиперплоскость, которая максимизирует расстояние между классами, называемое отступом. Максимизация отступа позволяет достичь наилучшей генерализации и повышает способность модели к классификации новых точек данных.

Однако в некоторых случаях данные могут быть нелинейно разделимыми в исходном пространстве. Тогда вместо построения разделяющей гиперплоскости в исходном пространстве, можно преобразовать данные в новое пространство более высокой размерности, где они станут линейно разделимыми. Этот процесс называется спрямляющим пространством.

Ядерная функция - это функция, которая позволяет вычислять скалярное произведение двух векторов в спрямляющем пространстве, не вычисляя самих векторов. Ядерные функции позволяют избежать вычислительной сложности преобразования данных в спрямляющее пространство. Они определены в исходном пространстве и могут быть выбраны для различных типов данных.

Некоторые из известных ядерных функций включают линейное ядро, полиномиальное ядро, радиальное базисное функциональное ядро (RBF) и сигмоидное ядро. Выбор подходящей ядерной функции влияет на способность SVM к разделению данных и к обобщению на новые точки данных.

10. Древовидные модели. Деревья решений. Алгоритм CART. Дерево регрессии. Дерево классификации.

Древовидные модели являются гибкими и интерпретируемыми алгоритмами машинного обучения, которые представляют данные в виде древовидной структуры. Два часто используемых типа древовидных моделей - это деревья решений и деревья регрессии.

Дерево решений является алгоритмом классификации и регрессии. Оно строит древовидную структуру, где каждый узел представляет признак, а каждое ребро - ветвь, связывающая узлы в соответствии с правилом разбиения. Дерево решений предоставляет простой и понятный способ принятия решений, позволяя делать прогнозы на основе значений

признаков.

Алгоритм классификации и регрессии CART (Classification and Regression Tree) - это один из самых популярных алгоритмов построения дерева решений. Он использует принцип жадной оптимизации для поиска наилучшего разбиения признаков на каждом узле дерева. Алгоритм CART строит дерево последовательно, на каждом шаге разбивая данные на две новые группы, максимизируя прирост информации или уменьшение неопределенности.

Дерево регрессии используется для задач регрессии, где требуется предсказать непрерывное значение целевой переменной. Каждый лист дерева содержит предсказанное значение регрессии. Дерево регрессии может использоваться для решения проблем, таких как прогнозирование продаж, оценка стоимости недвижимости и т.д.

Дерево классификации используется для задач классификации, где требуется отнести объекты к определенным категориям или классам. Каждый лист дерева представляет классификацию. Деревья классификации могут быть использованы в различных областях, таких как медицина, финансы, маркетинг и другие.

11. Ансамблевые методы . Ансамбли (композиции) деревьев решений. Бэггинг. Случайный лес

Ансамблевые методы в машинном обучении объединяют несколько моделей, чтобы получить более сильную и устойчивую модель. Одним из популярных типов ансамблевых методов являются ансамбли, или композиции, деревьев решений.

Бэггинг (bagging) является одним из методов ансамблевого обучения, где каждая модель строится на основе подмножества обучающих данных с использованием процедуры рэсэмплинга с возвращением (bootstrap). В контексте ансамбля деревьев решений, каждое дерево обучается на новом подмножестве данных, что позволяет получить разнообразные деревья.

Случайный лес (random forest) является алгоритмом, основанным на бэггинге. В случайном лесу строится множество деревьев решений, где каждое дерево обучается на случайном подмножестве признаков.

Каждое дерево принимает решение, и финальная классификация или регрессия определяется путем голосования или усреднения результатов.

12. Ансамблевые методы. Ансамбли (композиции) деревьев решений. Бустинг.

Бустинг - это алгоритм ансамблевого обучения, который объединяет слабые модели для создания сильной модели. В отличие от бэггинга, где модели обучаются независимо, в бустинге модели добавляются последовательно и каждая новая модель фокусируется на исправлении ошибок, сделанных предыдущими моделями.

Другим алгоритмом бустинга является градиентный бустинг (Gradient Boosting). Градиентный бустинг работает путем обучения модели на остатках предыдущих моделей, что позволяет минимизировать ошибку на каждом шаге обучения. В градиентном бустинге используется градиентный спуск для настройки весов моделей, и финальная модель строится путем комбинирования всех моделей.

13. Ансамблевые методы. Ансамбль стекинга. Современные имплементации градиентного бустинга (CatBoost, LightGBM, XGBoost).

Ансамбль стекинга (stacking ensemble) — это метод ансамблевого обучения, который комбинирует прогнозы нескольких моделей с помощью метамодели. Вместо использования простого голосования или усреднения результатов, модели первого уровня обучаются на обучающем наборе данных, а затем их прогнозы передаются в метамодель для получения конечного прогноза.

Ансамбль стекинга состоит из двух основных компонентов: моделей первого уровня (базовых моделей) и метамодели. Модели первого уровня обучаются на обучающей выборке, затем их предсказания используются как входные данные для метамодели. Метамодель обучается на основе предсказаний моделей первого уровня и выдает конечный прогноз.

Несколько современных имплементаций градиентного бустинга включают:

1. CatBoost: CatBoost - это высокоэффективная библиотека градиентного бустинга, разработанная компанией Яндекс. Она предоставляет реализацию алгоритма градиентного бустинга с категориальными признаками. CatBoost автоматически обрабатывает

категориальные переменные без необходимости кодирования. Он также может работать с большими объемами данных и предлагает удобные инструменты для настройки гиперпараметров.

2. LightGBM: LightGBM - это библиотека градиентного бустинга, разработанная Microsoft. Она предлагает быструю и эффективную реализацию градиентного бустинга, оптимизированную для больших наборов данных. LightGBM использует алгоритм гистограммного градиентного бустинга, что позволяет достичь высокой скорости обучения и более эффективной работы с памятью.

3. XGBoost: XGBoost - это популярная библиотека градиентного бустинга, которая обладает высокой скоростью и хорошей точностью. XGBoost предлагает широкие возможности настройки гиперпараметров и поддерживает распределенное обучение для обработки больших объемов данных. Он также обеспечивает встроенную поддержку категориальных переменных и имеет множество возможностей для контроля структуры дерева и его сложности.

14. Задача кластеризации в ML. Постановка задачи кластеризации. Методы оценки качества кластеризации.

Задача кластеризации в машинном обучении относится к области без учителя, и ее цель состоит в группировании объектов данных в непересекающиеся кластеры на основе их сходства. Кластеризация не требует предварительной разметки данных и позволяет выявить скрытые структуры или паттерны в данных.

Постановка задачи кластеризации:
1. Имеется набор данных без разметки, состоящий из N объектов.
2. Необходимо разделить эти объекты на K непересекающихся кластеров.
3. Целью является максимизация сходства объектов внутри каждого кластера и минимизация сходства между различными кластерами.

Методы оценки качества кластеризации используются для оценки, насколько хорошо алгоритм кластеризации справляется со своей задачей. Вот некоторые из распространенных методов оценки качества кластеризации:

1. Индекс силуэта (Silhouette Score) измеряет, насколько каждый объект в кластере согласуется со своим кластером в сравнении с другими кластерами. Значение индекса силуэта может варьироваться от -1 до 1, где значения ближе к 1 указывают на хорошую кластеризацию, а значения ближе к -1 указывают на плохую кластеризацию.

2. Индекс Данна (Davies-Bouldin Index) оценивает разделение кластеров, основываясь на межкластерных и внутрикластерных расстояниях. Чем ниже значение индекса Данна, тем лучше качество кластеризации.

3. Коэффициент согласованности (Cluster Cohesion) и разделение (Cluster Separation) очень полезны для оценки иерархической кластеризации. Коэффициент согласованности оценивает компактность кластеров, а коэффициент разделения оценивает сепарацию между кластерами.

4. Нормализованная взаимная информация (Normalized Mutual Information) используется для оценки качества кластеризации, особенно в случаях, когда имеется наличие разметки для сравнения с результатами кластеризации.

15. Задача кластеризации в ML. Алгоритмы кластеризации. Алгоритм kсредних.

К-минс, DBSCAN, иерархическая, гаусовская смесь
Алгоритм k средних разделяет объекты данных на заранее заданное количество кластеров, которое обозначается как k. Он работает по следующему принципу:

1. Инициализация центроидов: Начальные положения центроидов выбираются случайным образом в пространстве признаков.

2. Присвоение кластеров: Каждый объект данных присваивается к ближайшему центроиду на основе выбранной метрики расстояния, обычно евклидова или манхэттенская.

3. Перемещение центроидов: Центроиды каждого кластера пересчитываются, как средние значения признаков объектов, отнесенных к этому кластеру.

4. Повторение шагов 2 и 3: Шаги присвоения кластеров и перемещения центроидов повторяются до тех пор, пока центроиды не стабилизируются или достигнуто максимальное количество итераций.

5. Представление результата: Результатом работы алгоритма является набор кластеров и их соответствующих центроидов.

16. Задача кластеризации. Алгоритм k-средних. Иерархическая кластеризация.

Иерархическая кластеризация - это метод кластеризации, который строит иерархическую структуру кластеров. Он не требует задания числа кластеров заранее и может быть представлен в виде дерева или дендрограммы.

В иерархической кластеризации объекты данных объединяются на основе их сходства. Существуют два основных типа иерархической кластеризации:

1. Агломеративная иерархическая кластеризация: Начинается с каждого объекта данных в отдельном кластере, а затем последовательно объединяет ближайшие кластеры, пока не получится единственный кластер.

2. Делительная иерархическая кластеризация: Начинается с одного кластера, содержащего все объекты данных, и затем последовательно разделяет кластеры на более мелкие на основе критерия разделения, пока каждый объект данных не будет отнесен к своему собственному кластеру.

В агломеративной иерархической кластеризации используется мера близости, такая как евклидово расстояние или коэффициент корреляции, для определения близости между кластерами. Различные меры близости, такие как односвязанность (single-linkage), полная связанность (complete-linkage) и средняя связанность (average-linkage), могут быть использованы для объединения кластеров.

17. Задача кластеризации. Алгоритм DBSCAN. Гуассовы смеси.

Алгоритм DBSCAN (Density-Based Spatial Clustering of Applications with Noise) - это алгоритм кластеризации на основе плотности. DBSCAN определяет кластеры, исходя из плотностей объектов данных

в их окрестностях. Он может обнаруживать кластеры любой формы и отделять выбросы, объекты не принадлежащие ни к одному кластеру.

Принцип работы DBSCAN:
1. Выбирается случайный объект данных, который еще не был посещен.
2. Если объект имеет достаточное число соседей внутри заданного радиуса (eps), то он считается ядром (core point).
3. Все объекты в заданном радиусе вокруг ядра добавляются к кластеру.
4. Процесс повторяется для каждого нового объекта в кластере, расширяя его, пока возможно.
5. Если объект не является ядром, но находится в пределах заданного радиуса другого ядра, он считается граничным (border point) и добавляется к кластеру.
6. Если объект не является ядром и не находится в пределах заданного радиуса ни одного ядра, он считается выбросом (noise point).

Гуассовы смеси (Gaussian Mixture Models - GMM) - это вероятностная модель, которая моделирует данные как смесь нескольких гауссовых распределений. Каждое распределение соответствует отдельному кластеру в данных.

Принцип работы GMM:
1. Выбирается количество компонентов распределения (кластеров) для моделирования данных.
2. Вычисляются параметры каждой компоненты, включая средние значения и ковариационные матрицы.
3. Вычисляется апостериорная вероятность каждого объекта данных принадлежать к каждой компоненте.
4. Каждый объект относится к компоненте с наивысшей апостериорной вероятностью.
5. Модель затем обновляется, пересчитывая параметры каждой компоненты на основе присваивания объектов кластерам.
6. Процесс повторяется до достижения сходимости или достижения максимального числа итераций.

18. Глубокое обучение. Глубокие нейронные сети (НС). Модель искусственного нейрона. Функции активации.

Глубокое обучение - это подраздел машинного обучения, который фокусируется на обучении глубоких нейронных сетей (нейронных сетей с несколькими слоями). Глубокое обучение позволяет моделям

автоматически извлекать представления из данных и решать сложные задачи, такие как распознавание изображений, обработка естественного языка и генерация текста.

Глубокая нейронная сеть - это архитектура модели, состоящая из множества последовательно соединенных слоев нейронов. Каждый слой принимает представление от предыдущего слоя и обрабатывает его с помощью весов и функции активации. Последний слой обычно является выходным слоем и используется для предсказания конечного результата.

Модель искусственного нейрона - это модель, которая имитирует работу биологических нейронов. Искусственный нейрон получает входные сигналы, взвешивает их и применяет функцию активации для вычисления выходного сигнала. Он является основным строительным блоком нейронной сети.

Функции активации - это функции, которые определяют активацию, или переходное состояние, искусственных нейронов в нейронной сети. Они добавляют нелинейность в сеть, позволяя ей воспроизводить сложные нелинейные функции и имитировать сложные взаимодействия в данных. Некоторые из распространенных функций активации включают в себя:

1. Сигмоидная функция (Sigmoid): Приводит значения к диапазону [0, 1]. Широко использовалась в прошлом, но сейчас менее популярна из-за проблемы затухания градиента.

2. Гиперболический тангенс (Tanh): Приводит значения к диапазону [-1, 1]. По сравнению со сигмоидной функцией, имеет сдвиг в 0, что упрощает обработку отрицательных значений.

3. Функция активации ReLU (Rectified Linear Unit): Очень часто используется на практике. Возвращает 0, если входное значение отрицательно, и возвращает само значение, если оно положительно.

4. Softmax: Часто используется в выходном слое для задач многоклассовой классификации. Преобразует значения в соответствующие вероятности, сумма которых равна 1.

19. Базовая архитектура нейронных сетей. Многослойный персептрон (MLP).

Базовая архитектура нейронных сетей включает в себя набор искусственных нейронов, организованных в слои, и связи между ними. Самые основные типы слоев в нейронных сетях включают входной слой, скрытые слои и выходной слой.

Многослойный персептрон (Multi-Layer Perceptron, MLP) является одной из самых распространенных архитектур нейронных сетей. Он состоит из нескольких слоев искусственных нейронов: входного слоя, одного или нескольких скрытых слоев и выходного слоя.

- Входной слой: Принимает входные сигналы или признаки и передает их в скрытый слой. Количество нейронов входного слоя соответствует количеству признаков в данных.

- Скрытые слои: Принимают входные сигналы от предыдущего слоя, обрабатывают их и передают выходные сигналы в следующий слой. Количество и размер скрытых слоев в MLP может варьироваться в зависимости от архитектуры нейронной сети.

- Выходной слой: Принимает выходные сигналы от последнего скрытого слоя и производит окончательные предсказания или результаты в соответствии с типом задачи (например, классификация или регрессия).

Каждый нейрон в MLP связан с нейронами предыдущего слоя и следующего слоя с помощью весов, которые определяют силу связи. Входные сигналы пропускаются через нейроны с применением функции активации (такой как ReLU или сигмоид) для генерации активации нейрона.

20. Полносвязные нейронные сети (FCNN). Алгоритм обратного распространения ошибки (Back-Propagation). Функция потерь.

Полносвязные нейронные сети (Fully Connected Neural Networks, FCNN), также известные как многослойные персептроны (Multi-Layer Perceptrons, MLP), представляют собой архитектуру нейронных сетей, в которой каждый нейрон в одном слое связан с каждым нейроном в следующем слое. Такие сети обладают свойством универсальной аппроксимации, то есть они могут приближать любые непрерывные

функции при достаточной архитектурной сложности.

Алгоритм обратного распространения ошибки (Back-Propagation) - это основной алгоритм обучения глубоких нейронных сетей, как FCNN. Он используется для обновления весов нейронов в сети, минимизируя функцию потерь (loss function) путем передачи ошибки в обратном направлении через сеть.

Процесс обратного распространения ошибки включает в себя несколько шагов:

1. Прямой проход (Forward pass): Данные приводятся к входному слою сети, и выходные значения передаются через каждый слой, с применением активационных функций, чтобы сгенерировать предсказания.

2. Вычисление ошибки (Error calculation): Вычисляется разница между предсказаниями сети и реальными значениями целевой переменной. Это позволяет определить, насколько сеть ошибается.

3. Обратное распространение (Backward pass): Ошибка распространяется назад от последнего слоя к первому, с использованием метода градиентного спуска для вычисления градиентов функции потерь по весам сети.

4. Обновление весов (Weights update): Веса сети обновляются путем вычитания градиента функции потерь по весам, умноженного на некоторый коэффициент обучения (learning rate). Это позволяет сети корректировать веса для достижения минимума функции потерь и улучшения качества предсказаний.

Функция потерь (Loss function) определяет, насколько хорошо модель сети предсказывает реальные значения. В задачах регрессии часто используется средняя квадратичная ошибка (Mean Squared Error, MSE), а для задач классификации - кросс-энтропия (Cross-Entropy Loss). Функция потерь также может зависеть от типа задачи и свойств данных.

21. Полносвязные нейронные сети (FCNN). Фреймворк TensorFlow и API Keras для построения FCNN. Решение задачи регрессии и классификации с помощью FCNN

Фреймворк TensorFlow - это открытый и гибкий фреймворк для глубокого обучения, который разработан компанией Google. Он предоставляет инструменты для создания и обучения различных моделей машинного обучения, включая полносвязные нейронные сети (Fully Connected Neural Networks, FCNN).

API Keras для построения FCNN. - является высокоуровневым интерфейсом для построения и обучения моделей глубокого обучения. Он работает поверх TensorFlow и предоставляет простой и интуитивно понятный способ создания нейронных сетей. Keras поддерживает различные типы моделей, включая FCNN.

Решение задачи регрессии

Подготовка данных: Загрузите данные и выполните предварительную обработку, такую как масштабирование функций или обработку пропущенных значений.

Создание модели: Используя TensorFlow и API Keras, определите архитектуру модели, включая входной слой, скрытые слои и выходной слой. В FCNN каждый нейрон в слое полностью связан со всеми нейронами в следующем слое.

Компиляция модели: Задайте функцию потерь и оптимизатор для обучения модели. В задаче регрессии обычно используется среднеквадратическая ошибка (Mean Squared Error, MSE) в качестве функции потерь.

Обучение модели: Подготовьте обучающий набор данных и запустите процесс обучения, вызвав метод fit(). Во время обучения модель будет настраивать веса своих нейронов для минимизации функции потерь.

Оценка модели: Оцените производительность обученной модели на отложенном наборе данных, используя метрики, такие как среднеквадратическая ошибка или коэффициент детерминации (R-squared).

классификации с помощью FCNN.

Подготовка данных: Кодируйте целевую переменную вектором значений (One-Hot Encoding) для многоклассовой классификации.

Создание модели: Последний слой модели должен быть настроен для соответствия количеству классов и использовать функцию активации, такую как softmax, для предсказания вероятности каждого класса.

Компиляция модели: В задаче классификации обычно используется функция потерь распространения задачи (Cross-Entropy Loss) и оптимизатор, такой как стохастический градиентный спуск (Stochastic Gradient Descent, SGD).

Обучение модели: Процесс обучения и оценки производительности модели аналогичен задаче регрессии.

22. Сверточные НС. Общая структура сверточного слоя. Операция «свертки». Параметры сверточного слоя.

Сверточные НС. - это тип нейронных сетей, которые обычно используются для анализа изображений и других типов данных с пространственной структурой, например, звуковых сигналов.

Общая структура сверточного слоя.

Анализируемая область (input feature map): это входные данные, которые обычно представляются в виде трехмерного тензора (высота x ширина x каналы).

Сверточные фильтры (convolutional filters): это матрицы весов, которые перемещаются по входным данным для выполнения операции свертки. Сверточные фильтры извлекают признаки из анализируемой области.

Матрица активаций (activation map): это результат операции свертки, представленный в виде трехмерного тензора. Каждое значение в матрице активаций представляет выход нейрона, который активируется при обработке сверточными фильтрами.

Полулокальная субдискретизация (pooling): эта операция уменьшает пространственное разрешение матрицы активаций, объединяя несколько значений в одно значение. Полулокальная субдискретизация позволяет уменьшить количество параметров в модели и обрабатывать более абстрактные признаки.

Нелинейное преобразование: на каждом шаге применяется нелинейная функция активации, например, ReLU (Rectified Linear Unit), для добавления нелинейности в модель.

Операция «свертки». -это основная операция в сверточных слоях. Она выполняется путем перемещения сверточных фильтров по всей анализируемой области и вычисления покомпонентного произведения элементов фильтра и соответствующих элементов входных данных. Затем произведения суммируются, образуя одно значение матрицы активаций. Этот процесс повторяется для каждого фильтра и каждой позиции входных данных.

Параметры сверточного слоя.

Размер сверточного фильтра (kernel size): определяет размер области, которая будет анализироваться сверточным фильтром.

Число сверточных фильтров (number of filters): определяет, сколько фильтров будет использоваться для извлечения различных признаков из входных данных.

Шаг свертки (stride): определяет, какое количество пикселей будет пропускаться при перемещении сверточного фильтра по анализируемой области.
Наружападание (padding): определяет, как обрабатывать краевые пиксели входных данных. Например, можно добавить нули по краям входных данных (padding) для сохранения размера.
Функция активации (activation function): определяет нелинейное преобразование, применяемое к выходу каждого нейрона в сверточном слое.
Другие параметры могут варьироваться в зависимости от конкретной реализации сверточного слоя.

23. Сверточные НС. Архитектура сверточных НС. Трансферное обучение (Transfer Learning) и тонкая настройка (Fine Tuning).

Архитектура сверточных НС. - является основным инструментом в обработке изображений и анализе видео. Она включает в себя несколько слоев, каждый из которых выполняет различные операции над входными данными и извлекает признаки с разной степенью абстракции.
Трансферное обучение (Transfer Learning) - это метод обучения нейронных сетей, который позволяет использовать знания, полученные из одной предварительно обученной модели, для решения другой задачи. Вместо того, чтобы обучать модель с нуля, можно использовать предобученную модель, которая была обучена на больших объемах данных (например, на изображениях ImageNet) и имеет хорошие особенности обобщения.
тонкая настройка (Fine Tuning). - это дополнительный шаг в трансферном обучении, который заключается в обучении предобученной модели на новых данных с небольшим коэффициентом обучения. Этот процесс позволяет адаптировать веса и параметры модели к новой задаче, учитывая специфические особенности нового набора данных. Часто тонкая настройка осуществляется на выходных слоях и не так сильно влияет на веса начальных слоев, которые могут быть сохранены или заморожены.

24. Задача понижения размерности. Метод главных компонент (PCA). Ал□горитм PCA. Нелинейный метод главных компонент – ядерный PCA.

Задача понижения размерности. - относится к области обработки данных и машинного обучения и заключается в уменьшении

количества признаков (измерений) в пространстве данных. Она обычно возникает, когда у нас есть множество признаков, описывающих объекты, и мы хотим уменьшить размерность данных, сохраняя при этом основные характеристики и понимание оригинальных данных.
Метод главных компонент (PCA). - это статистический метод, используемый для понижения размерности данных. Он позволяет найти линейные комбинации исходных признаков (называемые главными компонентами), которые объясняют наибольшую долю дисперсии в данных. Это позволяет сократить размерность данных, удалив нерелевантные и коррелирующие признаки, и сосредоточиться на важных аспектах данных.

Алгоритм PCA.
Стандартизация данных: Если признаки имеют разные масштабы, их следует стандартизировать, чтобы каждый признак внес равный вклад в анализ.
Вычисление ковариационной матрицы: В этом шаге вычисляется ковариационная матрица, которая показывает связь между признаками и их дисперсию.
Вычисление собственных значений и собственных векторов: Далее, вычисляются собственные значения и собственные векторы ковариационной матрицы. Собственные векторы представляют главные компоненты, а собственные значения показывают, сколько дисперсии объясняет каждая главная компонента.
Выбор главных компонент: Главные компоненты выбираются в порядке убывания соответствующих им собственных значений, чтобы сохранить наибольшую долю дисперсии данных.
Проекция данных: В конечном счете, исходные данные проецируются на выбранные главные компоненты, что позволяет получить новое представление данных с меньшей размерностью.
Нелинейный метод главных компонент – ядерный PCA. - является расширением метода главных компонент, который позволяет обрабатывать нелинейные зависимости в данных. В отличие от обычного PCA, ядерный PCA использует ядерные функции, такие как радиально базисная функция (RBF), чтобы преобразовывать данные в высокоразмерное пространство, где они могут быть разделены линейно. Затем применяется обычный PCA к преобразованным данным. Это позволяет обнаружить нелинейную структуру в данных и получить линейное представление в новом пространстве.

25. Задача понижения размерности. Методы снижения размерности. Не□линейные методы снижения размерности: t-SNE, Isomap

Методы снижения размерности. - используются для сокращения размерности данных, уменьшения количества признаков и представления данных в пространстве меньшей размерности. Это может быть полезно для визуализации данных, устранения шума, ускорения алгоритмов обучения или улучшения обобщающей способности моделей машинного обучения.
Нелинейные методы снижения размерности:
t-SNE, - это алгоритм снижения размерности, который широко используется для визуализации данных. Он основывается на вероятностной интерпретации набора точек в исходном пространстве и пытается сохранить относительные расстояния между этими точками в пространстве меньшей размерности. t-SNE обладает способностью обнаруживать сложные структуры данных, такие как кластеры и локальные группы точек. Он особенно полезен для визуализации высокоразмерных данных в двух или трех измерениях.
Isomap. - это метод снижения размерности, который моделирует геометрию данных, основываясь на идеи сохранения геодезических расстояний между близкими точками в исходном и сниженном пространствах. Он строит граф связей между точками данных, а затем вычисляет геодезические расстояния (наименьшие пути) на этом графе. Низкоразмерное представление данных получается путем нахождения глобальной конфигурации, которая сохраняет эти расстояния. Isomap позволяет улавливать нелинейные зависимости между признаками данных и может быть полезен при работе с временными рядами, изображениями или другими сложными структурами данных.

26. Задача понижения размерности. Методы выбора признаков (Feature Selection).

Задача понижения размерности.
Методы выбора признаков (Feature Selection). - Он заключается в отборе наиболее значимых признаков из исходного набора. Это позволяет упростить модель, снизить степень шума в данных, уменьшить вычислительную сложность алгоритма и повысить интерпретируемость модели.
Основанные на статистике: эти методы оценивают статистическую значимость каждого признака и выбирают наиболее важные. Примеры

таких методов включают t-тест, анализ дисперсии (ANOVA), информационные критерии (например, AIC и BIC) и корреляционный анализ.

Основанные на модели: эти методы используют модель машинного обучения для оценки важности каждого признака. Примеры таких методов включают регрессию с L1-регуляризацией (LASSO), методы отбора на основе деревьев (например, RandomForest и Gradient Boosting) и рекурсивный отбор признаков (RFE).

Основанные на вложенности: эти методы оценивают важность признаков, учитывая взаимодействие между ними. Примеры таких методов включают взаимную информацию, коэффициенты Шеррина и методы на основе деревьев (например, Recursive Feature Elimination with Cross-Validation, RFECV).

27. Основы обработки естественного языка. Базовые методы векториза□ции текста. Bag of words, TF-IDF. Задача тематического моделирования (мягкая кластеризация)

Основы обработки естественного языка.

Базовые методы векторизации текста.

Bag of words, - Метод "мешок слов" представляет текст как неупорядоченный набор слов, игнорируя порядок их следования.

Процесс включает следующие шаги:

Токенизация: разделение текста на отдельные слова или токены.

Построение словаря: создание уникальных слов (или токенов) из текстового корпуса.

Векторизация: преобразование текстов в векторы, где каждая позиция соответствует слову из словаря, а значение указывает на наличие или количество вхождений данного слова в текст. Это может быть просто бинарное (0/1) значение или количество вхождений слова.

Представление документов: каждый документ (текст) представляется вектором, где каждая позиция отображает слово из словаря, а значение указывает на вхождение или количество вхождений соответствующего слова в текст.

TF-IDF. - TF-IDF является методом, который учитывает не только количество вхождений слов в текст, но и их важность в контексте всего корпуса текстов. Процесс включает следующие шаги:

Токенизация и построение словаря: аналогично Bag of Words.

Вычисление TF: определение относительной частоты каждого слова в каждом документе (TF = (количество вхождений слова в документе) / (количество слов в документе)).
Вычисление IDF: определение инверсии частоты слова в документах (IDF = log((количество документов) / (количество документов, в которых встречается слово))).
Вычисление TF-IDF: перемножение значений TF и IDF для каждого слова и документа.
Задача тематического моделирования (мягкая кластеризация) - Задача тематического моделирования в NLP состоит в выявлении скрытых тематик в коллекции документов. Мягкая кластеризация предполагает, что документ может принадлежать нескольким тематикам с различной степенью принадлежности. Одним из популярных подходов к тематическому моделированию является модель Latent Dirichlet Allocation (LDA), которая основана на вероятностных графических моделях. LDA позволяет выделить темы и определить вероятности принадлежности каждого документа к каждой теме.