1. **Понятие машинного обучения**

Машинное обучение (МЛ) — это область искусственного интеллекта, которая занимается разработкой алгоритмов и моделей, позволяющих компьютерам обучаться и принимать решения без явного программирования. Основная идея машинного обучения заключается в том, чтобы создавать системы, которые могут автоматически улучшать свои результаты на основе анализа данных.

**Основные элементы машинного обучения:**

1. **Данные**: МЛ опирается на данные для обучения. Это могут быть числовые, текстовые, графические или иные виды данных.
2. **Алгоритмы**: Способы обработки данных и извлечения закономерностей, такие как линейная регрессия, деревья решений, нейронные сети и др.
3. **Модели**: Результат применения алгоритма, который может прогнозировать или классифицировать новые данные.

**Классификация машинного обучения:**

1. **Обучение с учителем (Supervised Learning)**: Используются размеченные данные (где известны входы и ожидаемые выходы). Например, классификация и регрессия.
2. **Обучение без учителя (Unsupervised Learning)**: Используются неразмеченные данные, цель — найти скрытые закономерности. Например, кластеризация и ассоциативные правила.
3. **Обучение с подкреплением (Reinforcement Learning)**: Алгоритм взаимодействует с окружением, получает награды или наказания и обучается на основе обратной связи.

**Применение машинного обучения:**

* Рекомендательные системы (например, в Netflix или Spotify).
* Компьютерное зрение (распознавание лиц, объектов).
* Обработка естественного языка (перевод, чат-боты).
* Финансовые прогнозы (анализ рынков, управление рисками).
* Диагностика в медицине.

Машинное обучение — мощный инструмент, который находит применение в самых разных областях благодаря своей способности решать сложные задачи, анализировать большие объемы данных и принимать решения.

1. **Постановка задачи машинного обучения**

**Постановка задачи машинного обучения** — это формализация проблемы, которую необходимо решить с помощью методов машинного обучения. Этот процесс включает определение цели, данных, подхода и критериев оценки модели. Четко сформулированная задача является важным шагом для успешного построения и обучения модели.

**Основные шаги постановки задачи машинного обучения:**

**1. Определение цели задачи**

* Что именно должна делать модель?  
  Примеры:
  + Классификация: Разделение объектов на заранее определенные категории (например, спам/не спам).
  + Регрессия: Прогнозирование численных значений (например, предсказание цен).
  + Кластеризация: Поиск групп в данных (например, сегментация клиентов).
  + Ранжирование: Упорядочивание объектов по степени релевантности (например, результаты поиска).

**2. Определение входных и выходных данных**

* **Входные данные**: Какие признаки (фичи) доступны для анализа?  
  Пример: В задаче прогнозирования цен на недвижимость входные данные могут включать площадь дома, местоположение, количество комнат.
* **Выходные данные**: Что нужно предсказать или классифицировать?  
  Пример: Цена дома, категория объекта.

**3. Сбор и анализ данных**

* Какие данные доступны для обучения?
  + Объем данных.
  + Качество данных (наличие пропусков, шумов).
  + Тип данных (структурированные или неструктурированные, текст, изображения и т.д.).
* Если данных недостаточно, нужно решить, где их можно получить.

**4. Выбор подхода к обучению**

* **Обучение с учителем**: Используется, если есть размеченные данные.
* **Обучение без учителя**: Используется, если метки (выходы) отсутствуют.
* **Обучение с подкреплением**: Используется для задач, где агент взаимодействует с окружением.

**5. Определение метрики качества**

Как оценивать, насколько хорошо модель выполняет задачу?  
Примеры метрик:

* Для классификации: Точность (accuracy), полнота (recall), F1-мера.
* Для регрессии: Среднеквадратичная ошибка (MSE), средняя абсолютная ошибка (MAE).
* Для кластеризации: Индекс силуэта, внутрикластерное расстояние.

**6. Разделение данных**

* **Обучающая выборка**: Используется для обучения модели.
* **Тестовая выборка**: Используется для проверки качества модели.
* **Валидационная выборка** (при необходимости): Для настройки гиперпараметров.

**7. Постановка ограничений**

* Ограничения по времени: Сколько времени доступно для обучения модели.
* Ограничения по ресурсам: Объем вычислительных мощностей.
* Ограничения по интерпретируемости: Насколько модель должна быть объяснимой.

1. **Задача восстановления регрессии**

Задача восстановления регрессии — это задача машинного обучения, в которой модель обучается предсказывать непрерывную числовую величину на основе входных данных. Цель регрессии — найти зависимость между входными признаками (фичами) и целевой переменной, чтобы на основе этой зависимости делать прогнозы для новых данных.

Основные элементы задачи регрессии:

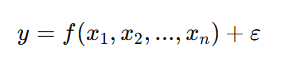
1. Данные: Входные признаки (фичи): Набор параметров, которые используются для построения модели.

Пример: площадь дома, количество комнат, возраст здания.

Целевая переменная: Непрерывная величина, которую нужно предсказать.

Пример: стоимость дома, температура, скорость ветра.

2. Модель: Модель регрессии аппроксимирует функцию, связывающую входные признаки с целевой переменной:

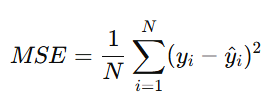


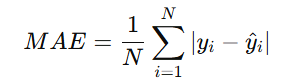
Где:

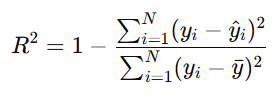
* x1,x2,..., xn​ — входные признаки,
* y — целевая переменная,
* f — функция зависимости,
* ε — случайная ошибка (шум).

3. Метрика качества:

Для оценки качества модели используются метрики, которые измеряют разницу между предсказанными значениями и реальными:

Среднеквадратичная ошибка (MSE): 

Средняя абсолютная ошибка (MAE): 

Коэффициент детерминации (R2): 

Где yi— реальные значения, yi^— предсказанные значения, yˉ ​ — среднее значение целевой переменной.

1. **Задача классификации**

Задача классификации состоит в присвоении объектам различных категорий или классов на основе определённых признаков. Решение задачи классификации – это построение модели, которая обучается на исходных данных, выявляя признаки, по которым определяется, к какому классу относится объект. Задача классификации решается, когда заранее известны классы, к которым необходимо отнести объекты.

***Некоторые методы классификации:***

-логистическая регрессия (используется для предсказания вероятности принадлежности объекта к одному из двух классов. Основана на логистической функции, которая преобразует линейную комбинацию входных признаков в вероятность)

-метод опорных векторов  
- k ближайших соседей (kNN)  
-деревья решений (модель в виде дерева, где каждый узел представляет собой проверку на определённый признак, а каждый лист — предсказание класса)  
-случайный лес (строит множество деревьев решений и объединяет их предсказания для улучшения точности и устойчивости модели).

В **классификации** результирующий признак дискретный (бинарный, номинальный / категориальный, упорядоченный)

1. **Метрическая классификация**

Метрические методы классификации основываются на расчете расстояния между объектами (отсюда и их название "метрические", от метрики). Существуют различные метрики, которые используются для расчета расстояния между объектами. Например, используются следующие меры расстояния:

1) Евклидово расстояния: длина отрезка прямой между двумя точками в евклидовом пространстве.

2) Манхэттенское расстояние: расстояние между двумя точками равно сумме модулей разностей их координат.

3) Косинусное расстояние — мера схожести между векторами, полезна в задачах обработки текстов.

Метрические методы, такие как метод k-ближайших соседей (kNN), классифицируют объекты на основе расстояния до ближайших соседей. Для определения класса нового объекта используется большинство голосов среди k ближайших соседей. Эти методы хорошо подходят для задач, где схожесть объектов важна, например, в биометрии, анализе изображений или текстов. Преимущества таких подходов включают простоту реализации, интуитивность и универсальность, поскольку метрики можно адаптировать под конкретные данные. Однако метрические методы чувствительны к выбросам, масштабированию признаков и требуют значительных вычислительных ресурсов при больших объёмах данных, что делает предварительную обработку данных и выбор подходящей метрики критически важными этапами.

1. **Линейная регрессия**

Линейная регрессия (Linear regression) — это математическая модель, которая описывает связь нескольких переменных.

Задача регрессии — это предсказание одного параметра (Y) по известному параметру X, где X — набор признаков, характеризующий наблюдение.

В **регрессии** результирующий признак непрерывный (количественный).

Суть линейной регрессии состоит в том, чтобы найти линейную функцию, которая будет отражать зависимость между целевой (зависимой) переменной и независимыми признаками.

Методы линейной регрессии: метод наименьших квадратов, градиентный спуск.

Линейная регрессия (Linear regression) — это математическая модель, которая описывает связь между зависимой переменной (целевой) и одной или несколькими независимыми переменными (признаками). Она используется для задач регрессии, где целевая переменная Y является непрерывной и количественной. Суть линейной регрессии состоит в нахождении линейной функции, которая аппроксимирует зависимость между переменными. Уравнение линейной регрессии имеет вид:

y=w0+w1x1+w2x2+⋯+wnxn+ ε

Где ε— случайная ошибка (шум).

Основные методы обучения:

1. **Метод наименьших квадратов**, который минимизирует сумму квадратов отклонений между реальными и предсказанными значениями.
2. **Градиентный спуск**, использующий итеративную оптимизацию параметров для минимизации функции потерь.

Линейная регрессия находит применение в прогнозировании стоимости недвижимости, анализе рыночных данных, моделировании производственных процессов. Однако её эффективность снижается при наличии нелинейных зависимостей, выбросов и мультиколлинеарности признаков. Для оценки качества модели используют метрики: среднеквадратичную ошибку (MSE), среднюю абсолютную ошибку (MAE) и коэффициент детерминации (R2).

1. **Бинарная классификация**

**Бинарная классификация** — это задача машинного обучения, заключающаяся в определении принадлежности объекта к одному из двух возможных классов. Примером может быть определение, является ли электронное письмо спамом или нет, болен ли пациент или здоров, или клиент покинет компанию или останется. В бинарной классификации объект описывается набором признаков (XXX), а результат предсказания представляет собой метку класса (yyy), которая может принимать значения, например, 000 или 111.

Цель бинарной классификации — создать модель, которая точно определяет принадлежность объектов к классам на основе данных. В случае линейно разделимых классов задача заключается в построении классификатора в виде прямой (в двумерном пространстве) или гиперплоскости (в многомерном пространстве), которая разделяет объекты с минимальной ошибкой. Для этого используются различные алгоритмы машинного обучения.

**Модели для бинарной классификации**

Существует множество моделей, которые применяются для решения задач бинарной классификации, среди которых наиболее популярные:

1. **Логистическая регрессия**:  
   Основана на логистической функции, которая преобразует линейную комбинацию признаков в вероятность принадлежности к классу. Эта модель хорошо подходит для линейно разделимых данных и часто используется благодаря своей простоте и интерпретируемости.
2. **Классификатор Байеса**:  
   Использует вероятностный подход и теорему Байеса для предсказания класса. Он особенно эффективен для задач, где признаки независимы друг от друга (наивный Байес).
3. **Метод k-ближайших соседей (kNN)**:  
   Этот алгоритм относит объект к классу, которому принадлежит большинство из kkk ближайших соседей, определяемых с помощью метрик расстояния, таких как Евклидово или Манхэттенское.
4. **Метод опорных векторов (SVM)**:  
   Строит оптимальную гиперплоскость, которая максимизирует расстояние между классами. Подходит как для линейных, так и для нелинейных данных при использовании ядерных функций.
5. **Деревья решений и случайный лес**:  
   Интуитивно понятные методы, где дерево решений последовательно проверяет признаки, а случайный лес объединяет предсказания множества деревьев для повышения точности.
6. **Логистическая регрессия**

Логистическая регрессия используется при поиске связей между набором входных переменных и категориальной выходной переменной и может применяться как в бинарной, так и в многоклассовой классификации.

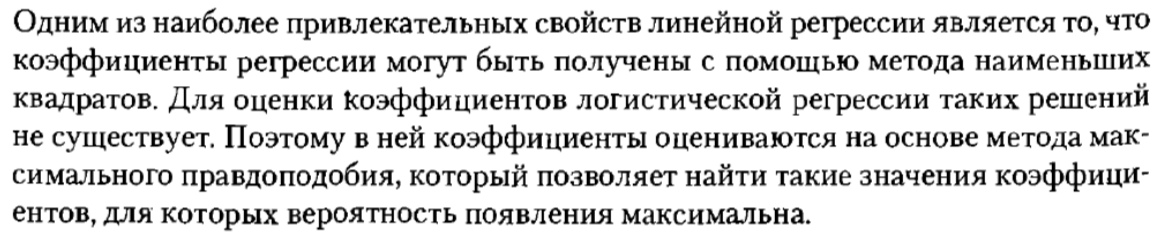
Признаковое пространство может быть разделено гиперплоскостью на два полупространства так, чтобы с одной стороны от этой гиперплоскости находились объекты с меткой класса «+», а с другой стороны – объекты с меткой класса «-» (или 1 и 0).

Логистическая регрессия прогнозирует вероятность отнесения объекта к классу «+». Для этого используется функция логистической кривой (сигмоиды).

Эта функция имеет вид ***, где z =***

*Принимает значения от 0 до 1, показывает вероятность отнесения объекта к классу.*

Логистическая регрессия может лежать в основе более точного классификатора, чем тот, в котором применяется линейная регрессия, поскольку сигмоидальная функция имеет форму кривой, а не прямой.



1. **Метод наименьших квадратов**

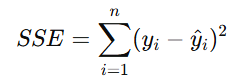
Метод наименьших квадратов минимизирует среднеквадратичную ошибку между реальным значением зависимой переменной и прогнозом, выданным моделью.

Метод наименьших квадратов в машинном обучении — это метод для подбора коэффициентов в линейной регрессии. Он заключается в минимизации суммы квадратов разностей между наблюдаемыми значениями и прогнозируемыми значениями, чтобы получить наилучшую линейную модель. Метод наименьших квадратов предполагает, что ошибки прогнозирования являются независимыми и случайными. Если это условие не выполняется, результаты могут быть недостоверными. Также метод наименьших квадратов предполагает линейную зависимость между предсказываемым и наблюдаемым значениями.

Предпосылки метода наименьших квадратов:

-случайные ошибки имеют нулевое математическое ожидание  
-случайные ошибки имеют нормальное распределение  
-случайные ошибки независимы (отсутствует автокорреляция ошибок)  
-дисперсия случайной ошибки одинакова для всех наблюдений (это называется гомоскедастичность).

**Метод наименьших квадратов (МНК)** минимизирует среднеквадратичную ошибку между реальными значениями зависимой переменной и прогнозом модели. Этот метод широко применяется для подбора коэффициентов в линейной регрессии. Цель метода — минимизировать сумму квадратов отклонений:



Он используется для нахождения наилучшей линейной зависимости между предсказаниями и наблюдаемыми значениями.

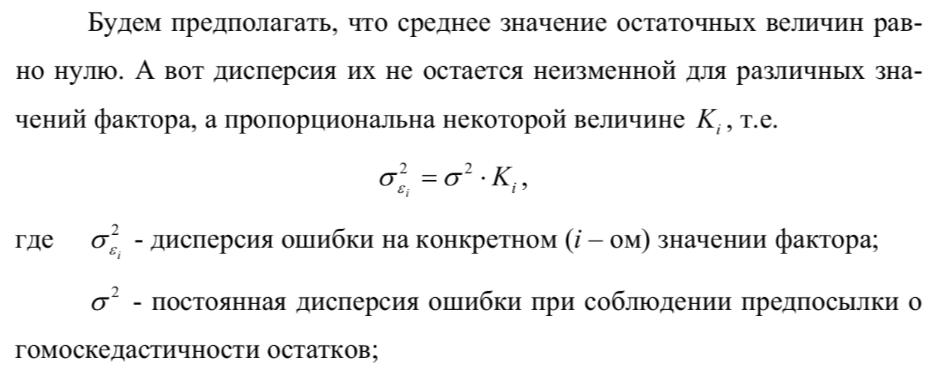
МНК предполагает выполнение следующих условий:

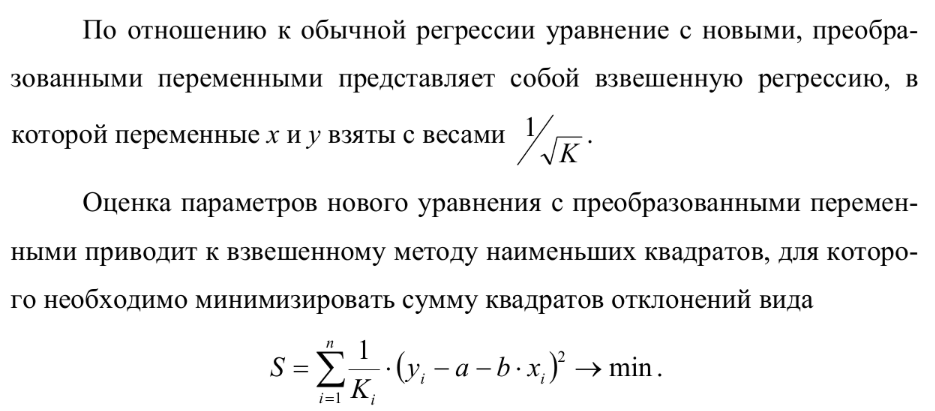
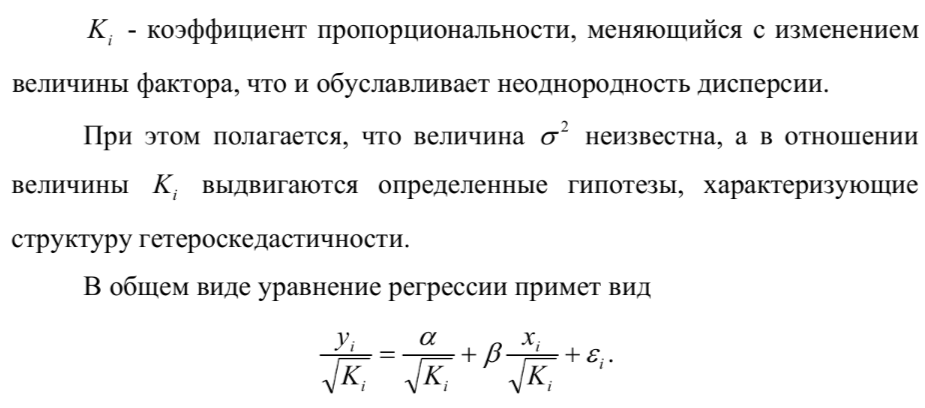
1. Случайные ошибки имеют нулевое математическое ожидание.
2. Ошибки имеют нормальное распределение.
3. Ошибки независимы (нет автокорреляции).
4. Дисперсия ошибок одинакова для всех наблюдений (гомоскедастичность).

Если эти предпосылки нарушаются, результаты метода могут быть ненадёжными. Например, при наличии мультиколлинеарности между признаками или гетероскедастичности ошибки метода возрастают. В таких случаях используют модификации, такие как регуляризация (Lasso, Ridge).

1. **Обобщенный метод наименьших квадратов**

Обобщенный метод наименьших квадратов применяется тогда, когда традиционный метод наименьших квадратов не очень точен (не выполняется условие гомоскедастичности или есть автокорреляция ошибок).





1. **Многоклассовая классификация**

Многоклассовая классификация — это задача машинного обучения, в которой модель должна классифицировать объекты (например, изображения, тексты или другие данные) на более чем два класса. В отличие от бинарной классификации, где есть только два возможных класса (например, "да" или "нет"), в многоклассовой классификации количество классов может быть любым, начиная с трех и более.

Для решения задач многоклассовой классификации используются различные алгоритмы, такие как логистическая регрессия, деревья решений, случайные леса, нейронные сети и другие.

Для решения задач многоклассовой классификации существует очевидный подход – это взять какой-либо алгоритм бинарной классификации и расширить его на M классов.

Существует два основных подхода к созданию моделей многоклассовой классификации на основе методов бинарной классификации:

* One-vs-All (один против всех): для каждого класса создается отдельный бинарный классификатор, который обучается различать один класс от всех остальных. Однако использование данного подхода обладает некоторыми недостатками. Один из них заключается в том, что может потребоваться больше вычислительных ресурсов, так как количество классификаторов пропорционально количеству классов.
* All-vs-All (все против всех): создается бинарный классификатор для каждой пары классов. Данный подход может быть более точным, чем One-vs-Rest, особенно в случаях, когда классы имеют схожие характеристики, так как каждый классификатор обучается на конкретной паре классов. Также при создании моделей учитывается взаимодействие между классами. Однако, такой подход требует значительно больше вычислительных ресурсов, так как количество классификаторов растет квадратично с увеличением числа классов.

1. **Метрики классификации**

1**. Accuracy**

Точность — это доля правильно классифицированных объектов от общего числа объектов. Она вычисляется по формуле:



2. **Recall**

Полнота (или чувствительность) измеряет, насколько хорошо модель находит положительные примеры для каждого класса. Для класса k она вычисляется как:

Изображение выглядит как текст, Шрифт, белый, дизайн

Автоматически созданное описание

3. **Precision**

Точность измеряет, насколько правильно модель предсказывает положительные примеры для каждого класса. Для класса k она вычисляется как:

Изображение выглядит как текст, Шрифт, белый, дизайн

Автоматически созданное описание

1. **Метрики регрессии**

Метрики регрессии позволяют оценить, насколько точно модель предсказывает значения зависимой переменной. Основные метрики включают MAE, MSE, RMSE и R2, каждая из которых имеет свои особенности и области применения.

**Mean Absolute Error (MAE).** Измеряет среднюю абсолютную ошибку между предсказанными и реальными значениями. Показывает, насколько в среднем предсказания модели отклоняются от истинных значений.

**Mean Squared Error (MSE).** Измеряет среднеквадратичную ошибку. Она более чувствительна к большим ошибкам, чем MAE, так как квадратичная функция ошибки увеличивает влияние больших отклонений. MSE полезна для задач, где важно минимизировать большие ошибки.

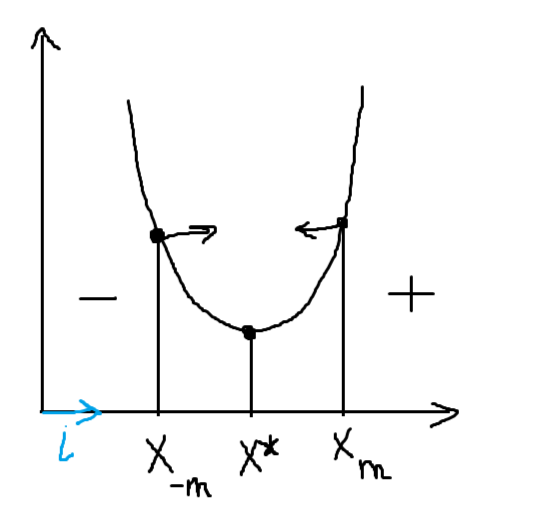
**Root Mean Squared Error (RMSE).** RMSE — это квадратный корень из MSE. Она также чувствительна к большим ошибкам, но выражена в тех же единицах, что и исходные данные. RMSE часто используется в задачах, где важно учитывать как средние, так и большие ошибки.

**R-squared (R²).** Измеряет долю дисперсии, объясненную моделью. Значение ближе к 1 указывает на хорошую модель. R² показывает, насколько хорошо модель объясняет вариативность данных, и является важной метрикой для оценки общей производительности модели.

1. **Метод стохастического градиентного спуска**

Стохастический градиентный спуск (англ. Stochastic gradient descent, SGD) — итерационный метод для оптимизации функции потерь. Отличается от обычного градиентного спуска тем, что градиент оптимизируемой функции считается на каждом шаге не как сумма градиентов от каждого элемента выборки, а как градиент от одного, случайно выбранного элемента.

Градиент – это вектор, который направлен в сторону наиболее резкого возрастания функции (указывающий в сторону роста функции). Чтобы найти минимум, нужен антиградиент – тот же вектор, но с минусом.

***Метод градиентного спуска***

***wnew = wпред – ню\*L’w***

***ню*** – шаг, скорость, с которой точки будут прыгать по кривой, пока не допрыгают до x\* = min. Чем больше ню, тем быстрее обучение (движение по кривой), но это не очень эффективно, так как можно не попасть в нужную точку, поэтому ню всегда маленькое.

Это используется во всех обучающих алгоритмах. Оптимальный метод для поиска минимума функции многих переменных.

***Суть:*** берем произвольную точку на кривой, от него считаем новое значение. Смотрим, минимум это или нет. Если нет, считаем следующее значение. Так до тех пор, пока не найдем минимум.

1. **Меры расстояний**

Метрики расстояния позволяют определить близость или расстояние между точками данных и понять, можно ли их сгруппировать.

Лучше всего было бы создать кластеры с помощью алгоритма кластеризации, такого как кластеризация методом k-средних или алгоритм k-ближайших соседей (knn), который использует ближайших соседей для решения задач классификации или регрессии. Как вы определите сходство между разными наблюдениями? Как мы можем сказать, что две точки похожи друг на друга? Это произойдёт, если их характеристики похожи, верно? Когда мы построим график этих точек, они будут ближе друг к другу по расстоянию.

Меры расстояния:

1) Евклидово расстояние — это кратчайшее расстояние между двумя векторами. Оно представляет собой квадратный корень из суммы квадратов разностей между соответствующими элементами. Большинство алгоритмов машинного обучения, в том числе K-Means, используют эту метрику расстояния для измерения сходства между наблюдениями.

2) Манхэттенское расстояние: расстояние между двумя точками равно сумме модулей разностей их координат.

3) Расстояние Минковского – это обобщенная форма евклидова и Манхэттенского расстояний.

4) Расстояние Хэмминга: измеряет сходство между двумя строками одинаковой длины. Это количество позиций, в которых соответствующие символы отличаются.

1. **Метод ближайшего соседа в методах классификации и регрессии**

К-ближайших соседей (K-Nearest Neighbors или просто KNN) — алгоритм классификации и регрессии, основанный на гипотезе компактности, которая предполагает, что расположенные близко друг к другу объекты в пространстве признаков имеют схожие значения целевой переменной или принадлежат к одному классу.

Алгоритм строится следующим образом:

1) сначала вычисляется расстояние между тестовым и всеми обучающими образцами;

2) далее из них выбирается k-ближайших образцов (соседей), где число k задаётся заранее;

3) итоговым прогнозом среди выбранных k-ближайших образцов будет мода в случае классификации и среднее арифметическое в случае регрессии;

4) предыдущие шаги повторяются для всех тестовых образцов.

