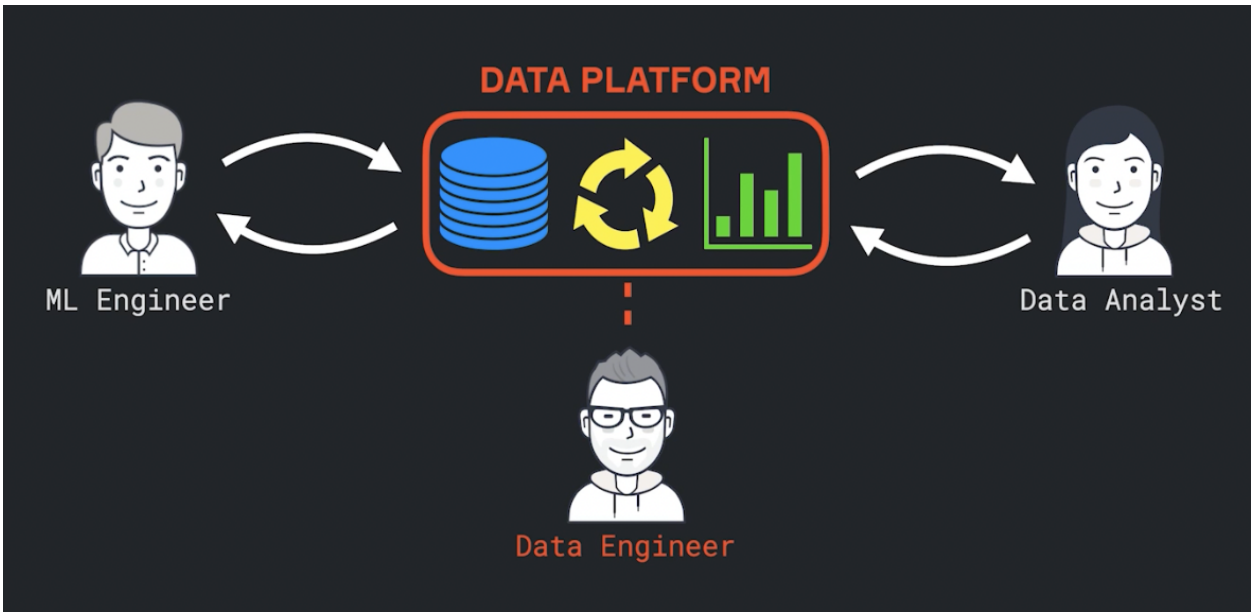
KARPOV.COURSES >>>
КОНСПЕКТ

> Конспект > 1 урок > Введение в машинное обучение

> Оглавление

> BIG DATA & ML

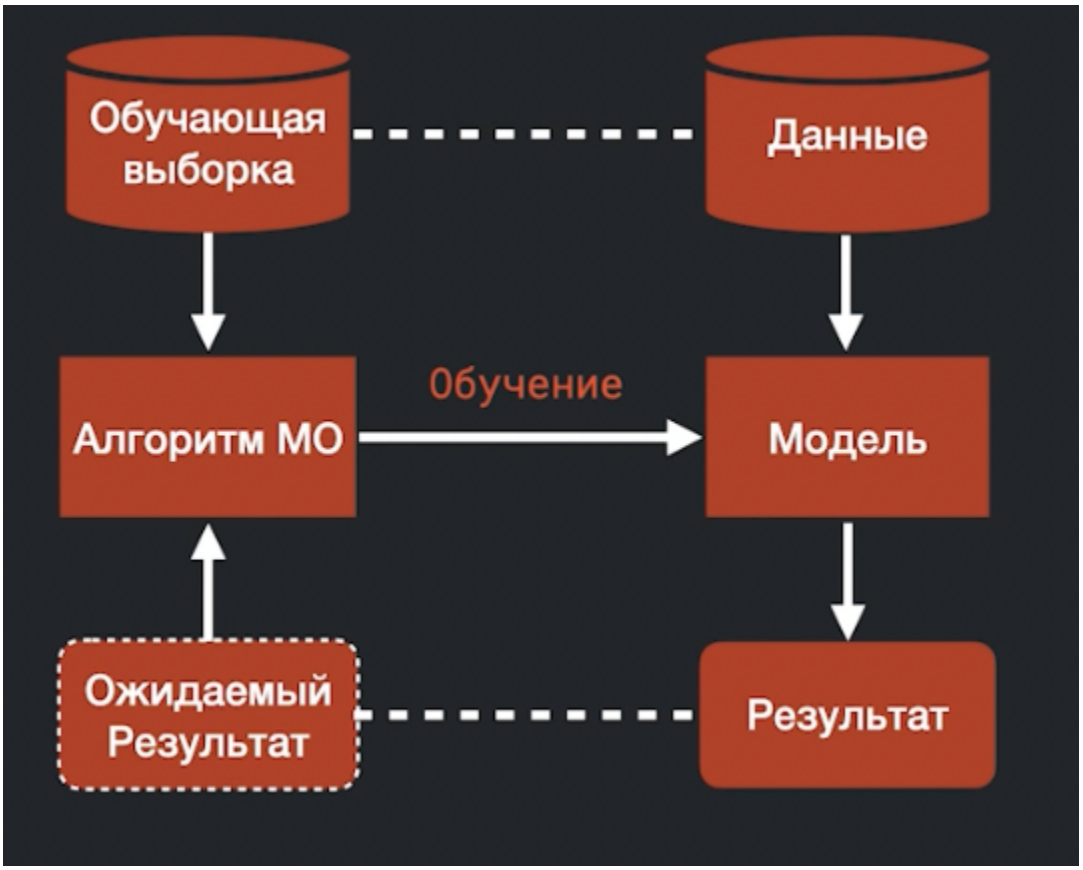
Зона ответственности дата инженера



Дата инженеры занимаются построением Data Platform, то есть полного набора инструментов, позволяющего работать с данными (хранить, использовать и обрабатывать). Аналитики и ML инженеры пользуются результатами работы инженера данных.

Алгоритм

При разработке некоторого алгоритма мы хотим, используя данные, передаваемые в него, получить какой-либо результат. Алгоритмы машинного обучения разрабатываются на порядок сложнее. На первый взгляд мы имеем схожих подход: есть данные и результат. Из данных формируется обучающая выборка, передаваемая в алгоритм МО. В зависимости от используемого алгоритма, мы можем иметь (или не иметь) ожидаемый результат. Далее обученная модель применяется к данным и формирует результат.



Алгоритм vs машинное обучение

	Алгоритм	Машинное обучение
Результат	Точный	Вероятностный
Проверка	Тестирование	Оценка качества (метрики)
Сложность	Относительно просто реализовать	Требует затрат как для подготовки данных, так и в обучении
Интерпретируемость	Прозрачен	Большинство методов трудно объяснимы
Применение	Широкое	Более широкое

> Составляющие машинного обучения

Данные - множество объектов (ситуаций) и их свойств для решения задачи.

Признаки - свойства или характеристики, используемые для обучения.

Алгоритм - метод, применяемый для создания модели.

Метрика - функция для оценки качества модели.

Основные проблемы в ML

Данные

- Где взять?
- Как хранить?
- Чем больше, тем лучше! (не всегда, зависит от качества данных)

Вычислительные мощности

- Где взять CPU/GPU?

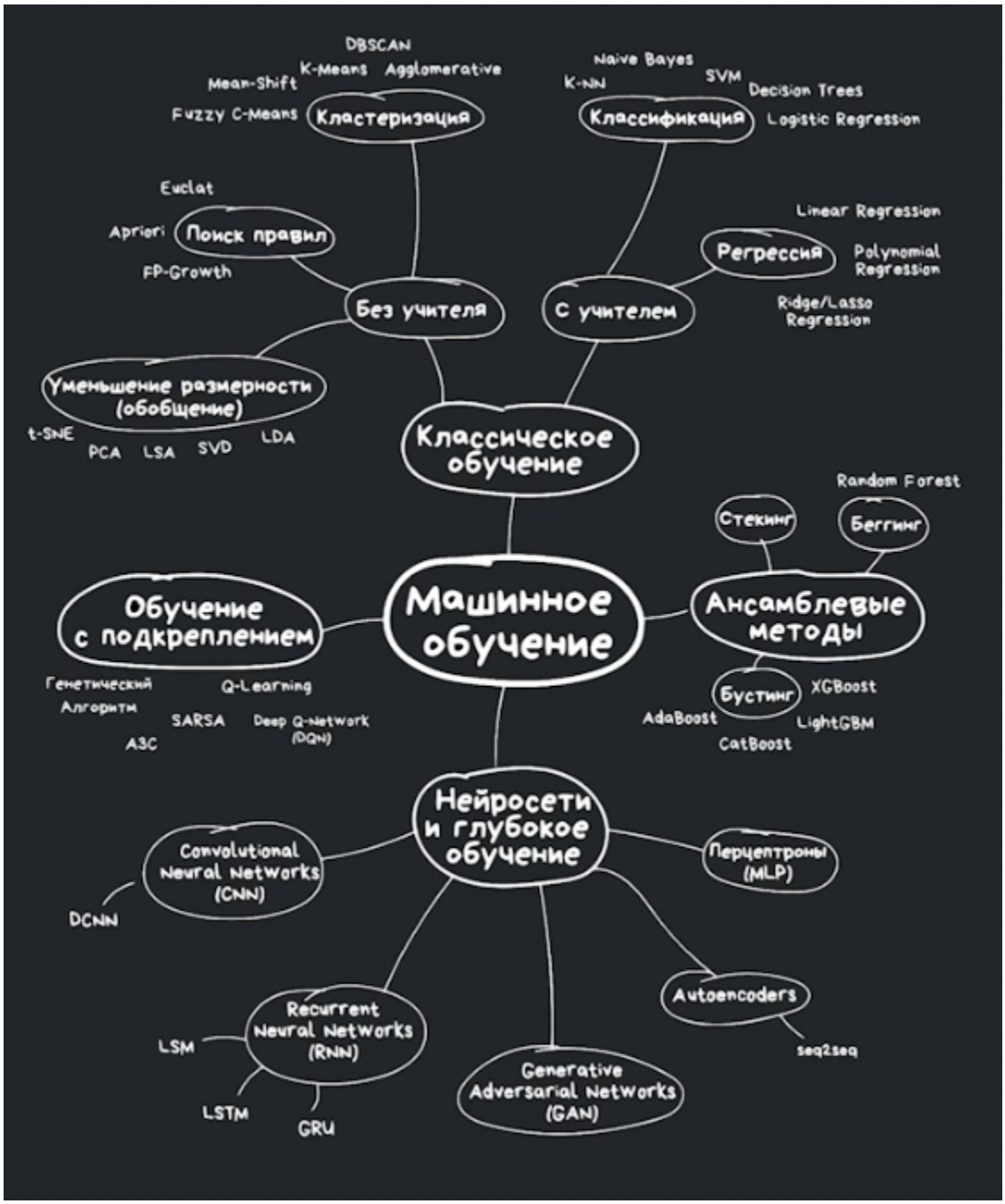
Результат по данным

- Ручная разметка
- Сбор результатов внутри системы
- Готовые датасеты

Производительность (Near Real Time)

- Оптимизация
- Масштабирование

Методы МО



В машинном обучении присутствует широкий набор методов, которые можно использовать для решения задач.

Разделяют 4 основных подхода:

- нейросети и глубокое обучение - работа с неструктурированными данными (изображения, звук, видео)

- ансамблевые методы - используя методы классического МО, позволяет повышать качество модели путем их комбинирования
- классическое обучение - подходит для решения простых задач МО
- обучение с подкреплением - подходит для интеграции с системой, на которую влияет модель МО и получать данные от этой системы для улучшения модели

> Классическое обучение

Обучение с учителем:

- Классификация - предсказать класс объекта
- Регрессия - предсказать значение

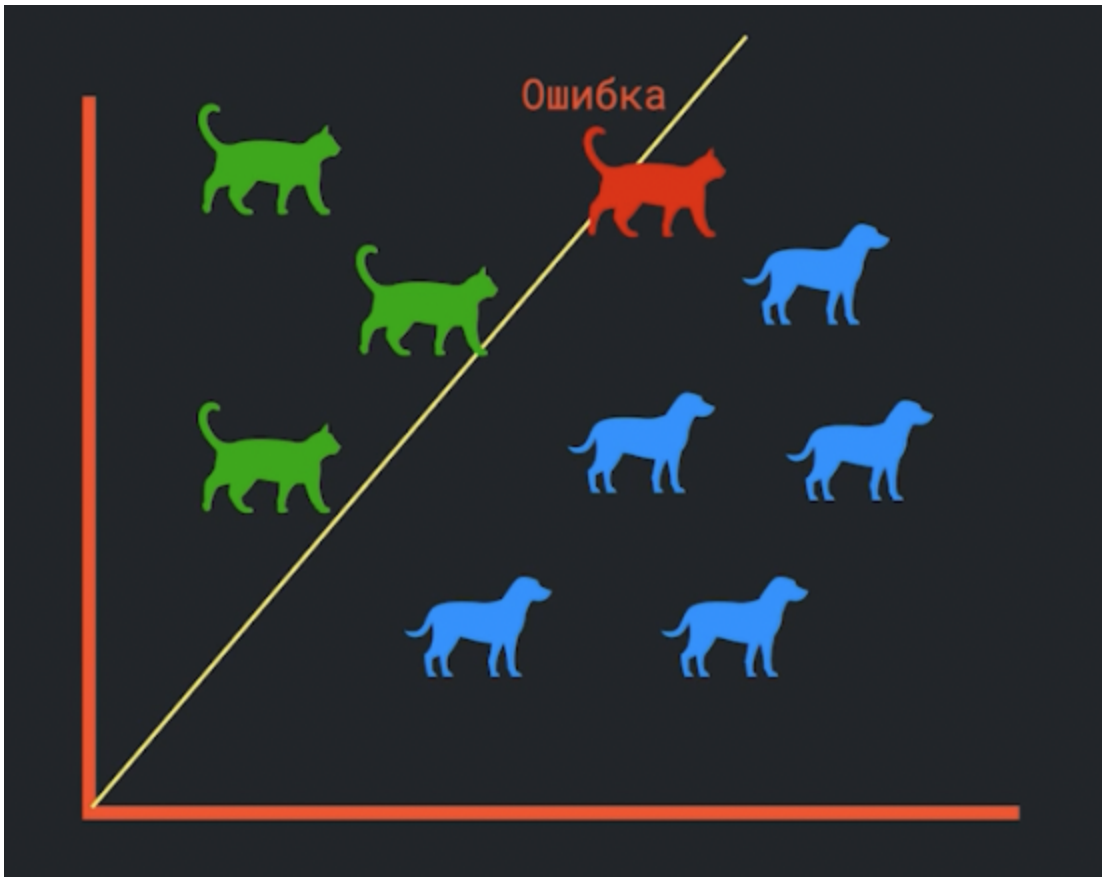
Обучение без учителя:

- Кластеризация - группировка объектов по сложности
- Ассоциация - выявление последовательностей
- Уменьшение размерности - выявление зависимостей

При обучении с учителем модель учится на данных и мы знаем, какой результат должны получить на этих данных. При обучении без учителя данные также есть, но мы не знаем, какой результат в итоге должен получиться на них.

Задача классификации

Задача классификации - получение категориального ответа на основе набора признаков.



Пример бинарной классификации

Пример такой задачи - классификация кошек и собак (бинарная классификация). По какому-либо набору параметров (вес, длина ушей, форма носа и т.д.) модели нужно определить, кто относится к коту, а кто к собаке. Эти параметры будут являться признаками, а класс будет являться целевой переменной.

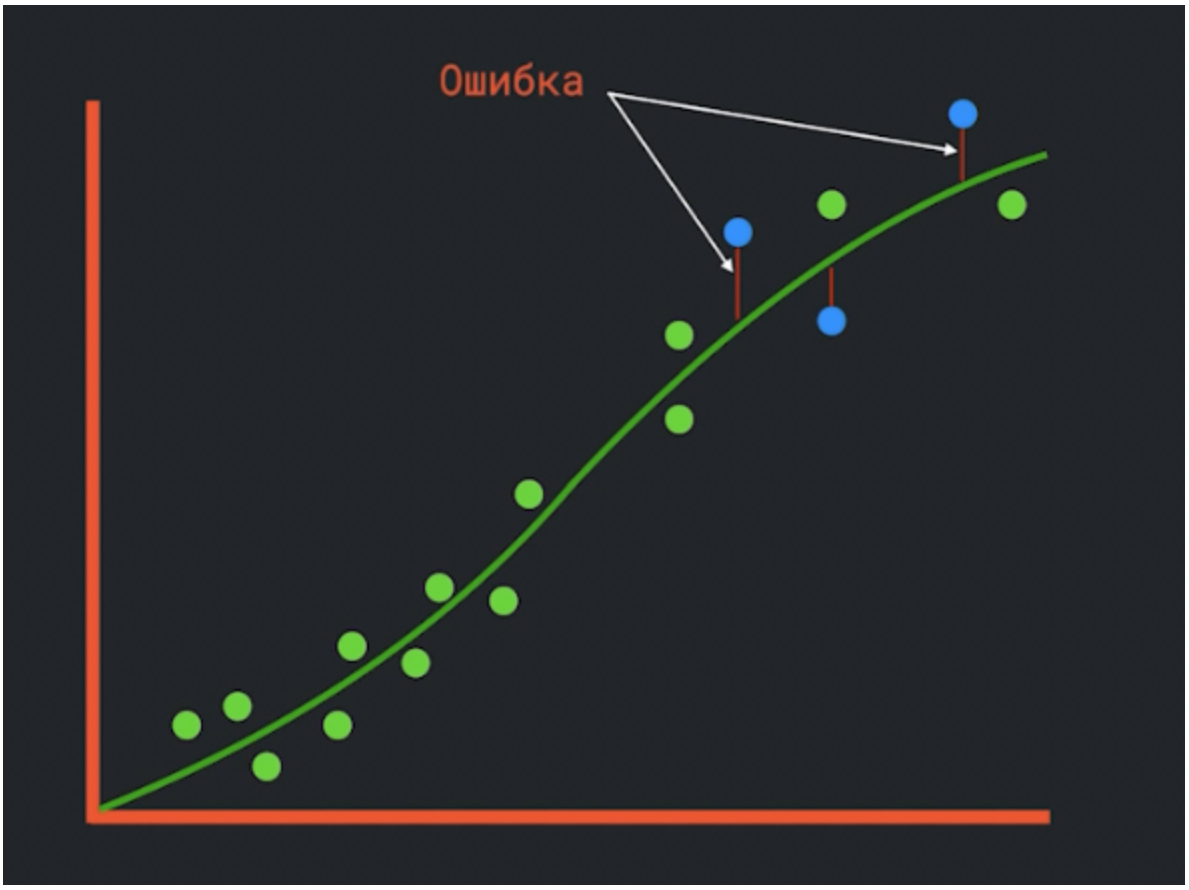
Метрики для определения качества модели в задачах классификации:

- Accuracy - доля правильных ответов в наборе данных
- F-мера - гармоническое среднее между точностью и полнотой

Данные метрики оценивают насколько модель точна в предсказании.

Задача регрессии

Задача регрессии - прогноз значения на основе выборки объектов с различными признаками.



Пример регрессии

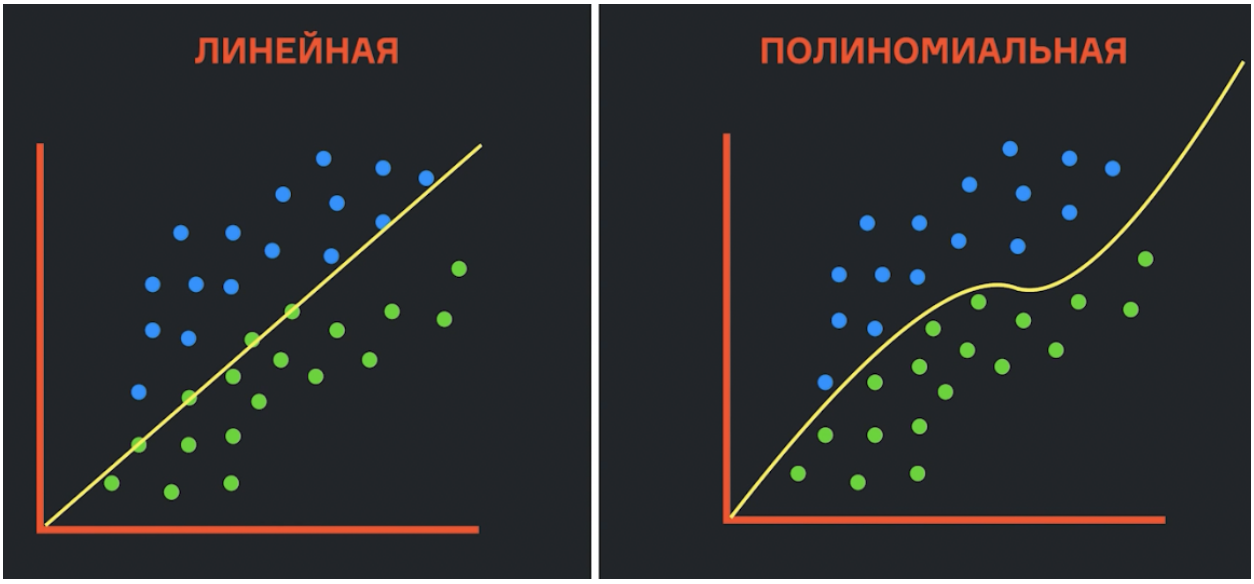
Пример задачи регрессии - прогнозирование цены. Здесь также имеются параметры, от которых зависима цена.

Метрики для определения качества в задачах регрессии:

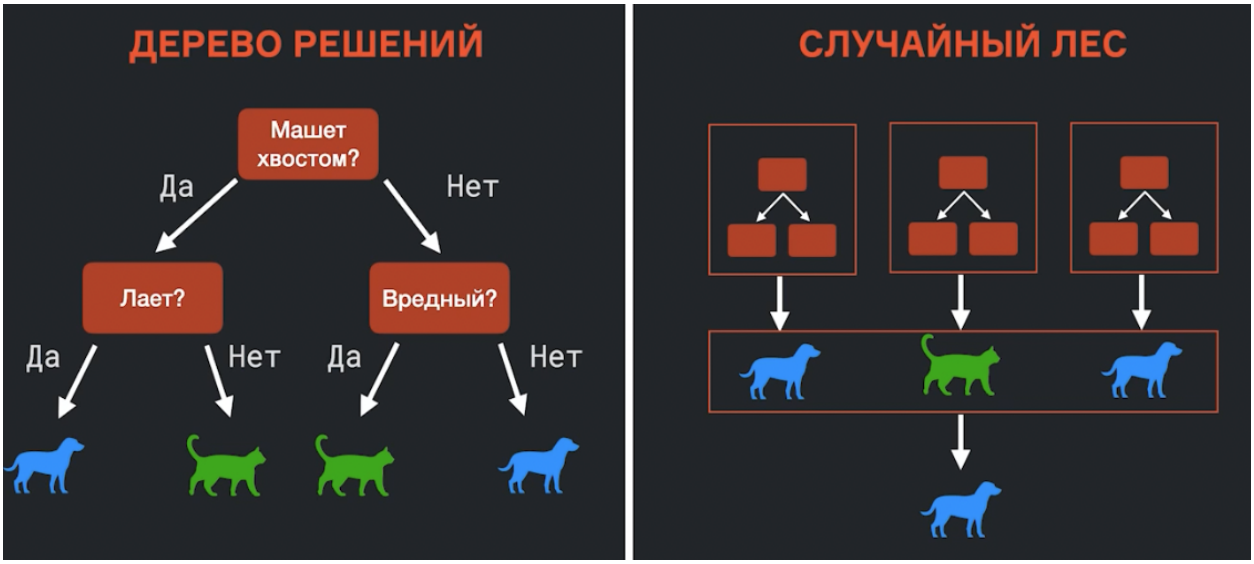
- MSE - средний квадрат отклонения
- RMSE - корень из среднего квадрата отклонения

Данные метрики оценивают насколько модель ошиблась в предсказании.

Регрессия может быть линейной или полиномиальной.



Дерево решений и случайный лес

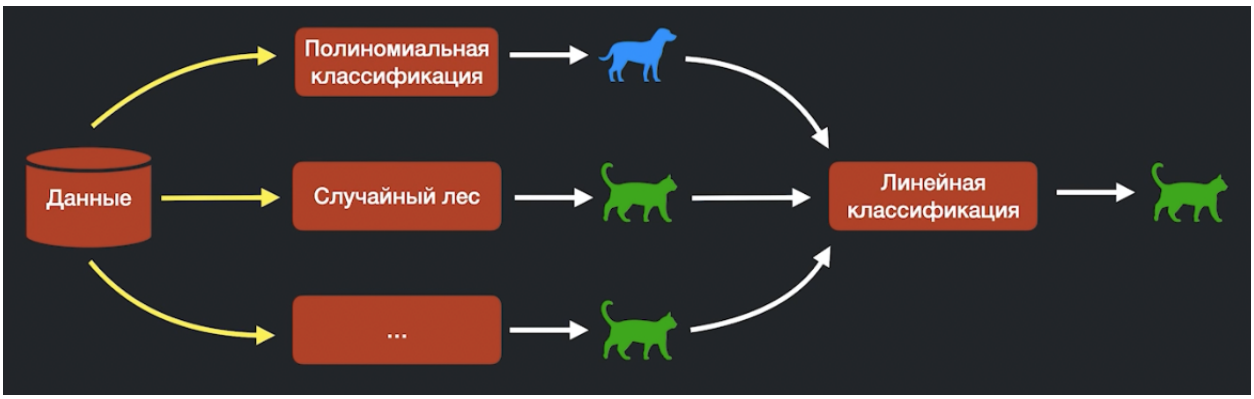


В вершине дерева задается вопрос к объекту, на основе ответа которого мы направляемся к следующему вопросу. Случайный лес стоит из комбинации деревьев, где каждое дерево выносит свой вердикт по объекту, но может иметь разные результаты. На основе этих результатов выносится общий вердикт.

> Ансамбли

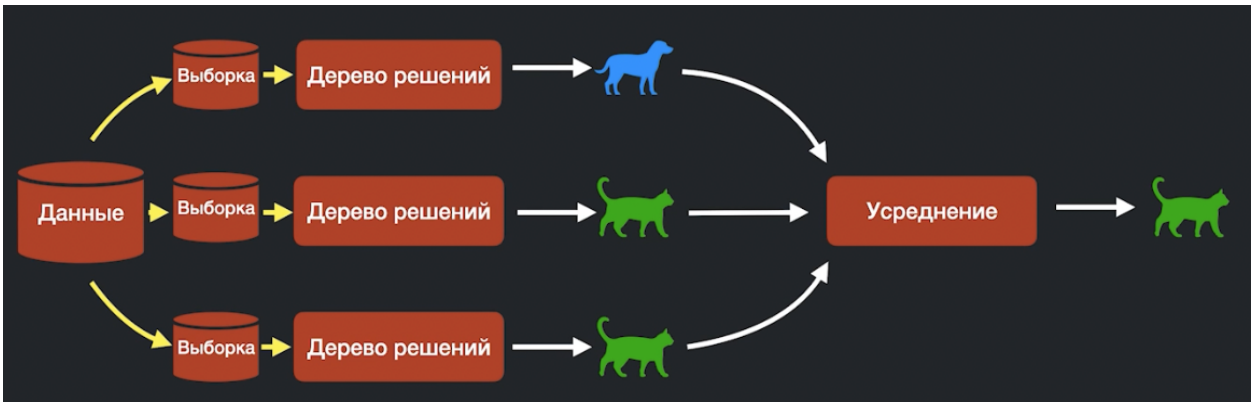
Стекинг

Стекинг - обучение набора разных алгоритмов и передача их результатов на вход последнему, который и принимает итоговое решение.



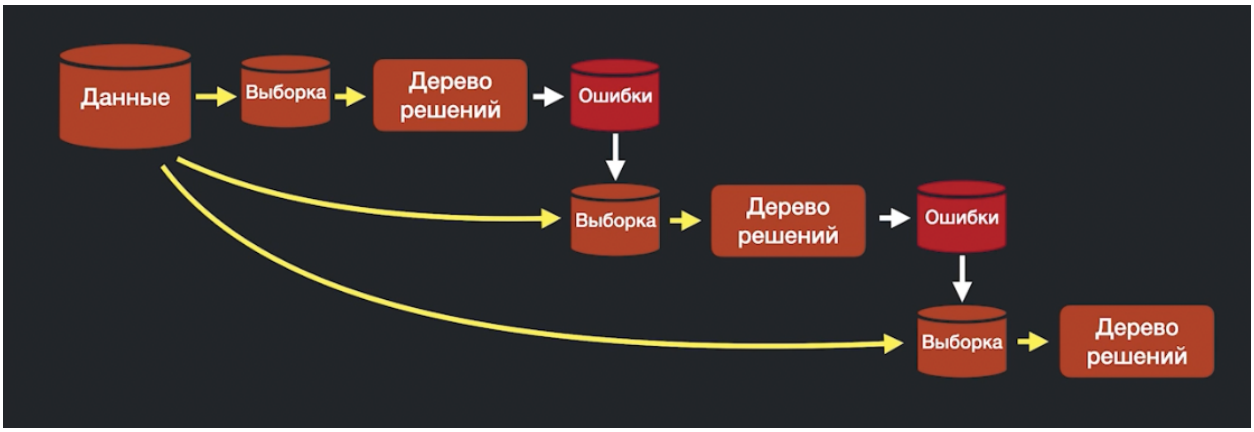
Бэггинг

Бэггинг - обучается один алгоритм много раз на случайных выборках из исходных данных, после чего ответы усредняются.



Бустинг

Бустинг - алгоритм обучается последовательно, каждый следующий уделяет особое внимание тем случаям, на которых ошибся предыдущий.



> Виды параметров модели и процесс обучения

Параметры модели - параметры, которые изменяются и оптимизируются в процессе обучения модели и итоговые значениях этих параметров являются результатом обучения модели.

Например, при обучении линейной регрессии, мы имеем уравнение прямой, где нам необходимо определить коэффициенты. В данном случае - эти коэффициенты будут являться параметрами модели. В случае нейронной сети к параметрам можно отнести веса нейронов.

Гиперпараметры модели - параметры, значения которых задаются до начала обучения модели и не изменяется в процессе обучения. При этом у модели может и не быть гиперпараметров.

На примере нейронной сети к гиперпараметрам можно отнести количество слоев в ней. В случае дерева решений гиперпараметром может являться глубина дерева или максимальное количество ветвей.

Оптимизация гиперпараметров - процесс поиска набора оптимальных гиперпараметров для алгоритма обучения.

Алгоритмы оптимизации гиперпараметров

Grid Search - алгоритм поиска гиперпараметров на основе перебора комбинаций гиперпараметров из заданного множества.

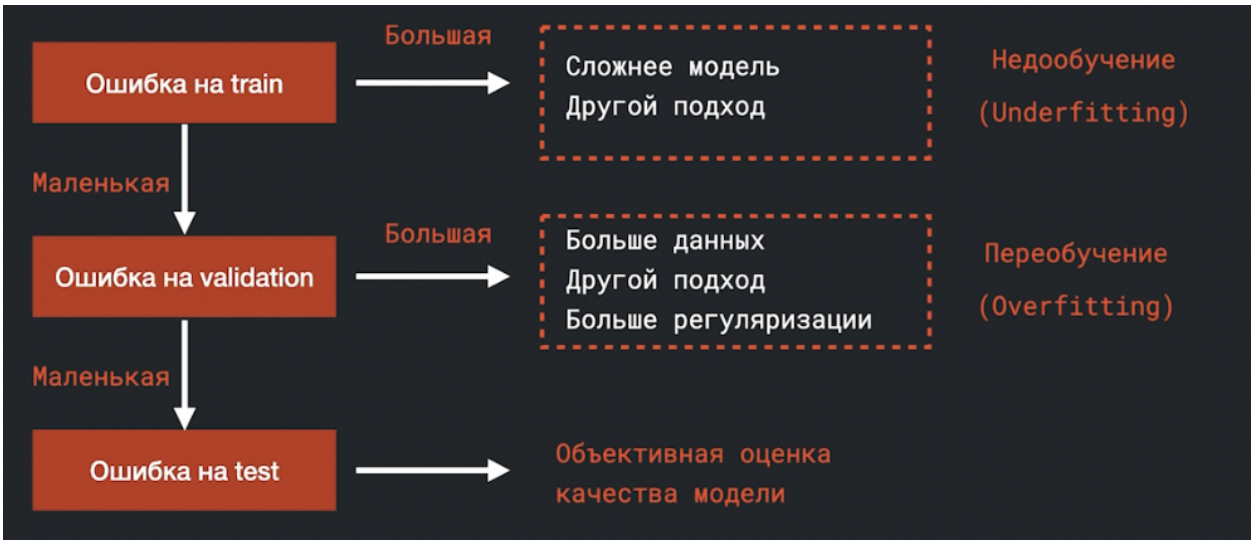
Random Search - алгоритм поиска гиперпараметров на овнове перебора случайно выбранных комбинаций гиперпараметров из заданного диапазона.

Gradient-based - алгоритмы поиска гиперпараметров на основе градиентного спуска.

Процесс обучения

При обучении модели у нас есть dataset (набор данных). Использовать весь dataset для обучения модели не является хорошей идеей, поэтому он делится на три выборки:

- train - применяется для обучения модели.
- validation - применяется для оптимизации параметров модели.
- test - применяется для итоговой оценки качества модели.



Underfitting и overfitting

Представим, что у нас есть dataset для обучения модели, который мы разбили на train, validation и test выборки.

Если на train наша модель выдает большую ошибку, то это говорит о том, что нам нужна более сложная модель или другой подход в обучении. Модель недообучивается (underfitting).

Если на train ошибка маленькая, то мы смотрим на validation. Большая ошибка на validation свидетельствует о переобучении (overfitting). Это значит, что модель запоминает, как ей нужно отвечать в той ситуации, в которой она находится. И с новыми данными она может показать обратную реакцию и сильно ошибаться. Здесь может быть полезно добавить больше данных, изменить подход или добавить больше регуляризации.

Если на validation ошибка маленькая, то можно применять модель на test данных для составления объективной оценки качества модели.