Машинное обучение

Входной ассессмент для программ

«Искусственный интеллект и предиктивная аналитика»

«Искусственный интеллект и бизнес-аналитика в реальном секторе экономики»

Лекция 3

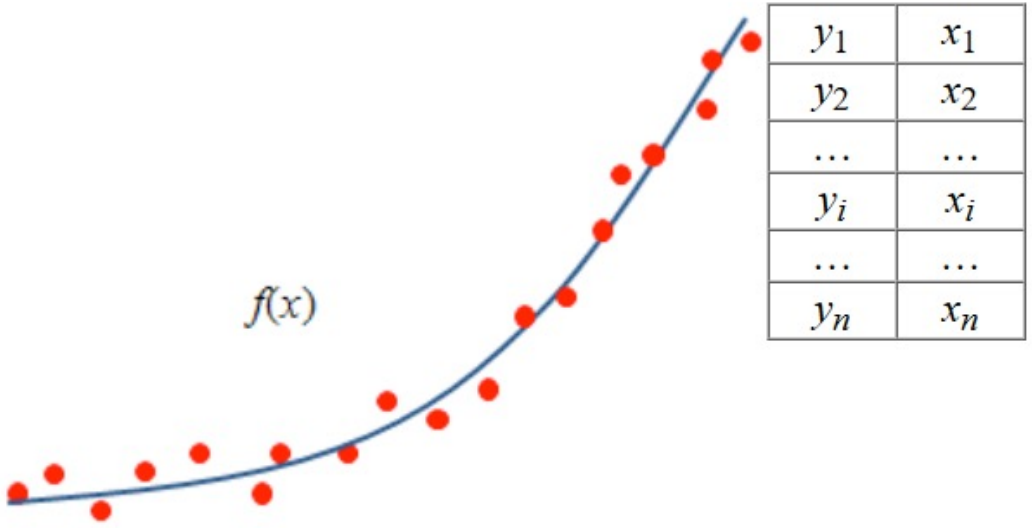
Машинное обучение

Машинное обучение – процесс решения некоторой задачи путем построения *статистической модели* на основе анализа некоторого *набора данных*.

Набор данных должен иметь минимально подходящий для решения задачи объем и содержать наблюдения о некотором объекте (сущности).

Наблюдения представляют собой некоторое множество значений показателей (атрибутов) объекта.

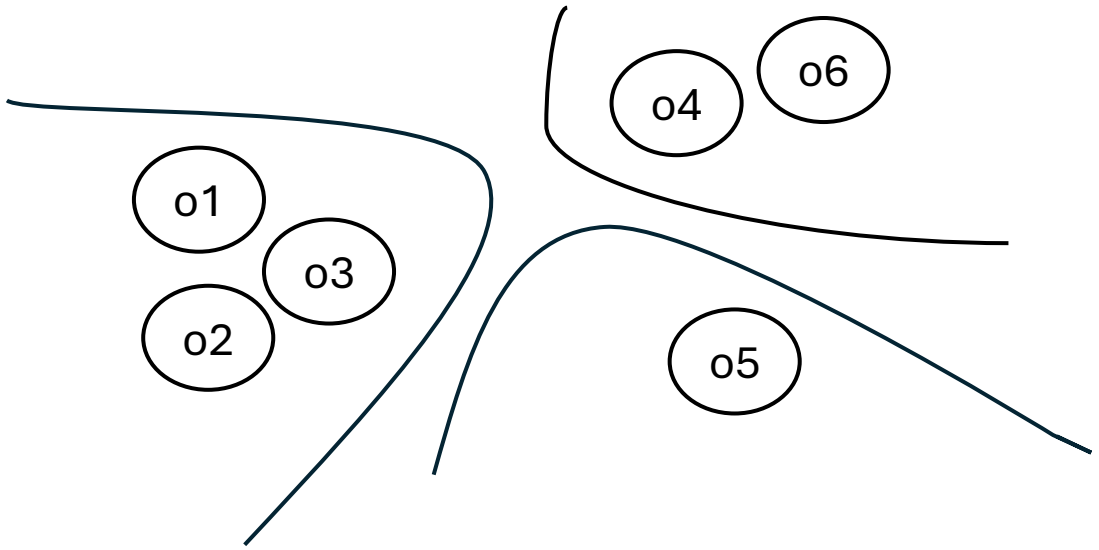
Аппроксимировать — это заменить одни результаты другими, близкими к исходным, но упрощенными.



y_1	x_1
y_2	x_2
…	…
y_i	x_i
…	…
y_n	x_n

Типы машинного обучения

1. *Обучение с учителем*. Обучить модель повторять поведение объекта.
2. *Обучение без учителя*. Обучить модель группировать объекты по степени похожести.
3. *Обучение с подкреплением*. Обучить модель действовать в некоторой неизвестной среде путем выполнения множества попыток и получения штрафов и вознаграждений.



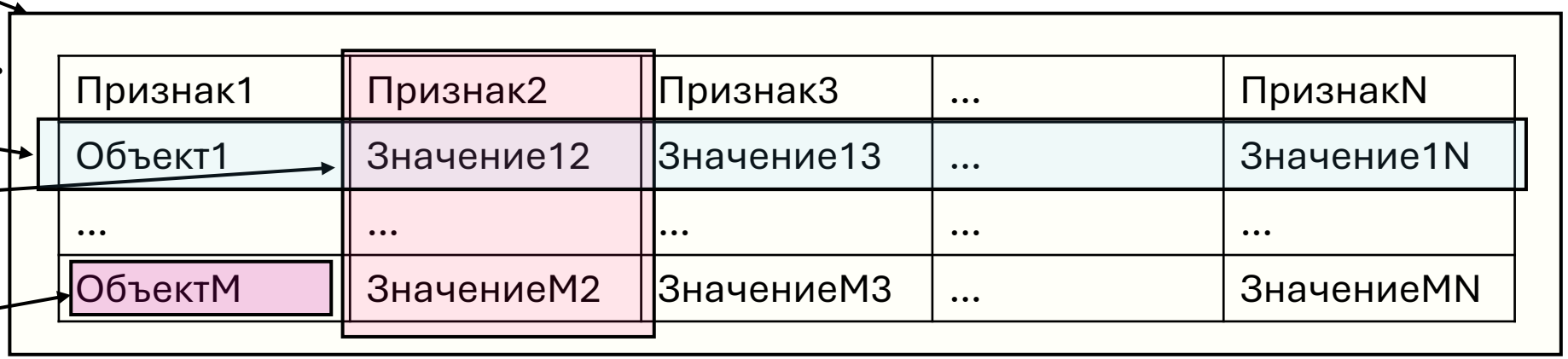
Данные в машинном обучении – 1

Набор данных – множество векторов X (таблица с данными об объекте/объектах).

Строка – наблюдение.

Столбец – признак.

Ячейка – значение признака наблюдения.



Признак1	Признак2	Признак3	…	ПризнакN
Объект1	Значение12	Значение13	…	Значение1N
…	…	…	…	…
ОбъектM	ЗначениеM2	ЗначениеM3	…	ЗначениеMN

Вектор X (строка) – входной вектор признаков (наблюдение, значения атрибутов объекта).

Признак (столбец/ячейка) – атрибут объекта (значение некоторого свойства объекта).

Данные в машинном обучении – 2

Неструктурированные (сырые, первичные) данные – данные для анализа в из «родном» формате: текст, изображения, звук и т. д. (jpeg, doc, docx, pdf, mp3, wav, …).

Слабоструктурированные (полуструктурированные) данные – текстовые данные с определенной структурой: логи, языки разметки, конфигурационные файлы и т. д. (xml, json, html, yaml, …).

Структурированные (аккуратные) данные – сформированный набор данных для анализа (таблица с набором наблюдений и признаков).

Для получения структурированных данных необходимо:

1. Выполнить конструирование признаков.
2. Структурировать данные.
3. Предобработать данные.

Если данные уже были структурированы, то конструирование признаков можно выполнять после предобработки данных.

Данные в машинном обучении – 3

При решении различных задач наблюдение может быть представлено как:

1. Вектор.
2. Матрица (многомерный вектор).
3. Тензор (многомерная матрица).

Для большинства методов машинного обучения структурированные данные следует представить в виде множества векторов числовых признаков.

При решении задачи обучения с учителем целевой признак Y выбирается из доступных признаков набора данных.

Данные в машинном обучении – 4

Для решения задачи машинного обучения следует делить набор данных на выборки:

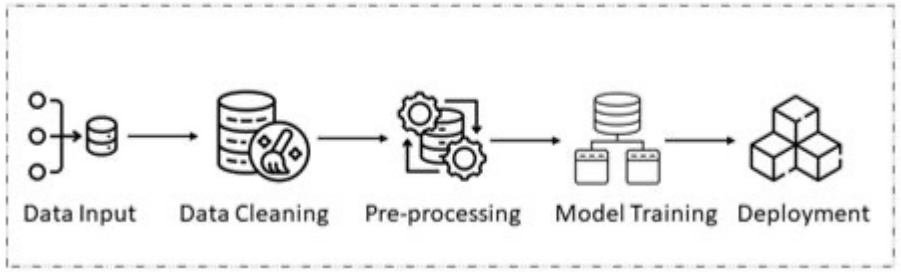
1. Обучающая выборка (70-80%). Обучение модели (подбор коэффициентов некоторой математической функции для аппроксимации).
2. *Контрольная выборка (10-15%)*. Выбор метода обучения, настройка гиперпараметров.
3. Тестовая выборка (10-15% или 20-30%). Оценка качества модели перед передачей заказчику.

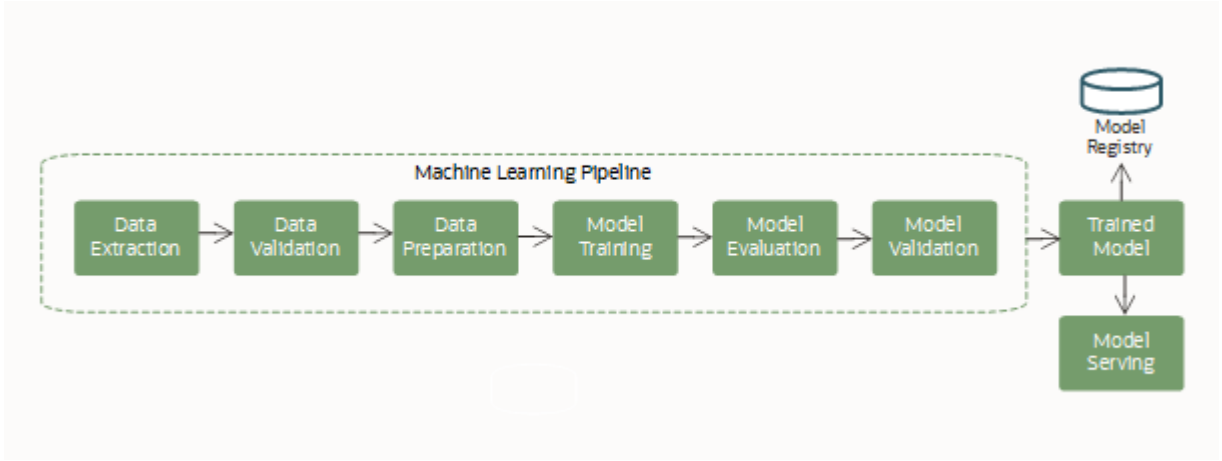
При обучении нельзя использовать данные из контрольной и тестовой выборок.

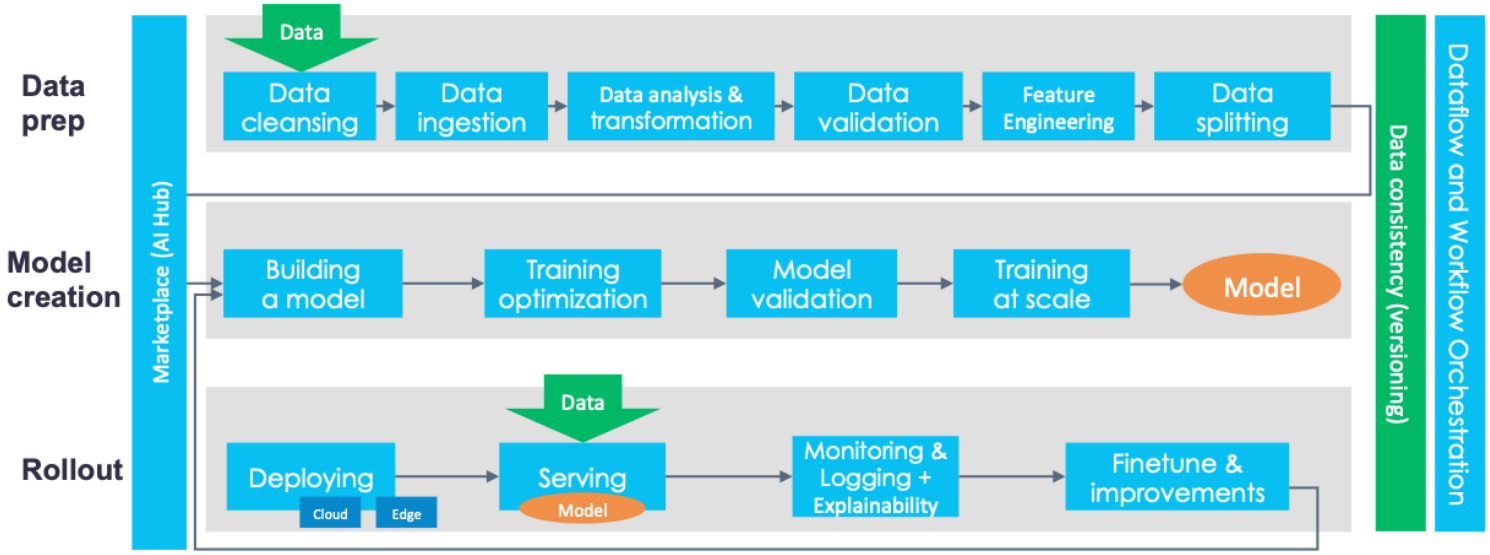
Иначе модель может «запомнить» все примеры и показывать превосходное качество на тестовых данных, но показывать посредственные результаты у заказчика на реальных данных.

Конвейер машинного обучения

Конвейер (pipeline) машинного обучения – набор типовых шагов для решения задачи машинного обучения: от получения данных до внедрения модели у заказчика.







Параметры и гиперпараметры

Гиперпараметры – входные данные для конвейера машинного обучения, которые влияют на качество полученной модели.

Гиперпараметры не входят в набор данных, но включают различные параметры самого метода машинного обучения, а также параметры и типы методов предобработки и подготовки набора данных.

Параметры – параметры (коэффициенты) модели, которые «настраиваются» в процессе обучения.

Например, для линейной регрессии:

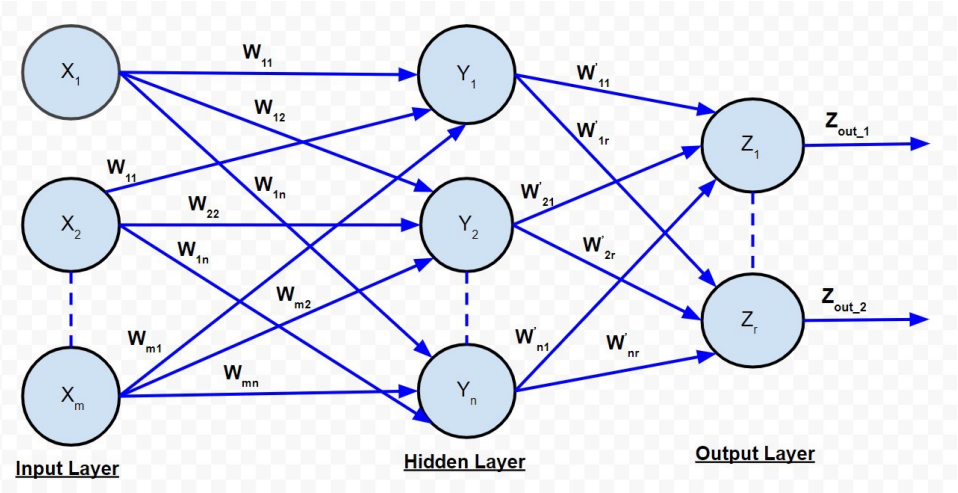
$$Y = \mathbf{w} * X + \mathbf{b}$$

Машинное обучение и глубокое обучение

Методы глубокого обучения входят в методы машинного обучения.

Однако методы глубокого обучения используют промежуточные слои, модели которых обучаются не на исходном наборе данных, а на выходных данных других слоев.

К методам глубокого обучения относятся различные архитектуры искусственных нейронных сетей.



Использование методов машинного обучения – 1

Не все задачи выгодно решать с помощью методов машинного обучения.

Когда следует использовать методы машинного обучения?

1. Задача слишком сложна для описания в виде традиционного алгоритма.
   Много условий и правил или не ясно как устроен объект.
2. Условия задачи постоянно меняются.
   Формат входных данных и/или требования к системе постоянно меняются.
3. Задача связана с распознаванием изображений или звуков.
4. Задача **НЕ** может быть решена традиционным способом.
5. Решение задачи экономически выгодно.

Использование методов машинного обучения – 2

Не все задачи выгодно решать с помощью методов машинного обучения.

Когда **НЕ** следует использовать методы машинного обучения?

1. Требуется объяснить как и почему система получила некоторый результат.
2. Высокая стоимость ошибки системы.
3. Ограничения по времени.
4. Можно использовать традиционные подходы к разработке с наименьшими затратами.
5. Объект слишком сложный, а данных для обучения недостаточно.

Инженерия машинного обучения

Инженерия машинного обучения (MLE, machine learning engineering) – совместное использование научных принципов, средств, технологий и методов *машинного обучения и программной инженерии* для проектирования и разработки интеллектуальных программных систем.

Поведение «обычной» программной системы детерминировано (определено, можно построить диаграмму состояний системы, блок-схемы алгоритмов).

Поведение «интеллектуальной» системы со временем может ухудшаться или становиться аномальным. Например, из-за изменений в данных или возникновения новых закономерностей, которых не существовало при обучении модели.

Жизненный цикл проекта машинного обучения

1. Понимание бизнес-цели.
   Что новая система даст бизнесу?
2. Понимание цели технического проекта.
   Что поступает на вход, что является целевым признаком, как оценить качество?
3. Сбор и подготовка данных.
4. Конструирование признаков.
5. Обучение модели.
6. Оценка качества модели.
7. Развертывание модели.
8. Выполнение модели.
9. Мониторинг модели.
10. Сопровождение модели.

Бизнес-цель и технические цели должны совпадать.

ЖЦ позволяет возврат к любому этапу.

Ключевые показатели проекта машинного обучения

1. Последствия внедрения проекта:
   - машинное обучение может заменить «сложную» часть проекта,
   - «дешевая» модель с невысоким качеством дает значительные преимущества.
2. Стоимость проекта:
   - сложность задачи (готовые реализации, вычислительные ресурсы, время и деньги),
   - стоимость данных (сбор, разметка),
   - необходимое качество модели (стоимость ошибки).

Высокие требования к качеству могут привести к использованию сложных моделей и/или необходимости больших объемов качественных данных, а также к повышению требований к вычислительным ресурсам.

Важные свойства модели машинного обучения

1. Учет бизнес-цели.
2. Учет особенностей проблемной области и данных.
3. Учет требований к качеству.
4. Экономическая выгода.
5. Полезность для пользователей.
6. Предсказуемость и воспроизводимость поведения.

Предсказуемость – получение аналогичного показателя качества на данных близких к обучающей выборке.

Воспроизводимость – получение аналогичной модели на основе тех же исходных данных: набор данных, алгоритм машинного обучения, гиперпараметры.

Члены команды

1. Аналитики, специалисты по анализу данных, data scientists (определяют цели, занимаются подготовкой данных, конструируют признаки, выбирают подходящие методы и гиперпараметры, пишут прототипы на Python).
2. *Эксперты (объясняют особенности проблемной области, помогают аналитику)*.
3. Инженеры по данным (занимаются получением сырых данных).
4. Разметчики (размечают набор данных, формируют целевой Y).
5. Программисты (переписывают код аналитика с Python на более производительный язык, оптимизируют код прототипов).
6. Специалисты DevOps/MLOps (настраивают среду, CI/CD).

Причины провалов проектов

С 2017 по 2020 годы от 74% до 87% проектов в области машинного обучения и анализа данных закончились провалом и не дошли до этапа эксплуатации.

Причины:

1. Организационные:
   - нехватка квалифицированных кадров (не ясно как нанимать, кандидаты прошли курсы и не имеют опыта коммерческих проектов, работали только с тестовыми данными в рамках обучения),
   - отсутствие поддержки со стороны руководства (аналитики работают в условиях неопределенности и тратят много времени на эксперименты, руководители ничего не понимают, руководители не видят прогресса, научные работники не могут презентовать результат),
   - разобщенность подразделений организации (данные и/или исполнители находятся в разных подразделениях),
   - невыполнимые проекты (завышенные ожидания и неадекватная оценка сложности проекта).
   - несогласованность бизнес-цели и цели технического проекта (аналитики не понимают бизнес-цели и решают научную задачу).
2. Технические:
   - отсутствует инфраструктура для работы с данными (данные собираются вручную или скриптами, сложно воспроизвести результаты),
   - проблемы разметки данных (аналитики сами размечают данные или используют аутсорсинг с низким качеством).