ЛЕКЦИЯ 5
ОСНОВНЫЕ МЕТОДЫ И ПОДХОДЫ МАШИННОГО ОБУЧЕНИЯ

Цель лекции:

Сформировать системное представление об основных подходах машинного обучения, показать различия между обучением с учителем, без учителя и с подкреплением, раскрыть назначение базовых методов и научить обоснованно выбирать тип задачи и алгоритм с учётом свойств данных и требований прикладной системы.

Понятие машинного обучения

Машинное обучение (Machine Learning, ML) — направление искусственного интеллекта, в котором модель формируется на основе данных и используется для прогнозирования, классификации или выявления структуры новых данных. В отличие от программы, где все правила поведения явно задаются разработчиком, в машинном обучении значительная часть закономерностей извлекается из обучающей выборки.

Важно различать алгоритм и модель. Алгоритм задаёт процедуру обучения, а модель является результатом применения этой процедуры к конкретным данным. Например, алгоритм дерева решений определяет, как выбирать признаки и строить разбиения, а конкретное обученное дерево с определёнными узлами и порогами — это модель.

Пример. В системе обнаружения подозрительных входов объектом является отдельная попытка входа. Признаками могут быть число неудачных попыток, время суток, страна подключения, тип устройства и длительность сессии. Целевой переменной может быть класс normal или suspicious.

Типовой цикл построения модели машинного обучения

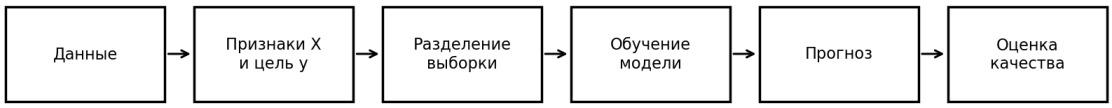


Рисунок 1 – Типовой цикл построения модели машинного обучения

Таблица 1. Основные компоненты задачи машинного обучения

Компонент	**Содержание**	**Пример для информационной безопасности**
Объект	Отдельное наблюдение, которое анализируется моделью	Сетевое соединение или попытка входа
Признак	Измеримая характеристика объекта	duration, packets, failed_logins
Целевая переменная	Значение, которое требуется предсказать	normal / attack
Обучающая выборка	Данные, на которых алгоритм подбирает параметры	Исторические размеченные события
Модель	Функция или структура, полученная после обучения	Обученное дерево решений
Метрика	Показатель качества результата	Accuracy, precision, recall или ошибка регрессии

Основные подходы машинного обучения

В учебном курсе выделяются три базовых подхода: обучение с учителем, обучение без учителя и обучение с подкреплением. Они различаются тем, какая информация доступна алгоритму в процессе обучения и какой результат требуется получить.



Рисунок 2 – Основные подходы машинного обучения

Таблица 2 – Основные подходы

Подход	**Какие данные используются**	**Типичная задача**	**Пример**
Обучение с учителем	Объекты и известные правильные ответы	Классификация, регрессия	Определить attack/normal; прогнозировать время обработки запроса
Обучение без учителя	Объекты без целевых меток	Кластеризация, выявление структуры	Сгруппировать сетевые узлы по профилю активности
Обучение с подкреплением	Взаимодействие агента со средой и сигнал награды	Выбор последовательности действий	Адаптивное управление ресурсами или стратегией реагирования

Обучение с учителем

При обучении с учителем каждый обучающий пример содержит входные признаки X и известный правильный ответ y. Алгоритм подбирает зависимость между X и y, а затем применяется к новым объектам. Две основные задачи этого подхода — классификация и регрессия.

1. Классификация

Классификация используется, когда целевая переменная принимает дискретные значения. При двух классах задача называется бинарной, при большем числе классов — многоклассовой.

Тип классификации	**Пример входных данных**	**Результат**

Бинарная	Характеристики электронного письма	phishing / legitimate
Бинарная	Параметры сетевого соединения	attack / normal
Многоклассовая	Признаки события безопасности	normal / DoS / brute force / malware
Многоклассовая	Характеристики документа	тип документа A / B / C

Пример. Если модель оценивает вероятность атаки как 0,92, система может отнести событие к классу attack. Однако сам порог принятия решения зависит от прикладной задачи и цены ошибок.

2. Регрессия

Регрессия применяется, когда необходимо прогнозировать числовое значение. В загруженном учебном пособии линейная регрессия рассматривается как базовая параметрическая модель, а подбор весов формулируется как задача минимизации функции потерь. Для линейной регрессии в качестве функции потерь приводится среднеквадратическая ошибка MSE.

Иллюстративный пример. Требуется прогнозировать время обработки запроса сервера. Признаки: число активных пользователей, объём передаваемых данных и загрузка процессора. Целевая переменная — время ответа в миллисекундах.

Линейная регрессия

Линейная регрессия описывает зависимость числовой целевой переменной от одного или нескольких признаков линейной функцией. В простейшем случае модель можно записать как $y = b_0 + b_1x$, где b_0 — свободный член, b_1 — вес признака. Для нескольких признаков используется вектор весов.

Обучение заключается в выборе таких коэффициентов, при которых расхождение между фактическими и прогнозируемыми значениями минимально. В учебном пособии подчёркивается чувствительность линейной регрессии к мультиколлинеарности — сильной зависимости между объясняющими признаками.

Преимущества	**Ограничения**
Простая интерпретация коэффициентов	Линейная форма зависимости подходит не для всех данных
Быстрое обучение	Чувствительность к выбросам может ухудшать результат
Удобна как базовая модель	Мультиколлинеарность затрудняет интерпретацию коэффициентов

Метод k-ближайших соседей (kNN)

Метод k-ближайших соседей используется и для классификации, и для регрессии. Для нового объекта находятся k наиболее близких объектов обучающей выборки. В классификации решение обычно принимается голосованием соседей, а в регрессии — по среднему или медианному значению их целевой переменной.

Пример. Новое сетевое соединение описывается признаками duration, packets и bytes. Если среди пяти ближайших по этим признакам соединений четыре относятся к классу attack, алгоритм может классифицировать новый объект как attack.

Выбор k существенно влияет на результат. Слишком малое k повышает чувствительность к шуму, а слишком большое может сглаживать границы между классами. В пособии рекомендуется определять подходящее значение с помощью кросс-валидации.

Метрика расстояния	Идея	Когда полезна
Евклидово расстояние	Прямая геометрическая дистанция между точками	Числовые признаки сопоставимого масштаба
Манхэттенское расстояние	Сумма абсолютных различий по координатам	Данные, где важны покоординатные различия
Чебышёва	Максимальное различие по одной координате	Задачи, где критичен самый большой разрыв
Косинусное расстояние	Ориентация векторов, а не только их длина	Высокоразмерные векторные представления

Практически важно масштабировать признаки: если bytes измеряется тысячами, а duration — единицами, немасштабированный признак bytes может непропорционально влиять на расстояние.

Деревья решений

Дерево решений представляет правила в иерархической структуре. Внутренние узлы содержат проверки признаков, ветви соответствуют результатам проверок, а листья содержат итоговое решение. В пособии построение дерева описывается как последовательное разбиение исходного множества примеров сверху вниз.

Для выбора признака разбиения используются критерии, оценивающие качество разделения. В описании алгоритма C4.5 рассматриваются энтропия и нормированный прирост информации. Чтобы уменьшить излишнюю сложность дерева, применяется отсечение ветвей (pruning).

Иллюстративный пример дерева решений для оценки риска входа

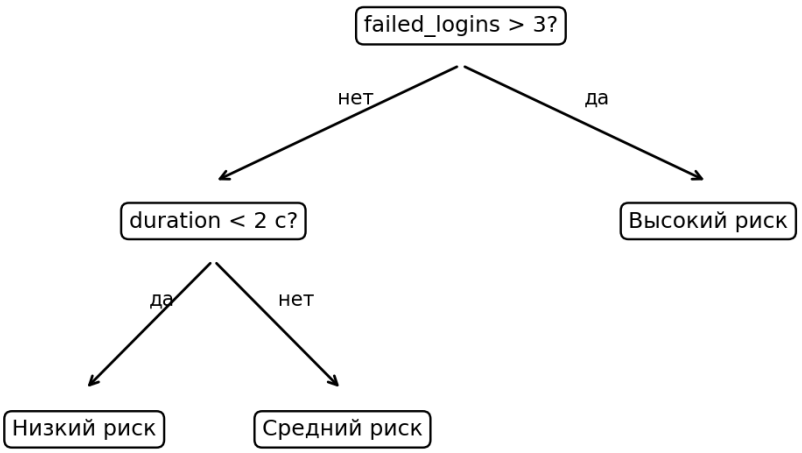


Рисунок 3 – Иллюстративный пример структуры дерева решений

На рисунке показан учебный пример, а не готовое правило реальной системы безопасности. Реальные пороги должны определяться на обучающих данных и проверяться на независимой выборке.

Преимущества дерева	Ограничения дерева
Наглядная логика решения	Глубокое дерево может переобучаться
Работа с нелинейными зависимостями	Небольшое изменение данных может изменить структуру
Можно оценивать важность признаков	Нужен контроль глубины и правил остановки
Удобно объяснять отдельные решения	Одно дерево нередко уступает ансамблевым методам по устойчивости

Логистическая регрессия

Логистическая регрессия применяется, когда исследуемая переменная является качественной и может быть представлена индикатором 0 или 1. В учебном пособии показано, что для такой задачи линейный отклик неудобен, поскольку прогноз вероятности должен оставаться в интервале от 0 до 1. Поэтому используется сигмоидальная логистическая функция.

Логистическая модель оценивает вероятность принадлежности к классу. Связь между линейным предиктором и вероятностью выражается через logit-преобразование — логарифм отношения шансов.

Пример. Пусть 1 означает подозрительный вход, а 0 — нормальный. Модель по признакам failed_logins, unusual_time и new_device оценивает вероятность класса 1. Значение 0,86 означает высокую оценённую вероятность подозрительного события, но окончательное действие системы зависит от выбранного порога и политики безопасности.

Линейная регрессия	Логистическая регрессия
Прогнозирует числовую величину	Оценивает вероятность класса
Цель непрерывная	Цель обычно 0/1 для бинарного случая
Типичный критерий — ошибка прогноза, например MSE	Параметры оцениваются вероятностным способом
Пример: время ответа сервера	Пример: вероятность подозрительного входа

Обучение без учителя

При обучении без учителя целевые метки отсутствуют. Алгоритм анализирует структуру признакового пространства и пытается обнаружить группы, сходство, скрытые закономерности или компактное представление данных.

1. Кластеризация

Кластеризация объединяет похожие объекты в группы. В информационной безопасности это может использоваться для исследования профилей сетевой активности: обычные рабочие станции, серверы, устройства с необычным поведением могут формировать различные группы. Кластер сам по себе не означает «атака» или «норма» — его смысл должен интерпретироваться после анализа.

2. Снижение размерности

При большом количестве признаков данные становятся сложнее анализировать и визуализировать. Методы снижения размерности формируют более компактное представление, сохраняя существенную структуру данных. Это может быть полезно перед визуальным анализом, кластеризацией или построением последующих моделей.

Обучение с подкреплением

В обучении с подкреплением агент взаимодействует со средой: наблюдает состояние, выбирает действие и получает награду. Цель обучения — сформировать стратегию поведения, увеличивающую ожидаемую суммарную награду.

Элемент	Смысл	Иллюстративный пример
Агент	Система, принимающая решение	Модуль управления
Среда	То, с чем взаимодействует агент	Вычислительная инфраструктура
Состояние	Текущая информация о среде	Нагрузка и доступные ресурсы
Действие	Выбор агента	Перераспределить ресурсы
Награда	Сигнал качества действия	Снижение задержки при сохранении ограничений

Этот подход принципиально отличается от обычной классификации: здесь важно не только отдельное предсказание, но и последовательность действий и их отложенные последствия.

Обучающая, валидационная и тестовая выборки

Качество модели нельзя оценивать только на тех данных, на которых она обучалась. Для объективной проверки данные разделяют. Обучающая часть используется для подбора параметров модели; валидационная — для выбора гиперпараметров и сравнения вариантов; тестовая — для итоговой независимой оценки.

Выборка	Назначение	Что нельзя делать
Train	Обучение параметров модели	Оценивать по ней итоговое качество обобщения
Validation	Настройка гиперпараметров и выбор варианта	Постоянно подгонять под неё решение без контроля
Test	Финальная независимая оценка	Использовать для обучения или выбора параметров

Переобучение и недообучение

Переобучение возникает, когда модель слишком хорошо приспосабливается к обучающим данным, включая их случайные особенности, но хуже работает на новых объектах. Недообучение возникает, когда модель слишком проста или обучена недостаточно и не описывает даже основные закономерности.

Состояние	Качество на train	Качество на новых данных	Типичная реакция
Недообучение	Низкое	Низкое	Улучшить признаки или выбрать более подходящую модель
Хорошее обобщение	Высокое	Сопоставимо высокое	Сохранить схему и подтвердить независимой оценкой

Переобучение	Очень высокое	Заметно хуже	Упростить модель, регуляризовать, увеличить данные, корректно валидировать

Оценка качества моделей

Метрика должна соответствовать задаче. Для классификации одной accuracy может быть недостаточно, особенно если классы несбалансированы. В задачах безопасности редкий класс атаки может быть значительно важнее многочисленного нормального класса.

Метрика	Что показывает	Практический смысл
Accuracy	Долю правильных ответов	Общая точность при относительно сбалансированных классах
Precision	Долю истинных положительных среди всех предсказанных положительных	Насколько часто тревога действительно относится к нужному классу
Recall	Долю найденных положительных среди всех реально положительных	Сколько реальных атак удалось обнаружить
F1-score	Баланс precision и recall	Удобна при необходимости учитывать обе характеристики
MSE / MAE	Величину ошибки числового прогноза	Используются в задачах регрессии

Пример. В наборе 10 000 событий только 100 атак. Модель, которая всегда отвечает normal, получит accuracy 99%, но не обнаружит ни одной атаки. Поэтому оценка должна учитывать структуру данных и последствия ошибок.

Сравнение рассмотренных методов

Метод	Тип задачи	Сильная сторона	Ограничение	Иллюстративное применение
Линейная регрессия	Регрессия	Простота и интерпретируемость	Линейность зависимости	Прогноз времени ответа
Логистическая регрессия	Классификация	Вероятностный результат, интерпретируемость	Линейная граница в исходном пространстве без усложнений	Оценка риска входа
kNN	Классификация/регрессия	Простая идея сходства объектов	Стоимость поиска и чувствительность к масштабу	Сравнение нового соединения с известными
Дерево решений	Классификация	Наглядные правила и	Риск переобучения	Классификация событий по

		нелинейные разбиения		последовательности условий
Кластеризация	Без учителя	Поиск естественных групп	Кластеры требуют интерпретации	Группировка профилей активности

Как выбирать подход

Выбор начинается не с названия алгоритма, а с постановки задачи. Если известны правильные ответы и требуется категория — это классификация. Если известны ответы и требуется число — регрессия. Если меток нет и требуется обнаружить группы — кластеризация. Если задача состоит в выборе последовательности действий на основе награды — обучение с подкреплением.

После определения типа задачи сравнивают несколько подходящих моделей. При этом учитывают качество, интерпретируемость, объём данных, масштаб признаков, вычислительную стоимость и требования к времени получения прогноза.

Вопрос	**Если ответ «да»**	**Направление выбора**
Есть известная целевая переменная?	Да	Обучение с учителем
Цель — категория?	Да	Классификация
Цель — число?	Да	Регрессия
Целевых меток нет?	Да	Обучение без учителя
Нужно найти естественные группы?	Да	Кластеризация
Решение является последовательностью действий с наградой?	Да	Обучение с подкреплением

Развёрнутый пример: обнаружение подозрительного сетевого события

Предположим, имеется набор исторических сетевых событий. Для каждого события известны duration, packets, bytes, failed_logins и метка normal/attack. Это задача обучения с учителем и бинарной классификации.

1. Проверить качество и состав данных, исключить некорректные записи и определить признаки.
2. Разделить данные на обучающую и тестовую части; при настройке модели использовать отдельную валидацию или кросс-валидацию.
3. Выбрать базовую модель, например логистическую регрессию, а затем сравнить её с kNN и деревом решений.
4. Для kNN масштабировать числовые признаки и подобрать k.
5. Для дерева решений подобрать ограничения сложности и проверить риск переобучения.
6. Сравнить модели не только по accuracy, но и по precision, recall и F1-score.
7. Проанализировать ошибки: какие атаки пропускаются и какие нормальные события ошибочно вызывают тревогу.

Такой пример показывает, что выбор алгоритма — только одна часть ML-задачи. Корректность данных, схема проверки и подходящая метрика не менее важны.

17. Практический пример на Python

Ниже приведён компактный учебный пример классификации. Он показывает общую последовательность действий и не заменяет полноценную проверку данных и настройку модели.

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report

X = data[['duration', 'packets', 'bytes', 'failed_logins']]
y = data['label']

X_train, X_test, y_train, y_test = train_test_split(
   X, y, test_size=0.2, random_state=42, stratify=y
)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

model = KNeighborsClassifier(n_neighbors=5)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```

Основные выводы:

– Машинное обучение строит модели на основе данных, а не только заранее заданных правил.

– Обучение с учителем используется при наличии правильных ответов; его основные задачи — классификация и регрессия.

– Обучение без учителя предназначено для поиска структуры в данных без целевых меток.

– Обучение с подкреплением основано на взаимодействии агента со средой и сигнале награды.

– Линейная регрессия, kNN, деревья решений и логистическая регрессия иллюстрируют разные принципы построения моделей.

– Выбор метода должен учитывать свойства данных, метрики, интерпретируемость и риск переобучения.

– В задачах информационной безопасности особенно важно учитывать дисбаланс классов и цену ложных срабатываний и пропусков атак.

Контрольные вопросы:

1. Что понимается под машинным обучением?
2. Чем алгоритм машинного обучения отличается от обученной модели?
3. Какие компоненты включает задача машинного обучения?
4. Какие три основных подхода машинного обучения выделяют?
5. Что такое обучение с учителем?

6. Чем классификация отличается от регрессии?
7. Приведите примеры бинарной и многоклассовой классификации.
8. Для каких задач используется линейная регрессия?
9. Какова роль функции потерь в линейной регрессии?
10. Что такое мультиколлинеарность?
11. В чём состоит принцип метода k-ближайших соседей?
12. Почему выбор k влияет на качество kNN?
13. Какие метрики расстояния могут применяться в kNN?
14. Почему для kNN важно масштабирование признаков?
15. Как устроено дерево решений?
16. Что содержат узлы, ветви и листья дерева?
17. Для чего применяется pruning?
18. Что показывает прирост информации при построении дерева?
19. Почему логистическая регрессия используется для классификации?
20. Что представляет собой сигмоидальная функция?
21. Что такое обучение без учителя?
22. Для чего применяется кластеризация?
23. Что понимается под снижением размерности?
24. Как устроено обучение с подкреплением?
25. Зачем разделять данные на train, validation и test?
26. Что такое переобучение и недообучение?
27. Почему accuracy может быть недостаточной при дисбалансе классов?
28. Что показывают precision и recall?
29. Как выбирать подход машинного обучения для прикладной задачи?
30. Какие особенности следует учитывать при применении ML в информационной безопасности?

Список источников:

1. Бессмертный И.А., Авдюшина А.Е., Кугаевских А.В., Королева Ю.А., Рущенко Н.Г. Системы искусственного интеллекта. Методическое пособие. Санкт-Петербург: Университет ИТМО, 2024. 281 с.

2. Беликова Е.К., Гудилина И.С., Круглова М.А., Саратовская Л.Б., Ямпольская А.В. English Reader in Artificial Intelligence = Хрестоматия по искусственному интеллекту на английском языке. Учебное пособие. Москва, 2024. 171 с.

3. Баланов А.Н. Искусственный интеллект. Понимание, применение и перспективы. Учебник для вузов. Санкт-Петербург: Лань, 2024. 312 с.

4. Рассел С., Норвиг П. Искусственный интеллект: современный подход. 2-е изд. Москва: Вильямс, 2018. 230 с.