ЛЕКЦИЯ 6
МЕТОДЫ КЛАССИФИКАЦИИ И РЕГРЕССИИ

Цель лекции

Изучить постановку задач классификации и регрессии, основные методы их решения, принципы обучения и оценки моделей; научиться интерпретировать результаты классификаторов и регрессионных моделей и выбирать подходящий метод для практической задачи.

Классификация и регрессия как задачи обучения с учителем

Классификация и регрессия относятся к обучению с учителем. В обоих случаях имеется набор объектов, для которых известны входные признаки и правильные ответы. Различие определяется типом целевой переменной. В классификации требуется определить класс объекта, а в регрессии — числовое значение.

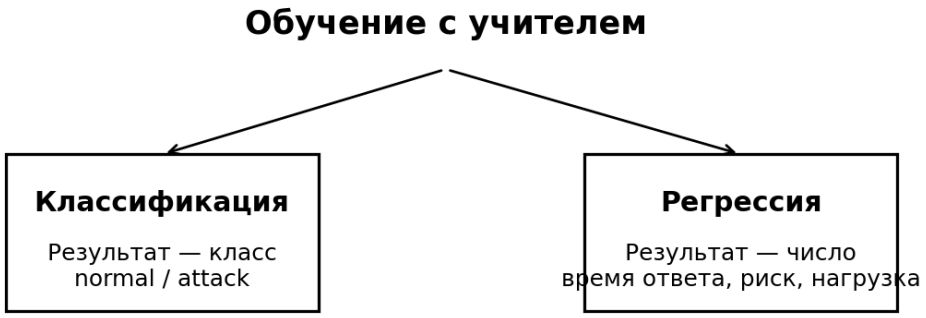


Рисунок 1 – Классификация и регрессия в обучении с учителем

Характеристика	**Классификация**	**Регрессия**
Целевая переменная	Категория или класс	Числовое значение
Пример	normal / attack	Время ответа сервера
Тип результата	Метка класса или вероятность класса	Числовой прогноз
Примеры методов	Логистическая регрессия, kNN, дерево решений, SVM	Линейная регрессия, kNN-регрессия, деревья регрессии
Примеры метрик	Accuracy, precision, recall, F1	MAE, MSE, MAPE, R^2

Постановка задачи классификации

Пусть каждый объект описывается набором признаков $x_1, x_2, \ldots, x_n$, а целевая переменная y принимает одно из конечного числа значений. Необходимо построить модель, которая по признакам нового объекта определит его класс.

Пример. Для сетевого соединения используются признаки duration, packets, bytes, failed_logins и protocol. Целевая переменная принимает значения normal и attack. Модель должна классифицировать ранее не встречавшееся соединение.

При двух классах говорят о бинарной классификации. При трёх и более классах — о многоклассовой классификации. Например, классификация сетевой активности может включать классы normal, DoS, brute force и malware.

Задача	Входные признаки	Класс
Фильтрация почты	Частота слов, ссылки, домен, признаки отправителя	spam / not spam
Обнаружение атаки	Параметры сетевого потока	attack / normal
Классификация вредоносного ПО	Статические и динамические признаки файла	семейство вредоносного ПО
Анализ события SOC	Параметры и описание события	категория инцидента

Логистическая регрессия

Логистическая регрессия является одним из базовых методов бинарной классификации. Несмотря на слово «регрессия» в названии, её результатом является оценка вероятности принадлежности объекта к определённому классу.

Если целевая переменная кодируется значениями 0 и 1, модель должна выдавать значение в интервале от 0 до 1. Для этого используется логистическая, или сигмоидальная, функция:

$$p = 1 / (1 + e^{(-z)}), \quad \text{где } z = b_0 + b_1x_1 + \ldots + b_nx_n.$$

При z = 0 вероятность равна 0,5. При увеличении z вероятность приближается к 1, а при уменьшении — к 0. Поэтому линейная комбинация признаков преобразуется в корректную вероятностную оценку.

Пример. Пусть модель оценила вероятность подозрительного входа как 0,84. При пороге 0,5 событие будет отнесено к классу suspicious. Если порог повысить до 0,9, то то же событие уже не будет отнесено к положительному классу. Следовательно, порог влияет на число ложных срабатываний и пропущенных событий.

Порог	Возможный эффект	Практический смысл
Ниже	Больше объектов относятся к положительному классу	Повышается чувствительность, но возможны дополнительные ложные тревоги
Выше	Меньше объектов относятся к положительному классу	Снижается число ложных тревог, но можно пропустить часть атак

Метод k-ближайших соседей для классификации

Метод k-ближайших соседей (kNN) классифицирует новый объект по классам наиболее близких к нему объектов обучающей выборки. В учебном пособии подчёркивается, что выбор k критичен: малое значение делает алгоритм чувствительным к шуму, а слишком большое может сгладить границы классов.

Пример. Для нового сетевого соединения найдены пять ближайших объектов. Четыре из них имеют метку attack, один — normal. При k = 5 результатом голосования будет класс attack.

k	Особенность	Риск
Малое, например 1–3	Решение определяется очень локальным окружением	Высокая чувствительность к шуму
Среднее	Компромисс между локальностью и устойчивостью	Требует подбора
Большое	Учитывается широкое окружение	Границы классов могут чрезмерно сглаживаться

Для kNN особенно важно масштабирование признаков. Если один признак измеряется тысячами, а другой — долями единицы, первый может доминировать при вычислении

расстояния. В качестве расстояния могут использоваться евклидова, манхэттенская, Чебышёва и другие метрики.

Деревья решений для классификации

Дерево решений представляет классификационные правила в иерархической структуре. Каждый внутренний узел проверяет некоторый признак, ветви соответствуют результатам проверки, а лист содержит итоговый класс.

При построении дерева необходимо выбрать критерий разбиения. В алгоритме C4.5 используется прирост информации. Последовательное разбиение продолжается до выполнения правила остановки. Для ограничения сложности дерева применяется отсечение ветвей — pruning.

Элемент дерева	**Назначение**	**Пример**
Корневой узел	Первая проверка	failed_logins > 3
Внутренний узел	Следующая проверка признака	duration < 2
Ветвь	Результат проверки	да / нет
Лист	Итоговый прогноз	attack

Преимущество дерева — наглядность. Можно проследить последовательность условий, приведшую к решению. Недостаток — склонность глубоких деревьев к переобучению.

Метод опорных векторов

Метод опорных векторов (Support Vector Machine, SVM) строит разделяющую границу между классами так, чтобы расстояние до ближайших обучающих объектов было максимально возможным. Эти ближайшие объекты называют опорными векторами.

Для линейно разделимых данных граница может быть гиперплоскостью. Для более сложных зависимостей применяются ядровые функции, позволяющие строить нелинейные разделения в преобразованном пространстве признаков.

Сильные стороны SVM	**Ограничения**
Эффективен в пространствах с большим числом признаков	Чувствителен к выбору параметров и масштабу признаков
Может строить нелинейные границы с помощью ядер	Обучение на очень больших наборах может быть затратным
Хорошо подходит для бинарной классификации	Интерпретация сложнее, чем у небольшого дерева решений

Пример. При классификации текстовых описаний инцидентов после преобразования текста в числовые векторы SVM может отделять одну категорию событий от другой по комбинации большого числа признаков.

Оценка качества классификации

Результат классификатора оценивается не только общей долей правильных ответов. Для бинарной классификации удобно использовать матрицу ошибок, которая показывает количество правильных и ошибочных решений для каждого класса.

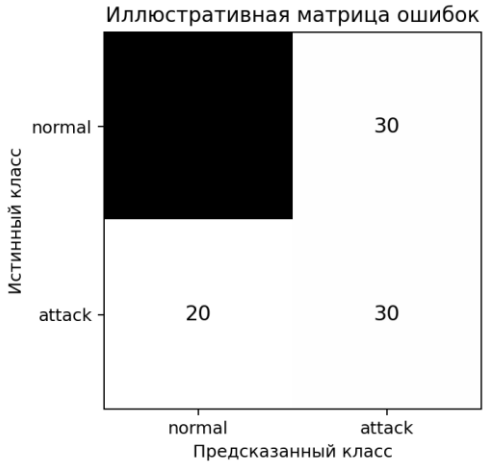


Рисунок 2 – Иллюстративная матрица ошибок: 1000 объектов, из них 50 атак

Обозначение	**Смысл**
TP — True Positive	Атака правильно распознана как атака
TN — True Negative	Нормальный объект правильно распознан как нормальный
FP — False Positive	Нормальный объект ошибочно признан атакой
FN — False Negative	Атака ошибочно признана нормальным объектом

Для показанной иллюстрации TP = 30, TN = 920, FP = 30, FN = 20. Из этих четырёх значений вычисляются основные метрики.

Accuracy = (TP + TN) / (TP + TN + FP + FN).

Precision = TP / (TP + FP).

Recall = TP / (TP + FN).

F1 = 2 × Precision × Recall / (Precision + Recall).

В нашем иллюстративном примере accuracy = 95%, precision = 50%, recall = 60%. Это показывает, почему высокая accuracy сама по себе не гарантирует хорошего обнаружения редкого класса.

Метрика	**Что показывает**	**Когда особенно важна**
Accuracy	Общую долю правильных прогнозов	При относительно сбалансированных классах
Precision	Насколько надёжны положительные срабатывания	Когда ложная тревога дорога
Recall	Какую долю реальных положительных объектов нашли	Когда пропуск атаки особенно опасен
F1-score	Баланс precision и recall	При дисбалансе и необходимости учитывать обе ошибки

ROC-кривая и AUC

Если классификатор выдаёт вероятность или оценку принадлежности к классу, его поведение можно исследовать при разных порогах. ROC-кривая показывает соотношение между долей правильно обнаруженных положительных объектов и долей ложноположительных решений при изменении порога.

AUC — площадь под ROC-кривой. Чем выше AUC, тем лучше модель в среднем разделяет классы при различных порогах. Однако выбор конкретного порога всё равно должен учитывать цену ошибок в прикладной системе.

В информационной безопасности нельзя выбирать порог только по максимальной общей точности. Для критичного сервиса может быть важнее высокий recall, тогда как для системы с дорогими ложными блокировками требуется контролировать false positive.

Постановка задачи регрессии

В регрессии целевая переменная является числовой. Требуется построить функцию, которая по входным признакам оценивает значение y.

Примеры: прогноз времени ответа сервера; прогноз нагрузки; оценка объёма трафика; прогноз количества обращений; оценка времени устранения инцидента.

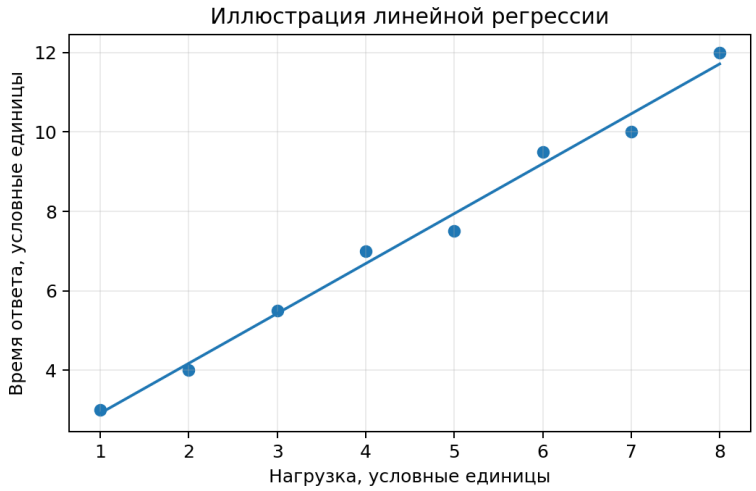


Рисунок 3 – Иллюстрация линейной зависимости и линии регрессии

Линейная регрессия

Линейная регрессия описывает зависимость целевой переменной от признаков линейной функцией. Для одного признака модель записывается как $\hat{y} = b_0 + b_1 x$. Для нескольких признаков используется сумма взвешенных признаков.

Задача обучения состоит в подборе коэффициентов, при которых прогнозы максимально близки к фактическим значениям. В загруженном пособии эта задача рассматривается как задача оптимизации; для простой линейной регрессии используется метод наименьших квадратов и среднеквадратическая ошибка MSE.

Пример. Пусть x — нагрузка на сервис, а y — время ответа. После обучения модель может показать, как ожидаемое время ответа изменяется при увеличении нагрузки. Если зависимость существенно нелинейна, простая линейная модель может быть недостаточной.

Преимущество	**Ограничение**
Простая математическая форма	Не описывает сложные нелинейные зависимости без преобразований
Интерпретируемые коэффициенты	Чувствительна к выбросам
Быстро обучается	Мультиколлинеарность признаков затрудняет интерпретацию

kNN и деревья в задачах регрессии

Идея kNN может применяться и к регрессии. Вместо голосования по классам берётся среднее или медианное значение целевой переменной у ближайших соседей. Например, время обработки нового запроса можно оценить по времени обработки наиболее похожих запросов.

Деревья также могут решать задачи регрессии. Пространство признаков разделяется условиями, а в листе формируется числовой прогноз, например среднее значение целевой переменной для объектов соответствующей области.

Метрики регрессии

Метрика	**Формула / идея**	**Интерпретация**
MAE	Среднее $\lvert y - \hat{y} \rvert$	Средняя абсолютная ошибка в единицах целевой переменной
MSE	Среднее $(y - \hat{y})^2$	Сильнее штрафует большие ошибки
MAPE	Средняя относительная ошибка в процентах	Удобна для относительной интерпретации, но проблемна при значениях около нуля
R^2	Доля объяснённой вариации относительно базового прогноза	Показывает, насколько модель описывает вариацию целевой переменной

Пример. Истинное время обработки трёх запросов равно 10, 20 и 30 мс, а прогнозы — 12, 18 и 33 мс. Абсолютные ошибки равны 2, 2 и 3 мс, поэтому MAE = (2 + 2 + 3) / 3 ≈ 2,33 мс.

Сравнение основных методов

Метод	**Классификация**	**Регрессия**	**Нужное масштабирование**	**Интерпретируемость**
Логистическая регрессия	Да	Нет в стандартной постановке	Желательно	Высокая
kNN	Да	Да	Особенно важно	Средняя
Дерево решений	Да	Да	Обычно не требуется	Высокая для небольшого дерева
SVM	Да	Возможна регрессионная модификация SVR	Важно	Средняя
Линейная регрессия	Нет	Да	Зависит от задачи	Высокая

Дисбаланс классов

В реальных задачах безопасности положительный класс часто встречается значительно реже нормального. Например, из 100 000 событий подавляющее большинство может быть нормальным. В такой ситуации модель способна получить высокую accuracy, почти всегда выбирая большинство, но плохо обнаруживать атаки.

Поэтому необходимо анализировать распределение классов, confusion matrix, precision, recall и F1-score. При разделении выборки полезно сохранять пропорции классов. При необходимости применяются методы балансировки, изменение весов классов и настройка порога решения.

Ошибки первого и второго типа в прикладной системе

Ошибка	**Для детектора атак**	**Возможное последствие**

False Positive	Нормальное событие признано атакой	Лишняя тревога, блокировка легитимного действия, нагрузка на аналитика
False Negative	Атака признана нормальным событием	Пропуск инцидента и возможное развитие атаки

Нельзя утверждать, что одна из ошибок всегда важнее другой. Их цена определяется контекстом. ***Например***, для системы блокировки критичных транзакций большое число false positive может нарушить работу пользователей, а для раннего обнаружения опасной атаки особенно критичен false negative.

Выбор метода классификации или регрессии

Условие	Возможный выбор
Нужна простая вероятностная бинарная модель	Логистическая регрессия
Важна наглядность правил	Дерево решений
Объекты хорошо описываются близостью	kNN
Много признаков и нужна разделяющая граница	SVM
Нужно прогнозировать число и зависимость близка к линейной	Линейная регрессия
Нужен локальный числовой прогноз по похожим объектам	kNN-регрессия

Выбор должен подтверждаться экспериментом. Несколько моделей обучают на одинаковой обучающей выборке и сравнивают на одинаковых независимых данных с использованием подходящих метрик.

Пример построения классификационной модели в Python

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix

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 = LogisticRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

Пример построения регрессионной модели в Python

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

X = data[['users', 'traffic', 'cpu_load']]
y = data['response_time']

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

model = LinearRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

print('MAE:', mean_absolute_error(y_test, y_pred))
print('MSE:', mean_squared_error(y_test, y_pred))
print('R2:', r2_score(y_test, y_pred))
```

Развёрнутый пример: классификация сетевого трафика

Рассмотрим набор сетевых соединений с признаками duration, protocol, packets, bytes и failed_logins. Цель — определить normal или attack. Сначала проверяется качество данных и распределение классов. Затем данные разделяются на обучающую и тестовую части. Для методов, чувствительных к масштабу, выполняется стандартизация.

В качестве первой модели можно использовать логистическую регрессию. Она формирует вероятностную оценку и удобна как интерпретируемая базовая модель. Затем результат можно сравнить с kNN, деревом решений или SVM. Сравнение проводится на одной тестовой выборке.

Если логистическая регрессия имеет высокий precision, но недостаточный recall, часть атак остаётся необнаруженной. Можно исследовать изменение порога или другую модель. Если дерево показывает почти идеальный результат на train, но значительно худший на test, вероятно переобучение. Для kNN необходимо проверить масштабирование и подобрать k.

Итоговая модель выбирается не по одному числу, а по совокупности требований: качество обнаружения, число ложных тревог, скорость работы, интерпретируемость и устойчивость на новых данных.

Основные выводы:

– Классификация прогнозирует дискретный класс, а регрессия — числовое значение.

– Логистическая регрессия оценивает вероятность класса и широко применяется в бинарной классификации.

– kNN использует близость объектов и требует внимательного выбора k и масштабирования признаков.

– Деревья решений формируют последовательность интерпретируемых правил, но требуют контроля переобучения.

– SVM строит разделяющую границу с максимальным зазором и эффективен в пространствах с большим числом признаков.

– Для классификации необходимо анализировать confusion matrix, precision, recall, F1 и при необходимости ROC-AUC.

– Для регрессии применяются MAE, MSE, MAPE и R^2.

– В задачах информационной безопасности выбор порога и модели должен учитывать цену false positive и false negative.

Контрольные вопросы:

1. Чем задача классификации отличается от задачи регрессии?
2. Что такое бинарная и многоклассовая классификация?
3. Как формулируется задача классификации сетевого трафика?
4. Почему логистическая регрессия относится к методам классификации?
5. Для чего используется сигмоидальная функция?
6. Как порог классификации влияет на результат?
7. В чём принцип метода k-ближайших соседей?
8. Почему выбор k является важным гиперпараметром?
9. Почему kNN чувствителен к масштабу признаков?
10. Как работает дерево решений?
11. Что такое критерий разбиения дерева?
12. Для чего применяется pruning?
13. В чём основная идея метода опорных векторов?
14. Что такое опорные векторы?
15. Что показывают TP, TN, FP и FN?
16. Как вычисляется accuracy?
17. Что показывает precision?
18. Что показывает recall?
19. Для чего используется F1-score?
20. Почему accuracy может вводить в заблуждение при дисбалансе классов?
21. Что показывает ROC-кривая?
22. Что характеризует AUC?
23. Как формулируется задача линейной регрессии?
24. Что такое MSE и почему она штрафует большие ошибки?
25. Что показывает MAE?
26. Какие ограничения имеет MAPE?
27. Что показывает коэффициент R^2?
28. Как kNN применяется к регрессии?
29. Как деревья решений применяются к регрессии?
30. Почему цена false positive и false negative зависит от прикладной задачи?

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.

2. Искусственный интеллект и машинное обучение. Учебно-методические материалы по дисциплине. Разделы по постановке задач машинного обучения, метрикам регрессии и классификации, SVM, kNN, линейной регрессии и деревьям решений.

3. Рассел С., Норвиг П. Искусственный интеллект: современный подход. Москва: Вильямс.