Д. Серікбаев атындағы
ШҚТУ

Основы искусственного интеллекта

Введение в контролируемое и неконтролируемое обучение.

План лекции

1 Типы машинного обучения.

- Контролируемое обучение (Supervised Learning)

- Неконтролируемое обучение (Unsupervised Learning)

2 Ключевые показатели эффективности моделей машинного обучения .

Типы машинного обучения

- **Машинное обучение (ML)** – это раздел искусственного интеллекта, который позволяет компьютерам учиться на данных и делать прогнозы.

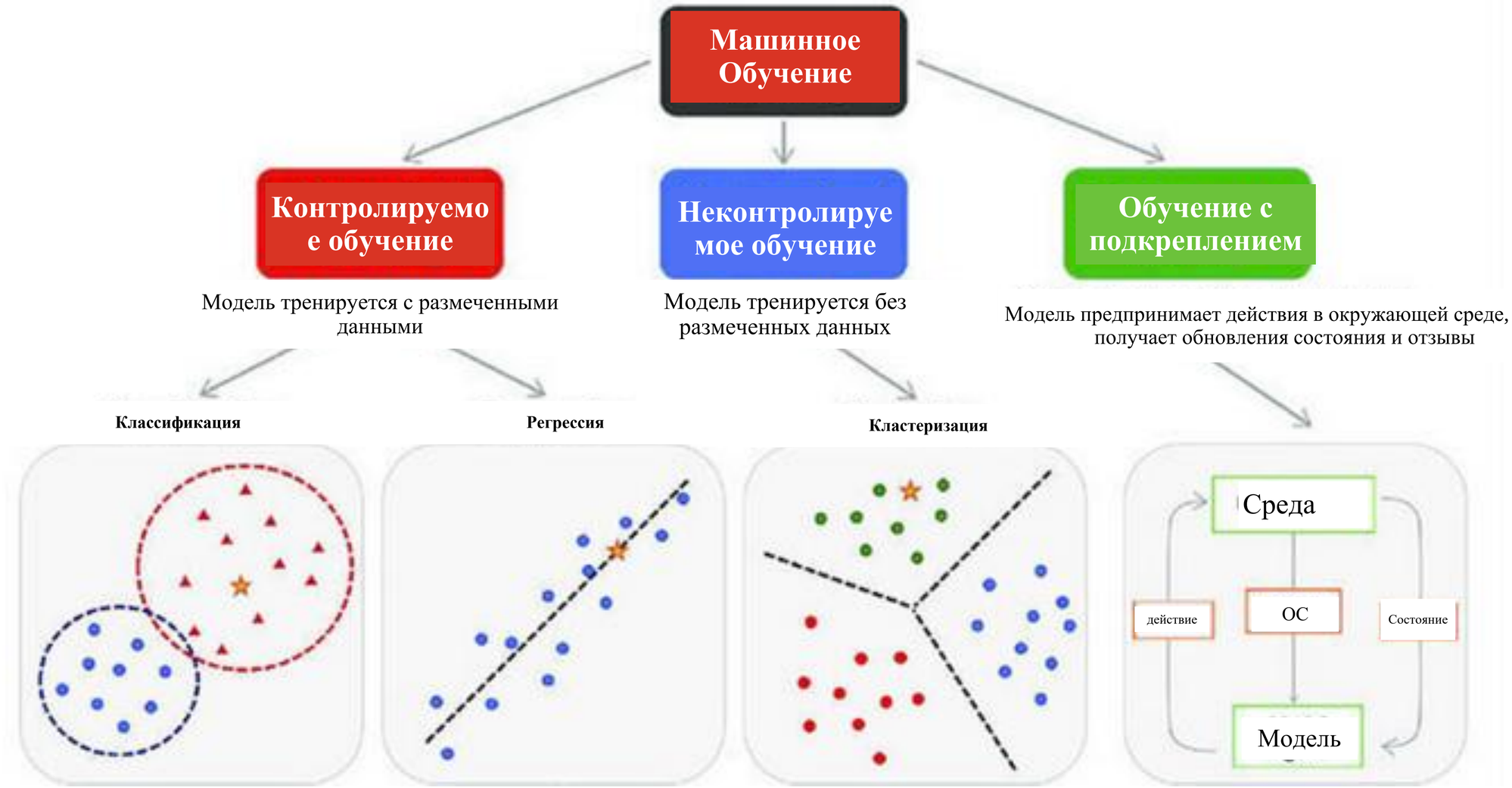
- В зависимости от типа данных и поставленных задач машинное обучение можно разделить на три основных категории:

  1. **Контролируемое обучение** (Supervised Learning)

  2. **Неконтролируемое обучение** (Unsupervised Learning)

  3. **Обучение с подкреплением** (Reinforcement Learning) (не рассматривается в данной лекции)

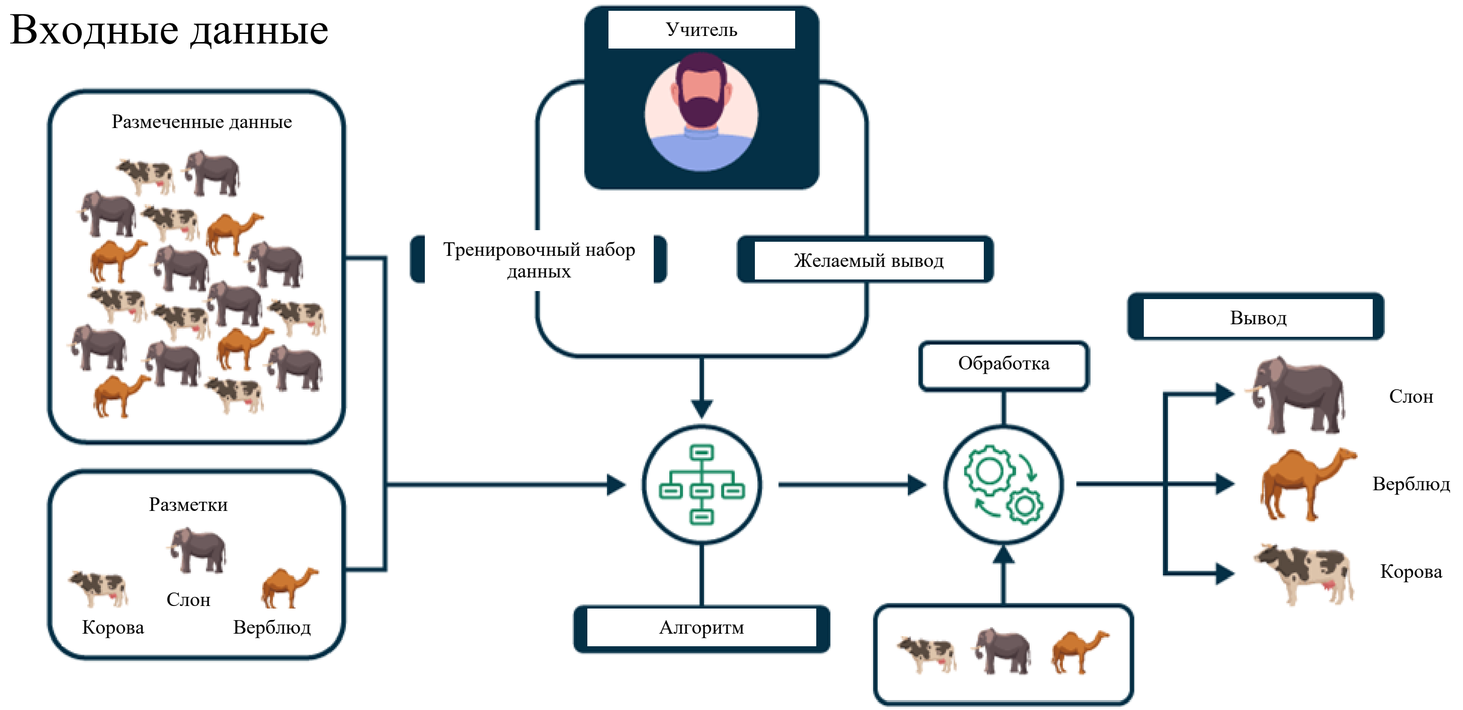
Типы машинного обучения



1 Контролируемое обучение

Определение

- **Контролируемое обучение** – это метод, при котором модель обучается на размеченных данных. Каждому входному примеру соответствует целевая переменная (метка), которую алгоритм пытается предсказать.



Примеры задач

- **Классификация**: модель предсказывает категорию (например, спам/не спам в почте, диагностика заболеваний по медицинским данным).
- **Регрессия**: модель предсказывает непрерывное значение (например, прогнозирование цен на недвижимость, предсказание температуры).

Алгоритмы Контролируемого обучения

- Линейная и логистическая регрессия
- Деревья решений (Decision Trees)
- Случайный лес (Random Forest)
- Градиентный бустинг (XGBoost, LightGBM, CatBoost)
- Метод опорных векторов (Support Vector Machines)
- Нейронные сети (MLP, CNN, RNN)

Преимущества и недостатки

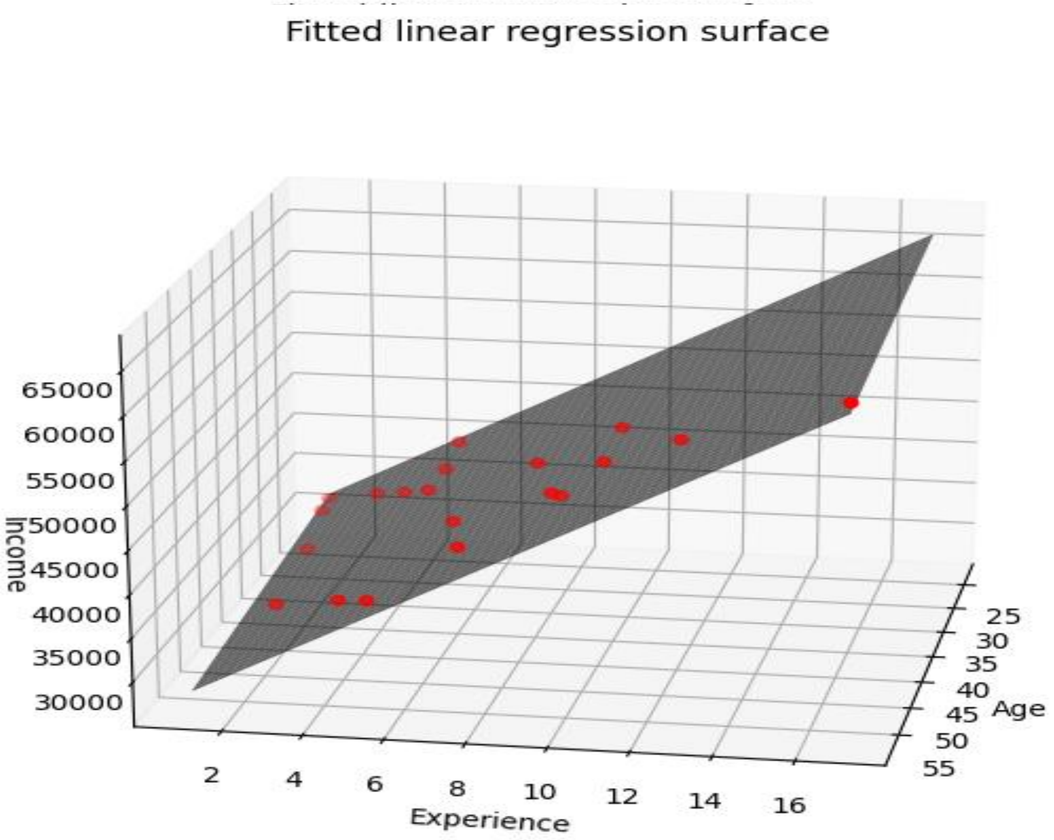
Контролируемое обучение

Преимущества	Недостатки
Высокая точность при наличии большого количества размеченных данных	Требуется размеченный набор данных
Возможность контроля качества обучения	Подверженность переобучению (overfitting)
Хорошо подходит для задач прогнозирования	Может требовать значительных вычислительных ресурсов

Алгоритмы Контролируемого обучения (Обучения с Учителем)
Линейная Регрессия

- В многомерном пространстве вместо линии или плоскости связь в данных будет описываться гиперплоскостью размерностью n-1.

- Формула: $y = wX + b$



Алгоритмы Контролируемого обучения (Обучения с Учителем)
Логистическая Регрессия

Описание: Классификационный алгоритм, предсказывающий вероятность принадлежности объекта к классу.

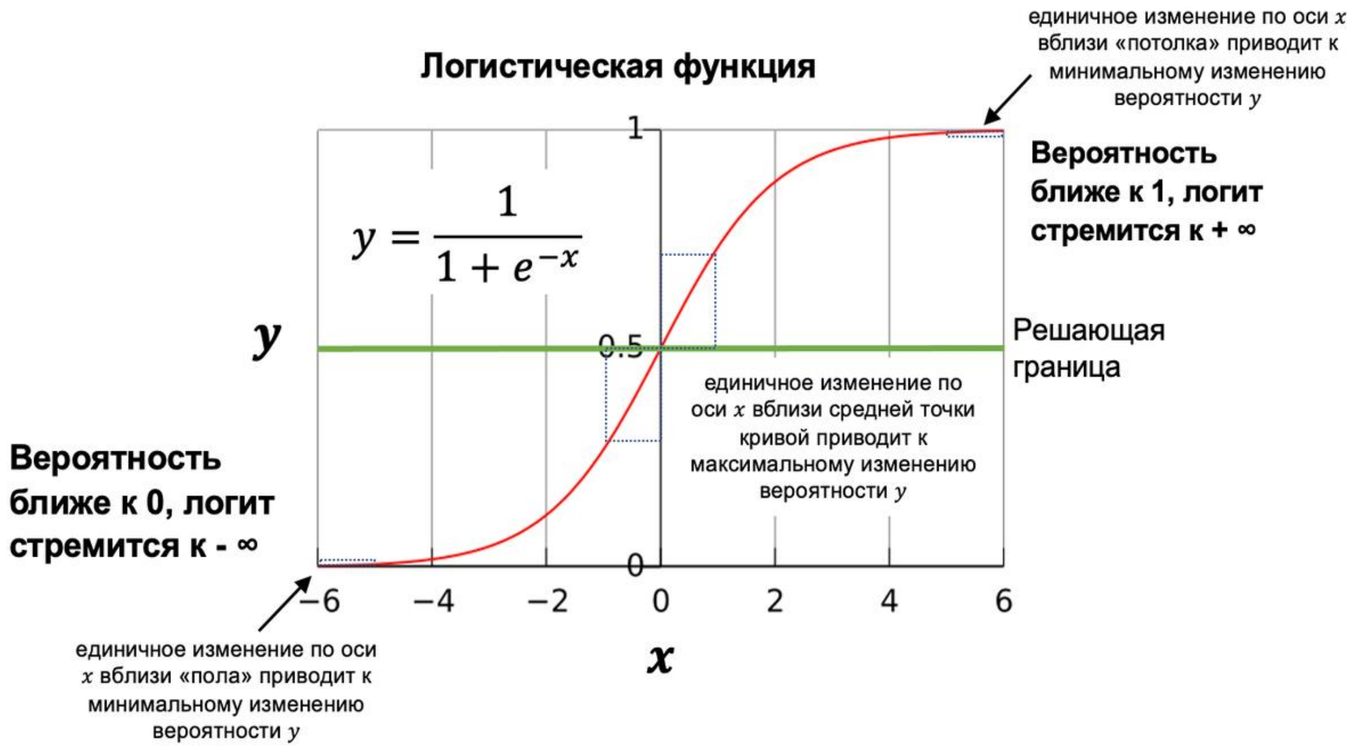
- Формула: $P(Y=1) = \frac{1}{1+e^{-z}}$, где $z = wX + b$

- Пример: Классификация спама в электронной почте

Плюсы и минусы:

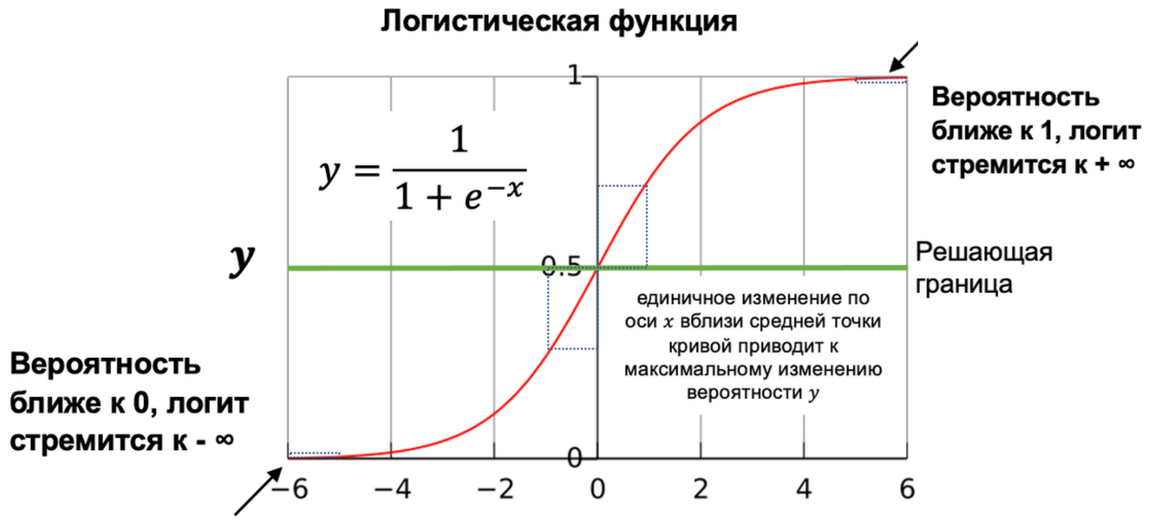
(+) Хорошо работает с бинарными задачами

(-) Плохо обрабатывает нелинейные зависимости



Логистическая регрессия (Logistic Regression)

- **Описание**: Логистическая регрессия используется для задач классификации, то есть для прогнозирования категориальных значений. Несмотря на название, это алгоритм классификации, а не регрессии.



Логистическая регрессия (Logistic Regression)

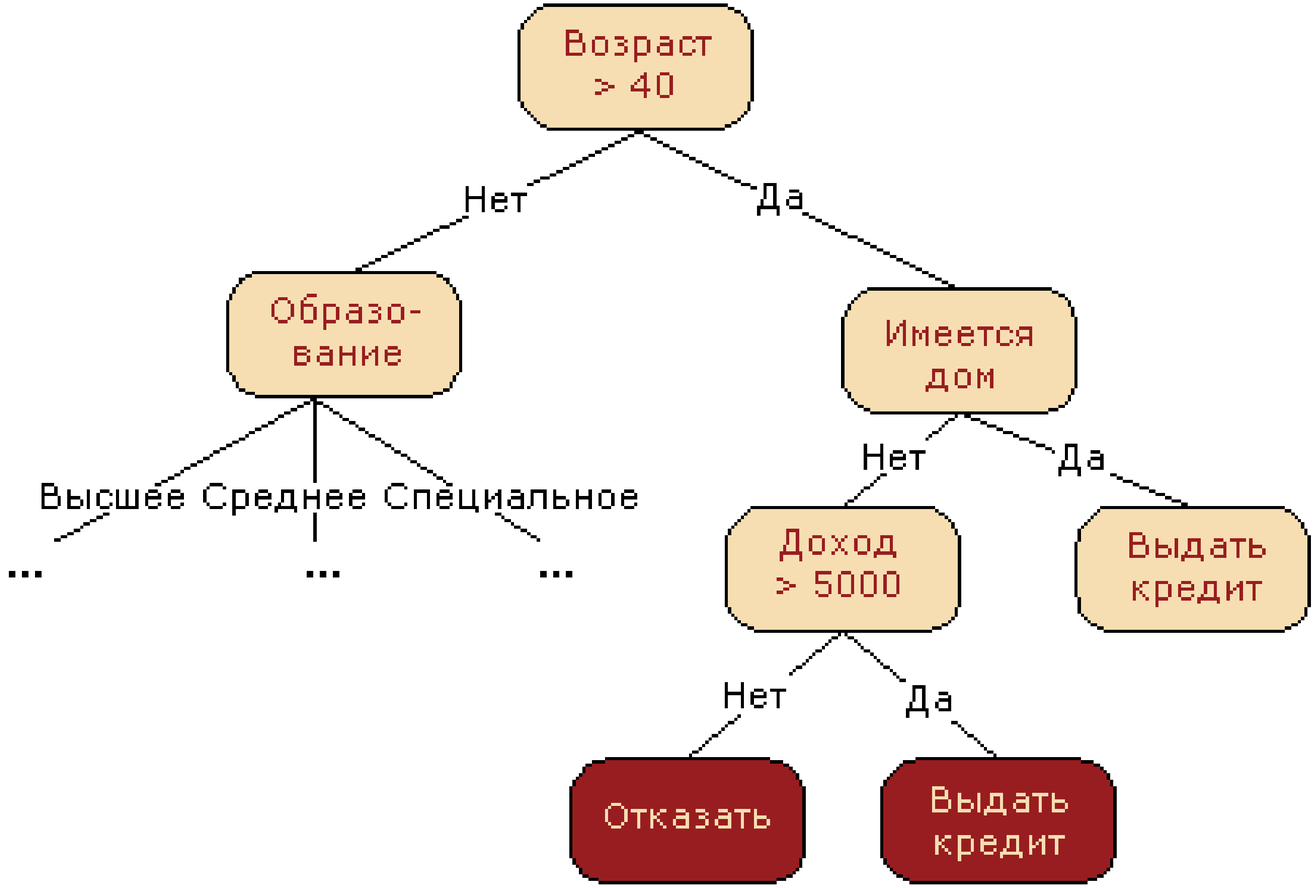
Пример на Python

```
from sklearn.linear_model import LogisticRegression
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split

# Загрузка данных
iris = load_iris()
X, y = iris.data, iris.target

# Разделение данных на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)

# Обучение модели
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)

# Оценка точности
accuracy = model.score(X_test, y_test)
print(accuracy)
```

Алгоритмы Контролируемого обучения (Обучения с Учителем) Деревья Решений

Описание: Алгоритм, принимающий решения, разделяя пространство данных по пороговым значениям.

- Пример: Определение, выдавать ли кредит клиенту банка.



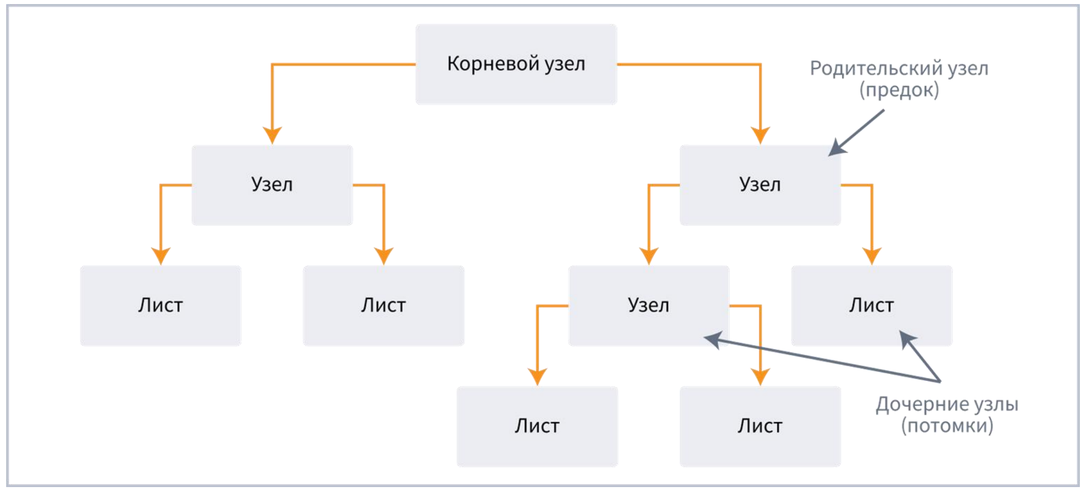
Плюсы и минусы:

(+) Легкость интерпретации

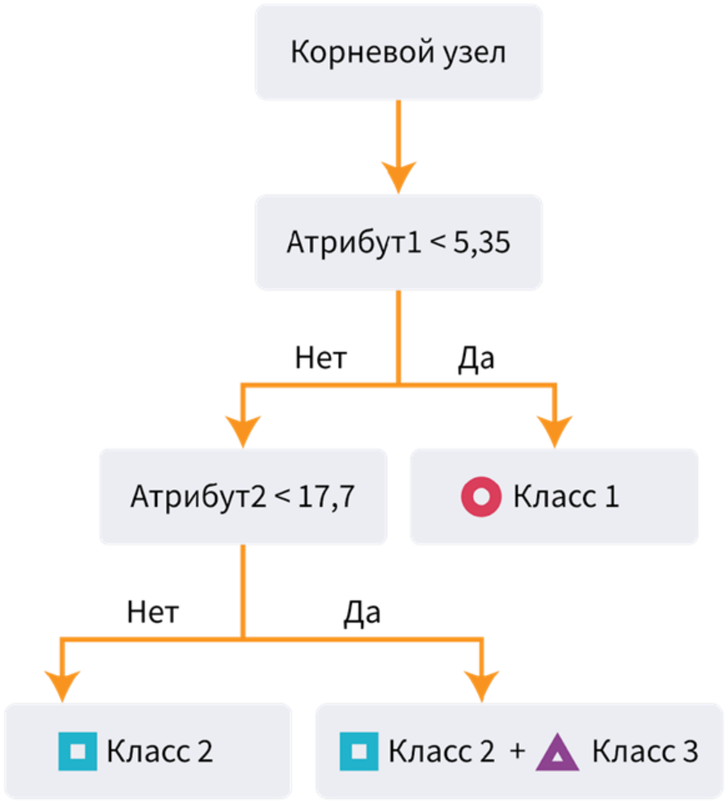
(-) Переобучение

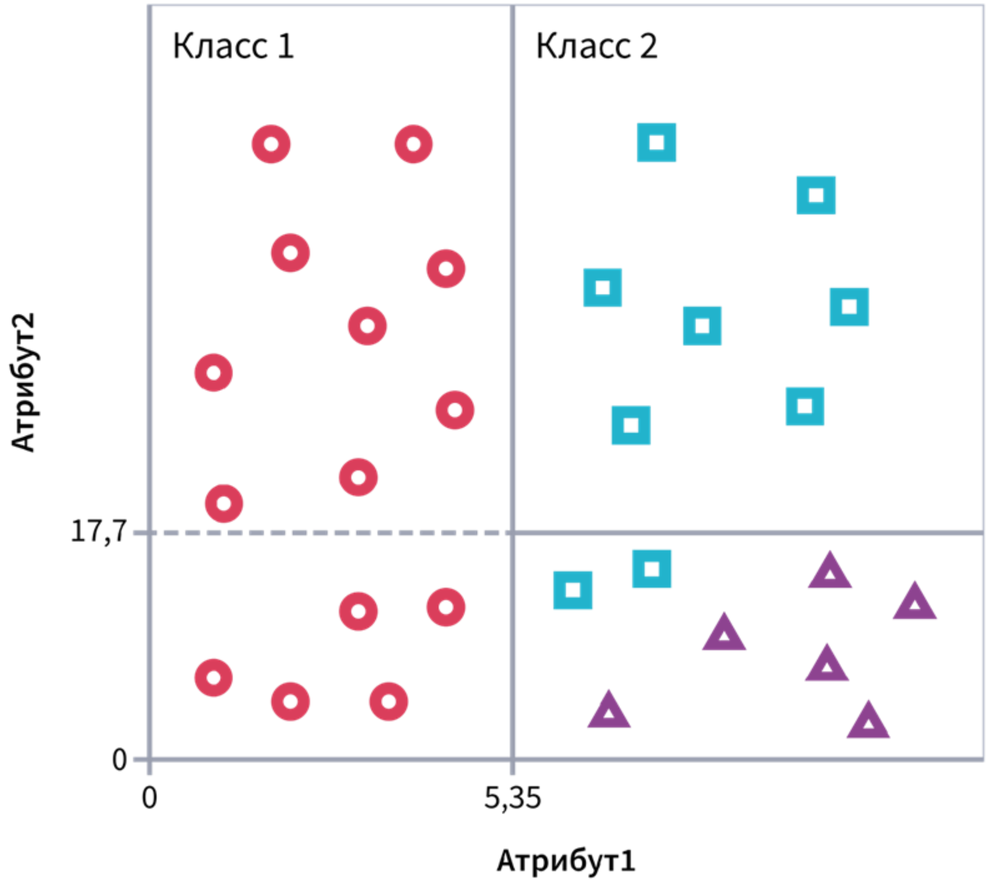
Деревья решений (Decision Trees)

- **Описание**: Деревья решений — это алгоритмы, которые принимают решения, основываясь на последовательности вопросов. Они могут использоваться как для задач регрессии, так и для задач классификации.



Деревья решений (Decision Trees)





Деревья решений (Decision Trees)

Пример на Python

```
from sklearn.tree import DecisionTreeClassifier


# Обучение модели

model = DecisionTreeClassifier()

model.fit(X_train, y_train)


# Оценка точности

accuracy = model.score(X_test, y_test)

print(accuracy)
```

Алгоритмы Контролируемого обучения (Обучения с Учителем)
Пример Дерева Решений для нескольких классов

Алгоритмы Контролируемого обучения (Обучения с Учителем)
Метод опорных векторов

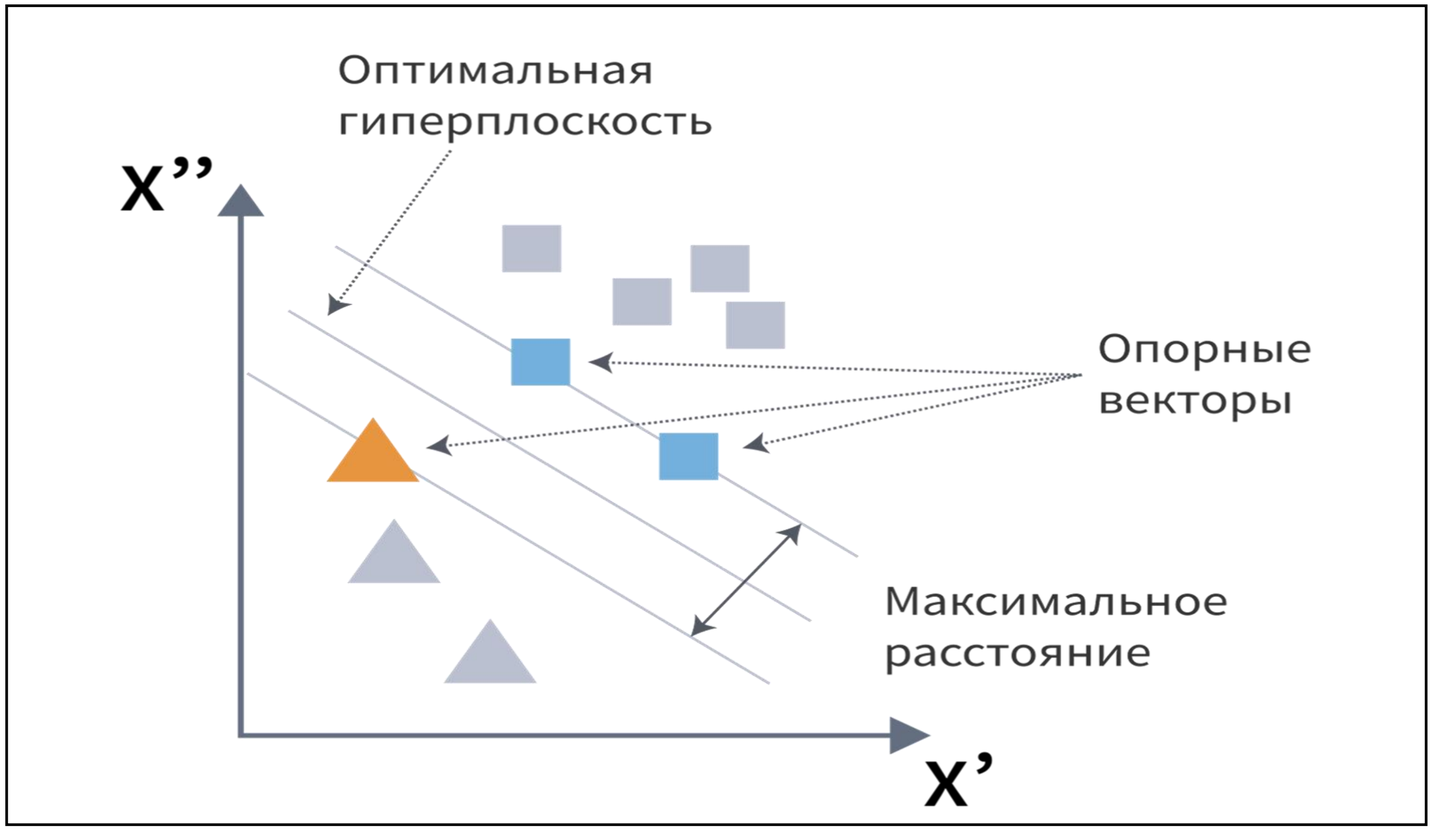
Описание: Создает гиперплоскость, разделяющую классы с максимальным отступом.

- Пример: Распознавание рукописных
цифр.

Плюсы и минусы:
(+) Хорошо работает на малых выборках
(-) Медленный на больших данных

Метод опорных векторов (Support Vector Machines, SVM)

- **Описание**: SVM — это мощный алгоритм, который может использоваться как для задач регрессии, так и для задач классификации. Он стремится найти оптимальную гиперплоскость, разделяющую классы.



Алгоритмы Контролируемого обучения (Обучения с Учителем) Метод опорных векторов

- Основная идея метода заключается в отображение векторов пространства признаков, представляющих классифицируемые объекты, в пространство более высокой размерности.

- Это связано с тем, что в пространстве большей размерности линейная разделимость множества оказывается выше, чем в пространстве меньшей размерности.

- Задача заключается в построении разделяющей гиперплоскость таким образом, чтобы максимизировать зазор — область пространства признаков между вспомогательными гиперплоскостями, в которой не должно быть векторов.

Метод опорных векторов (Support Vector Machines, SVM)

Пример на Python

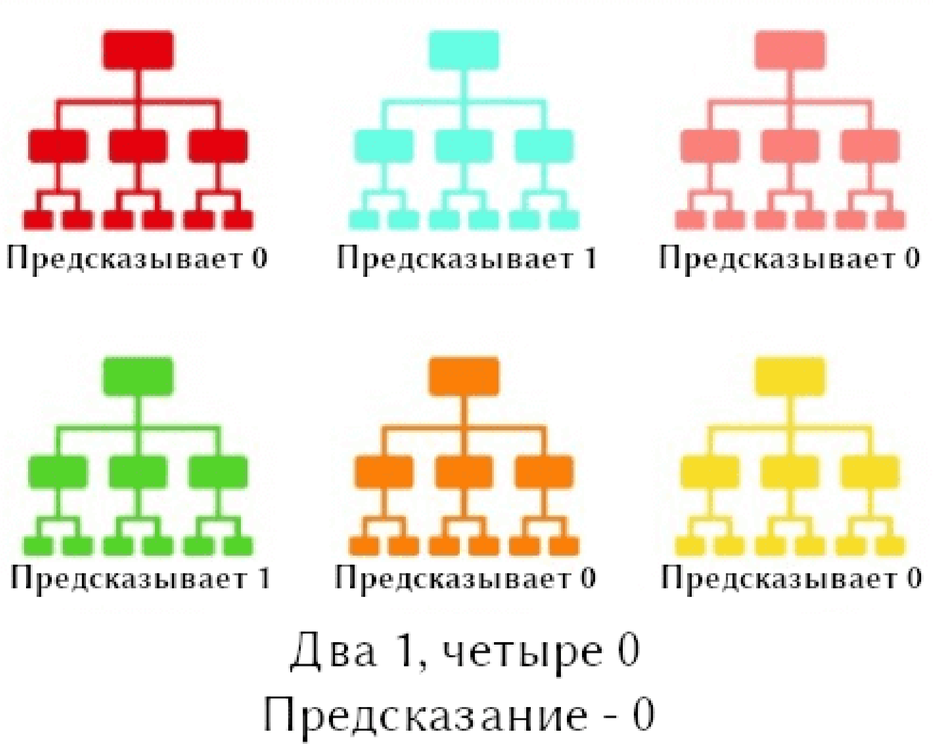
```
from sklearn.svm import SVC


# Обучение модели
model = SVC()
model.fit(X_train, y_train)


# Оценка точности
accuracy = model.score(X_test, y_test)
print(accuracy)
```

Алгоритмы Контролируемого обучения (Обучения с Учителем) Случайный Лес

- Описание: Ансамбль деревьев решений для повышения точности и снижения переобучения.

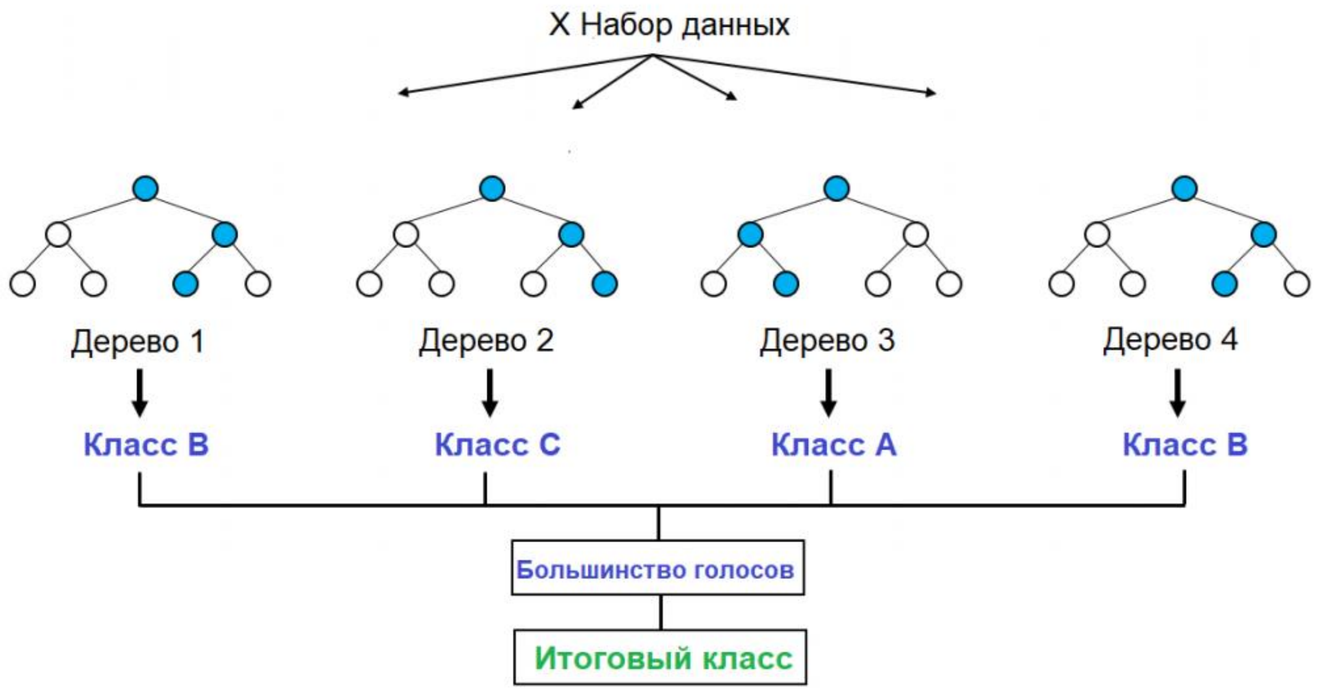
- Пример: Диагностика заболеваний по медицинским показателям.

Плюсы и минусы:
(+) Высокая точность
(-) Сложность интерпретации



Случайный лес (Random Forest)

- **Описание**: Случайный лес — это ансамблевый метод, который объединяет множество деревьев решений для повышения точности прогнозов.



Случайный лес (Random Forest)

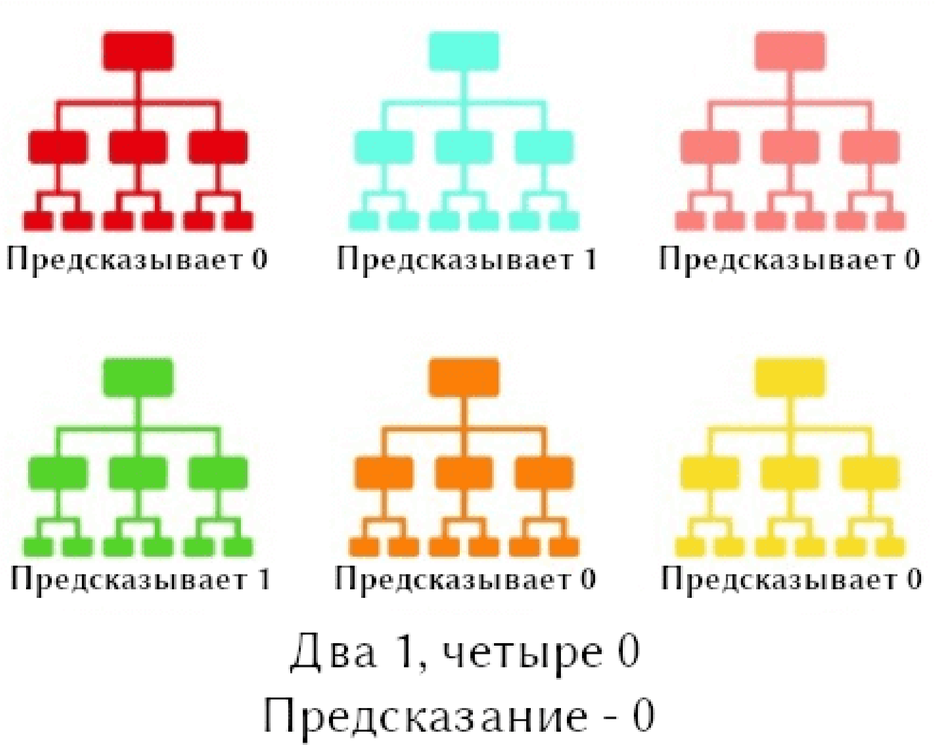
Пример на Python

```
from sklearn.ensemble import RandomForestClassifier


# Обучение модели

model = RandomForestClassifier()

model.fit(X_train, y_train)


# Оценка точности

accuracy = model.score(X_test, y_test)

print(accuracy)
```

Алгоритмы Контролируемого обучения (Обучения с Учителем)
Градиентный Бустинг

- Описание: Улучшенная версия случайного леса, строящая деревья последовательно для минимизации ошибки.

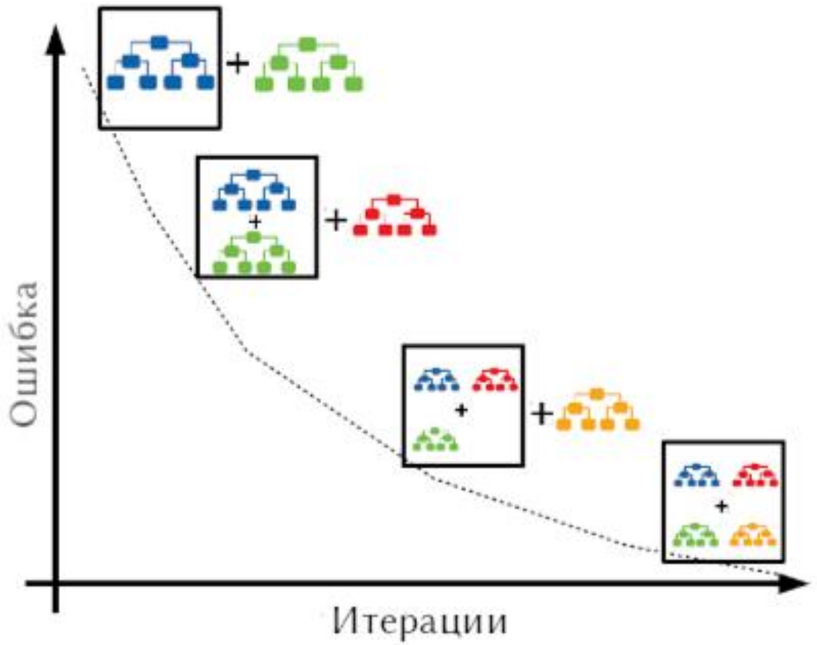
- Пример: Классификация клиентов в маркетинге.



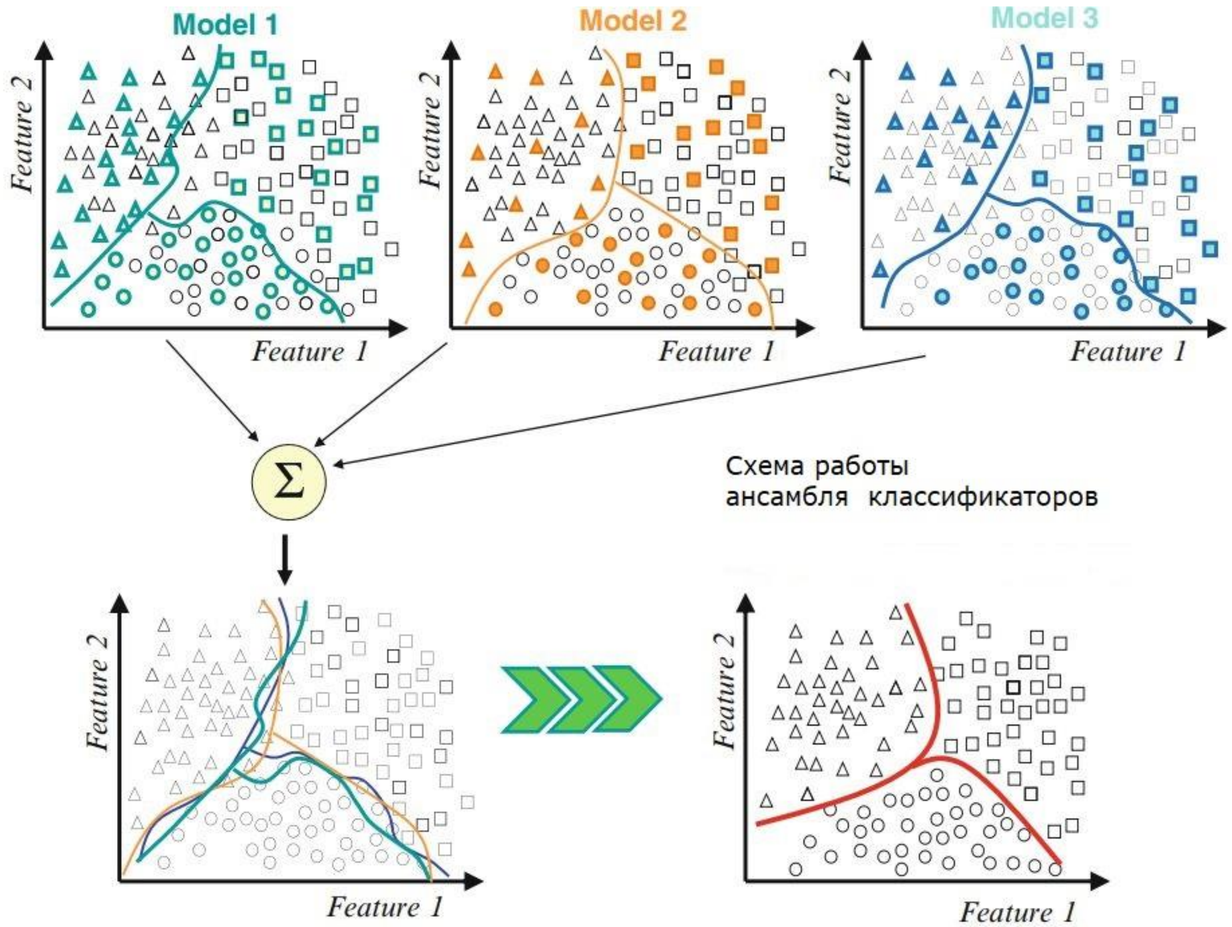
Плюсы и минусы:
(+) Высокая предсказательная способность
(-) Требует тщательной настройки гиперпараметров

Градиентный бустинг (Gradient Boosting)

- **Описание**: Градиентный бустинг — это еще один ансамблевый метод, который строит модель итеративно, исправляя ошибки предыдущих моделей.



Градиентный бустинг (Gradient Boosting)



Градиентный бустинг (Gradient Boosting)

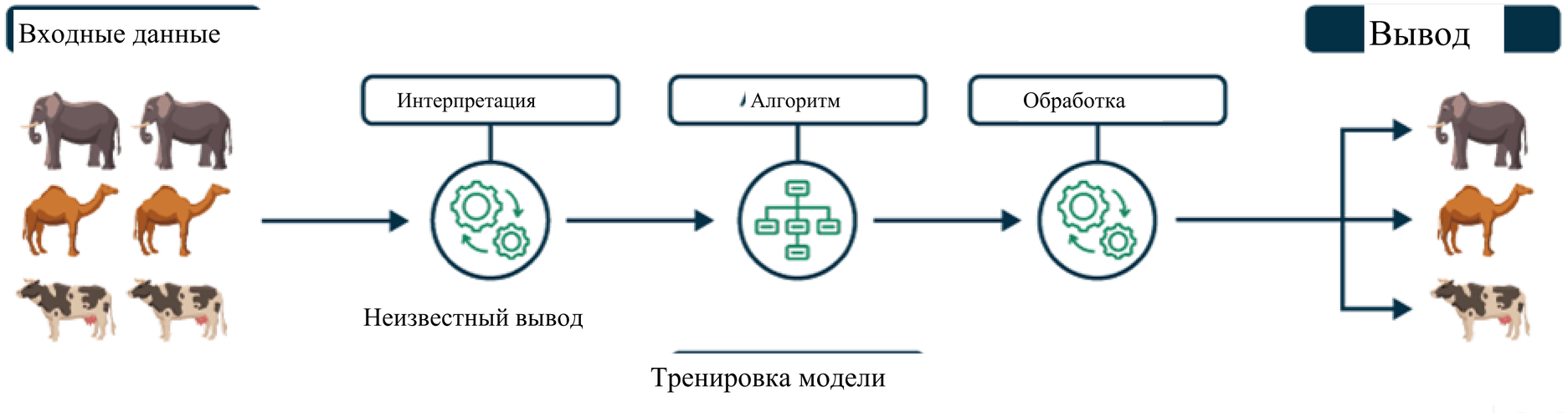
Пример на Python

```
from sklearn.ensemble import GradientBoostingClassifier


# Обучение модели
model = GradientBoostingClassifier()
model.fit(X_train, y_train)


# Оценка точности
accuracy = model.score(X_test, y_test)
print(accuracy)
```

2 Неконтролируемое обучение

Определение

- **Неконтролируемое обучение** – это метод, при котором модель обучается на неразмеченных данных, пытаясь выявить скрытые структуры и зависимости.



Примеры задач

- **Кластеризация**: группировка данных по сходству (например, сегментация клиентов в маркетинге, анализ ДНК).
- **Снижение размерности**: упрощение данных для визуализации или подготовки к последующим моделям (например, PCA, t-SNE).
- **Выявление аномалий**: поиск отклонений в данных (например, обнаружение мошенничества в банковских транзакциях).

Алгоритмы Неконтролируемого обучения

- K-means
- DBSCAN
- Иерархическая кластеризация
- Principal Component Analysis (PCA)

Преимущества и недостатки

Неконтролируемое обучение

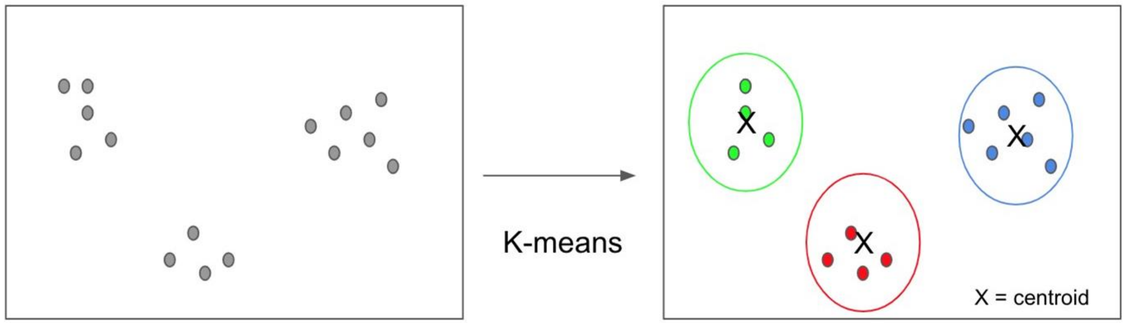
Преимущества	Недостатки
Не требует разметки данных	Сложность интерпретации результатов
Может выявлять скрытые закономерности	Менее точные прогнозы по сравнению с контролируемым обучением
Полезно для исследования данных	Алгоритмы чувствительны к выбору параметров

Контролируемое vs. Неконтролируемое обучение

Фактор	Контролируемое обучение	Неконтролируемое обучение
Тип данных	Размеченные	Неразмеченные
Цель	Предсказание выходного значения	Поиск закономерностей
Примеры задач	Классификация, регрессия	Кластеризация, снижение размерности
Требования к данным	Наличие разметки	Разметка не требуется

Кластеризация К-средних (K-means)

- **Описание:** Группирует данные в K кластеров на основе их схожести.
- **Пример:** Сегментация клиентов по покупательскому поведению.



- **Плюсы и минусы:**

(+) Простота

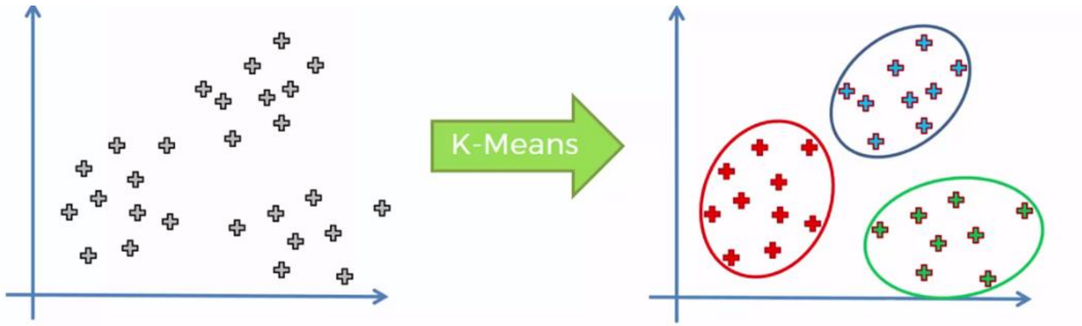
(-) Чувствительность к начальному положению кластеров

Кластеризация К-средних (K-means)

- **Кластеризацию** можно рассматривать как **классификацию без размеченных данных** (без учителя), то есть алгоритм должен классифицировать данные сам, без маркировки.
- Ученик должен сам отнести объекты к определенным классам, но ученик не знает, какие это могут быть классы, поэтому **нужно просто сказать ему, на сколько классов нужно разделить данные объекты** (на k классов).
- Затем ученик начинает разделять объекты по их похожести, на k классов, то есть **близко похожие друг на друга объекты ученик относит к одному классу.**

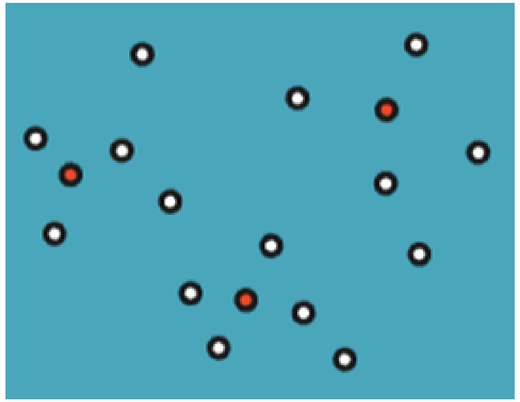
До K-means

После K-means

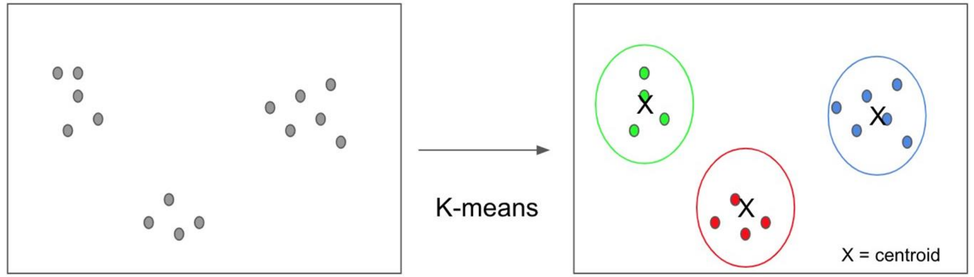


Кластеризация К-средних (K-means)

- Выбираются **k случайных центров кластеров** для набора данных (выбор случайным образом или используя специальные методы инициализации).
- **Каждая точка данных присваивается к ближайшему центру** кластера.
- **Центры обновляются как среднее значение всех точек**, принадлежащих каждому кластеру.
- **Этот шаг повторяется до тех пор, пока центры не перестанут меняться** или до достижения заданного критерия сходимости.

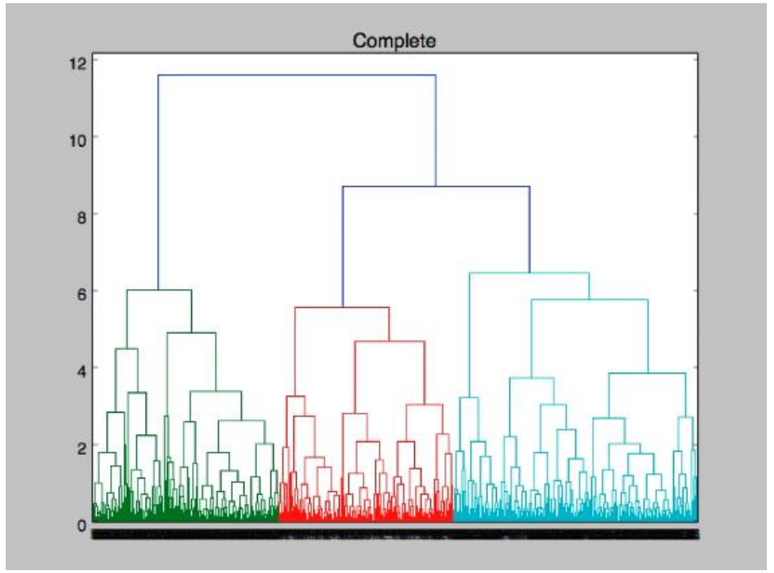


$$d(a,b) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$$



Иерархическая кластеризация

- **Описание:** Создает древовидную структуру кластеров.
- **Пример:** Классификация биологических видов.



- **Плюсы и минусы:**

  (+) Не требует заранее задавать число кластеров

  (-) Высокая вычислительная сложность

Иерархическая кластеризация

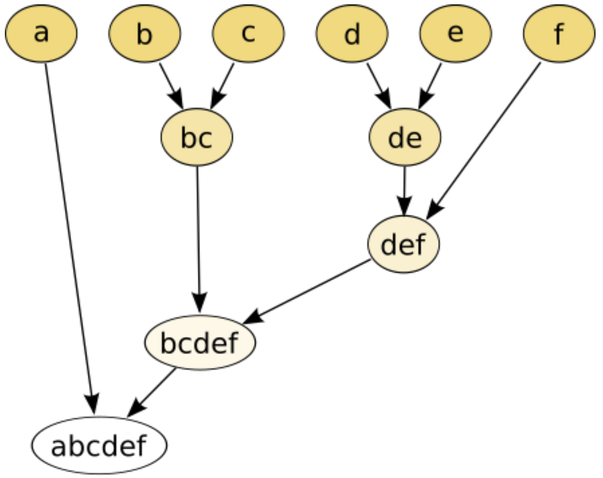
1. Предположим, у нас ряд точек данных a, b, c, d, e и f. Сгруппируем их согласно «жадной стратегии».

2. Мы видим, что b и c, а также d и e находятся ближе всего друг к другу, а потому объединяем их в первую очередь.

3. Далее мы видим, что f ближе всего к кластеру de, поэтому объединяем их всех в кластер def.

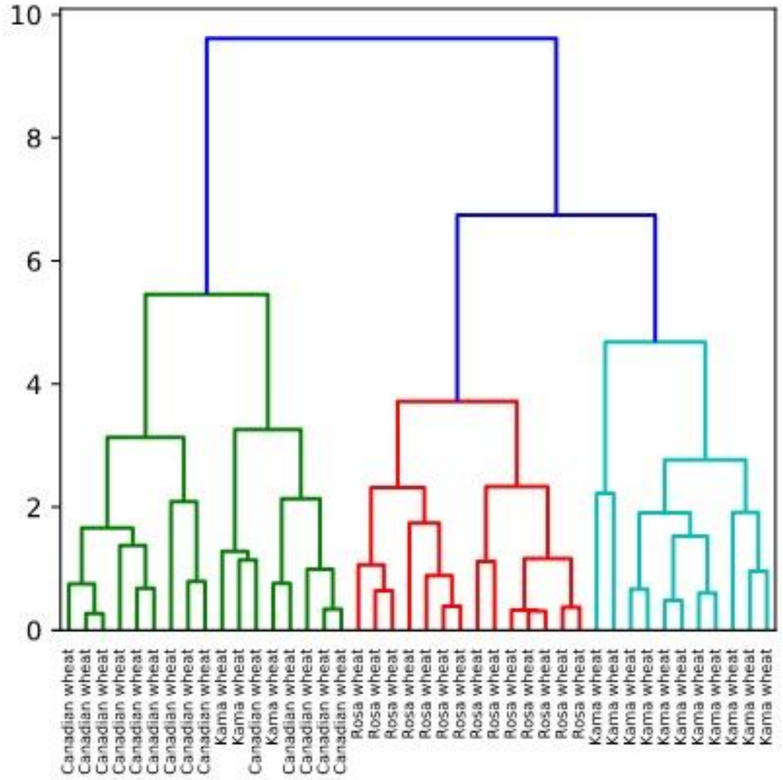
4. Далее, ближайшим к кластеру def является кластер bc, поэтому объединяем их в $bcdef$.

5. И, наконец, объединяем этот крупный кластер с оставшимся в одиночестве а в один большой кластер $abcdef$.

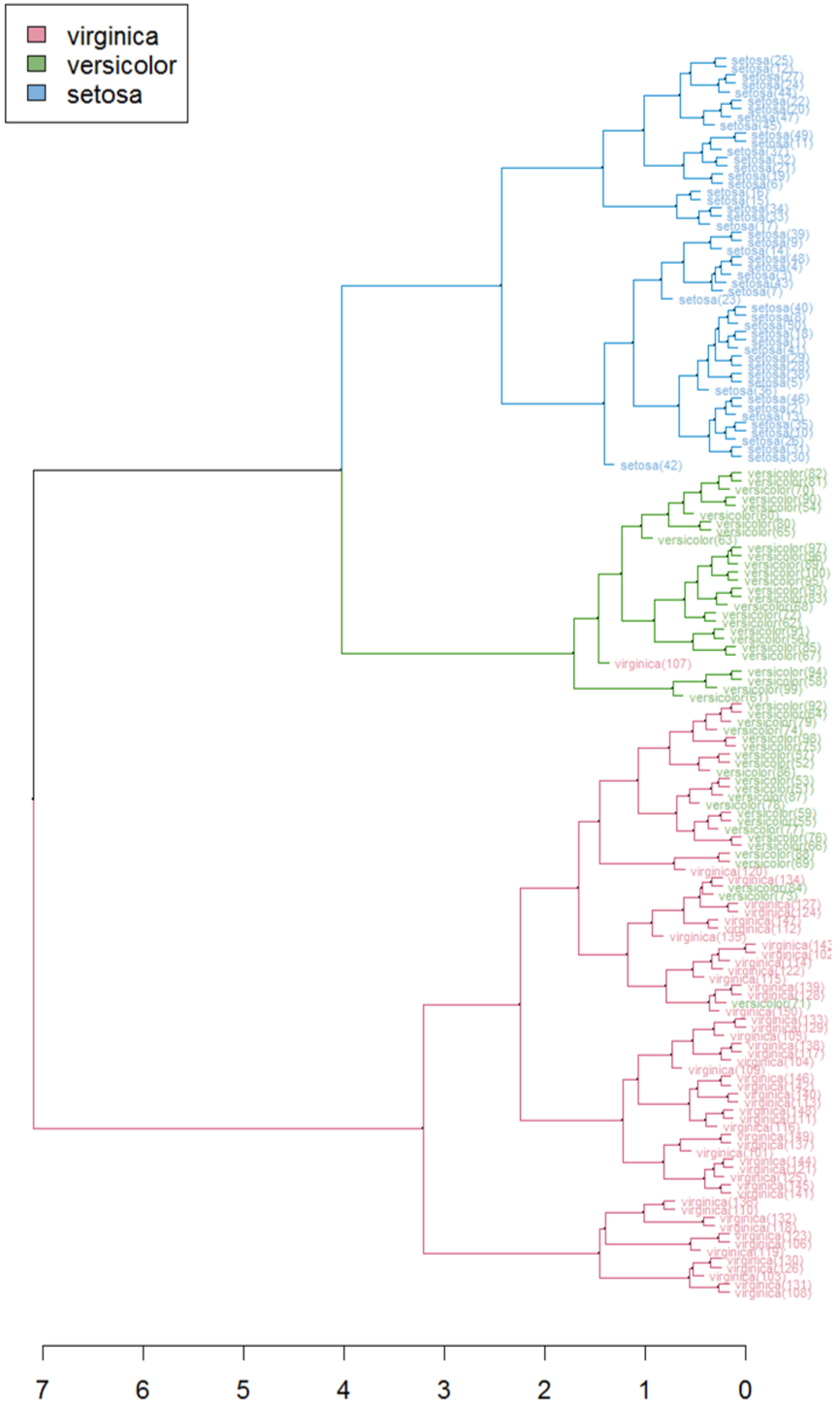


Иерархическая кластеризация

- График кластеризации – **дендрограмма**.
- Если вы наткнётесь на большой пропуск в дендрограмме, то будете знать, что два данных кластера находятся далеко друг от друга, а потому можно считать их отдельными.
- Некоторые популярные показатели расстояния: евклидово расстояние, квадрат евклидова расстояния, манхэттенская метрика, максимальное расстояние и др.



Иерархическая кластеризация: Кластеризация набора данных Iris



Метод Главных Компонент (PCA)

- **Описание:** Метод уменьшения размерности данных.
- **Пример:** Сжатие изображений.

- **Плюсы и минусы:**

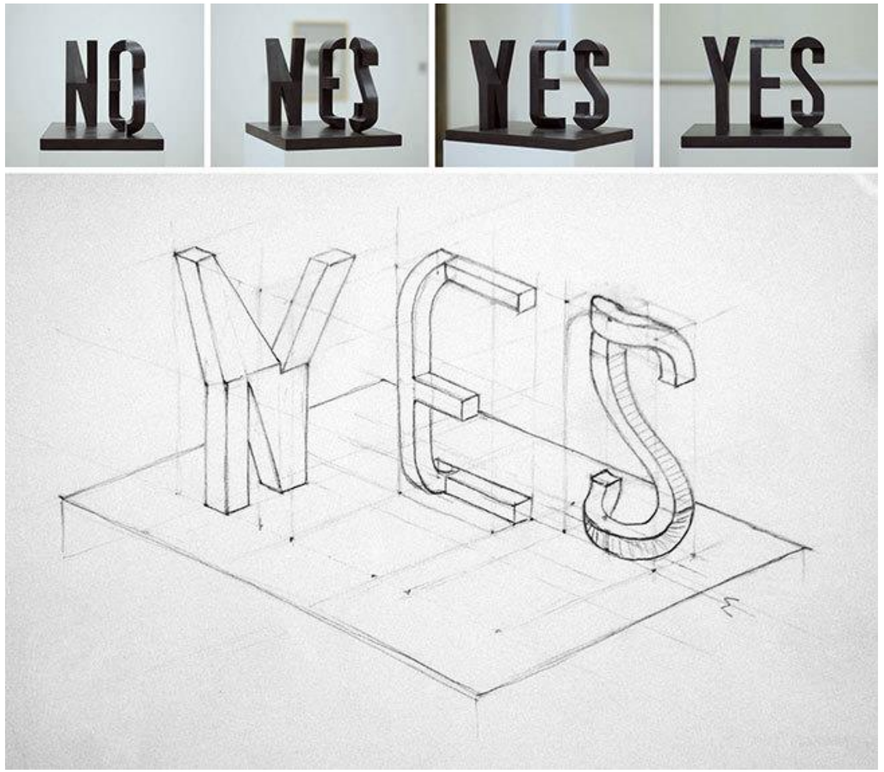
  (+) Уменьшает количество признаков

  (-) Потеря информации

Метод Главных Компонент (PCA)

- Метод главных компонент (Principal Component Analysis или же PCA) — алгоритм обучения без учителя, используемый для понижения размерности и выявления наиболее информативных признаков в данных.

- Его суть заключается в предположении о линейности отношений данных и их проекции на подпространство ортогональных векторов, в которых дисперсия будет максимальной.



Ассоциативные правила

- **Описание:** Выявляет закономерности между переменными в больших наборах данных.
- **Пример:** Анализ покупательских корзин в супермаркете.

- **Плюсы и минусы:**

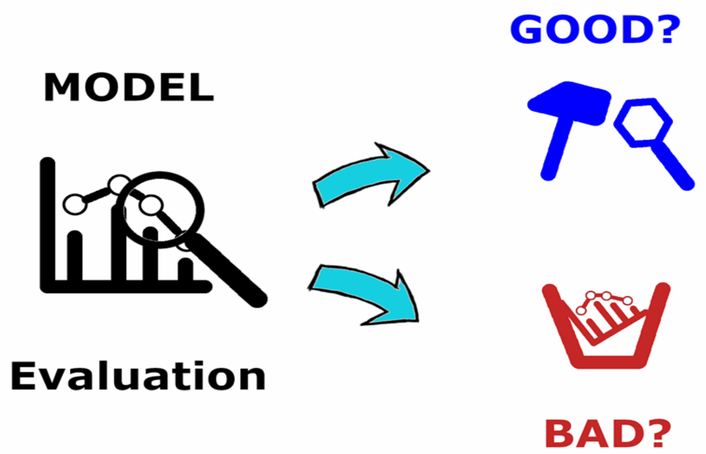
  (+) Полезен для рекомендаций

  (-) Требует большого количества данных

Ключевые показатели эффективности моделей машинного обучения (классификаторы)

Зачем оценивать модели машинного обучения?

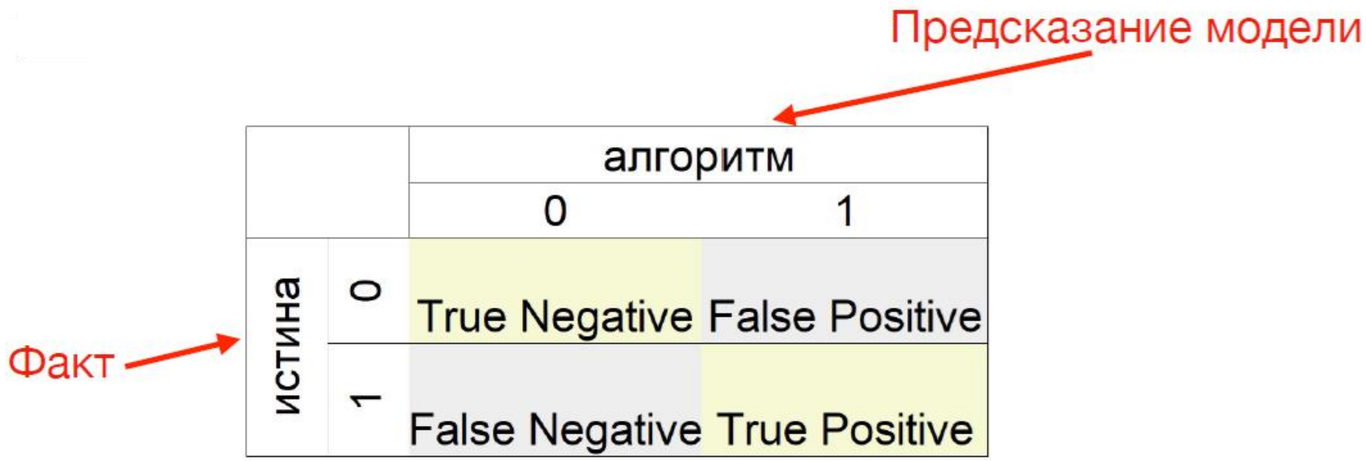
- После обучения модели машинного обучения важно понять, насколько хорошо она работает на новых, ранее невиданных данных.
- Ключевые показатели эффективности (метрики) помогают **количественно оценить качество прогнозов модели.**
- Выбор правильной метрики зависит от задачи машинного обучения и бизнес-целей.
- Сегодня мы рассмотрим три фундаментальные метрики для задач классификации: Accuracy, Precision и Recall.



Матрица ошибок (Confusion Matrix)

Основа для оценки

- **Матрица ошибок** - это таблица, которая показывает результаты работы классификатора.
- Она сравнивает фактические классы с классами, предсказанными моделью.



Матрица ошибок (Confusion Matrix)

- **TP (True Positive/Истинно Положительный):** Модель правильно предсказала положительный класс.
- • **TN (True Negative/Истинно Отрицательный):** Модель правильно предсказала отрицательный класс.
- • **FP (False Positive/Ложно Положительный):** Модель ошибочно предсказала положительный класс (ошибка первого рода).
- • **FN (False Negative/Ложно Отрицательный):** Модель ошибочно предсказала отрицательный класс (ошибка второго рода).

		алгоритм	
		0	1
истина	0	True Negative	False Positive
	1	False Negative	True Positive

Accuracy (Точность)

Общая доля правильных предсказаний

- **Accuracy** показывает, какая доля всех предсказаний модели является правильной.
- • Рассчитывается как отношение количества правильных предсказаний (TP + TN) к общему количеству предсказаний (TP + TN + FP + FN).
- • **Формула:**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy (Точность)

Пример

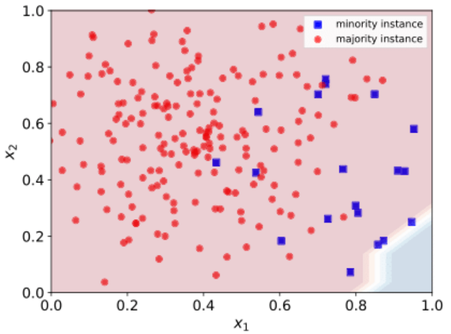
- Если из 100 примеров модель правильно классифицировала 90, то Accuracy = 90/100 = 0.9 или 90%.
- **Важно:** Accuracy может быть вводящим в заблуждение при несбалансированных классах.

Проблема несбалансированных классов (Accuracy)

- Представьте задачу классификации, где 95% объектов принадлежат к классу "А", а 5% - к классу “Б".

- • Модель, которая всегда предсказывает класс "А", достигнет Accuracy 95%, но будет абсолютно бесполезна для обнаружения объектов класса “Б".

- • В таких случаях Accuracy *не является надежной метрикой.*



Precision (Точность)

Точность положительных предсказаний

- • **Precision** отвечает на вопрос: "Из всех объектов, которые модель предсказала как положительные, какая доля действительно является положительными?”
- • Рассчитывается как отношение количества истинно положительных предсказаний (TP) к общему количеству положительных предсказаний (TP + FP).
- **Формула:**

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision (Точность)

Пример

- Если модель предсказала 10 объектов как "больные", и 8 из них действительно оказались больными, то Precision = 8/10 = 0.8 или 80%.•

- **Важно:** Высокий Precision означает, что модель редко ошибочно помечает отрицательные объекты как положительные.

Recall (Полнота)

Доля обнаруженных положительных объектов

- • Recall отвечает на вопрос: "Из всех фактических положительных объектов, какая доля была правильно обнаружена моделью?”
- • Рассчитывается как отношение количества истинно положительных предсказаний (TP) к общему количеству фактических положительных объектов (TP + FN).
- • **Формула:**

$$\text{Recall} = \frac{TP}{TP + FN}$$

Recall (Полнота)

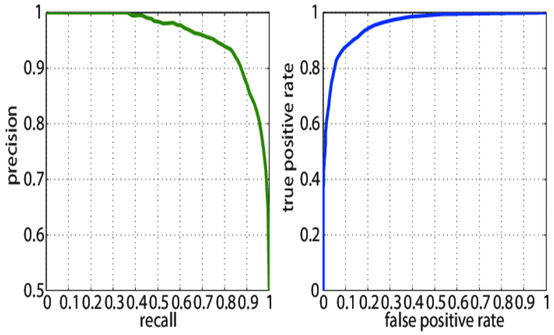
Пример

- Если в наборе данных 10 больных пациентов, и модель правильно выявила 7 из них, то Recall = 7/10 = 0.7 или 70%.
- **Важно:** Высокий Recall означает, что модель хорошо обнаруживает все положительные объекты и редко пропускает их.

Precision против Recall: Баланс

Две стороны одной медали

- Precision и Recall часто находятся в обратной зависимости.
- Повышение Precision может привести к снижению Recall и наоборот.• Выбор между Precision и Recall зависит от конкретной задачи.
- • **Примеры:**
    - **Высокий Precision важен:** Спам-фильтр (лучше пропустить несколько спам-писем, чем пометить важное письмо как спам).
    - **Высокий Recall важен:** Медицинская диагностика (лучше перестраховаться и отправить здорового человека на дополнительное обследование, чем пропустить больного).



F1-мера

Гармоническое среднее Precision и Recall

- • F1-мера - это взвешенное среднее Precision и Recall.
- • Она стремится найти баланс между Precision и Recall.
- • Особенно полезна при несбалансированных классах.
- • **Формула:**

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

F1-мера

Интерпретация

- • Значение F1-меры находится в диапазоне от 0 до 1.
- • Более высокое значение F1-меры указывает на более сбалансированную модель

Метрики в задачах регрессии

Основные метрики

- • **Mean Absolute Error (MAE)**: Среднее абсолютное отклонение.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- • **Mean Squared Error (MSE)**: Среднее квадратическое отклонение.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Метрики в задачах регрессии

Основные метрики

- **Root Mean Squared Error (RMSE)**: Квадратный корень из MSE (в тех же единицах, что и целевая переменная).

$$RMSE = \sqrt{MSE}$$

- • **Coefficient of Determination (R2)**: Доля объясненной дисперсии (от 0 до 1).

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Заключение

- Accuracy, Precision и Recall - важные метрики для оценки моделей классификации.
- • Accuracy показывает общую долю правильных предсказаний, но может быть ненадежной при несбалансированных классах.
- • Precision измеряет точность положительных предсказаний.
- • Recall измеряет полноту обнаружения фактических положительных объектов.
- • F1-мера обеспечивает сбалансированную оценку, учитывая и Precision, и Recall.
- • Выбор метрики зависит от конкретной задачи и бизнес-целей.

Спасибо за внимание!