

---

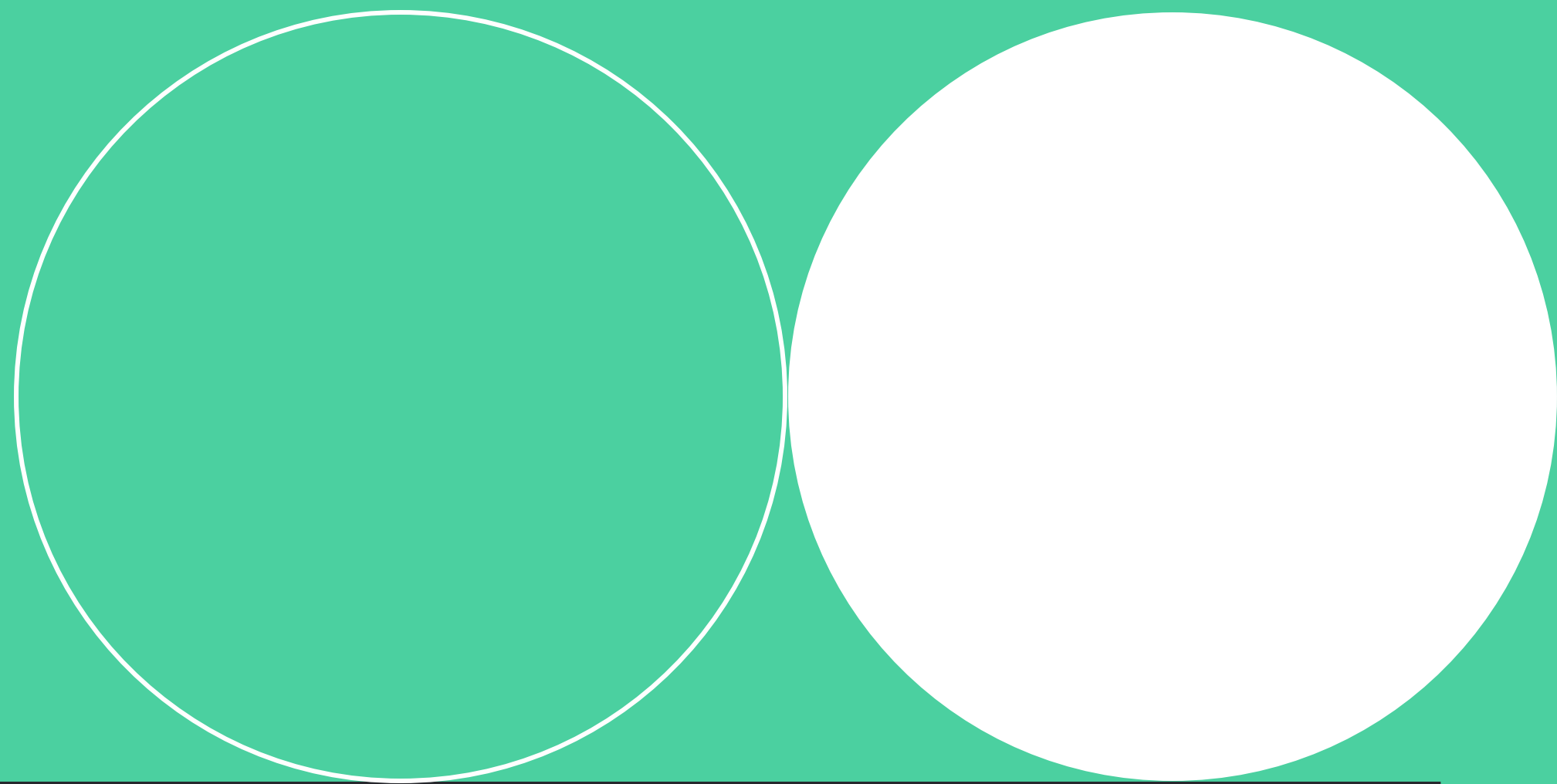
# Линейные модели. Линейная, полиномиальная и логарифмическая регрессия

---



---

# Цели занятия



- Основные задачи машинного обучения
- Линейные модели. Линейная регрессия.
- Sklearn
- Практика

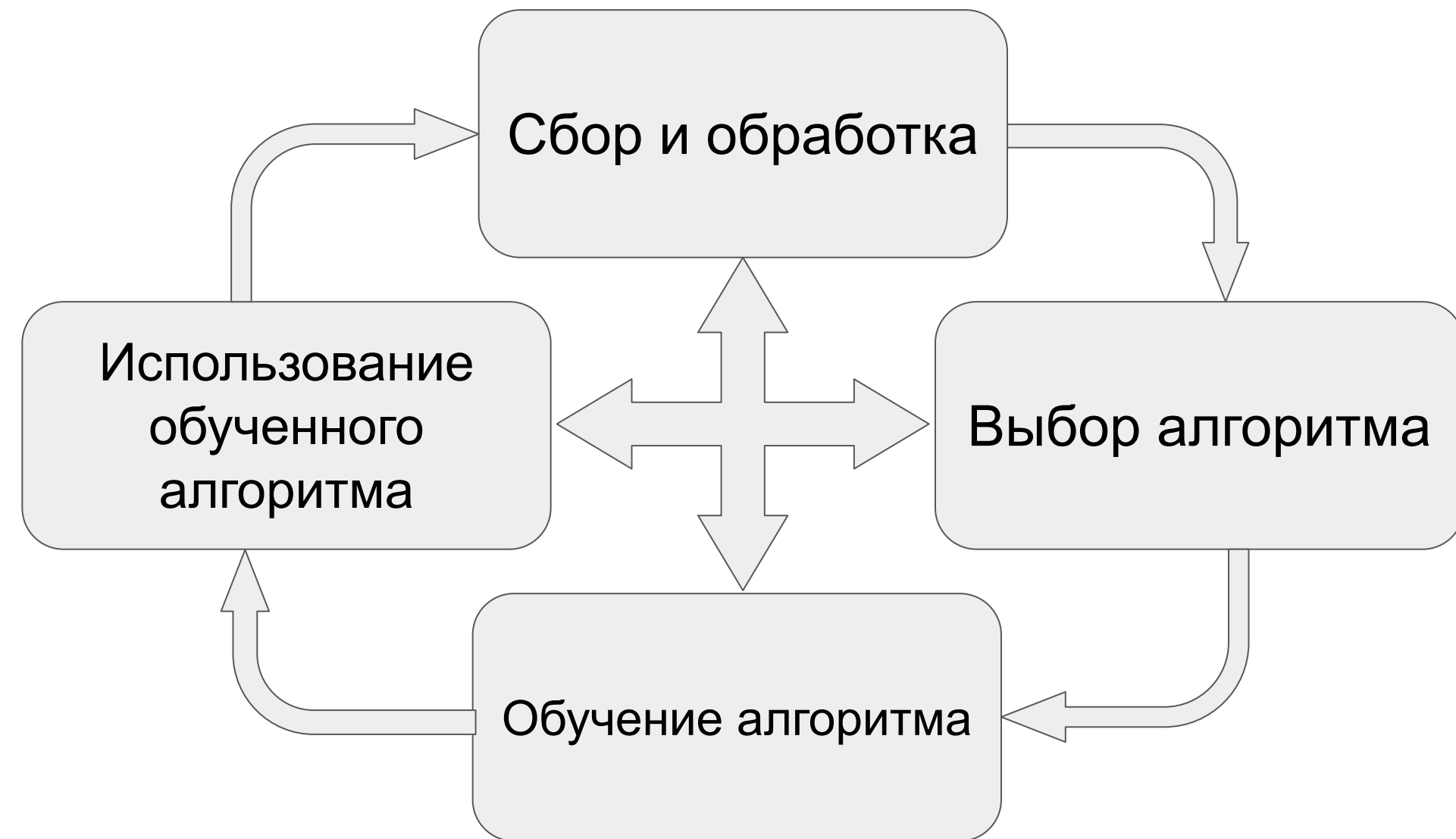


# Задачи машинного обучения

Машинное обучение - использование специальных алгоритмов самостоятельного нахождения решений различных задач путём комплексного использования статистических данных, из которых выводятся закономерности и на основе которых делаются прогнозы.

Процесс:

1. Сбор и обработка данных
2. Выбор алгоритма машинного обучения для поставленной задачи
3. Обучение алгоритма с проверкой качества работы
4. Использование обученного алгоритма с контролем качества



# Решаемые задачи

Задача **регрессии** – задача **предсказания вещественного значения**. Предсказания погоды (температура воздуха, влажность), координаты объекта на картинке, цена продукта, стоимость ценных бумаг, доход магазина.

Задача **классификации** – задача **предсказания категориального ответа (метки класса)** с конечным количеством вариантов. Тип объекта на фотографии, произнесенный звук (распознавание речи), распознавание персоналии по фото, болеет ли человек, фродовое ли объявление на сервисе о продаже.

Задача **кластеризации** – задача **распределение данных на группы**. Разделение всех клиентов мобильного оператора по уровню платёжеспособности, выделение тем в корпусе документов.

Задача **уменьшения размерности** – задача **сведение большого числа признаков к меньшему**. Сжатие информации, повышения качества данных для обучения, отображение информации графически.

Задача **выявления аномалий** – задача **отделение аномалий от стандартных случаев**. Похожа на задачу классификации, но есть одно существенное отличие: аномалии – явление редкое и поэтому мало примеров для обучения. Выявление мошеннических действий с банковскими картами, выявление аномалий в работе приборов и датчиков, аномалии временных рядов.

Задача **ранжирования - сортировка** по большому количеству признаков и по неполным данным. Релевантность поисковой выдачи. Рекомендации товаров в магазинах.



# Машинное обучение можно разделить на несколько основных подходов:

1. **Обучение с учителем** (*supervised learning*) - для обучения алгоритма **есть правильные примеры**. Для обучения сравниваются правильные и предсказанные значения добиваясь минимальной разницы.
  - 1.1. Классификация (*classification*)
  - 1.2. Регрессия (*regression*)
  - 1.3. Ранжирование (*learning to rank*)
2. **Обучение без учителя** (*unsupervised learning*) - **примеров с правильными ответами нет**. Обучение происходит в процессе обработки данных.
  - 2.1. Кластеризация (*clustering*)
  - 2.2. Уменьшение размерности (*dimensionality reduction*)
3. **Обучение с частичным привлечением учителя** (*semi-supervised learning*) - есть некоторое количество примеров с правильными ответами, на которые алгоритм опирается при обработки данных.
  - 3.1. Кластеризация (*clustering*) - даются несколько опорных кластеров (про которые нам точно известно, что они есть)
4. **Обучение с подкреплением** (*reinforcement learning*) - алгоритм обучается получая информацию о качестве решения им задачи, получая награду или штраф за полученное решение.







# Обучение алгоритма/ модели машинного обучения

**Модель** можно представить как **функцию с параметрами**

где  $\theta$  - параметры алгоритма

$\varepsilon$  - неустраняемая ошибка

$$y = f(\theta) + \varepsilon$$

Параметры алгоритма можно разделить на обучаемые (просто **параметры**) и необучаемые (**гиперпараметры**)

$$f(g, w)$$

**Параметры** модели задают **семейство функций**

**Метод максимального правдоподобия** -- метод **поиска** модели, **наилучшим** в каком-то смысле **образом описывающей обучающую выборку**, полученную с некоторым неизвестным распределением.

$$p(y_1, \dots, y_k / x_1, \dots, x_k) = \prod p(y_i / x_i)$$

$p(y_i / x_i)$  — вероятность получить  $y$  при входных данных  $x$  (значения признаков)

Наша задача с помощью различных подходов **выбрать функцию/модель** из всего семейства **добившись максимального правдоподобия/ похожести** реальных данных и данных получаемых с помощью алгоритма.





# Обучение алгоритма/ модели машинного обучения

Для обучения модели определяют:

1. Функцию потерь.
2. Технические метрики. С помощью которых определяют качество обучения модели.
3. Бизнес метрики. С помощью которых определяют бизнес эффект модели.

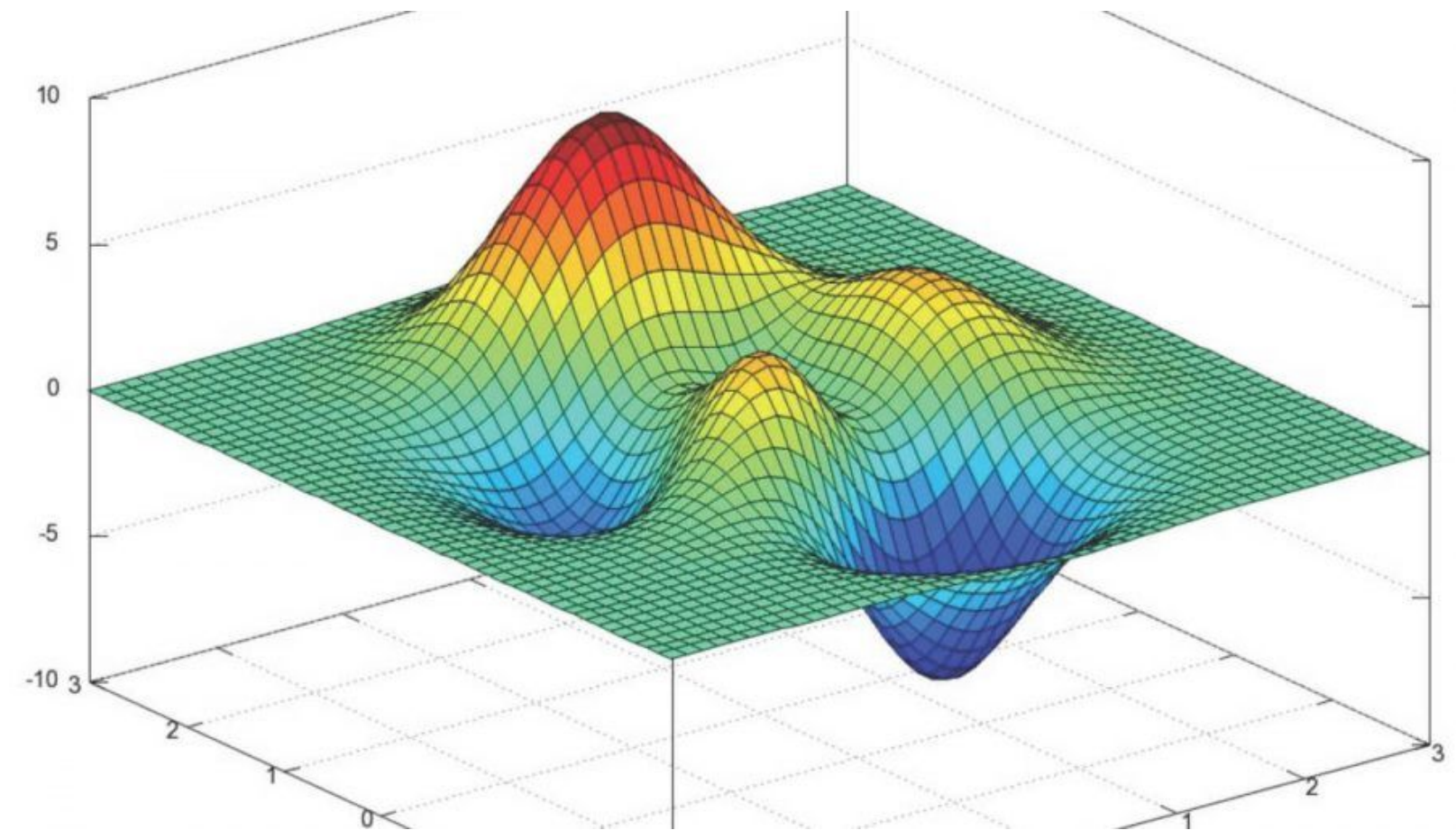
**Функция потерь** — это **мера** количества ошибок, которые наша модель делает на наборе данных.

Функция потерь задает **гиперповерхность в пространстве параметров модели** (пространство функции потерь)

**Минимизируя** функцию потерь мы **получаем** модель, которая **дает** минимальное отклонение предсказанных данных от реальных.

Ошибки которые совершает модель:

1. Ошибки **обучения** (на тренировочных данных)
2. Ошибки **обобщения** (на проверочных данных)



Данные для обучения/тренировки (train)

Данные для проверки (test)



# Линейные модели. Линейная регрессия.

**Линейные модели** - предполагают, что определяемый критерий *линейно зависит* от признаков описывающих объект или процесс.

## Плюсы:

1. Скорость и простота получения модели.
2. Интерпретируемость модели. Линейная модель является прозрачной и понятной для аналитика. По полученным коэффициентам регрессии можно судить о том, как тот или иной фактор влияет на результат, сделать на этой основе дополнительные полезные выводы.
3. Широкая применимость. Большое количество реальных процессов в экономике и бизнесе можно с достаточной точностью описать линейными моделями.
4. Изученность данного подхода. Для линейной регрессии известны типичные проблемы (например, мультиколлинеарность) и их решения, разработаны и реализованы тесты оценки статической значимости получаемых моделей.
- 5.

## Минусы:

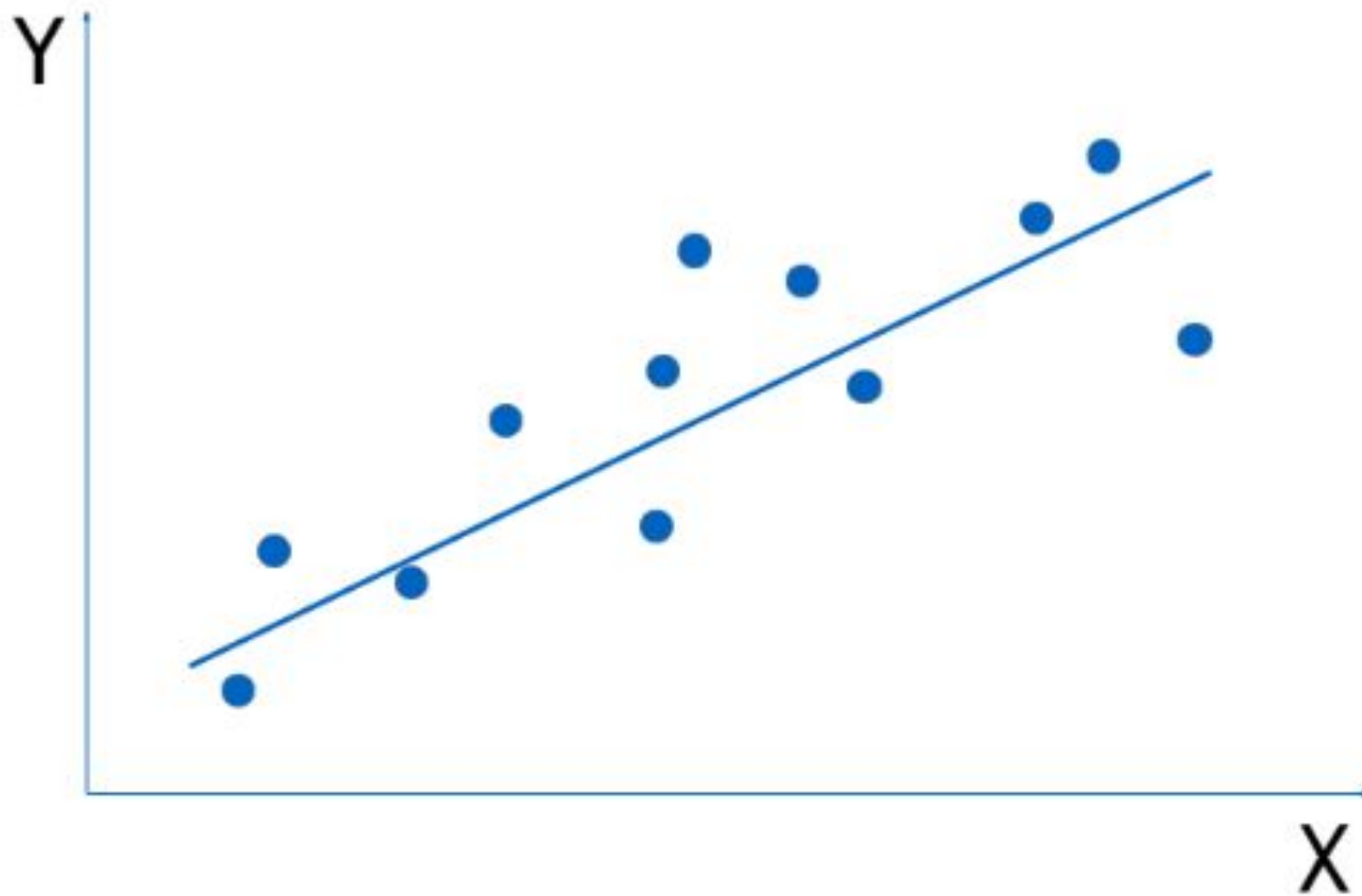
1. Не могут уловить сложные зависимости в данных.
2. Проблемы с мультиколлинеарностью



# Линейная регрессия

Регрессия - уравнение связи искомого критерия с признаками

Линейная регрессия - связь между искомым критерием и параметрами линейная.



$$y_i = \sum_{j=1}^m w_j X_{ij} + e_i$$

$Y$  – целевая переменная

$W$  – вектор весов модели

$X$  – матрица наблюдений

$e$  – ошибка модели



# Линейная регрессия

При обучении мы должны  
максимизировать правдоподобие

$$\hat{w} = \arg \max_w p(\vec{y} \mid X, \vec{w})$$

Идеальная модель

$$y = f(\theta) + \epsilon \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

$$p(y_i \mid X, \vec{w}) = \sum_{j=1}^m w_j X_{ij} + \mathcal{N}(0, \sigma^2) = \mathcal{N}\left(\sum_{j=1}^m w_j X_{ij}, \sigma^2\right)$$

$$\log p(\vec{y} \mid X, \vec{w}) = -\frac{n}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^n \left(y_i - \vec{w}^T \vec{x}_i\right)^2$$

$$\log p(\vec{y} \mid X, \vec{w}) = \arg \max_w -\frac{1}{2\sigma^2} \sum_{i=1}^n \left(y_i - \vec{w}^T \vec{x}_i\right)^2$$

Функция потерь -

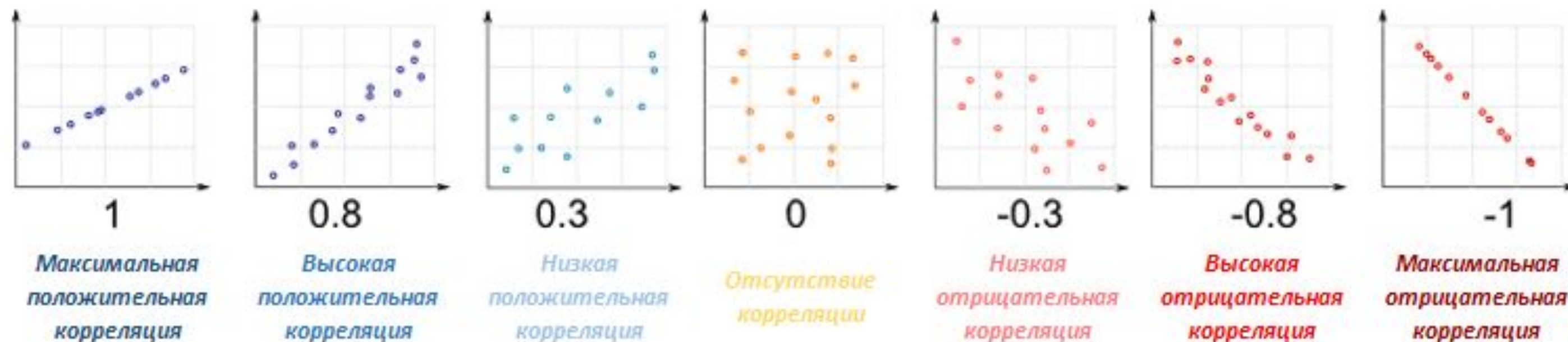
$$\mathcal{L}(X, \vec{y}, \vec{w}) = \frac{1}{2n} \sum_{i=1}^n \left(y_i - \vec{w}^T \vec{x}_i\right)^2$$



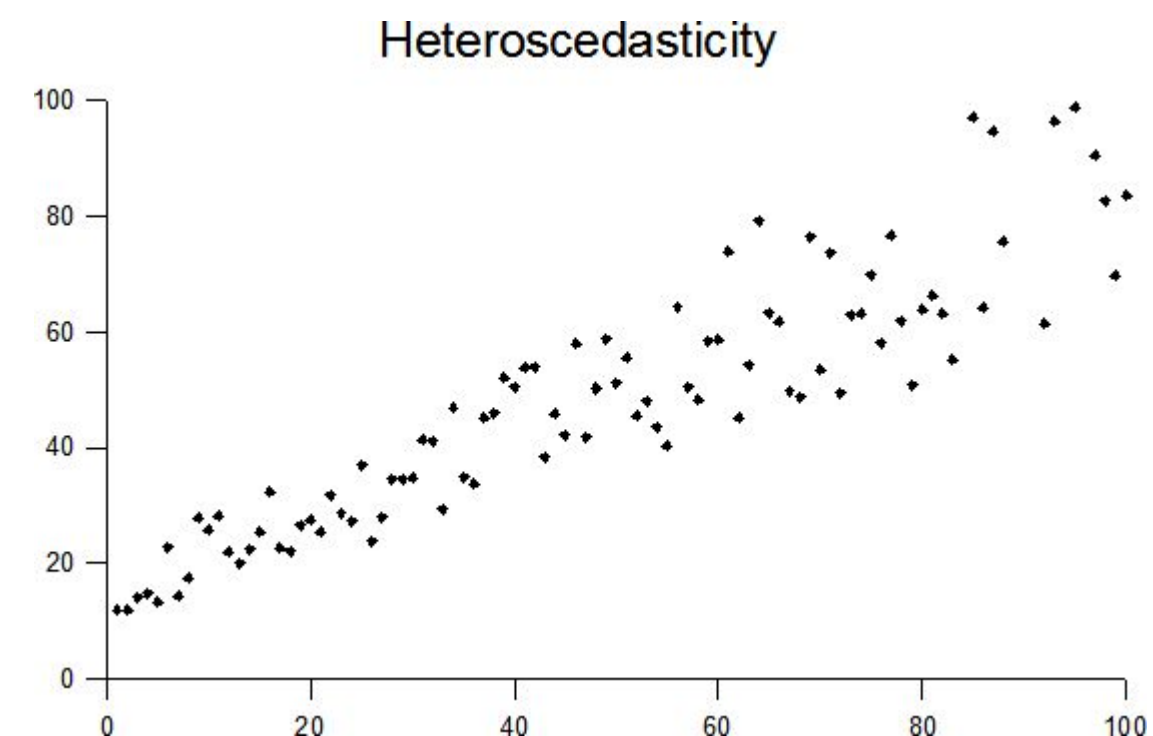
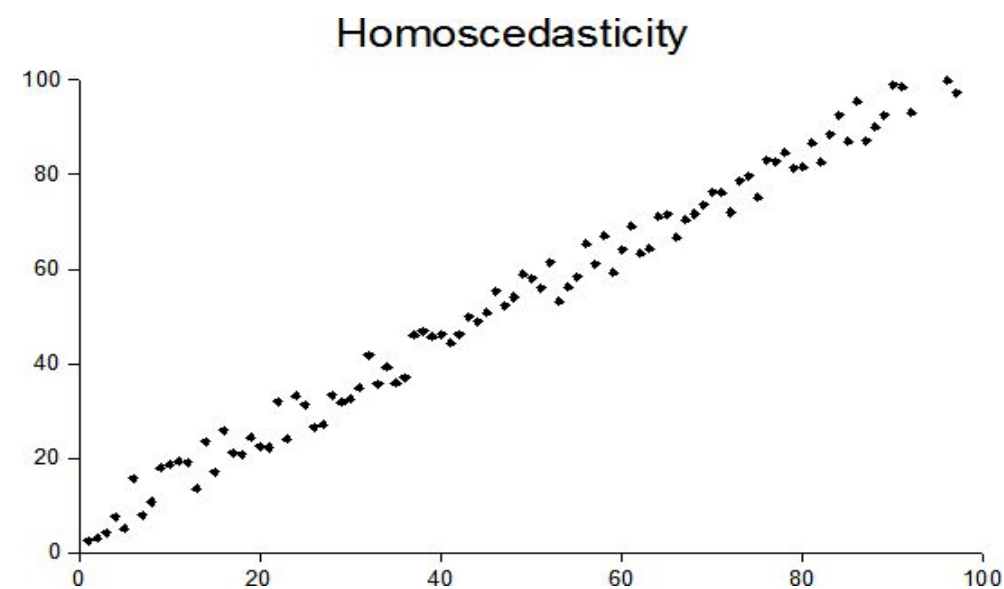


# Требования к данным

1. Независимость признаков (можно бороться с помощью регуляризации)



2. Разброс признаков и определяемого критерия не меняется во времени (условие гомоскедастичностью)



# Виды регрессий

Полиномиальная регрессия

$$f(X) = \sum_{i=1, j=1}^{n, m} w_i x_i^j + b$$

---

Логарифмическая регрессия

$$f(X) = \sum w_i \ln(x_i) + b$$

Экспоненциальная регрессия

$$f(x) = \sum e^{wx} + b$$

Гиперболическая регрессия

$$f(x) = \sum w \cdot 1/x + b$$

Линеаризация - процедура приведения к линейному виду

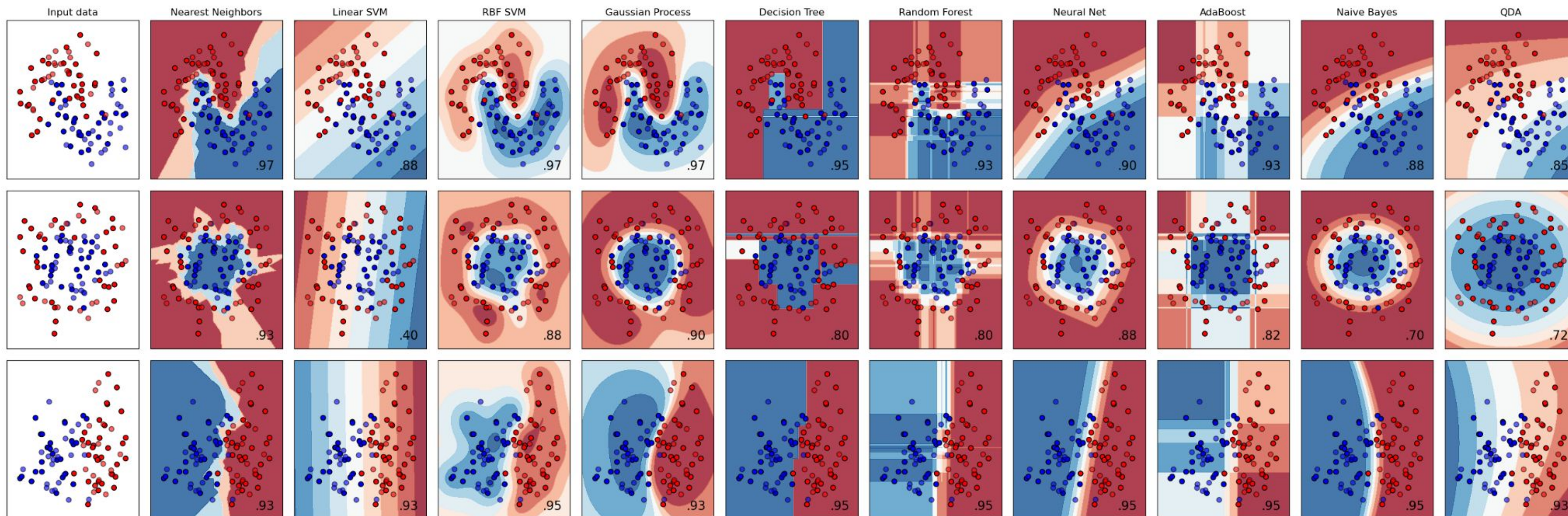




# Scikit-learn

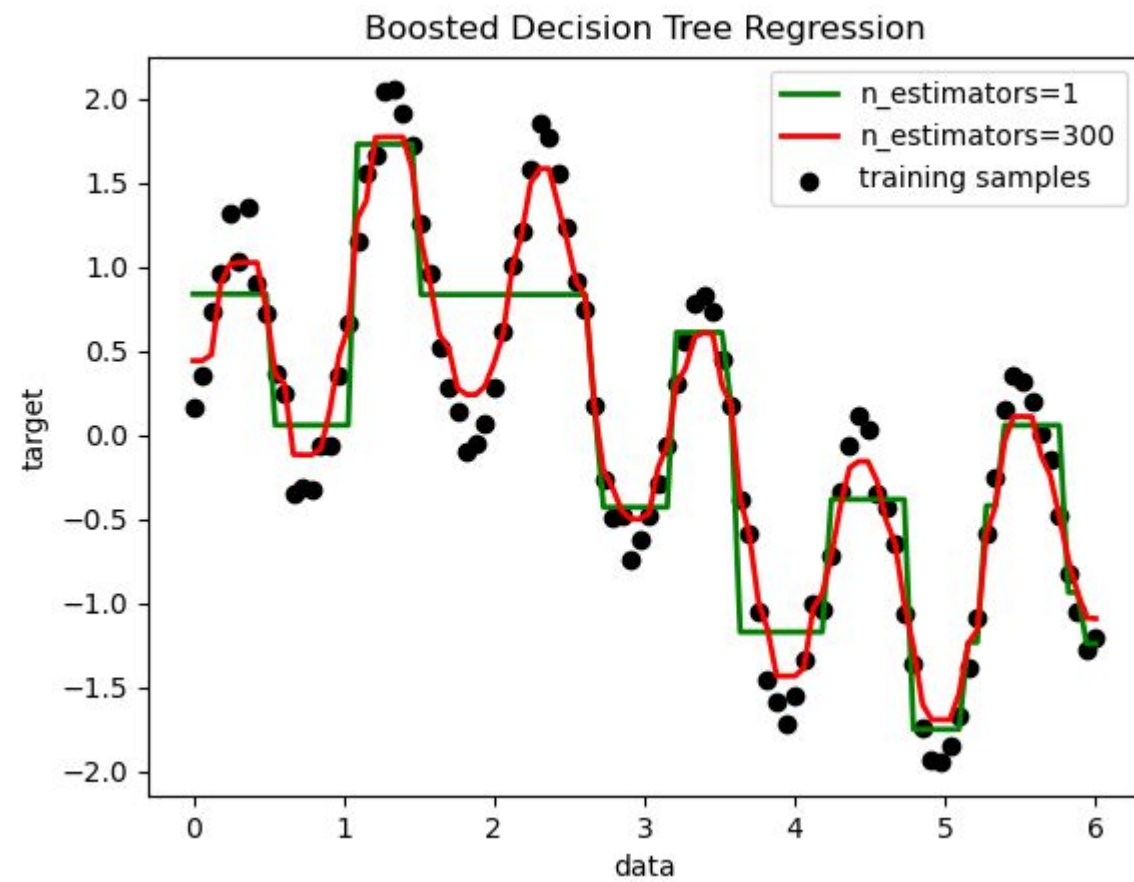
Библиотека Scikit-learn — самый распространенный выбор для решения задач классического машинного обучения. Она предоставляет широкий выбор алгоритмов обучения с учителем и без учителя.

## Классификация



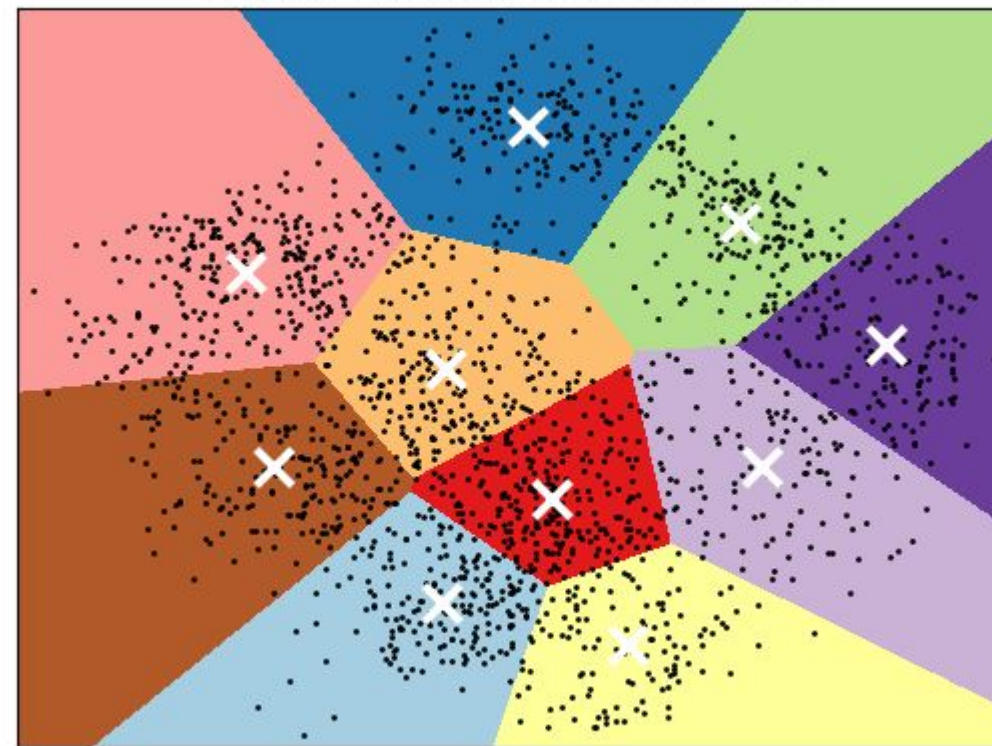


## Регрессия

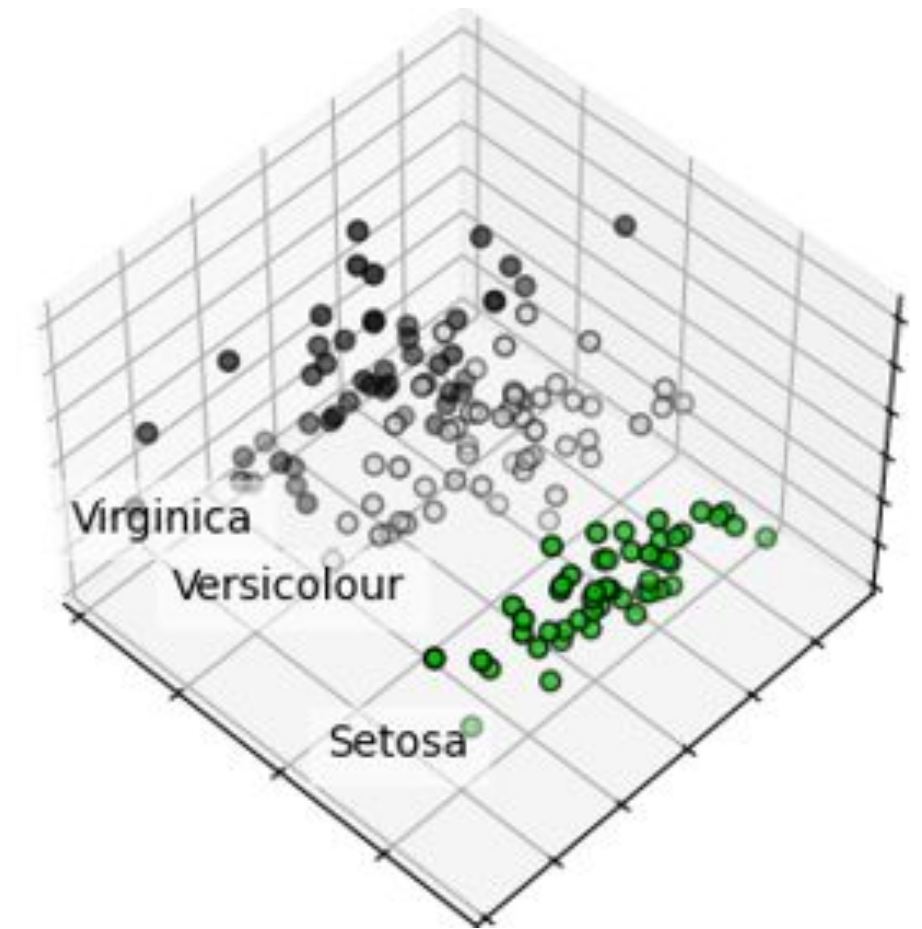


## Кластеризации

K-means clustering on the digits dataset (PCA-reduced data)  
Centroids are marked with white cross



## Понижение размерности



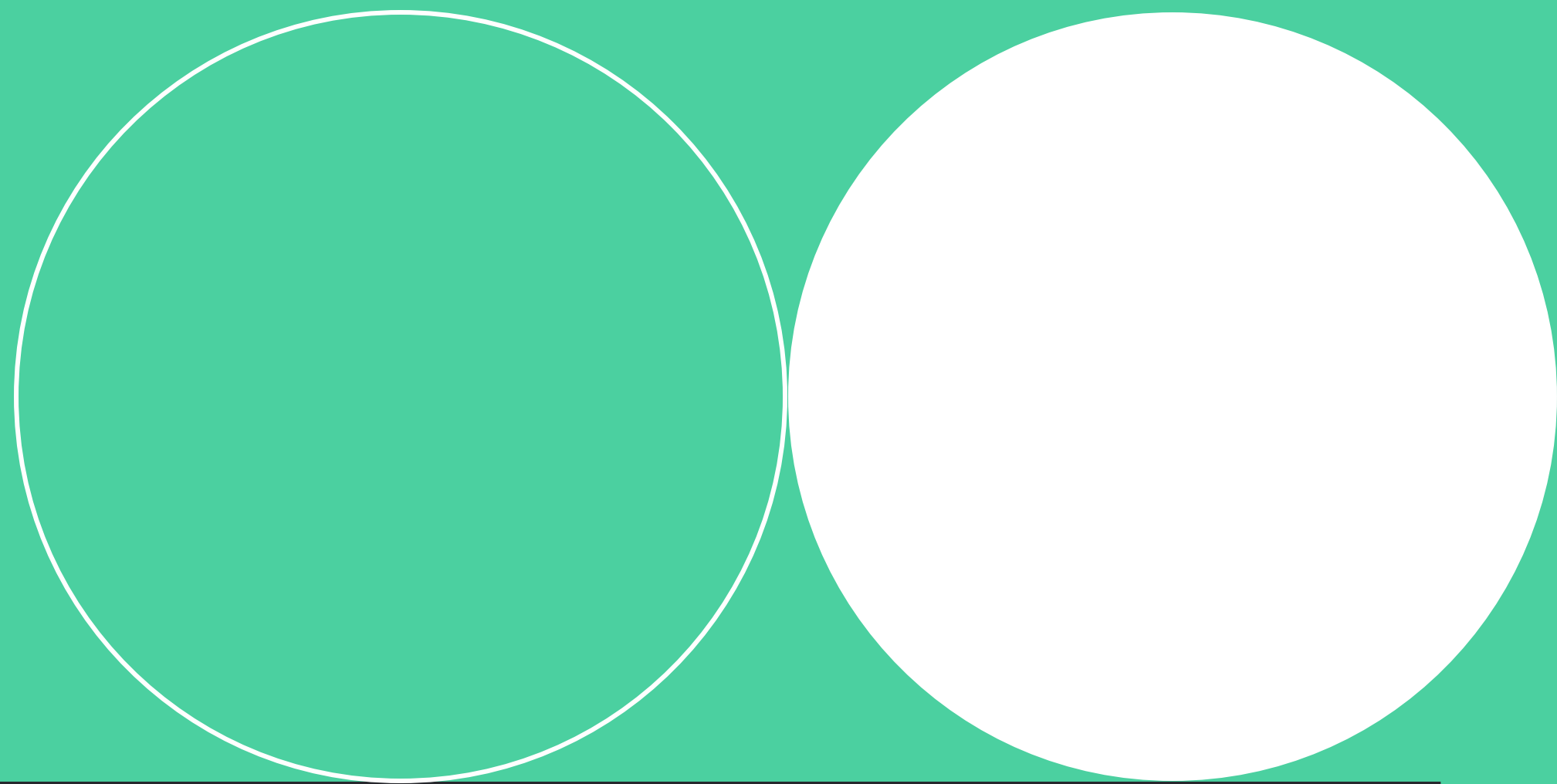
## Применение

```
1 from sklearn.linear_model import LinearRegression
2
3 reg = LinearRegression()
4
5 reg.fit(X_train, y_true)
6
7 y_pred = reg.predict(X_test)
8
```



---

# Практика



---

# Спасибо за внимание!

