



Блок

# FEATURE ENGINEERING



ЕГОР  
САЧКО

Lead Data Scientist

Сбербанк



egsachko@gmail.com



sachkoe



**МАТЕРИАЛЫ ПО БЛОКУ**

## МАТЕРИАЛЫ ПО БЛОКУ

1

**“Learning scikit-learn:  
Machine Learning in Python”**

Raul Garreta,  
Guillermo Moncecchi,  
2013,  
Packt

2

**“Hands-On Machine  
Learning with Scikit-Learn  
and Tensorflow:  
Concepts, Tools and  
Techniques to Build  
Intelligent Systems”**

Geron, A., 2017, O'Reilly Media

3

<https://www.analyticsvidhya.com/blog> - много интересных статей и tutorиалов

4

[blog.kaggle.com/-  
No Free Hunch](https://blog.kaggle.com/-/No-Free-Hunch)



# ЦЕЛИ ЗАНЯТИЯ

# В КОНЦЕ ЗАНЯТИЯ ВЫ СМОЖЕТЕ

# 1

Уменьшать  
размерность  
пространство с  
помощью  
**Lasso**  
регрессии

# 2

Сжимать  
пространство  
признаков с  
помощью **Ridge**  
регрессии

# 3

Использовать  
**метод главных**  
**компонент**

# 4

Использовать  
**sklearn** для  
изменения  
размерности  
пространства  
признаков



**ЧТО БУДЕМ ОБСУЖДАТЬ**

## ПЛАН ЗАНЯТИЯ

---

- 1 Линейная регрессия
- 2 Ridge регрессия
- 3 Lasso регрессия
- 4 Метод главных компонент
- 5 Обсуждение домашнего задания



---

Часть 1-3

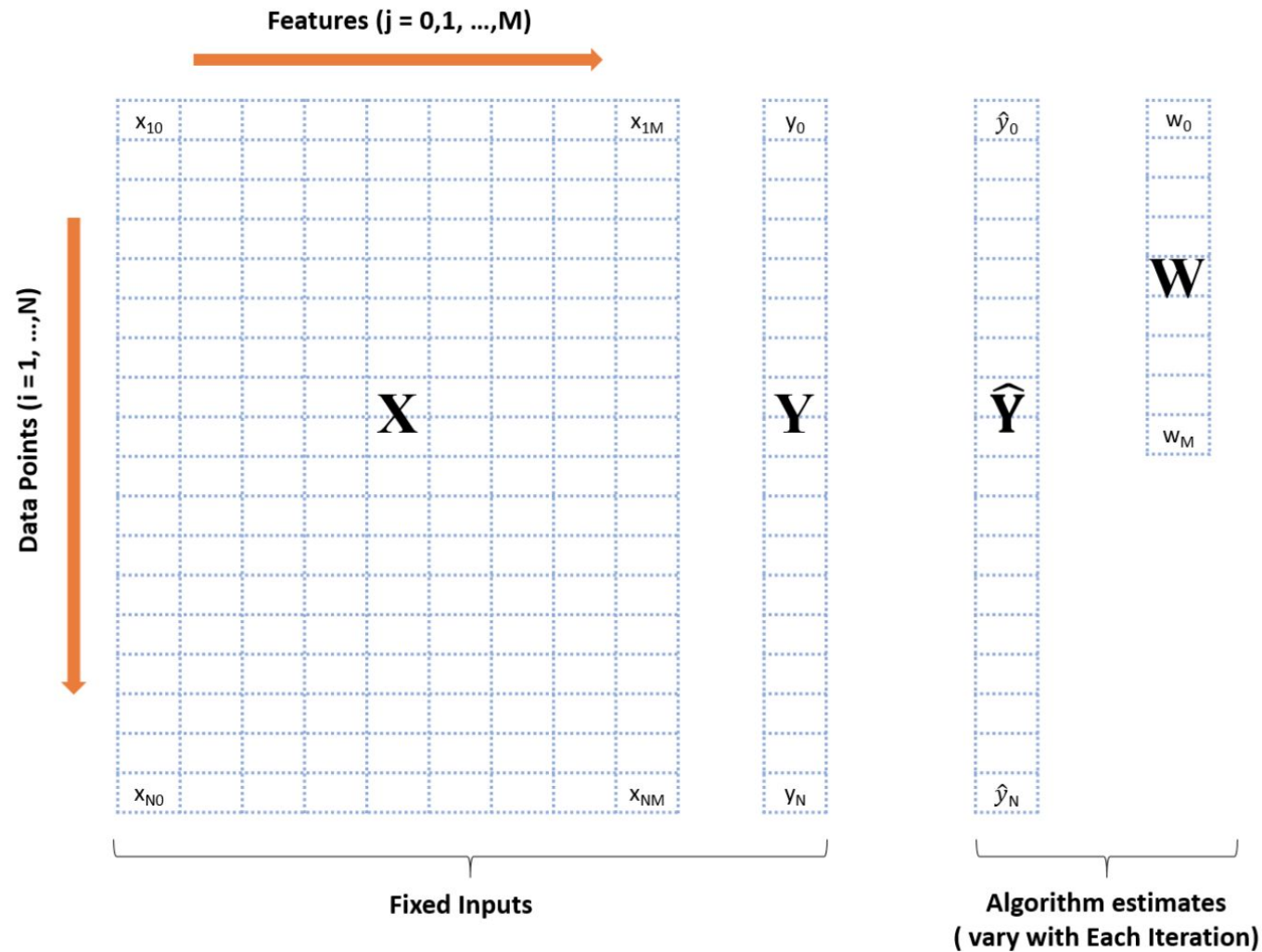
**Линейная регрессия**

**Ridge**

**Lasso**

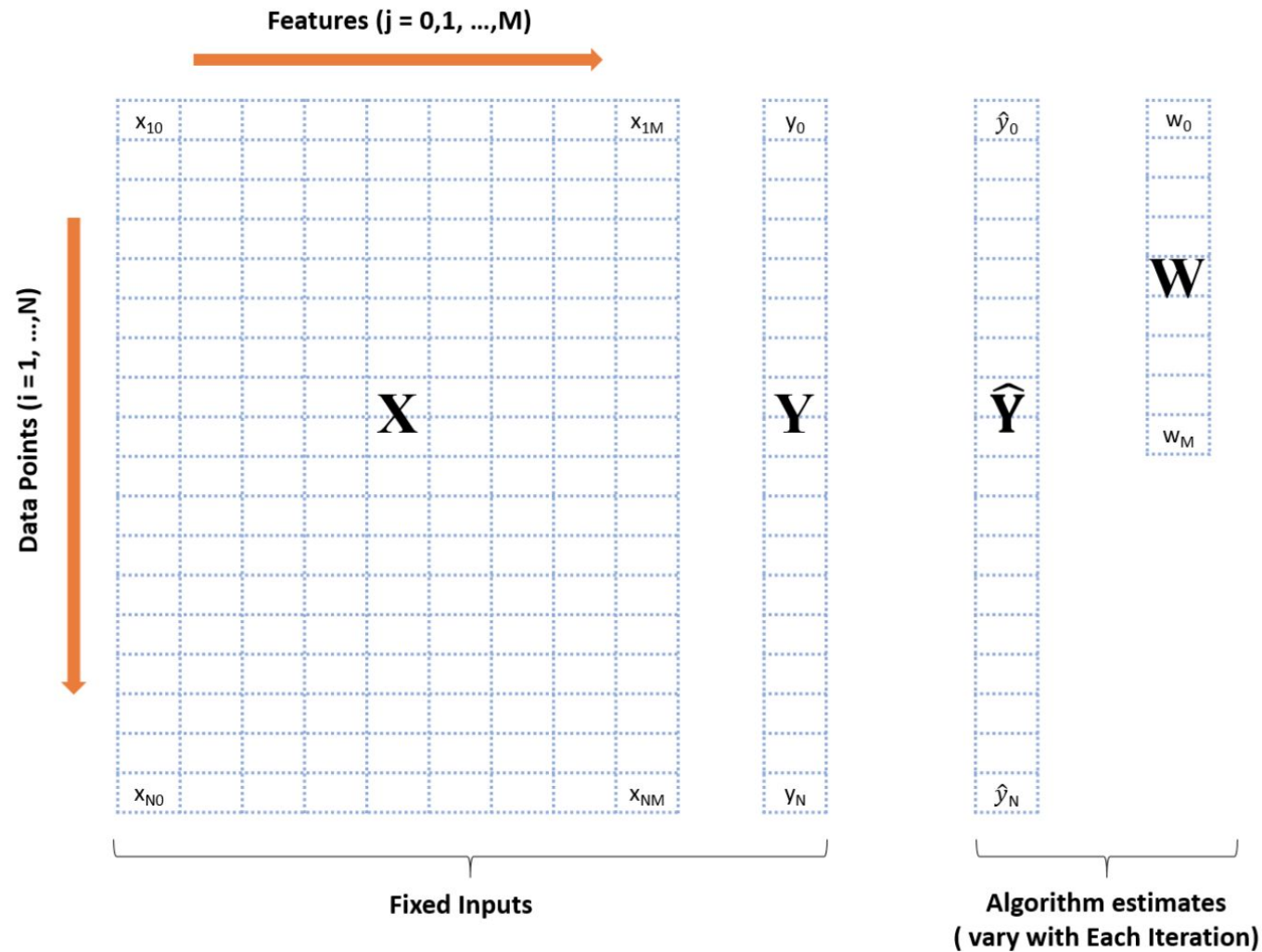
# Линейная регрессия

## Regression Data Representation



# Линейная регрессия

## Regression Data Representation



## Линейная регрессия

$$\hat{y}_i = \sum_{j=0}^M w_j * x_{ij}$$

$$Cost(W) = RSS(W) = \sum_{i=1}^N \{y_i - \hat{y}_i\}^2 = \sum_{i=1}^N \left\{ y_i - \sum_{j=0}^M w_j x_{ij} \right\}^2$$

## Линейная регрессия

$$w_j^{t+1} = w_j^t + 2\eta \sum_{i=1}^N x_{ij} \left\{ y_i - \sum_{k=0}^M w_k x_{ik} \right\}$$

$$\frac{\partial}{\partial w_j} Cost(W) = -2 \sum_{i=1}^N x_{ij} \left\{ y_i - \sum_{k=0}^M w_k x_{ik} \right\}$$

## Ridge регрессия

$$Cost(W) = RSS(W) + \lambda * (\text{sum of squares of weights})$$

$$= \sum_{i=1}^N \left\{ y_i - \sum_{j=0}^M w_j x_{ij} \right\}^2 + \lambda \sum_{j=0}^M w_j^2$$

## Ridge регрессия

$$w_j^{t+1} = w_j^t - \eta \left[ -2 \sum_{i=1}^N x_{ij} \left\{ y_i - \sum_{k=0}^M w_k * x_{ik} \right\} + 2\lambda w_j \right]$$

$$w_j^{t+1} = (1 - 2\lambda\eta)w_j^t + 2\eta \sum_{i=1}^N x_{ij} \left\{ y_i - \sum_{k=0}^M w_k * x_{ik} \right\}$$

$$\frac{\partial}{\partial w_j} Cost(W) = -2 \sum_{i=1}^N x_{ij} \left\{ y_i - \sum_{k=0}^M w_k x_{ik} \right\} + 2\lambda w_j$$

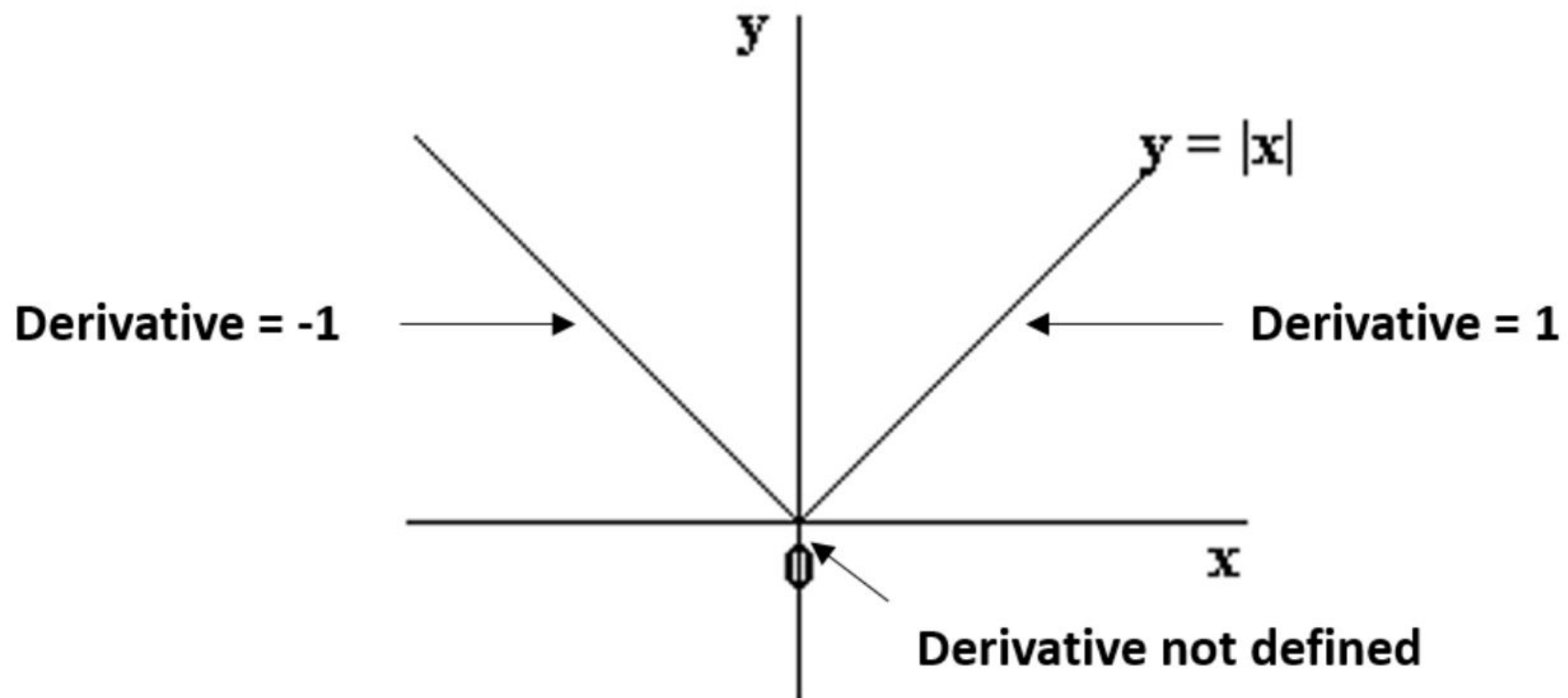
## Lasso регрессия

$$\text{Cost}(W) = \text{RSS}(W) + \lambda * (\text{sum of absolute value of weights})$$

$$= \sum_{i=1}^N \left\{ y_i - \sum_{j=0}^M w_j x_{ij} \right\}^2 + \lambda \sum_{j=0}^M |w_j|$$



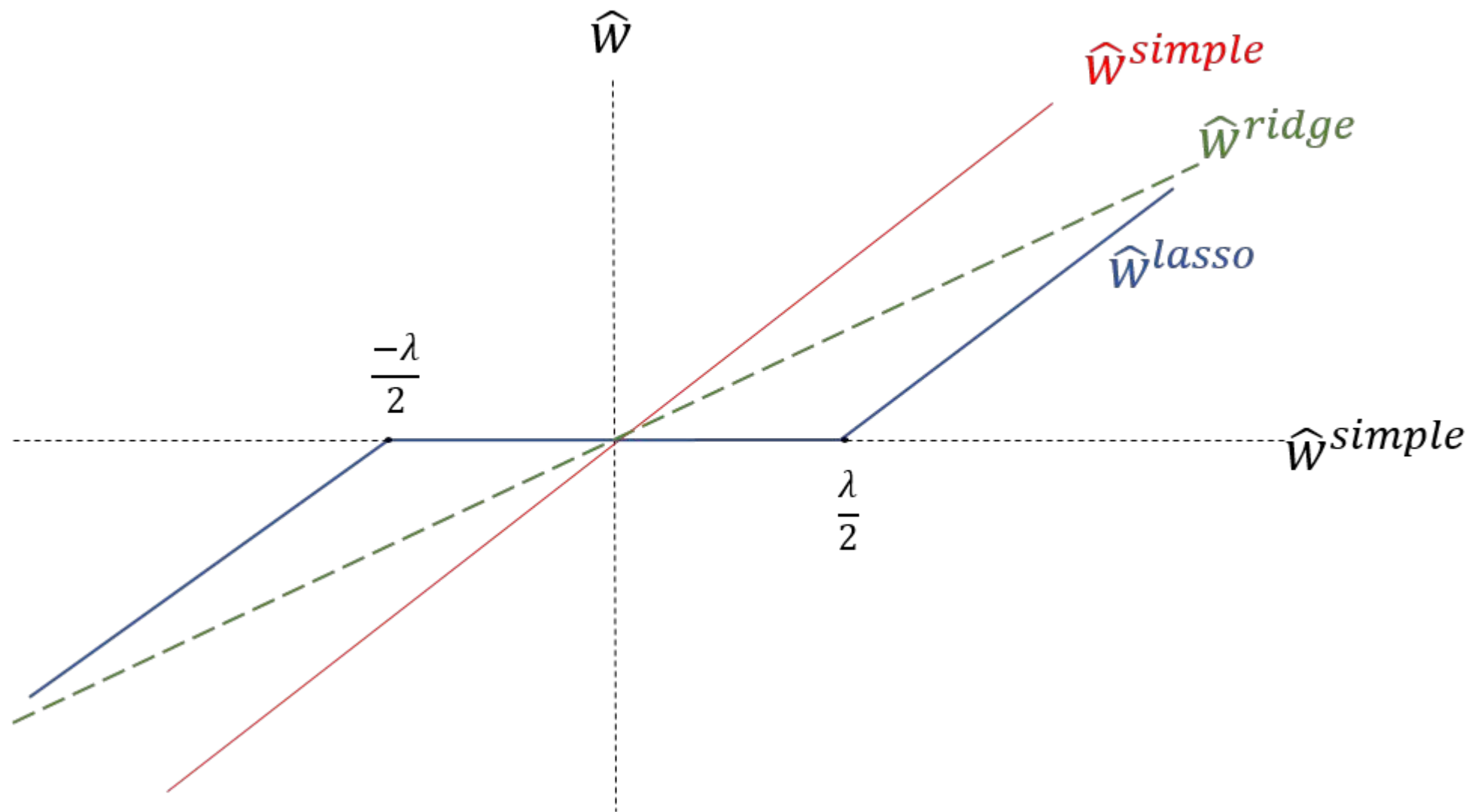
## Lasso regression



## Lasso регрессия

$$w_j = \begin{cases} g(w_{-j}) + \frac{\lambda}{2}, & \text{if } g(w_{-j}) < -\frac{\lambda}{2} \\ 0, & \text{if } -\frac{\lambda}{2} \leq g(w_{-j}) \leq \frac{\lambda}{2} \\ g(w_{-j}) - \frac{\lambda}{2}, & \text{if } g(w_{-j}) > \frac{\lambda}{2} \end{cases}$$

## Сравнение коэффициентов при разных видах регрессии



---

Практика

**Как изменяются  
коэффициенты?**

---

Часть 3-4

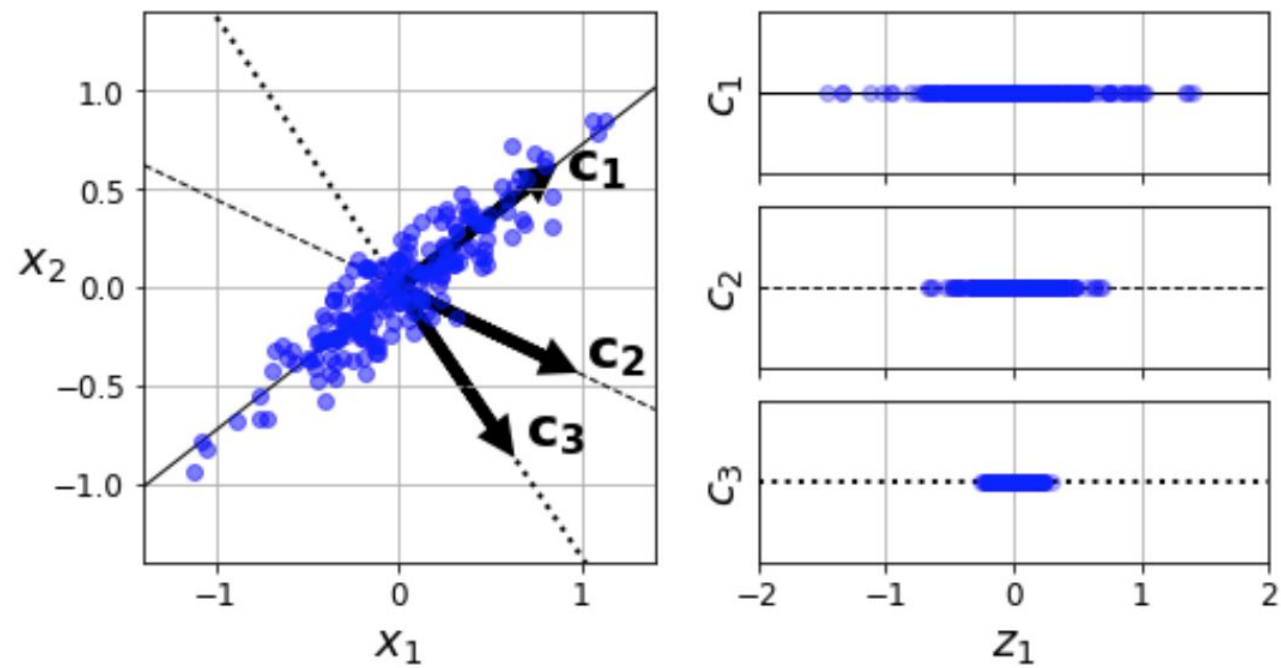
**Уменьшение размерности  
пространства**

**Метод главных компонент**

## Уменьшение размерности пространства

- Трансформирует  $p$  фич в  $M$  линейных комбинаций этих фич
- Новые фичи используются для построения модели
- Новые фичи должны:
  - Уменьшить размерность пространства
  - Сохранить как можно больший процент variance исходных данных

## Уменьшение размерности пространства



## Уменьшение размерности пространства

### Большое количество фич

- Замедляет работу алгоритмов ML
- Усложняет процесс поиска решений (curse of dimensionality)
- Среднее расстояние между двумя случайно выбранными точками в квадрате с длиной стороны 1 равно 0.52
- В 1,000,000-ом гиперкубе  $\approx 408.52$



## Уменьшение размерности пространства

# Сокращение размерности пространства признаков

- Уменьшить размерность данных
- Ускорить процесс обучения
- Уменьшить шумы
- Визуализировать данные

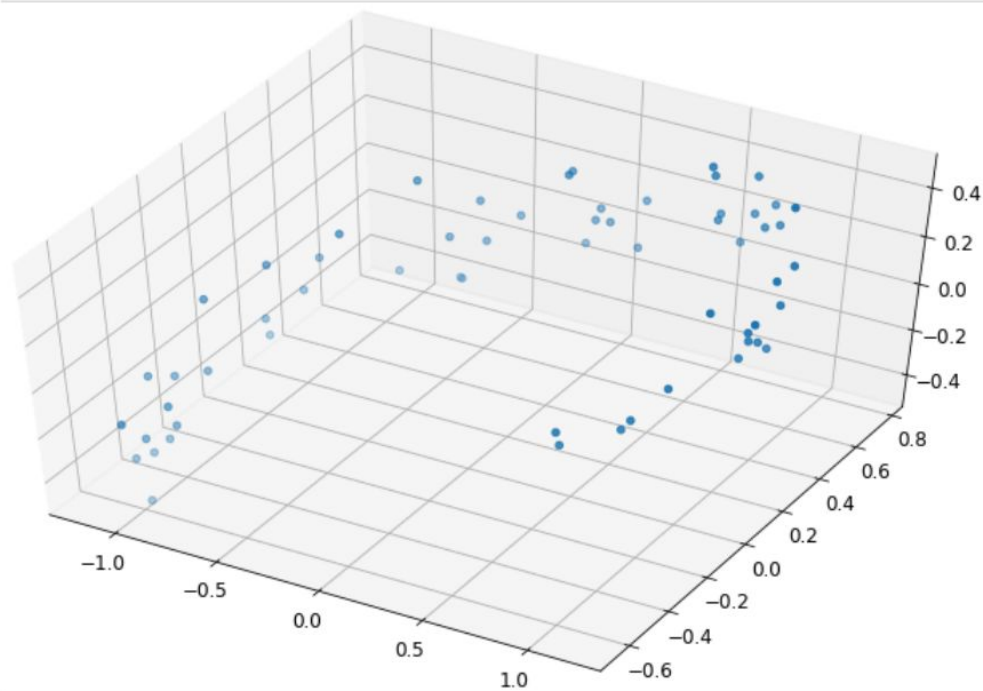
## Уменьшение размерности пространства

### Основные способы

- Projection
  - PCA
  - Kernel PCA
- Manifold learning
  - Isometric Mapping (Isomap)
  - Locally Linear Embedding (LLE)
  - t-distributed Stochastic Neighbour Embedding (t-SNE)

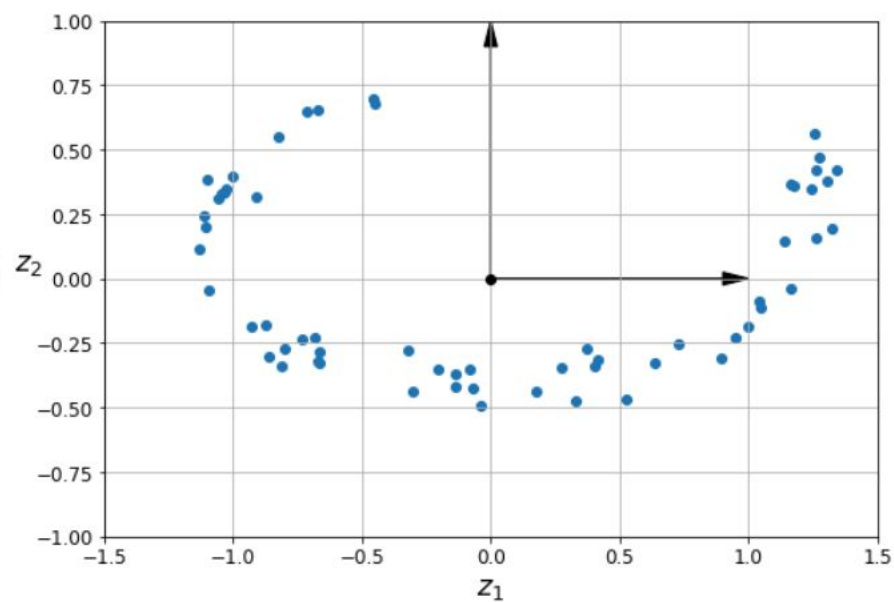
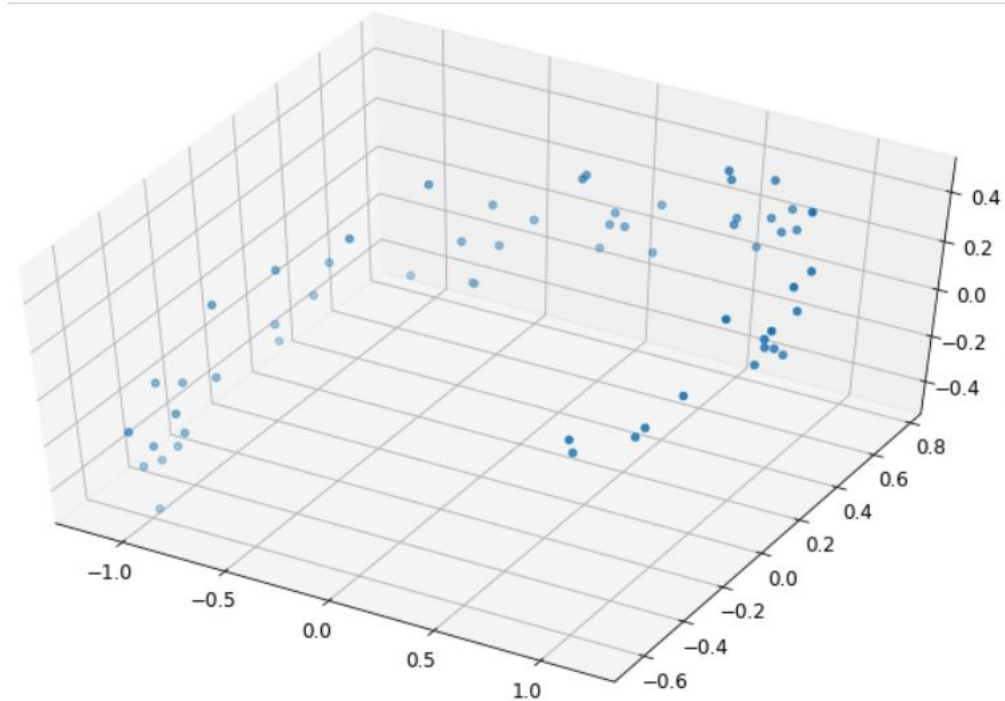
## Уменьшение размерности пространства

# Projection



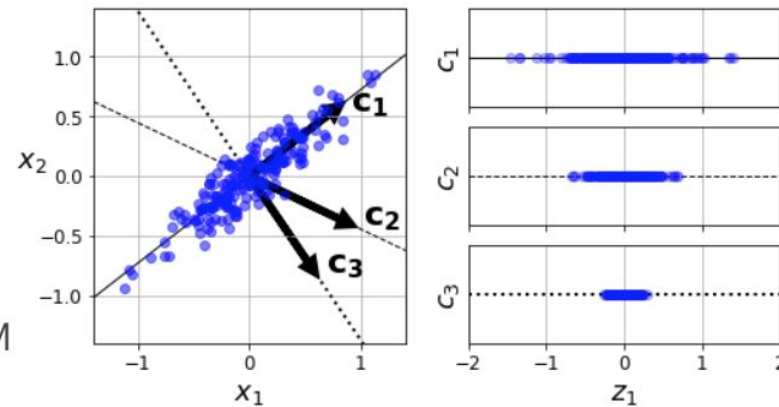
## Уменьшение размерности пространства

# Projection



## Principal component analysis (PCA)

- Самый популярный projection способ
- Проецирует исходные данные на компоненты
- Каждая следующая компонента:
  - ортогональна всем предыдущим
  - описывает максимальное количество остаточного variance



## ELBOW METHOD

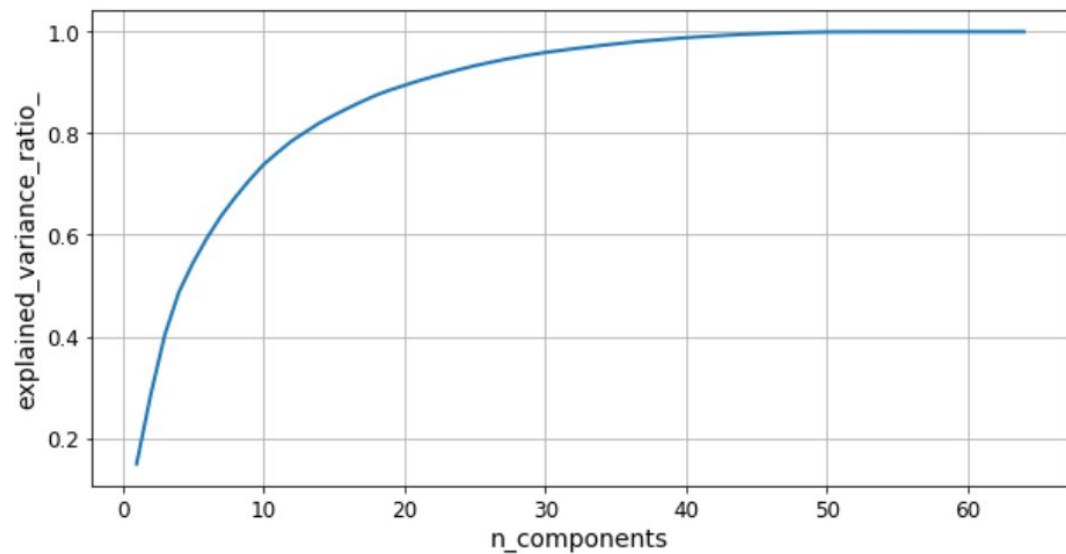
- Обязательный параметр PCA - количество компонент
- Для определения оптимального кол-ва компонент используется метод “локтя” (elbow method)

```
for n in range(1, n_features + 1):  
    x.append(n)  
    y.append(variance описанный n компонентами)  
plot(x, y)
```



## Локоть

Выбираем количество компонент по проценту сохраненного variance



## PCA в sklearn

- Основан на методе матричной декомпозиции SVD (Singular Value Decomposition, см. далее)
- Центрирует данные (PCA предполагает, что данные центрированы)
- Направление компонент не стабильно
- Можно указывать ожидаемый variance вместо количества компонент

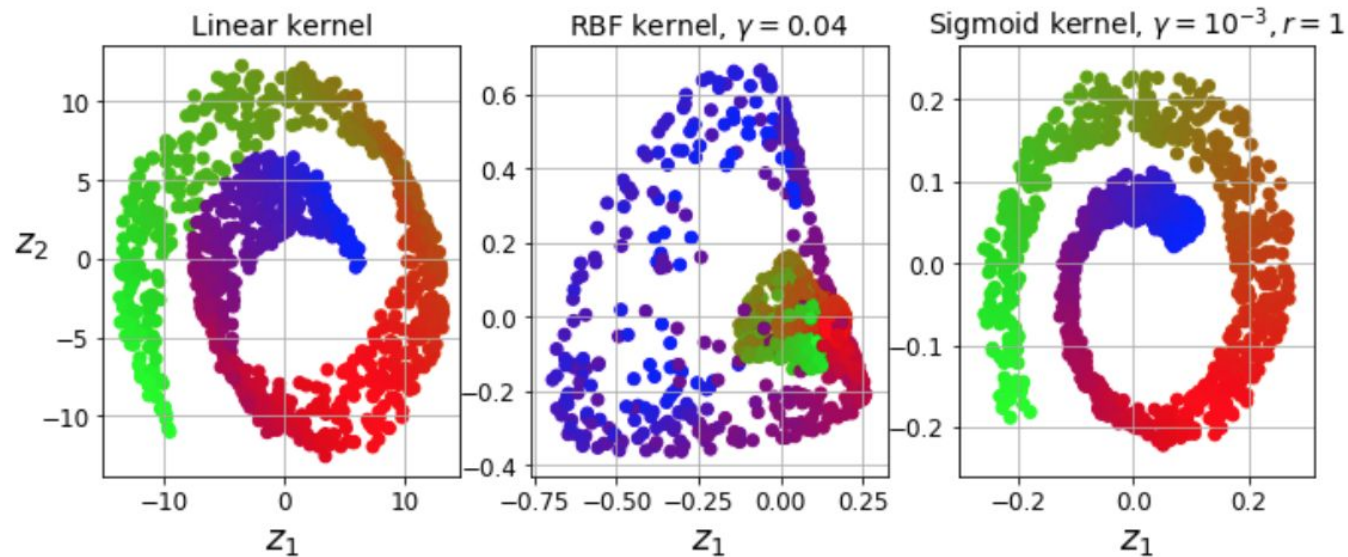


## PCA в sklearn

- Если данные очень большие и не помещаются в память
  - IncrementalPCA
  - Numpy memmap
- Нужна быстрая оценка первых  $d$  компонент?
  - RandomizedPCA (стохастический алгоритм)
  - $d \ll n$ ,  $n$  - количество фич

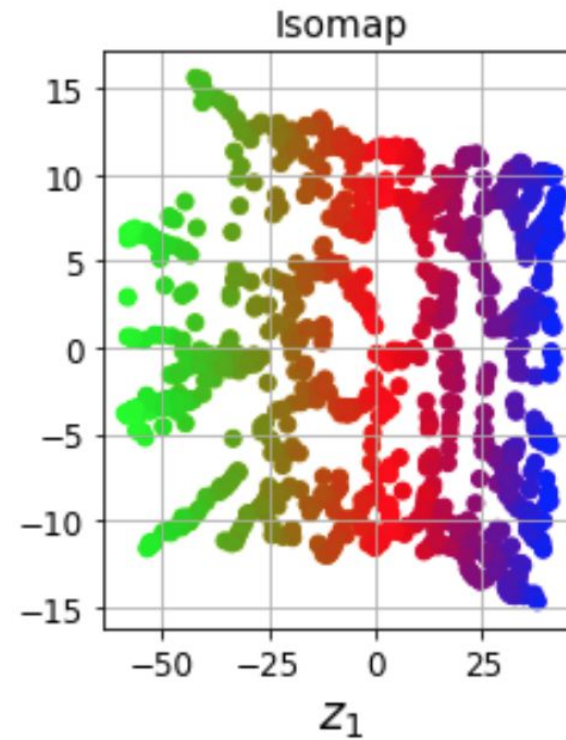
## Kernel PCA

- Kernel trick можно использовать в PCA (kPCA)
- Позволяет осуществить сложную нелинейную проекцию



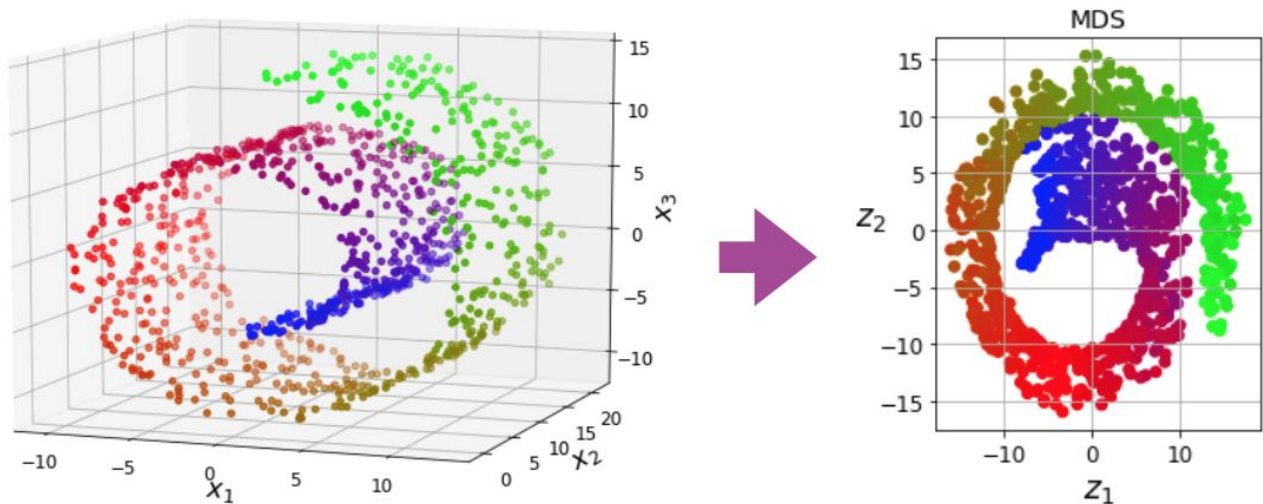
# ISOMAP

- Создает граф, соединяя экземпляры с их ближайшими соседями
- Уменьшает размерность, пытаясь сохранить геодезическое расстояние



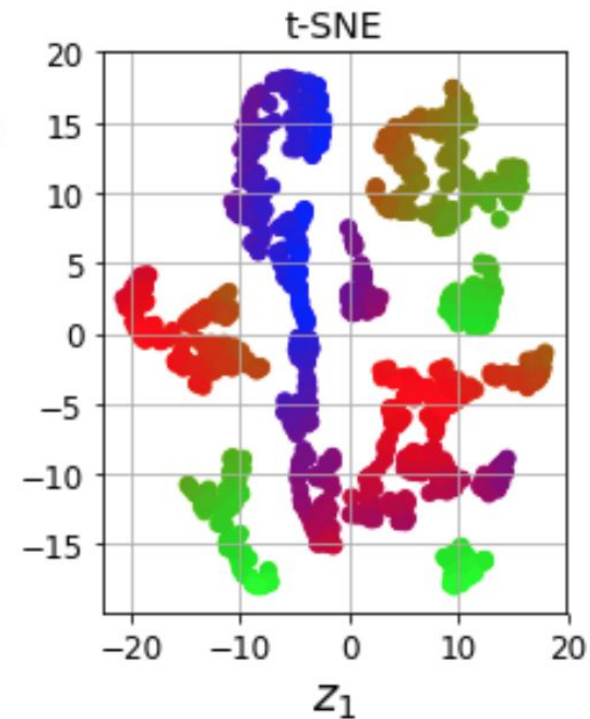
## MDS

Уменьшает размерность, пытаясь сохранить расстояние между объектами



## T-SNE

- Уменьшает размерность, сохраняя близость экземпляров в пространстве
- близкие в исходном = близкие в новом
- далекие в исходном = далекие в новом
- Используется для визуализации кластеров



---

Практика

# **АНАЛИЗ РЕКЛАМНЫХ БЮДЖЕТОВ**

---

Часть 5

# Обсуждение домашнего задания



## Обсуждение домашнего задания





---

**ЧТО МЫ СЕГОДНЯ УЗНАЛИ**

## ИТОГИ

1

Как устроена **линейная регрессия**

2

Отличия **Lasso** и **Ridge** регрессий

3

Математика под капотом **РСА**

4

Как использовать **метод главных компонент** в **sklearn**

---

**СПАСИБО ЗА ВНИМАНИЕ**