DOI 10.37539/2949-1991.2026.45.10.016

Феофанов Артем Тимофеевич, Студент,
Сахалинский государственный университет

Научный руководитель:
Осипов Геннадий Сергеевич,
д.т.н., профессор кафедры информатики,
Сахалинский государственный университет

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ СНИЖЕНИЯ РАЗМЕРНОСТИ С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕК SCIKIT-LEARN И MINISOM ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON

Аннотация. В работе рассматриваются три метода уменьшения размерности на наборе данных «Ирисы Фишера». Для обучения моделей использовалась библиотеки scikit-learn и MiniSom языка программирования Python.

Ключевые слова: Уменьшение размерности, машинное обучение, метод главных компонент, IsoMap, самоорганизующаяся карта Кохонена.

Введение

В текущих реалиях анализа данных и машинного обучения часто приходится сталкиваться с многомерными наборами данных, где количество признаков исчисляется в десятках, сотнях или тысячах. Их избыточность или малая информативность напрямую влияет на эффективность обучения модели [1]. Для решения данной проблемы используют различные методы уменьшения размерности. Их цель – отображение многомерного пространства в пространства меньшей размерности (например, двумерное или трехмерное для упрощения визуализации) с минимизацией потери полезной информации.

Для изучения методов уменьшения размерности воспользуемся возможностями языка программирования Python. Его особенность – широкое разнообразие доступных библиотек и инструментов для работы с машинным обучением. В работе были задействованы такие средства как scikit-learn [2] и MiniSom.

В рамках задачи уменьшения размерности будут рассмотрены три метода: метод главных компонент (PCA), IsoMap и самоорганизующаяся карта Кохонена (SOM).

Характеристика и инициализация методов уменьшения размерности

В работе будет использоваться датасет «Ирисы Фишера», содержащий 150 экземпляров цветков ирисов, равномерно распределенных по трем биологическим видам (по 50 объектов на каждый вид): Iris *Setosa*, Iris Versicolor и Iris Virginica. Данный набор данных характеризуется 4 признаками: длиной и шириной чашелистика, длиной и шириной лепестка. Scikit-learn предоставляет возможность воспользоваться этим датасетом с помощью функции load_iris () [3] (рис. 1).

```
# Инициализация датасета "Ирисы Фишера"
iris = load_iris()
X = iris.data
y_true = iris.target
```

Рисунок 1. Инициализация "Ирисов Фишера"

Чтобы исключить искажения из-за разного масштаба исходных признаков, следует применить z-масштабирование (нормализацию), которое описывается следующей формулой:

$x' = \frac{x-\overline{x}}{\sigma_x}$, где $\overline{x}$ – среднее арифметическое признака, σ_x – стандартное отклонение признака.

С помощью функции StandardScaler () выполняем данную операцию на датасете (рис. 2):

```
# Нормализация датасета
scaler = StandardScaler()
X = scaler.fit_transform(X)
```

Рисунок 2. Стандартизация датасета

Определим, что представляет собой каждый из трех выбранных методов уменьшения размерности:

- Метод главных компонент (PCA, Principal Component Analysis) – классический метод линейного снижения размерности, направленный на преобразование исходных взаимосвязанных признаков в набор новых некоррелированных переменных (главных компонент), упорядоченных по степени дисперсии [4]. Перед началом работы данные обязательно подвергаются центрированию (из каждого признака вычитается его среднее значение). Математическая постановка задачи формулируется следующим образом: пусть имеется матрица данных X размера n x m, где n – число объектов, m – число признаков. Первая главная компонента $t_1 = \mathrm{X}\mathrm{w}_1$ представляет собой линейную комбинацию исходных признаков с весовым вектором $w_1 \in \mathbb{R}^m$ (при условии $\|\mathrm{w}_1\| = 1$), которая максимизирует дисперсию проекции t_1: $\mathrm{Var}(t_1) = \frac{1}{n}\|\mathrm{X}\mathrm{w}_1\|^2 = w_1^T\left(\frac{X^TX}{n}\right)w_1 = w_1^T\mathrm{S}\mathrm{w}_1$, где $S = \frac{X^TX}{n}$ – выборочная ковариационная матрица для центрированных данных. Задача оптимизации записывается так: $w_1 = \arg\max_{\|\mathrm{w}\|=1} w^T\mathrm{S}\mathrm{w}$. Решением этой задачи является собственный вектор ковариационной матрицы S, соответствующий ее максимальному собственному значению λ_1, причем $\lambda_1 = \mathrm{Var}(t_1)$. Последующие главные компоненты строятся аналогично, будучи ортогональными предыдущим. Для использования метода в scikit-learn есть функция PCA (), где с помощью параметра n_components можно задать количество компонент до которых нужно уменьшить размерность датасета (рис. 3).

```
# Метод главных компонент (PCA)
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)
```

Рисунок 3. Инициализация PCA

- Метод IsoMap (Isometric Mapping) – метод нелинейного снижения размерности, относящийся к категории обучения на многообразиях. В отличие от PCA, который эффективен лишь в линейных пространствах, IsoMap разработан для работы со сложными «складчатыми»

поверхностями в высокоразмерном пространстве. Он использует геодезическое расстояние – кратчайшее расстояние между точками, измеренное вдоль поверхности многообразия. Алгоритм метода заключается в следующем: происходит поиск ближайших соседей для каждой точки и формирование графа смежности; затем рассчитываются геодезические расстояния между всеми парами точек как кратчайших путей в построенном графе (например, с помощью алгоритма Дейкстры); на основе полученных расстояний строится низкоразмерное представление данных, которое максимально сохраняет эти расстояния. Инициализируем этот метод с помощью функции Isomap (), где параметр n_neighbors обозначает число ближайших соседей для каждой точки при построении графа (рис. 4).

```
# Метод IsoMap
isomap = Isomap(n_neighbors=5, n_components=2)
X_isomap = isomap.fit_transform(X_scaled)
```

Рисунок 4. Инициализация IsoMap

- Самоорганизующаяся карта Кохонена (SOM, Self-Organizing Map) представляет собой нейронную сеть без учителя, предназначенную для проецирования многомерного пространства на дискретную сетку нейронов (обычно двумерную). При инициализации обучения алгоритм присваивает каждому нейрону карты j случайный вектор весов $w_j(0)\ \epsilon\ \mathbb{R}^d$, затем из обучающей выборки выбирается случайный входной вектор $x(t)$. Далее находится нейрон c, чей вектор весов наиболее близок к $x(t)$: $c = \underset{j}{\operatorname{argmin}}\|x(t) - w_j(t)\|$. Для нейрона-победителя и его топологических соседей обновляются веса: $w_j(t+1) = w_j(t) + \eta(t) * h_{c,j}(t) * \left(x(t) - w_j(t)\right)$,где $\eta(t)$ – скорость обучения, $h_{c,j}(t)$ – функция соседства, определяющая влияние нейрона-победителя на нейрон j. Процесс повторяется заново для всех входных векторов до тех пор, пока веса не стабилизируются. Для инициализации SOM используем библиотеку MiniSom (рис. 5). Первые два параметра grid_size определяют размеры двумерной сетки нейронов, input_len – количество признаков входного вектора, sigma – радиус функции соседства, learning_rate – шаг изменения весов нейронов [5].

```
# SOM
som = MiniSom(grid_size=7, grid_size=7, input_len=4, sigma=1.0, learning_rate=0.5)
som.random_weights_init(X_scaled)
```

Рисунок 5. Инициализация SOM

Характеристика и инициализация методов уменьшения размерности

В результате применения методов уменьшения размерности на датасет получаем следующие графики:

- Для метода PCA: на графике первой и второй главных компонент видно (рис. 6), что класс *Iris Setosa* образует полностью изолированный компактный кластер в левой части плоскости. Классы *Iris Versicolor* и *Iris Virginica* располагаются в правой части и образуют два вытянутых эллипсоида, имеющих незначительную зону перекрытия на границе, что отражает естественную морфологическую схожесть этих двух видов.

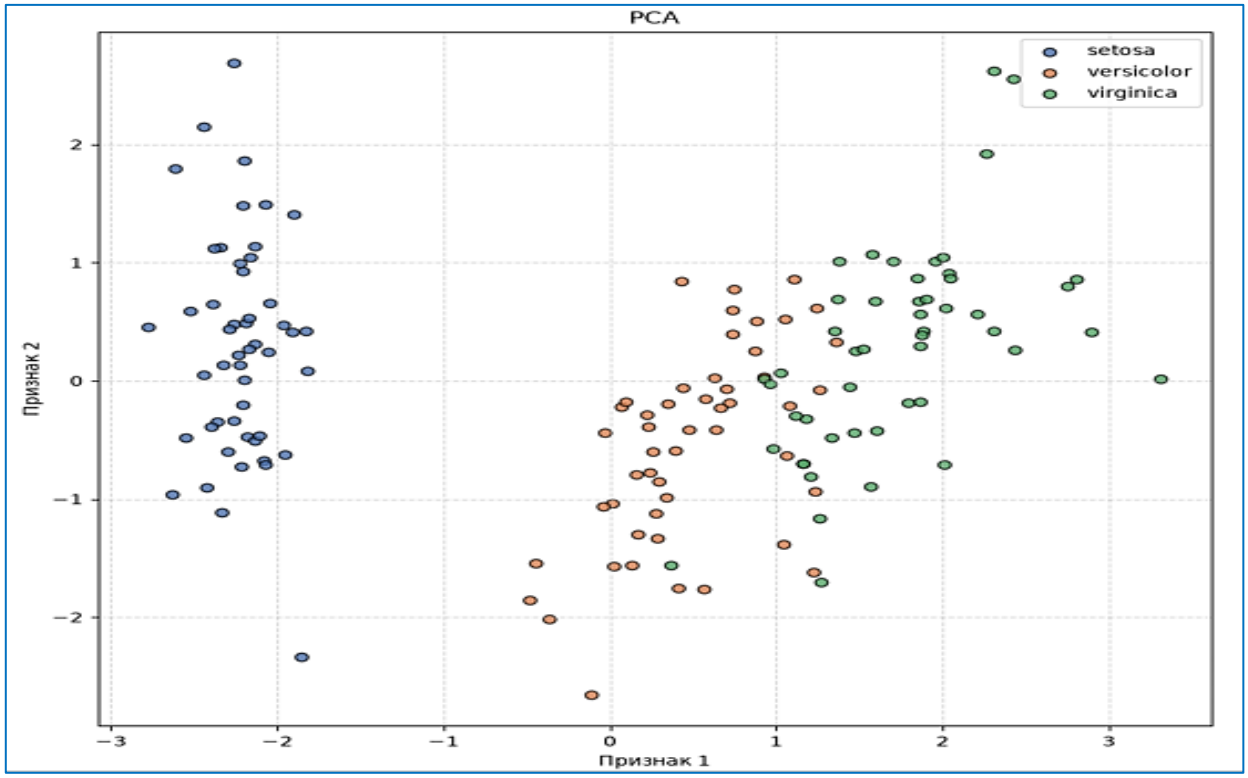


Рисунок 6. График PCA

- Для IsoMap: класс *Setosa* четко обособлен (рис. 7). Однако кластеры *Versicolor* и *Virginica* приобретают более изгибающуюся нелинейную форму вдоль графового многообразия, при этом граница между ними становится более извилистой, чем при PCA.

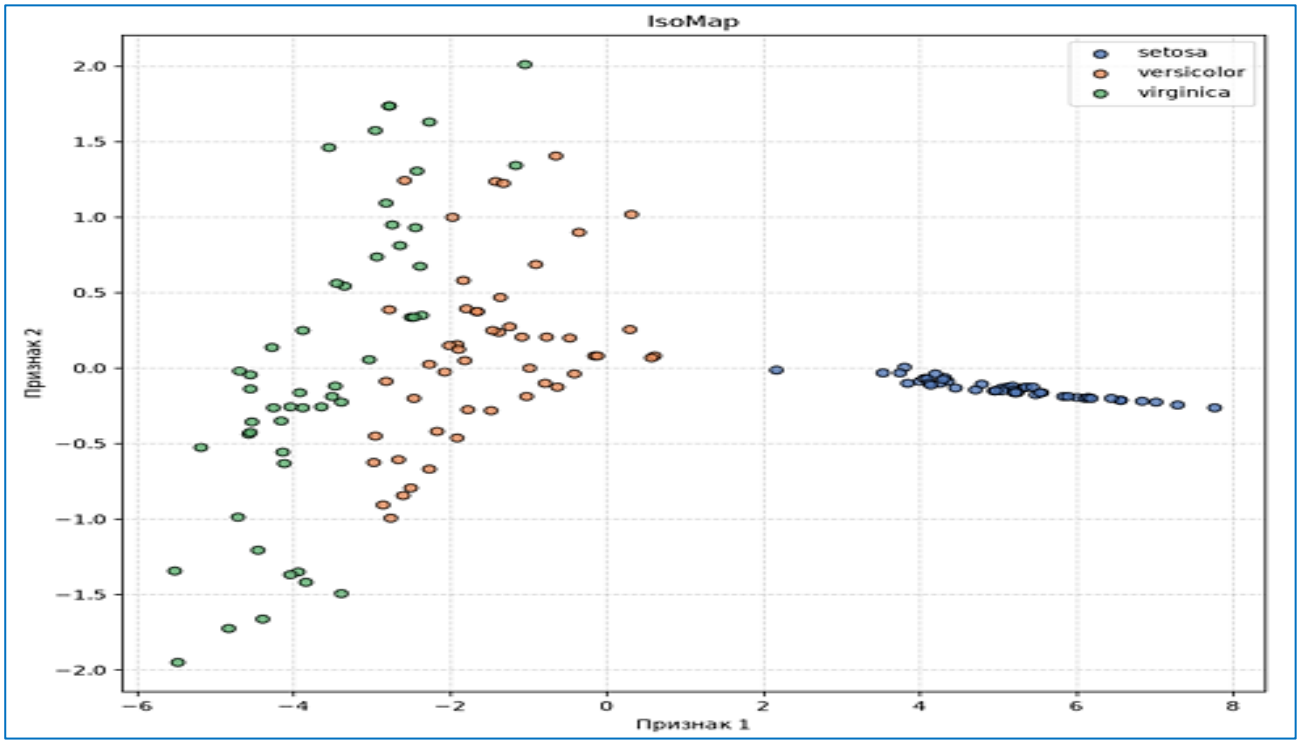


Рисунок 7. График IsoMap

- Для SOM: поскольку координаты получаются целочисленными, к ним добавляется небольшой случайный шум, чтобы точки не накладывались друг на друга. Класс *Setosa* образует полностью обособленную область в правой части карты, не имея ни одного пересечения с другими видами (рис. 8). Классы *Versicolor* и *Virginica* активируют смежные нейроны. *Virginica* преимущественно локализуется в левом сегменте сетки, а *Versicolor* – в верхнем.

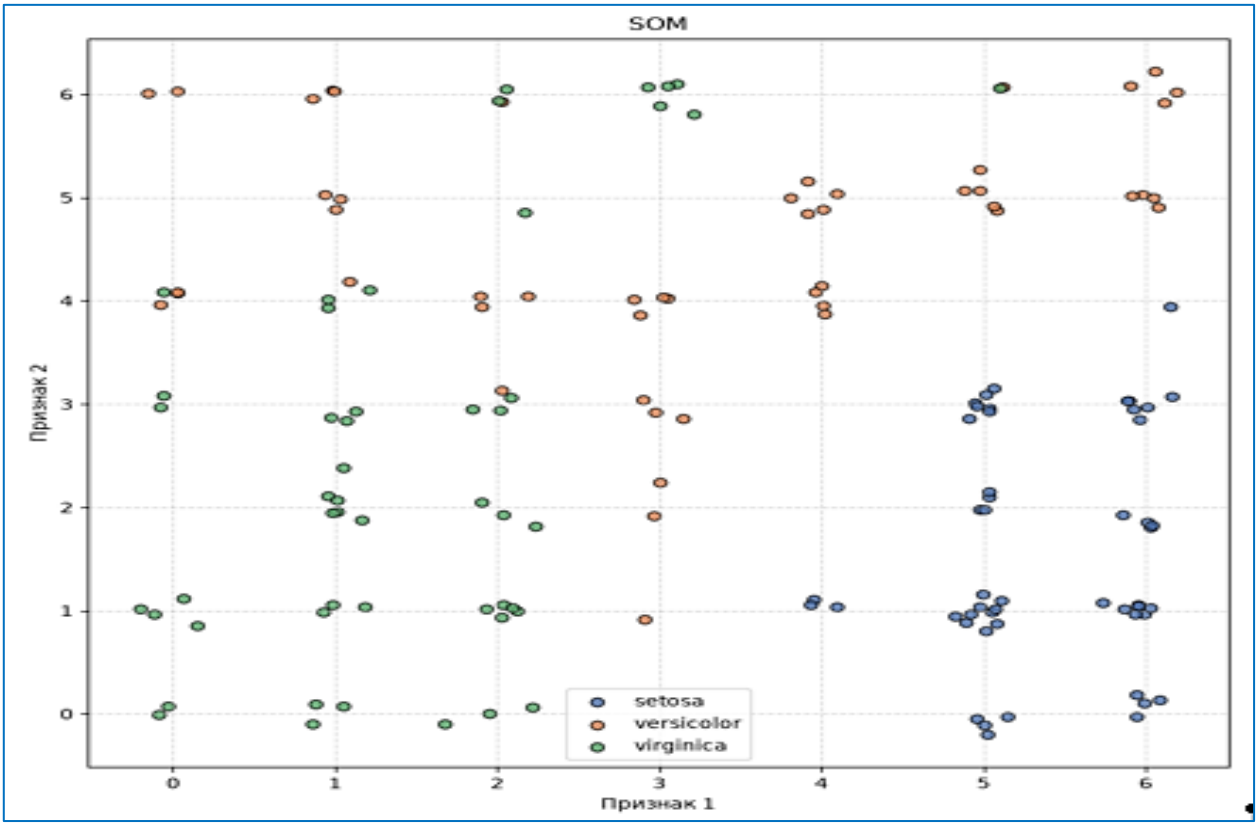


Рисунок 8. График SOM

Для того, чтобы сравнить методы, были использованы следующие метрики (табл. 1):

- Время, затраченное на кластеризацию.
- Коэффициент силуэта, являющийся критерием для оценки выраженности кластера ($\epsilon 1; 1$). Чем ближе к единице, тем лучше.
- Индекс Калински-Харабаса – отношение суммы межкластерной дисперсии к сумме внутрикластерной дисперсии ($\epsilon\ 0; \infty$)). Чем больше, тем лучше.
- Индекс Дэвиса-Болдина, сравнивающий расстояние между кластерами с размером самих кластеров ($\epsilon\ 0; \infty$)). Чем ближе к нулю, тем лучше.

Таблица 1

Сравнительная таблица методов уменьшения размерности

Метод кластеризации	**Время (мс)**	**Коэфф. силуэта**	**Калински-Харабас**	**Дэвис-Болдин**
PCA	**0.36**	0.4014	221.96	0.9555
IsoMap	5.36	**0.4927**	**693.17**	**0.7943**
SOM	12.80	0.3335	80.20	1.2428

Выводы

Все три рассмотренных метода уменьшения размерности успешно справились с базовой задачей визуализации и сепарации классов датасета «Ирисы Фишера», при этом во всех случаях вид *Iris Setosa* продемонстрировал четкую обособленность от остальных классов.

Метод IsoMap показал наилучшие результаты по всем метрикам качества кластеризации: он достигает наибольшего значения коэффициента силуэта (0,4927) и индекса Калински-Харабаса (693,17), а также наименьшего индекса Дэвиса-Болдина (0,7943). Это свидетельствует о том, что сохранение геодезических расстояний позволяет более эффективно разделять классы *Versicolor* и *Virginica* за счет учета нелинейной структуры многообразия данных.

Метод PCA продемонстрировал наивысшее быстродействие (0,36 мс), уступая IsoMap по качеству кластеризации, но показывая стабильно хорошие показатели (0,4014 по силуэту и 221,96 по Калински-Харабасу). Благодаря минимальным вычислительным затратам, PCA является наиболее оптимальным выбором для первичного анализа данных и структур.

Самоорганизующаяся карта Кохонена показала наименьшую эффективность на данном наборе данных, продемонстрировав худшие метрики (коэффициент силуэта 0,3335, индекс Дэвиса-Болдина 1,2428) и наибольшее время выполнения (12,80 мс). Это обусловлено спецификой алгоритма, проецирующего данные на дискретную сетку нейронов.

Список литературы:

1. Викиконспекты ИТМО. Уменьшение размерности [Электронный ресурс]. – URL: https://neerc.ifmo.ru/wiki/index.php?title=%D0%A3%D0%BC%D0%B5%D0%BD%D1%8C%D1%88%D0%B5%D0%BD%D0%B8%D0%B5_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%80%D0%BD%D0%BE%D1%81%D1%82%D0%B8#.D0.92.D1.8B.D0.B4.D0.B5.D0.BB.D0.B5.D0.BD.D0.B8.D0.B5_.D0.BF.D1.80.D0.B8.D0.B7.D0.BD.D0.B0.D0.BA.D0.BE.D0.B2/ (дата обращения: 29.10.2026).

2. Феофанов, А. Т. Сравнительный анализ методов кластеризации с использованием библиотеки scikit-learn языка программирования Python / А. Т. Феофанов // Вектор научной мысли. – 2026. – № 5 (34). – С. 484-494. – DOI 10.58351/2949-2041.2026.34.5.049. – EDN QUKFCF.

3. scikit-learn: официальная документация. The Iris Dataset [Электронный ресурс]. – URL: https://scikit-learn.org/1.4/auto_examples/datasets/plot_iris_dataset.html/ (дата обращения: 29.10.2026).

4. Метод главных компонент (PCA). Принцип работы и реализация с нуля на Python [Электронный ресурс]. – URL: https://habr.com/ru/articles/799001/ (дата обращения: 29.10.2026).

5. MiniSom: официальный репозиторий [Электронный ресурс]. – URL: https://github.com/JustGlowing/minisom/ (дата обращения: 29.10.2026).