7 практическая работа «Отбор, оценка значимости признаков и построение моделей машинного обучения»

Цель практики:

Освоить методы отбора и оценки значимости признаков, построение и анализ моделей с использованием различных подходов, а также визуализацию данных. Это включает как линейные модели, так и деревья решений, метод главных компонент (PCA), а также современные методы объяснимости моделей, такие как SHAP. Практика направлена на развитие навыков работы с большими наборами данных, интерпретации результатов и визуализации значимости признаков.

Оглавление

Задание 5

Выбор набора данных 7

1.1. Значение выбора набора данных 7

1.2. Критерии выбора набора данных 7

1.3. Рекомендуемые наборы данных 8

1.4. Практические аспекты работы с выбранными данными 10

Построение моделей и оценка значимости признаков 11

2.1. Значимость признаков 11

2.2. Построение линейной модели 11

2.3. Построение моделей на основе деревьев решений 12

2.4. SHAP-значения 13

2.5. Сравнение подходов к оценке значимости 14

2.6. Визуализация значимости признаков 14

2.7. Рекомендации для практической реализации 15

Визуализация значимости признаков 16

3.1. Значение визуализации значимости признаков 16

3.2. Методы визуализации значимости признаков 16

3.2.1. Гистограммы и столбчатые диаграммы 16

Пример гистограммы и столбчатой диаграммы 16

3.2.2. Графики SHAP (SHAP Summary Plot) 17

3.2.2.1 Типы графиков SHAP: 17

3.2.2.2 Пример Summary Plot 18

3.3.3. Графики важности признаков в моделях деревьев 18

3.3.4. Боксплоты для анализа распределения значимости 18
3.3 Особенности визуализации для разных моделей 18
3.4. Преимущества визуализации значимости признаков 19
3.5. Рекомендации по построению визуализации 19
3.6. Выводы по разделу 20
Сравнение качества моделей 21
4.1. Зачем сравнивать качество моделей 21
4.2. Метрики оценки качества моделей 21
4.2.1 Для задач классификации: 21
4.2.2 Для задач регрессии: 22
4.3. Подходы к сравнению моделей 23
4.3.1. Модели на всех признаках: 23
4.3.2. Модели на отобранных признаках: 23
4.3.3. Модели на преобразованных признаках (PCA): 24
4.3.4. Ансамблевые методы: 24
4.4. Проблемы при сравнении моделей 24
4.5. Как проводить оценку моделей на практике 25
4.6. Выводы по сравнению моделей 25
Снижение размерности и визуализация данных 26
5.1 Зачем снижать размерность данных 26
5.2. Методы снижения размерности 26
5.2.1. Principal Component Analysis (PCA) — метод главных компонент 26
5.2.1.1. Пример использования PCA: 27
5.2.2. t-SNE — Stochastic Neighbor Embedding 27

5.2.2.1. Пример использования t-SNE:..... 28
5.2.3. UMAP — Uniform Manifold Approximation and Projection..... 28
5.2.3.1. Пример использования UMAP 29
5.3. Применение метода снижения размерности 29
5.4. Визуализация данных..... 30
5.4.1. Пример визуализации с PCA 30
5.4.2. Пример визуализации с t-SNE..... 30
5.5. Преимущества и ограничения методов 30
5.6. Рекомендации по снижению размерности 31

ЗАДАНИЕ

1. **Выбор набора данных**:

- Выберите один из предложенных датасетов или другой подходящий набор данных с большим количеством признаков.

2. **Построение моделей и оценка значимости признаков**:

- Построение линейной модели:
- Использовать критерий значимости признаков (p-value или веса).
- Применить жадный алгоритм отбора признаков.
- Модель с Lasso-регуляризацией:
- Оценить влияние параметра alpha на зануление признаков.
- Построение моделей на основе деревьев решений:
- Случайный лес.
- Градиентный бустинг.
- Вывести метрики качества моделей.
- Оценка значимости признаков:
- Использовать встроенные методы моделей (например, feature_importances_).
- Применить SHAP-значения для интерпретации.

3. **Визуализация значимости признаков**:

- Визуализировать отсортированную значимость признаков для каждого алгоритма.

4. **Сравнение качества моделей**:

- Оценить качество моделей:
- На всех признаках.
- На основе наиболее значимых признаков.

5. **Снижение размерности и визуализация данных**:

- Применить метод PCA, обучить модель на преобразованных признаках.

- Визуализировать данные с помощью t-SNE или UMAP.

6. **Подготовка отчета**:

- Подробное описание хода выполнения работы.
- Листинг выполненных заданий.
- Скриншоты результатов.
- Защита работы.

ВЫБОР НАБОРА ДАННЫХ

1.1. Значение выбора набора данных

Выбор набора данных является ключевым этапом в любом проекте машинного обучения. Данные определяют возможности для анализа, выявления закономерностей и построения моделей. Для успешного выполнения задачи необходимо учитывать:

- **Цель исследования**: Например, прогнозирование, классификация или снижение размерности.
- **Сложность задачи**: Простые задачи требуют базовых наборов данных, а для сложных задач подойдут данные с большим количеством признаков и разнообразными типами переменных.
- **Качество данных**: Проверка на пропуски, выбросы, коррелированные признаки и несбалансированные классы.

1.2. Критерии выбора набора данных

Чтобы выбрать подходящий набор данных, нужно учитывать следующие аспекты:

1. **Размер набора данных**:

- Большие объемы данных предпочтительны для более точного обучения моделей, но требуют больше вычислительных ресурсов.
- Меньшие наборы данных подходят для начальных этапов работы, обучения методам и быстрого тестирования.

2. **Тип задачи**:

- Для задачи классификации нужны данные с меткой класса (целевой

переменной).

- Для регрессии целевая переменная должна быть числовой и непрерывной.
- Для кластеризации и снижения размерности целевая переменная может отсутствовать.

3. **Доступность и источник данных**:

- Наборы данных, доступные через библиотеки Python, такие как sklearn.datasets или платформы (например, Kaggle), упрощают загрузку и предварительную обработку.

4. **Количество признаков и их разнообразие**:

- Наборы данных с большим количеством признаков предоставляют больше возможностей для отбора и оценки значимости.
- Наличие как числовых, так и категориальных данных позволяет тестировать гибридные подходы.

5. **Проблема пропущенных данных**:

- Проверить, есть ли пропуски в данных и насколько они критичны.
- Выявить, какой тип обработки требуется: удаление строк, заполнение или создание новых признаков.

6. **Устойчивость к несбалансированным классам**:

- В задачах классификации важно учитывать распределение целевой переменной. Несбалансированные классы требуют специальных методов обработки, например, ресэмплинга или взвешивания метрик.

1.3. Рекомендуемые наборы данных

В данной работе предлагается использовать один из следующих наборов данных:

1. **Стоимость домов** *(House Prices)*:

- Задача: Регрессия.
- Источник: sklearn.datasets.fetch_openml(name="house_prices", as_frame=True).
- Особенности: Большое количество числовых признаков, важность работы с корреляцией и жадными алгоритмами.

2. **Одобрение кредита** *(Home Loan Approval)*:

- Задача: Классификация.
- Источник: Kaggle.
- Особенности: Смесь числовых и категориальных признаков, работа с пропущенными значениями.

3. **Расходы на рекламные кампании** *(Media's Cost Prediction)*:

- Задача: Регрессия.
- Источник: Kaggle.
- Особенности: Исследование взаимосвязи затрат на рекламу с доходами компании.

4. **Сердечный приступ** *(Heart Attack Analysis)*:

- Задача: Классификация.
- Источник: Kaggle.
- Особенности: Медицинский набор данных с особенностями обработки бинарных и числовых переменных.

5. **Сердечная недостаточность** *(Heart Failure Prediction)*:

- Задача: Классификация.
- Источник: Kaggle.
- Особенности: Изучение значимости клинических факторов в прогнозировании заболеваний.

6. **Классификация курильщиков** *(Smoking Drinking Dataset)*:

- Задача: Классификация.
- Источник: Kaggle.
- Особенности: Социальные и поведенческие данные с акцентом на

категориальные признаки.

7. **Оценка расходов клиентов магазина** *(Customer's Dataset)*:

- Задача: Регрессия.
- Источник: Kaggle.
- Особенности: Анализ поведения потребителей и расходов.

8. **Астероиды** *(NASA Nearest Earth Objects)*:

- Задача: Регрессия или классификация.
- Источник: Kaggle.
- Особенности: Объемный датасет с астрономическими характеристиками, возможность работы с высокоразмерными данными.

1.4. Практические аспекты работы с выбранными данными

После выбора набора данных необходимо:

1. Загрузить данные с источника.
2. Изучить структуру данных: типы переменных, наличие пропусков, размеры.
3. Провести начальную предобработку:

- Очистка данных.
- Кодирование категориальных признаков (если требуется).
- Нормализация или стандартизация числовых данных.

4. Провести базовую визуализацию для понимания распределения данных.

Этот этап создаёт базу для построения моделей и анализа признаков, что будет выполнено в следующих частях задания.

ПОСТРОЕНИЕ МОДЕЛЕЙ И ОЦЕНКА ЗНАЧИМОСТИ ПРИЗНАКОВ

2.1. Значимость признаков

Значимость признаков (feature importance) оценивает вклад каждого признака в предсказания модели. Анализ значимости позволяет:

- Определить, какие признаки оказывают наибольшее влияние на целевую переменную.
- Упростить модель за счёт исключения нерелевантных признаков.
- Повысить интерпретируемость модели.

Методы оценки значимости зависят от типа модели:

- **Линейные модели** используют коэффициенты при признаках.
- **Деревья решений** рассчитывают значимость на основе уменьшения неопределённости (например, Gini impurity или информационной энтропии).
- **Модели с регуляризацией** добавляют штрафы, зануляя незначимые признаки.
- **Методы SHAP** предоставляют универсальный подход к объяснению значимости признаков.

2.2. Построение линейной модели

Линейные модели основываются на линейной зависимости между признаками X и целевой переменной y. Формула 1:

$$y = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b \quad (1)$$

Где: w_i — веса (коэффициенты) при признаках, b — смещение.

1. **Коэффициенты модели**:

- Коэффициенты w_i указывают на важность каждого признака.
- Знак коэффициента показывает направление влияния: положительное или отрицательное.

2. **Оценка значимости признаков**:

- **P-value**: Используется в статистических методах для проверки гипотезы о значимости коэффициента w_i. Малое значение p-value (< 0.05) говорит о том, что признак статистически значим.
- **Регуляризация**:
  - Метод Lasso (L1-регуляризация) добавляет штраф, который приводит к занулению коэффициентов малозначимых признаков.
  - Метод Ridge (L2-регуляризация) штрафует большие значения коэффициентов, уменьшая их влияние, но не зануляет их.

3. **Жадный алгоритм**:

- Последовательное добавление признаков в модель с оценкой её качества (например, метрики R^2 или F1-меры) позволяет отбирать только значимые признаки.

2.3. Построение моделей на основе деревьев решений

Модели на основе деревьев решений, такие как случайный лес (Random Forest) и градиентный бустинг (Gradient Boosting), определяют значимость признаков на основе их влияния на разбиения данных.

1. **Случайный лес**:

- Основан на ансамбле деревьев решений.
- Значимость признака оценивается на основе:
- Уменьшения критерия расщепления (например, Gini impurity или

энтропии).

- Количества разбиений, в которых участвует признак.

2. **Градиентный бустинг**:

- Использует ансамбль слабых моделей (например, деревьев решений), обученных последовательно.
- Значимость признаков вычисляется аналогично случайному лесу, но с учётом веса каждого дерева.

3. **Метрики качества моделей**:

- **Классификация**:
  - Accuracy
  - Precision
  - Recall
  - F1-score
  - ROC-AUC.
- **Регрессия**:
  - MSE (среднеквадратичная ошибка)
  - RMSE (корень из MSE)
  - MAE (средняя абсолютная ошибка).

2.4. SHAP-значения

SHAP (Shapley Additive Explanations) — метод объяснения предсказаний модели. Он вычисляет вклад каждого признака в предсказания, основываясь на концепции теории игр. Особенности и преимущества представлены в таблице 2.4.1

Таблица 2.4.1 - SHAP

Особенности SHAP	Преимущества SHAP
Универсальность: работает с любыми моделями	Чёткая интерпретация значимости признаков

Продолжение таблицы 2.4.1

Визуализация: позволяет интерпретировать влияние признаков как на уровне набора данных, так и на уровне отдельных наблюдений	Возможность выявить взаимозависимости между признаками
Баланс между локальной (для одного предсказания) и глобальной (для всего набора данных) объяснимостью	

2.5. Сравнение подходов к оценке значимости

Таблица 2.5.1 – Сравнение подходов

Метод	**Линейные модели**	**Lasso-регуляризация**	**Деревья решений**	**SHAP**
Основа оценки	Коэффициенты w_i	Коэффициенты w_i	Уменьшение неопределённости	Теория игр
Преимущества	Простота, интерпретируемость	Автоматический отбор признаков	Не требуется предварительная обработка	Гибкость, высокая интерпретируемость
Ограничения	Линейность ограничивает точность	Зависимость от выбора α	Может переобучаться	Высокая вычислительная сложность

2.6. Визуализация значимости признаков

Для улучшения интерпретации используются графики значимости:

- **Линейные модели**: Гистограммы коэффициентов.
- **Деревья решений**: Диаграммы важности признаков (feature_importances_).
- **SHAP-значения**:
  - Глобальные графики (summary plots): показывают общую значимость признаков.
  - Локальные графики (force plots): демонстрируют вклад каждого признака в предсказание.

2.7. Рекомендации для практической реализации

- Используйте несколько методов для оценки значимости признаков, чтобы получить более полное понимание.
- Проводите сравнение моделей на всех признаках и на основе отобранных значимых признаков.
- Визуализируйте результаты для каждого метода, чтобы упростить интерпретацию.

ВИЗУАЛИЗАЦИЯ ЗНАЧИМОСТИ ПРИЗНАКОВ

3.1. Значение визуализации значимости признаков

Визуализация значимости признаков помогает:

- Интерпретировать вклад признаков в предсказания модели.
- Выявлять наиболее и наименее важные признаки.
- Понимать, как изменение значений признаков влияет на целевую переменную.

Графики значимости могут быть полезны как на этапе разработки моделей, так и при представлении результатов для заинтересованных сторон.

3.2. Методы визуализации значимости признаков

3.2.1. Гистограммы и столбчатые диаграммы

- Самый простой способ визуализации.
- Основаны на значениях важности, предоставляемых моделями, например:
- Коэффициенты линейных моделей.
- Значения атрибута feature_importances_ для деревьев решений.
- Признаки сортируются по убыванию значимости.

Пример гистограммы и столбчатой диаграммы

Пример приведен в виде кода в листинге 3.2.1.1.

Листинг 3.2.1.1 – Гистограммы и столбчатые диаграммы

```
import matplotlib.pyplot as plt
import numpy as np
# Пример данных
features = ['feature1', 'feature2', 'feature3']
importances = [0.8, 0.5, 0.2]
indices = np.argsort(importances)[::-1]
# Визуализация
plt.figure()
plt.title("Feature Importances")
plt.bar(range(len(importances)),
np.array(importances)[indices], align="center")
plt.xticks(range(len(importances)),
np.array(features)[indices])
plt.show()
```

3.2.2. Графики SHAP (SHAP Summary Plot)

- Глобальный график значимости признаков для всей модели.
- Показывает, как каждый признак влияет на предсказания.
- Основывается на концепции SHAP-значений, которые отражают средний вклад признака.

3.2.2.1 Типы графиков SHAP:

1. **Summary Plot**:
- Отображает значимость всех признаков.
- Визуализирует распределение влияния каждого признака с помощью цветовой схемы.
2. **Force Plot**:
- Демонстрирует локальную интерпретацию: как каждый признак влияет на конкретное предсказание.
3. **Dependence Plot**:
- Показывает взаимосвязь между значением признака и его

влиянием на предсказание.

3.2.2.2 Пример Summary Plot

В листинге 3.2.3.1 приведен пример кода для создания Summary Plot

Листинг 3.2.3.1 – Summary Plot

```
import shap
shap.summary_plot(shap_values, X)  # X — матрица признаков
```

3.3.3. Графики важности признаков в моделях деревьев

- Основаны на атрибуте feature_importances_ в таких моделях, как Random Forest или Gradient Boosting.
- Сортировка признаков по их важности и построение столбчатого графика.

3.3.4. Боксплоты для анализа распределения значимости

- Используются для анализа распределения важности признаков в ансамблях деревьев.

3.3 Особенности визуализации для разных моделей

Таблица 3.3.1 – Особенности разных моделей

Наименование модели	Описание
Линейные модели	Графики с положительными и отрицательными коэффициентами. Гистограммы или горизонтальные столбчатые диаграммы для наглядности.

Продолжение таблицы 3.3.1

Деревья решений	Диаграммы на основе feature_importances_. Возможность включения межпризнаковых взаимодействий через SHAP.
Градиентный бустинг	Диаграммы значимости на основе частоты использования признаков. Применение SHAP для более глубокой интерпретации.
Модели с регуляризацией	Lasso: Показывает зануленные признаки при увеличении параметра α Ridge: Демонстрирует сужение значений коэффициентов.

3.4. Преимущества визуализации значимости признаков

Интерпретируемость: Упрощает понимание моделей для пользователей без глубоких технических знаний.

Сравнение: Позволяет оценить вклад каждого признака в различных моделях.

Принятие решений: Облегчает выбор признаков для дальнейшей работы или их исключение.

3.5. Рекомендации по построению визуализации

1. **Сортировка признаков**:

- Сортировать признаки по значимости для наглядности.
- Выделять наиболее важные признаки цветом или стилем графика.

2. **Адаптация к аудитории**:

- Для технической аудитории используйте подробные графики, такие как SHAP summary.
- Для пользователей — простые столбчатые диаграммы или гистограммы.

3. **Подписи и пояснения**:

- Добавляйте заголовки, оси и пояснения к графикам.
- Используйте легенды для различия положительного и отрицательного влияния.

3.6. Выводы по разделу

Визуализация значимости признаков — важный инструмент для анализа моделей машинного обучения. Она не только позволяет понять, какие признаки оказывают наибольшее влияние, но и помогает в выборе оптимального набора признаков, улучшении интерпретации модели и подготовке отчётов.

СРАВНЕНИЕ КАЧЕСТВА МОДЕЛЕЙ

4.1. Зачем сравнивать качество моделей

Сравнение качества моделей позволяет определить, какая из них лучше всего подходит для решения конкретной задачи. Качество модели зависит от её способности:

- **Точно предсказывать целевую переменную**.
- **Обобщать результаты** на новых данных (не переобучаться).
- **Обеспечивать интерпретируемость и устойчивость к шуму** в данных.

Цель этого этапа — оценить влияние отбора признаков на качество модели и выбрать оптимальную конфигурацию признаков для обучения.

4.2. Метрики оценки качества моделей

4.2.1 Для задач классификации:

1. **Accuracy** (точность):
- Процент правильных предсказаний от общего числа примеров.
- Хорошо работает для сбалансированных данных.
2. **Precision** (точность положительного класса)(Формула представлена под номером 4.2.1.1:
- Доля истинно положительных предсказаний среди всех, которые модель классифицировала как положительные.

$$Precision = \frac{TP}{TP + FP} \qquad (4.2.1.1)$$

3. **Recall** (полнота)(Формула представлена под номером 4.2.1.2):

- Доля истинно положительных примеров, которые модель правильно классифицировала.

$$Recall = \frac{TP}{TP + FN} \quad (4.2.1.2)$$

4. **F1-score** (Формула представлена под номером 4.2.1.3):

- Гармоническое среднее между Precision и Recall. Используется, когда важно учитывать баланс между этими метриками.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4.2.1.3)$$

5. **ROC-AUC**:

- Площадь под кривой ROC (Receiver Operating Characteristic). Показывает соотношение True Positive Rate (TPR) и False Positive Rate (FPR).

4.2.2 Для задач регрессии:

1. **Mean Squared Error (MSE)**:

- Среднеквадратичная ошибка. (Формула представлена под номером 4.2.2.1)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (4.2.2.1)$$

- Чем меньше MSE, тем лучше.

2. **Root Mean Squared Error (RMSE)** (Формула представлена под номером 4.2.2.2):

- Корень из MSE. Облегчает интерпретацию в единицах целевой переменной.

$$RMSE = \sqrt{MSE} \quad (4.2.2.2)$$

3. **Mean Absolute Error (MAE)**(Формула представлена под номером 4.2.2.3):

- Средняя абсолютная ошибка.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \qquad (4.2.2.3)$$

- Лучше всего подходит для данных с выбросами.

4. **R-squared (R^2)**(Формула представлена под номером 4.2.2.4):

- Показывает долю объяснённой вариации целевой переменной моделью.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum i = 1^n (y_i - \bar{y})^2} \qquad (4.2.2.4)$$

4.3. Подходы к сравнению моделей

4.3.1. Модели на всех признаках:

- Модель обучается на полном наборе признаков.
- Результаты служат базой для сравнения.

4.3.2. Модели на отобранных признаках:

- Используются только значимые признаки, отобранные ранее (например, с помощью Lasso, Random Forest или SHAP).
- Сравнивается метрика качества с результатами базовой модели.

4.3.3. Модели на преобразованных признаках (PCA):

- Модель обучается на главных компонентах, выделенных методом PCA.
- Это позволяет уменьшить размерность данных и проверить, как это влияет на качество.

4.3.4. Ансамблевые методы:

- Сравнение ансамблей (например, случайный лес против градиентного бустинга).
- Использование метрик важности признаков для уточнения моделей.

4.4. Проблемы при сравнении моделей

Таблица 4.4.1 – Проблемы моделей

Наименование проблемы	Описание	Решение
Переобучение	Если модель слишком сложна, она может показать высокую точность на обучающих данных, но низкую на тестовых.	Кросс-валидация
Несбалансированные классы	Высокий accuracy может скрывать плохую работу с миноритарным классом.	Использовать Precision, Recall, F1-score или ROC-AUC.
Избыточные признаки	Использование всех признаков может привести к ухудшению обобщающей способности модели	Отбор наиболее значимых признаков
Выбор метрики	Неправильная метрика может исказить результаты	Выбирать метрику, подходящую для задачи

4.5. Как проводить оценку моделей на практике

1. **Разделение данных**:
- Разделите данные на обучающую и тестовую выборки.
- Оптимально использовать разбиение 70/30 или 80/20.
2. **Кросс-валидация**:
- Разделите данные на k-блоков и обучите модель на k-1 блоках, оставив один для тестирования.
- Итоговая метрика — среднее значение по всем блокам.
3. **Сравнение моделей**:
- Для каждой модели вычислите выбранные метрики.
- Сравните результаты на всех признаках, отобранных признаках и признаках после PCA.
4. **Интерпретация результатов**:
- Используйте визуализацию, чтобы показать разницу в метриках для разных моделей (например, графики ROC, боксплоты ошибок).

4.6. Выводы по сравнению моделей

Сравнение моделей позволяет выбрать оптимальную комбинацию алгоритма и набора признаков.

Использование различных подходов к отбору признаков и снижению размерности позволяет улучшить качество модели и её интерпретируемость.

Итоговая модель должна показывать хорошее качество как на обучающей, так и на тестовой выборках.

СНИЖЕНИЕ РАЗМЕРНОСТИ И ВИЗУАЛИЗАЦИЯ ДАННЫХ

5.1 Зачем снижать размерность данных

Снижение размерности данных позволяет:

- Упростить модели за счёт уменьшения количества признаков.
- Уменьшить вычислительную сложность.
- Устранить избыточные или скоррелированные признаки.
- Улучшить интерпретируемость данных.
- Визуализировать многомерные данные в 2D или 3D для их лучшего понимания.

Снижение размерности особенно полезно для обработки данных с большим количеством признаков, где часть из них может быть нерелевантной или избыточной.

5.2. Методы снижения размерности

5.2.1. Principal Component Analysis (PCA) — метод главных компонент

PCA преобразует исходные признаки в новое пространство меньшей размерности, сохраняя при этом максимальную долю дисперсии данных.

1. **Основные этапы PCA**:

- Центрирование данных (вычитание среднего значения для каждого признака).

- Вычисление ковариационной матрицы.
- Нахождение собственных векторов и собственных значений ковариационной матрицы.
- Отбор главных компонент (собственных векторов) с наибольшими собственными значениями.

2. **Особенности PCA**:

- Главные компоненты ортогональны (некоррелированы).
- Снижение размерности осуществляется за счёт выбора первых k компонент, которые объясняют большую часть дисперсии.

3. **Преимущества PCA**:

- Простота реализации.
- Сохранение максимальной информативности данных.

4. **Ограничения**:

- Не учитывает нелинейные зависимости.
- Преобразует данные, что затрудняет интерпретацию исходных признаков.

5.2.1.1. Пример использования PCA:

Листинг 5.2.1.1.1 – Использование PCA

```
from sklearn.decomposition import PCA
pca = PCA(n_components=2)  # Снижение до 2 компонент
X_pca = pca.fit_transform(X)
```

5.2.2. t-SNE — Stochastic Neighbor Embedding

t-SNE — метод для визуализации данных в низкоразмерном пространстве, который сохраняет локальную структуру данных.

1. **Как работает t-SNE**:

- Вычисляет вероятности близости между объектами в исходном пространстве.
- Проецирует данные в низкоразмерное пространство, минимизируя расхождение между вероятностями в исходном и целевом пространстве (по дивергенции Кульбака-Лейблера).

2. **Особенности t-SNE**:

- Хорошо подходит для визуализации кластеров в данных.
- Сохраняет локальные расстояния, но может искажать глобальную структуру.

3. **Ограничения t-SNE**:

- Высокая вычислительная сложность.
- Результаты могут меняться при повторных запусках.

5.2.2.1. Пример использования t-SNE:

Листинг 5.2.2.1.1 – Пример t-SNE

```
from sklearn.manifold import TSNE
tsne = TSNE(n_components=2, perplexity=30)
X_tsne = tsne.fit_transform(X)
```

5.2.3. UMAP — Uniform Manifold Approximation and Projection

UMAP — современный метод снижения размерности, который сохраняет как локальную, так и глобальную структуру данных.

1. **Принципы работы UMAP**:

- Создаёт граф ближайших соседей в исходном пространстве.
- Проецирует граф в низкоразмерное пространство, оптимизируя расстояния.

2. **Особенности UMAP**:

- Быстрее, чем t-SNE.

- Лучше сохраняет глобальную структуру данных.

3. **Ограничения UMAP**:

- Параметры модели (например, расстояние и количество соседей) могут сильно влиять на результаты.
- Требует предварительной настройки.

5.2.3.1. Пример использования UMAP

Листинг 5.2.3.1.1 – Пример UMAP

```
import umap
umap_model = umap.UMAP(n_components=2)
X_umap = umap_model.fit_transform(X)
```

5.3. Применение метода снижения размерности

1. **Выбор метода**:

- Для интерпретации данных и сохранения структуры: PCA.
- Для визуализации кластеров: t-SNE или UMAP.

2. **Проверка сохранённой информации**:

- При снижении размерности важно оценить, насколько хорошо объясняется дисперсия данных (например, для PCA — доля объяснённой дисперсии).

3. **Визуализация данных**:

- Визуализация снижения размерности помогает обнаружить скрытые структуры в данных, такие как кластеры или аномалии.

5.4. Визуализация данных

1. **2D и 3D графики**:

- Снижение размерности до 2 или 3 компонент позволяет визуализировать данные в понятной форме.
- Используются для предварительного анализа данных.

2. **Инструменты для визуализации**:

- Matplotlib или Seaborn для 2D-графиков.
- Plotly для интерактивных графиков.

5.4.1. Пример визуализации с PCA

Листинг 5.4.1.1 – Пример визуализации с PCA

```
import matplotlib.pyplot as plt
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=labels)
plt.title("PCA Visualization")
plt.show()
```

5.4.2. Пример визуализации с t-SNE

Листинг 5.4.2.1 – Пример визуализации с t-SNE

```
plt.scatter(X_tsne[:, 0], X_tsne[:, 1], c=labels)
plt.title("t-SNE Visualization")
plt.show()
```

5.5. Преимущества и ограничения методов

Таблица 5.5.1 – Преимущества и ограничения методов

Метод	Преимущество	Ограничения
PCA	Быстро, сохраняет глобальную структуру	Линейность, трудности интерпретации

Продолжение таблицы 5.5.1

t-SNE	Хорошо визуализирует кластеры	Медленно, переменные результаты
UMAP	Быстро, сохраняет локальную структуру	Требует настройки параметров

5.6. Рекомендации по снижению размерности

1. **Определите цель**:
- Если нужно снизить вычислительную сложность — используйте PCA.
- Если нужна визуализация данных — используйте t-SNE или UMAP.
2. **Проверяйте объяснённую дисперсию**:
- Для PCA убедитесь, что выбранное число компонент объясняет большую часть дисперсии (например, 90%).
3. **Анализируйте результаты**:
- Проверяйте, не потерялись ли важные зависимости в данных.
4. **Используйте визуализацию**:
- Помогите понять данные не только себе, но и другим участникам проекта.