|  |
| --- |
|  |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования  **"МИРЭА - Российский технологический университет"**  **РТУ МИРЭА** |
| Институт информационных технологий (ИИТ) |
| Кафедра прикладной математики (ПМ) |

|  |  |
| --- | --- |
| **ОТЧЕТ**  **ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 2** | |
| **по дисциплине** |  |
| **«Технологии и инструментарий анализа больших данных»** | |
| Выполнил студент группы ИКБО-20-22 | Шумахер М.Е. |
| Принял преподаватель кафедры ПМ | Горячев А.А. |

|  |
| --- |
|  |
|  |

Москва 2025

1. Найти и выгрузить многомерные данные (с большим количеством признаков – столбцов) с использованием библиотеки pandas. В отчёте описать найденные данные.

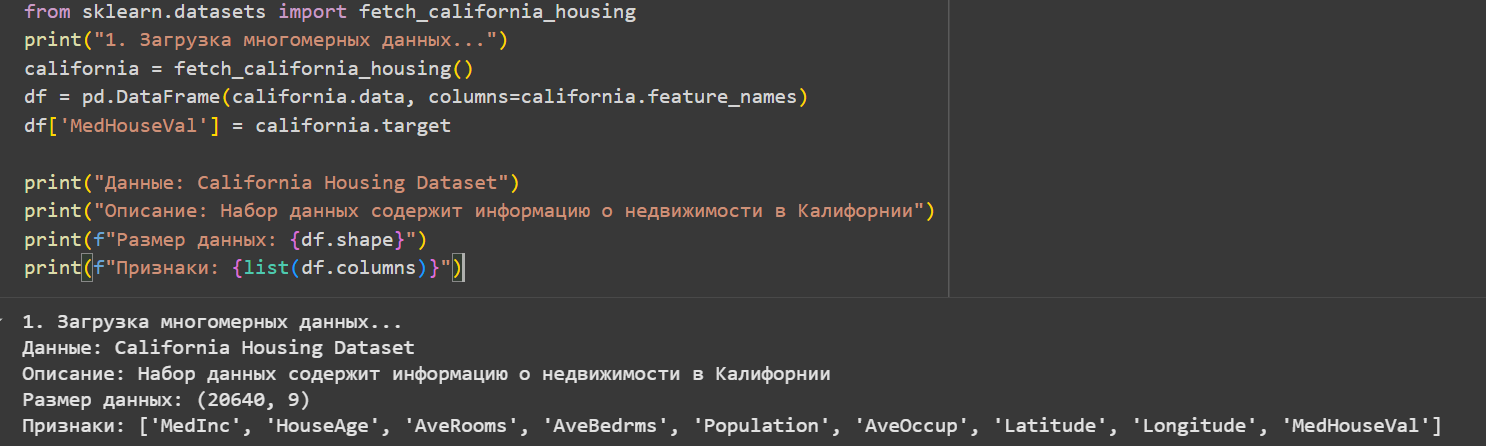


Рисунок 1 – Код и результат программы

Описание признаков:

MedInc - медианный доход населения

HouseAge - медианный возраст домов

AveRooms - среднее количество комнат

AveBedrms - среднее количество спален

Population - население

AveOccup - средняя занятость

Latitude - широта

Longitude - долгота

MedHouseVal - медианная стоимость дома (целевая переменная)

2. Вывести информацию о данных при помощи методов .info(), .head(). Проверить данные на наличие пустых значений. В случае их наличия удалить данные строки или интерполировать пропущенные значения. При необходимости дополнительно предобработать данные для дальнейшей работы с ними.

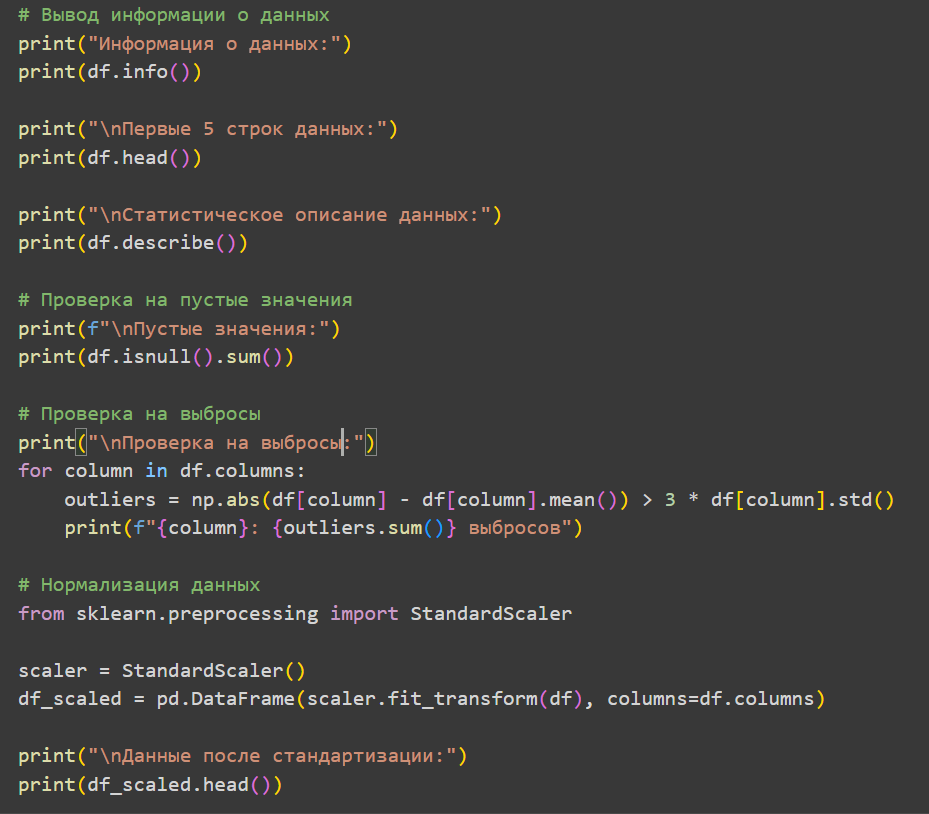


Рисунок 2 – Код решения задачи

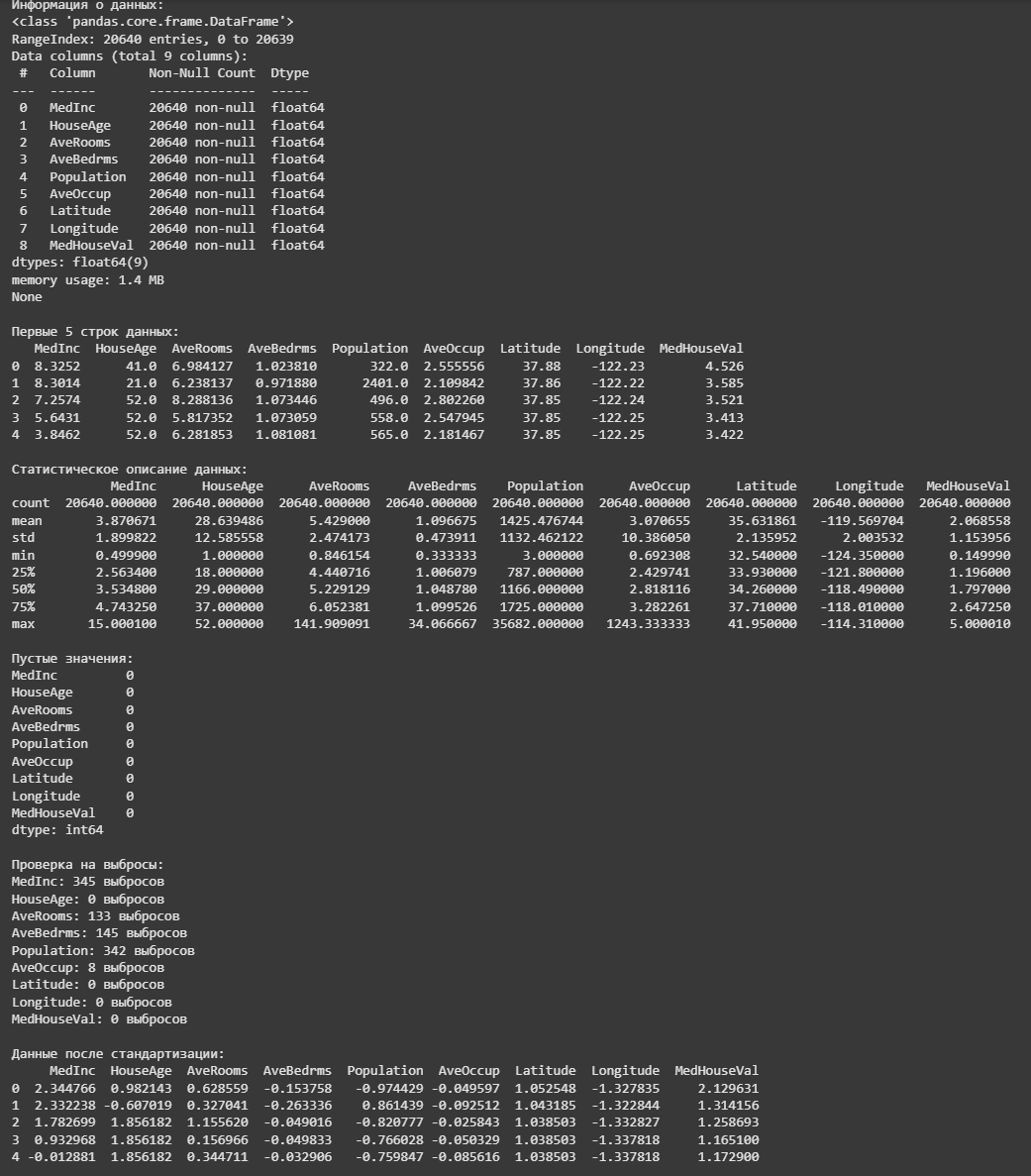


Рисунок 3 – Результат работы кода

3. Построить столбчатую диаграмму (.bar) с использованием модуля graph\_objs из библиотеки Plotly со следующими параметрами:

3.1. По оси Х указать дату или название, по оси У указать количественный показатель.

3.2. Сделать так, чтобы столбец принимал цвет в зависимости от значения показателя (marker=dict(color=признак, coloraxis="coloraxis")).

3.3. Сделать так, чтобы границы каждого столбца были выделены чёрной линией с толщиной равной 2.

3.4. Отобразить заголовок диаграммы, разместив его по центру сверху, с 20 размером текста.

3.5. Добавить подписи для осей X и Y с размером текста, равным 16. Для оси абсцисс развернуть метки так, чтобы они читались под углом, равным 315.

3.6. Размер текста меток осей сделать равным 14.

3.7. Расположить график во всю ширину рабочей области и присвоить высоту, равную 700 пикселей.

3.8. Добавить сетку на график, сделать её цвет 'ivory' и толщину равную 2. (Можно сделать это при настройке осей с помощью gridwidth=2, gridcolor='ivory')

3.9. Убрать лишние отступы по краям.



Рисунок 4 – Код решения задачи

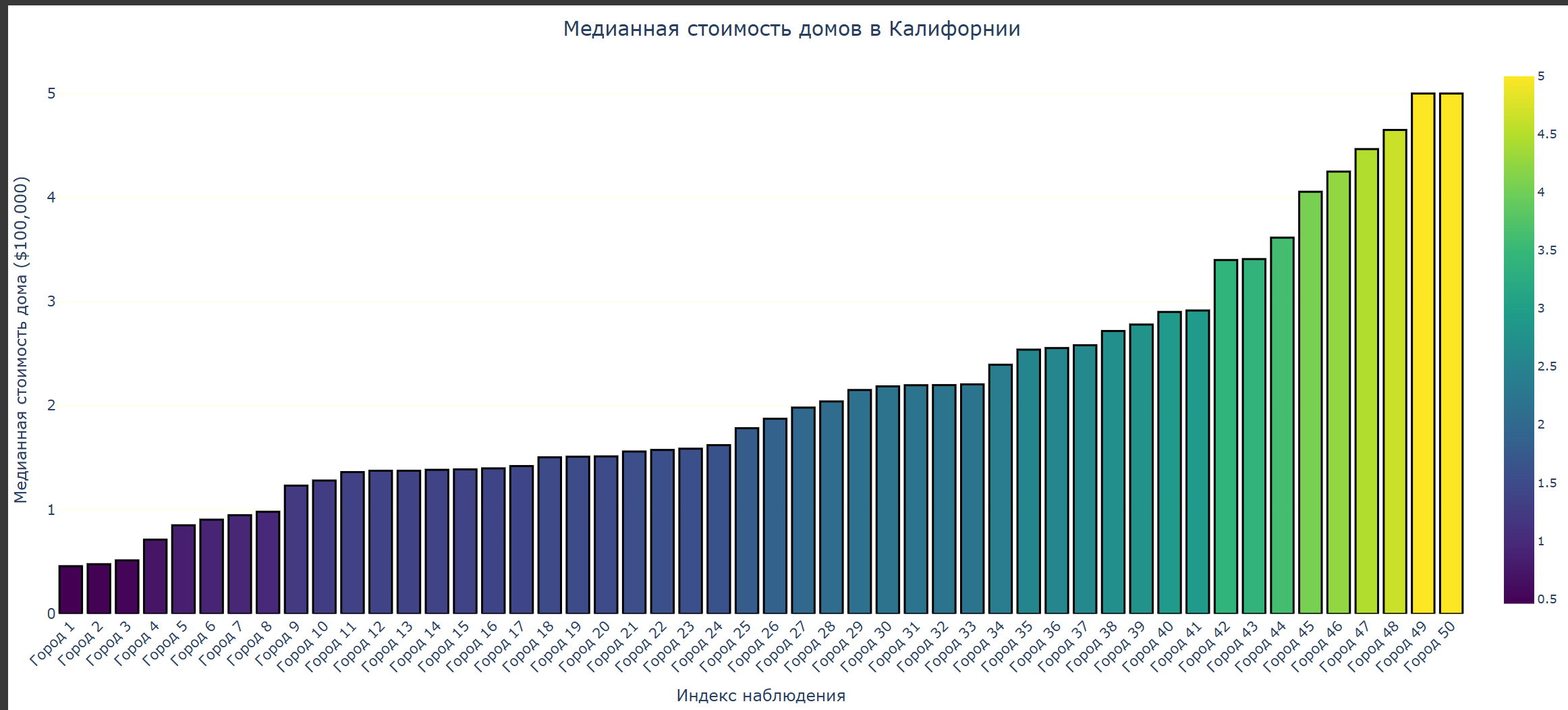


Рисунок 5 – Результат работы кода

4. Построить круговую диаграмму (go.Pie), использовав данные и стиль оформления из предыдущего графика. Сделать так, чтобы границы каждой доли были выделены чёрной линией с толщиной, равной 2 и категории круговой диаграммы были читаемы (к примеру, объединить часть объектов).

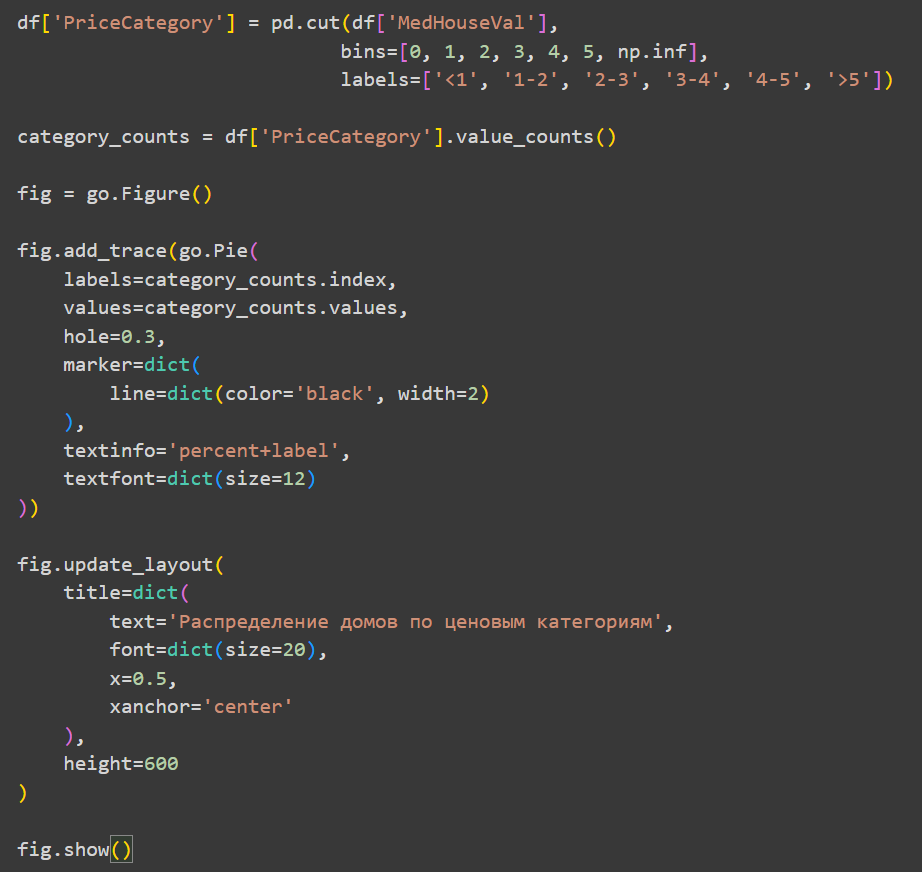


Рисунок 6 – Код решения задачи



Рисунок 7 – Результат работы кода

5. Построить линейные графики, взять один из параметров и определить зависимость между другими несколькими (от 2 до 5) показателями с использованием библиотеки matplotlib. Сделать вывод.



Рисунок 8 – Код решения задачи

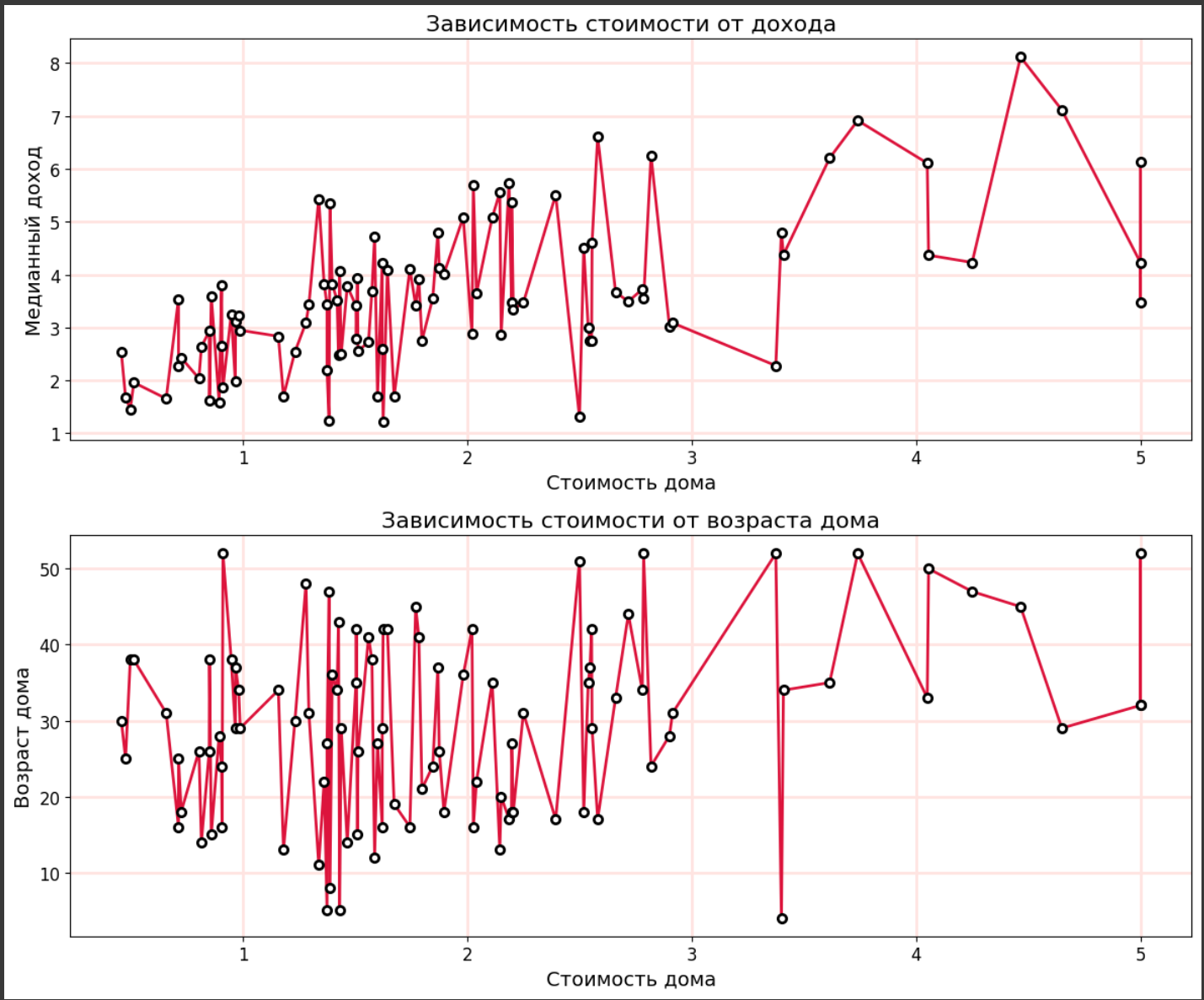


Рисунок 9 – Результат работы кода

Выводы по линейным графикам:

* Наблюдается положительная корреляция между доходом и стоимостью домов
* Возраст домов показывает слабую корреляцию со стоимостью

6. Выполнить визуализацию многомерных данных, используя t-SNE. Необходимо использовать набор данных MNIST или fashion MNIST (можно использовать и другие готовые наборы данных, где можно наблюдать разделение объектов по кластерам). Рассмотреть результаты визуализации для разных значений перплексии.

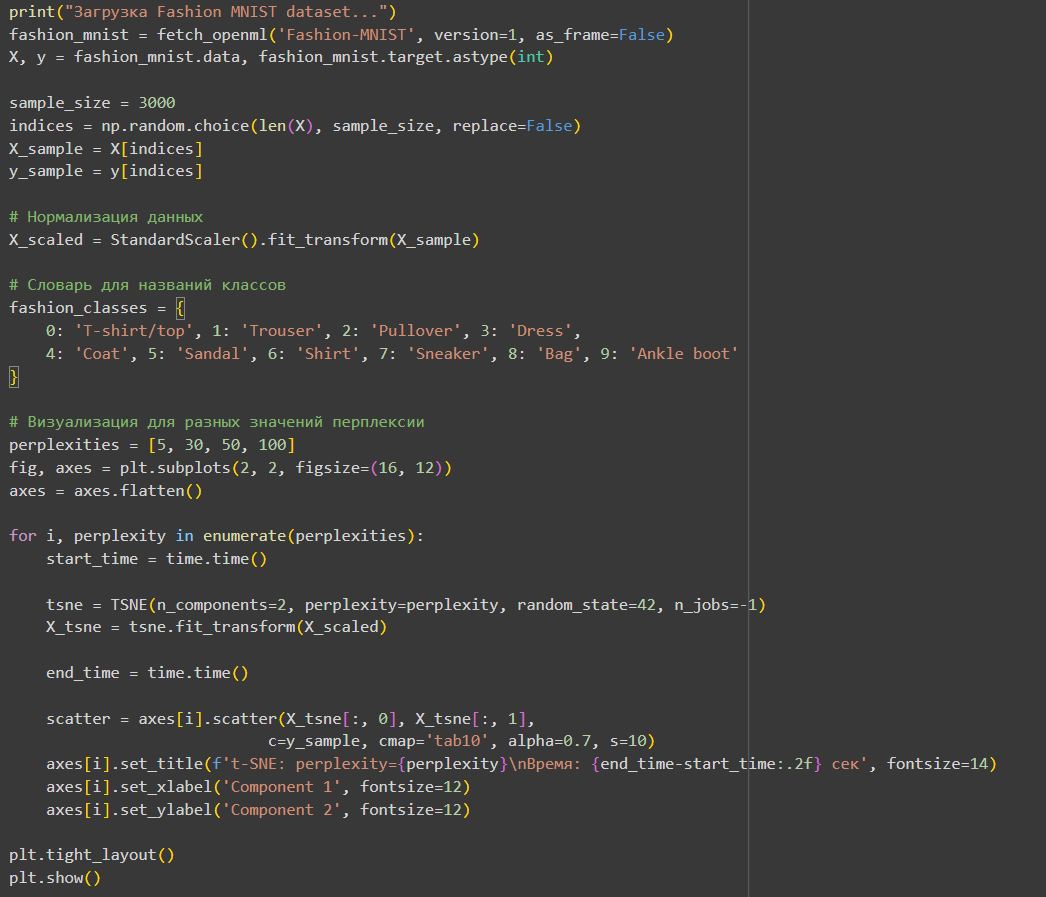


Рисунок 10 – Код решения задачи

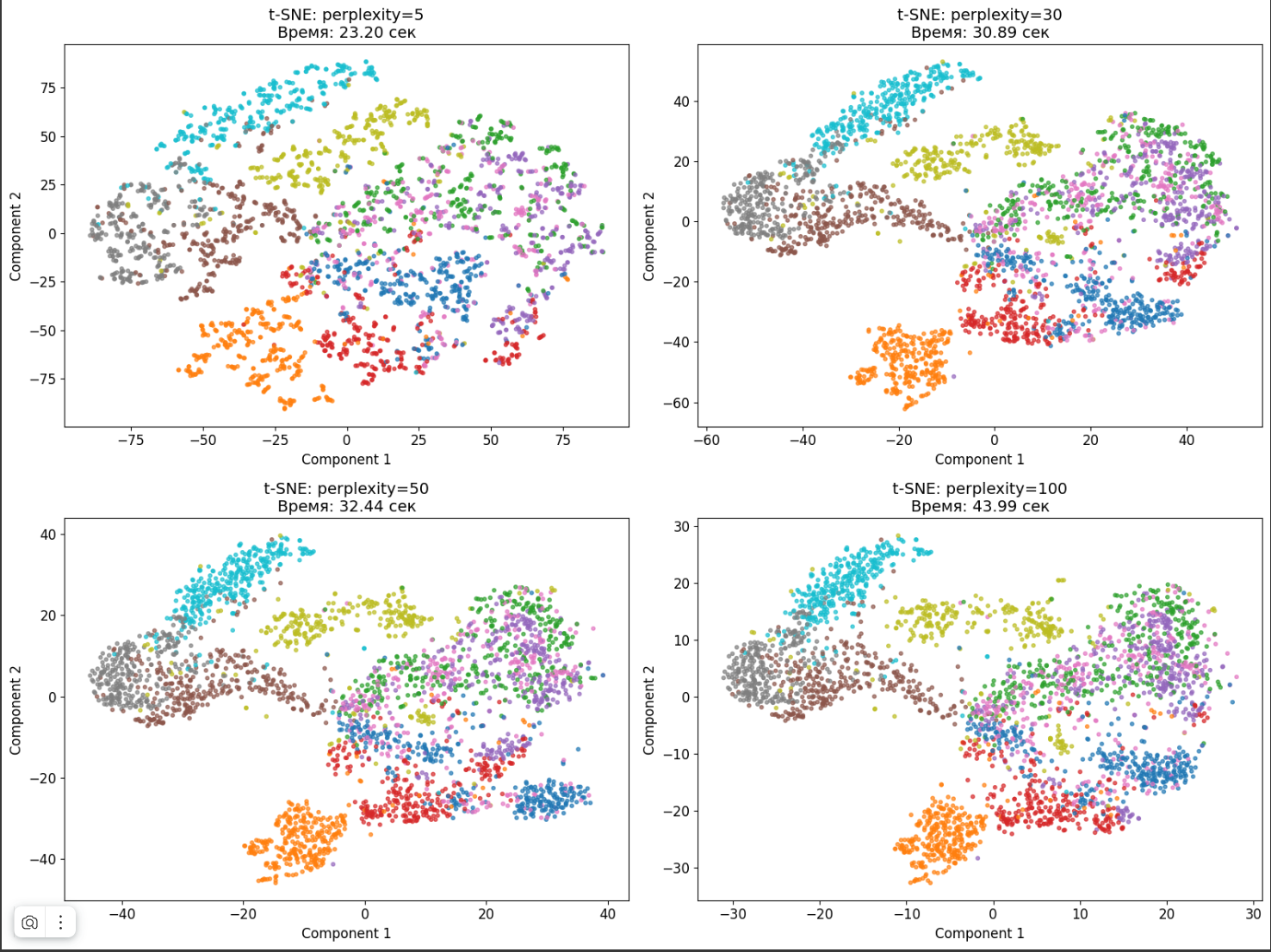


Рисунок 11 – Результат работы кода

Выводы по t-SNE:

* Perplexity=5: Маленькие изолированные кластеры, возможна переобучение
* Perplexity=30: Хорошее разделение кластеров, оптимальный баланс
* Perplexity=50: Более крупные кластеры, но хуже разделение
* Perplexity=100: Кластеры начинают сливаться

7. Выполнить визуализацию многомерных данных, используя UMAP с различными параметрами n\_neighbors и min\_dist. Рассчитать время работы алгоритма с помощью библиотеки time и сравнить его с временем работы t-SNE.

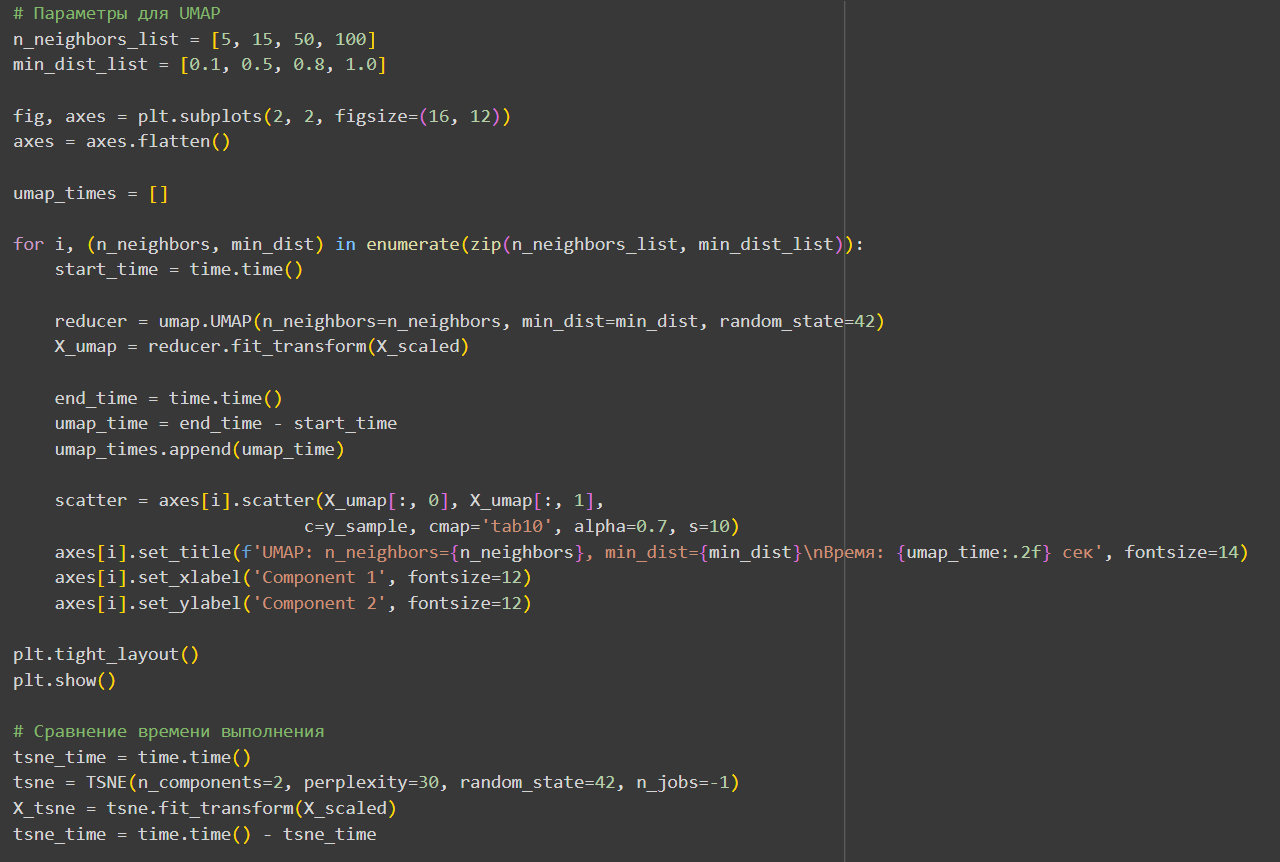


Рисунок 12 – Код решения задачи

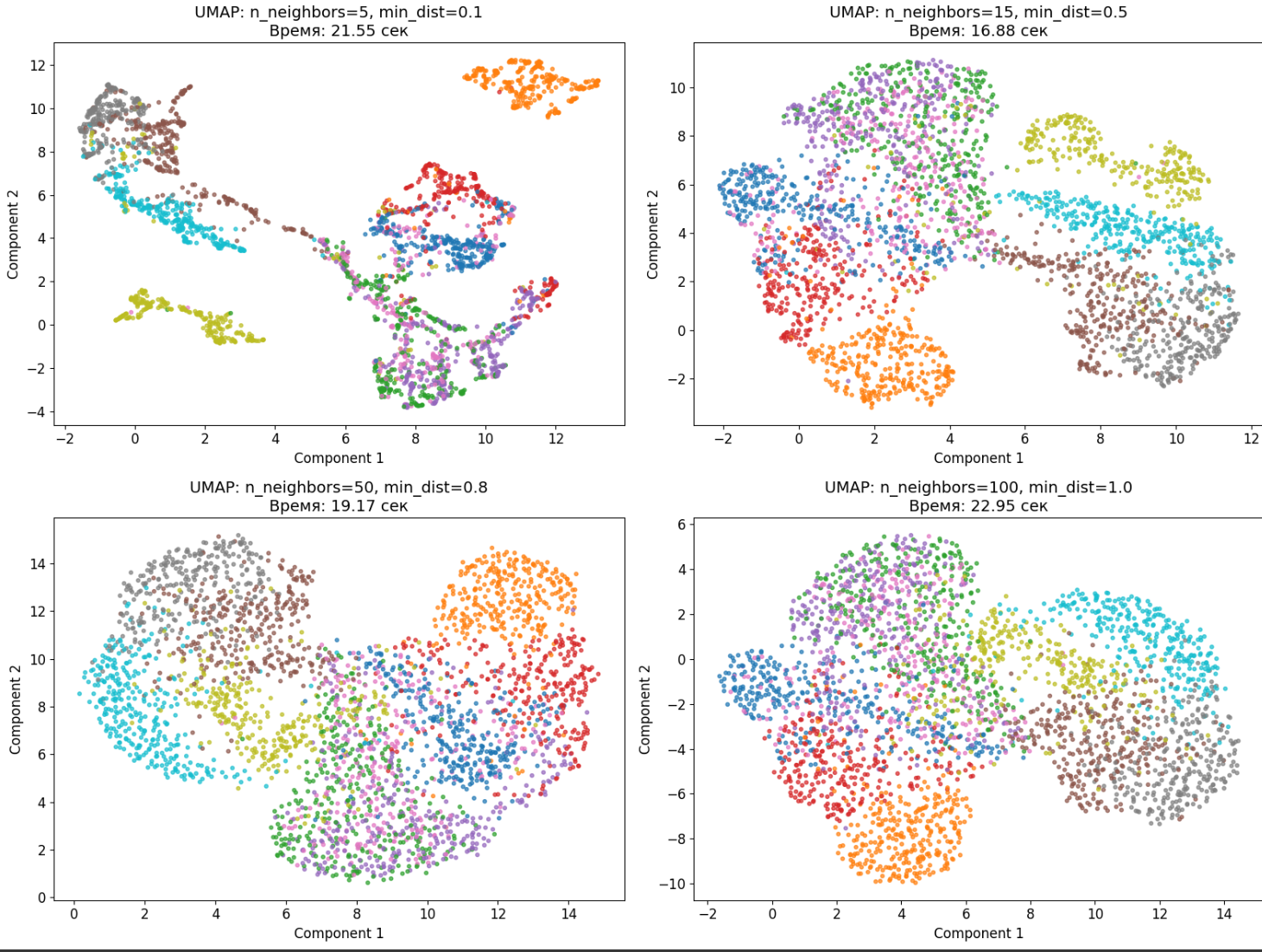


Рисунок 13 – Результат работы кода

Сравнение времени выполнения:

* t-SNE (perplexity=30): 27.65 секунд
* UMAP (в среднем): 20.14 секунд
* UMAP быстрее t-SNE в 1.4 раз

Выводы по UMAP:

* n\_neighbors=5: Мелкие локальные структуры, возможен шум
* n\_neighbors=15: Хороший баланс между локальными и глобальными структурами
* n\_neighbors=50: Акцентирует глобальную структуру данных
* n\_neighbors=100: Максимальное сохранение глобальной структуры