|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА** – **Российский технологический университет»**  **РТУ МИРЭА** | | | | |
| Институт информационных технологий (ИТ) | |
| Кафедра прикладной математики | |

|  |  |
| --- | --- |
| **ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 2** | |
| **по дисциплине «Технологии и инструментарий анализа больших данных»** | |
|  | |
| Выполнил студент группы ИКБО-20-21 | Фомичев Р.А. |
| Проверил ассистент кафедры ПМ ИИТ | Тетерин Н.Н. |

**Задание 1**

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

В качестве многомерных данных будем использовать набор данных о самых прослушиваемых треках в Spotify за 2023 год. Данные можно получить по ссылке: https://www.kaggle.com/datasets/nelgiriyewithana/top-spotify-songs-2023.

**Задание 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. Код представлен на рисунке 1.

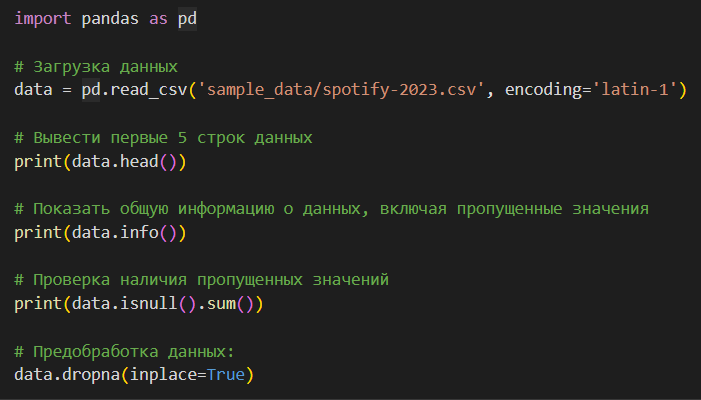


Рисунок 1 – Код

Результат работы программы представлен на рисунках 2-3.

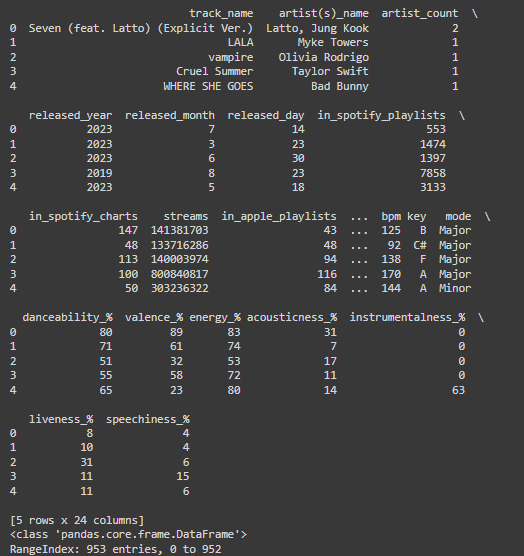


Рисунок 2 – Результат работы программы

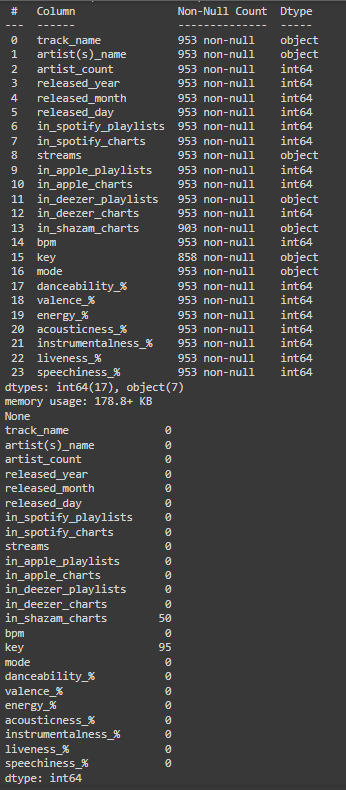


Рисунок 3 – Результат работы программы

**Задание 3**

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

* По оси Х указать дату или название, по оси У указать количественный показатель.
* Сделать так, чтобы столбец принимал цвет в зависимости от значения показателя (marker=dict(color=признак, coloraxis="coloraxis")).
* Сделать так, чтобы границы каждого столбца были выделены чёрной линией с толщиной равной 2.
* Отобразить заголовок диаграммы, разместив его по центру сверху, с 20 размером текста.
* 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.
* Размер текста меток осей сделать равным 14.
* Расположить график во всю ширину рабочей области и присвоить высоту, равную 700 пикселей.
* Добавить сетку на график, сделать её цвет 'ivory' и толщину равную 2. (Можно сделать это при настройке осей с помощью gridwidth=2, gridcolor='ivory')
* Убрать лишние отступы по краям.

Код представлен на рисунке 4.

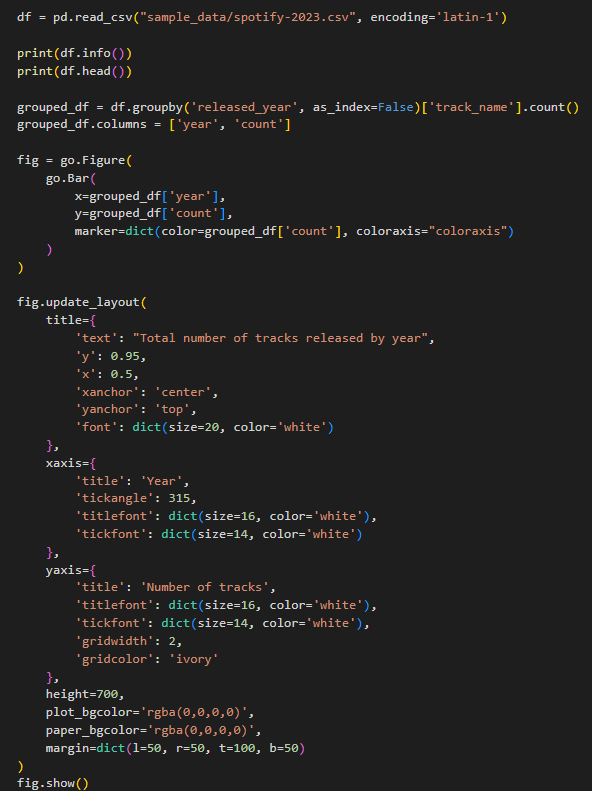


Рисунок 4 – Код

Результат работы программы представлен на рисунке 5.

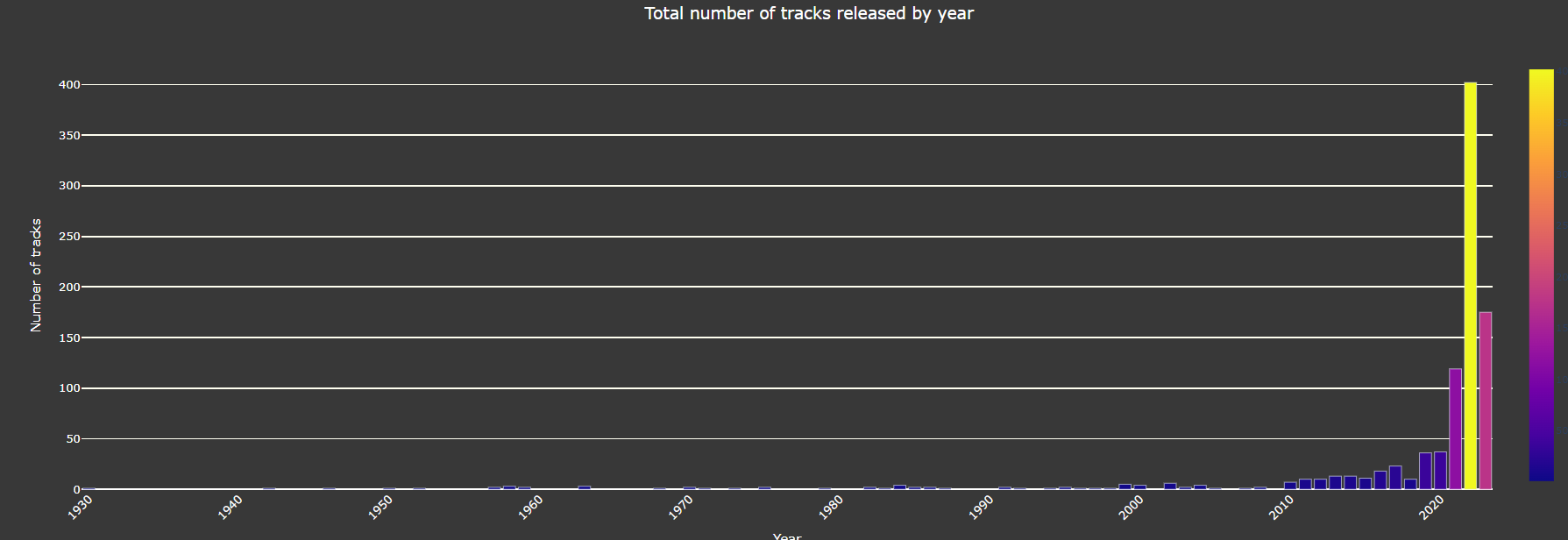


Рисунок 5 – Результат работы программы

**Задание 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)

Напишем код, который отражает количество треков, находящихся в плейлистах на платформе Spotify (начиная от 25 000). Код представлен на рисунке 6.

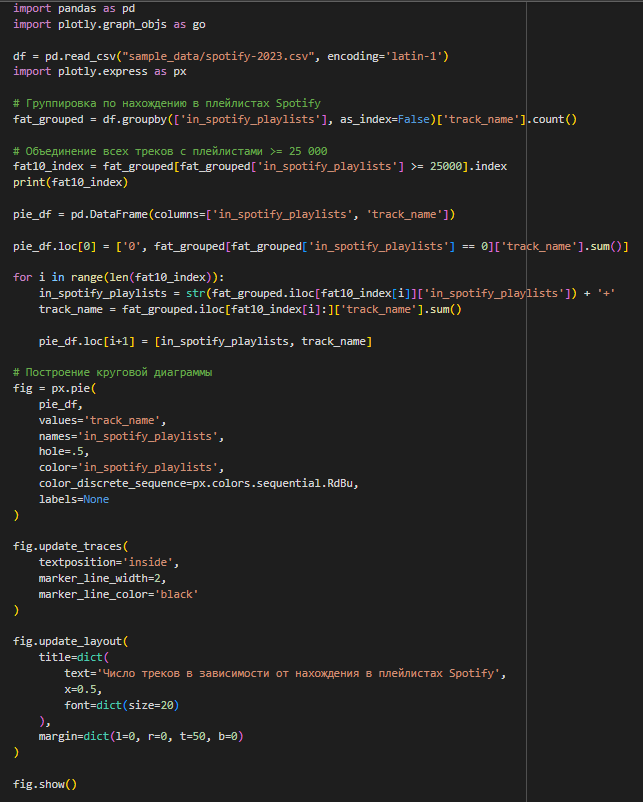


Рисунок 6 – Код

Результат работы программы представлен на рисунке 7.

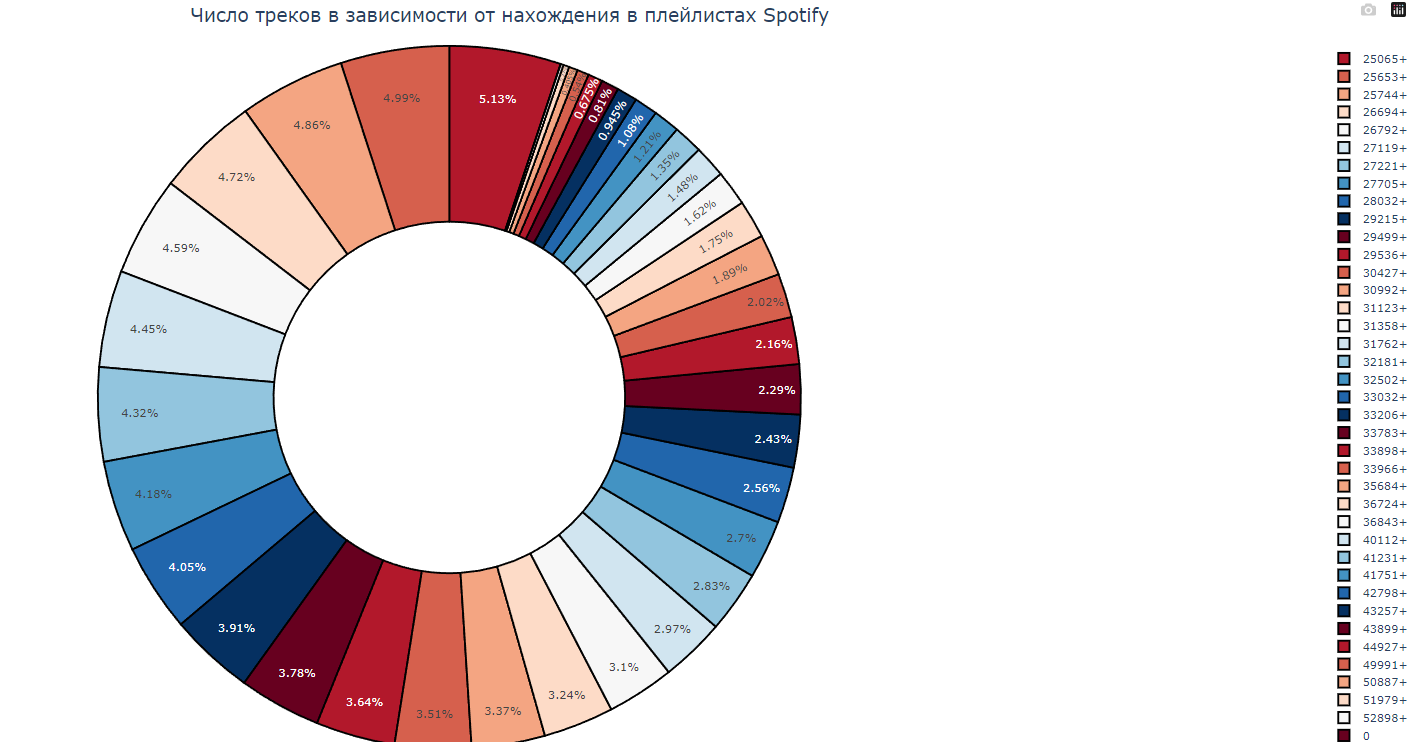


Рисунок 7 – результат работы программа

**Задание 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.

* Сделать график с линиями и маркерами, цвет линии 'crimson', цвет точек 'white', цвет границ точек 'black', толщина границ точек равна 2.
* Добавить сетку на график, сделать её цвет 'mistyrose' и толщину равную 2. (Можно сделать это при настройке осей с помощью linewidth=2, color='mistyrose').

Построим линейный график, чтобы узнать зависимость числа релизов от года. Код представлен на рисунке 8.

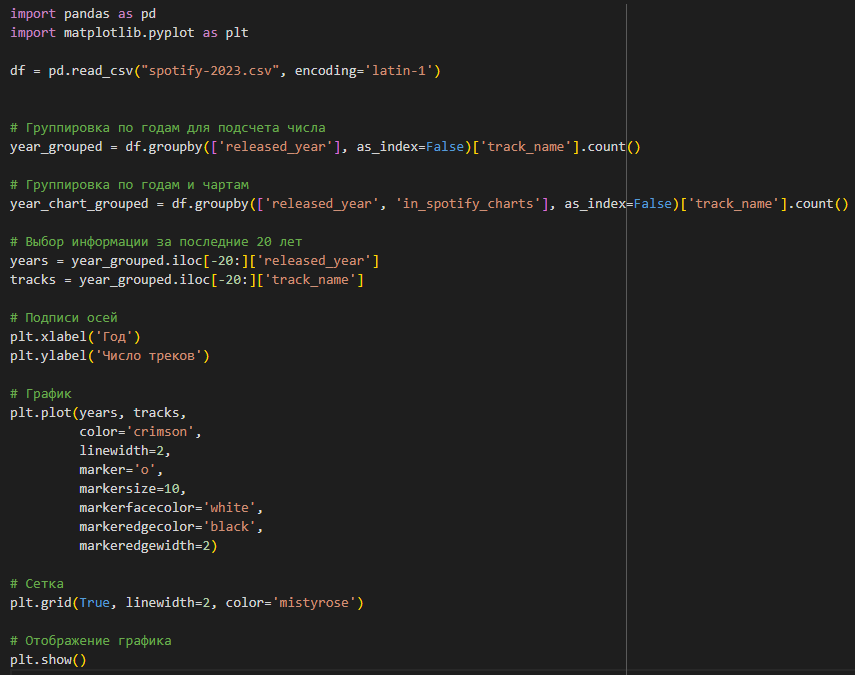


Рисунок 8 – Код

Результат работы программы представлен на рисунке 9.

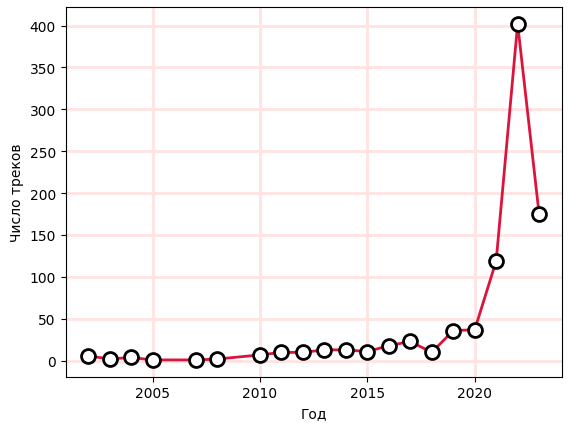


Рисунок 9 – Результат работы программы

Проанализировав получившийся график, можно сделать вывод, что рост количества, выпущенных треков происходит после 2020 года.

Построим еще один линейный график, чтобы узнать зависимость числа релизов треков от месяца. Код представлен на рисунке 10.

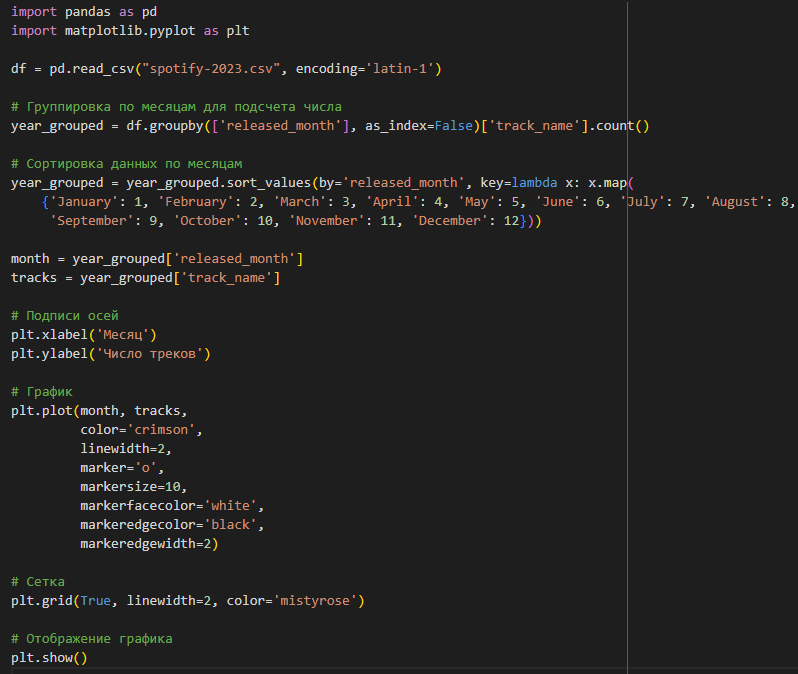


Рисунок 10 – Код

Результат работы программы представлен на рисунке 11.

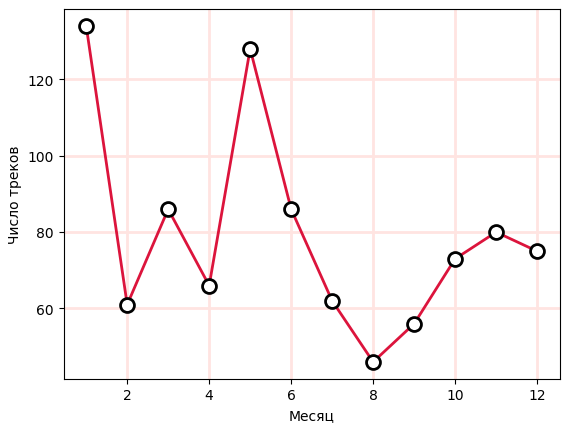


Рисунок 11 – Результат работы программы

Проанализировав получившийся график, можно сделать вывод, что самые популярные песни в большинстве пишутся либо в январе, либо в мае. **Задание 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.

Напишем код, который строит график набора данных MNIST, содержащий изображения рукописных цифр от 0 до 9. Код представлен на рисунке 12.

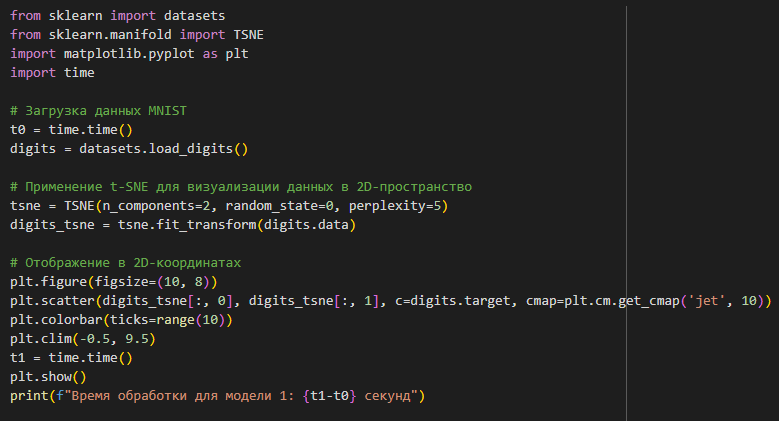


Рисунок 12 – Код

Результат работы программы представлен на рисунке 13.

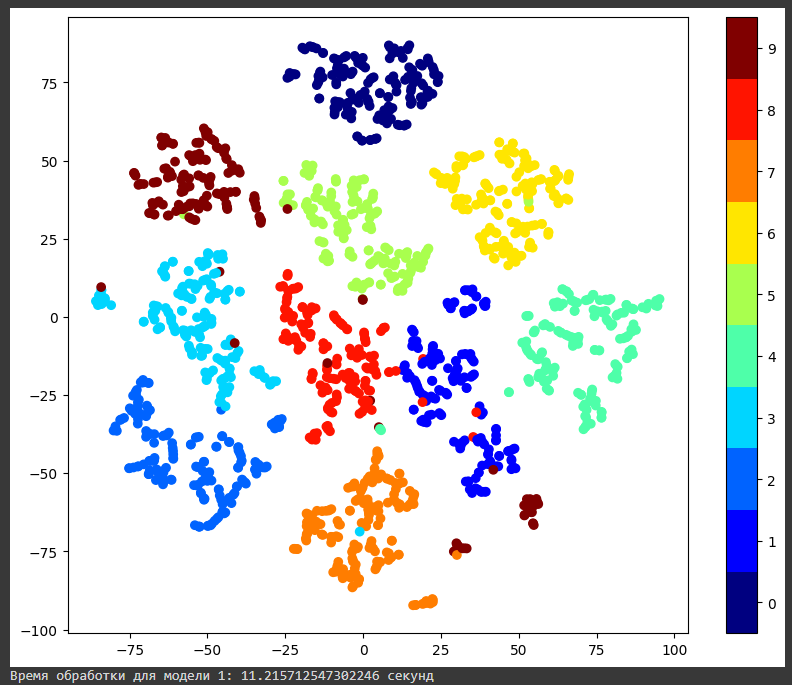


Рисунок 13 – Результат работы программы

Изменим значение перплексии с 5 на 15. Результат работы программы представлен на рисунке 14.

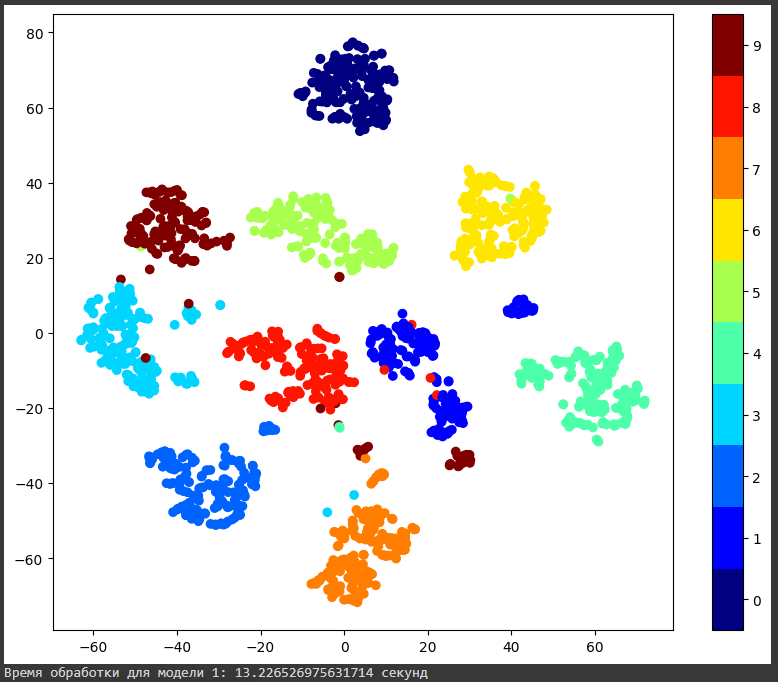


Рисунок 14 – Результат работы программы

**Задание 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.

Напишем код, который строит график набора данных MNIST, содержащий изображения рукописных цифр от 0 до 9, c помощью библиотеки umap. Код представлен на рисунке 15.

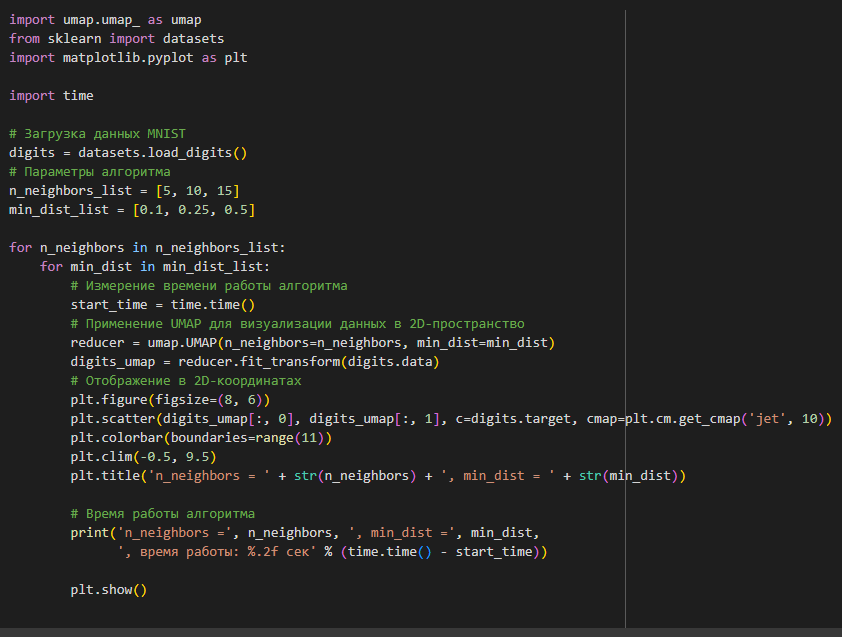


Рисунок 15 – Код

Результат работы программы представлен на рисунках 16 - 24.

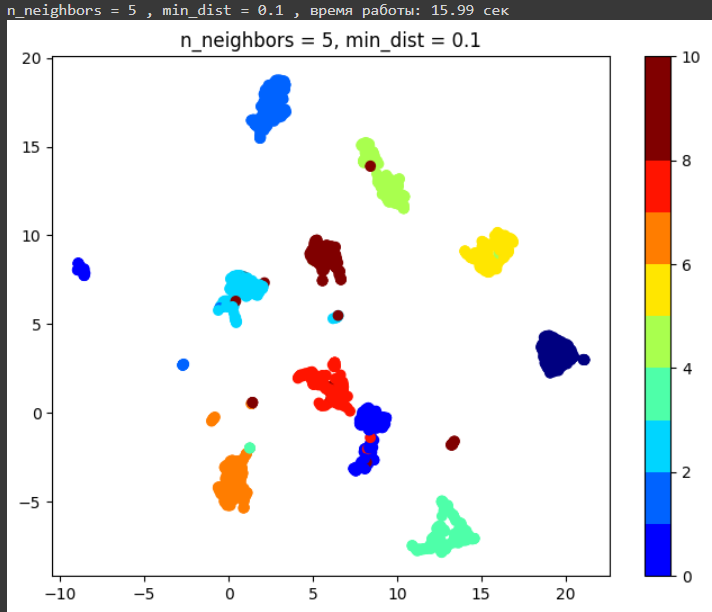


Рисунок 16 – Результат работы программы

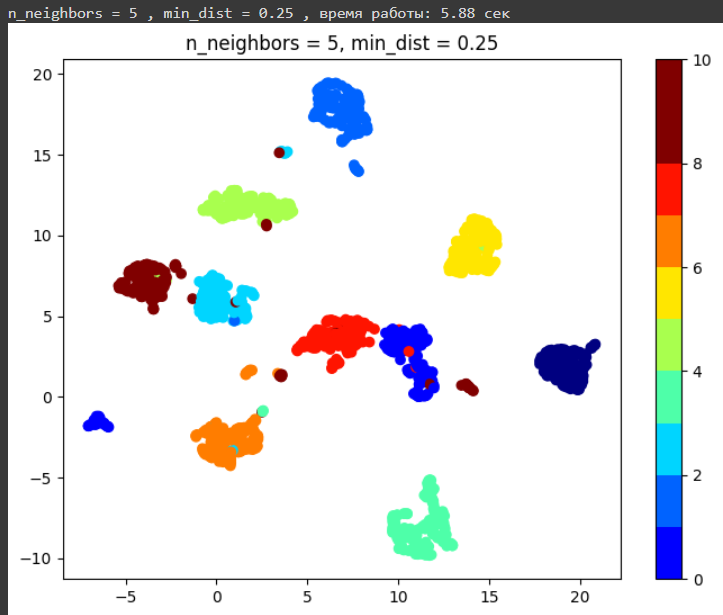


Рисунок 17 – Результат работы программы

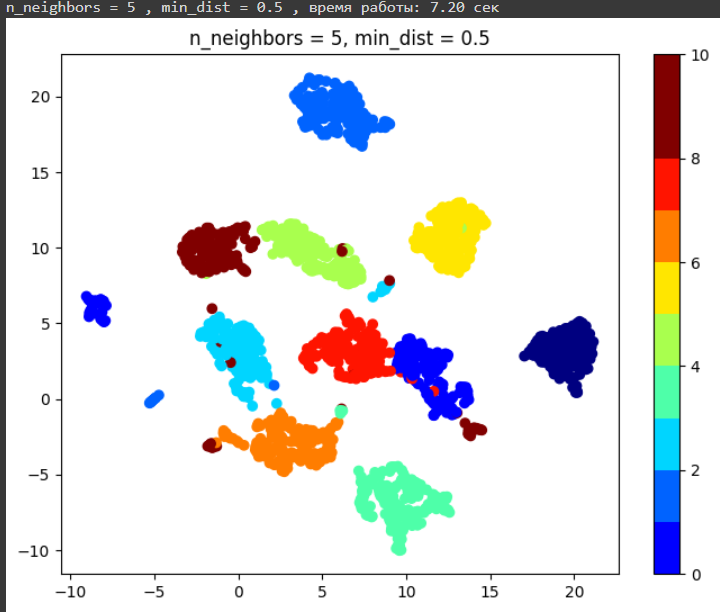


Рисунок 18 – Результат работы программы

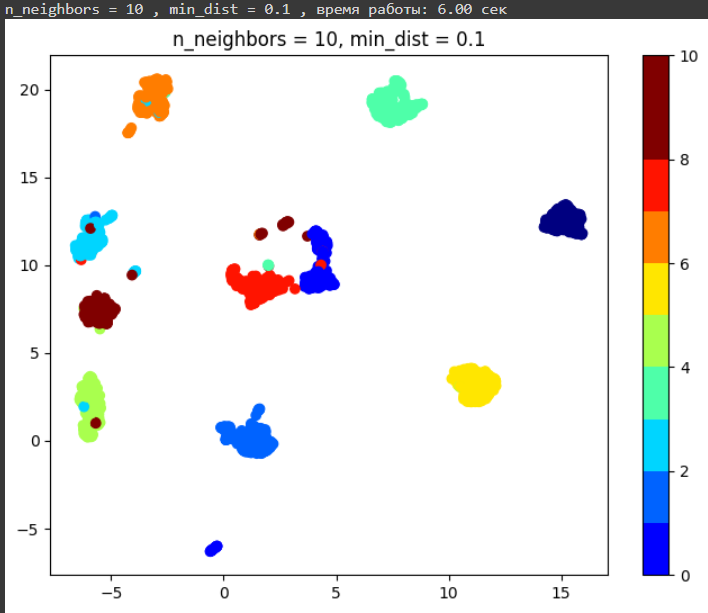


Рисунок 19 – Результат работы программы

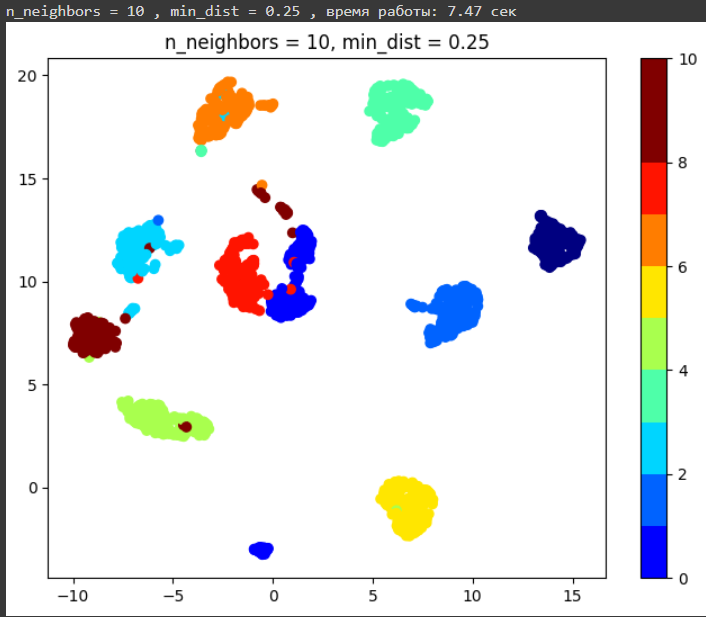


Рисунок 20 – Результат работы программы

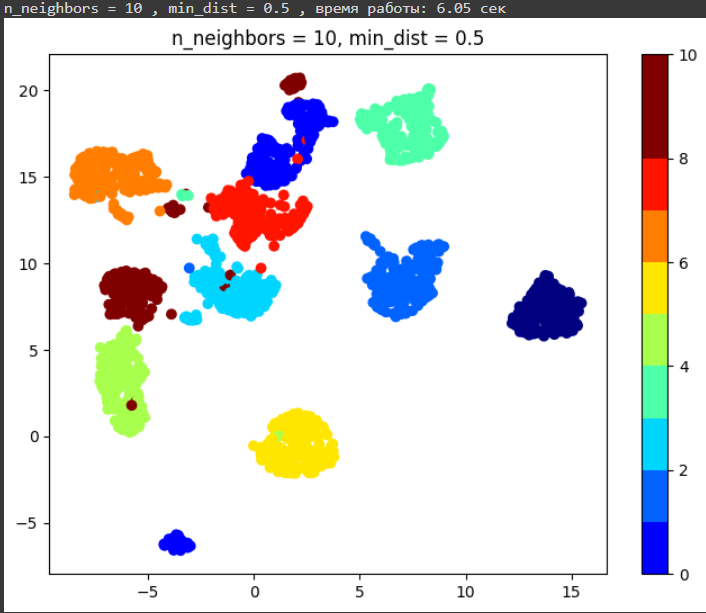


Рисунок 21 – Результат работы программы

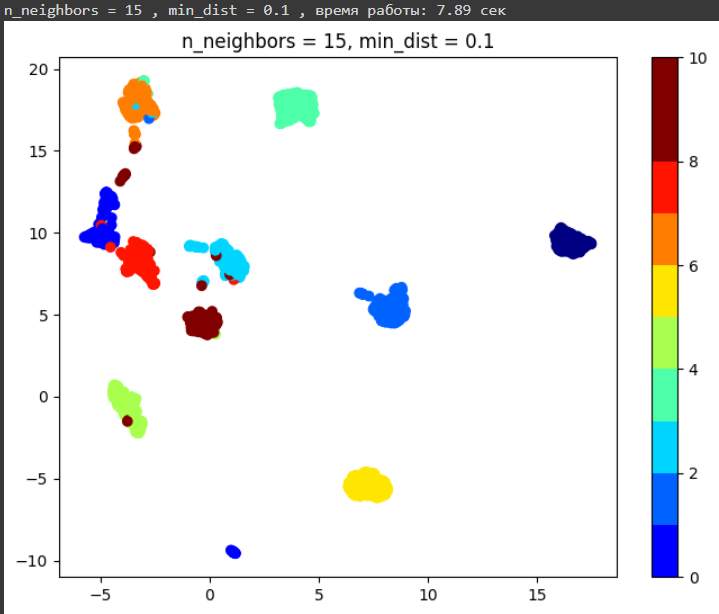


Рисунок 22 – Результат работы программы

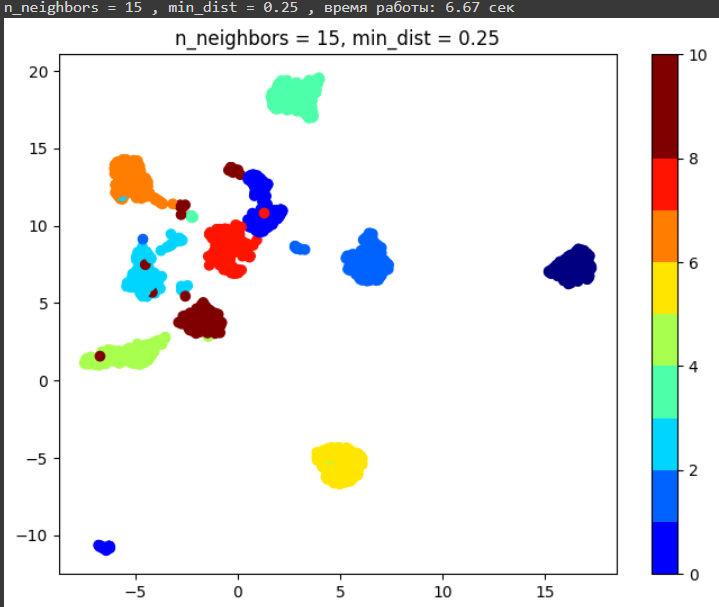


Рисунок 23 – Результат работы программы

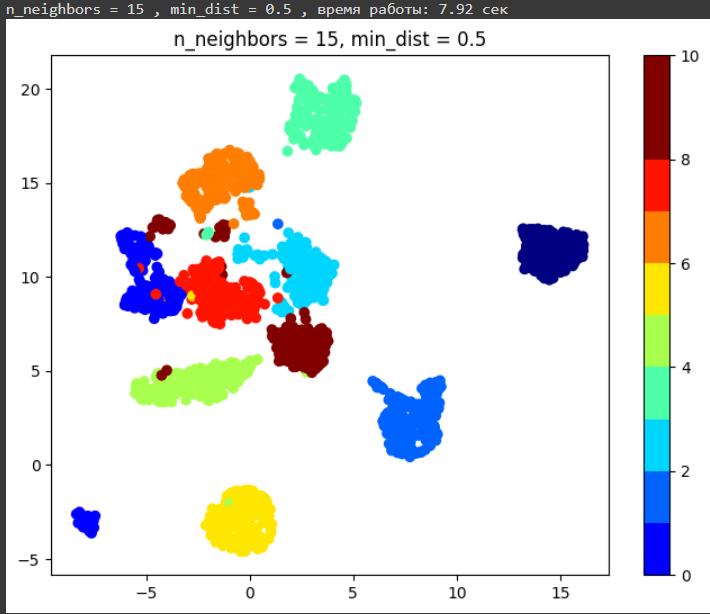


Рисунок 24 – Результат работы программы

Можно заметить, что время отрисовки каждого графика umap меньше, чем время отрисовки графика t-SNE.