|  |
| --- |
| Изображение выглядит как эмблема, символ, герб, нашивка  Автоматически созданное описание  МИНОБРНАУКИ РОССИИ |
| 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**

по дисциплине «Технологии и инструментарий анализа больших данных»

**Практическая работа № 2**

|  |  |  |  |
| --- | --- | --- | --- |
| Студент группы ИНБО-01-17 | *ИКБО-01-22 Прокопчук Роман Олегович* | (подпись) | |
| Ассистент | *Тетерин Николай Николаевич* | (подпись) | |
| Отчёт представлен | «\_\_» сентября 2025 г. | |  | |

Москва, 2025 г.

**ЦЕЛЬ**

Ознакомиться с различными библиотеками визуализации данных (matplotlib, plotly, TSNE, UMAP) и особенностями работы с ними в среде программирования Python.

**ХОД РАБОТЫ**

**Задание**

1. Найти и выгрузить многомерные данные (с большим количеством признаков – столбцов) с использованием библиотеки pandas. В отчёте описать найденные данные.
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.
3. Построить столбчатую диаграмму (.bar) с использованием модуля graph\_objs из библиотеки Plotly со следующими параметрами:
   1. По оси Х указать дату или название, по оси У указать количественный показатель.
   2. Сделать так, чтобы столбец принимал цвет в зависимости от значения показателя (marker=dict(color=признак, coloraxis="coloraxis")).
   3. Сделать так, чтобы границы каждого столбца были выделены чёрной линией с толщиной равной 2.
   4. Отобразить заголовок диаграммы, разместив его по центру сверху, с 20 размером текста.
   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.
   6. Размер текста меток осей сделать равным 14.
   7. Расположить график во всю ширину рабочей области и присвоить высоту, равную 700 пикселей.
   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')
   9. Убрать лишние отступы по краям.
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)
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.
   1. Сделать график с линиями и маркерами, цвет линии 'crimson', цвет точек 'white', цвет границ точек 'black', толщина границ точек равна 2.
   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').
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.
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.
8. На основе проделанной работы составить отчёт с описанием и скриншотами полученных результатов, сделать выводы о выбранных данных на основе полученных графиков, сравнить библиотеки. Начиная с 6 пункта отчёт дополнительно должен содержать результаты визуализации для разных значений параметров и выводы.

В качестве данных для практической был выбран датасет с данным о пит стопах Формулы-1 с 1950 по 2024 года. Ссылка - <https://www.kaggle.com/datasets/akashrane2609/formula-1-pit-stop-dataset?resource=download>.

На рисунке 1 представлен скрипт для выгрузки и предобработки данных.

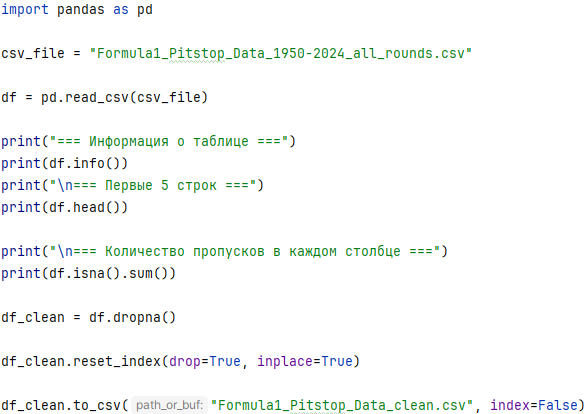


Рисунок 1 – Загрузка и преобразование данных

На рисунке 2 представлен код для создания столбчатой диаграммы.

Созданная диаграмма представлена на рисунке 3.



Рисунок 2 – Код для создания столбчатой диаграммы

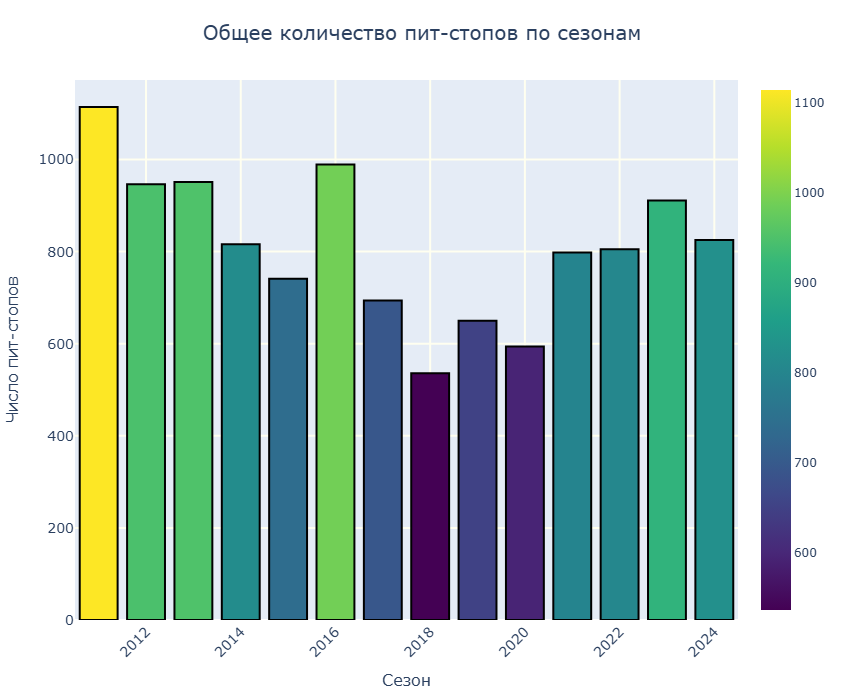


Рисунок 3 – Созданная столбчатая диаграмма

Код для создания круговой диаграммы представлен на рисунке 4. Созданная диаграмма представлена на рисунке 5.



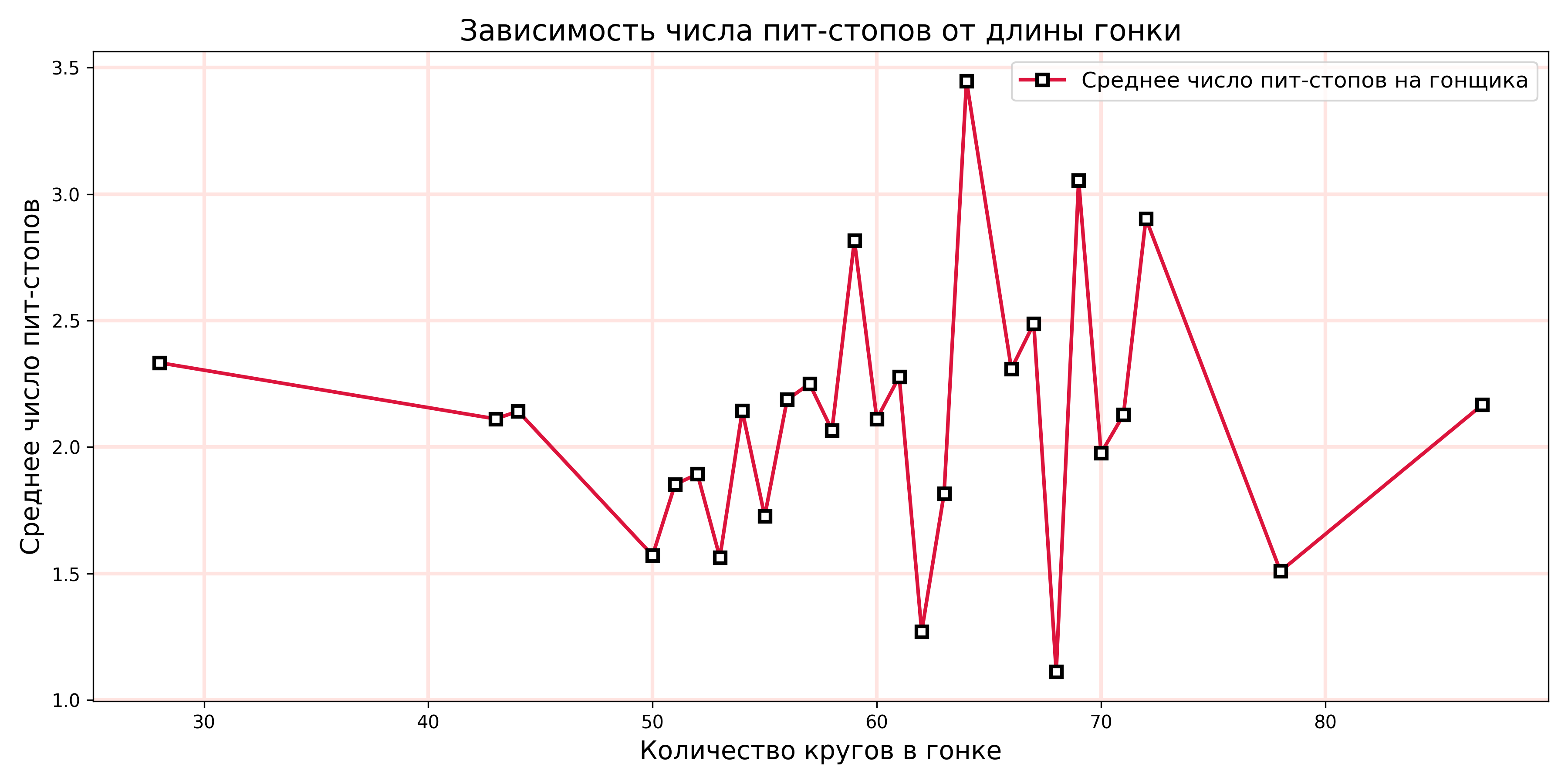
Рисунок 4 – Код для создания круговой диаграммы

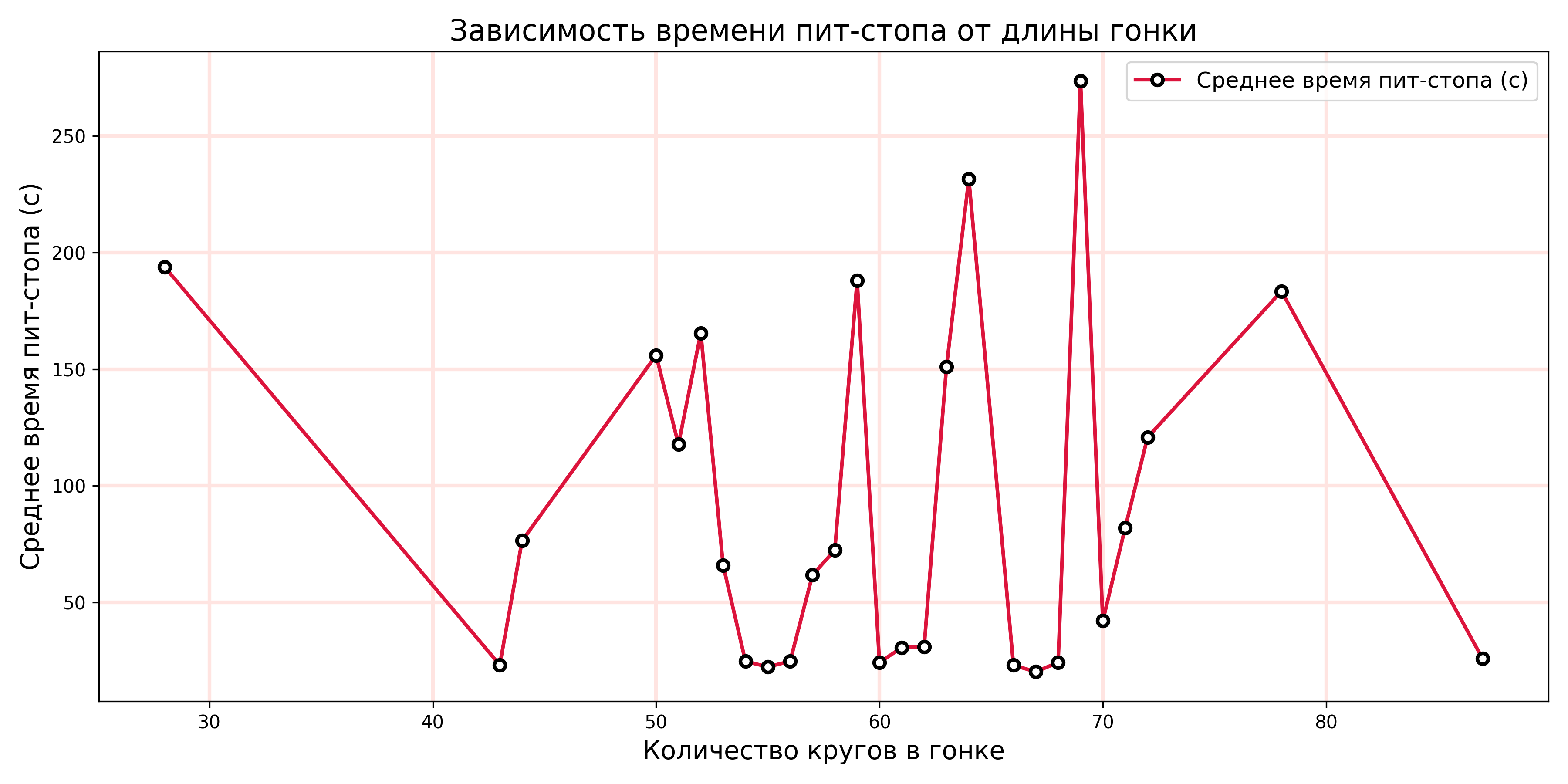


Рисунок 5 – Созданная круговая диаграмма

Код для создания линейных графиков представлен на рисунке 6. Созданные графики – на рисунка 7 и 8.

  
Рисунок 6 – Код для создания линейных графиков

  
Рисунок 7 – Зависимость числа пит-стопов от кол-ва кругов в гонке

  
Рисунок 8 – Зависимость времени пит-стопов от кол-ва кругов в гонке

Из полученных графиков видно, что кол-во пит-стопов в гран-при формулы 1, а также их длительность никак не соотносятся с кол-вом кругов в гонке. Эти параметры сильно зависят от внешних факторов. В первую очередь это касается температуры трассы, возможных осадков во время гонок, кол-ва аварий.

На рисунках 9 и 10 представлен код для визуализации многомерных данных при помощи t-SNE и UMAP соответственно.

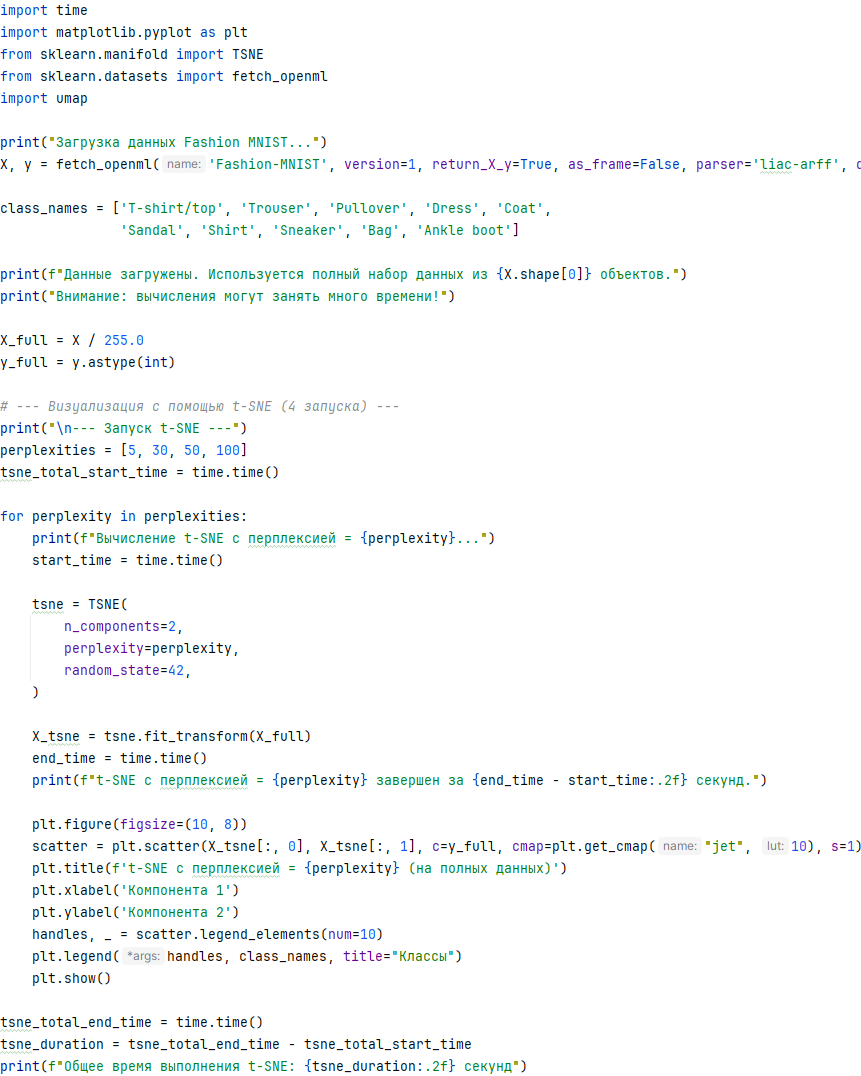


Рисунок 9 – Загрузка данных и их визуализация при помощи t-SNE

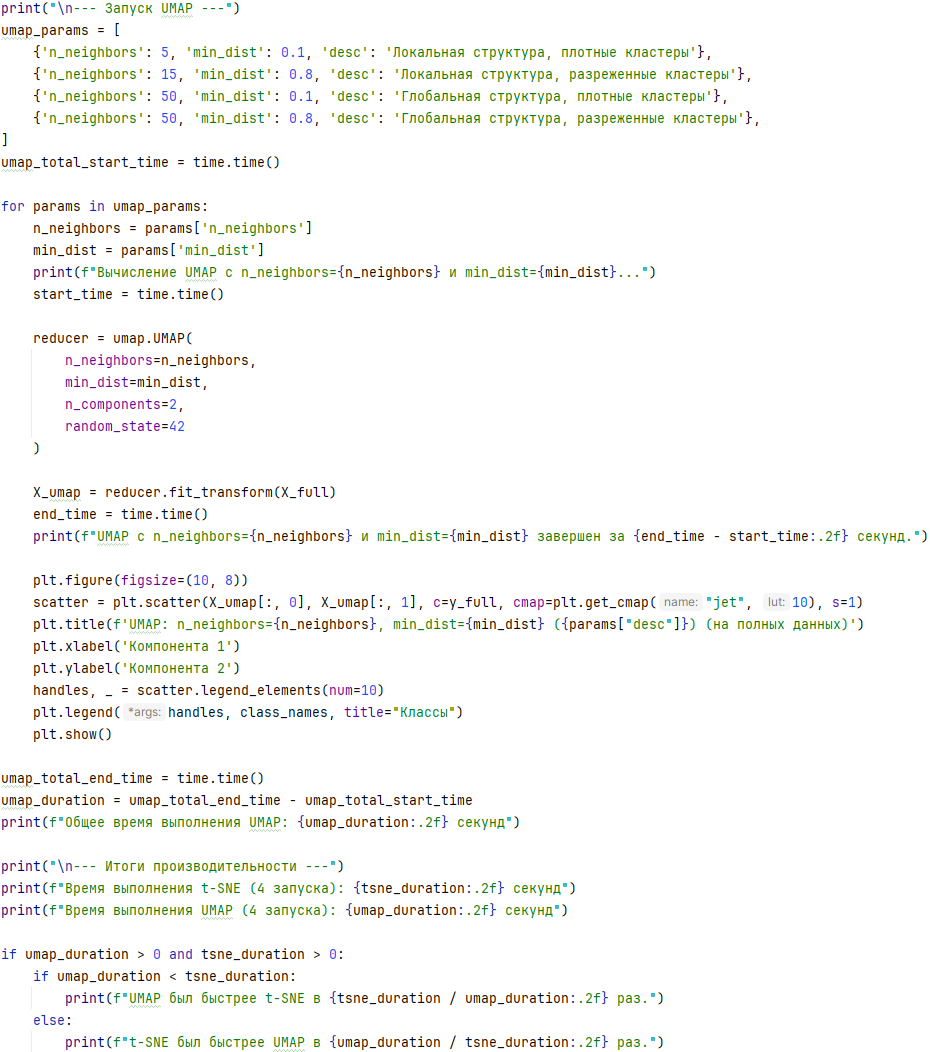


Рисунок 10 – Визуализация данных при помощи UMAP и сравнение по времени выполнения

Визуализации, созданные при помощи t-SNE представлены на рисунках 11-14, при помощи UMAP – на рисунках 15-18.

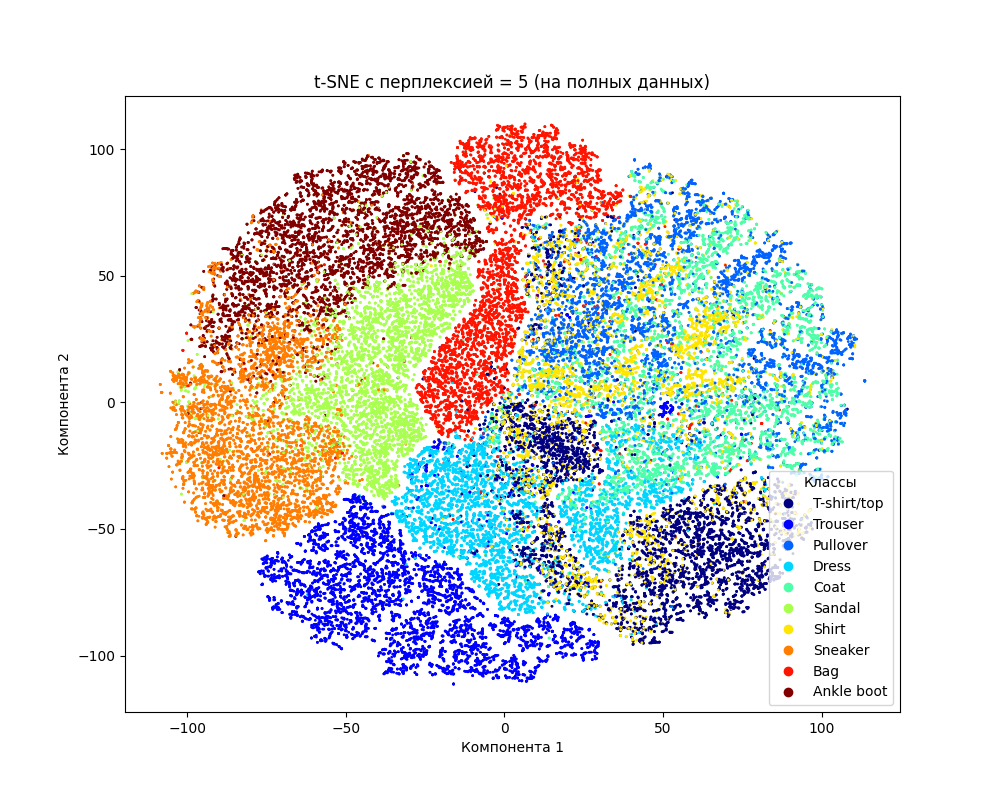


Рисунок 11 – Визуализация данных при помощи t-SNE (1/4)

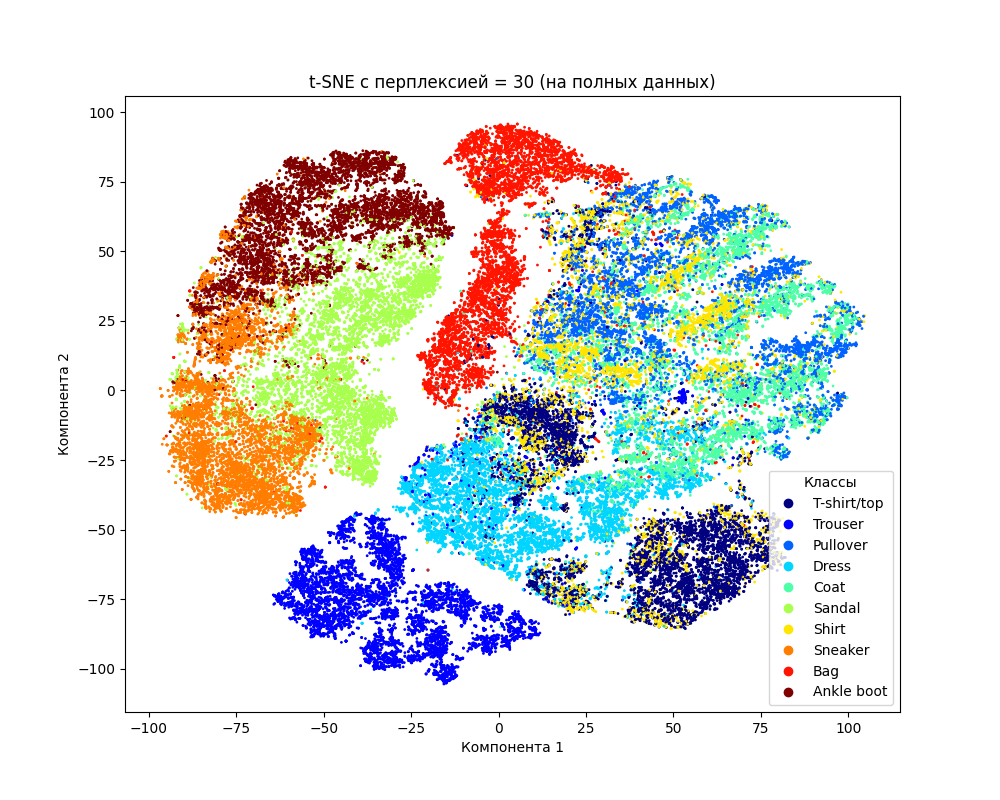


Рисунок 12 – Визуализация данных при помощи t-SNE (2/4)

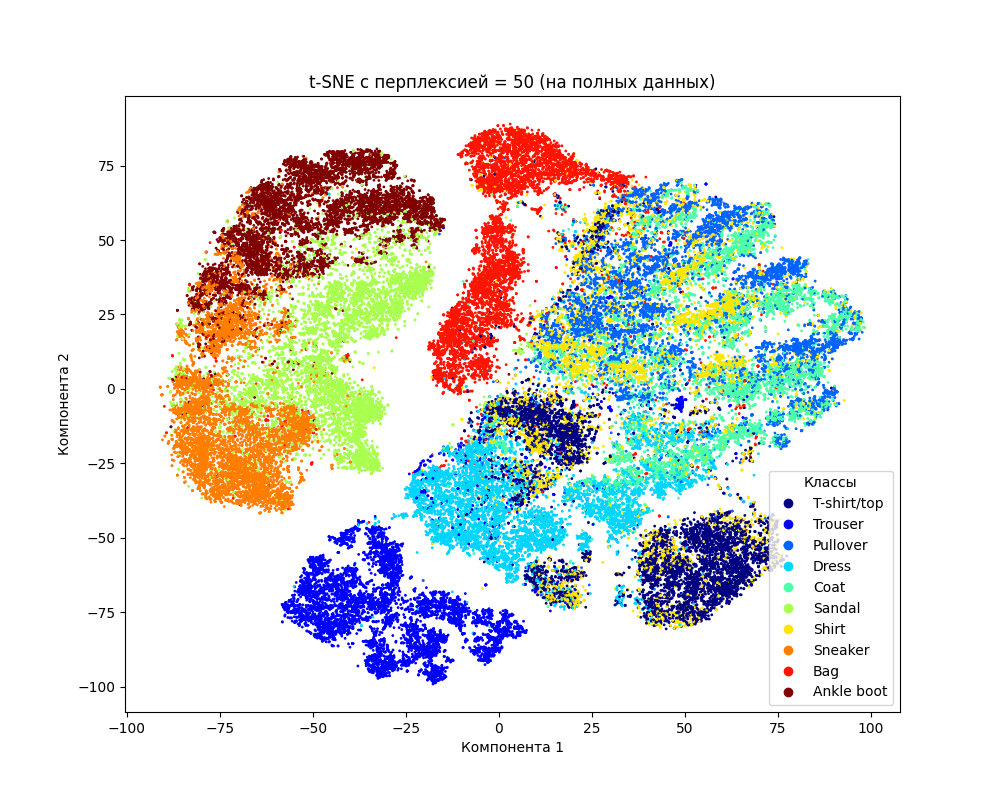


Рисунок 13 – Визуализация данных при помощи t-SNE (3/4)

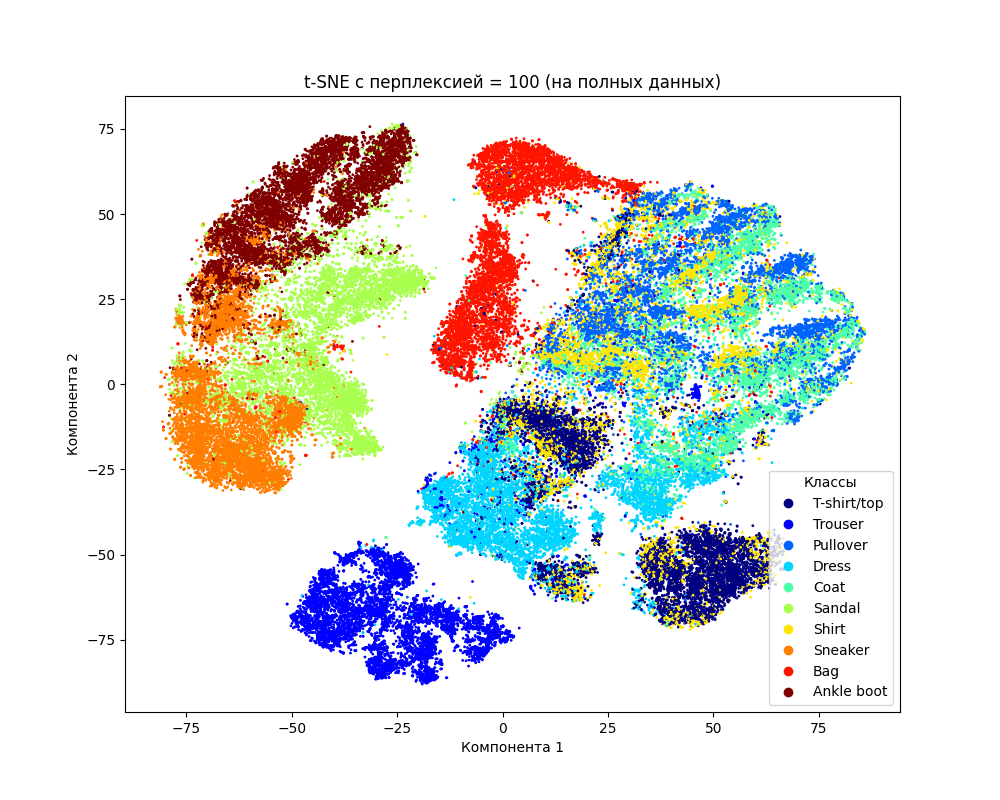


Рисунок 14 – Визуализация данных при помощи t-SNE (4/4)

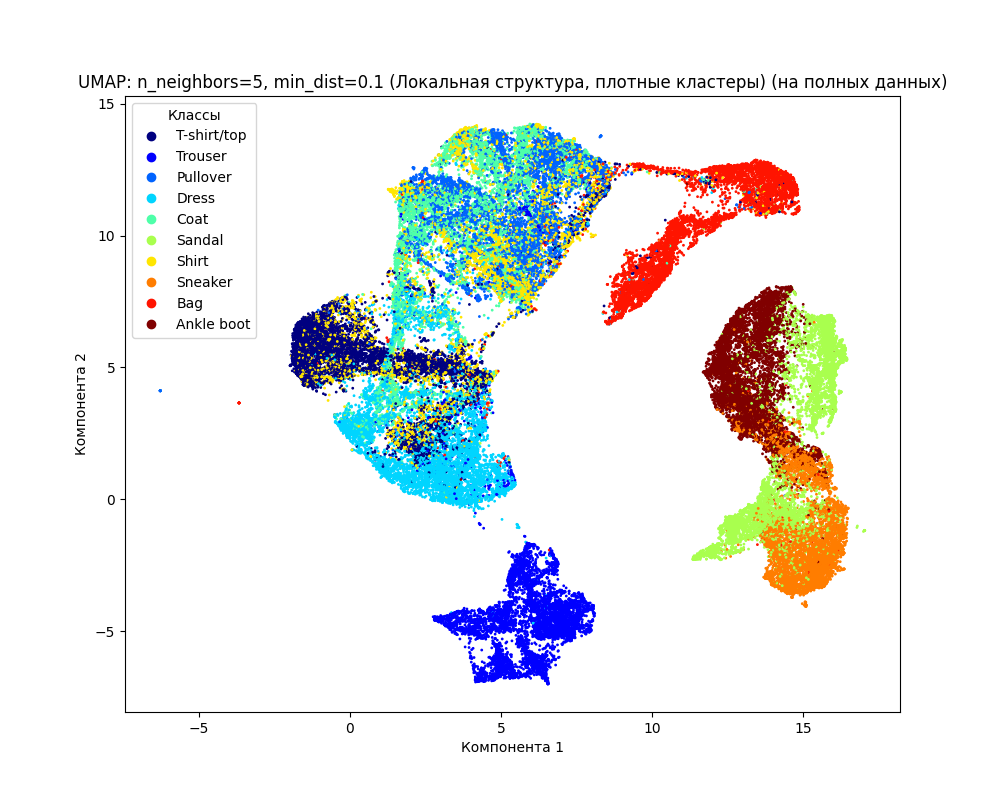


Рисунок 15 – Визуализация данных при помощи UMAP (1/4)

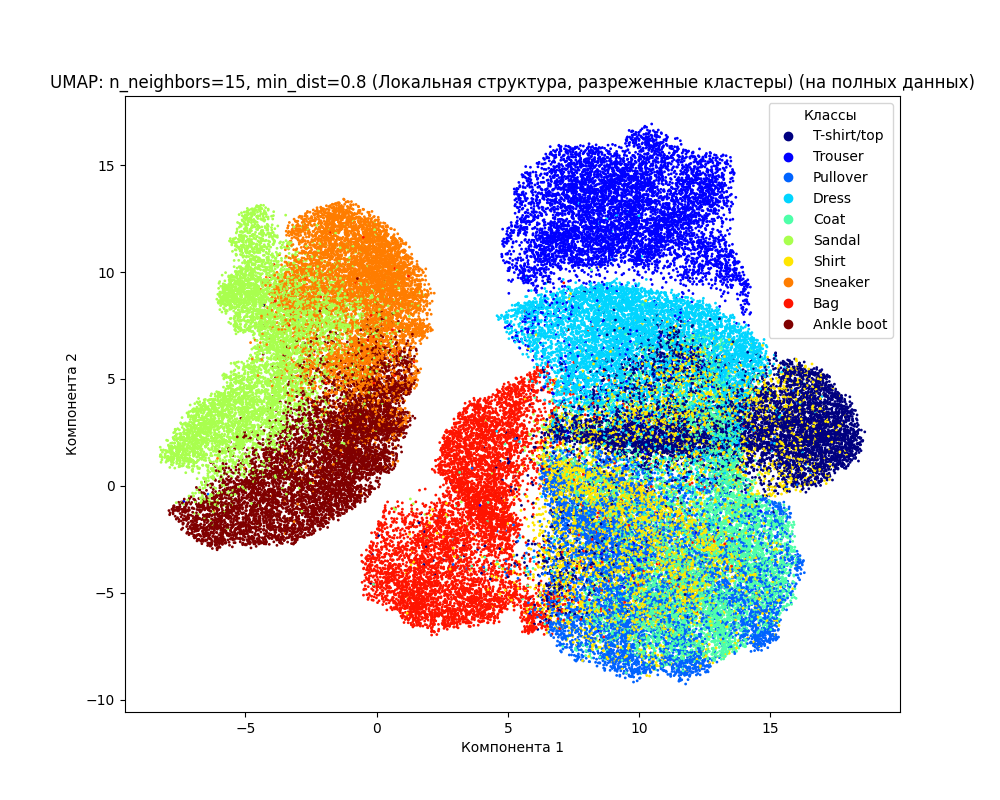


Рисунок 16 – Визуализация данных при помощи UMAP (2/4)

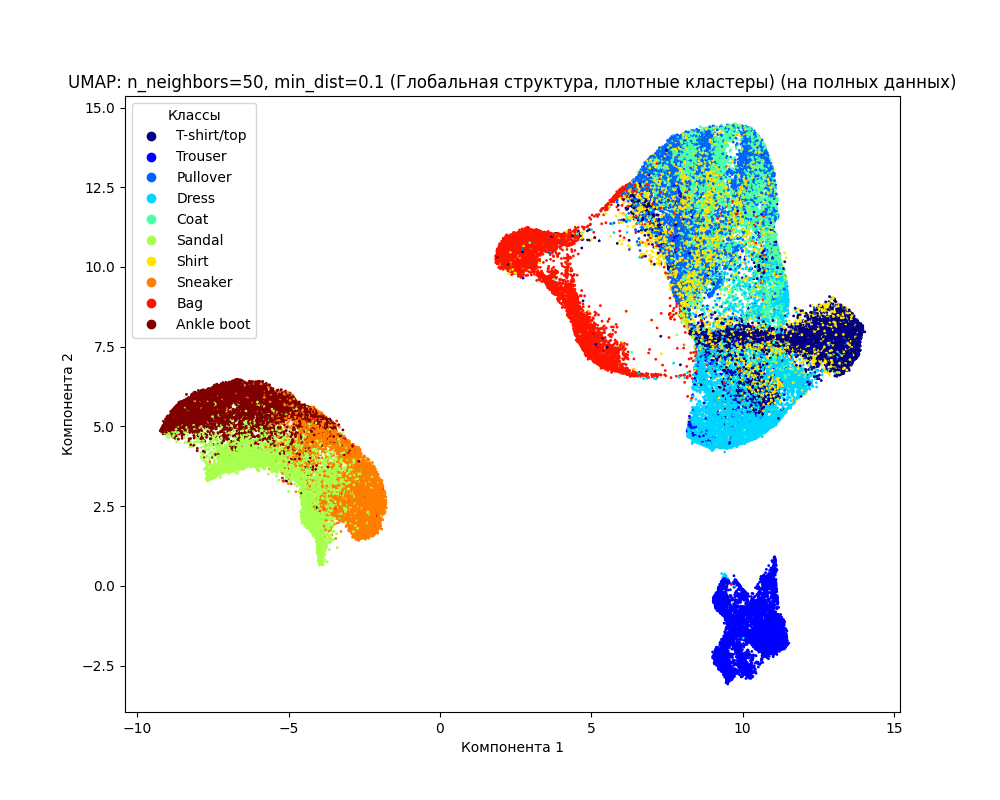


Рисунок 17 – Визуализация данных при помощи UMAP (3/4)

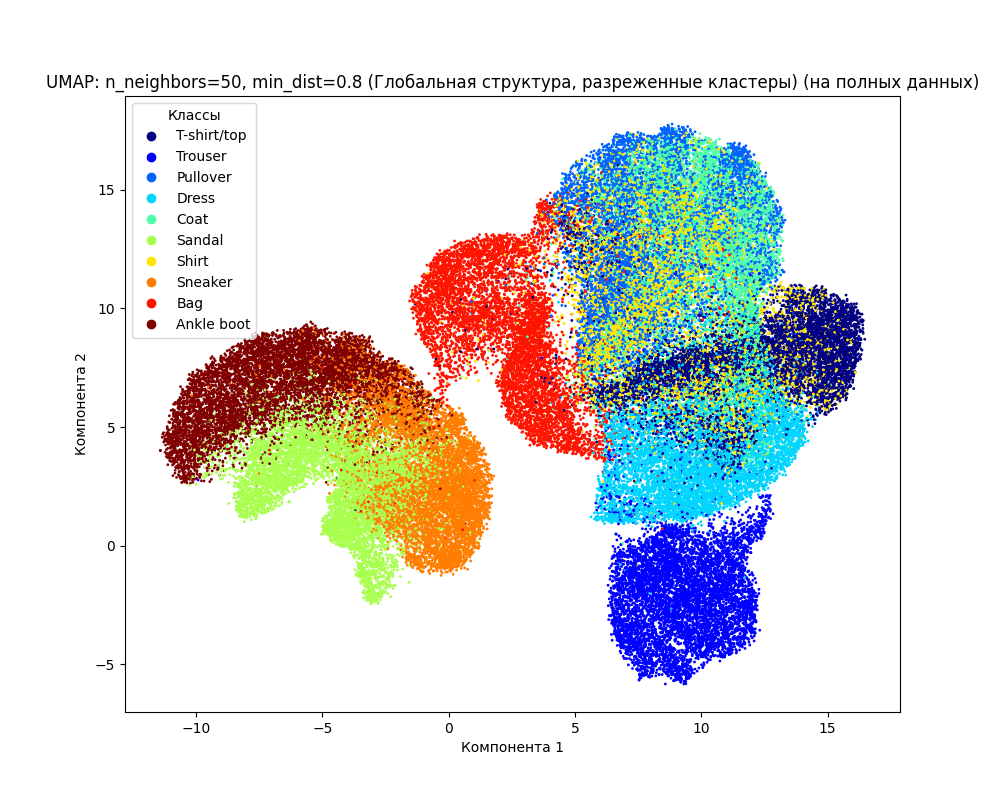


Рисунок 18 – Визуализация данных при помощи UMAP (4/4)

Сравнение двух способов по времени работы представлено на рисунке 19.

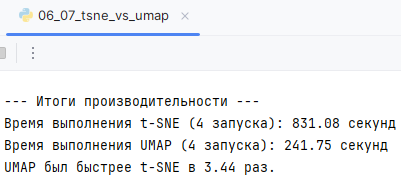


Рисунок 19 – Сравнение времени работы t-SNE и UMAP

**ВЫВОД**

В ходе выполненной практической работы проведено ознакомление с различными библиотеками визуализации данных (matplotlib, plotly, TSNE, UMAP) и особенностями работы с ними в среде программирования Python.