|  |  |
| --- | --- |
| **Домашнее задание** | |
|  | |
| по дисциплине | Практикум по программированию |

|  |  |  |  |
| --- | --- | --- | --- |
| Студента | Загребельного Владислава Александровича | | |
|  | фамилия, имя, отчество полностью | | |
| Курс | 2 | Группа | ФИТ-221 |
| Направление | 02.03.02. Фундаментальная информатика и информационные технологии | | |
|  | код, наименование | | |
| Руководитель | ст.преподаватель | | |
|  | должность, ученая степень, звание | | |
| Саматов А. П. | | |
|  | фамилия, инициалы, дата, подпись | | |
| Выполнил |  | | |
|  | дата, подпись студента(ки) | | |

|  |  |
| --- | --- |
| Итоговый рейтинг |  |

Омск 2023

ВВЕДЕНИЕ 3

1.Поиск и загрузка данных 4

2.Разведывательный анализ данных 5

3.Предварительная обработка данных 9

ЗАКЛЮЧЕНИЕ 11

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 12

# **ВВЕДЕНИЕ**

Анализ данных стал неотъемлемой частью современного мира, где информация играет ключевую роль в принятии решений. Он позволяет извлекать ценные знания из больших объемов данных и выявлять скрытые закономерности, что делает его не только крутым, но и востребованным в различных областях.

Для успешного анализа данных существует множество инструментов и библиотек, которые упрощают и автоматизируют процесс обработки и визуализации данных. Одним из таких инструментов является *Matplotlib*, который предоставляет мощные возможности для создания графиков и визуализации данных в различных форматах. Библиотека *numpy* предоставляет удобные функции для работы с массивами и матрицами, что позволяет удобно и эффективно оперировать числовыми данными. *Pandas*, в свою очередь, предоставляет высокоуровневые структуры данных и инструменты для работы с ними, что позволяет легко обрабатывать и анализировать табличные данные. *Sympy* и *scipy* предоставляют возможности для символьных и численных вычислений соответственно, что помогает в решении сложных математических задач. И наконец, *seaborn* предоставляет удобные функции для визуализации данных и создания стильных графиков.

Таким образом, использование данных инструментов значительно упрощает процесс анализа данных, позволяя исследователям и специалистам в различных областях получить более точные и надежные результаты.

# **1.Поиск и загрузка данных**

Использован набор данных *spotify songs*[[5]](https://www.kaggle.com/code/carlmarco/spotify-songs/data).

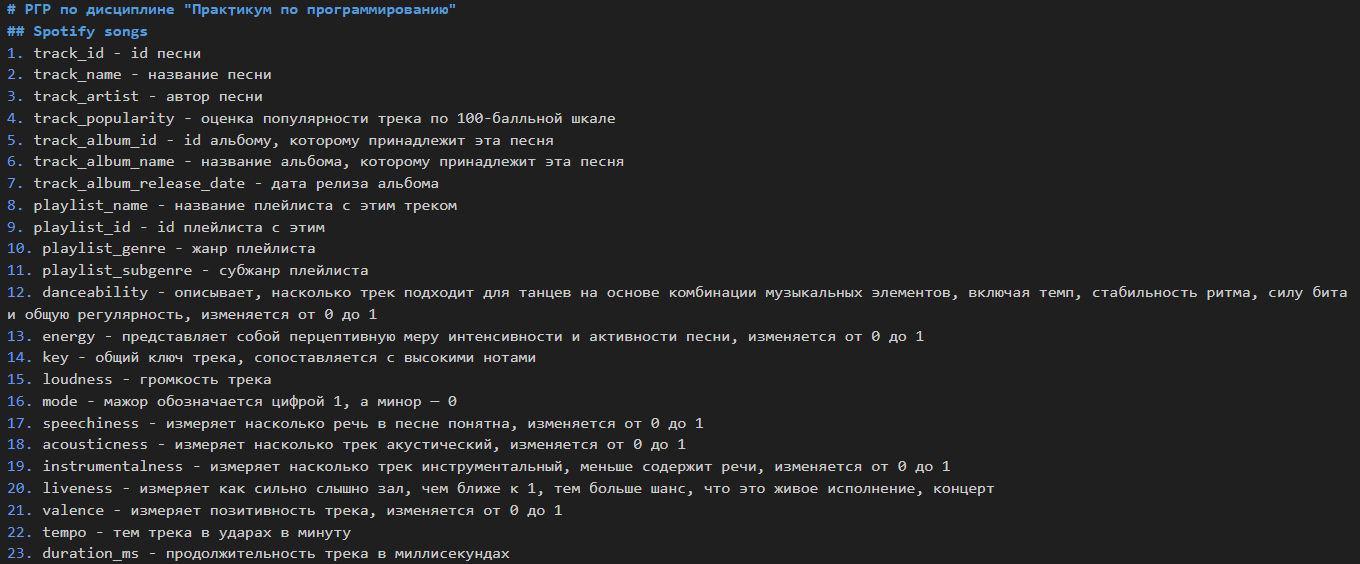


Рисунок 1 - Файл Readme.md

Набор данных был загружен в ноутбук командой *read\_csv()*, импортированной из библиотеки *Pandas.*

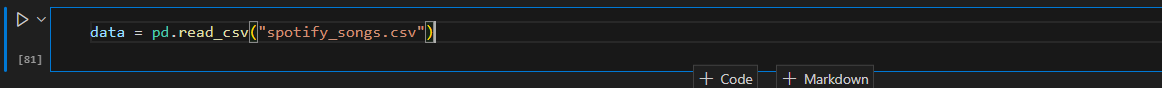


Рисунок 2 - Загрузка набора данных

В данном наборе данных содержится 32 833 строки и 23 столбца, в которых указаны все возможные характеристики для каждой песни.

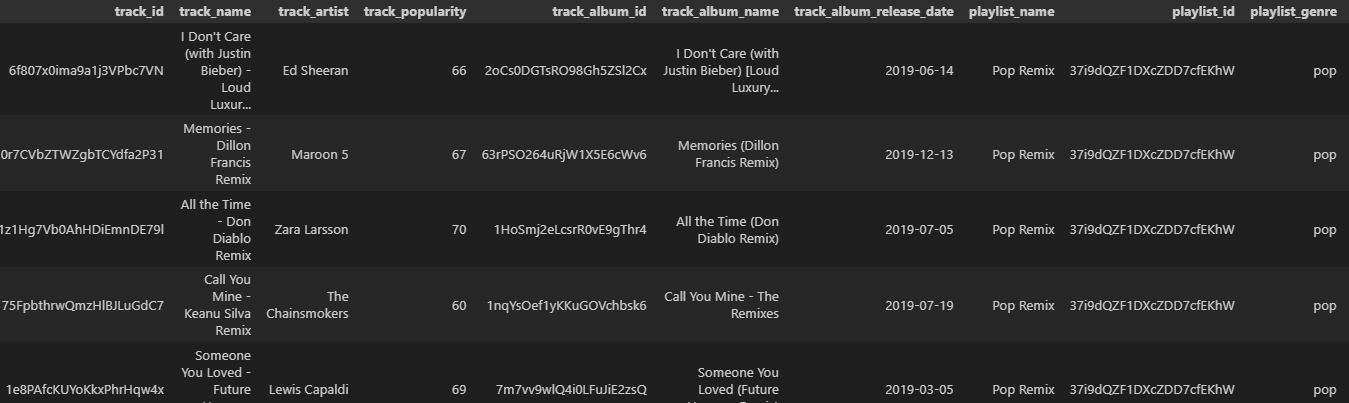


Рисунок 3 - Часть набора данных

# **2.Разведывательный анализ данных**

Гистограмма — это графическое представление распределения данных, которое позволяет наглядно представить, как часто встречаются определенные значения или диапазоны значений в наборе данных. Она представляет собой столбчатую диаграмму, где по горизонтальной оси отображаются возможные значения переменной, а по вертикальной оси отображается количество наблюдений, попадающих в каждый столбец.

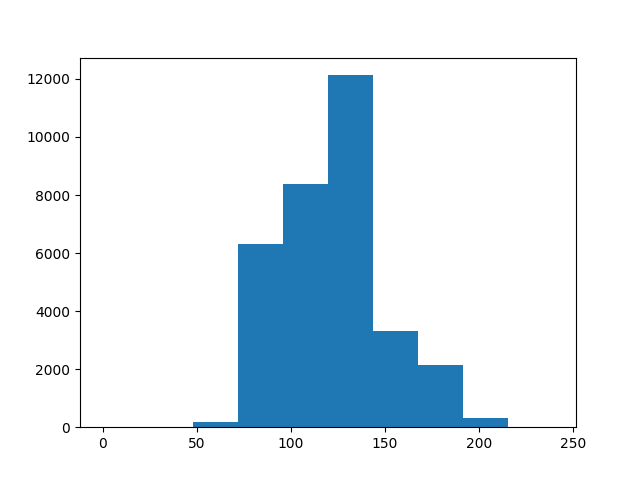


Рисунок 4 - Гистограмма для признака tempo

Судя по гистограмме, большинство песен в наборе данных имеет темп ~130 ударов в минуту.

Диаграмма "ящик с усами" (или "ящик с усами") - это графическое представление распределения данных, которое позволяет визуально оценить основные характеристики набора данных, такие как медиана, квартили, выбросы и размах.

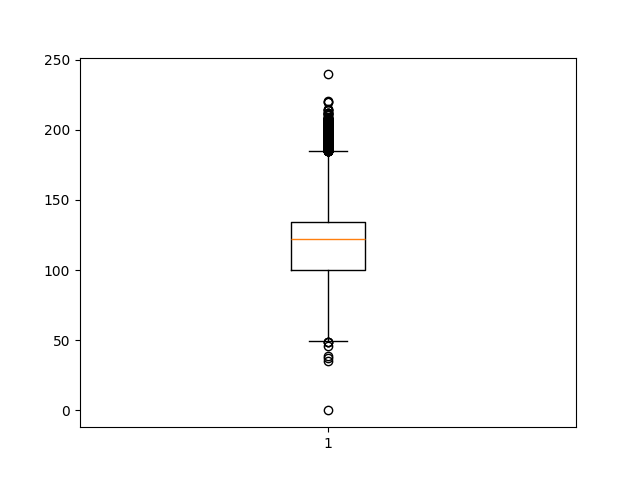


Рисунок 5 - Диаграмма "Ящик с усами" для признака tempo

Судя по диаграмме «Ящик с усами», в столбце tempo присутствуют выбросы, а медиана для tempo ~125.

Круговая диаграмма (или "пироговая диаграмма") — это графическое представление данных, которое использует круг для визуализации составляющих частей целого. Круговая диаграмма состоит из секторов, пропорциональных относительным значениям, которые они представляют.

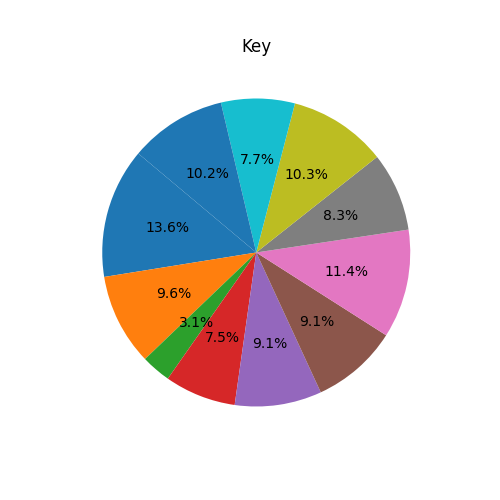


Рисунок 6 - Круговая диаграмма для признака key

Судя по круговой диаграмме, ключи в наборе данных распределены равномерно.

Тепловая карта (heatmap) или карта корреляции - это графическое представление данных, в котором значения каждой ячейки представлены цветом в соответствии с их числовым значением. Такие карты часто используются для визуализации матрицы корреляции, которая показывает степень линейной зависимости между парами переменных.

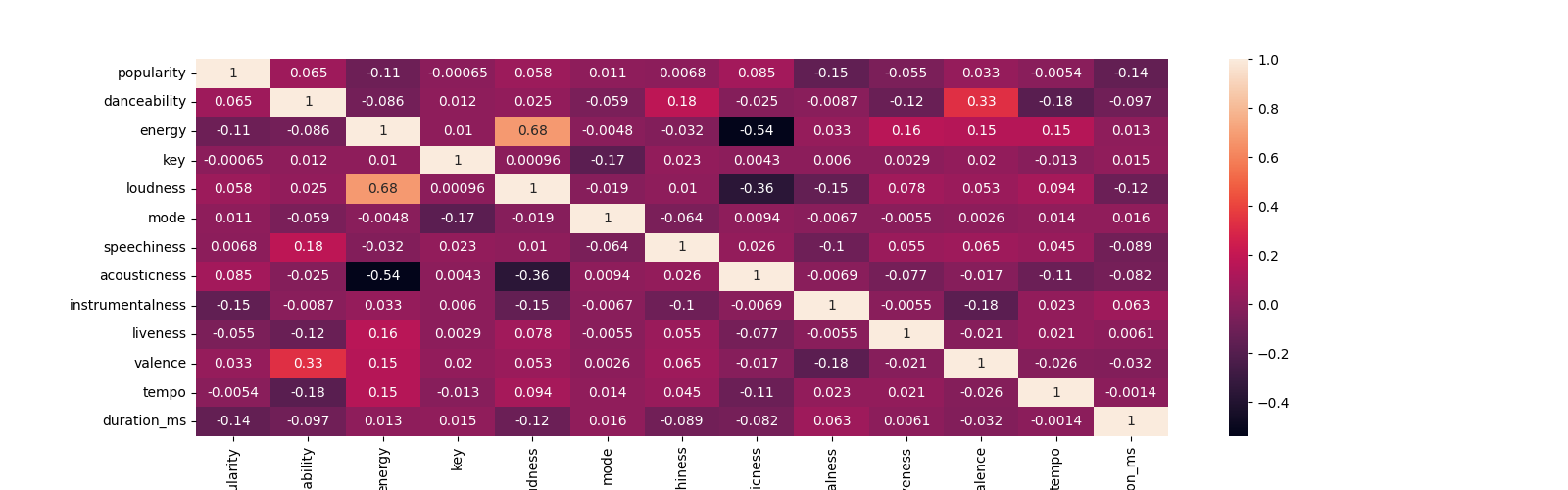


Рисунок 7 - Тепловая карта

Судя по тепловой карте, признаки loudness и energy хорошо коррелируют.

Диаграмма countplot — это графическое представление данных, которое показывает количество наблюдений в каждой категории переменной. Обычно countplot используется для визуализации распределения категориальных переменных. Это может быть полезно для быстрого анализа частоты появления различных значений в категориальных данных.

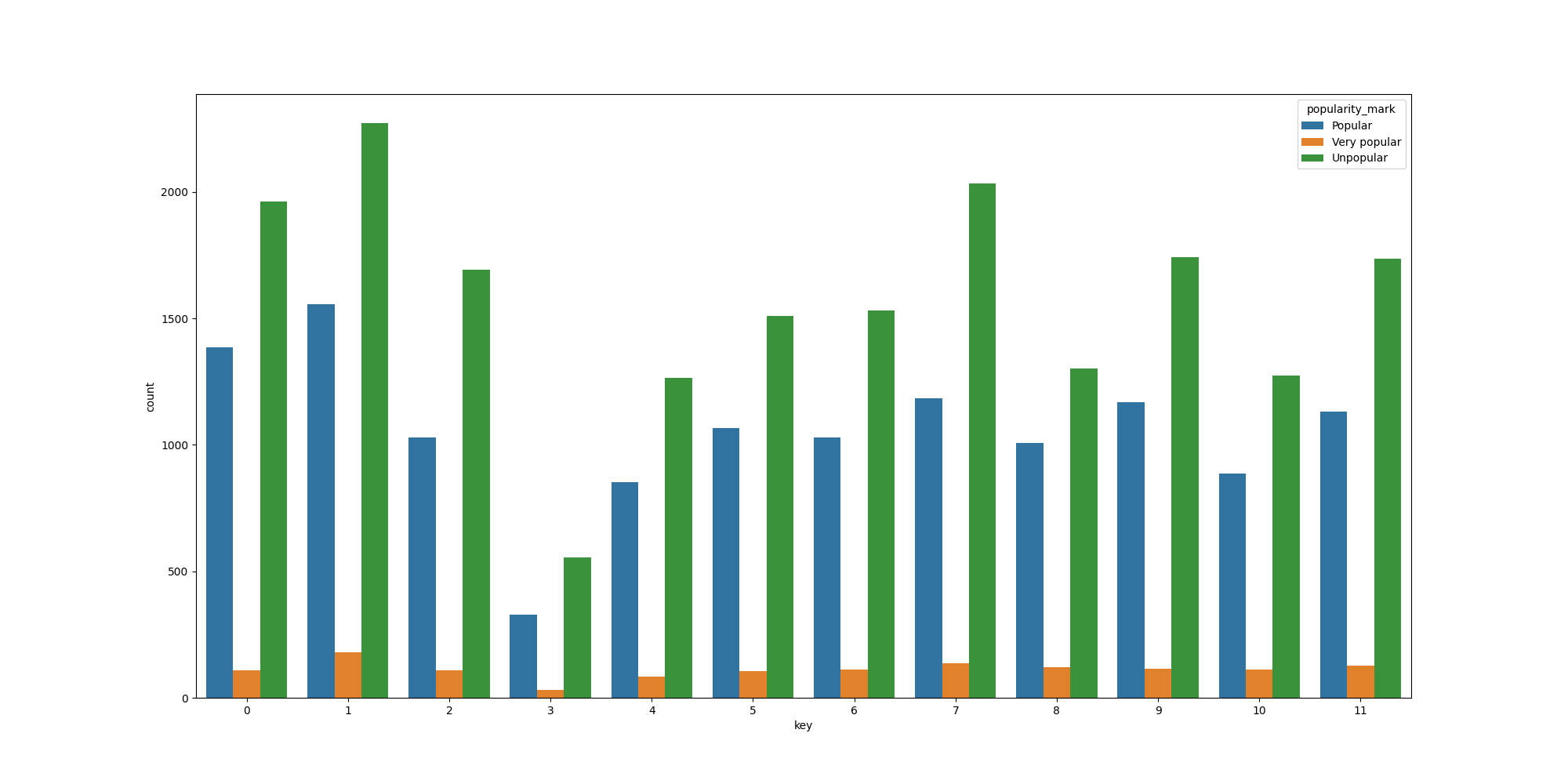


Рисунок 8 - График countplot для key и popularity

Судя по графику, среди наиболее популярных песен встречается первый ключ.

# **3.Предварительная обработка данных**

Для начала нужно проверить есть ли пустые значения в наборе данных.

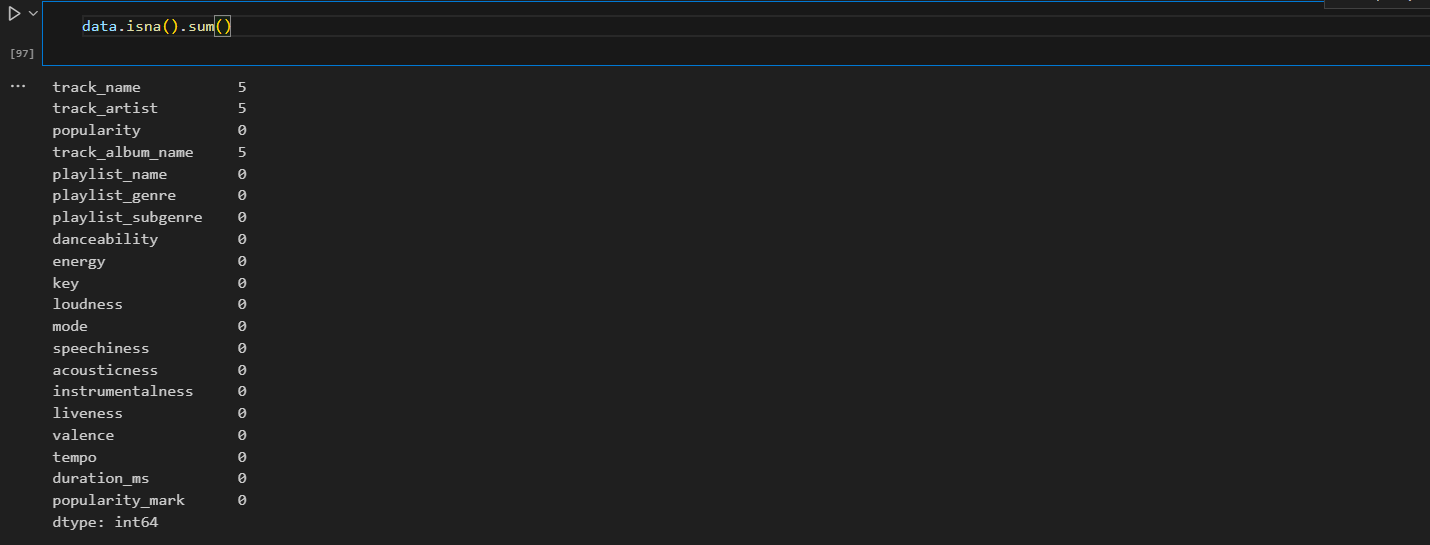


Рисунок 9 - Вывод количества пропусков в наборе данных

Пропущенный значения присутствуют, но заменять данные пропущенные значения модой нелогично. Поскольку набор данных имеет внушительные 32 833 строки, строки с пропущенными значения можно дропнуть.

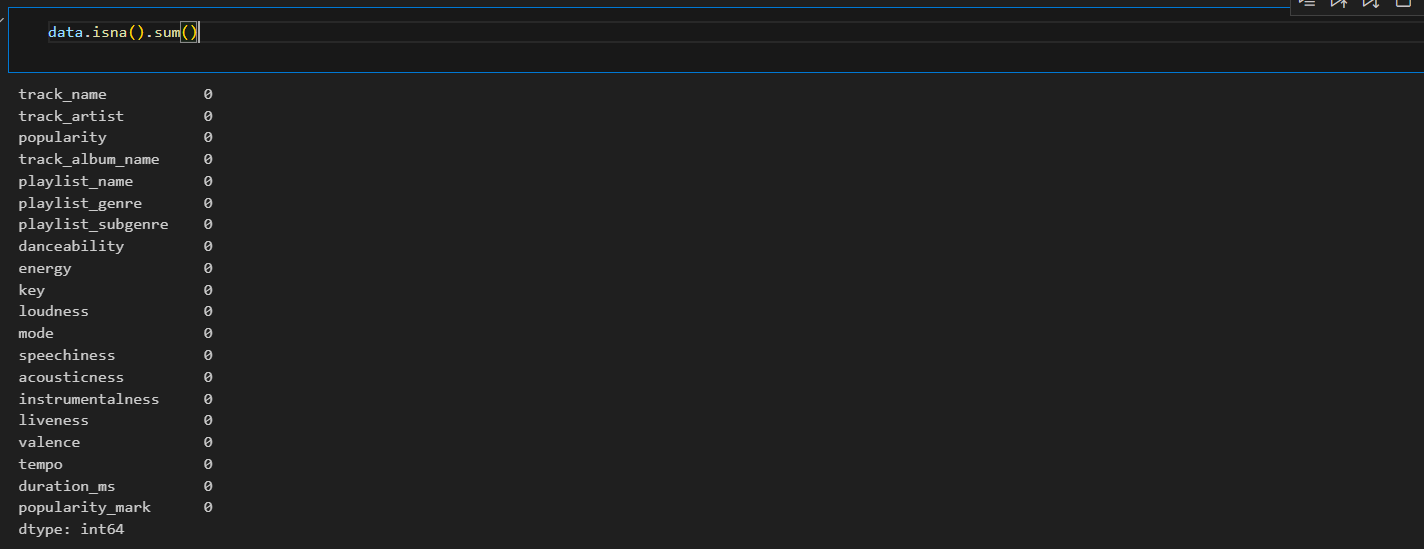


Рисунок 10 - Вывод количества пропущенных значений после дропа

После удаления пустых значений были удалены дубликаты.

После удаления дубликатов были удалены ненужные столбцы.



Рисунок 11 - Удаление ненужных столбцов

Также нужно было применить для некоторых категориальных признаков one-hot кодирование.

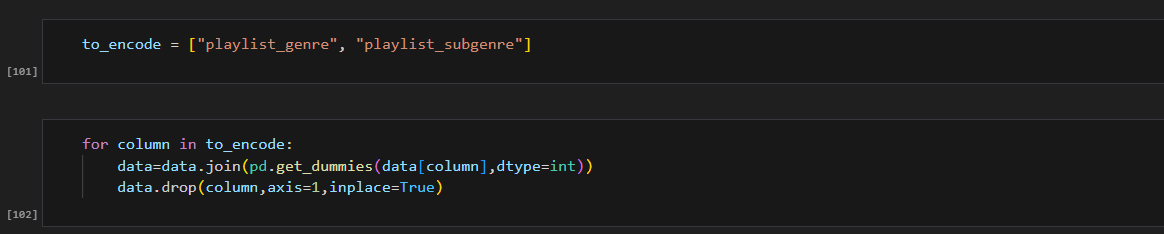


Рисунок 12 - Кодирование категориальных признаков

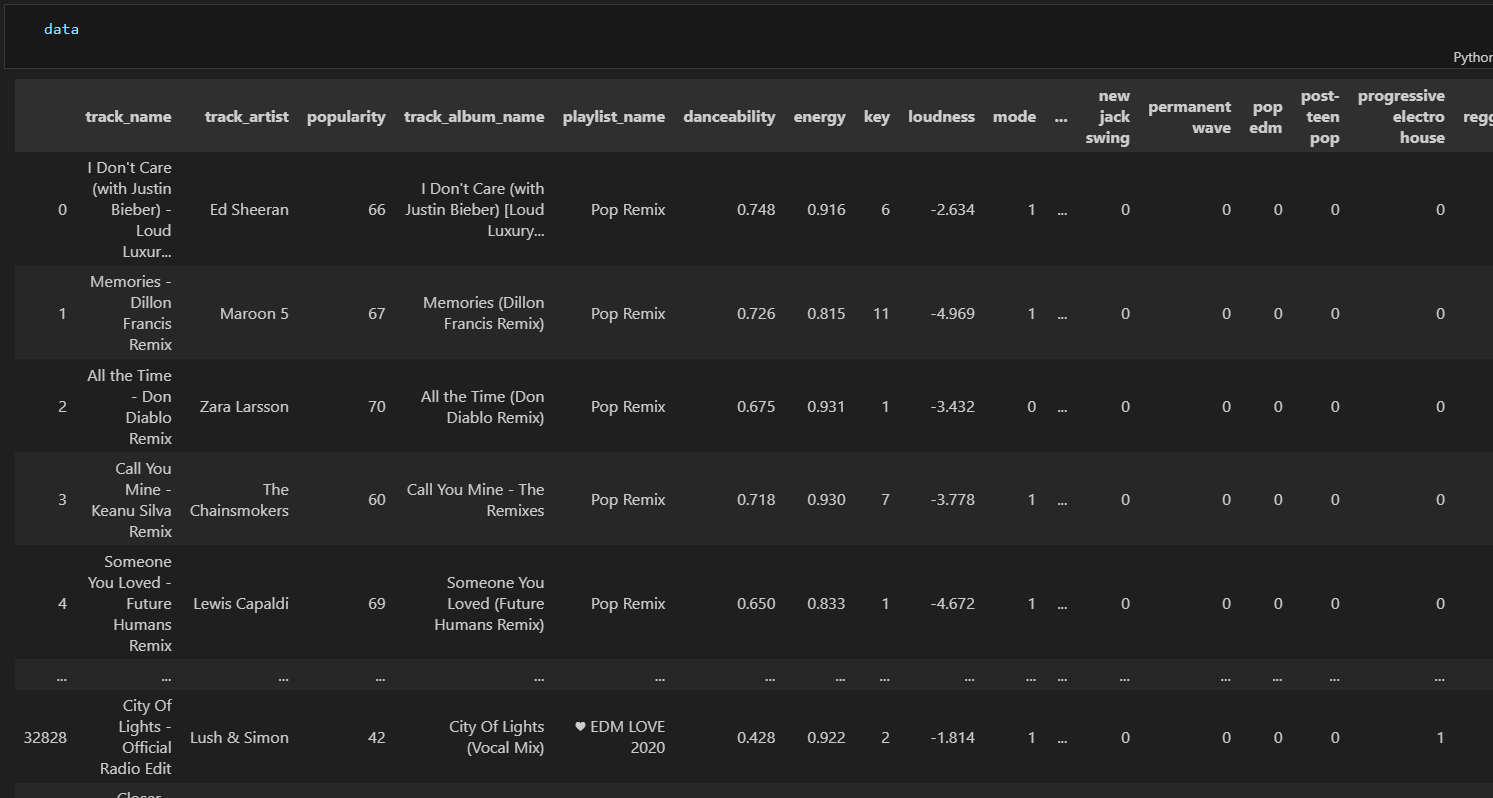


Рисунок 13 - Набор данных после выполненной предобработки

После выполненной предобработки набор данных был сохранен.

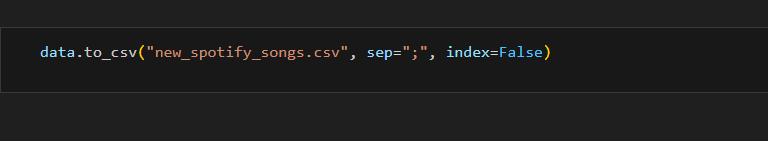


Рисунок 14 - Сохранение предобработанного набора данных

# **ЗАКЛЮЧЕНИЕ**

В ходе практики были изучены и применены ключевые библиотеки Python для анализа данных, включая Matplotlib, NumPy, Pandas, SymPy, SciPy и Seaborn. Это позволило углубить понимание основных инструментов анализа данных и визуализации, а также научиться применять их в реальных проектах.

В частности, были изучены возможности Matplotlib для создания различных видов графиков и диаграмм, NumPy для работы с массивами и матрицами, Pandas для обработки и анализа данных, SymPy для символьных вычислений, SciPy для выполнения научных и инженерных расчетов, а также Seaborn для создания статистических графиков.

Кроме того, в рамках практики была выполнена предобработка набора данных, включающая в себя очистку данных от выбросов и пропущенных значений, преобразование категориальных переменных, масштабирование признаков и другие методы подготовки данных для дальнейшего исследования.

Полученные знания и навыки по использованию указанных библиотек и предобработке данных являются важным шагом в освоении анализа данных с помощью Python и будут полезны в дальнейшей профессиональной деятельности.

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

* + 1. <https://numpy.org/doc/stable/reference/generated/numpy.matrix.html>(дата обращения: 24.12.23).
    2. <https://seaborn.pydata.org/installing.html>(дата обращения: 24.12.23).
    3. <https://pandas.pydata.org/docs/reference/api/pandas.get_dummies.html>(дата обращения: 24.12.23).
    4. <https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.tight_layout.html>(дата обращения: 24.12.23).
    5. <https://www.kaggle.com/code/carlmarco/spotify-songs/data> (дата обращения: 24.12.23).