МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Омский государственный технический университет»

Факультет информационных технологий и компьютерных систем

Кафедра «Прикладная математика и фундаментальная информатика»

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

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].

```
# РГР по дисциплине "Практикум по программированию"
## Spotify songs
1. track_id - id песни
2. track_name - название песни
3. track_artist - автор песни
4. track_popularity - оценка популярности трека по 100-балльной шкале
5. track_album_id - id альбому, которому принадлежит эта песня
6. track_album_name - название альбома, которому принадлежит эта песня
7. track_album_release_date - дата релиза альбома
8. playlist_name - название плейлиста с этим треком
9. playlist_id - id плейлиста с этим
10. playlist_genre - жанр плейлиста
11. playlist_subgenre - субжанр плейлиста
12. danceability - описывает, насколько трек подходит для танцев на основе комбинации музыкальных элементов, включая темп, стабильность ритма, силу бита
и общую регулярность, изменяется от 0 до 1
13. energy - представляет собой перцептивную меру интенсивности и активности песни, изменяется от 0 до 1
14. key - общий ключ трека, сопоставляется с высокими нотами
15. loudness - громкость трека
16. mode - мажор обозначается цифрой 1, а минор – 0
17. speechiness - измеряет насколько речь в песне понятна, изменяется от 0 до 1
18. acousticness - измеряет насколько трек акустический, изменяется от 0 до 1
19. instrumentalness - измеряет насколько трек инструментальный, меньше содержит речи, изменяется от 0 до 1
20. liveness - измеряет как сильно слышно зал, чем ближе к 1, тем больше шанс, что это живое исполнение, концерт
21. valence - измеряет позитивность трека, изменяется от 0 до 1
22. tempo - тем трека в ударах в минуту
23. duration_ms - продолжительность трека в миллисекундах
```

Рисунок 1 - Файл Readme.md

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

```
[81]  data = pd.read_csv("spotify_songs.csv")
```

+ Code + Markdown

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.

track_id	track_name	track_artist	track_popularity	track_album_id	track_album_name	track_album_release_date	playlist_name	playlist_id	playlist_genre
6f807x0ima9a1j3VPbc7VN	I Don't Care (with Justin Bieber) - Loud Luxur...	Ed Sheeran	66	2oCs0DGTsRO98Gh5ZSl2Cx	I Don't Care (with Justin Bieber) [Loud Luxury...	2019-06-14	Pop Remix	37i9dQZF1DXcZDD7cfEKhW	pop
0r7CVbZTWZgbTCYdfa2P31	Memories - Dillon Francis Remix	Maroon 5	67	63rPSO264uRjW1X5E6cWv6	Memories (Dillon Francis Remix)	2019-12-13	Pop Remix	37i9dQZF1DXcZDD7cfEKhW	pop
1z1Hg7Vb0AhHDiEmnDE79l	All the Time - Don Diablo Remix	Zara Larsson	70	1HoSmj2eLcsrR0vE9gThr4	All the Time (Don Diablo Remix)	2019-07-05	Pop Remix	37i9dQZF1DXcZDD7cfEKhW	pop
75FpbthrwQmzHlBJLuGdC7	Call You Mine - Keanu Silva Remix	The Chainsmokers	60	1nqYsOef1yKKuGOVchbsk6	Call You Mine - The Remixes	2019-07-19	Pop Remix	37i9dQZF1DXcZDD7cfEKhW	pop
1e8PAfcKUYoKkxPhrHqw4x	Someone You Loved - Future	Lewis Capaldi	69	7m7vv9wlQ4i0LFuJiE2zsQ	Someone You Loved (Future	2019-03-05	Pop Remix	37i9dQZF1DXcZDD7cfEKhW	pop

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.

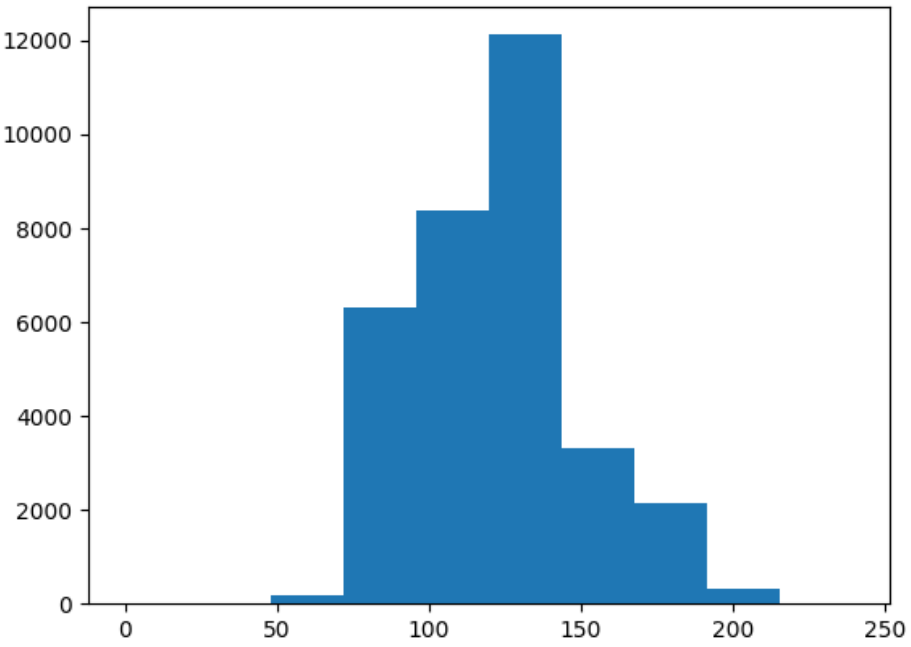

Рисунок 4 - Гистограмма для признака tempo

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.

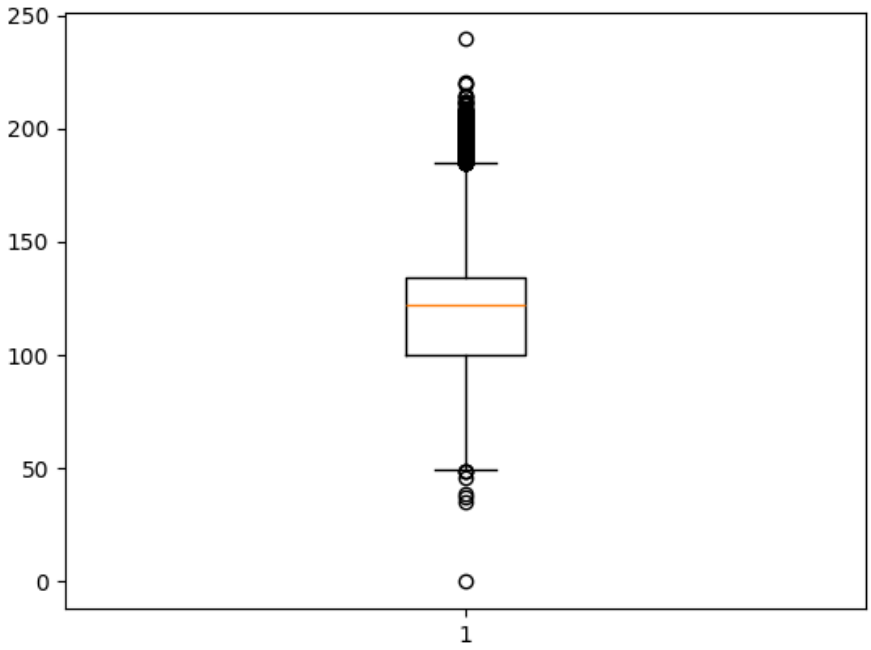

Рисунок 5 - Диаграмма "Ящик с усами" для признака tempo

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.

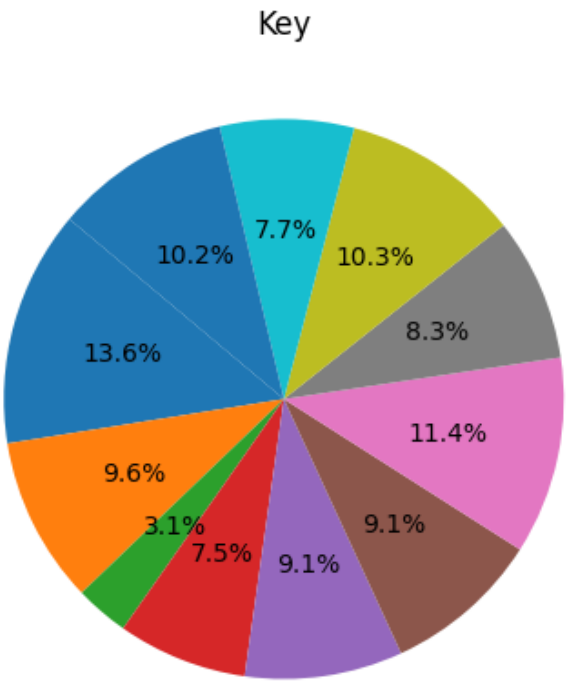

Рисунок 6 - Круговая диаграмма для признака key

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.

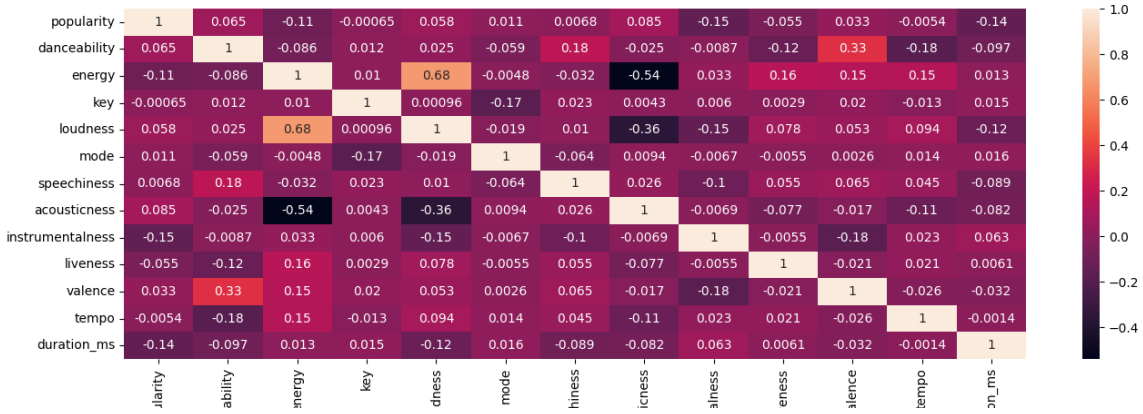

Рисунок 7 - Тепловая карта

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.

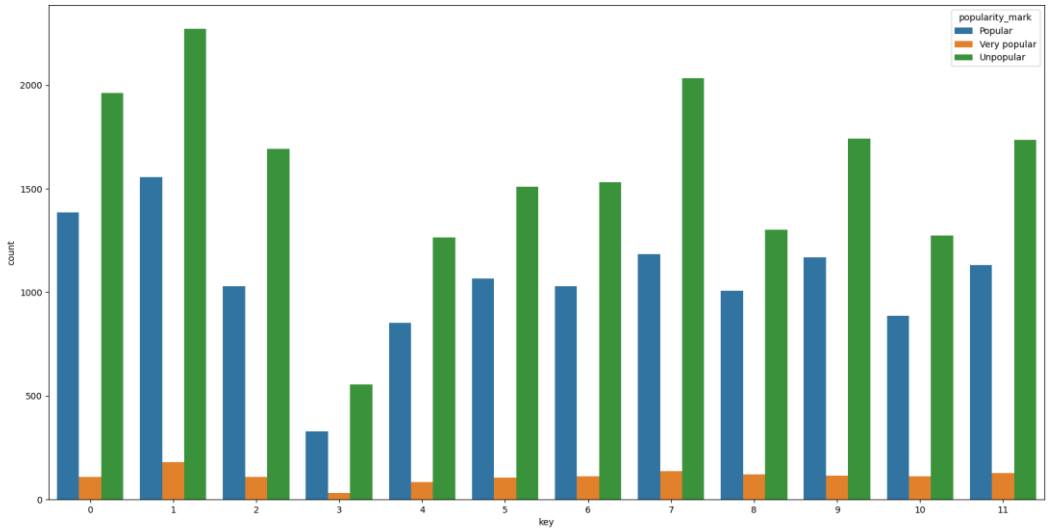

Рисунок 8 - График countplot для key и popularity

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.

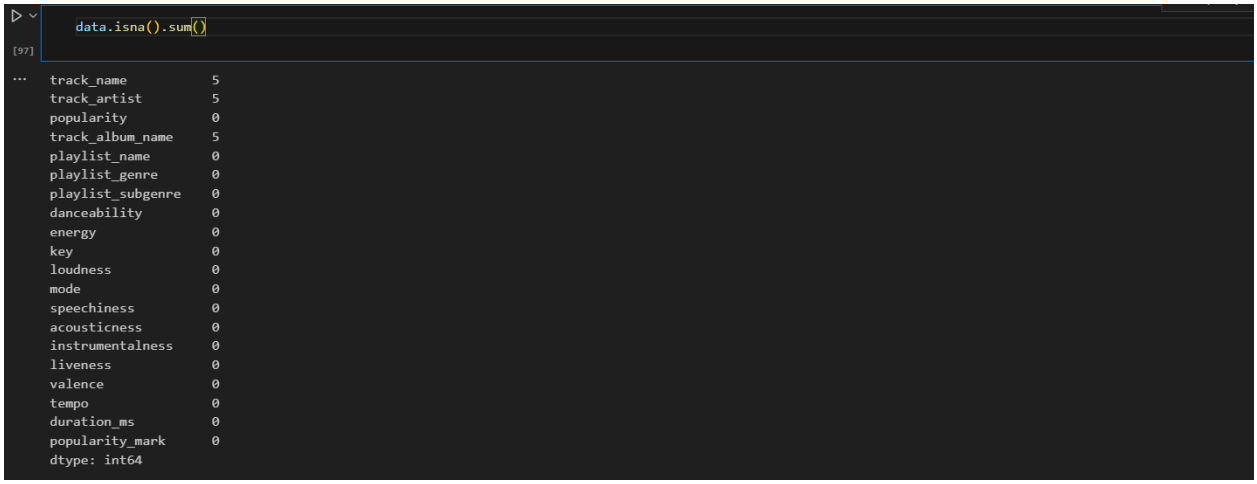

Рисунок 9 - Вывод количества пропусков в наборе данных

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

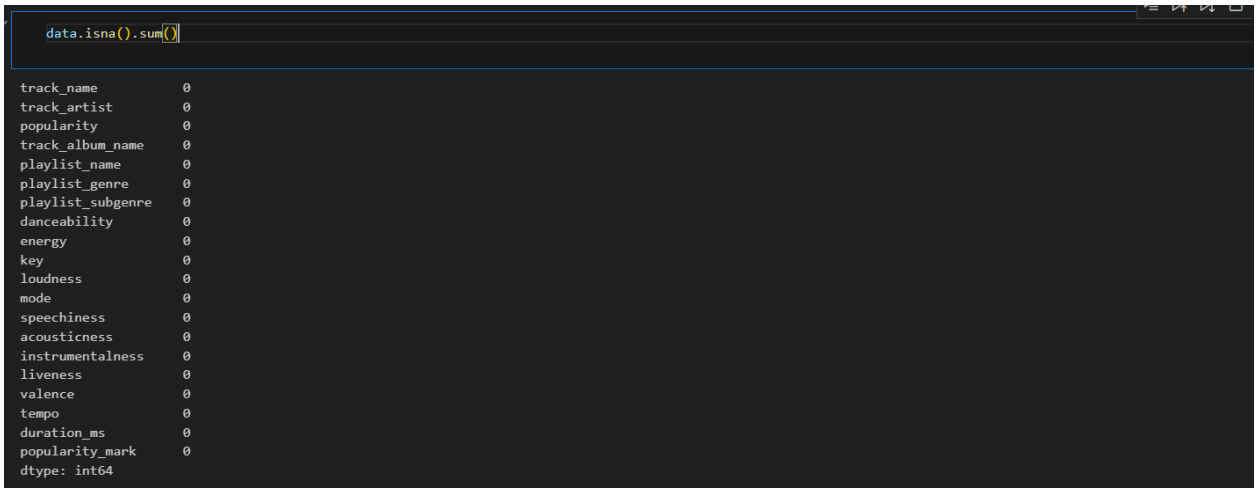

Рисунок 10 - Вывод количества пропущенных значений после дропа

После удаления пустых значений были удалены дубликаты.

После удаления дубликатов были удалены ненужные столбцы.

```
data.drop(['track_id', 'track_album_id', 'playlist_id', 'track_album_release_date'], inplace=True, axis=1)
```

Рисунок 11 - Удаление ненужных столбцов

Также нужно было применить для некоторых категориальных признаков

one-hot кодирование.

```
[101]
to_encode = ["playlist_genre", "playlist_subgenre"]

[102]
for column in to_encode:
    data=data.join(pd.get_dummies(data[column],dtype=int))
    data.drop(column,axis=1,inplace=True)
```

Рисунок 12 - Кодирование категориальных признаков

```
data
```

	track_name	track_artist	popularity	track_album_name	playlist_name	danceability	energy	key	loudness	mode	...	new jack swing	permanent wave	pop edm	post-teen pop	progressive electro house	regg
0	I Don't Care (with Justin Bieber) - Loud Luxur...	Ed Sheeran	66	I Don't Care (with Justin Bieber) [Loud Luxury...	Pop Remix	0.748	0.916	6	-2.634	1	...	0	0	0	0	0	
1	Memories - Dillon Francis Remix	Maroon 5	67	Memories (Dillon Francis Remix)	Pop Remix	0.726	0.815	11	-4.969	1	...	0	0	0	0	0	
2	All the Time - Don Diablo Remix	Zara Larsson	70	All the Time (Don Diablo Remix)	Pop Remix	0.675	0.931	1	-3.432	0	...	0	0	0	0	0	
3	Call You Mine - Keanu Silva Remix	The Chainsmokers	60	Call You Mine - The Remixes	Pop Remix	0.718	0.930	7	-3.778	1	...	0	0	0	0	0	
4	Someone You Loved - Future Humans Remix	Lewis Capaldi	69	Someone You Loved (Future Humans Remix)	Pop Remix	0.650	0.833	1	-4.672	1	...	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
32828	City Of Lights - Official Radio Edit	Lush & Simon	42	City Of Lights (Vocal Mix)	♥ EDM LOVE 2020	0.428	0.922	2	-1.814	1	...	0	0	0	0	1	

Рисунок 13 - Набор данных после выполненной предобработки

После выполненной предобработки набор данных был сохранен.

```
data.to_csv("new_spotify_songs.csv", sep=";", index=False)
```

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 (датаобращения: 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 (датаобращения: 24.12.23).
5. https://www.kaggle.com/code/carlmarco/spotify-songs/data (дата обращения: 24.12.23).