

## Сбор данных о музыкальных композициях, выпущенных на Spotify

*Фатеенков Данила Витальевич*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

### Аннотация

В статье описывается процесс сбора данных о музыке, которая была выпущена в 2021-м году на музыкальной платформе Spotify. Рассматривается сторонний модуль для работы с API Spotify – Spotipy, а также представлен процесс преобразования данных в .csv формат с сохранением необходимых параметров для последующего анализа.

**Ключевые слова:** анализ данных, Python, JSON, Spotipy, Spotify.

## Collecting of music track data released on Spotify

*Fateenkov Danila Vitalievich*

*Sholom-Aleichem Priamursky State University*

*Student*

### Abstract

The article describes the process of collecting data about the music, which was released in 2021 on the music platform Spotify. The third-party module for working with API Spotify - Spotipy is considered, and also the process of data conversion into .csv format with saving the necessary parameters for further analysis is presented.

**Keywords:** data analysis, Python, JSON, Spotipy, Spotify.

## 1. Введение

### 1.1 Актуальность

Сбор и анализ большого количества данных актуален в настоящее время для многих сфер человеческой деятельности. Анализ данных позволяет получать необходимые сведения в удобном для человека формате.

Такой вид деятельности также востребован в музыкальной сфере развлечений. В настоящее время музыка является таким же бизнесом, как и киноиндустрия и другие сферы развлечения. Поэтому понимать и уметь определять тренды и тенденции в музыкальной сфере в настоящее время очень важно.

Для решения такой задачи необходимо пользоваться инструментами для анализа данных, например, языком программирования Python или специализированным программным обеспечением (например, Orange).

Одной из самых популярных музыкальных платформ является Spotify, на котором ежедневно выпускаются сотни композиций. Для работы с данным

сервисом для разработчиков создан API, который также позволяет собирать данные о музыке и анализировать их.

## 1.2 Обзор исследований

В.А. Мельникова и Д.А. Медведев описали алгоритм сравнительного анализа данных с использованием языка программирования Python и модулей NumPy и Pandas [1]. М.Э. Чибирова описала методику регрессионного анализа данных с реализацией алгоритма на Python и R [2]. В.Ю. Ильичев и Е.А. Юрик представили алгоритм для обработки больших массивов данных на Python с применением модуля Pandas [3]. И.Х. Утакаева провела статистический анализ данных автомобильного рынка с применением Python и модуля Pandas [4]. А.О. Захаренков и Г.С. Евдокимова провели анализ языков Python, R и Ruby в контексте анализа данных на примере задачи кластеризации [5].

## 1.3 Цель исследования

Цель – собрать данные о музыкальных произведениях, выпущенных на музыкальной платформе Spotify для последующего анализа.

## 2. Материалы и методы

Для решения поставленной задачи используется язык Python, сторонняя библиотека Spotify.

## 3. Результаты и обсуждения

Перед началом сбора данных был получен список всех доступных на данный момент музыкальных жанров в Spotify. В настоящее время представлено 126 жанров. Получить список можно, используя функционал API платформы, через с URL-запрос по следующему URL:

```
https://api.spotify.com/v1/recommendations/available-genre-seeds
```

После формирования списка жанров был загружен модуль для языка программирования Python: Spotify. Данный модуль предназначен для работы с API музыкальной платформы и предоставляет разработчику весь необходимый функционал для работы с сервисом в своих приложениях. В список основных функций модуля можно отнести:

1. Получение данных об исполнителе (его имя, популярные музыкальные композиции, альбомы и сборники, количество подписчиков и другие данные).
2. Получение данных о слушателе (его любимые композиции, плейлисты, ID и количество подписчиков, а также другие данные).
3. Получение данных о музыке, альбомах, сборниках (название, исполнитель, популярность по 100 бальной шкале и другие параметры).
4. Изменение плейлиста пользователя, создание новых плейлистов.

## 5. Осуществлять поиск по платформе.

Модуль позволяет выполнять также и другие функции, но в рамках поставленной задачи необходимо применить методы получения данных о музыкальных композициях.

Установить `Spotipy` можно, применяя `pip`:

```
pip install spotipy
```

Для взаимодействия с функционалом модуля, необходимо получить токен авторизации. Получение токена происходит через протокол OAuth. OAuth – протокол авторизации, который позволяет предоставить третьей стороне доступ к защищённым ресурсам пользователя без передачи логина и пароля. Данный протокол применяется во многих сервисах, в том числе и отечественных (ВКонтакте, например) и последняя версия протокола (она же OAuth 2.0) надёжные методы для обеспечения безопасности данных пользователя при авторизации (одним из преимуществ являются токены с коротким сроком действия, которые предоставляет пользователю долгосрочную авторизацию).

Также получить доступ к API можно путём создания отдельного приложения в системе Spotify for Developers. В таком случае для работы с API необходимо получить Client ID и Client Secret ID. Данные идентификаторы генерируются автоматически при создании нового приложения.

Авторизация с использованием Client ID и Client Secret ID представлена следующим образом: задаются разработчиком (или алгоритмом при подключении к удалённому серверу приложения) ключи доступа к приложению, после ключи указываются в качестве параметров в функции `SpotifyClientCredentials()` и вызовом метода `Spotify` производится подключение к API музыкальной платформы:

```
import spotipy
from spotipy.oauth2 import SpotifyClientCredentials

client_id = "CLIENT_ID"
client_secret = "CLIENT_SECRET"
client_credentials_manager =
SpotifyClientCredentials(client_id=client_id,
client_secret=client_secret)
sp =
spotipy.Spotify(client_credentials_manager=client_credentials_manager)
```

После установления соединения с API Spotify, можно начать собирать данные. Для получения данных по поисковому запросу необходимо использовать метод “Search”, являющийся одним из основных методов API. Данный метод позволяет искать контент на платформе и получать результат

в формате JSON. Искать можно исполнителей, музыкальные композиции, альбомы, плейлисты, шоу и эпизоды (также допускается поиск нескольких категорий, в таком случае параметры поиска разделяются запятой). Можно установить ограничение на количество возвращаемых результатов поиска (от 0 до 50), задавать страну, музыка из которой должна быть найдена и смещение (от 0 до 1000). Запрос посылается на сервер API сервиса: «[api.spotify.com/v1/search](https://api.spotify.com/v1/search)» и может выглядеть следующим образом:

```
https://api.spotify.com/v1/search?q=genre%3Aall&type=track%2
Cartist&limit=10&offset=3
```

В качестве результата выполнения поиска по данным параметрам будет представлен список из максимум 10-и музыкальных композиций и исполнителей всех жанров со смещением в 3 элемента поиска. Метод “Search” может принимать следующие параметры:

q – поисковый запрос. Можно искать исполнителей, музыку, производить поиск по определённому промежутку времени, искать по идентификационному номеру музыкального произведения (ISRC) или штрих-коду произведения (UPC). Также для метода представлены 2 уникальных тега: new и hipster. Первый выполняет поиск среди музыки, которая была выпущена в течение последних двух недель, а второй тег производит поиск среди музыки, популярность которой ниже 10%.

type – тип контента, который необходимо найти.

limit – ограничение на количество возвращаемых результатов.

offset – смещение на определённое количество элементов.

Получить треки определённого жанра через Spotify можно следующим образом:

```
query = ["genre:acoustic year:2019"]
result = sp.search(query, limit=10, offset=0, type='track')
for i in range(len(result['tracks']['items'])):
    print(result['tracks']['items'][i])
```

Данный код отправляет на сервер запрос на поиск первых 10-и музыкальных произведений в жанре Acoustic, релиз которых состоялся в 2019-м году. После выполнения запроса выводятся данные каждого трека в формате JSON.

Возвращаемый с сервера JSON-файл принимает формат:

```
{  "tracks": {
    "href": "*ссылка*",
    "items": [],
    "limit": 10,
    "next": null,
    "offset": 0,
    "previous": null,
```

```

        "total": 0 }
    }

```

Набор данных, после выполнения сбора, представлен в виде массива, который содержит следующий набор параметров:

1. Название трека
2. Имя автора
3. Жанр музыкального произведения
4. ID произведения
5. URL
6. URI
7. Дата релиза на платформе Spotify
8. Параметр Acousticness
9. Параметр Instrumentalness
10. Параметр Energy
11. Параметр Danceability
12. Параметр Valence
13. Темп (количество ударов в минуту)
14. Громкость произведения
15. Популярность трека

Массив формируется при обработке JSON-контейнера. Конечный результат сохраняется на ПК в формате .csv и может быть использован для последующего анализа.

JSON-контейнер содержит данные, которые не требуются для анализа (например, страны, в которых можно слушать трек), поэтому сформировать массив без предварительной обработки не получится (см. рис. 1).

```

{
  'album': {
    'album_type': 'single',
    'artists': [
      {
        'external_urls': {
          'spotify': 'https://open.spotify.com/artist/2uFUBdaVGtyMqckSeCl0Qj'
        },
        'href': 'https://api.spotify.com/v1/artists/2uFUBdaVGtyMqckSeCl0Qj',
        'id': '2uFUBdaVGtyMqckSeCl0Qj',
        'name': 'Ludovico Einaudi',
        'type': 'artist',
        'uri': 'spotify:artist:2uFUBdaVGtyMqckSeCl0Qj'
      }
    ],
    'available_markets': [
      'AD', 'AE', 'AG', 'AL', 'AM', 'AO', 'AR', 'AT', 'AU', 'AZ', 'BA', 'BB', 'BD', 'BE', 'BF', 'BG', 'BH', 'BI', 'BJ', 'BN', 'BO', 'BR', 'BS', 'BT', 'BW', 'BY', 'BZ', 'CA', 'CD', 'CG', 'CH', 'CI', 'CL', 'CM', 'CO', 'CR', 'CV', 'CW', 'CY', 'CZ', 'DE', 'DJ', 'DK', 'DM', 'DO', 'DZ', 'EC', 'EE', 'EG', 'ES', 'FI', 'FJ', 'FM', 'FR', 'GA', 'GB', 'GD', 'GE', 'GH', 'GM', 'GN', 'GQ', 'GR', 'GT', 'GW', 'GY', 'HK', 'HN', 'HR', 'HT', 'HU', 'ID', 'IE', 'IL', 'IN', 'IQ', 'IS', 'IT', 'JM', 'JO', 'JP', 'KE', 'KG', 'KH', 'KI', 'KM', 'KN', 'KR', 'KW', 'KZ', 'LA', 'LB', 'LC', 'LI', 'LK', 'LR', 'LS', 'LT', 'LU', 'LV', 'LY', 'MA', 'MC', 'MD', 'ME', 'MG', 'MH', 'MK', 'ML', 'MN', 'MO', 'MR', 'MT', 'MU', 'MV', 'MW', 'MX', 'MY', 'MZ', 'NA', 'NE', 'NG', 'NI', 'NL', 'NO', 'NP', 'NR', 'NZ', 'OM', 'PA', 'PE', 'PG', 'PH', 'PK', 'PL', 'PS', 'PT', 'PW', 'PY', 'QA', 'RO', 'RS', 'RU', 'RW', 'SA', 'SB', 'SC', 'SE', 'SG', 'SI', 'SK', 'SL', 'SM', 'SN', 'SR', 'ST', 'SV', 'SZ', 'TD', 'TG', 'TH', 'TJ', 'TL', 'TN', 'TO', 'TR', 'TT', 'TV', 'TW', 'TZ', 'UA', 'UG', 'US', 'UY', 'UZ', 'VC', 'VE', 'VN', 'VU', 'WS', 'XK', 'ZA', 'ZM', 'ZW'
    ],
    'external_urls': {
      'spotify': 'https://open.spotify.com/album/43wcKhgYUpeh2cwvBtqDYc'
    },
    'href': 'https://api.spotify.com/v1/albums/43wcKhgYUpeh2cwvBtqDYc',
    'id': '43wcKhgYUpeh2cwvBtqDYc',
    'images': [
      {
        'height': 640,
        'url': 'https://i.scdn.co/image/ab67616d0000b2738a9699b9398b39c074e226c4',
        'width': 640
      },
      {
        'height': 300,
        'url': 'https://i.scd

```

Рисунок 1. Необработанные данные в формате JSON, полученные при проведении поиска музыки через API Spotify

Из необработанного JSON-контейнера были получены название трека, исполнитель, URL, URI и ID (он также используется для поиска остальных параметров композиции), а также дата релиза произведения и параметр популярности. Так как JSON-контейнер содержит в себе большое количество индексов, то легче будет присваивать новой переменной значение полученного результата с обращением к индексу 'items':

```
query_data = query['tracks']['items']
```

Для получения необходимых данных из JSON представлены следующие обращения к элементам по индексу:

1. Название композиции: `query_data['name']`
2. Исполнитель композиции: `query_data['artists'][0]['name']`
3. ID: `query_data['id']`
4. URL: `query_data['external_urls']['spotify']`
5. URI: `query_data['uri']`
6. Дата релиза: `query_data['album']['release_date']`
7. Популярность: `query_data['popularity']`

Также необходимо получить следующие параметры: Acousticness, Instrumentalness, Energy, Danceability, Valence, темп и громкость. Для получения таких параметров необходимо использовать метод "audio\_features()". Данный метод предоставляет пользователю характеристики определённого трека. На вход метод принимает один параметр: ID композиции (также возможно получать характеристики нескольких треков, для этого необходимо разделить ID запятой). Вызов метода представлен следующим образом:

```
features_query = sp.audio_features(track_id)
```

Для получения необходимых характеристик также нужно обращаться к определённым элементам JSON-контейнера:

1. Acousticness: `features_query[0]['acousticness']`
2. Instrumentalness: `features_query[0]['instrumentalness']`
3. Energy: `features_query[0]['energy']`
4. Danceability: `features_query[0]['danceability']`
5. Valence: `features_query[0]['valence']`
6. Tempo: `features_query[0]['tempo']`
7. Loudness: `features_query[0]['loudness']`

Также в итоговом массиве представлены жанры, к которым относятся те или иные произведения. Так как список жанров был получен заранее и для поиска по жанрам используется файл, в котором представлен данный список, то этот параметр вносится в массив как элемент файла, из которого происходит считывание в цикле.

Для метода "search" были установлены следующие ограничения: `limit = 50`, `offset = 0-500` (увеличивается после обработки каждых 50-и элементов), `type = 'track'`.

Полученный массив лучше всего представить в виде объекта DataFrame, который представлен в модуле Pandas. DataFrame можно легко экспортировать в формат .csv с необходимым разделителем (для работы с полученными данными в Orange можно использовать точку с запятой как разделитель) и кодировкой:

```
df = pandas.DataFrame(data)
df.head()
df.to_csv("music_spotify_data.csv", sep=';', encoding='utf-8', index=False)
```

	name ...	URI
0	Pano ...	spotify:track:08MFgEQeVLF37EyZ7jcwLc
1	Habang Buhay ...	spotify:track:0zANeX4R6uWb82gCQAguOD
2	Habang Buhay ...	spotify:track:63bmIgh9sS6sX5Sc7MetGq
3	Give Me Your Forever ...	spotify:track:4mzP5mHkRvGxdhdGdAH7EJ
4	You and Me On The Rock (feat. Lucius) ...	spotify:track:61X73CJcpwGIxgQgKEXbw0
...	...	...
56475	Canciones para Clases de Yoga ...	spotify:track:169N2g4a4CLMHD30b4Eul7
56476	Running - Live ...	spotify:track:4WPWbqafJhHvN02eXAIN5Z
56477	Our First Whale ...	spotify:track:19CTmWud1ieKRQj9KsB2iL
56478	Sonidos de Agua para Dormir: Bosque, Pt. 27 ...	spotify:track:3AFxeCtMvrLajoD9KSalIT
56479	Sorongong ...	spotify:track:7tWkX19sjhfr70tj1Zvamr

[56480 rows x 15 columns]

Рисунок 2. DataFrame с данными, которые были получены при поиске через API Spotify

Таким образом был сформирован массив данных с 56480 уникальными элементами и выгружен в формате .csv (см. рис. 3).

1	name	author	genre	release date	ID	acoustic	instrumenter	energy	danceabil	valence	tempo	loudness	popularity	URL	URI
2	Pano	Zack Tabu	acoustic	06.12.2021	08MFgEQeVLF37EyZ7jcwLc	0.868	0.0145	0.457	0.375	0.415	174.839	-7.018	82	https://oç.spotify:track:08MFgEQeVLF37EyZ7jcwLc	spotify:track:08MFgEQeVLF37EyZ7jcwLc
3	Habang Buhay	Zack Tabu	acoustic	17.09.2021	0zANeX4R6uWb82gCQAguOD	0.088	0.0	0.668	0.62	0.183	118.013	-5.719	78	https://oç.spotify:track:0zANeX4R6uWb82gCQAguOD	spotify:track:0zANeX4R6uWb82gCQAguOD
4	Habang Buhay	Zack Tabu	acoustic	15.10.2021	63bmIgh9sS6sX5Sc7MetGq	0.088	0.0	0.668	0.62	0.183	118.013	-5.719	76	https://oç.spotify:track:63bmIgh9sS6sX5Sc7MetGq	spotify:track:63bmIgh9sS6sX5Sc7MetGq
5	Give Me Your Forever	Zack Tabu	acoustic	15.10.2021	4mzP5mHkRvGxdhdGdAH7EJ	0.279	0.0	0.363	0.627	0.301	99.905	-8.127	81	https://oç.spotify:track:4mzP5mHkRvGxdhdGdAH7EJ	spotify:track:4mzP5mHkRvGxdhdGdAH7EJ
6	You and Me On The Rock (feat. Lucius)	Brandi Caracoustic	acoustic	01.10.2021	61X73CJcpwGIxgQgKEXbw0	0.15	1.81e-06	0.686	0.568	0.725	172.075	-6.635	65	https://oç.spotify:track:61X73CJcpwGIxgQgKEXbw0	spotify:track:61X73CJcpwGIxgQgKEXbw0
7	See the World	Brett Deni	acoustic	23.04.2021	0YPX6ztcV0.0245	0.0245	3.59e-05	0.638	0.616	0.507	89.989	-7.981	64	https://oç.spotify:track:0YPX6ztcV0.0245	spotify:track:0YPX6ztcV0.0245
8	Worry No	Amos Lee	acoustic	11.08.2021	6Sf1jXhbf0.568	0.568	0.00128	0.435	0.272	0.27	78.721	-10.096	63	https://oç.spotify:track:6Sf1jXhbf0.568	spotify:track:6Sf1jXhbf0.568
9	Right on Time	Brandi Caracoustic	acoustic	01.10.2021	5i4WgW0i0.626	0.626	1.83e-05	0.269	0.419	0.152	121.695	-9.388	60	https://oç.spotify:track:5i4WgW0i0.626	spotify:track:5i4WgW0i0.626
10	Binibini	Zack Tabu	acoustic	26.03.2021	2X5AFygz0.423	0.423	1.54e-06	0.374	0.642	0.0979	129.863	-10.606	75	https://oç.spotify:track:2X5AFygz0.423	spotify:track:2X5AFygz0.423
11	This Time	Brandi Caracoustic	acoustic	01.10.2021	1Cu8Qne0.841	0.841	0.0	0.326	0.606	0.414	91.626	-10.031	61	https://oç.spotify:track:1Cu8Qne0.841	spotify:track:1Cu8Qne0.841
12	Jericho	Andrew R	acoustic	14.05.2021	0MwxVd70.189	0.189	0.0	0.716	0.501	0.363	70.919	-4.661	58	https://oç.spotify:track:0MwxVd70.189	spotify:track:0MwxVd70.189
13	Range Rover	Ben Recto	acoustic	21.05.2021	1hcl2TWB0.111	0.111	0.0	0.68	0.712	0.882	144.966	-6.051	59	https://oç.spotify:track:1hcl2TWB0.111	spotify:track:1hcl2TWB0.111
14	Binibini	Zack Tabu	acoustic	15.10.2021	55RMhrPC0.423	0.423	1.54e-06	0.374	0.642	0.0979	129.863	-10.606	73	https://oç.spotify:track:55RMhrPC0.423	spotify:track:55RMhrPC0.423
15	Pontiac	Mat Kear	acoustic	26.02.2021	1dNxCqZy0.12	0.12	5.27e-05	0.905	0.379	0.38	82.253	-6.466	60	https://oç.spotify:track:1dNxCqZy0.12	spotify:track:1dNxCqZy0.12
16	Anywhere	Mat Kear	acoustic	02.04.2021	0AHUTBer0.368	0.368	0.00101	0.788	0.602	0.404	112.987	-6.837	59	https://oç.spotify:track:0AHUTBer0.368	spotify:track:0AHUTBer0.368
17	Broken Heart	Brandi Caracoustic	acoustic	01.10.2021	0vAGNjNf0.00695	0.00695	1.8e-06	0.729	0.444	0.388	95.258	-5.442	58	https://oç.spotify:track:0vAGNjNf0.00695	spotify:track:0vAGNjNf0.00695
18	Feels Like	Drew Holc	acoustic	12.02.2021	38mvvvoJ0.088	0.088	8.82e-05	0.508	0.603	0.157	72.983	-8.671	57	https://oç.spotify:track:38mvvvoJ0.088	spotify:track:38mvvvoJ0.088
19	Paul Newman	Brett Deni	acoustic	30.07.2021	4pAjn6Pcc0.0158	0.0158	0.000164	0.766	0.645	0.665	157.963	-8.284	59	https://oç.spotify:track:4pAjn6Pcc0.0158	spotify:track:4pAjn6Pcc0.0158
20	Without a Doubt	Drew Holc	acoustic	01.10.2021	1CYBnHY0.656	0.656	0.000327	0.435	0.493	0.485	87.952	-12.334	58	https://oç.spotify:track:1CYBnHY0.656	spotify:track:1CYBnHY0.656
21	heavenly	Wilder W	acoustic	26.03.2021	3EVNm0d0.626	0.626	0.0	0.628	0.481	0.607	174.002	-6.118	56	https://oç.spotify:track:3EVNm0d0.626	spotify:track:3EVNm0d0.626
22	Atlantis	Noah Gun	acoustic	08.10.2021	4Lp2SmcF0.54	0.54	0.0462	0.555	0.591	0.459	83.459	-10.702	60	https://oç.spotify:track:4Lp2SmcF0.54	spotify:track:4Lp2SmcF0.54
23	I Believe	JJ Heller	acoustic	01.01.2021	3FRARAKL0.449	0.449	1.2e-05	0.359	0.534	0.267	131.804	-9.311	60	https://oç.spotify:track:3FRARAKL0.449	spotify:track:3FRARAKL0.449
24	Stand by Me	Boyce Ave	acoustic	15.01.2021	068qo0w10.872	0.872	1.03e-05	0.266	0.728	0.307	115.189	-7.322	69	https://oç.spotify:track:068qo0w10.872	spotify:track:068qo0w10.872

Рисунок 3. Данные о музыкальных композициях, представленные в виде таблицы

Таким образом, был сформирован набор данных и приведён к виду, при котором возможен дальнейший анализ. Анализ можно проводить в

программном обеспечении Orange, которое было также разработано с использованием языка программирования Python.

Из полученных данных, при использовании модуля Violin Plot, можно определить жанры, в которых выходила самая популярная музыка в 2021-м году: pop, dance, latin и hip-hop. Наименее популярными стали следующие жанры: Chicago-house, honky-tonk, Detroit-techno и музыка для детей (Spotify выделяет 'kids' как отдельный жанр). Самая энергичная музыка встречалась в жанрах: Death-metal, Happy, Metalcore, Grindcore и Metal. Музыка с самым высоким темпом: Drum and Bass (среднее значение темпа = 175 ударов в минуту), Happy (170 ударов в минуту) и Hardstyle (около 160 ударов в минуту).

Благодаря гистограммам модуля Distributions можно определить следующее:

1. Количество треков, популярность которых не превысила 2%, составляет 3.6% (то есть 2050 треков) от общего числа. Почти половина композиций (47.86% или 27045 треков) достигла отметки на промежутке от 36% до 60% (средний показатель). Меньше всего композиций, популярность которых превысила 80% - 1.33% (747 треков) от общего количества (см. рис. 4).

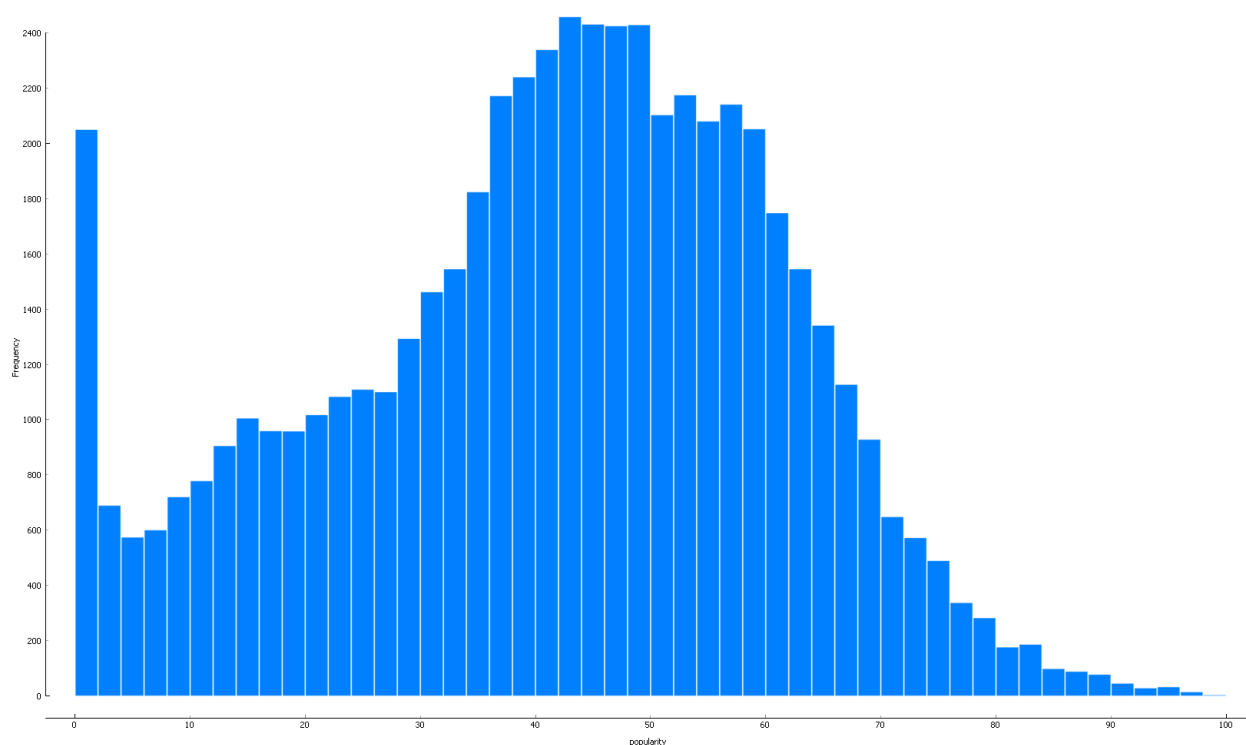


Рисунок 4. Гистограмма по параметру Popularity

2. Темп преобладает на промежутке от 120-и ударов в минуту до 130-и. 10421 композиция попадают в этот промежуток (то есть 18.45% от общего числа) (см. рис. 5).

На данном этапе нельзя сказать – какой жанр является преобладающим у аудитории музыкального сервиса. Используя модуль Feature Statistics



можно определить средние значения каждого параметра: среднее значение параметра Tempo составляет 122 удара в минуту, а среднее значение популярности = 41%.

Также в набор данных попали музыкальные произведения, которые изначально были не выпущены в 2021-м году – это Remastered-версии ранее выпущенных произведений, поэтому они имеют свой уникальный ID и учитываются при анализе.

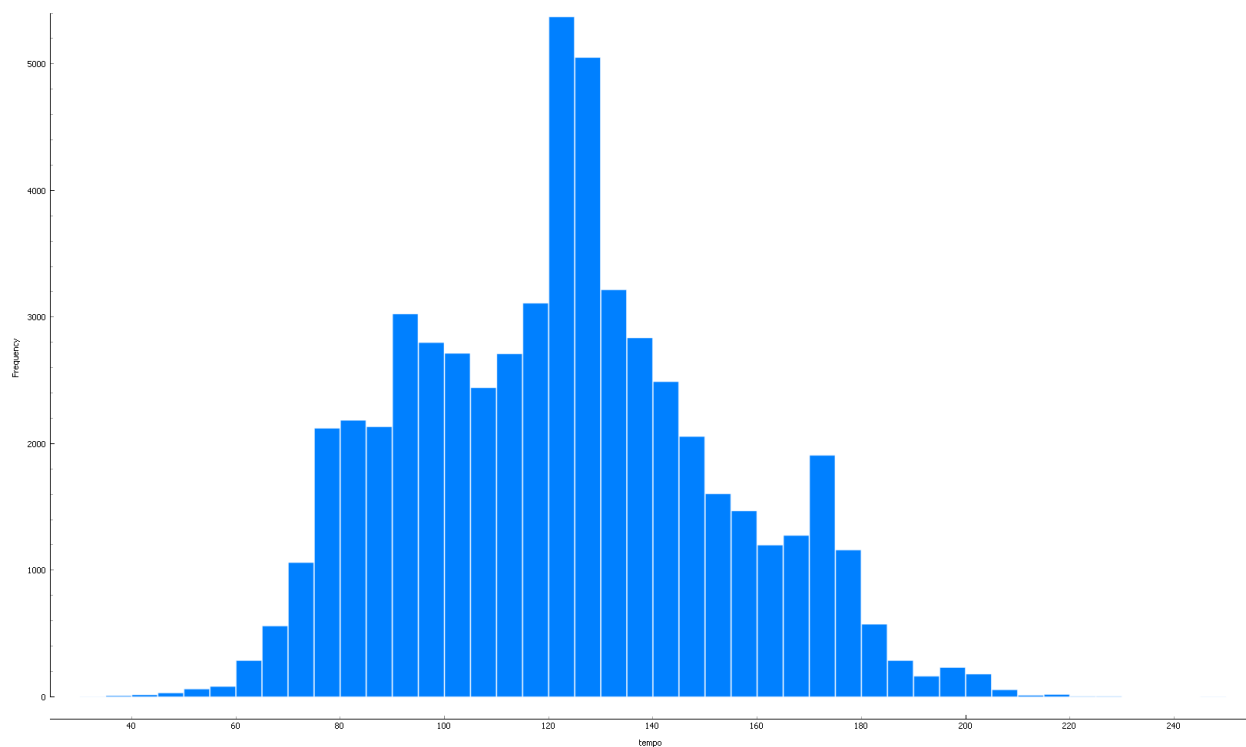


Рисунок 5. Гистограмма по параметру Tempo

Заметных корреляций популярности с другими параметрами не наблюдалось, но можно определить следующие зависимости:

1. Параметр Energy падает при повышении Acousticness (коэффициент корреляции:  $acousticness-energy = -0.792$ , преобладает обратная зависимость).

2. При повышении энергичности, увеличивается и громкость музыкального произведения ( $energy-loudness = +0.782$ , преобладает прямая линейная зависимость).

Также Orange имеет в качестве встроенного функционала модели обучения, которые можно использовать для прогнозирования результатов при определённых значениях. Но практической пользы от прогнозирования на основе заданных пользователем значений параметров может не быть, так как коэффициент популярности также зависит от алгоритмов и музыкальных редакторов Spotify. Значительный прирост популярности исполнитель может получить после добавления его музыки в алгоритмические или редакторские плейлисты, поэтому параметр популярности независим от остальных параметров характеристики музыкального произведения.

Таким образом, с применением языка программирования Python и стороннего модуля Spotipy был собран набор данных и рассмотрен в программе Orange. Полученные данные могут быть подвергнуты более глубокому анализу. Можно отметить удобство работы с API Spotify, а также короткий промежуток времени получения ответа с сервера.

В статье был описан процесс получения данных о музыкальных произведениях, выпущенных на музыкальной платформе Spotify.

### **Библиографический список**

1. Мельникова В.А., Медведев Д.А. Анализ больших данных с использованием Python // Труды братского государственного университета. Серия: естественные и инженерные науки. 2019. Т.1. С. 46-49
2. Чибирова М.Э. Анализ данных и регрессионное моделирование с применением языков программирования Python и R // Научные записки молодых исследователей. 2019. Т.7. № 2. С. 37-45
3. Ильичев В.Ю., Юрик Е.А. Анализ массивов данных с использованием библиотеки Pandas для Python // Научное обозрение. Технические науки. 2020. - № 4. - С. 41-45
4. Утакаева И.Х. Применение пакета статистического анализа Python для анализа данных автомобильного рынка // Вестник Алтайской академии экономики и права. 2019. № 2-2. С. 346-351
5. Захаренков А.О., Евдокимова Г.С. Сравнительный анализ языков Ruby, R и Python в вопросах анализа данных на примере задачи кластеризации // Системы компьютерной математики и их приложения. 2020. № 21. С. 38-44
6. Spotify for Developers URL: <https://developer.spotify.com>. (Дата обращения: 01.04.2022).
7. Github: Spotipy URL: <https://github.com/plamere/spotipy>. (Дата обращения: 25.03.2022).