initial

December 29, 2023

1 Analiza danych

Na podstawie drugiej wersji danych

[85]:
```
import os
import pandas as pd
from config_file import data_path
import matplotlib.pyplot as plt

pd.options.mode.chained_assignment = None  # default='warn'
```

[ ]:
```

```

1.1 artists.jsonl

Plik zawiera informacje o artystach, których utwory można znaleźć w systemie.

Informacje: - id artysty - nazwa / pseudonim artysty - lista gatunków artysty

[86]:
```
df = pd.read_json(os.path.join(data_path, "artists.jsonl"), orient="records",␣
↪lines=True)
```

[87]:
```
print(f"Liczba artystów: {len(df['id'].unique())}")
```

```
Liczba artystów: 27650
```

1.2 tracks.jsonl

Plik zawiera informacje o utworach różnych artystów.

Informacje: - id utworu - nazwa utworu - id artysty - długość - popularność - statystyki odnoszące się do taneczności itp.

[88]:
```
df = pd.read_json(os.path.join(data_path, "tracks.jsonl"), orient="records",␣
↪lines=True)
```

[89]:
```
print(f"Liczba utworów: {len(df['id'].unique())}")
```

```
Liczba utworów: 129648
```

1.3 users.jsonl

Plik zawiera informacje o użytkownikach platformy

Informacje: (kluczowe) - id użytkownika - nazwa

[90]:
```
df = pd.read_json(os.path.join(data_path, "users.jsonl"), orient="records",␣
↪lines=True)
```

[91]:
```
print(f"Liczba użytkowników: {len(df['user_id'].unique())}")
```

```
Liczba użytkowników: 500
```

1.4 sessions.jsonl

Plik zawiera informacje o sesjach użytkowników

Informacje: (kluczowe) - id sesji - id utworu - typ zdarzenia - timestamp

[92]:
```
df = pd.read_csv(os.path.join(data_path, 'sessions_timestamp_track.csv'))
```

[93]:
```
print(f"Liczba sesji odtworzeń utworów: {len(df['track_id'])}")
```

```
Liczba sesji odtworzeń utworów: 13741737
```

1.5 Odtworzenia artystów

[94]:
```
df_sessions = pd.read_csv(os.path.join(data_path, 'sessions_timestamp_track.
↪csv'))
df_tracks = pd.read_csv(os.path.join(data_path, 'tracks_ids.csv'))
df_tracks.rename(columns={'id': 'track_id'}, inplace=True)
df_session_artist = df_sessions.merge(df_tracks, on='track_id')
df_session_artist = df_session_artist[['timestamp', 'id_artist']]
df_artists = pd.read_csv(os.path.join(data_path, 'artists_ids.csv'))
```

[95]:
```
session_artists_len, artists_len = len(df_session_artist['id_artist'].
↪unique()), len(df_artists['id'].unique())
```

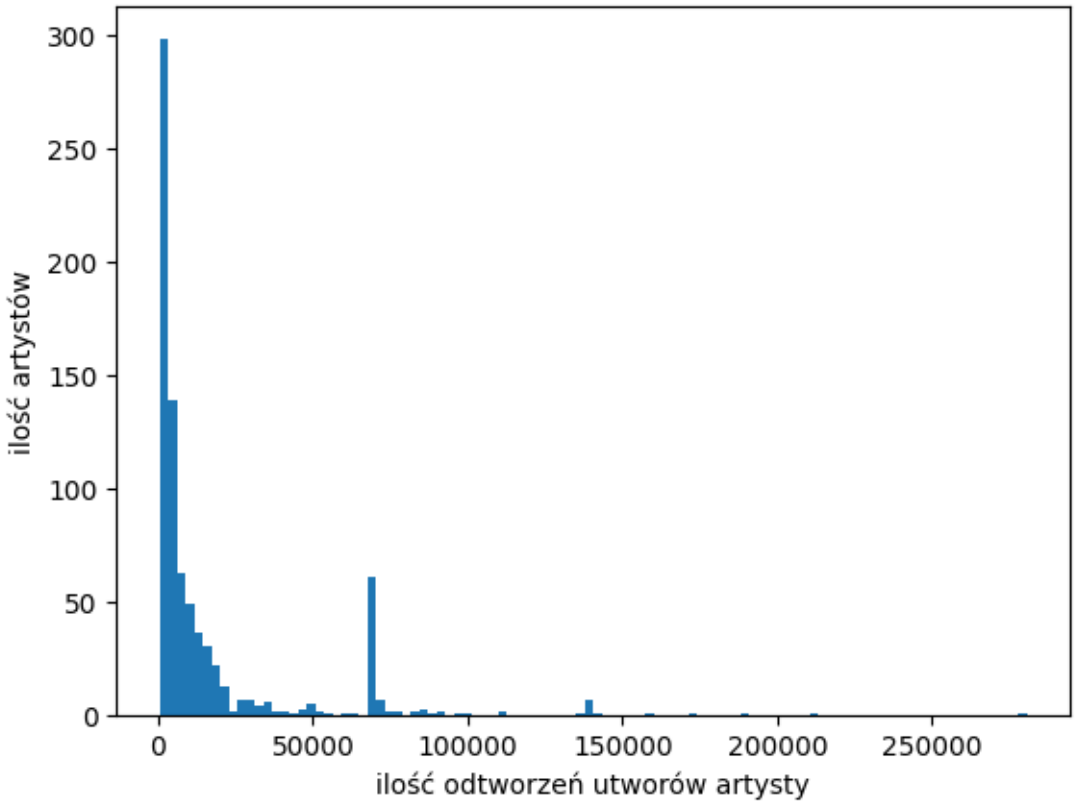
[96]:
```
print(f"Liczba artystów: {artists_len}")
print(f"Liczba odtwarzanych artystów: {session_artists_len}")
```

```
Liczba artystów: 27650
Liczba odtwarzanych artystów: 794
```

1.5.1 Wykres odtworzeń artystów

[97]:
```
session_counts = df_session_artist['id_artist'].value_counts()
df_session_counts = pd.DataFrame(session_counts)
session_counts_list = df_session_counts['count'].tolist()
```

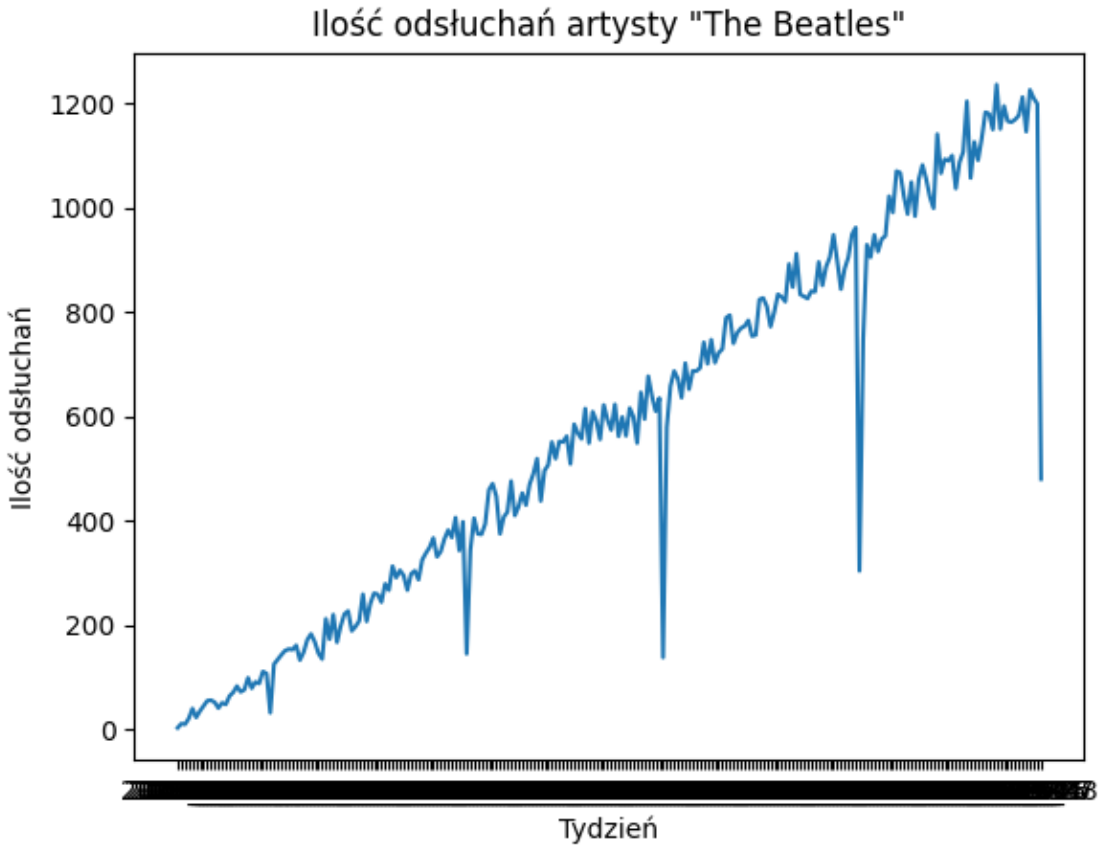
```
plt.hist(session_counts_list, bins=100)
plt.xlabel('ilość odtworzeń utworów artysty')
plt.ylabel('ilość artystów')
plt.show()
```



1.5.2 Wykres odtworzeń artystów (tygodniowy)

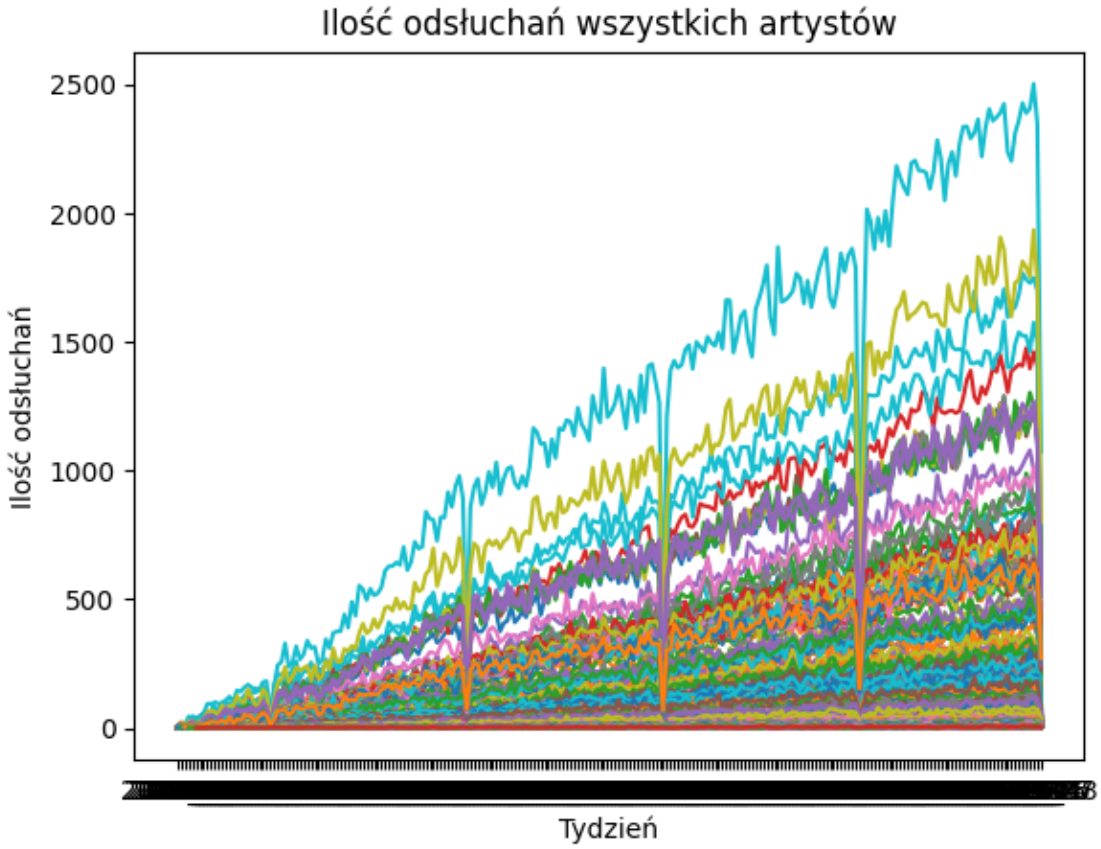
[98]:
```
df_session_artist['timestamp'] = pd.to_datetime(df_session_artist['timestamp'])

def get_sessions_per_week(artist_id: str, df: pd.DataFrame) -> pd.DataFrame:
    df_artist = df[df['id_artist'] == artist_id]
    df_artist['week'] = df_artist['timestamp'].dt.dayofyear // 7
    df_artist["week"] = df_artist["week"].astype(str).str.zfill(2)
    df_artist['year'] = df_artist['timestamp'].dt.year
    df_artist['year'] = df_artist['year'].astype(str)
    df_artist['year_week'] = df_artist['year'] + '_' + df_artist['week']
    df_artist = df_artist[['year_week', 'id_artist']]
    df_artist = df_artist.groupby('year_week').count()
    # sort by year_week desc
    df_artist.sort_index(ascending=False, inplace=True)
    # reverse index
```

```
    df_artist = df_artist.iloc[::-1]
    # df_artist.reset_index(inplace=True)
    df_artist.rename(columns={'id_artist': 'count'}, inplace=True)
    return df_artist

artist_weeks = get_sessions_per_week('3WrFJ7ztbogyGnTHbHJFl2',␣
 ↪df_session_artist)

plt.plot(artist_weeks['count'])
plt.title("Ilość odsłuchań artysty \"The Beatles\"")
plt.ylabel('Ilość odsłuchań')
plt.xlabel('Tydzień')
plt.show()
```



[99]:
```
all_artists = df_session_artist['id_artist'].unique()
plays_per_week = [get_sessions_per_week(artist_id, df_session_artist) for␣
 ↪artist_id in all_artists]

for artist_weeks in plays_per_week:
    plt.plot(artist_weeks['count'])
```

```
plt.title("Ilość odsłuchań wszystkich artystów")
plt.ylabel('Ilość odsłuchań')
plt.xlabel('Tydzień')
plt.show()
```



2 Wnioski

2.1 Ogólne

- jedynie 794 artystów z 276500 jest odsłuchiwanych przez użytkowników
- większość artystów jest odsłuchiwana sporadycznie
- odsłuchiwanie dużej liczby artystów w małej ilości, a niewielkiej liczby artystów często jest zgodne z dystrybucją Pareto, która opisuje analogiczne zjawiska do słuchania muzyki w serwisie streamingowym np. ilość przegranych godzin w grach komputerowych czy efektywność pracowników

2.2 Problemy z danymi

- wykryto anomalię objawiającą się, że niektórzy artyści nie zostali ani razu odsłuchani

- wykryty anomalię w rozkładzie polegającą na tym, że część artystów była słuchana podobną ilość razy (środkowy słupek w wykresie pierwszym)
- obie anomalie mogą być spowodowane zbyt małą ilością użytkowników w paczce danych - uczestkownicy serwisów streamingowych mają zazwyczaj tendendcję do słuchania ograniczonej i wąskiej liczby artystów, dlatego modelowanie powinno odbywać się na większej ich próbce.

2.3 Założenia

- na podstawie zgromadzonych danych rozsądnych okresem przewidywania byłby tydzień
- w zależności od otrzymanych nowych danych będzie możliwe pełne podjęcię decyzji w sprawie modelowania czasu odsłuchań w całym serwisie czy modelowania czasu odsłuchań każdego artysty z osobna.