



› PROJEKT C) SPOTIFY

„WELCHE MUSIKTRENDS SIND IN BETRACHT DER ERFOLGSFAKTOREN DER BELIEBTESTEN KÜNSTLER DER JAHRE 2017 BIS 2020 AUF SPOTIFY ZU ERWARTEN?“

BETRIEBSWIRTSCHAFTLICHE DATENANALYSE (282135)

Avram Ivanovic, David Pehar, Suphi Pembe / Fakultät WV / WIN | WiSe 2021/2022

EINLEITUNG

- meistgenutzter Streamingdienst für Musik weltweit
- täglich privat im Gebrauch
- einer der Autoren Teil von zwei aktiv verfolgten Musikprojekten
- Die beliebtesten Künstler werden miteinander verglichen
- Erfolgsfaktoren bestimmen
- Prognosen treffen

THEORETISCHE GRUNDLAGEN

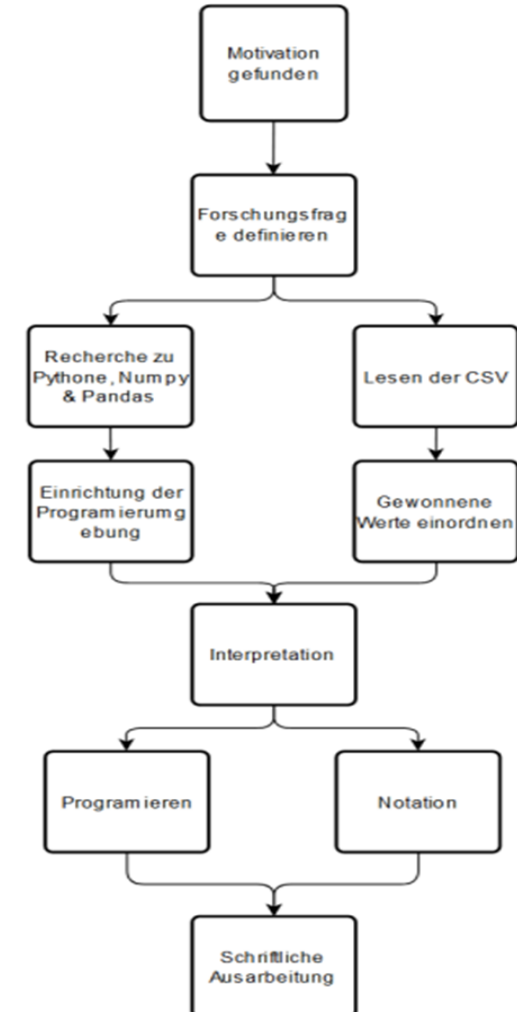
- Spotify sammelt Daten über unsere Musikvorlieben
- Daten werden verwendet für optimiertes Benutzererlebnis
- Auch verwendet zu Leistungsanalyse
- Enthalten Informationen wie z.B.:
 - Länge der Songs
 - Explizität
 - Genre
 - Vermittelte Emotion

FORSCHUNGSDESIGN

AUFBAU METHODISCHER ANSATZ

Gewählte Parameter zur Untersuchung:

- Häufigkeit des Vorkommens der Künstler in allen Jahren
- Genre der Songs
- Songlänge
- Häufigste Wörter in Titeln
- "Explicit"-Status
- "Danceability"
- "Loudness"
- "Speechiness"
- "Acoustic"
- "Geschwindigkeit"



FORSCHUNGSDESIGN

VERWENDETE LIBRARYS

- Genutze Librarys



FORSCHUNGSDESIGN

DATENGEWINNUNG

- Import der CSV Dateien
- Datensäuberung (NaNs entfernen)

```
# entferne alle NaNs  
df_k1 = df_k1.dropna(axis=0)
```

FORSCHUNGSDESIGN

DATENGEWINNUNG

- Top 20 Künstler
 - Chartzeitraum 2017 bis 2020
 - Globale Charts

```
#2017
df2017 = df1
df2017['date'] = pd.to_datetime(df2017['date'])
start_date = '01/01/2017'
end_date = '01/01/2018'
mask17 = (df2017['date'] > start_date) & (df2017['date'] <= end_date) & (df2017['country'] == "Global")
```

FORSCHUNGSDESIGN

DATENGEWINNUNG

- Top 20 Künstler
 - Filter anwednung

```
df_k4 = df2017.loc[mask17]  
df_k4_countCountry = df_k4.groupby(["artist"]).count()["count"]
```


FORSCHUNGSDESIGN

DATENGEWINNUNG

- Top 20 Künstler
 - Daten auswerten und Grafik erstellen

```
#2017
# Entferne alle NaNs
df_k4 = df_k4.dropna(axis=0)

#wähle die top 20 aus
df_k4h = df_k4.head(20)

# Visualisierungsparameter
Lsize17 = df_k4h.groupby(["artist"]).count()["count"]
LLables17 = df_k4h.groupby(["artist"]).count()["count"].index

# Figur
plt.figure(figsize=(10, 10))
plt.pie(Lsize17, labels=LLables17, autopct="%1.1f%%")
plt.tight_layout()
plt.title("")
plt.savefig("PieSpot2017.png")
```

FORSCHUNGSDESIGN

DATENGEGWINNUNG

- Durchschnittliche Songlänge

```
#Fügt alle Künstler vom Dataframe1 zu einer Liste hinzu
list = df_k1h["artist"].tolist()

#generiert den Filter mithilfe der list
filter = df_k1["artist"].isin(list)
df_k1f = df_k1.loc[filter]
#entfernt Duplikate in df1
df_k1f.drop_duplicates(subset="track", inplace=True, ignore_index=True)

#fügt alle uri von df1 zu einer Liste hinzu
df1_urilist = df_k1f["uri"].tolist()

#erstellt filter und filter direkt df2 nach den passenden Tracks
duration_filter = df2["Uri"].isin(df1_urilist)
df2f = df2.loc[duration_filter]

#entfernt Duplikate in df2
df2f.drop_duplicates(subset="Uri", inplace=True, ignore_index=True)

#gibt Durchschnittslänge als String aus
length_string = df2f["duration_ms"].mean()
print("Länge: " + str(length_string))
```

FORSCHUNGSDESIGN

DATENGEWINNUNG

- Wörter im Titel vergleichen und darstellen

```
#filter und fügt den text als variable hinzu
df_text = df_k1["title"]

#entfernt Duplicate
df_k1.drop_duplicates(subset="title", inplace=True, ignore_index=True)

#zählt die am häufigst genutzten wörter und fügt sie zu einer .CSV Datei hinzu
df_wordtitle = pd.Series(' '.join(df_k1['title']).lower().split()).value_counts()[: 142].head(20)
df_wordtitle.to_csv("WordTitle.csv")
```

FORSCHUNGSDESIGN

DATENGEGWINNUNG

- Genre der Tracks der beliebtesten 20 Künstler grafisch darstellen

```
# Visualisierung von Genre
# Hier werden die Genre aus df2 gezählt sowie auch ihre Namen als Index gespeichert
LsizeGenre = df2f.groupby(["Genre"]).count()["count"]
LLablesGenre = df2f.groupby(["Genre"]).count()["count"].index

# kuchendiagramm wird erstellt und gespeichert
plt.figure(figsize=(10, 10))
plt.tight_layout()
plt.pie(LsizeGenre, labels=LLablesGenre, autopct="%1.1f%%")
plt.title("")
plt.savefig("Genre.png")
```

FORSCHUNGSDESIGN

DATENGEWINNUNG

- Explicit der Tracks vergleichen und grafisch darstellen

```
# Visualisierung von Explicit
# Hier werden die Explicit aus df2 gezählt sowie auch ihre Namen als Index gespeichert
LsizeGenre = df2f.groupby(["Explicit"]).count()["count"]
LLablesGenre = df2f.groupby(["Explicit"]).count()["count"].index

# Figure
plt.figure(figsize=(10, 10))
plt.pie(LsizeGenre, labels=LLablesGenre, autopct="%1.1f%%")
plt.tight_layout()
plt.title("")
plt.savefig("Explicit.png")
```

FORSCHUNGSDESIGN

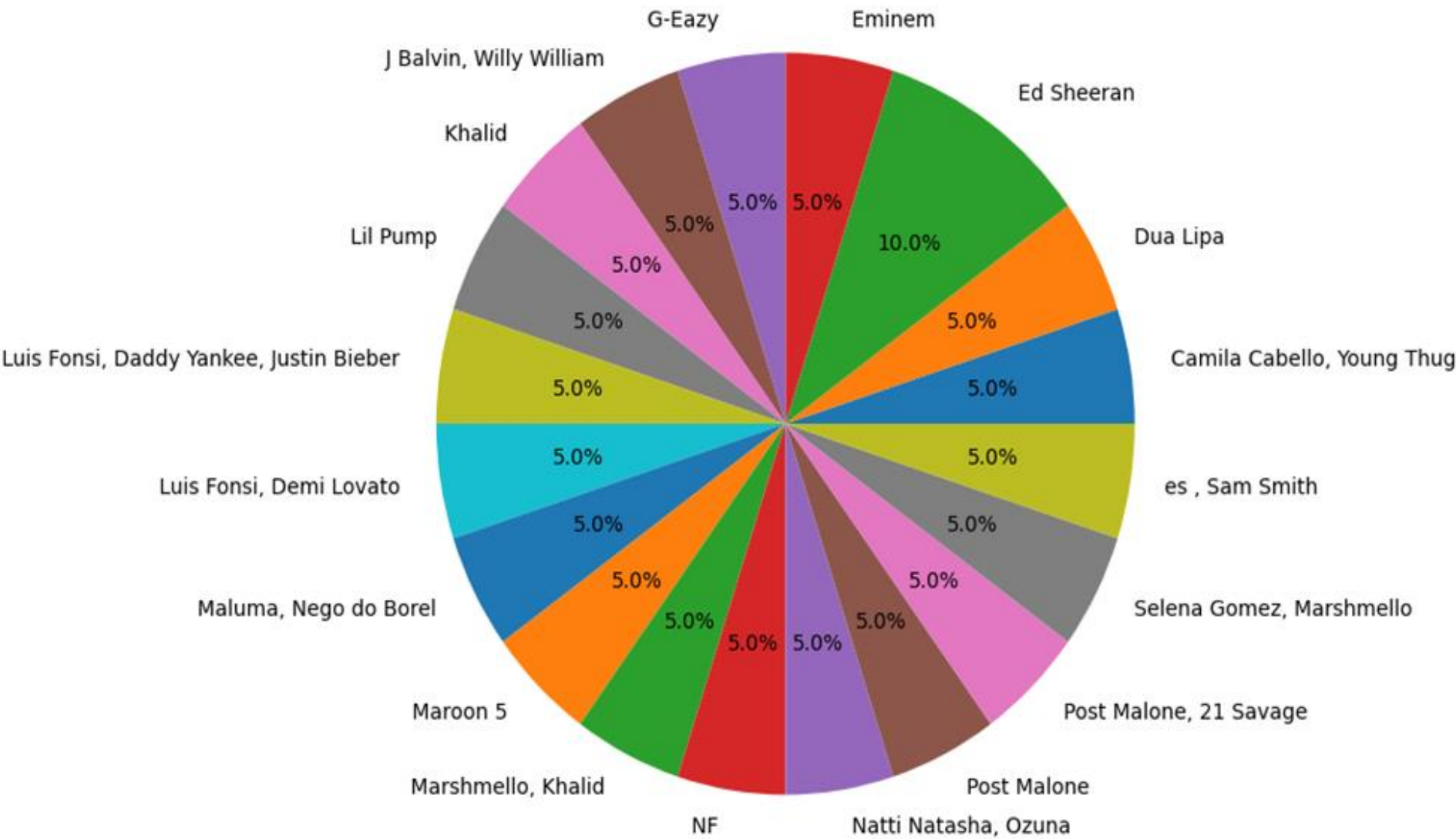
DATENGEWINNUNG

- Mittelwert der Danceability, Loudness, Speechiness, Acoustics & Tempo

```
#berechnet den durchschnitt der deancability  
d2f2dance_string = df2f["danceability"].mean()  
print("Danceability: " + str(d2f2dance_string))
```

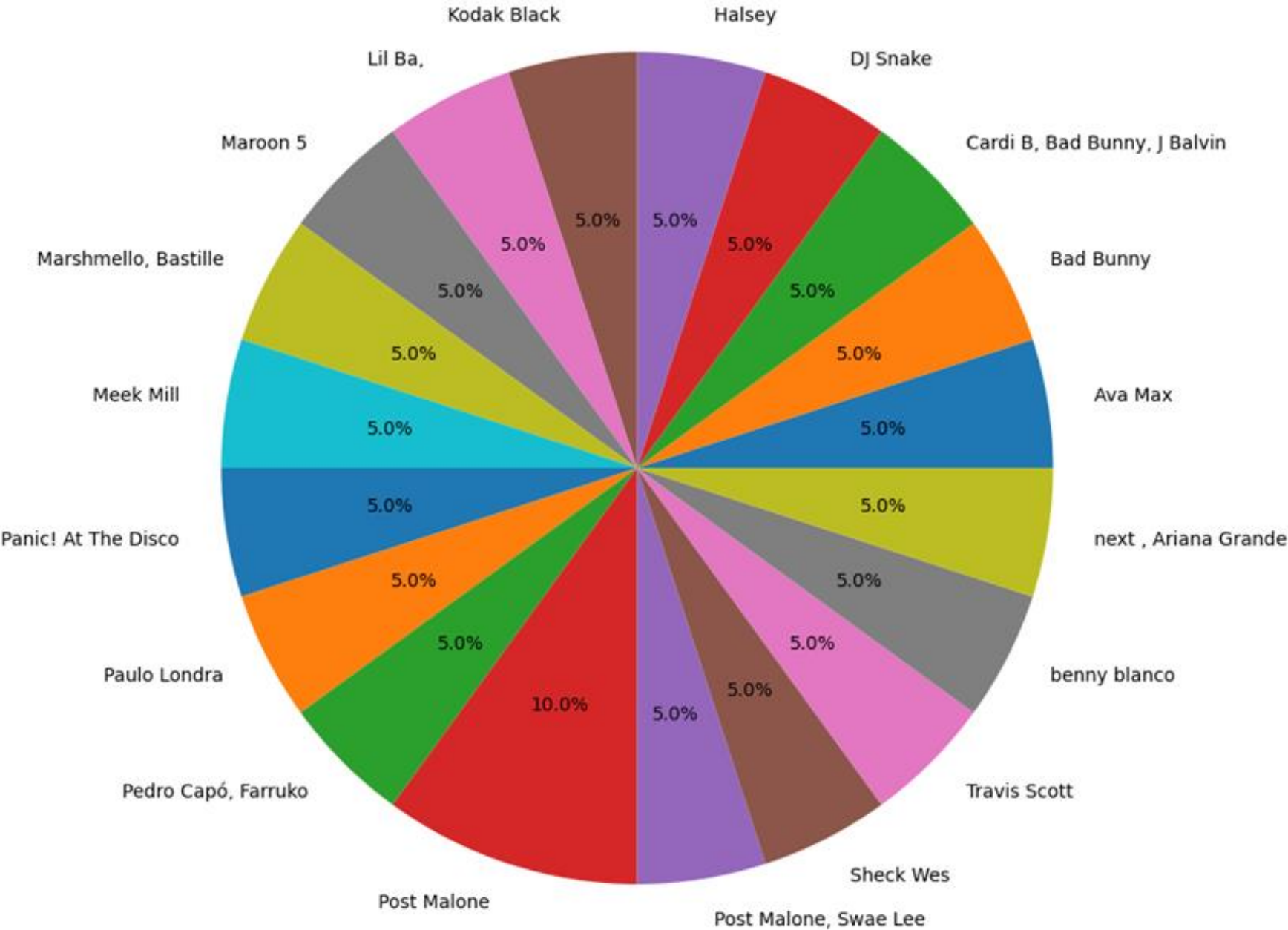
ERGEBNISSE

Top 20 Künstler 2017



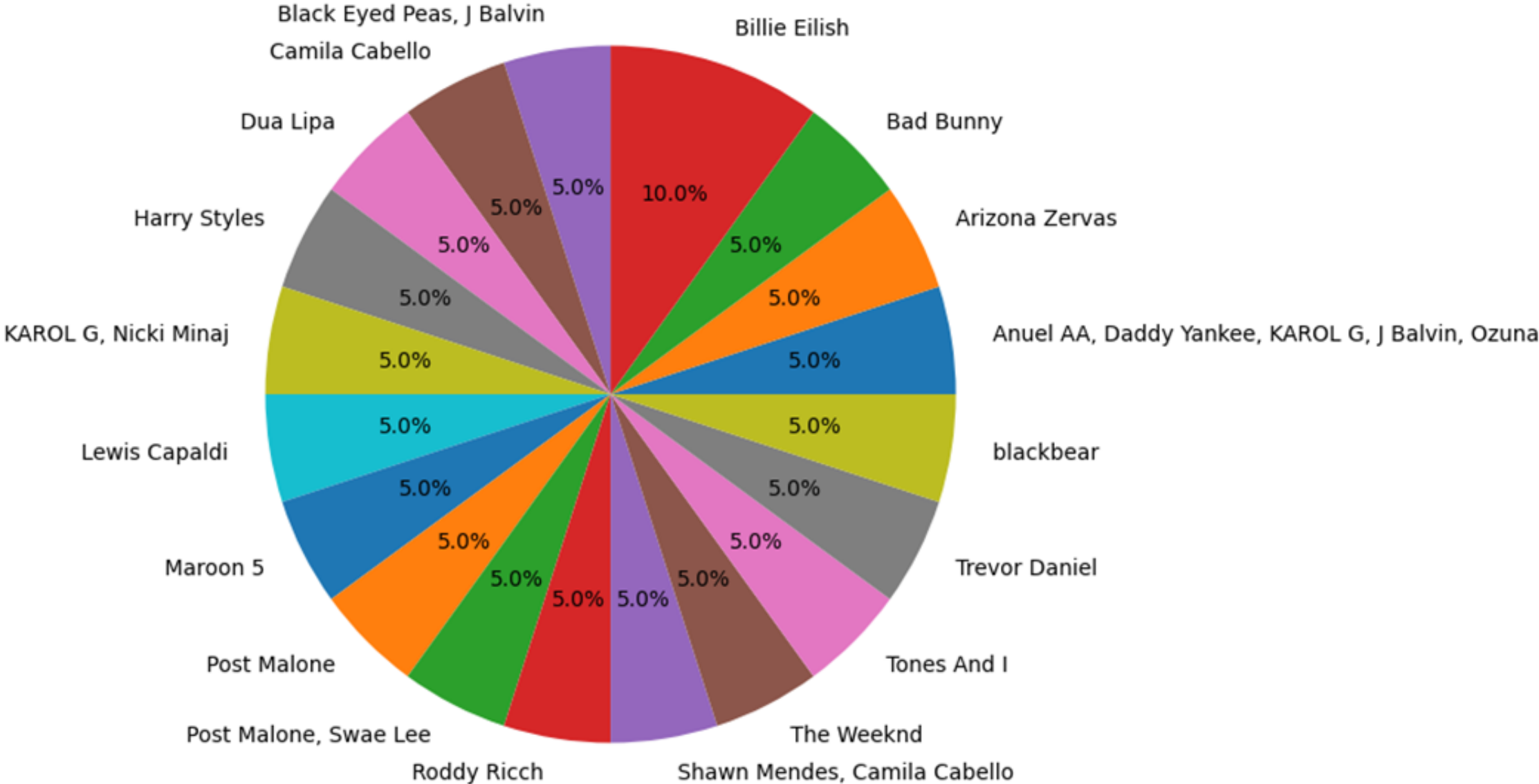
ERGEBNISSE

Top 20 Künstler 2018



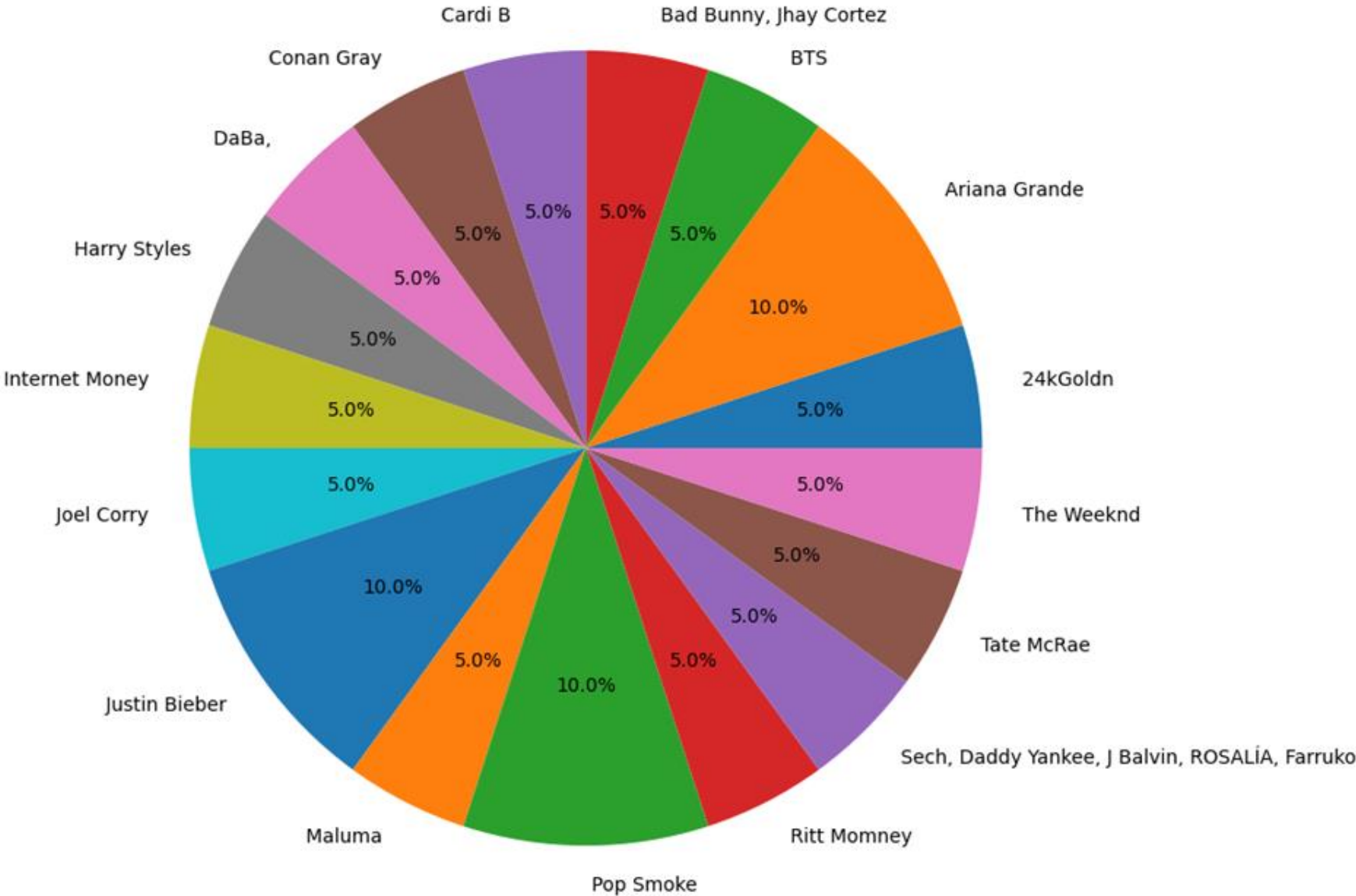
ERGEBNISSE

2019



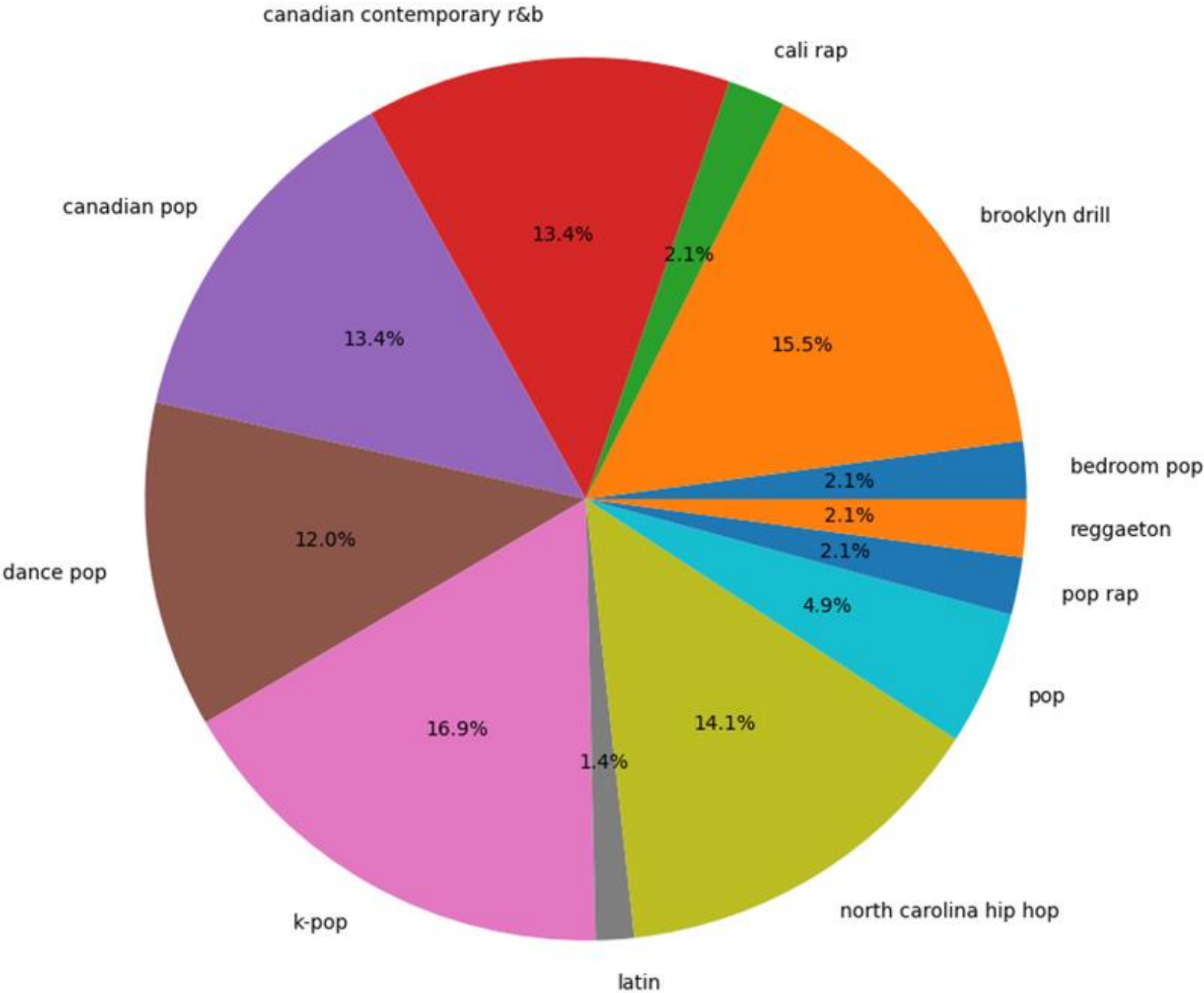
ERGEBNISSE

2020



ERGEBNISSE

Beliebteste Genres 2020



ERGEBNISSE

- Durchschnittliche Songlänge beträgt 3min und 12s
- Häufigste Wörter in Titeln: „the“ 75 mal, „me“ 63 mal, „you“ 59 mal, „remix“ 40 mal und „my“ 38 mal
- 52,2% der Songs sind "Explicit"
- Durchschnittliche "Danceability": 0,67
- Durchschnittliche "Loudness": -6,3 LUFS
- Durchschnittliche "Speechiness": 0,145
- Durchschnittlicher "Acoustic"-Wert: 0,2
- Durchschnittliche Geschwindigkeit: 125 BPM

LIMITATION

- Eigene Programmierfähigkeit
- Umfang des Datensatzes
- Wahl der Forschungsfrage
- Wahl der zu untersuchenden Parameter/Größen

FAZIT

- Neigung zu elektronischem Sound und Tanzbarkeit
- Neigung zu lauten aber kurzen Songs
- Parameter der Künstler deuten auf Erfolgsfaktoren hin

Prognose: Aktueller Trend wird bleiben, bzw sich in eine mögliche Richtung steigern

ABBILDUNGSVERZEICHNIS

Titelfolie Abbildung 1

<https://pixabay.com/de/illustrations/spotify-streaming-musik-1360002/>

Code Abbildung & Charts
eigene Grafik

LITERATUR

Dörn, Sebastian (2020): Python lernen in abgeschlossenen Lerneinheiten. Programmieren für Einsteiger mit vielen Beispielen. Wiesbaden, Heidelberg: Springer Vieweg (Lehrbuch).

Gomes, Inês; Pereira, Inês; Soares, Inês; Antunes, Mariana; Au-Yong-Oliveira, Manuel (2021): Keeping the Beat on: A Case Study of Spotify. In: Álvaro Rocha, Hojjat Adeli, Gintautas Dzemyda, Fernando Moreira und Ana Maria Ramalho Correia (Hg.): Trends and Applications in Information Systems and Technologies, Bd. 1366. Cham: Springer International Publishing (Advances in Intelligent Systems and Computing), S. 337–352.

Rocha, Álvaro; Adeli, Hojjat; Dzemyda, Gintautas; Moreira, Fernando; Ramalho Correia, Ana Maria (Hg.) (2021): Trends and Applications in Information Systems and Technologies. Cham: Springer International Publishing (Advances in Intelligent Systems and Computing).