# Spotify Song Recommender

Bena Smith

This program uses 400,000 playlists from the Spotify Million Playlist dataset https://www.aicrowd.com/challenges/spotify-million-playlist-dataset-challenge

The program uses Word2Vec to find songs similar to a searched song. This can be used to suggest similar songs to a playlist that a user creates

In [1]:

```
# Import packages

import pandas as pd
from itertools import combinations
import numpy as np
from sklearn.model_selection import train_test_split
from gensim.models import Word2Vec
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
```

In [2]:

```
# Read datafile (400,000 playlists from https://www.aicrowd.com/challenges/s
df = pd.read_parquet('playlists.parquet')
```

In [3]:

```
# test/ train split. Testing has not currently been developed
df_train, df_test = train_test_split(df, test_size=0.3, random_state=42)
```

In [4]:

```
# Reset indices
df_train = df_train.reset_index(drop=True)

# make a dictionary of track ids to song name and artist
all_tracks = set()
track_uri_dict = dict()

for playlist in df_train["tracks"]:
    for track in playlist:
        all_tracks.add(track["track_uri"])
        track_uri_dict[track["track_uri"]] = (track["track_name"], track["ar
```

## Train Model

In [5]:

```
# Create "sentences" for word2vec
playlist_list = []

for playlist in df_train["tracks"]:
    track_list = []
    for track in playlist:
        track_list.append(track["track_uri"])
    playlist_list.append(track_list)
```

In [6]:

```
# Train Word2Vec model
model = Word2Vec(sentences=playlist_list, vector_size=100, window=5, min_cou
```

## Visualize Vector embeddings

In [51]:

```
# Select track_uri values to visualize
chosen_track_uris = [
    "spotify:track:7yCqehT9dKACTFy7YugP0J", "spotify:track:1Bqxj0aH5KewYHKUg
    "spotify:track:0S3gpZzlT9Hb7CCSV2owX7", "spotify:track:7xyauCIams30DLpAp
    "spotify:track:6cJLfIqwh0tCKRjYM3WpZ5", "spotify:track:0DQd0tWurMHUAv0cM
    "spotify:track:2HJQcyUpmUuvzS5vBAICIc", "spotify:track:5FrxJwJyC8Wy9rgj0
    "spotify:track:6Hu6dzwlvoyg3zBUC8k4BK", "spotify:track:0pKGbgGzZ4yLydQvc
    "spotify:track:1EaKU4dMbesXXd3BrLCtYG", "spotify:track:2771LMNxwf62FTAdp
    "spotify:track:4cA6e3U7mtJfAOq3dWNAUZ"]
```

In [52]:

```
print("Songs to plot: \n")

for track in chosen_track_uris:
    print(track_uri_dict[track])
```

```
Songs to plot:

('You & Me', 'Marc E. Bassy')
('Some Kind Of Drug', 'G-Eazy')
('Morning', 'Marc E. Bassy')
("Mama, I'm Coming Home", 'Ozzy Osbourne')
('Not Going Away', 'Ozzy Osbourne')
('With You Tonight (Hasta El Amanecer)', 'Nicky Jam')
('Darte un Beso', 'Prince Royce')
('Stand by Me', 'Prince Royce')
('Roses', 'The Chainsmokers')
('Working For It - Bonus Track', 'ZHU')
('Faded - Original Mix', 'ZHU')
('Never Be The Same', 'Christopher Cross')
('Sailing', 'Christopher Cross')
('Pour Some Sugar on Me', 'Def Leppard')
('Soy Peor', 'Bad Bunny')
('ELEMENT.', 'Kendrick Lamar')
('Bodak Yellow', 'Cardi B')
('In My Feelings', 'Kehlani')
```

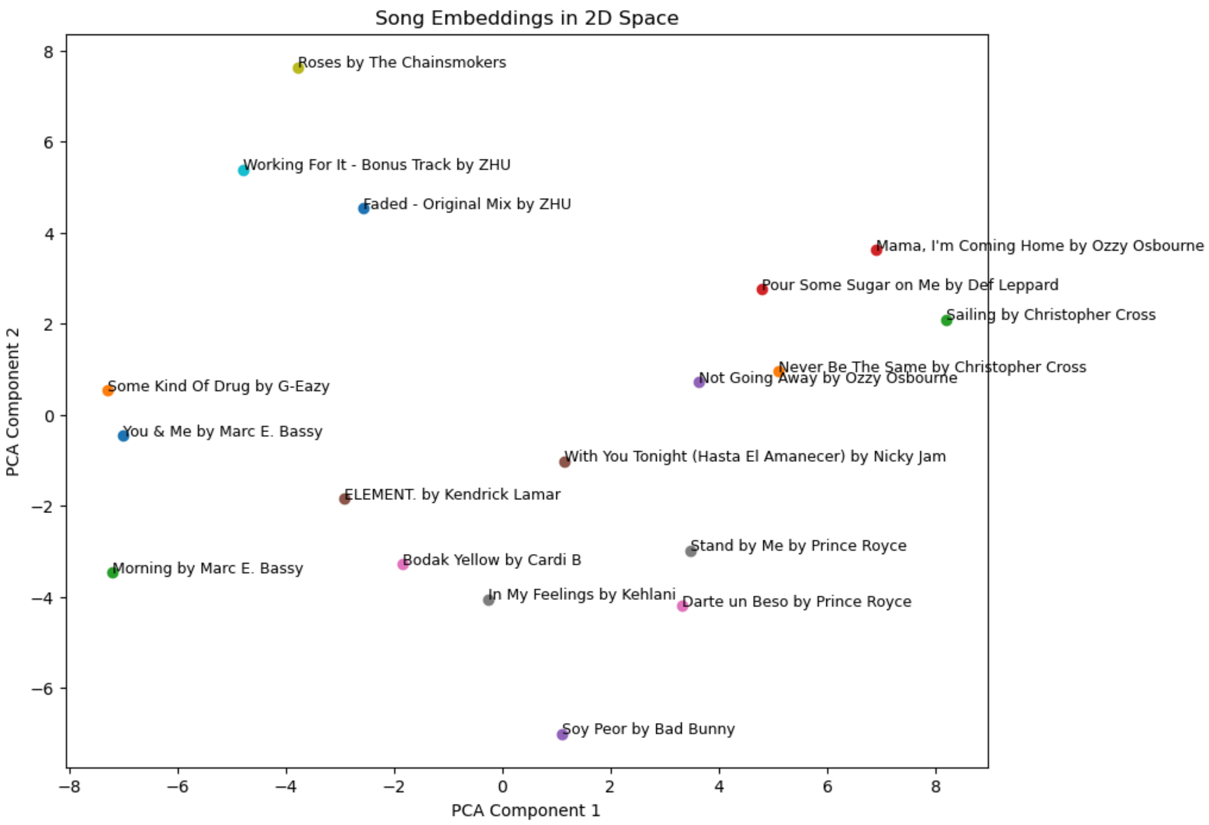
In [53]:

```
# Get the embeddings for the chosen tracks
embeddings = [model.wv[uri] for uri in chosen_track_uris]

# Reduce dimensionality to 2D for visualization
pca = PCA(n_components=2)
reduced_embeddings = pca.fit_transform(embeddings)

# Plot the reduced embeddings
plt.figure(figsize=(10, 8))
for i, (x, y) in enumerate(reduced_embeddings):
    track_name, artist_name = track_uri_dict[chosen_track_uris[i]]
    label = f"{track_name} by {artist_name}"
    plt.scatter(x, y, label=label)
    plt.text(x , y , label, fontsize=9)

plt.title("Song Embeddings in 2D Space")
plt.xlabel("PCA Component 1")
plt.ylabel("PCA Component 2")
plt.show()
```



We can see similar songs close together in this graph. Songs by the same artists are close together. For example, Christopher Cross tracks are close together, as are Ozzy Osbourne, Prince Royce and ZHU tracks. Songs in Spanish are clustered together, as are EDM songs. Rap and R&B tracks are also close together. I remember listening to Marc E. Bassy and G-Eazy songs together when I was in high school because they have a few songs together. They are also plotted close together. Note that the proximities shown in this plot are not exactly to scale with similarity because we are only looking at the first 2 Principal Components. Songs may be further or closer in reality in the higher dimensional space.

# Search Similar Songs

In [54]:

```
# Dictionary of Song titles/ artist to track ids
reversed_track_uri_dict = {value: key for key, value in track_uri_dict.items
```

In [11]:

```
# Function to get similar songs
def search(song_title):
    searched_song = reversed_track_uri_dict[song_title]
    #searched_song = "spotify:track:7qiZfU4dY1lWllzX7mPBI3" #"spotify:track:

    most_similar_songs = model.wv.most_similar(searched_song, topn=10)


    print("Searching for similar songs to:" + str(track_uri_dict[searched_sc
    for track in most_similar_songs:
        print(track_uri_dict[track[0]])
```

```
In [12]: search(('Shape of You (Major Lazer Remix) [feat. Nyla & Kranium]', 'Ed Sheer
```

```
Searching for similar songs to:('Shape of You (Major Lazer Remix) [feat. Nyl
a & Kranium]', 'Ed Sheeran')

('Shape of You - Galantis Remix', 'Ed Sheeran')
("You Don't Know Me - Radio Edit", 'Jax Jones')
('The Heat (I Wanna Dance With Somebody)', 'Ralph Felix')
('I Love You', 'Axwell /\\ Ingrosso')
('Swalla (feat. Nicki Minaj & Ty Dolla $ign) - Wideboys Remix', 'Jason Derul
o')
('Show You Love', 'Kato')
('Feels Right - Galloway Remix', 'Jocelyn Alice')
("It Ain't Me - Tiësto's AFTR:HRS Remix", 'Kygo')
('No Lie - Sam Feldt Remix', 'Sean Paul')
('On My Way', 'Tiësto')
```

After searching for similar songs to Ed Sheeran's song that is remixed by Major Lazer, we get other remixed pop songs including another remix of Shape of You.

```
In [13]: search(('Best I Ever Had', 'Drake'))
```

```
Searching for similar songs to:('Best I Ever Had', 'Drake')

('Find Your Love', 'Drake')
('Headlines', 'Drake')
('Make Me Proud', 'Drake')
('Take Care', 'Drake')
('Practice', 'Drake')
('Marvins Room', 'Drake')
('Over', 'Drake')
('BedRock', 'Young Money')
('Shot For Me', 'Drake')
('Come Thru', 'Drake')
```

After searching for similar songs to Drake's song, we get other Drake songs and other Rap and R&B tracks.

In order to suggest songs for a playlist, we can select random songs within the created playlist and find similar songs using this algorithm