k-NN Machine Learning Model in app.py

After loading the data, we can start building our machine learning model to recommend songs. There are many ways to approach this. A simple approach is to use a k Nearest Neighbor model to get the top songs that are closest in distance with the set of feature inputs selected by the user. These “feature inputs” include the genre of interest, release year range (start year and end year), and a set of audio features (acousticness, danceability, energy, instrumentalness, valence, tempo).

We can use Sklearn to build the k-NN model and return the top-k results given a test point. To perform the above functions, we write the function `n_neighbors_uri_audio` , which will return the Spotify URIs and audio feature values of the top neighbors in ascending order of their rank (a point closest to the input features is ranked first).

```
def n_neighbors_uri_audio(genre, start_year, end_year, test_feat):
   genre = genre.lower()
   genre_data = exploded_track_df[(exploded_track_df["genres"]==genre) &
(exploded_track_df["release_year"]>=start_year) &
(exploded_track_df["release_year"]<=end_year)]
   genre_data = genre_data.sort_values(by='popularity', ascending=False)[:500]    neigh =
NearestNeighbors()
   neigh.fit(genre_data[audio_feats].to_numpy())    n_neighbors = neigh.kneighbors([test_feat],
n_neighbors=len(genre_data), return_distance=False)[0]    uris =
genre_data.iloc[n_neighbors]["uri"].tolist()
   audios = genre_data.iloc[n_neighbors][audio_feats].to_numpy()
   return uris, audios
```

Note that we only use the top 500 most popular songs belonging to a certain genre so that the songs recommended by our system will be more popular. I

play around with this number, remove it to being recommended less popular tracks.