Candidate Generators Assignment Summary

Jonathan Shang-Wen Chang (jsc2279): Two Tower generator, content embeddings & similarities comparison.
I did content embeddings by selecting top artist styles, top sources and top seeds to identify the top contents and then generated embeddings accordingly. The output is *user_features_tensor* in the *two_tower.py*.
I developed code to implement the similarity comparison of user and content embeddings. I also implemented the code to get the two tower model connected to the application and generate results. The involved files are *TwoTowerANNGenerator.py* and *BetaController.py*.

Nicole Yu (ny2334): Two Tower generator, user embeddings.
I did user embeddings by combining the user features in professor's code and our own-designed user features and then generated embeddings accordingly. The new features interact with content features. The output is *user_features_tensor* in the *two_tower.py*.
Our own-designed features are the following conditional probabilities:

- **"indie" score:** P(a user will like a content | this content is an unfavorable contents)
- **"vigilante" score:** P(a user will dislike a content | this content is a popular contents)
- **"basic" scores:** P(a user will like | this content is a popular contents), P(a user will dislike | this content is an unfavorable contents)

● *The idea of the above scores is generated by Jonathon. I am responsible for implementing and incorporating the code.*

Jingru Chen (jc5898): User-based collaborative filtering.
I implemented collaborative filtering with the user-item matrix, see code file services/backend/src/data_structures/user_based_recommender/beta/UserBasedRecommender.py and services/backend/src/reccomendation_system/recommendation_flow/candidate_generators/beta/CollabertiveFilteredSimilarUsersGenerator.py

Rohan Sheelvant (rns2167): Our choice generator.
I worked on the your choice generator. For this generator, I implemented the popularity-based generator. The logic behind the popularity generator is to retrieve all items having highest number of likes. I implemented this using SQL command. Additionally, I also helped Jingru decode errors in Collaborative based filtering code and

integrate her work to deploy the generator on docker. I was responsible for creating the submission repo and creating a PR.