Content-Based Filtering:

- Mood Mapping: Match song attributes to movie moods or genres. For example:
  - High tempo & high energy: Action or thriller movies
  - High acousticness & low energy: Drama or romance movies
  - High danceability: Musical or comedy movies
- The challenge is in creating a mapping from Spotify attributes to movie genres/tones. We have to define this manually at first and then refine based on user feedback.
  - This is what the mapping document is for

Collaborative Filtering:

- This is based on user behavior. If many users who liked a particular song or type of music also liked a particular movie, then recommend that movie.
- The challenge is collecting enough user data for meaningful correlations.
  - Probably not a good option for us now since we won't have that many users

Hybrid Method:

- Combine content-based and collaborative filtering. Use content-based recommendations initially, and as you gather more user data, transition or combine it with collaborative filtering.
- We may want to add a feature where users can rate their recommendation that our algorithm gave them. We can use this data to further refine our algorithm - it can recognize recommendations that were perceived as bad and those that were perceived as good to aid it in making decisions in the future.

Matrix Factorization:

- This involves decomposing the user-item interaction matrix into multiple matrices representing latent factors. Techniques like Singular Value Decomposition (SVD) can be used.
- For this, we can start by building a user-song matrix (users who liked a song) and a user-movie matrix (users who liked a movie). By comparing these matrices, you can infer movie preferences.
  - We don't have access to the data of what movie users like, so it won't work.

Deep Learning:

- With TensorFlow, we can use deep learning models to understand the complex relationships between musical attributes and movie preferences.
- We can develop a neural network where the input layer consists of Spotify attributes and the output layer suggests possible movie genres.
  - If there is a public dataset or recommendation algorithms for this, it'd be perfect, but we couldn't find any.

Association Rule Mining:

- Discover associations between songs or musical attributes and movies. For instance, if a significant number of users who liked songs with high danceability also liked musical movies, then this forms a rule.
- This can be implemented using algorithms like Apriori or FP-growth.
    - Again, we don’t really know what movies users have liked in the past.

Clustering:

- Use unsupervised learning algorithms like K-Means to cluster users based on their Spotify attributes.
- If a user falls into a particular cluster, recommend movies popular or rated highly in that cluster.

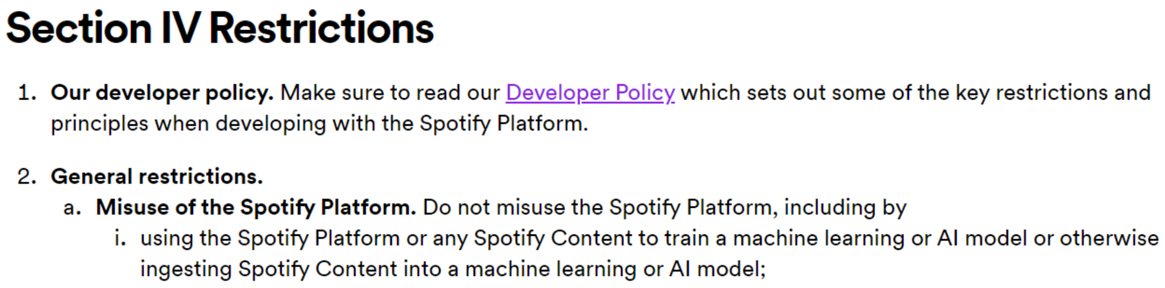
Reinforcement Learning:

- Implement a model where the system learns the best movies to recommend over time by receiving feedback (rewards/punishments) based on user interactions.
- For instance, if a user regenerates recommendations frequently, it could be a signal that the recommendations are not satisfactory.
    - This would take a long time to start working well, it can be a future goal

For the scope of this project, it might be optimal to start out with content-based filtering, and as we get more user data, we can start out with a simple feed-forward neural network where input layer has song features and output layer has probability of liking different movie genres. Over time, the model can be refined. We can start by training with manually defined mappings and refine with user feedback.

For this semester, we could focus on getting the content-based filtering algorithm ready, and work on the deep learning aspect next semester.

****NOTE**:**



Section IV Restrictions

1. **Our developer policy.** Make sure to read our Developer Policy which sets out some of the key restrictions and principles when developing with the Spotify Platform.
2. **General restrictions.**
    a. **Misuse of the Spotify Platform.** Do not misuse the Spotify Platform, including by
        i. using the Spotify Platform or any Spotify Content to train a machine learning or AI model or otherwise ingesting Spotify Content into a machine learning or AI model;

For deep learning:

Data Collection:

- First, we’ll need a dataset containing Spotify track features linked to movies or movie genres.
- We can use the initial content-based mapping (like mapping song attributes to specific movie genres) to generate training data.

Data Preprocessing:

- Normalize song attributes between 0 and 1
- Convert movie genres into a one-hot encoded vector. If a song maps to both "Action" and "Thriller" genres, the vector could look like [1, 0, 0, 1, ...].

Model Building:

- Use a deep learning framework like TensorFlow or PyTorch.
- Define a neural network architecture:
  - Input Layer: Number of nodes equal to the number of song attributes.
  - Hidden Layers: 2-3 layers with a decreasing number of nodes. Activation functions like ReLU can be used.
  - Output Layer: Number of nodes equal to the number of movie genres, with a sigmoid activation function to get probabilities for each genre.

Model Training:

- Use the preprocessed dataset to train the model.
- Define a suitable loss function
- Split the data into training and validation sets to monitor for overfitting.
- Train the model for a certain number of epochs until validation loss stops improving.

Evaluation:

- Once the model is trained, evaluate its performance on a test set.
- Measure metrics like accuracy or F1 score to determine how well our model is performing.