HOLLY LOWE
FINAL REPORT

MUSIC RECOMMENDATION SYSTEM

Executive Summary

This project is suggesting a **hybrid model** using a **Decision Tree** to determine which models are used in order to create a trustworthy and innovative top 10 song recommendations to a user.
The models that will be used include the following, all* of which have been trained, tweaked and tested:

- **Popularity Based**
- **Content-Based**
- **Song to Song Similarity**
- **User-User Similarity**
- **SVDMatrix Estimation**
- **Content-Based (*not yet trained on new data)**

The reason for the hybrid approach is to maximise the power of these models and avoid a recommendation that decreases with accuracy over the 10 songs. Individually all the models give good top 2 or 3 recommendations but fail to give a consistently good top 10. By using **combinations of models** we are thoughtfully creating a recommendation for the user based on many different factors. I am also implementing various methods to ensure the recommendations of songs include **new and independent artists** and not just the most popular ones.

Next Steps:
In order to accommodate brand new bands and artists we will **require more data** that was not made available to me for this period. I would suggest a fast beta roll-out for the NLTK to analyse song reviews of new songs that look for comparable artists within the review. This can be used to train the Content Based Model.

I also suggest that instead of cutting off songs with more than 5 play counts, we simply cap them so they can be included in the dataset. This way we do not punish songs for being popular with a user.

Given the further action required I am suggesting a **further 4-6 months of data collection, model building and Decision Tree structuring**.

Conclusion:
This brand new master model aims to gain the **trust of the user** and keep them from leaving for another platform. It is hoped that they will promote the quality of the music discovery algorithm to their friends and that we will become the most trusted and enjoyed platform. It also gains the respect of the artists and the listeners who care about new music. Our long-vision platform can be proud to boast not only an **engaged and happy usership**, but also a platform that is **thoughtfully and ethically proactive about the music industry** by avoiding feedback loops and inspiring creativity and new music.

Problem Summary

Music is important. It can express what we are feeling when we are unable to and it makes us feel better / alive / seen / part of something. It is not a product to be consumed, but it is a necessity to our soul. Finding the music that speaks to you is so important.

The music industry has changed so much since the internet and streaming came along. It collapsed and broke for a while but it is now finding its feet again. With platforms such as ours we can use this as a tool for good - for independent artists and bands to seek out their own fans no matter where they are in the world. That is something very special indeed and it overcomes fashion, scenes and major labels who simply pay their way to the radio plays.

I want to create a song recommendation system that serves two purposes:

1) To feed the user the songs they love and will love. To keep them musically happy and to gain their trust so that they are more likely to listen to the songs in their recommendations and stay with the platform.
2) To help ensure that new and independent bands and artists have a chance to find potential new fans by placing them inside top 10 song recommendations to listeners who are likely to appreciate their style.

This project looks at various different recommendation models that make predictions about how many times a user will listen to a particular song. The measurements of success and accuracy of these models will go on to form a conclusion of which models will be used for a final recommendation system that will adhere to both of the points made above.

Solution Design

I took a dataset of user IDs, song IDs, play count (how many times a user played a particular song), and I combined it with another so we also had the song titles, album name and year of release. With this data I created and trained various models that could predict how many times a user would play a particular song. I could also use these predictions to rank the songs specific to a user and produce for them a top 10 songs recommendation.
These are the models that were created and tested:

Popularity Based - This is a simple model that looks at play counts and the number of unique listeners for each song and produces a ranked order of the most popular songs.

User-User Similarity - (Collaborative Filtering) This looks at how the user of interest has been interacting with songs and finds other similar users, and indeed similar users of the similar users. It then uses popular songs with these similar users as recommendations for this user.

Song-Song Similarity - (Collaborative Filtering) This looks at a particular song (so we can use a song with a high play count from our user) and finds other songs that are similar based on user interaction with it and other songs.

SVD (Singular Value Decomposition) This is a means of matrix estimation and finding latent variables by pulling apart a matrix and factorising it into 3 separate matrices. It is highly personalised and can be good in sparse matrices.

Clustering - This groups (or clusters) users together depending on their play count of songs. It produces recommendations based on other users in the cluster to the user of interest.

Content Based - This model takes the Song Title, Artist Name and Album Name and uses the Natural Language Toolkit to analyse the words and produces recommendations based only on this. It requires no user interaction and can also be used for brand new songs with no previous play count.

The success of these models in terms of predictions was calculated with the following methods. After experimenting with different values, I used the threshold value of >1.5. This value is to represent a user playing a song at least 1.5 times. This becomes the boundary that makes this regression problem become more like a classification problem. We are saying here that if a user plays the song 1.5 times or more, then they are considered to like the song that was recommended and it would be considered a successful prediction.

- **Root Mean Square Error (RMSE)** - measuring the distance of error of the prediction to the actual number of times a song was played. (Lower score is better)
- **Precision@K** - This measures the percentage of songs recommended being a success ie. how many true positives there were in the recommendation. (Higher score is better)
- **Recall@K** - This measures the percentage of songs that would have been a success being recommended to the user. Ie. Of all the songs a user would like - how many of them made it into the recommendation? (Higher score is better)
- **F1@K** - This is the harmonic mean of Precision and Recall. (Higher score is better)

Model	RMSE	Precision	Recall	F_1 score
sim_user_user	1.0878	39.6%	69.2%	50.4%
sim_user_user_opt	1.0521	41.3%	72.1%	52.5%
algo_knn_item	1.0394	30.7%	56.2%	39.7%
algo_knn_item_opt	1.0328	40.8%	66.5%	50.6%
svd	1.0252	41.0%	63.3%	49.8%
svd_optimized	1.0143	41.2%	63.3%	49.9%
clust_baseline	1.0487	39.7%	58.2%	47.2%
clust_tuned	1.0654	39.4%	56.6%	46.5%

Metric of Interest:

I decided to focus on **Precision** as my main metric since I want our users to trust in our recommendations and their top 10 to be all songs of interest to them. You can see from the table that the sim_user_user_opt (User - User Similarity Optimised) has the best results for almost all of the metrics including Precision. With SVD having the best RMSE score.

There were also two other models not in this table as they were unable to be measured with these metrics. They were Popularity Based and Content Based Models.

I also tested each model by asking for recommendations for a particular user on songs they have and have not interacted with. I also asked each optimised model to create a top 5 songs recommendation for a particular user. And the Content Filtering produced a Top 10 Recommendation based on the song ‘Learn To Fly’ by The Foo Fighters.

.

USER-USER SIMILARITY

	song_id	play_freq	predicted_plays	corrected_plays
0	5531	618	2.553335	2.513109
2	317	411	2.518269	2.468943
1	5943	423	2.390723	2.342101
3	4954	183	2.406776	2.332854
4	8635	155	2.396606	2.316284

```
['The Maestro by Beastie Boys',
 "You've Got The Love by Florence + The Machine",
 'Undo by Björk',
 'Secrets by OneRepublic',
 'Una Confusion by LU']
```

SONG-SONG SIMILARITY

	song_id	play_freq	predicted_plays	corrected_plays
4	2342	111	2.653903	2.558987
2	5101	130	2.386577	2.298871
3	139	119	2.313727	2.222057
1	7519	168	2.270864	2.193712
0	8099	275	2.212702	2.152399

```
['I Got Mine by The Black Keys',
 'Toxic by Britney Spears',
 'A Dustland Fairytale by The Killers',
 'White Sky by Vampire Weekend',
 'Alaska by Camera Obscura']
```

SVD MATRIX ESTIMATION

	song_id	play_freq	predicted_plays	corrected_plays
2	7224	107	2.331359	2.234685
4	8324	96	2.102053	1.999991
3	6450	102	2.075093	1.976078
0	5531	618	1.976763	1.936538
1	5653	108	2.019267	1.923042

```
['Secrets by OneRepublic',
 'Transparency by White Denim',
 'Brave The Elements by Colossal',
 "Victoria (LP Version) by Old 97's",
 'The Big Gundown by The Prodigy']
```

CLUSTERING

	song_id	play_freq	predicted_plays	corrected_plays
2	7224	107	3.094797	2.998124
4	8324	96	2.311498	2.209436
1	9942	150	2.215039	2.133390
0	5531	618	2.124563	2.084337
3	4831	97	2.123783	2.022248

```
['Dog Days Are Over (Radio Edit) by Florence + The Machine',
 'Heaven Must Be Missing An Angel by Tavares',
 'Secrets by OneRepublic',
 "Bigger Isn't Better by The String Cheese Incident",
 'The Big Gundown by The Prodigy']
```

Content Filtering Top 10 Songs Similar to 'Learn To Fly' by The Foo Fighters:

```
1. 'Everlong' by The Foo Fighters
2. 'The Pretender' by The Foo Fighters
3. 'Nothing Better (Album)' by Postal Service
4. 'From Left To Right' by Boom Bip
5. 'Lifespan Of A Fly' by Bird and Bee
6. 'Under The Gun' by The Killers
7. 'I Need A Dollar' by Aloe Blacc
8. 'Feel The Love' by Cut Copy
9. 'All The Pretty Faces' by The Killers
10. 'Bones' by The Killers
```

As you can see, for each model, the predicted plays (corrected_plays) decrease as we go down the list. Also the highest precision value for any of the models is only 42%, which means in a top 10 recommendations, the best model is still only getting about 4 songs that are of interest to our user. The top n recommendations from an SME point of view are quite good and consistent to each other in style and sound, but they are not consistent within themselves and often have very strange suggestions inside them that are completely out of character to the rest of the songs.

Therefore, why use a model whose top 10 recommendations will get worse and less accurate as they go through the list?

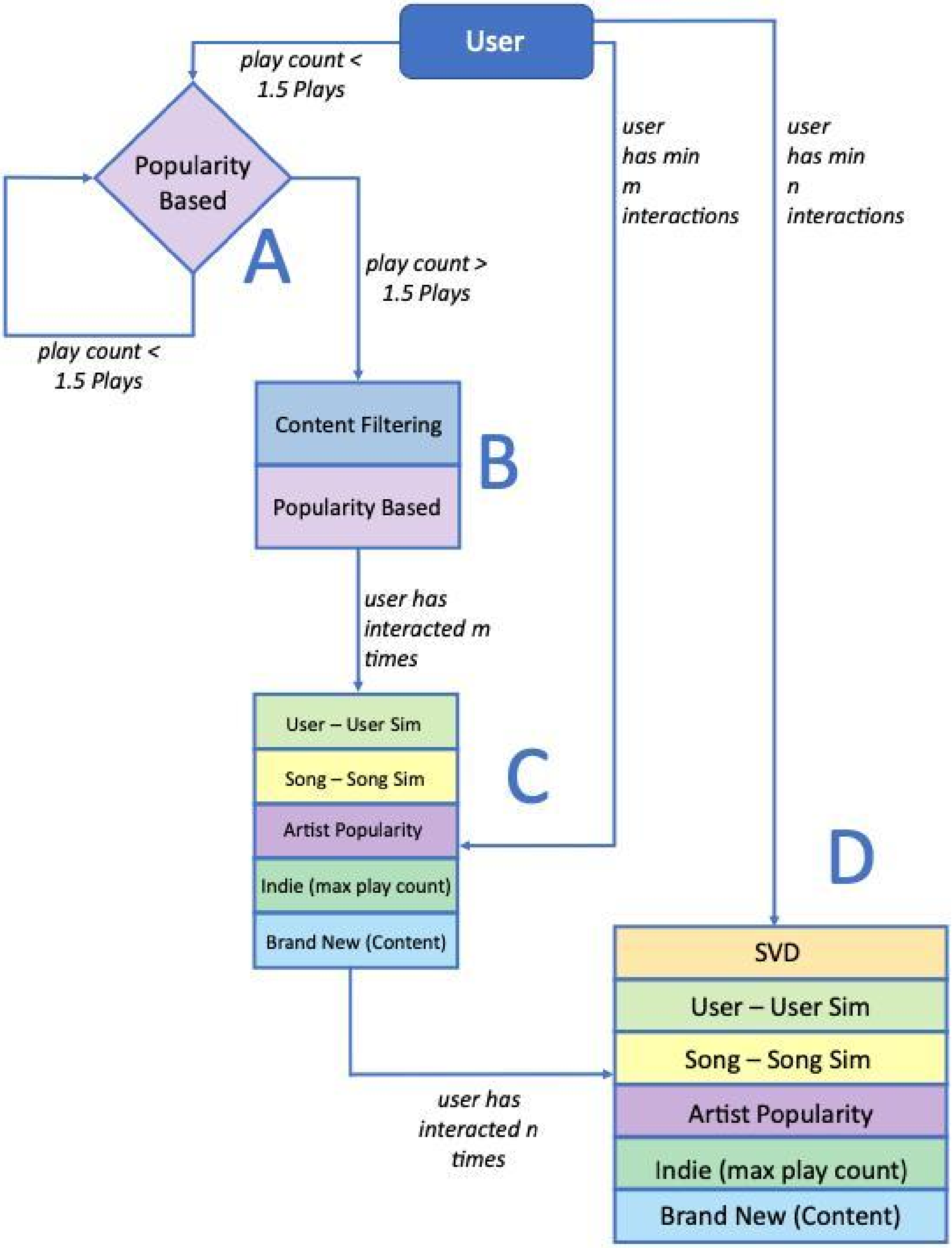
My proposal is to use a **hybrid model** constructed within a Decision Tree consisting of combinations of the best performing models that will be invoked at different times depending user interactions.

SUPER H!

(H IS FOR HYBRID!)

The ‘Super H’ hybrid system would consist of the optimised versions of the following models:

- **Popularity Based**
- **Content-Based Filtering**
- **Song-Song Similarity**
- **Artist Popularity Based**
- **User-User Similarity**
- **SVDMatrix Estimation**
- **Content-Based Filtering for New Artists (not yet trained)**



This graphic shows how the hybrid system will work. The number of interactions with songs a user has had will decide on which model/s will be producing their recommendations.

System A
A user who is new and has not played any songs or has not played any songs more than 1.5 times will be simply fed recommendations from a **Popularity** model.

System B
As soon as they play a song more than 1.5 times, they will drop through to system B which is a combination of the **Content Based** model as well as the **Popularity** model since these models do not require the user of interest to have any data.

System C
When a user has had *m* number of interactions with the platform they will drop through to system C which invokes the **User-User Similarity** model. This is the model that performed the best in the metrics tests but we couldn't use it until the user of concern gives us enough data since this model finds other similar users and creates recommendations based upon the activity of the similar users. This is why we require a user to have *m* number of interactions with the platform so that the model has enough data to work.

System C is a hybrid of mostly **User-User Similarity** but also uses **Song-Song Similarity, Popularity** (subset within Artists that the user likes). **Indie** is a User-User Similarity set with a maximum play count threshold that is lower to open the door to less popular indie artists). The '**Brand New'** model is a Content-based model that uses natural language analysis of song reviews for brand new artists (determined by a very low play count and play frequency, release date from the present year and no prior album releases).

System D
When a user has had *n* number of interactions with the platform, they will drop through to the final system D which is almost identical to System C but with the additional use of **SVD Matrix Estimation** is a mathematically sophisticated model that only looks at the user's prior interactions to find the latent variables. This model is highly personalised to each individual user and adds value to the final hybrid model.

The ultimate Top Ten Song Recommendations System D for a user will look something like this (with the order to be determined through further testing):

Source Model	No. of Songs for Final Model	Key
User – User Sim	4	a
SVD	2	b
Song – Song Sim	1	c
Artist Popularity	1	d
Indie (max play count)	1	e
Brand New (Content)	1	f

Final Song Recommendation	$Source_k$
1	a_1
2	a_2
3	b_1
4	c_1
5	d_1
6	a_3
7	e_1
8	b_2
9	f_1
10	a_4

where $_i$ = recommendation number from source model

Problems Solved by the Super H Hybrid System:

New Users - Systems A and B take care of the Cold Start problem of new users by using models that require the least amount of user data.

Failing Models - None of the models were not robust enough on their own, but with the hybrid solution, together they give much accurate recommendations that are consistent through the 10 songs.

Feedback Loops - By using many different models that work in different ways we are avoiding feedback loops where the user will fall into a very narrow channel of songs being recommended. The additional placement of independent and new artists, and the thought behind each and every model, ensures that the user is not simply receiving very safe recommendations of one type. But they are also going to experience an exciting and rewarding music discovery journey.

New Songs - New songs by popular artists will appear in many of the models. And new songs by new artists are also accommodated in the 'Brand New' Content Filtering model. This model does not require play frequency or count from the songs in order for them to be picked for recommendations.

Going Forwards:

This is a complex hybrid system that requires two models that have not yet been trained, tweaked and tested on the data that I would like them to be working with:

- Artist Popularity - Finding popular songs from Artists that the user likes
- 'Brand New' Content Based using online song reviews

The **Artist Popularity** model requires creating, training and testing. There must be data manipulation in order to create a dataset of artist_frequency which will give us how many times the user has played songs by a particular artist. A popularity model can then be run within the dataset of a particular artist and we can also include an IF statement to recommend the most popular song by that Artist that the user has not played.

Brand New Content Based model needs new data to be able to be created, trained and tested. NLTK (Natural Language Toolkit) will scrape online reviews of songs in the dataset that meet the following criteria: very low play count, release date from present year and no prior album releases from the artist)

Play Count Capping:
In the models I made for this project I used a cut-off in the dataset for any songs with more than 5 play counts by a user. This was because the play counts higher than 5 were very rare and it would have made the matrix sparse. But I propose instead of cutting out the play count greater than 5, that we instead cap them. This will mean the higher play count songs are still included and songs are not being punished for being popular with a user. However, we may want to cut them off at an excessively high number as this could be an indication of the presence of bots.

I also took issue with the fact that play counts were given as integers whereas our predicted play counts were floating point values. It would be of interest for the future to gather data sets with decimal place values for play counts to be able to measure more accurately how well the models are performing.

Limitations and Potential Issues / Costs

This 'Super H' hybrid model has a lot of thought behind it, solving many problems and with a long-term vision to give Independent and Brand New artists a door to their potential fans. However, a complex model such as this is not without some issues and costs.

More Data Needed:
In order to accommodate brand new bands and artists we will **require more data** that was not made available to me for this period. I would suggest a fast beta roll-out for the NLTK to analyse song reviews of new songs that look for artists of comparison within the review. This can be used to train the Content Based Model.

The monetary **costs** of this new data should be relatively low as we don't have to pay for the data itself as it will be scraped from the internet. It will cost time and resources to create and maintain the model that will do this. We will probably require additional cloud storage to accommodate an ever-increasing large dataset.

More Recent Data:
It is also worth noting that the data used to create these models was collected in 2011 and users may have changed the way they interact with the platform by 2023. I would also suggest finding more recent data sets to train all the final models on.

The costs of finding this data are unknown to me and something to be considered and also cloud storage for when this model is scaled up for launch as the datasets will be very large.

More Time:

- More data must be found and more models must be built, tested and tuned.

- The Decision Tree requires the values of *m* and *n* to be found. So we know when is the most ideal time for a user to be moved to the next System within it. This will take some experimenting and tuning.

- The final blend of Top 10 Recommendations that come from different models needs time to be tweaked and decided - for example what to do with overlapping recommendations and when to use songs the user has already listened to.

- I would also suggest that the hybrid systems B, C and D should be treated as complete models themselves and be tested for Error Measurement, Precision and Recall in the final testing and tuning stage.

Given the further action required I am suggesting a **further 4-6 months** of data collection, model building and Decision Tree structuring.

CONCLUSION

This brand new master model aims to gain the **trust of the user** and keep them from leaving for another platform. It is hoped that they will promote the quality of the music discovery algorithm to their friends and that we will become the most trusted and enjoyed platform. It also gains the respect of the artists and the listeners who care about new music. Our long-vision platform can be proud to boast not only an **engaged and happy usership**, but also a platform that is **thoughtfully and ethically proactive about the music industry** by avoiding feedback loops and inspiring creativity and new music.