Music Recommender Report

Background: I have taken a job with my favorite music streaming service and they've asked me to come up with a way to make recommendations based on a user's play count.

Objective: Build and compare two types of recommender systems (a SVD and KNN) using a subset of the Million Song Dataset

Getting the Data

Merge three datasets into a final dataset, the final data contains user id, song id, play counts, track id, artist, song name. Total 1450942 observations.

Preprocessing

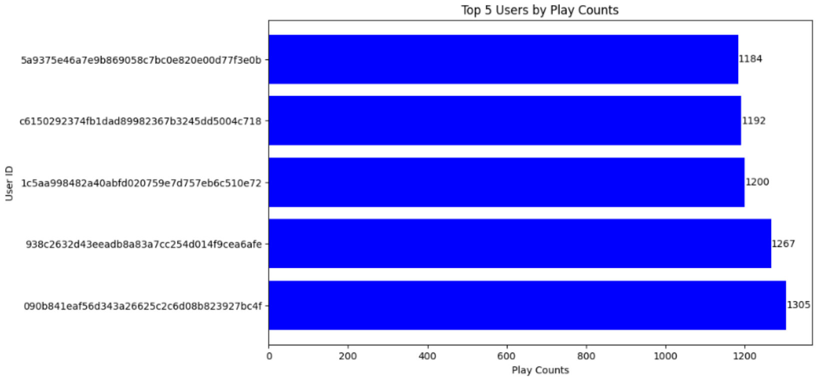
Use the play count column as a surrogate for an explicit rating, play counts range from 1 to 923. BIN the data 1 through 10. The assumption is that if someone listened to a song only once then they didn't love it but if they listened to something say 5 or more times, they probably enjoyed it.

Exploratory Data Analysis

Basic analysis of the users

1) What users listen the most, how many play counts?

The top 5 users who listen the most and their play counts shows as follows:



2) How many unique songs etc?

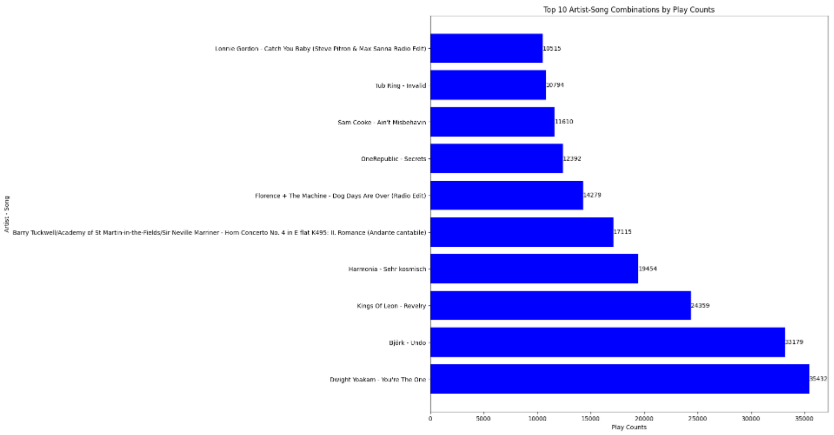
- The number of unique songs are 163209.

- The number of unique user are 110002.

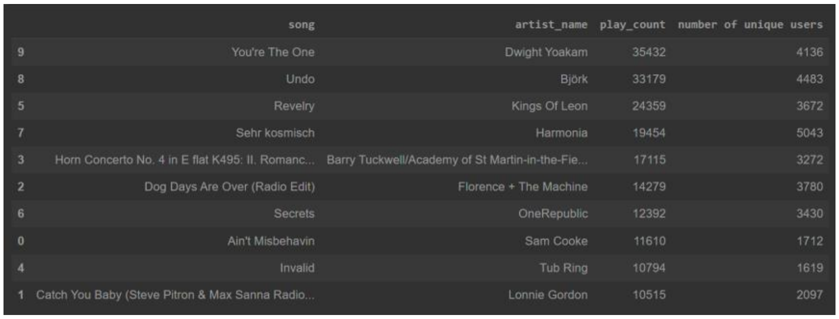
- The number of unique play count are 110002.

Basis analysis of the artists and songs

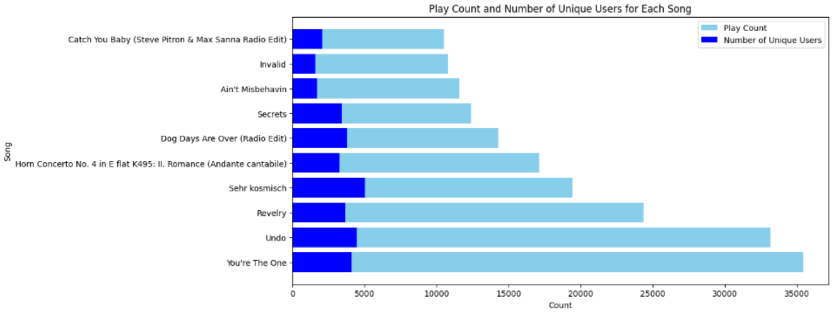
1) What are the most popular songs and artists, how many unique users have listened to them?



2) How many unique users have listened to them?



	song	artist_name	play_count	number of unique users
9	You're The One	Dwight Yoakam	35432	4136
8	Undo	Björk	33179	4483
5	Revelry	Kings Of Leon	24359	3672
7	Sehr kosmisch	Harmonia	19454	5043
3	Horn Concerto No. 4 in E flat K495: II. Romanc...	Barry Tuckwell/Academy of St Martin-in-the-Fie...	17115	3272
2	Dog Days Are Over (Radio Edit)	Florence + The Machine	14279	3780
6	Secrets	OneRepublic	12392	3430
0	Ain't Misbehavin	Sam Cooke	11610	1712
4	Invalid	Tub Ring	10794	1619
1	Catch You Baby (Steve Pitron & Max Sanna Radio...	Lonnie Gordon	10515	2097



Train & Evaluate Recommenders

Since the dataset is huge, I random a sample of dataset named filtered_data

1) Create a Global BaselineOnly model, set 'method': 'als','n_epochs': 50, 'reg_u': 5, 'reg_i': 5

Evaluating Global Mean Baseline: RMSE: 2.3583, MAE: 1.6748

2) Create a basic SVD model, default parameter value

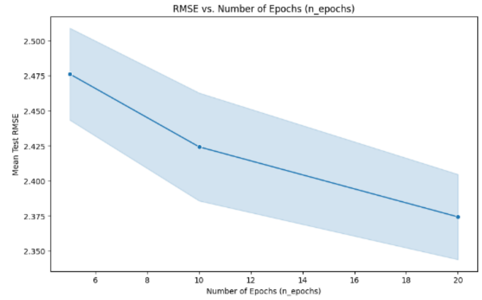
Evaluating basic SVD recommender model: RMSE: 2.3756, MAE: 1.6970

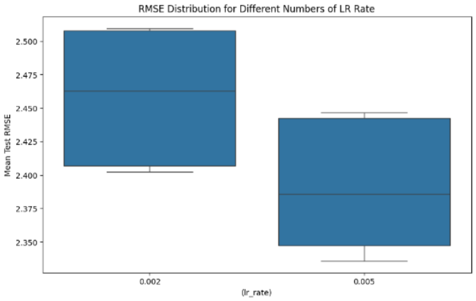
3) Hyperparameter tuning on the SVD

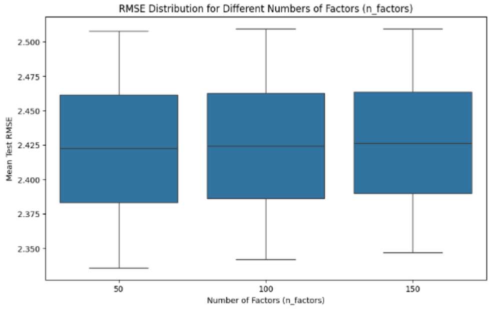
The values for hyperparameters are the number of factors (n_factors): [50, 100, 150], the number of epochs (n_epochs): [5, 10, 20], the learning rate (lr_all): [0.002, 0.005], the regularization term (reg_all): [0.02, 0.05]

SVD models often have hyperparameters that can be fine-tuned for better performance. Grid search or other hyperparameter optimization techniques can be applied to find the optimal set of parameters.

Upon visualizing the parameters, we observed that RMSE decreases as the number of Epochs decrease. Additionally, setting the learning rate as 0.005, it will has a lower RMSE. However, the number of factors has a relatively small effect on the value of RMSE.







Finally, we got the best parameter is the number of factors (n_factors) = 50 , the number of epochs (n_epochs) = 20 ,the learning rate (lr_all) = 0.005, and the regularization term (reg_all) = 0.002

4) Refit with best parameters

Evaluating SVD recommender model: RMSE: 2.3707,MAE: 1.6905

5) Compare the Baseline() performance to that of the SVD.

Evaluating SVD vs. Baseline:

SVD RMSE: 2.3707

SVD MAE: 1.6905

Baseline RMSE: 2.3583

Baseline MAE: 1.6748

We can see that Baseline() RMSE/MAE is a little lower than the SVD but we still prefer using SVD. The reason is

- First, baselines are simple models used as a point of comparison for more complex algorithms. They are crucial for understanding how much better a sophisticated model is compared to a simple approach. Thus, it is a fundamental in recommendation systems and any sophisticated model should exceed to be considered valuable, such as the model we used SVD should have a greater performance compare to the model used baseline.

- In this case I used global baseline algorithms. This is the simplest form of baseline. It involves calculating a single average rating across all users and items in the dataset and using this average as the prediction for all user-item pairs. It is used as a starting point.

- Secondly, Singular Value Decomposition (SVD) is a sophisticated matrix factorization technique commonly used in recommendation systems, particularly for collaborative filtering. It used to decompose user-item rating matrix into matrices representing latent (hidden) factors for users and items.

- In this case, I try to learn the latent factors that best approximate the original user-item rating matrix. The goal is to minimize the difference between predicted and actual play_count_binned in the training set, thereby learning the preferences and characteristics of users and songs.

- It's important to understand that RMSE and MAE are metrics used to evaluate the accuracy and effectiveness of a model's predictions. While these are common metrics, there are other ways to evaluate a model's performance, such as diversity and relativity. One way to use SVD is to recommend songs to users that they haven't listened to before but would likely enjoy based on their previous listening habits.

Answer the following:

1. For a random sample of 5 users with 10 or more song plays make 5 recommendations of songs they have not listened to with your SVD.

I discovered that there's a partial overlap in the song recommendations for the five users, indicating that the model has a certain diversity.

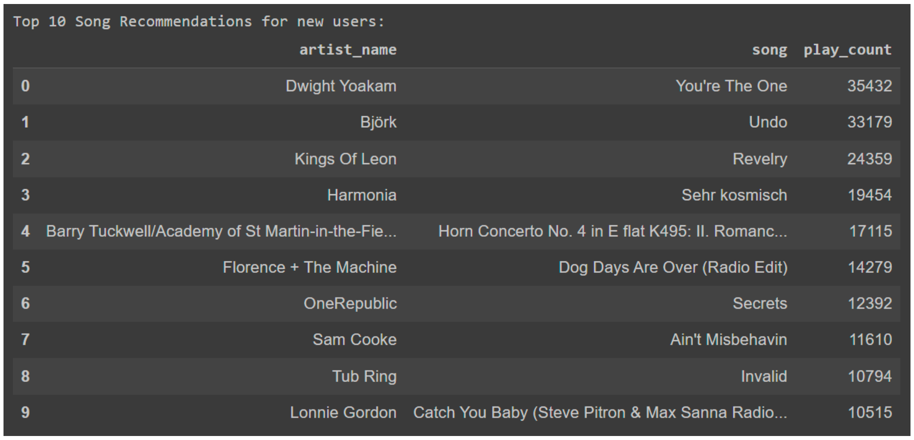
2. Recommendation systems should provide “relevant” recommendations expanding the user’s pool of options (songs in our case) what would you do to improve your recommendation to expand a user’s relevant song recommendation pool?

- If we could have more information about song's genre, lyrics, we could incorporate content-based filtering to recommend songs. This helps in diversifying recommendations by focusing on the characteristics of the items.For example, if the user has listened to one piece of music from the album, we recommend other pieces of music from the album to the user, and if the user listens to the song and listens to it many times, we can record the performance and update the database so that we can combine the content-based and user-based model to make better recommendations.

- Since we used a random sample of 10000 records, the model we created can’t cover all the features of original data, thus, there still have some room to increase the relativity and diversity of recommendations.

3. What are your top 10 recommendations for a net new user? That is a user with no user/song play count? essentially the cold start problem.

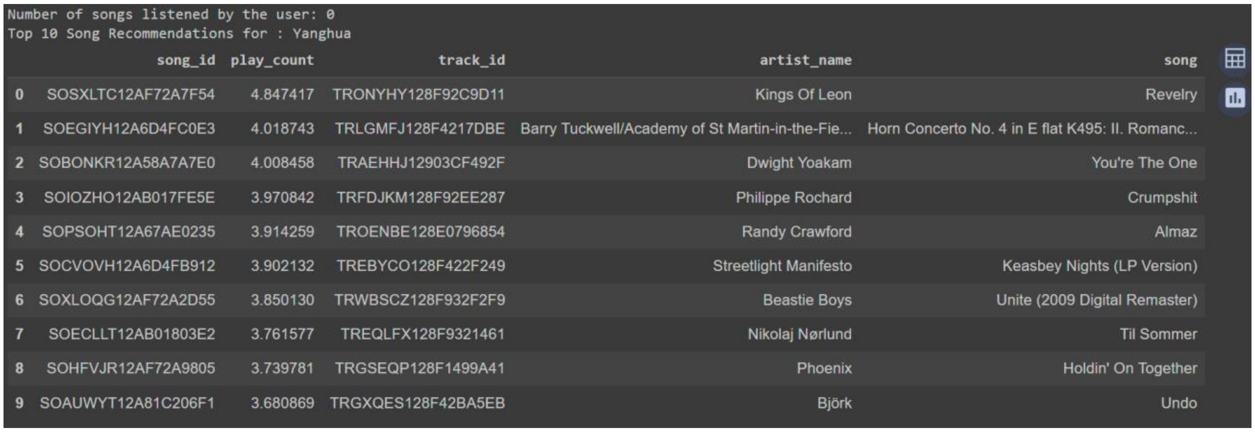
For a net new user, since we have limited information, we could make a non-personalized recommendation of songs. For example, recommend the songs that has the top 10 play count. These are likely to be well-received by a board audience.



```
Top 10 Song Recommendations for new users:
```

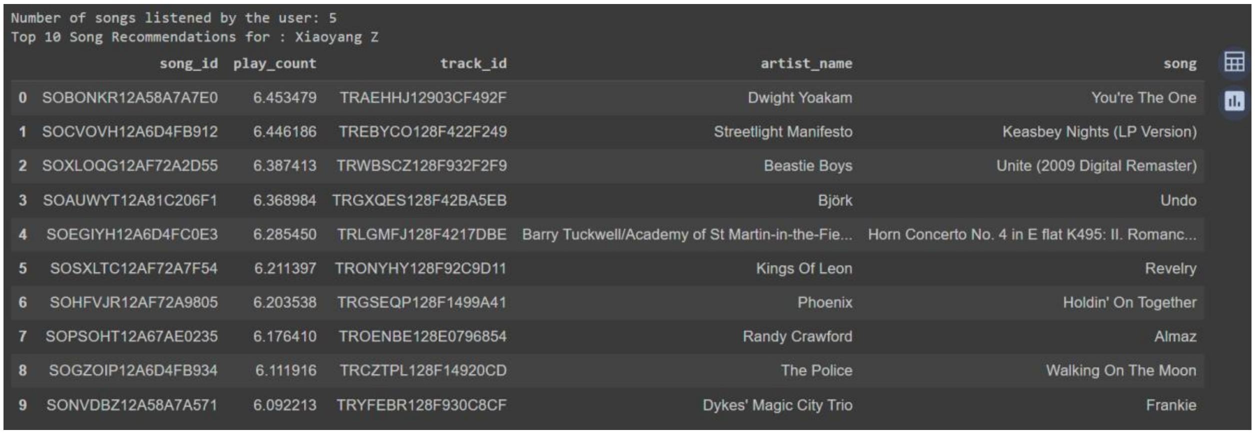
	artist_name	song	play_count
0	Dwight Yoakam	You're The One	35432
1	Björk	Undo	33179
2	Kings Of Leon	Revelry	24359
3	Harmonia	Sehr kosmisch	19454
4	Barry Tuckwell/Academy of St Martin-in-the-Fie...	Horn Concerto No. 4 in E flat K495: II. Romanc...	17115
5	Florence + The Machine	Dog Days Are Over (Radio Edit)	14279
6	OneRepublic	Secrets	12392
7	Sam Cooke	Ain't Misbehavin	11610
8	Tub Ring	Invalid	10794
9	Lonnie Gordon	Catch You Baby (Steve Pitron & Max Sanna Radio...	10515

4. What are your top 10 recommendations for you and your peer?



```
Number of songs listened by the user: 0
Top 10 Song Recommendations for : Yanghua
```

	song_id	play_count	track_id	artist_name	song
0	SOSXLTC12AF72A7F54	4.847417	TRONYHY128F92C9D11	Kings Of Leon	Revelry
1	SOEGIYH12A6D4FC0E3	4.018743	TRLGMFJ128F4217DBE	Barry Tuckwell/Academy of St Martin-in-the-Fie...	Horn Concerto No. 4 in E flat K495: II. Romanc...
2	SOBONKR12A58A7A7E0	4.008458	TRAEHHJ12903CF492F	Dwight Yoakam	You're The One
3	SOIOZHO12AB017FE5E	3.970842	TRFDJKM128F92EE287	Philippe Rochard	Crumpshit
4	SOPSOHT12A67AE0235	3.914259	TROENBE128E0796854	Randy Crawford	Almaz
5	SOCVOVH12A6D4FB912	3.902132	TREBYCO128F422F249	Streetlight Manifesto	Keasbey Nights (LP Version)
6	SOXLOQG12AF72A2D55	3.850130	TRWBSCZ128F932F2F9	Beastie Boys	Unite (2009 Digital Remaster)
7	SOECLLT12AB01803E2	3.761577	TREQLFX128F9321461	Nikolaj Nørlund	Til Sommer
8	SOHFVJR12AF72A9805	3.739781	TRGSEQP128F1499A41	Phoenix	Holdin' On Together
9	SOAUWYT12A81C206F1	3.680869	TRGXQES128F42BA5EB	Björk	Undo

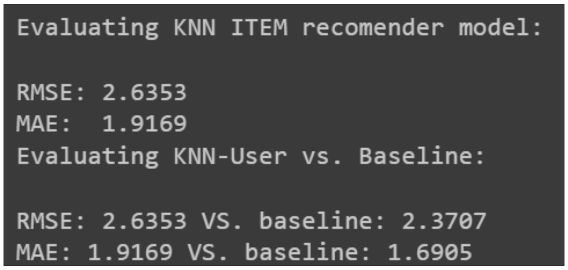


```
Number of songs listened by the user: 5
Top 10 Song Recommendations for : Xiaoyang Z
```

	song_id	play_count	track_id	artist_name	song
0	SOBONKR12A58A7A7E0	6.453479	TRAEHHJ12903CF492F	Dwight Yoakam	You're The One
1	SOCVOVH12A6D4FB912	6.446186	TREBYCO128F422F249	Streetlight Manifesto	Keasbey Nights (LP Version)
2	SOXLOQG12AF72A2D55	6.387413	TRWBSCZ128F932F2F9	Beastie Boys	Unite (2009 Digital Remaster)
3	SOAUWYT12A81C206F1	6.368984	TRGXQES128F42BA5EB	Björk	Undo
4	SOEGIYH12A6D4FC0E3	6.285450	TRLGMFJ128F4217DBE	Barry Tuckwell/Academy of St Martin-in-the-Fie...	Horn Concerto No. 4 in E flat K495: II. Romanc...
5	SOSXLTC12AF72A7F54	6.211397	TRONYHY128F92C9D11	Kings Of Leon	Revelry
6	SOHFVJR12AF72A9805	6.203538	TRGSEQP128F1499A41	Phoenix	Holdin' On Together
7	SOPSOHT12A67AE0235	6.176410	TROENBE128E0796854	Randy Crawford	Almaz
8	SOGZOIP12A6D4FB934	6.111916	TRCZTPL128F14920CD	The Police	Walking On The Moon
9	SONVDBZ12A58A7A571	6.092213	TRYFEBR128F930C8CF	Dykes' Magic City Trio	Frankie

a. Do the recommendations make sense? What could you do to improve them? Do you think an Item based KNN might better?

Yes, we all like the songs recommended. Also it showed that the model could recommend different songs for different users.



```
Evaluating KNN ITEM recomender model:

RMSE: 2.6353
MAE:  1.9169
Evaluating KNN-User vs. Baseline:

RMSE: 2.6353 VS. baseline: 2.3707
MAE: 1.9169 VS. baseline: 1.6905
```

- Item-based KNN relies on the similarity between items (songs) to make recommendations, which can be easy to understand and interpret can potentially offer more serendipitous recommendations since it relies on item similarities rather than underlying latent factors .If we prefer increase relativity, we prefer use KNN. But KNN may face challenges when new items are introduced since they lack historical interaction data for similarity calculations. Also, KNN is less diversity compare to SVD. The recommendation based on KNN algorithm is relatively fixed, and the recommended song list has a higher repetition rate.

- SVD's recommendations are based on latent factor that represent user preferences and item characteristics. The recommendation based on SVD algorithm is more diversity, and the recommended song list has a lower repetition rate.

- Thus, my recommendation is we could use both models to recommend songs and combine the recommended songs to the users.