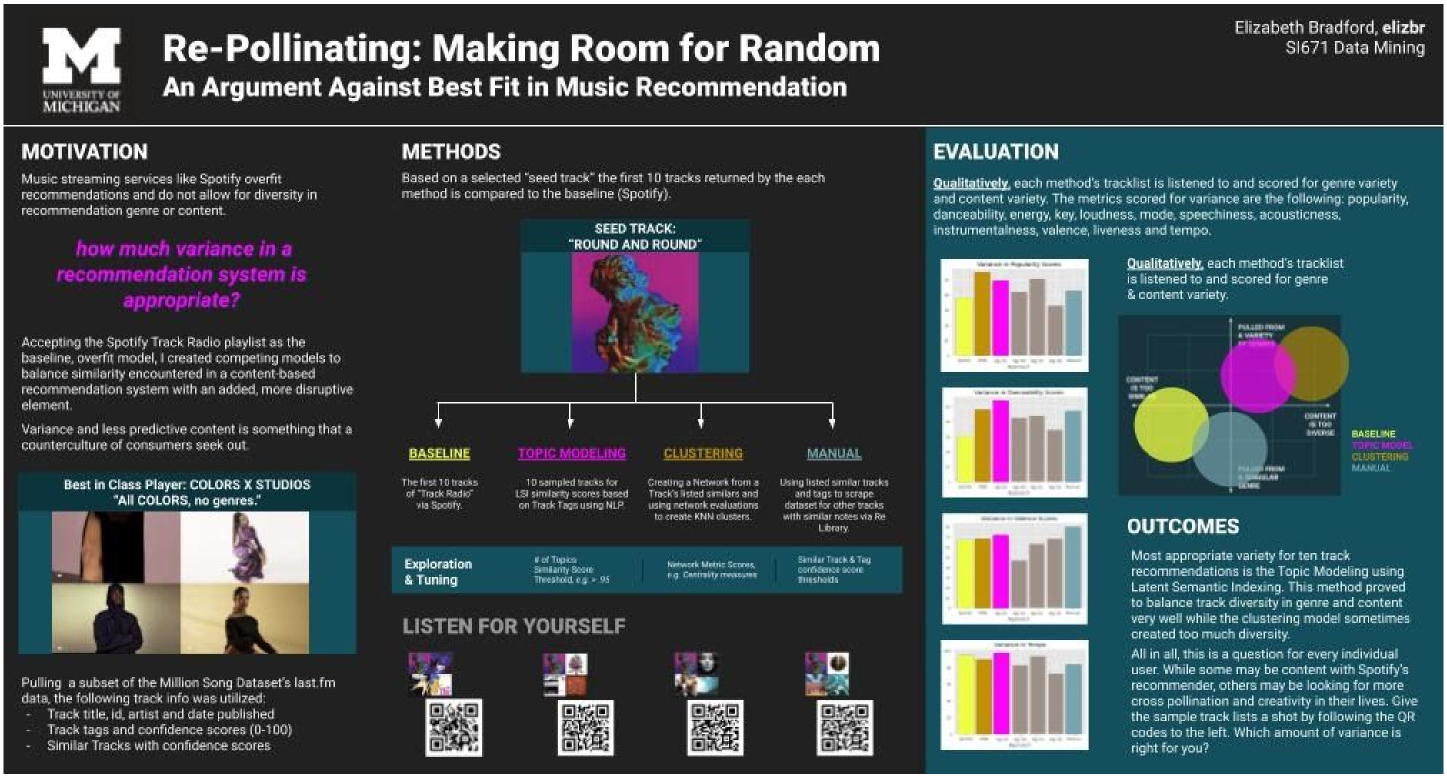


Re-Pollinating: Making Room for Random

An Argument Against Best Fit in Music Recommendation

Elizabeth Bradford, **elizbr**
SI671 Data Mining

MOTIVATION

Music streaming services like Spotify overfit recommendations and do not allow for diversity in recommendation genre or content.

how much variance in a recommendation system is appropriate?

Accepting the Spotify Track Radio playlist as the baseline, overfit model, I created competing models to balance similarity encountered in a content-based recommendation system with an added, more disruptive element.

Variance and less predictive content is something that a counterculture of consumers seek out.

Best in Class Player: COLORS X STUDIOS "All COLORS, no genres."

Pulling a subset of the Million Song Dataset's last.fm data, the following track info was utilized:

- Track title, id, artist and date published
- Track tags and confidence scores (0-100)
- Similar Tracks with confidence scores

METHODS

Based on a selected "seed track" the first 10 tracks returned by the each method is compared to the baseline (Spotify).

SEED TRACK: "ROUND AND ROUND"

BASELINE
The first 10 tracks of "Track Radio" via Spotify.

TOPIC MODELING
10 sampled tracks for LSI similarity scores based on Track Tags using NLP.

CLUSTERING
Creating a Network from a Track's listed similars and using network evaluations to create KNN clusters.

MANUAL
Using listed similar tracks and tags to scrape dataset for other tracks with similar notes via Re Library.

Exploration & Tuning
# of Topics Similarity Score Threshold, e.g. > .95
Network Metric Scores, e.g. Centrality measures
Similar Track & Tag confidence score thresholds

LISTEN FOR YOURSELF

EVALUATION

Qualitatively, each method's tracklist is listened to and scored for genre variety and content variety. The metrics scored for variance are the following: popularity, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, valence, liveness and tempo.

Qualitatively, each method's tracklist is listened to and scored for genre & content variety.

Variance in Popularity Scores
Variance in Danceability Scores
Variance in Valence Scores
Variance in Tempo
PULLED FROM A VARIETY OF GENRES
CONTENT IS TOO SIMILAR
CONTENT IS TOO DIVERSE
PULLED FROM A SINGULAR GENRE
BASELINE
TOPIC MODEL
CLUSTERING
MANUAL

OUTCOMES

Most appropriate variety for ten track recommendations is the Topic Modeling using Latent Semantic Indexing. This method proved to balance track diversity in genre and content very well while the clustering model sometimes created too much diversity.

All in all, this is a question for every individual user. While some may be content with Spotify's recommender, others may be looking for more cross pollination and creativity in their lives. Give the sample track lists a shot by following the QR codes to the left. Which amount of variance is right for you?

[POSTER LINK]

Re-Pollinating: Making Room for Random

An Argument Against Best Fit in Music Recommendation

Elizabeth Bradford
UMSI, School of Information
University of Michigan, Ann Arbor
elizbr@umich.edu

1 Problem Statement & Motivation

Within streaming services, we encounter a lot of overfitted music recommendation tools in playlist generators and weekly Releases. Services like Spotify have invested and delivered on algorithms that promote and recommend tracks that highly correlate to user's tastes. Because of this, it's often hard to break out of a particular behavioural pattern. The beauty of music is its ability to translate emotional depth and through adding interest and enjoyment to our lives. Listening is both social and antisocial.

In this project, my goal was to explore the question of *how much variance in a recommendation system is appropriate?* Accepting the Spotify Song Radio playlist as the baseline, overfit model, I will work to create a competing model that balances the similarity encountered in a content-based recommendation system with an added, more disruptive element. Looking at the success of curator types like COLORS x STUDIOS and Metrograph Online who both exist in recommendation-system dominated arenas of music and film, variance and less predictive content is something that a counterculture of consumers seek out. The goal of this Project is to explore the limits of random input into playlist generation.

2 Methodological Approach

The Dataset For the exploration and testing of this project, the publicly accessible Million Song Dataset's last.fm dataset was utilized. This dataset contains one million songs with information including the following:

- Track title, id, artist and date published
- Tags – The tags are words or labels that have been assigned to the track. Within the dataset, the amount of tags varies from none to about 70. Each tag is listed as a sublist containing the tag and the score of the tag from 0 to 100 for accuracy of the tag.
- Similars – Similar to tags, similars are a list of lists of a track's similar tracks with the associated track's id and a similarity score from 0 to 1. The amount of similar songs ranges from none to about 40.

Preprocessing Because this dataset is truly 1,000,000 tracks, I selected a subset to use for exploration and analysis of the dataset. Using the glob library, the dataset imports in sections easily. From a collection of three subsets, of about 260,000 tracks each, I selected a random sample of 20,000 tracks. From this selection, I again filter down to exclude members of this sample that have no listed tags resulting in just over 10,000 tracks. The tracks included in the subset were optimized for a better-connected network since a completely random sample of 10,000 from 1 million results in a not usable graph. This optimization was done by prioritizing tracks mentioned as similar to other tracks in the output subset.

Data Contents Within the sampled database, we see a distribution of tags. Similarly, there is a variable amount of similar tracks listed for each track_id. For preprocess, the tags are converted into list format and reattached to the dataset.

Approaches Below is a list of approaches which will be covered in more detail in the following section. This project relies heavily on comparison of recommendations based on a sample. For these purposes, I approached the subject of similarity ranking in a few different manners. Using an example track from the dataset, I compared the first ten songs recommended based on each experimented approach. Overall, the goal of the project is to explore and understand appropriate variance for a user experience. Using the Spotify Song Radio playlist as a baseline of too similar, not enough variance, each of the first 10 songs recommended through these explorative approaches will be measured qualitatively and quantitatively against one another and the Spotify example.

1. *(Manual) Raw Track Content* Utilizing a track's artist, similar tracks and tag information, this approach was meant to be a measure against the other two. This approach pulls tracks mentioned as similar in the dataset as well as from tracks with tags in common.
2. *Topic Modeling* Using the track tags available in the dataset, I explored a system of track similarity ranking based on topic modeling with a Latent Semantic Index structure.
3. *Clustering in Network Analysis* Through analysis of a track's listed similar tracks, I build out and explore track similarity scores based on network graph metrics such as degree, centrality and clustering coefficient. From the node scores for these network metrics, I experimented with and fit machine learning clustering models including KNN.

Designed Evaluation The models outlined above are measured and evaluated in both qualitative and quantitative measures:

- *Quantitative* Pulling from the Spotify API, track metrics like danceability, accousticness and genre will be utilized to create a measurement for similarity for each approach's produced playlist. These scores will easily quantify diversity between the explored approaches and the baseline Spotify playlist. This requires me to manually add the returned recommendation playlists to spotify and pull their information from the API. With the returned JSON, I was able to easily convert content into dataframes using Pandas for analysis.
- *Qualitative* With three selected seed tracks, I spent about 5 hours listening to each variation of the recommendation playlists for each track (10 tracks each). This piece of feedback helped round out the user experience of randomness in the playlist, supplementing the quantitative findings and paying homage to the curatorial players mentioned in the introduction. Qualitative feedback and analysis is central to any user-centric problem or design. This was a leg of the project where I could flex my past professional skillset in user-centric design with specified qualitative evaluation.

3 Experimentation

3.1 (Manual) Raw Track Content

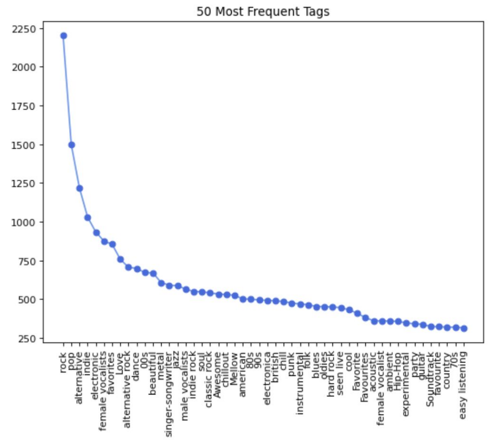
Overview The goal of this approach is to manually scrape the contents of the available dataset to create a recommendation list based on a combination of similar artists, similar tracks as listed in the dataset and tracks with highly correlated tags.

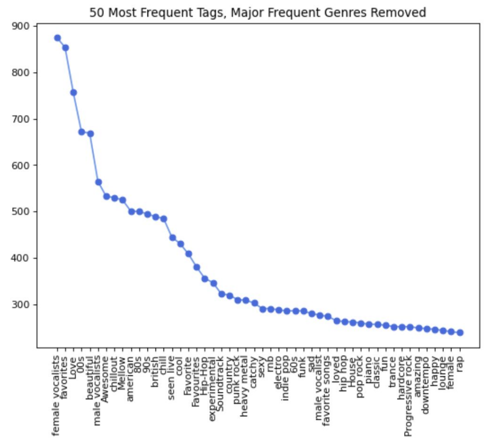
Within the subset of the dataset being used, there is a lot of variance in the type and amount of information available for each track which highly impacts the ability of this method to perform well. For example, when running this model for Björk's only track included in the dataset, an empty set is returned since this track has no listed similar tracks or tags as well as no artist similarity. My hypothesis about this approach was that it would overfit in recommendation homogeneity, similar to Spotify's "Track Radio" which is serving as a baseline.

Experimentations In selecting importance, I found that giving priority or more weight to similar tracks for inclusion in the recommended playlist over similar tags results in better results qualitatively.

Within the dataset, weights of similarity and accuracy are given to both listed similar tracks and tags for each track. For example, for the Third Eye Blind track listed above, we see scores associated for similar tracks ranging from 0 to 1. Similarly, each listed tag is ranked for accuracy or match on a scale of 0 to 100. In terms of content selection, I found that allowing similar tracks with a score in the range of .7 to 1.0 was appropriate. For tags, it was important to be more selective by setting a higher threshold, returned tracks would have to have a similar tag with a match score of over .95. Through experimenting with these two hyperparameters, I am able to prioritize the inclusion of similar tracks listed over tracks with shared tags. This improves the overall accuracy of the model and allows the model to perform without returning a completely randomized recommendation list.

Still, tags play a crucial role in this structure of recommendation. Through looking at the tags column content example above, it is not surprising for this method to return a heavily genre-lead 10 song playlist.



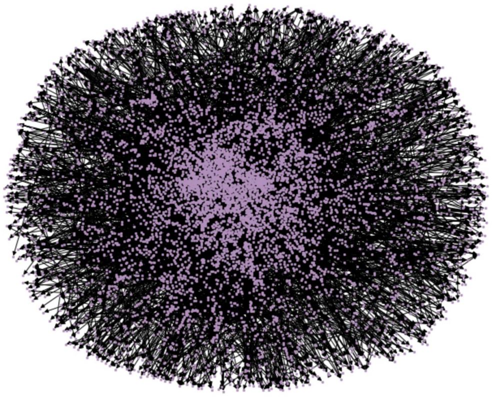


Considering the tag content throughout the entire document, we can see that some of the larger umbrella genres have exponentially more mentions than other types of tags. In an effort to negate recommendations for highly scored 'rock' or 'pop' tracks from having very similar outputs, I cleaned the tags up by removing the top 50 most occuring tags inside of the dataset, targeting general genre tags. The resulting returns from this approach allowed for more diversity among seed songs in rock or pop by requiring the system to pull tracks with high matching scores for smaller tag groups such as 'beautiful,' 'piano,' or fun.

As we look into the tag contents, we can see some general biases in content distribution for genres. While pop, rock and alternative are all highly occurring tags, other genres (of equal size and cultural weight) are tagged much less frequently like 'rap.' This is something I had to keep in mind throughout and into evaluation comparisons with the baseline Spotify set.

3.2 Clustering with Network Analysis

Overview The subset of the million song database being used is 10,693. For the creation of the network graph, I used track_id and the track_ids listed in similars to create an edgelist from the dataset contents. Through the addition of the track_ids outlined as similar tracks to the initial tracks in the 10,693, the number of nodes in the graph increases to 274,913 (about ⅓ of the total dataset before sampling) with a total of 910,492 edges. As seen in the graph depiction to the right, this is a massive map with appropriate rich information for only about 4% of the nodes included. Because of this, it was first necessary to reform the node list by requiring included nodes to contain one of the originally listed 10,693 track_ids as either the from_node or to_node for each edge. This approach allowed the results of the similarity scoring to be more meaningful by ensuring all tracks included in the graph were fully flushed out with information.



Resulting Filtered Graph

Experimentations Moving forward, I assigned network metric scores to the nodes of the graph including the following:

- *Clustering Coefficient* – estimation of the amount of triangles for each node; extended measurement of node connectivity
- *PageRank* – the ranking score of each node based on the structure of incoming edges
- *Degree Centrality* – another measure of connection equivalent to the fraction of $^1/_n$ where n is the number of nodes it is connected to
- *Closeness Centrality* – a measurement of centrality for a node where a higher score indicates a closer distance to the center of the graph
- *Betweenness Centrality* – computes the shortest path based on the shortest paths of connected nodes
- *Degree* – the number of edges adjacent to the node

Even with the filtered, more tightly-knit graph in play, a lot of null values result from these scoring methods. Because, for many methods, zero and zero-like values are the scores affixed to nodes, there was not a great difference in results between null values that were imputed versus replaced with zero. For the sake of simplicity I moved forward by replacing all null network scoring values with 0.

The initial plan or idea behind the network driven recommendation approach was to completely disregard the genre and subgenre tags included in the original dataset. After each node received its scores for each of the network metrics, I used SKLearn's K-Nearest Neighbor to create groups with about ten track members based on the network metric scores. The resulting group for each seed track was translated into the recommended playlist for testing against the other approaches.

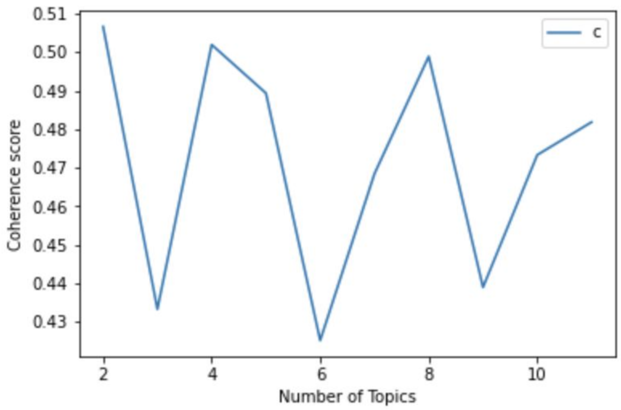
3.3 Latent Semantic Topic Modeling

Overview This wouldn't be a true recommendation project without an information retrieval method and an index implementation. Using the tags as a body of text, I chose to process this with a Latent Semantic Index to model topics and rank tracks for similarity based on their scores.
First, I had to format the text body:

1. Format tag text content
2. Tokenize with the NLTK library & stop word removal
3. Stem the tokens with NLTK's PorterStemmer method

Next, I built the dictionary and corpus matrix based on the cleaned and processed tag text. Using the Gensum library, I built an LSA model which then allowed me to measure track similarity based on the model and through setting some hyperparameters.

Experimentations With topic modeling, topic coherence is a primary goal in order to have useful and interesting results. In order to create a model with the best coherence, I experimented with the number of topics of the model. From the resulting model topic contents, it became clear that for this body of text, a text that was largely only keywords to begin with, some of the models I tried with only a few topics were more coherent. Still, I was able to graph out coherence scores as outlined by the gensim method of compute_coherence_values() which gave me more visibility into overall coherence performance.



From this point, I chose to pull recommendation track lists for lsa models with 4, 6 and 8 topics.

Based on the coherence scores visualized above, I expected to see slightly more accurate/literal recommendations based on the lsa models with 4 and 8 topics when compared to the 6 topic model. Still, between the recommendations from the 4 topic model and 8 topic model, I anticipated a preference for the 4 because there would be slightly less specific scores applied, and therefore I expected the resulting playlist recommendation to be slightly less prescribed.

Like with the raw content approach above, I reworked this approach and experimentation with a list of tags cleaned of the top 50 tags containing umbrella genres like 'pop' and 'rock.' I anticipated that this model would do a noticeably better job of recommending songs that were both similar but also not closely linked by genre.

4 Analysis of Results

Evaluation and Measures As mentioned in the outline of approach above, three tracks were selected within the dataset to serve as constants in the weighing and evaluating process between the experimental approaches and the baseline Spotify playlist. Each was selected for diversity in amount of listed tags, amount of similar tracks and general time period of creation. By selecting a diverse but small subset of seeds, I am able to understand a decent amount of the variance within the dataset's tracks without sacrificing quality due to timing limitations. The three seed tracks are listed below:

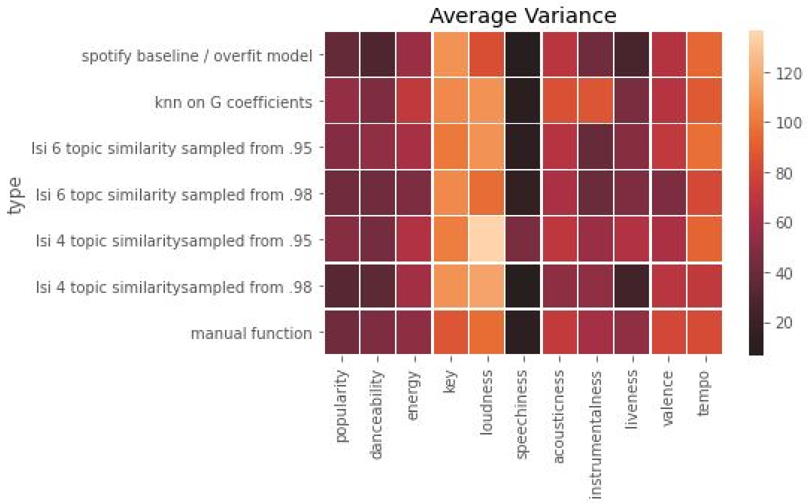
1. *"Losing A Whole Year"* by Third Eye Blind (Alternative) | Released: March 1998
2. *"In Dreams"* by Roy Orbison (Country rock, oldies) | Released: February 1963
3. *"Round & Round"* by New Order (Electronica, New Wave) | Released: February 1989

Qualitative & Quantitative Results For each of the seed songs listed above, I pulled more track data for returned tracks for each of the methods with the Spotify API. Through access to additional song data beyond the provided track content in the million song dataset, I am able to have a more consistent and quantitative approach to scoring each method's returned ten track playlist. The track information that I utilized from the Spotify API is listed below:

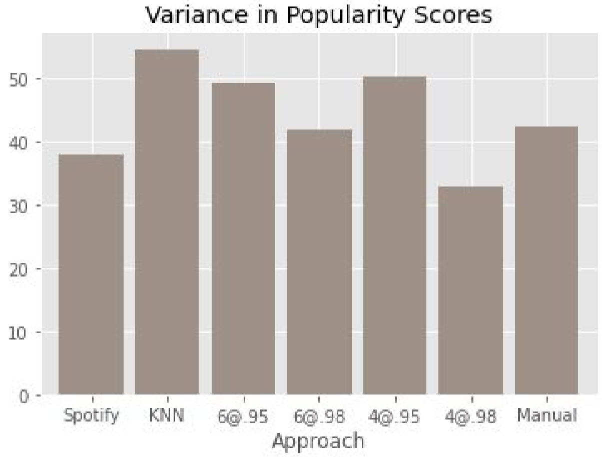
1. Popularity – On a scale of 0 to 100, the popularity ranking score of the track on Spotify.
2. Danceability – On a scale of 0 to 1, a Spotify provided metric for one's ability to dance to a track.
3. Energy – On a scale of 0 to 1, this measures intensity and activity. Spotify gives the example of Black Metal (1) vs Bach transcriptions (0) – could correlate highly with genre
4. Key – The estimated primary key of the song using pitch metrics.
5. Loudness – Overall loudness of the track measured in decibels (dB).
6. Speechiness – Measures the presence of spoken word in the track on a scale from 0 to 1.
7. Acousticness – On a scale of 0 to 1, a confidence measure of the track's acoustic content.
8. Instrumenalness – Scoring between 0 and 1, intended to measure if track contains vocals or not.
9. Liveness – On a scale of 0 to 1, a confidence measure of if an audience is in the recorded track.
10. Valence – On a scale of 0 to 1, valence measures tone and positiveness where 0 is more sad and 1 is euphoric.
11. Tempo – Overall estimated tempo of a track in beats per minute (BPM).

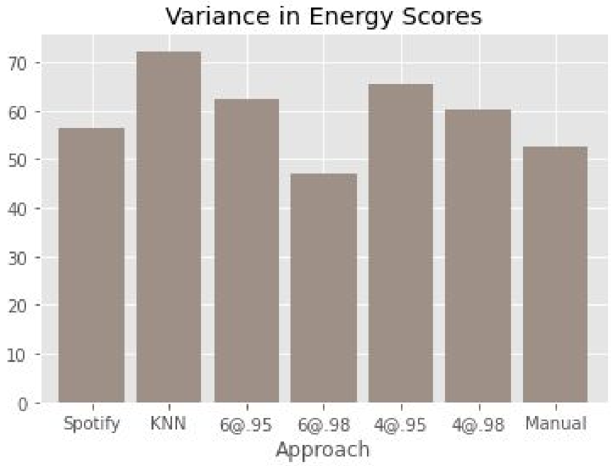
12. Mode – Modality of track's melodic content: 0 - Minor, 1: Major

Using these metrics, I calculated the variance for each method's returned playlist. Through analysis of variance within the playlists, we can add quantitative perspective to the evaluation of these methods of playlist creation. Since the goal of the project is to explore and improve upon the small amount of diversity in Spotify's playlist makers, the methods' variance scores are measured against the first ten tracks of Spotify's "Track Radio."



Quantitative Overall Among the utilized metrics provided by the Spotify API, popularity, energy, acousticness and valence were especially interesting to compare and understand each method's performance in track similarity compared to one another. While there were twelve separate metrics to observe, for the sake of time and conciseness, I have selected four of the most interesting variance comparisons among the methods to outline and explain. The graphs show below were created by averaging the variance among the three selected seed songs' playlists. I also created analyses for the variance between the seed songs to see the differentiation based on input track.

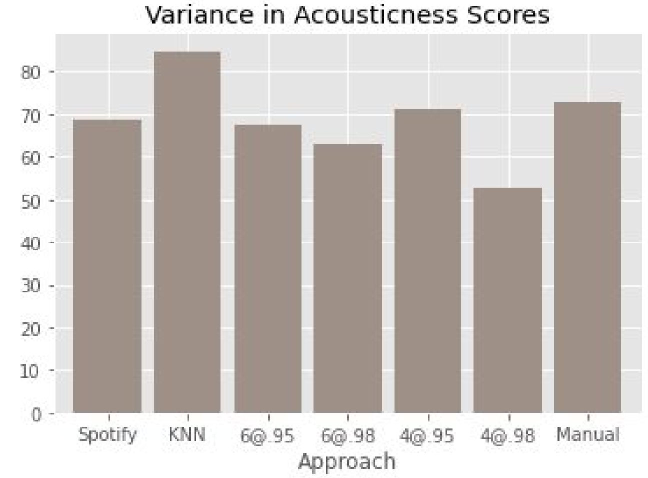




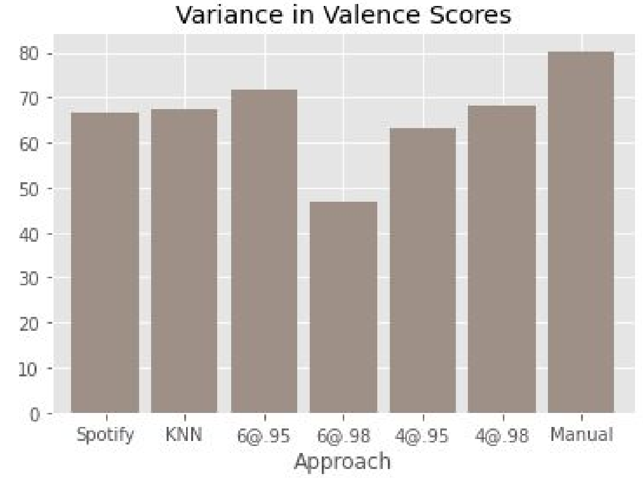
While the baseline tracklist from Spotify was expected to score the lowest on all metrics' measures of variance, Popularity scoring was one of the categories where the tracklist had the least amount of variety in comparison to the other methods. Both KNN application to the track network and the 6 topic LSI

With the interpretation of energy score translating relatively cleanly to genre grouping, both Spotify and the Manual playlist creation (utilizing song tags and tracks listed as similars) score very low on diversity and variance. While this was qualitatively understood, it's pretty validating and interesting to see it appear model that sampled above .95 performed the best. The 4 topic LSI method sampling at a .98 similarity score is also performing very poorly for diversity and variance within popularity of songs. This is reflective of the high similarity score point of 98% as well as with the selection of only 4 topics (more limited approach to the index)

visibly in quantitative scoring. The KNN application in the network method performs the best which makes sense because the nearest neighbor lists are being formed from nodes with similar positions and identities within the graph, who are not necessarily connected to one another (i.e. a genre breaking approach).



Throughout these selected bar graphs, we can clearly see that including less perfect similar scores for the LSI methods increases overall variety in the output playlist. This is again true for acousticness, but we see that the Spotify baseline playlist and the manually selected playlist outperform the LSI methods for variation in this category. This category represents the other scoring metrics such as danceability and liveness which work as confidence scores invented by the Spotify team. The results for this category were definitely interesting. As a listener, I would generally associate specific genres with more acoustic or less acoustic track options, so it is very interesting to see this come through. KNN again is a top performer – prioritizing diversity over similarity and breaking the consistent genre rule of traditional recommendation systems. The only question will be if the KNN approach is producing a playlist that is too diverse from the seed track.



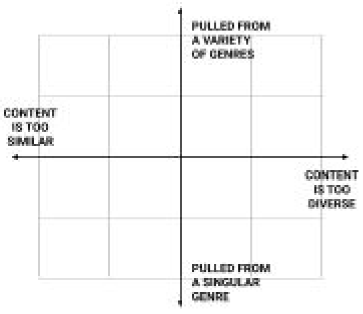
Valence, which is meant to measure a general mood of a track, is another interesting variance metric to observe. Based on the selection tactics of the manual method, I really expected to see the lowest variance for valence there. Instead, we see it as the top performance method for variation in this category. Because this was so different from what I expected to see, it prompted me to return to exploration and trial phases by returning to remove the top 100 genre tags from the original dataset (this step is listed above in methodology). The main motivation was to take out some of the largest genre tags from the dataset inorder to favor track tags about emotional content. Still, this did create a dramatic drop in variation for the manual method.
We again see the LSI methods perform as expectedly for mood-centric variation being relatively low for these methods since they work off of semantics and subjects. KNN and LSI 6 @ .95 again outperform the Spotify baseline on variation here.

Quantitatively, overall, the top performers for providing a ten track playlist that offers variation and diversity are the KNN model based on the Track Network as well as the Latent Semantic Indexing with 6 topics, sampling at .95 similarity scoring. Overall, I expected the KNN to provide the most diverse offering, but I expected it to have dramatically more variance than the baseline Spotify because it is largely working off a model that does not weigh or consider genre. However, in the results of these quantitative analyses, I was pleasantly surprised to see that it only slightly outperforms other methods, including the baseline. The next level of verification of the methods' results will be to listen to each created playlist and see if the KNN model provides just enough variation or too much variation.

Additionally, I am pleasantly surprised by the performance of the LSI model with 6 topics. Early on in the work of this project, I felt pretty confident in the performance and ability of KNN to deliver, but the LSI model was an exploration that worked out better than expected. Through analyzing and tuning the amount of topics included in the model, I was able to understand and improve the performance step by step. Additionally, I made the decision to pull samples from the returned tracks with similarity scores of a certain threshold, like .95, instead of returning the top 10 scoring similar tracks. In my trials, this shift to added randomization improved the models' overall ability to work outside of genre lines and improve variation scores.

Method	Quantitative Analysis	Overall Qualitative Analysis
Baseline: Spotify "Track Radio"	Lowest variance scores; most significantly for variance in popularity	No genre diversity, not enough content diversity
Manual Selection	Low variance scores, for some measures, lower than baseline method	No genre diversity, Slightly more content diversity
KNN on Tracks Network(G) (top - performer)	Top scoring for variance in the majority of 12 measures; potentially includes too much variance for happy listening	Best for genre diversity, Content diversity, borderline too much
LSI: 6 topics sampled at .95 sim (top - performer)	A top scorer for variance, compared to other methods and Baseline Spotify method	Genre diversity, Best for content diversity
LSI: 6 topics sampled from tracks matching with at least .98 similarity score to seed track	Low variance for valence especially	Not enough genre diversity, A small amount of content diversity
LSI: 4 topics sampled from tracks matching with at least .95 similarity score to seed track	Decent variance scores for most measures	Some genre diversity, Varying amounts of content diversity
LSI: 4 topics sampled from tracks matching with at least .98 similarity score to seed track	Low variance for popularity and acousticness and other confidence measures	No genre diversity, not enough content diversity

Qualitative Overall Spending about 5 hours per each seed song, I was able to sit down and qualitatively analyse each method's recommendations by listening to the resulting 10 track playlists. Ultimately, it will be the users' feedback that would validate the success of each of these methods, so this was an important piece of evaluation to understand overall performance and recommending a best fit model. In an effort to document differentiation, I scored each playlist for each seed song on a double axis graph:



While there was some variance in the scores received for each method for each seed song, the overall overlap of scoring and qualitative notes on each method's playlist had a lot of overlap that only reinforced some of the findings and concerns with the quantitative scoring. For both the Roy Orbison and New Order tracks, the KNN model on the Track network delivered on genre diversity and the right amount of variation in track content. However, for the Third Eye Blind track, this method produced a playlist that delivered on genre diversity but that provided too much content diversity to work well.

Another top performer, consistent with quantitative scoring, was the LSI 6 topic model which sampled from tracks matching at .95 similarity and above. This was an easy best method for the Third Eye Blind song -- providing both genre diversity and the right balance of content similarity and differentiation. This method also did very well for the Roy Orbison track throwing in tracks that definitely fit well without being an obvious addition, such as a Misfits song. Qualitatively, as a listener, I am able to experience and understand how each of these playlists deliver on genre and artist content as well as overall mood, tone and potentially subject matter via lyrical content. However, qualitative analysis is more of an art than a science, and it's important to understand that each listener may pull out different key nuances and prefer different amounts of matching. That being said, the Spotify playlist, searching as a baseline underperformed in the qualitative scoring with less than desired genre blending and too consistent of content. For both of the LSI models that sample tracks at and above .98 similarity scores to the seed scores, we see too heavy of a correlation in a genre and not enough diversity in the returned playlists. However, I was surprised to see that the manual method playlists scored similarly to the Spotify playlist in not enough genre diversity but it did outperform this baseline model in content diversity.

5 Discussion and Conclusion

Interpreted Outcomes All in all, I would select the LSI 6 topic model which samples songs which match the seed track with a similarity score of .95 or greater. This was an obvious front runner in evaluation, both quantitatively with the Spotify API track data as well as quantitatively as a listener. While we did see high performance from the KNN model on the Network Analysis of the seed track, there was reason to believe this method would potentially raise the risk of recommending playlists that were slightly too different in content to be enjoyable to the listener. The LSI 6 topic model selecting playlist tracks with a similarity score of .95 or higher outperformed the Spotify “Track Radio” baseline playlist in both genre diversity and appropriate content diversity. The research that fueled the experimentation in this project was *How much variety is appropriate variety in a song recommendation system?* Through the methods and approaches I used to try to solve this question, I have a much better understanding for the limits of appropriate variation. While some models provided too much variation, like the KNN model, plenty of the LSI models outperformed the Spotify baseline for diversity in genre content as well as overall contextual content of the tracks for the first ten tracks based on a seed track. Based on this experimentation, I think it is safe to make the statement that, for an audience that is looking for a more unique and creative listening experience, there are plenty of scoring and ranking improvements that could be made to improve the variety of a recommendation system.

Learnings Some of the limitations on this assignment were due to time constraints. In terms of processing power, my computer could not handle the entire dataset I had planned on using. To counteract this, I sampled the main dataset and created a subset dataset of around 10,000 tracks. From this track list, I found my seed tracks and proceeded to do analyses from here. While this decision dramatically improved my ability to make progress on this project, it had implications on the roll out and solutions of some of the methods I tested, both those included in the paper and those that did not work at all. For example, when using the Networkx library to build a graph from a node list based on the subset data, I struggled to see any connectivity. To counteract this, I compacted the dataset for that step a bit further, removing some of the disconnected nodes to improve scores for the KNN model in the Network method.

Similarly, recommendation systems in use in the world today work from a hybrid layout of both track similarity as well as from liked tracks or preferences from individual users. Because of this, the baseline ten track playlist being used for each of the evaluations of methods has some margin of error since I accessed these lists through my own personal spotify account with years of listening history. To counteract this effect, I tried to choose seed songs that would be both familiar but also not something that would highly correlate to highly or most recently played.

Future Scope Overall, this project was a lot of fun to work on. If I had more time, the obvious add or change I would make to the structure of this project would be to build a system that could allow for user feedback which could help improve the model. In the same way, the addition of more metadata could potentially improve the methods included and expand from there. For example, if each track had lyrical information included in the data, the LSI models would improve on their genre-centric pulls to complete a more mood or vibe recommendation which would be very cool to see. Similarly, I would just like to have a timeframe that allowed for users beyond myself to interact and rate each method's responses to the question.

6 References

Jannach, Dietmar & Kamehkhosh, Iman & Bonnin, Geoffray. (2018). Music Recommendations: Algorithms, Practical Challenges and Applications. 10.1142/9789813275355_0015. [PDF]

For more information on the Million Song Dataset, please visit their website.

The Python Libraries used throughout this project include the following:

- Matplotlib & Seaborn for Visualizations
- Pandas, Numpy and SkLearn (including KNN from SKLearn implementation)
- Networkx for Track Network creation and analysis
- NLTK and Gensim for Topic Modeling and NLP Text Preprocessing

7 Appendix

Appendix A – Example of Track Data in Original Dataset

Example Dataset Track Entry	
Artist	Third Eye Blind
Timestamp	2011-08-03 19:25:53.648756
Title	Losing A Whole Year
Track_id	"TRXEFKY128F147C7C0"
Similars	[['TRLGTNS128F147C7C3', 1], ['TRVVRIZ128F147C76 .6]]
Tags	[['alternative rock', '100'], ['rock', '66'], ['alternative', '66'], ['90s', '66'], ['one star', '33'], ['bpluscoop', '33'], ['woocoop', '33'], ['aitch', '33'], ['4 Stars', '33'], ['Pop-Rock', '33'], ['good songs', '33'], ['pop', '33'], ['indie', '33'], ['Usual', '0'], ...