<table>
<tr><th>Slide 1</th><th>Slide 2</th></tr>
<tr><td>NONE</td><td>In this presentation, I would talk about the current problem with music recommendation system, prior research, our proposal, datasets and the result.</td></tr>
<tr><th>Slide 3</th><th>Slide 4</th></tr>
<tr><td>
Major music recommendation
Spotify
Pandora
Google
(slides) As you can see, Current music recommendation relies on music’s meta data
Songs from popular artists are more likely to be recommended
So, it is becoming increasingly difficult for new artists to gain popularity with current music recommenders..
Limits user’s ability to discover new songs.
Leads unfair result for unknown artists.

In first 8 albums, artist has 16million monthly listeners on average -> they are popular
It is obvious why unsupervised recommendation system recommend popular songs.
If one artist is popular already, new songs by that artist will be more likely to be recommended.
Induces unfair result for unknown artists.
user limited ability to discover new artists.

</td><td>Problem is especially serious for unknown artists.

Left: Shows the distribution of artists.
Right: Shows the distribution of sales revenue

It is clear that very handful of popular artists dominates the music sales

The top 1 percent of bands and solo artists now earn 80 percent of all revenue from recorded music in 2015.

There is a need for music recommendation system that can analyze music without being influenced by meta data.</td></tr>
<tr><th>Slide 5</th><th>Slide 6</th></tr>
<tr><td>NONE</td><td>This paper was one of the first successful example of using the music audio content as a factor to determine what song to recommend. It uses 30 seconds clip of music as well as metadata to predict user’s preference of songs, using boolean value.

Prior to this, the most common method for recommenders was either to rely solely on metadata or to use a traditional audio content feature such as Mel-frequency cepstral coefficient, which is known to perform poorly.

This paper was able to predict whether or not user will like the song with Root Mean Squared Error of 0.255.</td></tr>
</table>

Slide 7	Slide 8
This paper took a brilliant approach to predict the users' preference for new songs. Because this only looks at music raw data, the result of this recommended system does not depend on any metadata, resonate with our goal What is done: • (slides) Network categorizes songs based on raw audio data into clusters - (slides) • (slides) When a new song is introduced, it is classified based on the trained model - (slides) • If the distance from a cluster of 'liked' songs is small, it is likely the user enjoys that song	(slides) Introduce LSTM LSTM is a type of recurrent neural network 1. RNN is a neural network with loop in it, allowing information to persist. Therefore, it is able to use what it has learned to understand something new, just like how human brain acts. 2. However, a traditional RNN has difficult time 'remembering' past information for long time. That's why LSTM: LSTM is designed to remember for longer time than other types of neural network. Because of that, it is well suited to analyze the series of data, making LSTM popular tool for audio analysis Recent success in speech recognition or translation are due to this model. powerful
Slide 9	**Slide 10**
Applying these researches, we came up with unique 'potential' solution for music recommendation system. Given the problem with current music recommendation system, we wanted to research if raw song data itself is useful for predictions. Because if that is possible, one could create a recommendation system that would not be influenced by metadata or popularity. In addition, what distinguishes this project with majority of recommender system is that we would use LSTM and regression model to predict how many times users are likely to listen to a particular song. Slides We believe that taking this approach can be more useful for the industry, such as music artists and record companies, because their interests are how many time the song is likely to be listened to. And that is what their profits come from.	NONE

<table>
<tr><td>Slide 11</td><td>Slide 12</td></tr>
<tr><td>

Million Song Dataset

1,000,000 songs
Metadata: artist, song name, etc
Features: key, tempo
NO GENRE

Echonest Taste Profile Subset

~ 350,000 songs from Million Song Dataset
Over 1,000,000 users
Records have User ID, Song ID and Play Counts

Million Song EchoNest Mapping Archive

Echonest API is deprecated
Maps Echonest Song ID to Spotify & Other services

</td><td>

Data Selection Process

Records were limited to 200 plays
Outliers w/ 1000s of plays per song
Top 5 users selected by total number of plays
Only songs listened to by at least 2 of the top 5 users were selected

Reason for reducing the dataset to 150 songs:

Computation time
Allotted disk space

</td></tr>
<tr><td>Slide 13</td><td>Slide 14</td></tr>
<tr><td>

Individual user data -> Model for each user
scikit-learn was used to standardize the num_plays column

</td><td>

PyDub ➔ Slice song into 5s clips
LibROSA ➔ Load songs as numpy array
Next talk about models & results

</td></tr>
<tr><td>Slide 15</td><td>Slide 16</td></tr>
<tr><td>NONE</td><td>

Input shape of LSTM was determined by parameters passed to network
Each LSTM cell has 50 neurons
Single layer LSTM to fully connected dense layer w/ single output
Two layer LSTM: LSTM to fully connected LSTM to fully connected Dense layer w/ single output

</td></tr>
</table>

<table>
<tr><th>Slide 17</th><th>Slide 18</th></tr>
<tr><td>

Data:
Numpy vector of mono audio clip
Time steps:
Number of clips of 5s data to consider for history in LSTM
Training Samples:

Left side:
3 songs
1 time step per song
Right Side
3 songs
3 time steps per song

</td><td>

User 1 was the worst user for all models
Possibly too much variance in data or targets?
Possible outlier in training
For all other users
All show either stability or clear decrease
MSE was under 0.4 std deviations

</td></tr>
<tr><th>Slide 19</th><th>Slide 20</th></tr>
<tr><td>

All models for User 2 performed fairly well
The final MSE ~.82 at highest ~.14 at lowest

</td><td>

No model performed universally well for ALL users
2 LSTM Layers with 18 steps appeared to be the most ‘universal’ model
Model still has to be trained per user
Possibility of online learning to update weights as user adds/listens to new songs
This concludes our presentation. Any questions?

</td></tr>
<tr><th>Slide 21</th><th>Slide 22</th></tr>
<tr><td>DEMO</td><td>NONE</td></tr>
</table>