1. **Table of Contents**
    - **a.** The presentation will be held in the following order starting from a brief introduction of our project into a discussion about our model and the results that our project produced.
2. **Project Introduction**
    - a. Our project's goal was to develop a music recommendation system that recommends music based on sentiment analysis of a user's KakaoTalk profile status message. We have since slightly changed the scope of the project from KakaoTalk status messages to a user input system due to feasibility issues. The primary motivation for our project was the prevalence of social media and messaging applications. The method we used was a collaboration with sentiment analysis of a given text and a predetermined list of songs and the cosine similarity of their softmax probability values calculated by our model.
3. **About the Team**
    - a. Our team is composed of 5 members of which 4 were contributing towards model development and one member solely focused on delivering progress report presentations. The role each member played in this project is listed as follows..
4. **Goals and Novelty**
    - a. As mentioned before, our project's goal was to develop a system that can accurately recognize emotion and recommend songs to the appropriate emotion. We also planned to develop a web application for our model to be run and used by others. We believe that the novelty of our project lies in its ability to recommend songs based on the distribution of probability factors for each emotion label. Other existing systems utilize only the emotion label, such as happy, sad, or anger, to provide a recommendation. However our system provides recommendations based on the distribution of emotion probability values. This means that our system can much more effectively express the nuances of emotions in the recommendations because although one can be happy, the reason, intensity and quality of happiness are variables that play into the classification of "happy." Therefore, we believe that our system reflects the nuances in emotion with its recommendations. Additionally we believe that through the labeling of songs and fine-tuning based on user feedback, song recommendations were made standardized
5. **Architecture: Model**
    - a. For the model, we used the pretrained model, KoELECTRA, and fine-tuned it on our dataset. No major changes to the pretrained model's architecture were made.
6. **Architecture: Web Application**
    - a. For the web application system, we developed a flask-based EC2 server. This server runs a saved, trained model to perform sentiment analysis. The following figure depicts the high level overview of our web application. User input is taken and sentiment analysis is performed on the input. The cosine similarity is found between the analyzed emotion and a list of songs which then provides an appropriate recommendation to the user. A more in-depth explanation will be given in the following slide.

7. **Architecture: Web Application**
    a. On the front-end side, the user will see three fields, name, phone number and current mood. With this inputted information, the fine-tuned trained model performs sentiment analysis on the current mood input. It then predicts the user's emotion probability distribution based on seven predefined emotions. An RDS-based database stores the predicted emotion distribution based on lyrics of about 7,500 songs. Songs are retrieved from the database based on the user's predominant emotion distribution. Predominant emotion distribution refers to the emotion label that has the highest softmax probability value. The server then recommends the most similar song to the user. Afterwards, user feedback on the recommendation they received is taken.
8. **Dataset**
    a. We utilized two different datasets. One dataset, the primary training dataset, was used to train and test our model while the other data, the music dataset, was used for the music recommendation system and for fine-tuning our model. We acquired the music dataset to further augment and fine tune our model after we deployed the system and received user feedback. More on this will be explained in the following slides. The training dataset was sourced primarily from AI-Hub and we utilized a compiled version of three datasets, Emotions in Continuous Korean Conversations Dataset, Emotions in Discrete Korean Conversations Dataset, and Audio of Korean Conversations for Sentiment Analysis Dataset. The music dataset was sourced from Melon using a web crawling tool to gather song information from a collection of playlists from Melon's May 15th ~ 21st popularity chart.
9. **Dataset: Preprocessing (training dataset)**
    a. We took four steps in preprocessing the training dataset. First we compiled the three datasets into one file. Then we normalized the language of the labels to korean. Next, since these datasets were manually created, we performed typo correction to correct any human error that could have occurred. We then performed label normalization by choosing 9,000 random entries for all the labels to prevent data skewing. The value 9,000 was chosen due to the least occurring label having 9,697 entries.
10. **Dataset: Preprocessing (music dataset)**
    a. For the preprocessing of the music dataset we took three steps. First we excluded lyrics that contain English words not present in the trained data. Then we treated 4 lines of lyrics as 1 sentence. Then we calculated the emotion probability for all **sentences** in a song then averaged the values for said song. The figure on the right depicts the structure of the music dataset file post preprocessing

11. **Model**
    a. For the model we utilized a pre trained KoELECTRA model. We made numerous changes throughout the project to end up with our final, fine-tuned model. The table shown in the slide compares the base model and our final fine-tuned model.

The base model used a KoELECTRA-Base pretrained model, trained on 13,500 entries, had a test accuracy of 57% and suffered from poor performance in large part due to overfitting. The fine-tuned model used a smaller pre trained model, KoELECTRA-Small, and trained on 45,000 entries. This model exhibited a test accuracy of 74% after applying dropout and L2 Regularization. The large increase in our final fine tuned model depicts the addition of our music dataset.

12. Model: Performance

a. As the base model as our starting point, we explored various ways to improve our model's performance. We changed the pretrained model size, utilized K-fold cross validation, and tried to alleviate the rampant overfitting present in the base model through dropout and L2 regularization. As a result, our fine-tuned model exhibited much greater performance compared to our base model.

13. Model: Evaluation

a. As stated before, we concluded that the base model exhibited poor performance due to overfitting. By using a smaller pretrained model and cross validation, we ensured dataset diversity and improved the model's generalization performance. Additionally, by applying dropout and L2 regularization, we further alleviated the problem with our model overfitting on the training data and also improved its generalization performance

14. Model: Observations

a. Throughout our journey of trying to improve the performance of the model, we made some key observations. Large-scale pretrained models such as KoELECTA-Base, do not scale well with a small training dataset size. Cross validation, dropout and L2 regularization are useful methods in improving a model's performance, but cannot completely eliminate overfitting. Finally, we saw that any further improvement to our model would require changing its architecture, training on more data and receiving much more feedback data to fine-tune the model even more.

15. Music Recommendation

a. As stated earlier, we utilized the Melon dataset for our music recommendation component. The recommendations are produced by retrieving songs that have matching labels with the user input. For example, if a user's input is classified as "happy," although it has other emotion probability values, songs from the "happy" list are retrieved. Then the cosine similarity between the emotion probability values of the songs and user text is performed. Finally, the most similar song, the result of the cosine similarity process, is recommended.

16. Music Rec: Performance

a. We were faced with a critical problem concerning the effectiveness and accuracy of our system's recommendations. Providing an objective result on the very subjective and nuanced topic of emotion is near impossible. How can we tell if the recommendation that was given to the user was appropriate? We decided that collecting user feedback about their recommendations was the best course of action in trying to address this problem. Our system received feedback from 100 users regarding the satisfaction of their recommendation. The first iteration of our model trained on 30% of our dataset had a satisfaction rate of 34%. The second iteration, trained on the entire data, exhibited a user satisfaction rate of 51%. Lastly, the fine-tuned model exhibited a satisfaction rate of 67%.

17. System in Action

a. The following video shows our application in practical use.

18. Accomplishments

a. We believed that we accomplished three things with our project. First, we successfully utilized sentiment analysis to ultimately provide accurate music recommendations that users were satisfied with. Next, our system provides music recommendations that are correct more than 50% of the time. Finally, we were successful in creating a minimum viable product, or at the very least a working prototype that could be iterated further to deliver a much better system.

19. Limitations

a. However we do recognize some limitations we faced during our journey with this project. First, similarity in emotion probability values from user input and a song does not guarantee that the user will actually like the recommended song. Next, we realized that because emotion is such a nuanced and subjective concept, it is near impossible to objectively analyze using only similarity values. Additionally, we feel that a fundamental re-evaluation of our current method of measuring similarity is required to improve the performance of our system. Finally, a limited size of the music dataset inevitably means a limited range of recommendable songs, which can only decrease the performance of our system.

20. Future Directions

a. We believe that a dedicated processing unit is required to further advance our system's performance. We conducted this project solely relying on the specifications provided by the free version of Google Colab. As a result, we faced lengthy training times ranging from 1 to 12 hours. Next, we believe that changing the means of measuring similarity must also be considered in order to further improve performance. Lastly, adjusting the model's architecture would be necessary to further improve the performance of our model.