Musical Robot

Amre Abken, Olga Dorabiala,
Katie Johnston, Raphael Liu

Background

- There is high interest in working with audio data.
- Music Information Retrieval (MIR) field has made much progress with analysing music data, e.g. ‘Librosa’ python package.
- We want users to learn more about the music they like. Given a music file,
    - What is the genre of the song?
    - What are other popular songs in that genre?
    - What other genres are similar to the genre of the uploaded song?

Result: Users can discover new music that they like easier.

Data

Free Music Archive: A Dataset For Music Analysis

Source: https://github.com/mdeff/fma

- fma_small.zip: 8,000 tracks of 30s, 8 balanced genres
- tracks.csv : per track metadata such as ID, title, artist, genres, tags, and play counts
- genres.csv: all genres present in data

Use Cases

1) User has an audio file. User wants to know the genre of the music file, as well as discover music of similar genres.

    a) The user uses the Musical Robot UI to upload their audio file, predict the genre, learn the most popular song in the genre, and learn what the other most similar genres are.

Use Cases

2) User wants to predict the genre for a batch of audio files using a pre-trained ML model and return an accuracy report. The user wants to be able to query the datasets. User knows how to operate a Jupyter Notebook and use Python.

    a) The user can follow the tutorial in GenrePredictionTutorial to learn how to interact with both the ML Model and the Data Manager to predict genre from music audio.

    b) The user can use the functions in the Data Manager to create and query dataframes.

Use Cases

3) The user has a fundamental understanding of machine learning and wants to train their own Support Vector Machine for music genre prediction. The user knows how to operate a Jupyter Notebook and use Python.

    a) The user can follow the tutorial in TrainSVMTutorial to learn how to interact with the Data Manger and the ML Model to train their own SVM and return an accuracy report.

Design- Data Manager

- Provides functionality to load the data as pandas dataframes, split the data into training, test, and validation sets, create a custom audio feature dataset for ML model training, and query subsets of the data.

1) All users use the Data Manager to create the dataframes used for music audio analysis.
2) Users can ask the Data Manager to return datasets to train a model on. The system splits the data and create the Audio Feature datasets.
3) Users can query the dataframes using the features in the Data Manger.

Design- ML Model

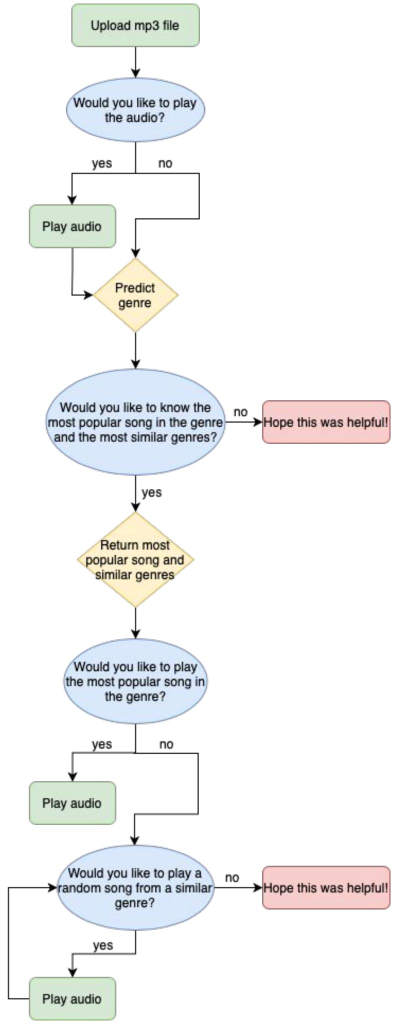
- Run a Support Vector Machine on data to predict genre and provide an accuracy report.

1) Use the Data Manager to create audio analysis dataframes and custom Audio Feature datasets.
2) Predict the genre of audio clips using an ML Model and optionally return an accuracy report.

Design- User Interface

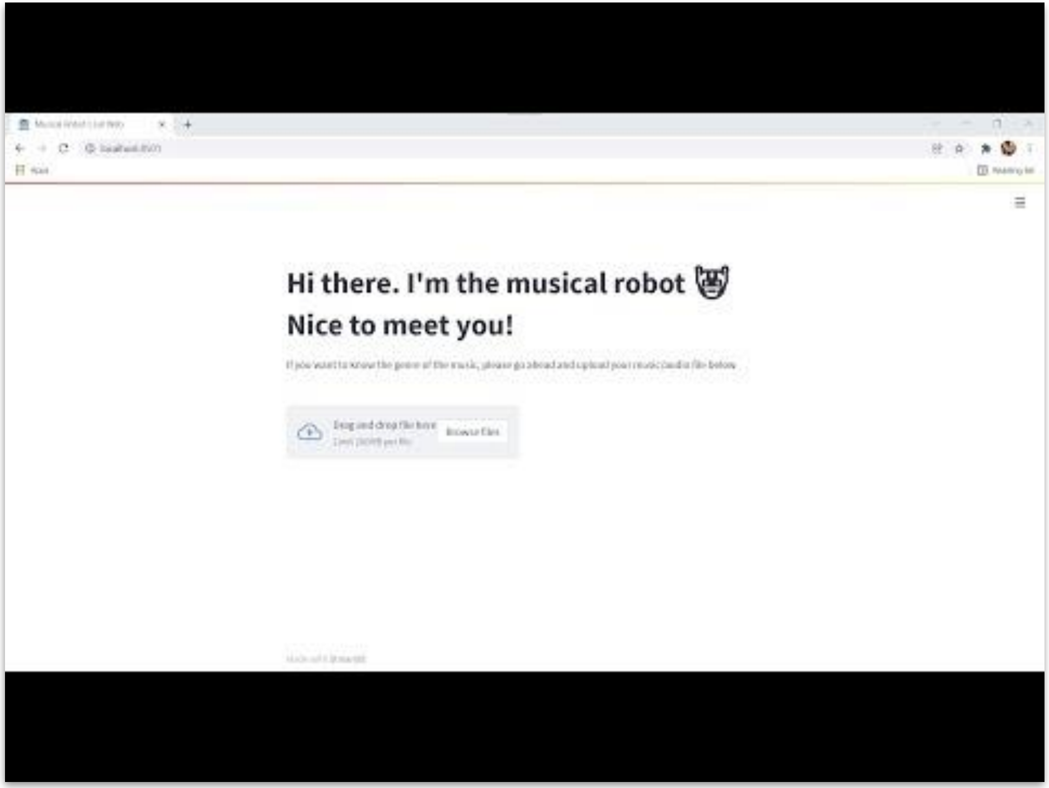
- Implementation of the musical robot, which the user can interact with.

1) Predict genre interacts with both the Data Manager and the ML model in the background.
2) Return most popular song and similar genres interacts with the Data Manager.



Upload mp3 file
Would you like to play the audio?
yes
no
Play audio
Predict genre
Would you like to know the most popular song in the genre and the most similar genres?
no
Hope this was helpful!
yes
Return most popular song and similar genres
Would you like to play the most popular song in the genre?
yes
no
Play audio
Would you like to play a random song from a similar genre?
no
Hope this was helpful!
yes
Play audio

Demo



Hi there. I'm the musical robot 🤖

Nice to meet you!

If you want to know the genre of the music, please go ahead and upload your music (audio file) below

Project Structure

```
.
├── LICENSE
├── README.md
├── docs
│   ├── Technology\ review.pdf
│   ├── component_specification.md
│   └── functional_specification.md
├── environment.yml
├── musical_robots
│   ├── __init__.py
│   ├── data
│   │   ├── fma_small/
│   │   ├── fma_metadata/
│   │   ├── genre_df
│   │   ├── svm_model.pkl
│   │   ├── total_genre_df
│   │   └── track_df
│   ├── dataset_queries.py
│   ├── demo.py
│   ├── session_state.py
│   ├── spectrogram_dataset.py
│   ├── svm_prediction.py
│   └── tests
│       ├── __init__.py
│       └── tests.py
└── examples
    ├── GenrePredictionTutorial.ipynb
    └── TrainSVMTutorial.ipynb
```

- /docs/: Contains project documentation.
- /musical_robots/data/: Raw data that was used for training the ML model and data containing track and genre information.
- /musical_robots/demo.py: Musical Robot User Interface
- /musical_robots/tests/: Unit tests.
- /examples/: Tutorial notebooks for SVM training and genre prediction.

Lessons Learned

- This project is mainly developed in collaboration with Github, Python, and Jupyter notebook.

1) Set up Github repo, environment configuration, and package structure
2) Familiarize intermediate collaboration with Github in team
3) Gain insights for technology review
4) Develop unit tests and integrate Travis CI for git repo
5) Design interactive user interface using Jupyter notebook & Streamlit
6) Utilize SVM Model for classification tasks

Future Work

- This project can be further improved in several directions to promise better user experience. These include but are not limited to: visualization, more features, web development, and prediction precision.
  1) Visualization: bring charts like percentage of genre classification into user interface
  2) Features: include features like given multiple audio files the music robot predict their genres in the same time.
  3) Web Development: deploy the streamlit app to cloud provider, which might include Heroku, AWS.
  4) Prediction precision: some ideas include letting the ML-model train more audio clips (currently 8,000) and classify more detailed genres.