Project Proposal Draft Outline

### Abstract

Automating the classification of music genres based on sound data has a potential to be applied to many music labelling and streaming companies for indexing and recommendations. The team aims to design and train a neural network model that can accurately classify a song’s music genre. The project plan requires the following steps: data collection, data processing/cleaning, feature extraction, model design and construction, training, validation, and testing. The type of network architecture to be applied is the Convolutional Neural Network (CNN). CNNs have been successful in pattern recognition and are often used for speech recognition, lyric transcription, and image recognition. The level of accuracy in the model will be determined using the frequency in which the network classifies the song according to previously known and accepted genre labels. Furthermore the neural network will be compared to a baseline model of nearest neighbour classification. The CNN must outperform this baseline model to a considerable degree to be considered successful. Some risks associated with the project include biased data, overfitting, and copyright issues, however these will be mitigated according to certain contingency plans. In addition, ethical considerations such as informed consent and privacy, have also been considered. The team has broken down the steps to constructing an accurate model, and associated each step with internal and formal deadlines. The project’s outcome will be a trained CNN model that has the ability to accurately classify large sets of audio data into music genres.

### 

### 1.0 Introduction (Tyler)

With the increased popularity of music streaming apps and new artists, as well as the number of published songs has been increasing exponentially [<https://ieeexplore.ieee.org/abstract/document/9489022?casa_token=J84bkD1Pmp0AAAAA:YJtvdJnl3eHMre-Pk20VJ6JN5oXwZN3xY52CnD-xz-DhzVnv282Rq_hyKY0_2wTitBZw9K61jw>

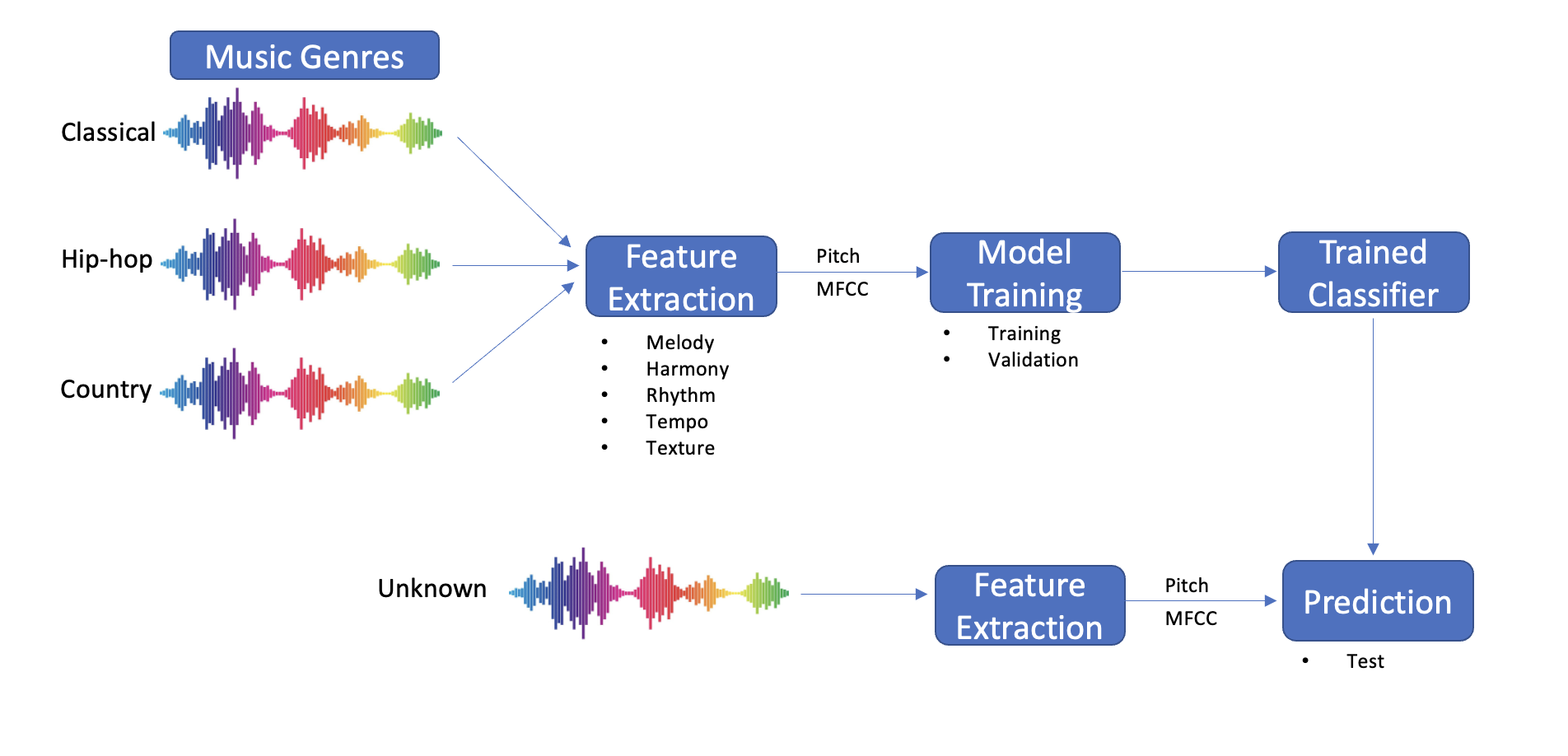
]. Companies such as Spotify or Apple Music would benefit from automated, and accurate music genre classification to organise and recommend their libraries better for their users [<https://www.sciencedirect.com/science/article/pii/S1877050919310646>]. The team’s goal will be creating a neural network that accurately classifies music genres from audio content. The milestones in this project include collecting data, building and training the neural network, optimising its parameters, and reporting on its accuracy. The team is interested in the project since all team members have some connection to music such as learning an instrument or are frequent listeners. Deep learning will improve user experiences on recommendations, and organisation of music, in addition to being faster and able to handle large amounts of data. While artists may classify their own music as belonging to a genre, a deep learning model can provide other genres a song may apply to, or classify them more specifically; especially in ambiguous genres such as Indie.

### 2.0 Illustration/Figure (Blythe)

“A well thought out model that communicates the core idea of your project and architecture immediately”

Figure must:

* Illustrate overall model/idea for project
* Aid in making report accessible

\*can use hand drawn illustration or powerpoint 

### 3.0 Background & Related Work (Anna)

“Briefly describes 1-2 prior work related to your project to put your project into context. Your descriptions need not be complete but should contain important work related to the project”

* Most familiar with applications like Shazam that are able to identify a song based on a sample of music
  + Genre recognition is more complex and can be subjective
* Music Genre Recognition is an important part of the research field related to Music Information Retrieval (MIR)
* Also a form of music label classification; other forms include:
  + Writer Classification
  + Emotion Classification
  + Region Classification
  + Applicable Scene Classification
* Potential Applications of music genre classification include:
  + Storage management
  + Search engine
  + Recommendation systems
  + Music creation
* Research is predominantly conducted using MP3 and WAV audio files
* Related work 1:
  + What is it?: Used multilayer classification system using Support Vector Machine (SVM) learning
    - Feature selection included beat spectrum (rhythm and tempo)
  + By Who: Xu, Maddage, et al.
  + When: 2009
  + Relevance: First application of a “deep confidence network” to music genre classification
* Related work 2:
  + What is it?: model used “two dimensional feature map of pitch and rhythm” to identify music genres”
    - Pitch and rhythm map helped to identify coefficients for feature vectors which could represent melody
  + By Who: McKinney and Breebaart
  + When: 2011
  + Relevance: had 81% accuracy

Music genre recognition is a significant subset of the research field related to Music Information Retrieval (MIR) [a]. While most are familiar with music identification applications like *Shazam*, which can recognize songs from short samples, genre recognition is a more complex and subjective process. Music genre classification is also a form of music label classification, with other forms being writer, emotion, and region classification, among others [b]. Some potential applications of music genre recognition include music organisation and storage management within databases, internet search engines, recommendation systems within streaming services, and tools for music inspiration and creation [b].

Since the 1990s, genre recognition has been studied by many data and computer scientists [b]. This work is predominantly conducted using MP3 and WAV audio files [b]. In 2006, researchers in Singapore developed the first application of a “deep confidence network” to music genre classification. This was done by using a multilayer classification system with Support Vector Machine (SVM) learning. One of the main methods of feature selection used was through using beat spectrum analysis, which automatically classifies rhythm and tempo [c]. To develop predictions, the research team utilized the theory that a subsection of a piece of music can be approximated by a linear combination of other related music samples [c].

In 2011, another group of researchers developed feature vectors for their genre recognition machine learning model through the use of a two-dimensional feature map which accounted for pitch and rhythm [b]. The feature vector coefficients generated were then used as approximations for melody [b]. This method of identifying feature vectors led to an 81% accuracy rate with model [b].

### 4.0 Data Processing (Alex)

“Clearly describes and cites sources of data, and the steps you will take to clean and format your data. The descriptions are clear enough for another classmate to follow and reproduce”

* Data Sources:
  + Where were they collected:
  + Description:
* Team Scratch Data Collection:
  + Collection Process:
  + Repurposing Process:
  + Cleaning Process:

The steps involved for our Music Genre Classification data processing are as follows:

1. Data Collection:

In order to produce a model that will output accurate information, we need a large dataset that has various classes of music and has already labelled elements. For these reasons, we will be using the GTZAN dataset, which is referred to as the MNIST dataset for music. This dataset is suitable because it contains 10 genres, each represented by 100 tracks. The tracks are all 22050Hz Mono 16-bit audio files in .wav forma. The dataset has various genres such as ‘rock’, ‘jazz’, ‘classical’, etc. This allows for easy splitting of the data in order to train our model. The presence of a variety of genres could potentially be useful when classifying songs that may fit more than one genre. An alternative dataset that was considered was th Million Song Dataset (MSD) but we decided against this dataset because it did not actually contain any audio files, but instead contained the metadata of the songs in the set. The file format was also in HDF5 which was slightly difficult to work with.

1. Data Preprocessing:

The data comes in audio form as .wav files. This is not suitable to be input into a model such as a Convolutional Neural Network (CNN). The means of addressing this issue would be to somehow visualize the audio files. This can be done by using spectrograms. Spectrograms are visual representations of the spectrum of frequencies of a signal as it varies with time. This would essentially involve applying a Fourier transform to the raw audio data, then dividing it into small time frames and applying a windowing function to each frame. The resulting spectrogram images are typically grayscale and can be used as input to a CNN. This can be done using PyTorch audio transform methods.

1. Augmentation and Normalization:

The dataset can be augmented to further extend the data used to train the model. This can be done by adding background noise, changing the pitch, etc. This is done in order for the training to be extended, improving the accuracy of the model. For example, the data can be split into 700 for training, 150 for validation and 150 for testing. The training set can then be augmented by changing a specific property of the audio files as previously indicated. After this, the data should be normalized to have zero mean and unit variance, which helps to improve the stability during the training process.

After performing these operations, the data is ready to be fed into the model and then the next steps would be to test its accuracy and grade it accordingly. The goal is to be able to test the model and have it perform well using data generated by the team. This would be in the form of recording one another playing an instrument or possibly using songs from one of our playlists. The model should still maintain a high accuracy when fed data that it has never encountered before.

### 5.0 Model Architecture (Anna)

“Rough description of the type(s) of neural networks that you will use, and the relevant components”

* Chosen Neural Network(s): Convolutional Neural Networks (CNN)
  + What is it: artificial neural network
  + How does it work:
    - Convolutional layers (each layer has a set of filters)
  + Why choose it?:
    - Convolutional layers are good for detecting patterns
      * The patterns detected could be rhythm, pitch, and other aspects of music
* Potential hyperparameters/layers:
  + ???
  + Number of layers
    - Number of filters, kernel size,
  + Learning rate
  + Epochs
  + Batch size

For the purposes of this project, the team will be building a music genre classifying model using a convolutional neural network (CNN). CNNs are types of artificial neural networks that contain convolutional layers. Each layer contains a set of filters which transforms (convolves) the data such that a prediction can be made as to what the data is depicting. CNNs are commonly used for image classification due to their ability to detect patterns. This pattern recognition shows promise for audio files, as the network’s detection capabilities could be used to detect rhythm, pitch, and other aspects of music. In order for the CNN to use the data, the audio files will need to be converted to images in the form of spectrograms through the use of Fourier transform [f]. An example of spectrogram of Blues music is provided below.

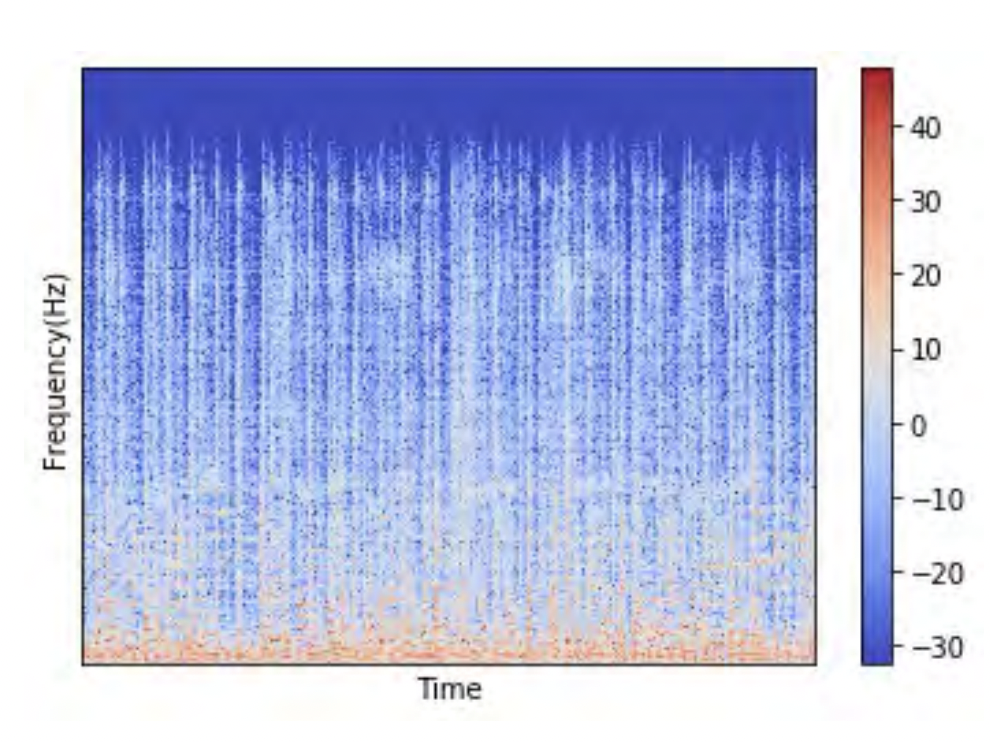


Figure 2: An example of the spectrogram of Blue’s genre. []

CNNs have been used frequently in the past for other music classification projects. For example, a study at Gazi University in Turkey used GTZAN data to train a CNN model, obtaining a testing accuracy rate of over 90% [e]. Further, a group of researchers in the Faculty of Economics at the Institute of Information Economics and Marketing (IISM) in Germany, were able to carry out the same task with the same data, with relatively similar results [f]. Research projects such as these indicate the viability of the chosen model.

### 6.0 Baseline Model (Anna, Blythe)

“A reasonable choice of baseline, accompanied by a description of the baseline so that a knowledgeable classmate can either find or reproduce”

* Model to compare neural network against (simple ML model, hand-coded non-ML heuristic model, or other):
  + KNN (K-nearest neighbour)

The goal of this project is to perform a classification task. As such, a baseline model by which the proposed model can be compared would be the K-Nearest Neighbour algorithm (KNN). KNN is used for regression and classification algorithm that separates data into categories. The theory behind the algorithm is that if a sample has the same characteristics as the majority of samples within a certain class, the sample must be part of that class[p]. This algorithm has been used to assess the capabilities of other music classification models[q].

The algorithm is first initialized by plotting training data points based on specific properties as denoted by plot axes (i.e., pitch rhythm, repetition, etc.). These training points are labelled based on their classification and should form clusters by class on the plot. When using the algorithm, an unclassified data point is introduced to the model. The new data point is classified based on the training group with the highest representation among *k* nearest training samples. If more than one class has the same proportion of representation, the class with training points having the minimum average distance from the new data point is chosen[p].

Detailed steps to reproduce this KNN models are outlined below for any reproduced purpose:

1. Import Required Libraries (np, pd, wav, mfcc, TemporaryFile, os, math, pickle, random, and operator)
2. Define a function that calculates distance between feature vectors and find neighbours. To elaborate further, this function should find each point’s distance with another point in the training data set, find all the nearest *k* neighbours, and return all neighbours to calculate the distance of two points.
3. Define a function to identify the class that has the maximum neighbours count. Therefore, we can store the class and its respective count of neighbours.
4. Define a function to evaluate the model and check accuracy and performance of the algorithms.
5. Conduct feature extraction including high-level features such as chords, rhythms, melody; mid-level features such as beat level attributes, pitch-like fluctuation patterns, and MFCCs; low-level features such as energy and a zero-crossing rate. Note: MFCC helps to extract mid-level and low-level features.
6. Clean up samples including dividing audio files to 20-40 ms long, separate linguistic frequencies from the noise through discrete cosine transform (DCT) of the frequencies
7. Train the classifier through the use of training data set, validation data set, and testing data set.

The paper has achieved approximately 70% of the accuracy and the relevant codes are provided for the team to compare with this baseline model.

<https://dl.acm.org/doi/abs/10.1145/3459665>

* Simple model to classify data into categories
* The model begins by plotting data points by specific properties denoted by the plot axes (i.e., pitch, rhythm, repetition, etc.)
* An unclassified data point is added to the plot
* K data points closest to the unclassified data point are retrieved, and the category with the highest proportion of retrieved points denominates the classification of the new data point

### 7.0 Ethical Considerations (Alex)

“Thoughtful consideration of ethical issues in data collection, and the impact of using the model”

* Ethical Issues:
  + Data Collection:
  + Model Use:

Ethical considerations in data collection and impact of using the model are as follows:

1. Informed Consent:

This is a key factor because the dataset contains audio data that was created by a individuals, which is why it is important to ensure that the creators are informed about their music being used to compile a dataset, how their music will be used and obtaining permission. This is more so to prevent intellectual theft and copyright infringements. This also implies us as collectors being transparent with our data collection means and making the collected data replicable.

1. Bias and Fairness:

It is imperative that the dataset be representative of a variety of genres and styles to avoid bias. Selection of music should be based on a clear and objective criteria, not personal biases or preferences. A consequence that should be avoided is publishing a false classification as this could harm the sales of the artists.

1. Privacy:

The music creators or rights holders should be assured that their personal information, such as their name and contact information, will not be shared or disclosed without their consent. They should also be informed about how their music will be stored, secured, and used to ensure their privacy is protected.

* Limitations:
  + Model:
  + Training Data:

The limitations of the model and the training data are as follows:

1. Small Dataset Size:

The GTZAN dataset is rather small, as it only contains 1000 elements. This is sufficient for training but if the dataset were bigger, it would go a long way in terms of improving the performance of the CNN

1. Limited Diversity and Cultural Bias:

The dataset has only 10 genres, which limits the ability of the CNN to classify music outside of those 10 specific models.

1. Data Quality:

The GTZAN dataset has been shown to have inconsistent audio quality and some samples contain noise or other artifacts that are capable of affecting the performance of the CNN. Another issue is with the labels in the dataset. The labels are not entirely accurate or consistent, which may introduce errors into the CNN and harm its performance.

### 8.0 Project Plan (Blythe)

| Team Members | Tasks |
| --- | --- |
| Alex | Project Proposal:   * Data Processing * Ethical Considerations   Data Processing |
| Anna | Project Proposal:   * Background & Related Work * Architecture * Baseline Model   Baseline Model |
| Blythe | Project Proposal:   * Illustration/Figure * Project Plan * Baseline Model   CNN Model |
| Tyler | Project Proposal:   * Introduction * Risk Register * Abstract   CNN Model |

* When will the team meet?

The team will meet **weekly from 6:00 - 7:00 pm on Monday** and can either be virtual or in-person depending on the assigned tasks. This meeting will be used to provide any weekly updates from each team member, track project progress, distribute work, and discuss any difficulties in the assigned tasks. If additional meeting is required, the team will schedule additional meeting based on everyone’s availabilities on when2meet platform.

* How will the team communicate with each other?

The team will communicate through WhatsApp group as the main source of channel. This ensures if team members have any questions or concerns outside the meeting time, other team members can be reached directly.

* How to prevent overwrite other’s code?

The team will use a shared Google Collab coding script to prevent overwriting other’s code. In addition, clear tasks and expectations will be outlined to prevent overlapping in responsibilities.

* Deadlines and Internal Deadlines

| Project Deliverables | Deadlines | Internal Deadlines | Descriptions |
| --- | --- | --- | --- |
| Project Proposal | Friday,  Feb 17 | Thursday, Feb 16 | The deliverable demonstrates the initiation on the project that covers the goals, motivation, training dataset, rough idea of the neural network, related work, measure of success, and team dynamics. |
| Project Progress Report | Friday, March 17 | Thursday, March 16 | This deliverable demonstrates the team is on track on the project and should have collected all the data, produced a baseline model, and produced at least one result from training the neural network model. |
| Project Presentation | Friday, April 7 | Wednesday, April 5 | This deliverable contains a final at most 7 min video presentation covering the project goals, the deep learning model, and a live demonstration of the model results. |
| Project Final Report | Friday, April 14 | Wednesday, April 12 | This deliverable contains addendum that covers their personal contribution to the project and more details will be provided by the teaching team. |

Additional Deadlines:

* Collect the data in wav. Format (Feb 27th)
* Data Preprocessing (2 people) (Feb 27th)
  + Transfer audio files into spectrograms through Fourier Transform
* Produce the Baseline Model (March 12) (Anna)
* Feed sample data to the baseline for accuracy
* Building the model (March 10) (2 people min)
  + Feature extraction (March 5)
  + Filter generation (March 11)
    - Weights
    - Kernel size
    - Layers (Convolutions/ Pooling)
    - Epochs
    - Batch Size
    - Learning Rate
  + Training with testing and validation dataset
* Preliminary Results (producing 1-2 results) (March 12)
* Project Progress Report (March 1)
* Adjust hyperparameters on testing and validation dataset
* Assess Results through final testing
* Final Deliverables
  + Report
  + Presentation

“Plan provides enough detail about the breakdown of tasks, internal deadlines, and team member responsibilities so that a new team member can replace an existing one and know roughly what their responsibilities are. Work is divided evenly amongst team members”

* Task Assignment, Current Progress, and Deadlines table
* Team Charter:
  + How will we work together?:
  + Scheduled Meeting times:
  + Communication Method:
  + Code Overwriting Prevention:

### 9.0 Risk Register (Tyler)

“A thorough analysis of the major risks and their solutions. The risks are well aligned to the proposed project”

1. Risk 1 - Biased Data (Project):

Description: The collected data could come from the same source of genre classifier to train and validate the model. Therefore the model could be biased towards a specific method and taste of classifying music genres.

Likelihood: Unlinkely

Contingency: Using a set of diverse datasets, the music is more likely to have been classified by numerous sources as well as cover a wider variety of genres.

1. Risk 2 - Overfitting (Project):

Description: If the model performs exceptionally well on the training set, but poorly on the validation or testing set, then the model is overfitted. Thus the model has not learned how to classify new data since its memorised patterns specific to the training set. [<https://www.v7labs.com/blog/overfitting> ]

Likelihood: Likely

Contingency: The model can apply cross validation, reducing the number of epochs, and regularisation to prevent overfitting.

1. Risk 3 - Accuracy (Project):

Description: If the model is unoptimized and inaccurate, the music genres will be classified incorrectly and user experiences will decrease.

Likelihood: Somewhat

Contingency (Project): Applying a large, accurate data set to train the model on will allow the model to learn more. In addition, parameters should be optimised for the lowest validation error and loss. [<https://www.freecodecamp.org/news/improve-image-recognition-model-accuracy-with-these-hacks/#:~:text=Increase%20Epochs,of%20data%20in%20your%20dataset>. ]

1. Risk 4 - Copyright (Project):

Description: Larger online data sets may be limited to music that is free from copyright which does not accurately represent the most relevant music. Likelihood: Somewhat

Contingency: The team can obtain data using an account to legally download music from streaming sites, and ensure the music data will be only used for research and not distribution purposes.

1. Risk 5 - Rushed or unfinished work (Team)

Description: If the team starts work late, there is a possibility of missed deadlines, and incomplete/unedited sections since the project will be rushed. This will result in a poorly constructed and documented neural network.

Likelihood: Unlikely

Contingency: The team will set many internal deadlines and check-ins to ensure work is completed gradually. This will give time to ask questions to the TAs or Professor, edit work, and propose new ideas.

\*can be team or project related

### 10.0 Project Link

“Includes link to public Colab notebook or a public GitHub repo”

* Link:

### 11.0 References

“At least 5 references, formatted consistently using the standard format”