Capstone Two - DL Music Genre Classifier deployed as a web application

What is the problem you want to solve?

Deep Learning has many applications to the world of music & audio. Traditional machine learning on digital audio data requires the extraction of features from audio samples, features such as band energy ratio, spectral flux, amplitude envelope, etc. Deep Learning has become a preferred way of doing AI-powered audio tasks because it skips the need for traditional feature engineering for audio.
The problem we are setting out to solve is that of classifying musical genres automatically. The idea for this stemmed from the theory that each musical genre has distinct characteristics that distinguish it from other genres (e.g. bass-heavy drums in hip-hop, chord progressions in jazz, etc.). There could be many use cases for this. A streaming provider such as Spotify may not have the time or resources to listen to every song that hits the platform and categorize it manually. A DJ may wish to take their existing catalog of music and organize it by genre so that he can choose different songs that fit the event they are playing. A large record label might want to delegate different genres of demo tapes they receive to different departments (say, a hip-hop department vs. a pop department). An automated system that can classify songs automatically drastically reduces the time and resources needed to do this manually.

Who is your client and why do they care about this problem? In other words, what will your client do or decide based on your analysis?

Client will be a new streaming provider that wants to improve their recommendations to users by recommending more songs from their favorite genre. They want a web interface for people to upload a songs and have them instantly classified by genre.

What data are you using? How will you acquire the data?

The data we will be using is the famous GTZAN dataset. It is the most-used dataset for exploring problems in Music Genre Recognition. It contains 10 genres with 100 30-second audio files each. The files were collected in 2000-2001 from a variety of sources including personal CDs, radio, microphone recordings, in order to represent a variety of recording conditions. The data is available on Kaggle
https://www.kaggle.com/andradaolteanu/gtzan-dataset-music-genre-classification
Optionally, we could expand the classifier to get more 30 second samples from labeled data using Spotify's API, this would probably improve the algorithm's accuracy as more data is usually better.

Briefly outline how you'll solve this problem. Your approach may change later, but this is a good first step to get you thinking about a method and solution.

Components:
Pre-processing pipeline

- Mersyas / GTZAN dataset

- Will perform digital signal processing and extraction of MFCC’s on which the model will be trained
- Potentially use other genre-labeled data such as Spotify
- Potentially a cloud function to periodically train the model
- Using Python DSP library called *librosa*

Model

- Neural Network that takes the pre-processed audio data as input
- Will use TensorFlow / Keras
- Closely follow the tutorial, and maybe try and achieve even better accuracy by exploring parameters in Keras / diff types of ANN’s

Deployment / Web app

- Will deploy the model on the backend of a Web Application as a Flask API. Will use Docker to containerize the app
- Will use nginx as the web traffic handler. Dockerized nginx
- Kubernetes to have the containers talk to each other
- All of this accessible through a simple React frontend

- What are your deliverables?
  - A link to the deployed website
  - Short video walkthrough tutorial of the website
  - Github with source code
  - Paper describing the process, challenges faced, further work, etc.
  - Slide deck describing the process, design challenges faced, etc.