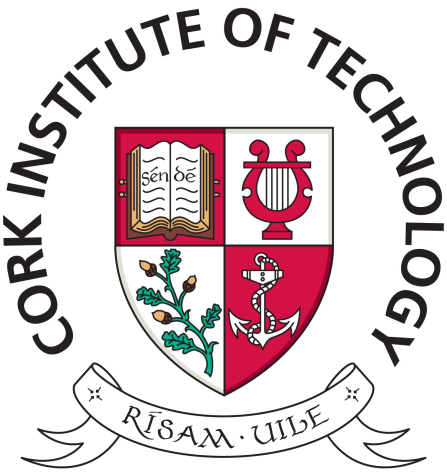

Enabling a Neural Network to Recognize Key Features in Music

by

Cian Roddis

This thesis has been submitted in partial fulfillment for the degree of Bachelor of Science in Software Development

in the
Faculty of Engineering and Science
Department of Computer Science

May 2019

Declaration of Authorship

I, Cian Roddis , declare that this thesis titled, 'Enabling a Neural Network to Recognize Key Features in Music' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for an undergraduate degree at Cork Institute of Technology.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at Cork Institute of Technology or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this project report is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

CORK INSTITUTE OF TECHNOLOGY

Abstract

Faculty of Engineering and Science
Department of Computer Science

Bachelor of Science

by Cian Roddis

The goal of this project is to train a machine learning model to be able to categorize a piece of music. This will be done by user submitted audio. As it is an unseen piece of audio, the program will have to extract and analyze key features of the audio and try to classify it by comparing it to seen, classified data. This is known as supervised learning. A major factor when it comes to a piece of music is genre. As music can be extremely complex, five algorithms, independent of each other are run on the piece of audio to get a more rounded outcome. The desired outcome is an accurate classification of the unseen audios genre. The classifications will be then fed through a graphing library for user friendly feedback.

Acknowledgements

I would like to thank the following people that have or will assist me to achieve my goals in my Final Year:

- Dr. Sean Mc Sweeney (Term 1 Supervisor)
- Byron Tracy (Term 2 Supervisor)
- Anthony Roddis (Father)
- Sarah Roddis (Mother)
- Dr. Alejandro Arbelaez (Machine Learning Lecturer)

Contents

List of Figures

List of Tables

Abbreviations

ML	**M**achine **L**earning
AI	**A**rtificial **I**ntelligence
KNN	**K** **N**earest **N**eighbour
NB	**N**aive **B**ayes
AWS	**A**mazon **W**eb **S**ervices
MIDI	**M**usical **I**nstrument **D**igital **I**nterface
MVP	**M**inimal **V**iable **P**roduct
SAS	**S**tatistical **A**nalysis **S**ystem
ANN	**A**rtificial **N**eural **N**etwork
UI	**U**ser **I**nterface

Dedicated to this my Parents for enabling me to attend such a prestigious Institute

Chapter 1

Introduction

Imagine being a music lover but not being able to accurately identify the music you are listening to and wanting to share it with friends. I and a lot of people I know struggle when asked, "What kind of music is it?", to narrowly define what it is we are listening to. It's difficult to describe some songs using just words. Sometimes you can't even narrow down the genre. Some songs are easy, such as Jimi Hendrix's Voodoo Child or Pink Floyd's Shine on you Crazy Diamond. These are easily classified because of the age we live in now and how much we know about the rock explosion of the 70's. But these artists were pioneers and people probably struggled back then to describe what they were hearing. They stretched boundaries and shaped music as we know it today, yet they still managed to captivate the ears of listeners back in the day. Imagine trying to explain King Crimson's music through words back in the 70's when it was nothing like you've ever heard before. You may even struggle to do that today without a convenient way of listening, such as streaming. Experimental music is in abundance and lots of crazier, almost-unidentifiable (in some cases, e.g Iglooghost) music is being created every day. This project is being made to combat this problem, to narrow down what confuses many music-lovers about the music they want to share with friends and allow them to give accurate descriptions.

1.1 Motivation

The motivation for this project was a personal interest. I wanted to help myself and music lovers around the world be able to accurately classify the music they are listening to, using AI. In doing this, I will be giving a rough view, using percentages, to say how much the machine thinks a song is this much of one genre, so much of another genre, etc. Another big motivator of mine is getting to work with machine learning algorithms.

Machine learning is exploding in usage. If any major IT company refuses to use ML these days, in some aspects, they are behind the curve. Hence, I have a major interest in this project and this will be a huge driving factor for me to get it right.

My initial idea for my project was to produce music using AI after seeing a couple of videos I found on YouTube, browsing. I found the Daddys Car video [1], in which the music (minus the vocals) was entirely created by Sonys CSL Al. The constraint here Is that the Al was able to produce music similar to the Beatles but incapable of actually arranging the music nor sticking to a sound or structure. The music used to create the song is extracted in snippets and arranged to sound like a song that is familiar to our ears, by a human. To back up this constraint even more, the second video by Carykh [2] shows us the non listenable garbage that an Al that produces on its own. Sony does not disclose their entire sample set. Carykh however, gives us the entirety of the songs [3] that his machine produced in .mp3, .wav and raw MIDI formats. He tries to make the machine produce jazz. The video allowed me to understand an important aspect about machine learning though. The music it pumped out after only a few songs were fed to it, was ridiculously bad and sounded nothing like what he was going for. The more data/music he fed the program, the closer it got to being jazz. This progress halted at a certain point. A machine can only do so much and in this case the diminishing returns on data became abundant eventually. I found these videos when thinking of a project. My initial Idea was to do what these videos were aiming for: produce music entirely by Al. However, examining their results, I decided to change my Idea and thats where I got the idea to classify music using a machine learning program.

1.2 Contribution

The goal of this project, as I said, is to train a machine with supervised learning to be able to classify music by listening for key features such as beats per minute, rhythm, also possibly key and time signature. I will be using Tensor Flow along with various libraries such as: NumPy and Pandas. I will be building a suggestion program that will accurately feed back to the user what percentage of a certain genre the machine thinks the song is. For example: Brockhampton's song, 'Tape' would be 75 percent hip-hop and 25 percent orchestral/classical, roughly speaking.

With my knowledge of Python and parsing JSON objects with Google API manager, creating an environment that is easy and straightforward for a user to get to grips with should be a breeze. The user should also want to use it. Otherwise, what's the point in this research project.

1.3 Structure of This Document

In this chapter I introduced you to the area of research in question and why I want to focus my studies here. I've given a brief overview of my thoughts. In the next chapter I will be talking about what is being done about my problem area right now and also, what has been done (such as Shazam for song recognition/identification). Chapter 2 will be the bulk of the research I've done in this field. With AI being so comprehensive, I will be drawing inspiration and extracting information from a multitude of places.

Chapter 3 will focus in on my problem statement and the challenges that I will have to overcome in order to complete my mission. Basically, it will be what I want to achieve. I will be talking about the functional and non-functional requirements.

Chapter 4 will address the actual Implementation methods I expect to use. This includes which technologies I will be implementing, the hardware I will be using to achieve my goals. I will provide a high level of my system and how I will implement the infrastructure plan to use. This chapter will also evaluate my risk assessment, methodology, my schedule, evaluation of my work ethic and I will talk about my prototype.

Finally, I will have a Conclusion section to round up all I've learned in my studies of this problem statement. I will provide conclusive thoughts about my project and I will let you know what I plan to do with this project in the future.

Chapter 2

Background

Machine learning is one of the most cutting edge technologies the tech world are working on at the moment of writing this paper and I believe it will be for the foreseeable future. SAS defines Machine Learning as: "a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention." [4].

2.1 Thematic Area within Computer Science

The thematic area within Computer science I will be looking at will be software engineering based using suitable data structures for efficient feedback to a user. I will be trying to optimize my design of my machine learning network for Boolean satisfy-ability. My network will be artificially intelligent and will try to solve problems by themselves. Machine learning will be the specific avenue I venture down in Artificial intelligence, where the machine will base its answer off a large set of data and come to a conclusion about some unseen data [5].

The three levels I will step into will be: **Software Engineering** as I am developing an application for a user; **Artificial Intelligence** as choices will be made without human persuasion; and even deeper we will venture into AI with **Machine Learning** by allowing a machine to learn from a dataset to perform a specific task.

2.2 A Review of Machine Learning and Artificial Intelligence

Machine Learning is seen as a branch of Artificial Intelligence because AI is the seek for knowledge. Machine learning is acquiring knowledge and implementing that knowledge [6]. AI is intelligence that is created by humans and has not formed naturally [7]. Machine learning then, is an application of AI that provides systems the ability to automatically learn and improve from experience without being explicitly programmed by humans [8]. There are 4 main kinds of learning a machine can incorporate. These are:

1. Supervised: The machine is given normalized data by us and learns from this. It can then try to provide an answer back to a user when given unseen data. This is what we will be using in this project. An example of this would be visual learning such as facial recognition Facebook use for tagging people in pictures software. [9]

2. Unsupervised: The machine learns on its own with no data input from us. This can be used for transactional data such as learning similar viewers interests on YouTube and suggesting back videos a viewer may find interesting. This would predict human behaviour that has never been told what kind of videos we watch but choose things based on what it has observed of our behaviour. [10]

3. Semi-supervised Learning: This is a mix of Supervised and Unsupervised Learning. This method takes very minimal data and a lot of unseen data. An example of such would be Google's Image Captcha software. The software only sees a handful of road signs but learns from users more and more what road signs look like as it's used. [11]

4. Reinforcement learning: This method is a trial and error way of machine learning. The machine will act differently after it makes a mistake to try to achieve the accumulative reward. There are many examples of this such as video games being fully completed by machines (e.g Super Mario Bros.), chess masters and Go masters getting beat at their own game by machines, etc. [12]

Artificial Neural Networks are modelled after our brains. Notice the terminology even. How we pass information around the brain is through neurons by synapses firing an electric current and the receiving neuron can either accept or reject the signal. We model artificial networks similarly. A node or neuron in our artificial system will receive some form of data. This data will then be analyzed and if it passes the test is sent to the next layer of nodes/categorization. If there are only 2 layers of nodes, an

input layer and an output layer, then a network is classified as a shallow network as there's not much categorization layers to pass through. Anything more than 2 layers is known as a deep network as the layers in between have hidden layers that we cannot see. [13]

Machine learning has 2 kinds of networks: shallow networks and deep networks with multiple popular algorithms for implementing these networks. The difference between deep learning and shallow, is that deep learning has hidden layers of nodes that a user cannot see operations or steps the algorithm takes as it works its way towards a conclusion or the output layer. In a shallow network a user can see the input layer (e.g. Picture of a Cat) and an output layer (e.g. The machine predicting the picture contains a cat). Example:

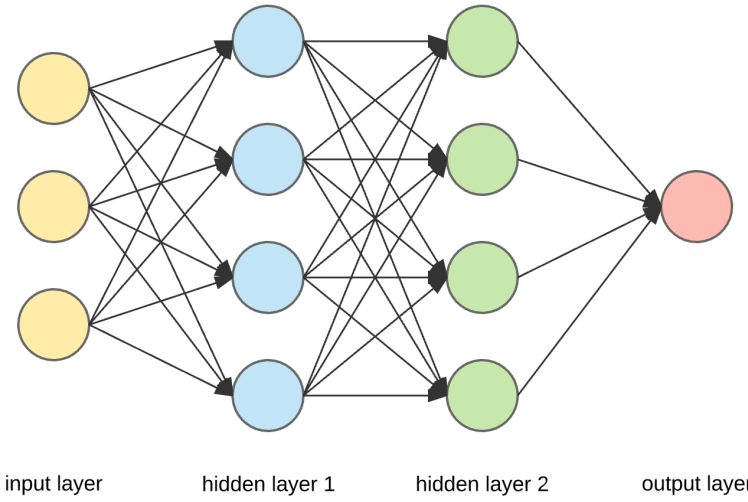

FIGURE 2.1: Deep Neural Network[14]

Deep networks however do increase complexity as you are increasing tasks that have to be done in order to achieve a desired outcome. Shallow networks are still perfectly viable for most systems but deep networks have picked up in popularity due to improved accuracy and computational power[15]

There are numerous algorithm classifiers for machine learning and each have their own characteristics. Each has a certain edge somewhere within data analysis. Classifiers are a way of recognizing data and predicting a certain data point into the classification an algorithms sees fit. Such as a musical piece's genre. Some algorithms are more likely to be more accurate than others for different tasks. Some of the algorithms most likely to assist my project are (in order of most relevant):

- K Nearest Neighbour (KNN): K Nearest Neighbour is a wonderful algorithm that maps out data and finds the most likely classification for point x given the details of K nodes closest to x.

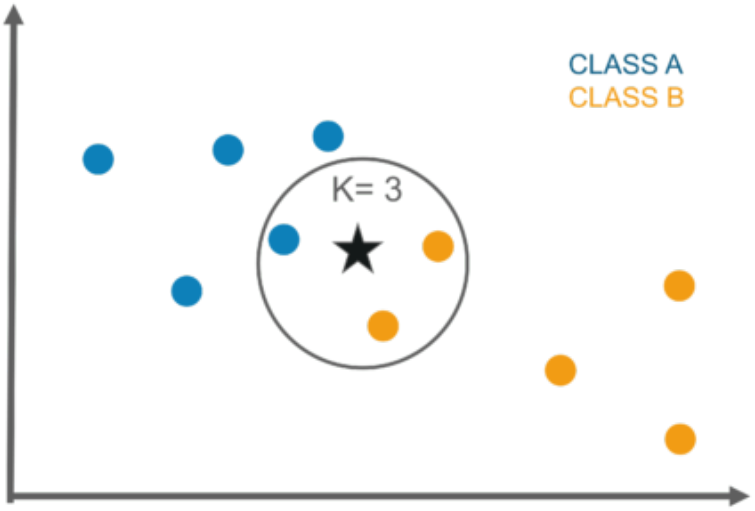

FIGURE 2.2: KNN Visual[16]

KNN can require a lot of memory or space to store all of the data but only performs a calculation (or learn) when a prediction is needed [16], which in our case is perfect as we will be sending data to be processed, instance by instance. KNN in this regard will be wonderfully efficient. The figure below shows our target we want to identify. The circle is an indication of the 3 closest nodes to our star. In this case the star would be classified as yellow considering no weights are in effect for the strength of the blues. We have to be able to calculate distances from our desired node in order to get a result for this algorithm. Here are variations of KNN using different kinds of differences.
Euclidean[17]:

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \tag{2.1}$$

Manhattan[17]:

$$\sum_{i=1}^{k}|x_i - y_i| \tag{2.2}$$

- Naive Bayes is a good percentage predictor which could come in handy when I am looking for a percentage based answer on how accurate the system is when predicting a genre. However Naive Bayes is generally used for text classification but can definitely be useful elsewhere. There are a few different Naive Bayes algorithms with respective equations. This includes Gaussian, Multinomial and

Bernoulli. Each has a different way of calculating but in essence are all variations of Naive Bayes. The Bernoulli equation is written as such:

$$\mathbf{P}(h_j \mid d) = \frac{P(d \mid h_j) P(h_j)}{P(d)} \tag{2.3}$$

However Bernoulli is only effective when it's binary predictions. One or the other. Multinomial Naive Bayes is used in classification and it assumes that features follow a normal distribution [18]. The Equation Multinomial follows is:

$$\mathbf{P}(w \mid c) = \frac{count(w, c) + 1}{count(c) + |V|} \tag{2.4}$$

The most important factors of KNN vs NB: NB is good if you have a lot of data points and speed is important to your program. KNN is good if you know that not having conditional independence in your data will have a highly negative influence on classification, or if the decision boundary is not linear, elliptic or parabolic [19].

There are generally 2 ways of identifying music for machine usage and machine learning [20]. These are a symbolic method and audio method. Symbolic being MIDI recognition as seen in Carykh's video on creating Jazz using a machine learning method [2], and Audio which is the more nuanced approach as each song sounds different, even if just minimal differences. MIDI is a lot easier to handle as it is purely signal based. This option will be useless to us as we want input from a user but I'll speak more on this in the next chapter.

2.3 Information Sources

Identifying the cutting-edge research in this branch of the industry is difficult to pinpoint as there are a plethora of companies, conferences, studies, etc. invested in Machine Learning and AI nowadays. However, I am going to evaluate some of the top studies, projects and conferences happening currently.

- Two of the leading conferences in Machine Learning is of course hosted by IEEE and NIPS(Neural Information Processing Systems) [21]. IEEE has a whole catalog of impressive feats under their belt in ML including their many videos on YouTube with Boston Dynamics [22]. NIPS have a long standing history in ML. The upcoming conference will be the Thirty-Second conference they host and will take place on the 2nd of December through to the 8th of December 2018. More discussed below:

- IEEE host a multitude of events all over the world[21] and have the number 1 most anticipated conference for 2019 in the world right now. This conference, CVPR, will have over 150 organizations showing off their latest equipment/-software as well as students, academic staff and more. The conference focus is on computer vision, machine learning and artificial intelligence. It is the best attended conference in computer vision and machine learning and it has a 91 percent positive feedback in gaining value from attendees it is clearly the leading conference in this field of study.
- NIPS is a top conference that takes place over the better part of a week. They host live competitions talks and seminars, as well as the first day being an exposition of ongoing work within the industry. The likes of Google, Amazon, Nvidia, Pony AI, Capital One and many more companies will be doing talks and presentations. A lot of them talking about AI and Machine learning. Every tech company nowadays are working continuously on automation and are eager to show off their latest achievements in artificial intelligence. This expo is a great opportunity to meet with top companies, see what they're doing and get to network with people in these positions of work. [23]
- Another long standing conference, ICML, took place in July 2018 and it was the 35th annual occurrence of this conference. Conferences that have this kind of longevity prove their worth and continue to deliver outstanding information and organization. If not they would not be one of the top and most-sought after conferences in machine learning, to date. ICML is supported by the International Machine Learning Society or IMLS. This one, was hosted this year in Sweden but will be hosted on Long Beach, CA next year, similar to the other two conferences on this list.
- 'Deep Convolutional Neural Networks for Predominant Instrument Recognition in Polyphonic Music'[24] is a Journal by Yoonchang Han, Jaehun Kim and Kyogu Lee of South Korea. They have developed a neural network that is able to single out individual instruments in pieces of music. This, according to them can be used to help identify genres or assist with transcription of music audio to sheet music. In this journal they discuss their system architecture, including their audio processing technique, their network architecture and training techniques which will be beneficial information for this project.
- 'Deep Learning and Music Adversaries' discusses ways of music context analysis. This paper can be used to further accuracy on recommendation engines for music, auto-generated music playlists like Spotify or Apple create, and much more. Music classification is only growing in demand as music becomes

more available, which it very much is with the introduction and popularity explosion in streaming services.

- Books and Journals have always been a great way to help with academic pursuits. In fact, they are academic pursuits in themselves. A few books I have been looking at to assist my learning are:
  - Guest Editorial: Machine learning in and for music by Gerhard Widmer. Written in 2006, Widmer speaks about key features to be identified in music. He does a melodic analysis where he identifies notes, pitch, duration and breaks. [25]
  - A paper published at '2017 Signal Processing and Communications Applications Conference' goes into specific techniques about genre classifying music. Although in Turkish, I can understand their graphical data and algorithms used. They use K-NN as a way to gauge what kind of music it is. They note volume as a key feature and graph that to show us the difference between Metal music and Classical. The decibel difference is enormous. [26]
  - Introduction to Machine Learning by Ethem Alpaydin and Francis Bach [27] comes highly recommended and highly rated as it is a well rounded book for students, professionals and more to come to grips with the techniques and algorithms used in machine learning today. The book was published in 2014, so the technology is recent enough to retain its relevancy. Due to the well rounded nature of the book, not much has changed as far as basics go for machine learning. The book discusses things from supervised learning and decision trees to reinforcement learning and dimension reduction. The contents of this book is highly relevant and substantial to progressing towards developing my machine learning algorithm and choosing the most suitable method to process my data.
- As companies will consider what you do for your Final Year Project in their reasoning for hiring you, you need to make sure that your project is desirable in the eyes of employers. As machine learning is on the rise as a whole, the knowledge I will gain from this project alone should be extremely transferable on other projects at a corporate level. A few examples of companies in Ireland, looking for data scientists include [28]:
  - Master Card are currently trying to fill a graduate position in Dublin to work on ML testing, Minimum Viable Products and full project development. As I said, any data science job opening would be suited to this project. As they are working on experimental products, my project might grab a recruiter's interest.

- Amazon are also looking to fill ML positions. Speaking to a small team of them at CIT career fair, I got information from developers working there that they are looking to hire quite a number of developers soon as they have just completed building a brand new office building, built for developers.
- Of course companies directly related to the topic would be interested in this project. Shazam of course comes to mind when audio identification.
- Music streaming services, such as Spotify, Apple Music, Deezer, etc. are growing by the day. These companies implement machine learning into their software by suggesting music to the customer in forms of personal Daily playlists that expand as you listen. They have weekly playlists for the user that have a bunch of songs similar to what they believe the customer will like. They are often accurate and this is exactly the type of work my project would play a role in.

On a more casual/high level of research, there are platforms that will greatly assist ease my learning curve on this project. Below are the most relevant to this project.

- Reddit host one of the best machine learning forums and the users are extremely active. This can be found at "r/MachineLearning" [29]. Users here can usually help you with anything. Usually a user will know a dataset you can use for your problem, they can help you with algorithm optimization/-chose what algorithm would suit your problem best, etc. These are just a few instances I witnessed on this sub. This is a valuable resource for anybody working with machine learning.
- Stack Overflow is the leading forum for any software development related questions/problems and of course that extends to machine learning. Stack Overflow has plenty of users willing to help other developers, in fact they receive over 50 million visitors a month, with about 21 million of those being professional developers or university level students. [30].
- Carykh is a YouTube channel where I originally began to develop my project idea and this is who inspired me to research and admire machine learning. With over 300,000 subscribers, he is one of the top machine learning channels on the platform. He makes interesting and fun videos on machine learning which may inspire many younger developers to go down the route of ML.[2]
- 1 Brown 3 Blue is another YouTuber with a subscriber count of 1.2 million. He breaks down the basics of machine learning and the mathematics behind it. He creates amazingly detailed videos with helpful info-graphics to enable views to visualize exactly what he's talking about. [31]

Chapter 3

Recognizing Musical features using Machine Learning

The key question to be addressed in this chapter is: "What do I want to achieve".

This chapter will discuss the problem I am trying to solve. I will dig into the specifics and will be discussing risks/potential hiccups I may encounter with my solution e.g. API problems with AWS. Also I will be discussing key functional requirements as well as non-functional requirements as to provide a better understanding of what I want the system to be/do.

3.1 Problem Definition

The problem I am trying to solve is enabling people to easily identify features within a musical piece. If a person can identify a piece of music with uncertainty, why can't a machine do it for them to be even more accurate? A user will record a piece of music with the application and it will do just that for them.

3.2 Objectives

The application will be developed on Linux and maybe ported to mobile in the future. Processing mp3 files will be easily accessible for this application. In terms of mobile development, this audio would have to be sent off to a server for audio classification due to mobile incompatibility with python. Audio classification is the most important objective of this project, I am aiming for high accuracy in testing unseen data and

hopefully just as high, if not relatively close accuracy for user recorded audio. Of course the whole point of this project is to be able to give accurate information to the user. When showcasing this information back to the user, it will have to be digestible and understandable. Possible servers to host our script could be: Amazon, they provide us with servers that can handle this kind of data transfer. They also provide professional level documentation [32] and there are libraries of video tutorials online for Amazon Web Services. Hopefully I will have little to no problems with implementing the API for AWS with android.

3.3 Functional Requirements

- A user should be able to open the application and clearly determine what everything does.
- A button that can be pressed in order to begin the analyzation process.
- An audio file will be sent for analysis.
- That audio will be analyzed and categorized.
- A result will be stored for continuous learning.
- A result will be sent back to the user and displayed as a percentage ring chart/pie chart.
- The user will be presented with a button to record again.
- The users history will store this result.
- A user can name results as they please.

3.4 Non-Functional Requirements

- Accessibility is open, there will be no user accounts as there is really no need.
- Efficiency is key in the transaction between user and server. If it takes too long a user may get bored quickly. Quick response times are key.
- While the server generates data, the user will not be left without acknowledgement. Aesthetic design will show an ongoing moving loading icon while waiting on the data.

- Quality within this product will be necessary as a user might not provide enough information to get an accurate conclusion for the program to come to. Some fault tolerance level will have to be met.
- Compatibility and Supportability will only apply to Linux for now and potentially the app will branch into the mobile market in the future.
- Reusability will be abundant with this application as each and every song can be categorized that a user has access to.

Chapter 4

Implementation Approach

I will discuss my approach below in detail, but as a quick overview: I will be implementing a user controlled application to submit and classify music. The classification will be done on audio sent by the user to be analyzed by our algorithm. The application will be running the audio through a machine learning process that will attempt to accurately classify the music into the genre it belongs and provide information on its findings.

4.1 Architecture

Description of the architecture of the solution that I have in mind:

- Machine capable of running a Kivy program.
- Python for algorithm scripting.
- Matplotlib is a library within Python to create graphical representations of data.
- Amazon Web Services for basic machine learning algorithm hosting. (Not Achieved)
- Amplify CLI to manage and integrate AWS cloud services. (Not Achieved)
- Android application for future work.

As I said above I will be designing a system that will classify music submitted by the user. The classification will be assessed by a program that can interpolate various values from a piece of music (e.g Tempo, Zero Crossing Rate, Chromagram measuring power levels, etc.). The application will be running the audio through a process that will accurately classify the music into the genre it belongs using sckit algorithms. The machine learning process will be done through the implementation of various algorithms

and then a value will be calculated based on the results from each algorithm. Mathplot Library has amazing graph features. This will be used for creating a pie chart to show our results in a readable fashion. The features this library provides are invaluable. An example of a pie chart that it can produce:

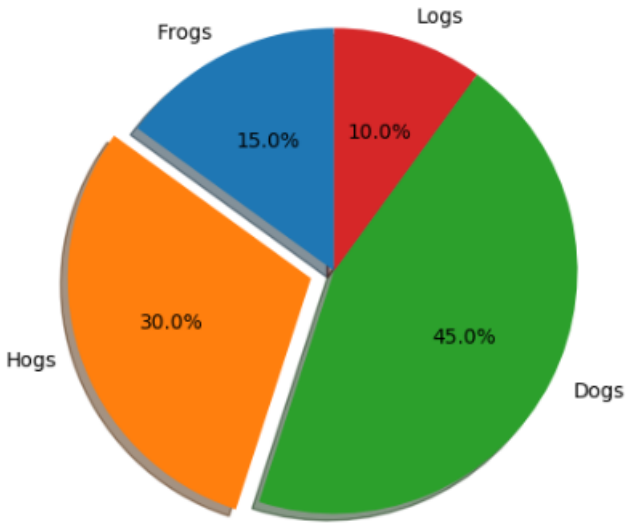

FIGURE 4.1: MathPlotLib Pie Chart[33]

Here we see an example of an animal dataset and have a clearly labeled, distinctive pie chart. We can see that it even has capabilities to highlight a certain section. In this scenario, it's hogs that are exacerbated. We can use this to our advantage and highlight the most likely genre within our music. This will allow for a user-friendly UI.

4.2 Risk Assessment

Potential risks to a project are often overlooked by students, which can lead to catastrophic outcomes. This can be somewhat, if not completely avoidable by careful planning and foresight into what is possible and not possible in your project. In this section I will try to outline some of the more difficult aspects of my project and assess the precautions that should be taken when approaching the implementation of these critical features.

Above we see a matrix of risk classification according to their importance. This matrix shows the possibility of a risk arising and enumerates the decisions you can make to anticipate them or mitigate them, in case problems ever do arise.

TABLE 4.1: Initial risk matrix

Frequency/ Consequence	**1-Rare**	**2-Remote**	**3-Occasional**	**4-Probable**	**5-Frequent**
4-Fatal					
3-Critical					
2-Major					
1-Minor					

Mistakes, blockages or wrong turns are inevitable so trying to identify and segregate them at an early stage will give a great advantage when it comes to designing and implementing. Below I will discuss potential risks and rate them based on the scale to the best of my knowledge. (e.g. Foo method failing: 3-1 = Occasional & Minor)

Potential Risks:

- **Audio Recording:** By limiting or specifying a fixed recording time for what we send to the server, we can potentially avoid long processing times and limit vague answers by the predictor. **Risk: 2-5**
- **Audio Processing:** Processing audio using Librosa should not be difficult but this is crucial to the project as we have to have values comparable to a dataset with labelled genres. Having comparable values means that we have to process our audio exactly how our dataset creators extracted their values. **Risk: 3-5**
- **Displaying Graphical Data:** The graph we generate using Mathplotlib may have to be sent as an image instead of being displayed from the server. My guess is this will have to be the way it is done as to avoid any scaling problems. This in itself comes with its own set of problems. Mainly that it'll be a more expensive operation for the server and the overall processing time. **Risk: 4-5**
- **Processing Time/Cost:** AWS offer free credit for students. Right now I don't know how much my operations will cost. I can't imagine it will be too expensive but I really do not want to run through the credit available to me. **Risk: 1-5**
- **Application Bugs:** The android application will be the only access the user has to the system, so the bugs have to minimal or users will just abandon the application. The app will not have too many crucial features so this should not be a major problem. **Risk: 1-1**
- **Cloud Services:** My idea for cloud services is to send data back and forth from a mobile device. This may prove difficult as I have never worked with this technology. AWS could give problems in processing audio using our algorithms. Hopefully I

can mitigate any problem we get in sending or receiving data with AWS. However, Amazon provides wonderful documentation and there are plenty of guides out there for communicating with AWS. **Risk: 3-3**

4.3 Methodology

This section will discuss how I plan to tackle different parts of this project. I will go over how I plan to build the application, devise a plan to integrate the app with an easy to use user interface and how I will manage my work flow as a whole.

- Using Research Done Already: This thesis has had me finding sources upon sources of information that will assist me greatly. Research phases of projects can help vastly in tackling a large project such as this. Research not only find useful links and informative videos but it also rules out less useful information and methodologies to stay away from.

- Implementing Skills: I have the knowledge needed to know what technologies I should be using now. Using wireframes/mock-ups and proper scheduling to meet self-determined deadlines, I will be able to develop the application I envision. I have most of the computational skills to implement an accurate system but experience with cloud services.

- Planning and Management: I plan to follow a Scrum work flow as I have put this in practice numerous times and it is proven to work from my observation. Scrum is an agile framework for planning out tasks for a specific amount of time [34]. It is usually in increments of 2 weeks or 3 weeks depending on team/manager preference but never longer than a month. The tasks to do within that time period is known as a sprint. It is quite often used in software development teams. Tasks are broken into small increments such that a task is doable within a few hours. This break down is essential to the Scrum methodology as it helps clear up what needs to be done on a week-to-week basis. For example, you don't want a task called 'Build App'. This is too vague and open to interpretation. Scrum is designed to minimized errors in production, so you want to be able to break big tasks such as 'Build App' into smaller ones like 'User Log In', 'Record & Send Data', 'Store user history', etc.
  For larger tasks and weekly/bi-weekly sprints, I will employ the use of a virtual kanban board with Trello. Trello is an online task management application that allows a user to specify what needs to be worked on, who's working on it and more

[35]. I have used this extensively for previous projects and found it very useful for planning sprints.

4.4 Implementation Plan Schedule

I have to prototype, which will be featured in its own section below. For prototyping I have been working with various machine learning algorithms, completing assignments using Naive Bayes and KNN algorithms.

As the implementation phase come to an end, my schedule shows how I was on top of the work load and how sprint planning helped focus tasks. I wrote down every task I could think of into a backlog list of activities on Trello. I then established a full list of Sprints that had a realistic workload for a week. These Sprints were created on a weekly basis and tasks were added/carried over as necessary.

As an overall, unrefined plan, I planned to build the initial framework for the application within the first sprint and add a functionality to simply analyze user audio. Next I attempted to program an accurate classifier, which took roughly 4 sprints if not a little more to get right. Then the next sprint. Any time I had left after I was satisfied with the performance of the application I began styling and creating a user-friendly environment, easy to navigate.

4.5 Evaluation

I will be evaluating my success on whether the core function is working or not. This is key and what the whole project revolves around: Classifying unseen audio. This is the Minimal Viable Product but I will be trying to satisfy every functional requirement. All of the functional requirements I outline in Chapter 3.3 are pretty critical to the design of the app bar the cloud services. If history or some other feature has to get left out in order to incorporate more core functionality then that shall be sacrificed but I will still see the project as a success overall if I can get an MVP going.

Non-Functional requirements outlined in chapter 3.4 are optional goals indeed, but I do believe they will all make the application more pleasing to a potential user and user experience is crucial for the lifespan of all applications. Non-functional requirements

vary in importance. For example, I would consider feedback that a process is occurring more important than efficiency. Now, of course efficiency is hugely important but at least if a user knows there is an ongoing process they will be more patient, therefore we can afford to be more lenient on efficiency.

My views on success will be in audio classification. If that means sacrificing some pretty frills then so be it but as long as a user can classify a particular track they like then I will deem this project a success.

4.6 Prototype

Delving into prototyping for this project, I started by making a basic wireframe to get a general understanding of what I want my app to look like.

A massive inspiration for the home page is most certainly Shazams app. They have a huge unmissable button as the core functionality of their application. This feels very intuitive and almost impossible to misinterpret. This button is used to start recording. Just like Shazam, once a user clicks the button in Figure B.1 they get a loading icon (Figure B.2) to inform them that the program is classifying their audio and then redirected to a screen similar to Figure B.3 to provide them with the genre distribution chart generated by the algorithm. These figures can be found in 'Appendix B, Wireframe Models' section at the end of this document.

I have been working with various machine learning algorithms over the past few months and I believe I have narrowed down which algorithms are best when it comes to processing audio. Below in Appendix A you will see python code I wrote for Naive Bayes, K-Nearest Neighbour using the SciKit learn package to predict whether a student will fail a module due to alcohol consumption with respect to how much study they do. I put the code snippets in the order of KNN (Figure A.1), then NB (Figure A.2) because KNN scored higher accuracy (on average) for this practice experiment. Also I am expecting KNN to be more accurate when it comes to classifying audio for this project.

I implemented a total of five algorithms to classify my audio: KNN, Naive Bayes, Decision Trees, Random Forest and Neural Networks from SciKits python package. The implementation made it easy to classify music on a scalar approach as intended.

Chapter 5

Implementation

This chapter I will go over what was done to complete this project, what caused difficulties in implementation and where I made compromises in order to complete this project/ get it to a point where I was happy with it.

5.1 Difficulties Encountered

I will be enumerating the different difficulties I had found when developing my solution approach. I will sort into three categories of difficulties:

- **Easy**: What I managed to solve with little difficulty
  - **Planning** - Planning and schedule keeping was easy to keep up with for the most part. Bar when there was a lot of other assignment work, time management was relatively simple to keep under control.
  - **Linux Development** - Switching over to Linux from Windows caused a lot less problems with python. Windows is known to harbour a lot of problems that just don't exist on Linux and switching my code base over to Linux was simple.
- **Medium**: It was not easy to solve, but I managed to develop a workaround or solution and still achieve the functionality I originally had in mind.
  - **Machine Learning** - as the main focus of the project, I had to make sure the right algorithms were looked into, aka which would provide the highest degree of accuracy. There are five machine learning algorithms implemented to achieve a rounded score. Each classification is independent of the other but form together to form a pie chart. See figure 5.1 for an example.

  - **Music Analyzing** - I had to find a dataset where I could clearly tell what was going on to achieve similar analyzation on an unseen piece of music. Librosa is a python library that analyzes sound and I managed to find a workaround where I was able to compare using machine learning, the targeted piece of audio against a dataset of similarly extracted features of pieces of music.
  - **GUI Development** - Development of the interface wasn't much effort to get working on Linux. It is build using Kivy. A python GUI development tool that is cross platform with Android, iOS, Windows, OSX and Linux if tweaked in certain ways.
  - **Database** - I had a CSV file with all the information about the music on my local machine that I was accessing for local development but I knew I'd have to access it through the web for a mobile app. Although the mobile app was unable to be completed, I still set up a Google API to query my data through an internet connection.

- **Hard**: The difficulty was so complicated that I was incapable of solving it. As a result, some functional requirement / non-functional requirement or use case from my solution approach was not achieved.

  - **Android GUI** - Porting the Kivy app over to Android came with many problems which I believed to had resolved until I found through Internet forums that some of the necessary libraries such as matplotlib and sklearn were not yet supported by the technology. This led to a scrapping of the Android application but it's not a problem because at that point I had a fully developed, working product for Linux.
  - **Cloud Solutions** - After getting hands on with AWS S3 and EC2 servers I found it difficult to understand how they could be used in the sense that I needed them to be used. I then looked into using Firebase with Google App Manager but this also proved to be completely above my expertise level. As these were new technologies to me I decided it would be best to abandon them and go for a different approach.

5.2 Actual Solution Approach

In December of last year, I set out on writing and researching all I could about this project. I looked into a variety of technologies that could help the development of this project. The project has come a long way since then. If you look at my mock ups made in the first phase, you can tell how much the project has changed from then to now. Difficulties were faced, some overcame and some not.

The architecture I had in mind when first developing this idea was to have the app be portable and for a user to be able to use their microphone on their mobile to record and classify music on the go. At the start this seemed like it would be an easy and effective implementation. I developed an app that recorded for 10 seconds when prompted by a user. The plan was to send this audio off to a server to be analyzed by my already working machine learning algorithms. However, due to lack of knowledge of cloud services such as AWS and Google Cloud, I was unable to develop this in an Android Studio development environment. This led me to check out if you can run python through android, where I found Kivy. Kivy is a python cross platform GUI developing tool. I developed an app that also ran on android and recorded for 10 seconds when prompted. My plan was to classify the song using this new setup but Kivy has yet to support sklearn's library on their python-for-android branch. So, in the end after developing a nice and clean looking GUI, it was unable to run on android. There had to be a compromise made in architecture then due to time constraints. I have now gone with just a desktop Linux application that reads music files from the local machine.

I completed most of my use cases besides server based ones, as mentioned previously. See Section 3.3 and 3.4 for the full list of functional and non-functional requirements but here I will got through and see how the requirements were achieved.

- A user is able to open the app and see clearly what to do. It's a straightforward app that couldn't really confuse a user.
- A button is pressed in order to begin processing. There is a singular button, that when pressed, music is processed
- A recording will be sent to a server for analysis. This was not achieved. I struggled to fathom the complexity of AWS and Google cloud systems when I needed to link them to my application.
- Audio will be processed and stored for continuous learning. This was never implemented as I spent too much time trying to get the aforementioned requirements working.
- A result will be displayed to the user as a pie chart. This was achieved and is working as envisioned from the beginning.
- The user will be presented with the option to record again. This is achieved by providing a user with a button that brings them back to the home page.
- A users history will be stored. This was not achieved due to time constraints. User can name result as they please. Not achieved due to time constraints.

My risk assessment took into mind the Android/portable application that failed to be developed. Therefore a lot of risks were mitigated and some amplified more than what I initially thought. Audio recording was simple. Audio processing was doable. Graphing data was a medium level of difficulty. However, even with all of these working perfectly, the application for mobile got stopped in it's tracks because of Cloud services. The code was there to do all of those things mentioned but I had no way of combining them to make an application that was able to do it all.

The methodology I had in the beginning to develop this project was to make a basic overall plan, have a stacked backlog that I would order by importance and try to stick to the plan as much as possible. The plan went well for the most part but when I couldn't solve my problem with cloud solutions there had to be compromises made and more research made to get a functioning application.

My implementation schedule composed of a Sprint being made every week to make sure things were getting done and I would move tasks from the backlog to the Sprint boards at the start of any given week and try to complete as much as possible. If a task was not finished in a week I would just move it to the next Sprint. Usually a task wouldn't linger in the Sprint boards for longer than 2 weeks.

My own evaluation of my project is that I did well for the most part but definitely could have done better when it came to the cloud computing aspects of the project. This crippled my android application, the original implementation I had in mind. Had I more experience or researched into the difficulties cloud solutions come with I might have been better equipped when it came to implementation and I might not have wasted a lot of time trying to make it work.

My prototype consists of a Kivy GUI for ease of use. The user specifies the song they would like to hear and just hit a button. Everything is automated from there, the extraction of features, querying the database using Google API Manager for the data set, and of course the classification of the music into its respective genres. After classification the user is brought to the result screen as seen in Fig 5.1. From here they have the choice to go back and try again if they wish.

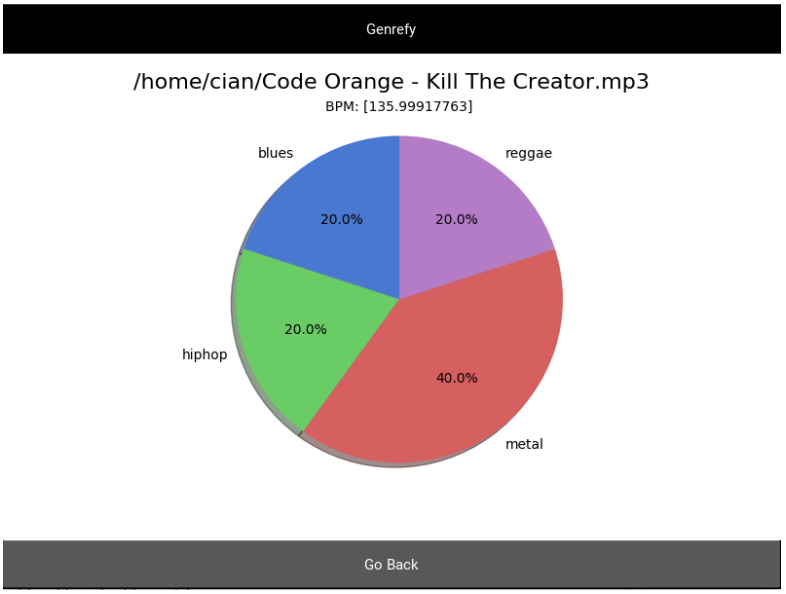

FIGURE 5.1: After song analysis

Chapter 6

Testing and Evaluation

The goal of this chapter is an objective evaluation of the final system in place. The evaluation will be quantitative and will go over analytic findings when comparing my project to any potential commercially available product (or lack there of) in a similar standing to my own.

On an initial search, it seems that commercially available software that classifies a users music into a scalar genre spectrum using machine learning networks is nowhere to be seen. It is however implemented behind the scenes in applications such as Spotify and Apple Music. However, when music is uploaded by an artist to these platforms, they specify the genre themselves (known from personal experience). Many of Spotify's genre based playlists are then are also curated. They do, in fact, use machine learning to generate user specific "Discover Playlists", which are unique to a user and takes into account what they already listen to[36]. Spotify has another 2 implementations of machine learning in which they, 1. try to fine tune recommendation engines based on user preference and 2. are trying to work with artists to assist in the creation of songs.

A product that will classify your music for you seems scarce in the market

6.1 Metrics

The metrics I have gathered are based on my particular view of whether a process is correct in it's classification or not. Also how accurate the training instances believe they are themselves.

From my testing and findings, for any particular song, the correct genre(according to the artist) was present in the classification around 74 percent of the time. I tested 10 songs all of different genres to see where the classifier performed well or poor.

6.2 System Testing

In testing for Table 6.1, 10 songs were collected, of various genres to put my algorithms to the test. The goal was to get songs in different genres from each other but it was also important to test songs of multiple genres such as Hank Williams - Lovesick Blues. Being a Country/Blues song, it was counted as a success if one of the two genres appeared in the classification.

To test these songs, the classification process was run on each song five times, to make a total of 50 unit tests. Every test ran on any particular song didn't change much in the outcome for that song. Meaning if the algorithm gave back a correct result once, it generally got that result every time. It could come back with a very slight change in result sometimes.

6.3 Results

TABLE 6.1: Classification Testing

Test Run/ Song	**Genre**	**1st Run**	**2nd Run**	**3rd Run**	**4th Run**	**5th Run**
Avantdale Bowling Club - Years Gone By	Jazz/Hip hop					
Bob Marley - Three Little Birds	Reggae					
Code Orange - Kill The Creator	Metal					
Hank Williams - Lovesick Blues	Country/Blues					
Justin Bieber - Baby	Pop					
Kevin Abstract - Peach	Pop/Hip hop					
Mahler - 8th Symphony snippet	Classical					
Queen - We Will Rock You	Rock					
The Beatles - Come Together	Pop/Rock					
XXXTentation - Look At Me!	Hiphop/Metal					

In the table we see a lot of positive results (Green) and a few negative results (Red). A green result means that one or more of the official genre(s), according to Google, was present in the applications classification of that particular track on that particular run.

Of the 50 unit tests carried out, 34 were successful giving this matrix and application a 74 percent success rate. If you look closely you can see that the classification algorithms had a difficult time getting an accurate reading on Pop music. Considering Pop music is a mesh of all types of Genres, there's no wonder why there was such a poor classification of pop in testing.

Pop being the only outlier in my testing, it is somewhat worrying. Pop is short for popular so if this was to be used in a commercially available product, the average user would not find it to be too accurate as the average user listens to pop music.

Chapter 7

Discussion and Conclusions

This chapter I will discuss challenges and problems I faced when completing this thesis. I will also discuss what I may do with the work I've done here in the coming future. I see this application as potentially having a future so I will discuss what plans I have with it.

7.1 Solution Review

My solution was initially meant to have a mobile phone application but I ran into problems with library incompatibilities so the final product still had the underlying goal completed of classifying music into scalar spread of genres. The only difference being that the application was completed as a Linux application. The application is minimalistic and classifies one song at a time but it is effective in displaying the critical information in an easily digestible way

7.2 Project Review

During the research phase, I have gathered much knowledge about this aspect of software engineering. That however didn't come without it's problems. For the most part I was extremely on top of what was expected of me, reaching word counts for chapters, weekly updates, attending meeting, etc. There came a time however towards the end where my work for this part of the project had to suffer due to other commitments and deadlines. I work well under pressure but I prefer having somewhat of a cushion of time in case something unforeseen comes up and I, like everyone else, can write more.

Prototyping was a small problem I encountered due to time constraints. Not much prototyping was actually done, because I was focusing on the report during this time. I had a wireframe model of what I want roughly from my design. I also have three fully fledged Android applications from years previous and I have one design in particular that I had imagined I would be implementing into this project (See Prototyping section - 4.6). I also have done some work with machine learning algorithms in the past. The problem I have with the amount of prototyping is putting all these avenues of software development together to form some cohesive, in-working-order app. I would have liked to have prototyped some form of program that incorporates at least a couple of aspects I plan to implement.

I went into the implementation phase of this project knowing to manage my time better and gave this project the time it needed. I planned out a sprints every week to ensure goals were completed.

I spent a lot of time trying to make the app work for mobile. If I could go back knowing what I know now, I would have spent more time trying to develop a server to host my machine learning classifier and graph generating software as these caused issues when porting to mobile. However, I ended up having a completely functional desktop application instead of mobile due to library incompatibilities.

The machine learning, which was the top priority of this project was a huge success as I was able to implement five different techniques to classify the music (Neural Network, Decision Trees, Random Forest, K-Nearest Neighbour and Naive Bayes). I initially thought I would only use one and maybe test another, but five gives a much more rounded outcome.

The application is basic but functional so I deem it a success. It can only read internal music files but it does an excellent job displaying information back to the user.

Cloud services were looked into heavily but my expertise was lacking in this field to say the least, and therefore this aspect let the project down the most.

7.3 Conclusion

To sum up what I have learned and how I implemented what I've learned I will discuss below:

- I learned what the most suitable machine learning algorithm is for my problem. This being a Neural network due to the nature of music and songs crossing genres

constantly. It allows for high confidence percentage predictions and more than likely give the best results. Neural networks given the hidden layers made for a highly accurate classifier for music.

- I tested out other algorithms because I've learned that machine learning algorithms are versatile. Algorithms such as Decision Trees and K-Nearest Neighbour surprised me with the accuracy they were getting.
- Naive Bayes was not amazing at predicting at all. It rarely got the correct overall genre but I left it in the process as it may pick up on something the other algorithms did not.
- I learned that if you don't plan and manage time you can fall behind. This was rectified in the implementation phase with proper sprint planning every week.
- I have concluded that music recognition is not only possible but extremely doable due to my findings.
- Kaggle was an invaluable source of datasets, a free dataset host where I came across an 1000 song dataset, all with assigned genres. I figured out the values they were working with and matched the extraction process to be able to accurately classify any piece of music.

7.4 Future Work

If I come back to this project after graduation I have a list of features that I would like to implement but aren't really feasible at this point in time. These are the plans I have in mind to make the app more user friendly:

- Adding accounts might be a feature that customers want as I will be putting this application on the Google Play store once there is a working build, for public use.
- If the application picks up any substantial traffic, implementing Google Ads is definitely a goal for the future but not so important for a college project.
- I also have plans to make this application use an API from Shazam and get the song name automatically. This goal is not in my initial goals because it is not necessary for the proof of concept application I am going for. It would definitely be a core feature that I would like to work on at a later time.
- Of course, a huge goal would be developing this for iOS. The iOS marketplace is a huge market but I do not have the programming tools for developing iOS and I am unfamiliar with their code base as of now.

The app will be public on GitHub so if any changes were looked for by users, an open source policy would be the way I go about developing this. I will be the only developer up to the point of submission however.

Bibliography

[1] S. C. R. Lab and F. Pachet. Daddy's car: a song composed by artificial intelligence - in the style of the beatles. Youtube. [Online]. Available: https://youtu.be/LSHZ_b05W7o

[2] carykh. Ai evolves to compose 3 hours of jazz! Youtube. [Online]. Available: https://youtu.be/nA3YOFUCn4U

[3] ——. Carykh drive. Google. [Online]. Available: https://drive.google.com/drive/folders/0B98DkR2AsGZaYk9JNFJScWRnMTQ

[4] SAS. Sas machine learning. SAS Institute Inc. [Online]. Available: https://www.sas.com/en_ie/insights/analytics/machine-learning.html

[5] D. of Science. Map of computer science. Youtube. [Online]. Available: https://www.youtube.com/watch?v=SzJ46YA_RaA

[6] H. Reese. Understanding the differences between ai, machine learning, and deep learning. TechRepublic. [Online]. Available: https://www.techrepublic.com/article/understanding-the-differences-between-ai-machine-learning-and-deep-learning/

[7] Wikipedia contributors, "Artificial intelligence — Wikipedia, the free encyclopedia," https://en.wikipedia.org/w/index.php?title=Artificial_intelligence&oldid=876255399, 2019, [Online; accessed 2-January-2019].

[8] Machine learning definition. Expert System. [Online]. Available: https://www.expertsystem.com/machine-learning-definition/

[9] J. Spacey. 3 examples of supervised learning. Simplicable. [Online]. Available: https://simplicable.com/new/supervised-learning

[10] ——. 3 examples of unsupervised learning. Simplicable. [Online]. Available: https://simplicable.com/new/unsupervised-learning

[11] F. Doukkali. Simple explanation of semi-supervised learning and pseudo labeling. Medium. [Online]. Available: https://towardsdatascience.com/simple-explanation-of-semi-supervised-learning-and-pseudo-labeling-c2218e8c769b

[12] S. Bhatt. 5 things you need to know about reinforcement learning. KDnuggets. [Online]. Available: https://www.kdnuggets.com/2018/03/5-things-reinforcement-learning.html

[13] J. Mahanta. Introduction to neural networks, advantages and applications. Medium. [Online]. Available: https://towardsdatascience.com/introduction-to-neural-networks-advantages-and-applications-96851bd1a207

[14] A. Dertat. Deep neural network graphic. Medium. [Online]. Available: https://towardsdatascience.com/applied-deep-learning-part-1-artificial-neural-networks-d7834f67a4f6

[15] M. Nielson. Chapter 5 - why are deep neural networks hard to train? [Online]. Available: http://neuralnetworksanddeeplearning.com/chap5.html

[16] J. Le. A tour of the top 10 algorithms for machine learning newbies. Medium. [Online]. Available: https://towardsdatascience.com/a-tour-of-the-top-10-algorithms-for-machine-learning-newbies-dde4edffae11

[17] K nearest neighbors - classification. [Online]. Available: https://www.saedsayad.com/k_nearest_neighbors.htm

[18] S. Ray. 6 easy steps to learn naive bayes algorithm. Analytics Vidyha. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

[19] O. Wibisono. Classification (machine learning): When should i use a k-nn classifier over a naive bayes classifier? Quora. [Online]. Available: https://www.quora.com/Classification-machine-learning-When-should-I-use-a-K-NN-classifier-over-a-Naive-Bayes-classifie

[20] T. L. Li, A. B. Chan, and A. Chun, "Automatic musical pattern feature extraction using convolutional neural network," in *Proc. Int. Conf. Data Mining and Applications*. sn, 2010.

[21] Guide2Research. Top conferences for machine learning arti. intelligence. Guide2Research. [Online]. Available: http://www.guide2research.com/topconf/machine-learning

[22] I. Spectrum. Boston dynamics' atlas robot shows off parkour skills. IEEE. [Online]. Available: https://spectrum.ieee.org/automaton/robotics/humanoids/boston-dynamics-atlas-robot-shows-off-parkour-skills

[23] NIPS. Nips 2018. NIPS. [Online]. Available: https://nips.cc/

[24] Y. Han, J. Kim, and K. Lee, "Deep convolutional neural networks for predominant instrument recognition in polyphonic music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 208–221, Jan 2017.

[25] G. Widmer, "Special issue on machine learning in and for music." *Machine Learning*, vol. 65, no. 2/3, pp. 343 – 515, 2006. [Online]. Available: http://search.ebscohost.com.cit.idm.oclc.org/login.aspx?direct=true&db=ofm&AN=501348299&site=ehost-live

[26] O. Y. Ali Karatana, "Music genre classification with machine learning techniques."

[27] Introduction to machine learning. [Online]. Available: https://ebookcentral.proquest.com/lib/cit-ebooks/reader.action?docID=3339851

[28] Various. Glassdoor ireland-machine-learning-jobs. Glassdoor. [Online]. Available: https://www.glassdoor.ie/Job/ireland-machine-learning-jobs-SRCH_IL.0,7_IN70_KO8,24.htm

[29] r/machinelearning. Reddit. [Online]. Available: https://www.reddit.com/r/MachineLearning

[30] Stack overflow developer survey 2018. Stack Overflow. [Online]. Available: https://insights.stackoverflow.com/survey/2018/

[31] . B. . Blue. But what *is* a neural network? — deep learning, chapter 1. Youtube. [Online]. Available: https://youtu.be/aircAruvnKk

[32] Aws android documentation. [Online]. Available: https://aws.amazon.com/developers/getting-started/android/

[33] matplotlib.org. Mathplotlib pie chart graphic. matplotlib.org. [Online]. Available: https://matplotlib.org/gallery/pie_and_polar_charts/pie_features.html

[34] Wikipedia contributors, "Scrum (software development) — Wikipedia, the free encyclopedia," https://en.wikipedia.org/w/index.php?title=Scrum_(software_development)&oldid=876610820, 2019, [Online; accessed 3-January-2019].

[35] How we effectively use trello for project management. WPCurve. [Online]. Available: https://wpcurve.com/trello-for-project-management/

[36] Spotify + the machine: Using machine learning to create value and competitive advantage. [Online]. Available: https://rctom.hbs.org/submission/spotify-the-machine-using-machine-learning-to-create-value-and-competitive-advantage/

Appendix A

Code Snippets

```
# SKLearn's K Nearest Neighbour algorithm
def knn():
    myData['total_alcohol'] = myData['Dalc'] + myData['Walc']
    myData['total_alcohol'] = myData['total_alcohol'].astype(int)

    threshold = sum(myData.total_alcohol) / len(myData.total_alcohol)
    myData['alcohol_level'] = [1 if i > threshold else 0 for i in myData.total_alcohol]

    y = myData['failures']
    X = myData.drop(['failures'], axis=1)

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)

    kNN = KNeighborsClassifier(n_neighbors=5)

    kNN.fit(X_train, y_train)
    y_pred = kNN.predict(X_test)
    print(accuracy_score(y_test, y_pred))

main()
```

```
pro2
D:\Python(ML)\Project2\venv\Scripts\python.exe D:/Python(ML)/Project2/pro2.py
D:\Python(ML)\Project2\venv\lib\site-packages\sklearn\externals\joblib\externals\cloudpickl
  import imp
Scikit K Nearest Neighbour:
0.8237547892720306

Process finished with exit code 0
```

FIGURE A.1: KNN w/ SciKit Learn

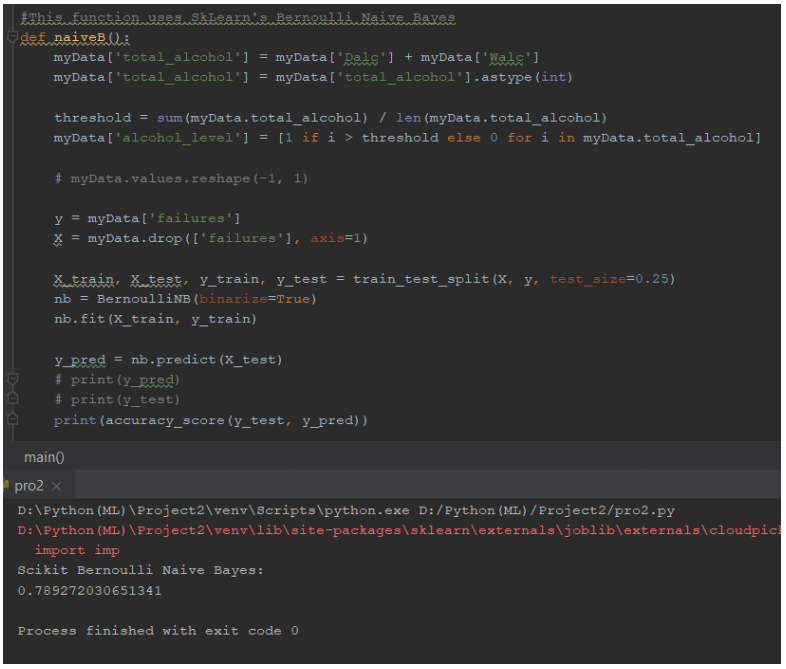

FIGURE A.2: NB w/ SciKit Learn

Appendix B

Wireframe Models

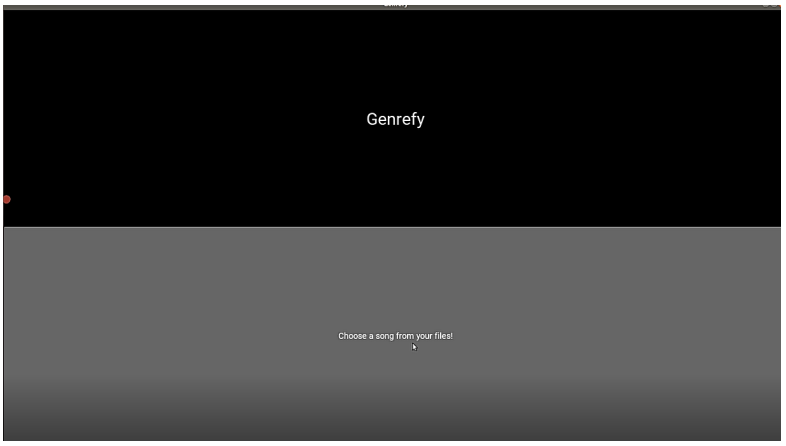

Figure B.1: App Home

Figure B.2: App File Chooser

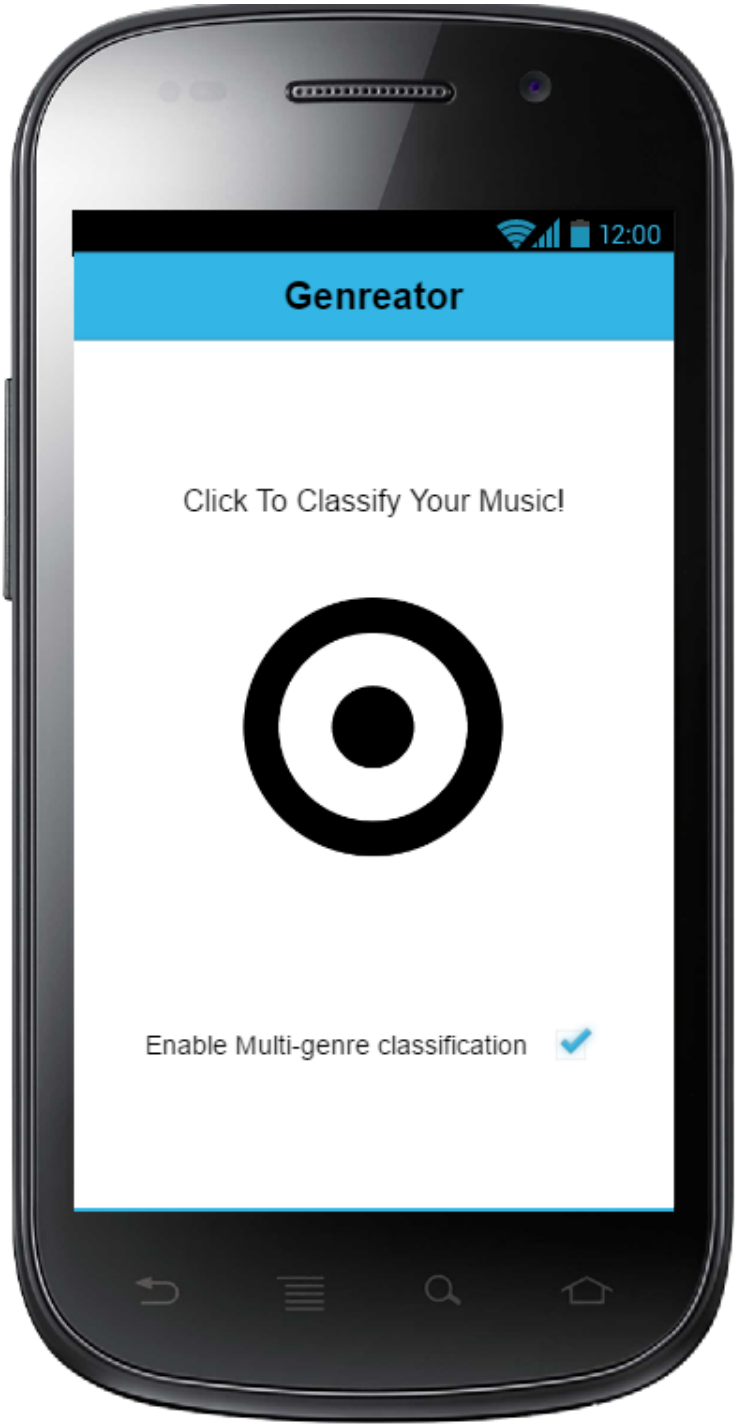

FIGURE B.3: Home Phone

FIGURE B.4: Loading Phone

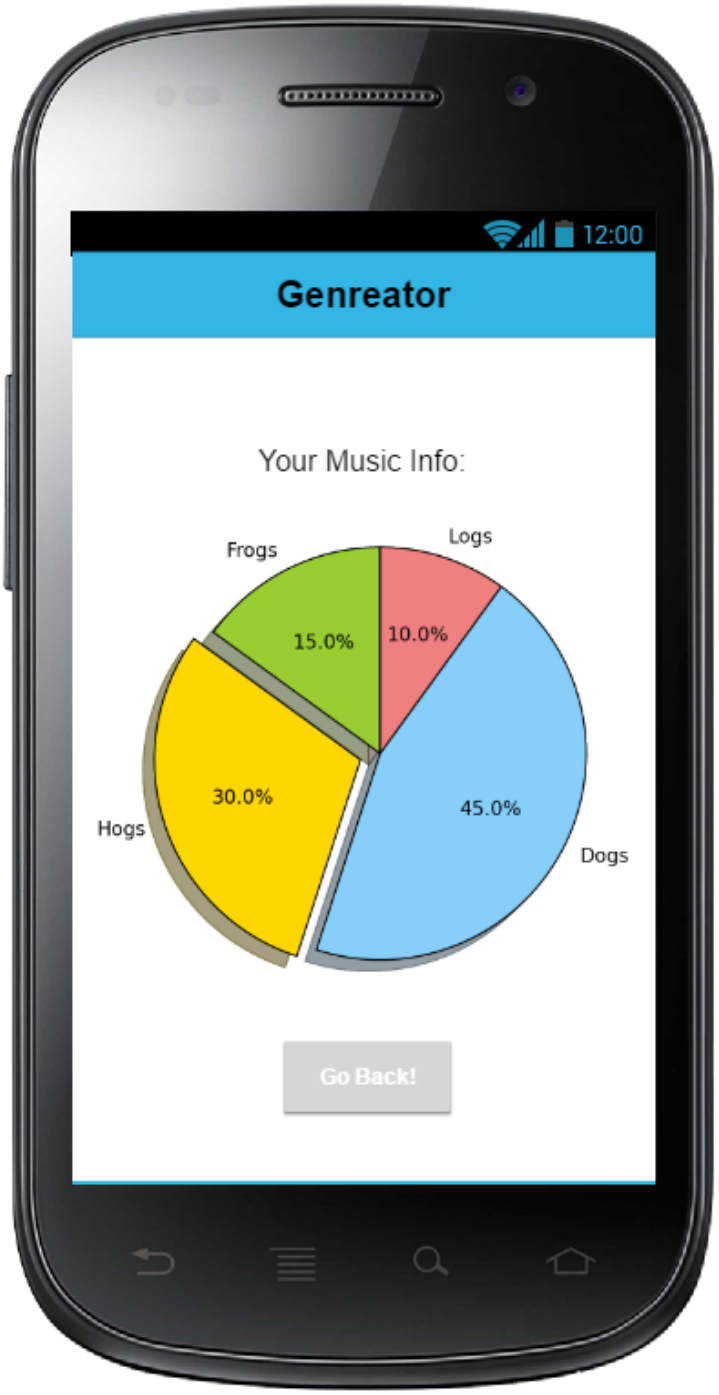

FIGURE B.5: Pie Chart Phone