Artificial Intelligence Incorporated into Audio Analysis of Electronic Music

Ned Fellenor

Let us know how access to this document benefits you

Recommended Citation
This Thesis is brought to you for free and open access by the Faculty of Arts, Humanities and Business Theses at PEARL. It has been accepted for inclusion in School of Society and Culture Theses by an authorized administrator of PEARL. For more information, please contact openresearch@plymouth.ac.uk.
Copyright ©2022 Ned Fellenor

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the author’s prior consent.
ARTIFICIAL INTELLIGENCE INCORPORATED INTO AUDIO ANALYSIS OF ELECTRONIC MUSIC

by

NED FELLENOR

A thesis submitted to the University of Plymouth in partial fulfilment for the degree of

RESEARCH MASTERS

School of Society and Culture

March 2023
Acknowledgements

This research project has very much been a personal endeavor. I jumped at the opportunity to attend this Computer Music course when I learned Plymouth University offered it. Music has been a core component of my life since early childhood. As a young boy I remember listening to abstract electronic music from the likes of Air and Groove Armada as my parents would play it in the kitchen. I would wonder how these sounds were created, and at the time, I did not realize this would shape my entire outlook on music.

Now in the present, my love for electronic music is strong and I desire to express this to others through producing my own music as well as performing live DJ sets. This early-developed interest for music combined with my background of computer science and an analytical mindset has urged me to delve into the technical aspects of digital music. This is where my research project stems from. I am researching into machine-learning based techniques applied to electronic music in the hopes my knowledge of the two fields can form higher potential for music systems in the future. As a user of DJ software, I have learned that one of the most time-consuming activities within it is organizing and forming playlists of music tracks. This research project aims to provide any information that could make this process more efficient.

I would like to thank my mentors Prof. Eduardo Miranda and Dr. David Moffat for supporting me throughout this thesis, especially considering the pressures ensued by remote working. Their valuable contact and personal interest in this topic have motivated me to see this project to the end. I am grateful to have been part of the ICCMR at Plymouth University and it has been interesting seeing the projects fellow course-mates have worked on.

I would also like to thank my father, Dr. John Fellenor, for guiding me through my university experience and being there for me in times where I really struggled. His academic background has given me an extra motivational boost as if it wasn’t for him, I maybe wouldn’t have seen the true value in academia.
Author’s Declaration

At no time during the registration for the degree of Research Masters has the author been registered for any other University award without prior agreement of the Doctoral College Quality Sub-Committee.

Work submitted for this research degree at the University of Plymouth has not formed part of any other degree either at the University of Plymouth or at another establishment.

A programme of advanced study was undertaken, which included Advanced Topics in Computer Music and Research Skills in the Arts, Humanities and Business.

Ned Fellenor, BSc Computer Science.

Student Reference Number – 10581049

Word count of main body of thesis: 18525

Signed: ____________________________

Date: 03/01/2023
Abstract

Artificial Intelligence Incorporated into Audio Analysis of Electronic Music

A thesis produced by Ned Fellenor

In recent decades the live music performance and DJ industry has undergone a transition from analog to digital music being the norm. Audio analysis and mixing software is now a necessity for modern DJs and their preferred equipment. However, it still stands that DJs must thoroughly learn their tracks and decide which transition well together in order to produce a smooth mix. This project looks at how machine learning-based systems can aid DJs in this process; it raises questions about the use of artificial intelligence within a heavily subjective form of live performance, and its potential effectiveness in this field. This project will also assess the nature of varying electronic genres and how they affect one another. Preliminary research will be conducted to assess how musical genre is determined, and how machine learning can utilize track features provided by Spotify research data to predict genres of electronic music. A KNN-based electronic genre prediction system will be developed from this, and its capabilities assessed.
# Table of Contents

Acknowledgements ........................................................................................................................................... 2
Author’s Declaration ......................................................................................................................................... 3
Abstract ............................................................................................................................................................ 4
List of Figures .................................................................................................................................................. 7
List of Tables & Matrices ................................................................................................................................. 8

1 – Introduction .................................................................................................................................................. 9
   1.1 – Research Questions ............................................................................................................................... 9
2 – Critical Evaluation of Pre-Existing Research .......................................................................................... 9
   2.1 – Research Topic Background ............................................................................................................... 10
   2.2 – Identifying the Research Gap Through Importance of Proposed Research .................................... 10
3 – Technologies, Research Methodologies & Legality ............................................................................... 11
   3.1 – Legal & Ethical Compliance ............................................................................................................... 11
   3.2 – Technologies to Aid Research & Data Collection ............................................................................. 11
      3.2.1 – Data Collection Software & APIs ............................................................................................... 11
      3.2.2 – MATLAB Integrated Development Environment (IDE) .......................................................... 12
   3.3 – Quantitative, Qualitative & Comparative Research Methods ............................................................. 12
4 – Research & Development Process ......................................................................................................... 13
   4.1 – Preliminary Studies to Aid Research .................................................................................................. 13
      4.1.1 – Understanding Spotify Features from Pilot Study (Feature Selection) ..................................... 13
      4.1.2 – Guidance Based on Pilot Study Findings .................................................................................... 18
      4.1.3 – Curse of Dimensionality .............................................................................................................. 21
      4.1.4 – Dimensionality Reduction ........................................................................................................... 22
   4.2 – Suitable Dataset .................................................................................................................................. 25
      4.2.1 – Forming a Spectrum of Electronic Genres .................................................................................... 26
      4.2.3 – Dataset Size Based on Collated Genres and Spotify Features .................................................. 29
      4.2.4 – Genre Selection Process .............................................................................................................. 29
      4.2.5 – Use of Generalized Spotify Analyzer (GSA) to Generate Track Feature Datasets .................. 31
   4.3 – Spotify Feature Analysis Using KNN ................................................................................................... 32
      4.3.1 – Developing KNN-based Algorithm ............................................................................................... 34
   4.4 – Initial Data Analysis ............................................................................................................................. 42
      4.4.1 – Generating & Collecting Data ........................................................................................................ 42

5
List of Figures

4.1.1 - Figures 1 & 2: Comparison of K-means clustering graphs
4.1.1 - Figures 3 & 4: Comparison of K-means silhouettes
4.1.1 - Figures 5: 3D projection of K-means analysis
4.1.1 - Figures 6: Example graphs of spread of normalized distances
4.1.3 - Figure 7: Equation for adjusted R-squared value
4.2.2 - Figure 8: Graph representing spectrum of electronic music genres
4.2.5 - Figure 9: Code extract of .pkl to .csv conversion function
4.2.5 - Figure 10: Screenshot of sub-genre .xlsx dataset
4.3.1 - Figure 11: Euclidean distancing formula
4.3.1 - Figure 12: Flowchart of independent variables in relation to KNN model
4.3.1 - Figure 13: Code extract of genre indexing function
4.3.1 - Figure 14: Code extract of tempo normalization
4.3.1 - Figure 15: Code extract of training & testing dataset preparation
4.3.1 - Figure 16: Flowchart of raw data to training & testing dataset process
4.3.1 - Figure 17: Code extract of PCA coefficient matrix function
4.3.1 - Figure 18: Code extract of mean array function
4.3.1 - Figure 19: Code extract of standard deviation array function
4.3.1 - Figure 20: Code extract of covariance and correlation coefficient matrices functions
4.3.1 - Figure 21: Code extract of KNN classifier function
4.3.1 - Figure 22: Code extract of confusion matrix function
4.3.1 - Figure 23: Code extract of average classification accuracy function
4.3.1 - Figure 24: Code extract of Ko function
List of Tables & Matrices

4.2.3 - Table 1: Dataset genre to sample size comparison

4.2.4 - Table 2: Table of parent & sub-genres, popularity ranking & playlist ID

4.4.2 – Table 3: Table of dataset mean values

4.4.2 – Table 4: Table of dataset standard deviation values

4.4.3 - Table 5: Table of parent genres covariances

4.4.3 - Table 6: Table of sub-genres genres covariances

4.4.3 - Table 7: Table of parent genres correlation coefficients

4.4.3 - Table 8: Table of parent sub-genres correlation coefficients

4.4.3 - Table 9: Table of parent genres PCA coefficients

4.4.3 - Table 10: Table of sub-genres PCA coefficients

4.4.4 - Table 11: Table of parent genres K-values and prediction accuracy

4.4.4 - Table 12: Table of sub-genres K-values and prediction accuracy

4.4.4 - Table 13: Confusion matrix of parent genre predictions

4.4.4 - Table 14: Table of reduced dataset prediction accuracies
1 – Introduction

This research project aims to contribute to developments in technology involved in live music performance, specifically DJing electronic music. It takes a majorly quantitative approach to analyzing a controversial and subjective field. Research into the nature of electronic music genres and ML-based prediction of such genres will be conducted in order to provide a foundation for any potential user-oriented systems developed or research conducted beyond this project. An effective ML-based all-encompassing analytical program will be detailed, as well as its development within MATlab and its use justified. Data collection and processing will be evaluated within this document; how the necessary data is obtained from Spotify and the decision as to what attributes will be implemented into the program will be assessed.

Key research points inferred by the outcome of the KNN algorithm, and its component analytical functions will be used to evaluate the success of the project and how much potential this machine learning algorithm has concerning the subjective topic of electronic music. Pre-existing qualitative data will be referred to throughout to justify discoveries made. Upon reflection, potential improvements and unforeseen hinderances will be addressed.

1.1 – Research Questions

The focal point of this study is to assess how capable a machine-learning based algorithm is at predicting genres of electronic music. This project aims to focus on electronic music specifically due to the pragmatic application it seeks to endorse, which is to provide a point of further development for DJ software. A subset of conversations stem from this purpose, one most notably being how are genres of music perceived, determined and how is the complex tree of electronic music genres justified?

Academically, music is often researched from a qualitative and human perspective. The nature of music certainly entails human qualities that arguably cannot be quantified; however, all music must share some fundamental composition. Questioning how music can be analyzed with a quantifiable approach spawns potential for studies to be performed such as this one. Considering we are in a digital age of music, it is mostly performed and enjoyed through software executed on computerized systems. These systems do not have capacity for human perception of music and so are limited to digital, quantified information. How can these values be expanded on software systems? Will the implementation of AI open doors to higher efficiency in music performance?

2 – Critical Evaluation of Pre-Existing Research

This chapter will examine pre-existing research into topics this project concerns. It will facilitate understanding the proposed work in relation to existing literature and what new discoveries can be made, as well as provide a gauge on how arguments for and against the research question may affect subsequent findings.
2.1 – Research Topic Background
The electronic dance music scene has exploded on a global scale since the 1990’s with the advent of inexpensive digital DJ equipment, such as all-in-one DJ controllers that simply require a laptop to work. The rise of highly accessible equipment means anyone can now DJ from the comfort of their home. Prior to this, DJs relied on connections to clubs and studios which owned their own, less versatile equipment. As hardware has evolved, so has the software that provides the functionality behind it. The history of DJing and electronic music is detailed in the book *Last night a DJ saved my life: The history of the disc jockey* (Brewster, 2014).

Development of DJ Software
Nowadays, DJ software such as Rekordbox and Serato can simply be run on a laptop rather than from the actual standalone DJ hardware. They provide a great deal of functionality to aid DJs in their mixes. Their development has certainly made the DJing process easier and more efficient. Visual aid takes pressure off using just audio queues to mix and the implementation of effects allows for smoother transitions within a mix. Despite this, DJs are still required to locate, analyze and determine themselves where they think every song belongs in their music library and what each song will mix best with. The use of a wide range of modern electronic genres to label music could be harnessed by machine-learning techniques to develop a system that takes the strain off musicians who must spend large amounts of time preparing their mix.

2.2 – Identifying the Research Gap Through Importance of Proposed Research

Why is this a pertinent question?
Ideally, the KNN-based machine-learning algorithm developed within this project will present a sufficient predictability accuracy in classifying genres of tracks. The success of this premise will provide a foundation for future ML track classification and playlist creation systems that may be implemented into DJing software.

Music is often defined in quantifiable and objective terms. Tempo, for example, comprises of a fixed numerical value. However, music can also be described in terms of subjective qualities, such as ‘energy’ or ‘danceability’. A given song will be favorably or unfavorably assessed by different individuals. As such, comparative, qualitative and quantitative data analysis is apposite.

Over recent years developments in technology have improved both the experience of DJing and music consumption. Technology that can further enhance these aspects is of import to the music industry; AI and machine-learning has already proven beneficial in other areas, such as Spotify’s AI-assisted user music recommendations.

There is existent research into genre classification, but not enough on electronic genres specifically and how it can benefit DJing oriented music and programs. Electronic music audibly, culturally and fundamentally differs from other genres of music and requires analysis, removed from the general musical consensus.
Research Literature Perspective

Artificial intelligence and machine-learning techniques are highly relevant topics in current academia as major advances have accelerated our understanding of the capabilities of these topics. They are becoming applicable to essentially all fields of research and in themselves are still developing. The application of modern machine learning techniques to traditional research methods and questions has generated a large amount of valuable information. There is little research into the compositional fundamentals of electronic music, especially from a quantitative stance. Electronic music is constantly developing and shows no sign of slowing down. It will undeniably remain relevant for the foreseeable future. Key information inferred from this project can provide guidance to the plethora of unanswered digital music questions that will inevitably be asked in the future. This project tackles specifically genres of electronic music, but the premise used here can concern electronic music in its entirety, such as the activity of mixing, effects used in performances and how various genres of music are produced.

Pragmatic Perspective

This research topic directly addresses a specified audience and popular activity. Electronic music is popular globally and varies massively. It is mixed on many different DJ systems in different environments, from city nightclubs, to festivals, to studios and even bedrooms. Research into advances on determining genres of electronic music for future implementation into DJ software is strongly supported.

3 – Technologies, Research Methodologies & Legality

3.1 – Legal & Ethical Compliance

This research project uses explicitly open-source scripts and research sources that allow for open use. Resources used are listed in section 3.2.1. This project evaluates through primarily quantifiable mathematical, analysis software-based techniques and does not require additional participants. There is no need for ethical approval of this study for this reason.

3.2 – Technologies to Aid Research & Data Collection

The proceeding section details the data collection resources used for this study. It also details the use of MATLAB to develop a Livescript program that encapsulates all analytical functions.

3.2.1 – Data Collection Software & APIs

Available links to the following information sources can be found in Open-source Code & Websites Used.

Spotify Research API

The Spotify research API was used to determine what track features could be collected using the GSA. Spotify’s definition of these features is available via the API website.
The Spotify research API provides data on track metadata and audience metrics. Track metadata describes the compositional data of a given track; this includes features that form this study’s datasets such as tempo, energy and valence. It also includes metrics a track such as name, artist, release date, duration and total plays or downloads. Track audience metrics include the number of plays by geographical breakdown and playlists that include concerned tracks. Similar metrics on Spotify playlists and albums are also provided.

*Every Noise*
This thesis used EveryNoise.com to identify all recognised Spotify genres and sub-genres and determine which genres should be selected for this study. These have been selected based on popularity in order to reflect the most dominant genres available.

*Generalized Spotify Analyzer*
GSA is an open-source script that sources track features for a given Spotify playlist. Opposed to using the Spotify research API itself to pull track data one track at a time, GSA allows for sourcing of large volumes of track feature data simultaneously. This maximizes efficiency analyzing genre-based Spotify playlists. Sourcing of track data within albums is also achievable through GSA.

3.2.2 – MATlab Integrated Development Environment (IDE)
MATlab is a data-analysis oriented procedural programming language that includes its own integrated development environment (IDE). It includes pre-built analysis functions, such as `fitcknn`, that enable reliable and efficient analysis. Self-made functions can be built on top of these analysis objects to form entire analysis scripts that perform a large range of analytical functionality. It is an ideal environment to develop script that will generate initial analytical values prior to running a main KNN method that will subsequently be evaluated with various techniques.

3.3 – Quantitative, Qualitative & Comparative Research Methods
Quantitative methods entail data collection and interpretation of scalable data, e.g., track tempo, and is predicated on theory testing; statistical techniques are characteristically employed. Qualitative methods focus on non-numerical data, such as talk in interaction, often analyzed inductively to understand people’s experiences etc. (Fielding et al. 2001). This thesis is primarily quantitative in nature, qualitative methods and research points may be used to justify discovered quantitative information.

In many disciplines, qualitative approaches form the first step to exploring new phenomena; with quantitative methods subsequently used to explore hypotheses about specific aspects of those phenomena; nominally deemed ‘mixed’ methods. ‘Comparative approaches’ afford an overall flexible and generally applicable set of methods that enables comparisons to be drawn between [typically] different populations. In the present study, while different populations are not involved in the typical sense, the work can be deemed comparative because the comparison between K-nearest neighbor is compared with the pre-determined human attribution of track genre.
4 – Research & Development Process
This chapter forms the main body of this thesis. It details preliminary research conducted to support the project, the development of the datasets, KNN MATLAB program as well as testing and critical evaluation.

4.1 – Preliminary Studies to Aid Research
This section will detail research undertaken prior to data collection and the main body of research conducted relating to the developed MATLAB KNN analysis program. Frequently referred to terms, features and pre-existing data will be provided with context here.

4.1.1 – Understanding Spotify Features from Pilot Study (Feature Selection)
The fundamental components of all electronic music can be used to hypothesize which Spotify features may have the greatest influence on defining electronic genres. A variance in values for a given feature provides greater distinguishability for a KNN algorithm interpreting them. This hypothetically should increase the KNN algorithm’s ability to correctly predict genres of tracks, making said feature more desirable.

It should be noted that most Spotify track features are valued between 0 and 1, where 1 expresses the greatest correspondence to the given feature. When describing the values of features, this scale can be assumed; the only exceptions are tempo, which is valued as beats per minute (BPM) and loudness using a metric of decibels (dB). A large benefit of track features that scale as 0-1 is the normalization process of raw data is a lot simpler.

How Does each Spotify Feature Apply to Electronic Music?
Every Spotify feature is all-inclusive of differing genres of music, electronic music seems to favor certain features, especially those highlighted in the pilot study. The following features that are detailed present significant influence on determining genre justified by findings within the pilot study and their relation to dimensionality reduction techniques. Justification of use of a wider range of the available features will follow.

Tempo
Tempo is a fundamental component within music. Usually measured in beats per minute, tempo dictates the speed at which a song should be ideally played. Within DJing adjustable tempo is essential as this parameter needs to be modified to beat-match tracks. This is often the first step a DJ takes after selecting a song to mix into their ongoing performance.

Parent genres of electronic music vary in tempo whereas sub-genres characteristically share the same tempo range. For example, House music contains a range of sub-genres such as ‘Minimal’, ‘Deep’ and ‘Tech House’; distinguishable from another by a trained ear, but all sharing a tempo of around 115-128 Beats per minute (BPM). This implies that tempo is a defining, tangible parameter within genres of music. Drum and Bass, an extremely popular family of genres in the UK, has a tempo range generally around 160-
180 BPM, about 40+ BPM above House music. It can be interpreted that the tempo sub-genres of music are bound within the tempo range defined by their more generalized parent genre.

Previous research indicates listeners can consistently and accurately predict the tempo of presented tracks and can make consistent tempo judgments as well as that the optimal tempo varies across extracts, making tempo a key parameter to use within this study. Moreover, rhythm also plays a role, but not by itself, in making temporal judgments. (Quinn, et al. 2006). Further, the use of tempo within this study is supported by the fact that many electronic genres of music include a strong, continuous rhythm which may increase a listener’s ability to estimate tempo. DJs are likely able to recognise and judge temporal differences in music, given that doing so is an implicit ‘tool’ of the trade. Within Spotify’s available research data, tempo values are presented as their default BPM value, as tempo values in this sense are quantifiable.

The pilot study highlighted the influence tempo has on several other Spotify track features, it is integrated into the calculation of energy and danceability which in themselves have been proven to be useful parameters. Tempo data collected from Spotify tracks used within the pilot study support the concept that different genres and their subparts are based around fixed ranges of values. Tempo is unavoidably a fundamental component in defining electronic genres and so will be a forerunner in KNN analysis using varying combinations of dimensions.

Valence

Spotify defines the valence to be the positiveness conveyed by the song. The greater the value, the more positive sounding the track. Human perception of melodies and textures of sound heavily affect valence. Minor chords induce a feeling of sadness or tension, whereas major chords elicit a more positive and happy emotional response. Studies suggest a link between valence and tempo also. For example, valence can be combined with a subjective arousal effect by altering tempo or instrumentation such that musical pieces are judged as more arousing with a fast tempo, and with an orchestral rather than a piano timbre (Droit-Volet, et al. 2013). This indicates that collected valence values may positively correlate with tempo.

PCA scores of valence values were used in the K-means analysis conducted within the pilot study. It was found to be a valuable parameter to use due to its range in average values of each genre, allowing for greater distinction between them. “The valence averages calculated show a strong correlation with pre-existing knowledge of the positivity/negativity of selected genres. Neurofunk is shown to have a low valence of 0.233; it is generally dark and negative sounding, whereas Liquid has an average of 0.593.” Two sub-genres of Drum and Bass that are structurally similar (same tempo range, beat pattern and danceability) are made distinguishable due to their valences.

Energy

Spotify defines the energy of a song as ‘a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy.’ As the tempo of electronic music can vary greatly, one can assume the energy value of such genres are similarly variable. Deep/Minimal House is considered less energetic than Techno or Tech House, hence, the energy value of tracks is a valuable parameter in modelling data.
Findings within the pilot study suggest electronic genres are on the more energetic side of music, and that a faster tempo and greater fullness has a marked influence on energy. The range in averages of energy values by genres previously tested was found to be 0.428; Neurofunk being the highest at 0.923 and UK Dubstep being the lowest at 0.495. Neurofunk scored high because of its instrumental fullness and high tempo. UK Dubstep was likely to be the lowest scorer because of its spaced out, slower beats. House and Techno genres may have scored higher than UK Dubstep, despite being slower, due to the fullness provided by the heavy rhythmic drums.

Danceability
The danceability of a track is described by Spotify as ‘how suitable a track is for dancing, based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity’.

It should be noted genres of electronic music lend themselves to this definition of danceability more so than mainstream and other alternative genres, e.g. ‘Heavy Metal’. Electronic music in general is often referred to as ‘dance music’, listeners tend to dance for much longer periods and in a more meaningful, confident manner. The following statement reflects that rave and dance culture was ignited by the advent of electronic sounds; Techno being a key influence.

“Little did I realize that just around the corner loomed a psychedelic dance culture, that the instruments and time-space coordinates of the neopsychedelic resurgence would not be wah-wah pedals and Detroit 1969, but Roland 303 bass machines and Detroit/Chicago 1987.” (Reynolds, 2013).

Comparison of Energy & Danceability
With reference to Spotify’s definition of Energy and Danceability, the pilot study concluded energy would provide greater differentiation between genres than danceability, due to how it is calculated, e.g., it incorporates more tangible features in its calculation; rather than just tempo and rhythm, intensity and loudness are also implicated. The latter parameters are deemed to be more useful in identifying differences between sub-genres as the rhythm is often identical within them as similar beat patterns are used. However, the intensity between sub-genres still varies; ultimately providing greater distinction between them.

An example of sub-genres maintaining the same rhythm but having different intensity and atmosphere would be a comparison between ‘Liquid Drum and Bass’ and ‘Neurofunk Drum and Bass’. Both genres use the classic amen breakbeat seen in almost all Drum and Bass genres, and both are played at a tempo of 160-180BPM, yet Liquid sounds lighter and more positive whereas Neurofunk is considered one of the darkest and most intense genres of electronic music.

A lack of differentiation between electronic genres would be witnessed when applying alone beat strength and overall regularity which are core components of danceability. Within the pilot study the average danceability of Deep House, Tech House and Techno was found to be 0.768, 0.806 and 0.745 compared to energy values of 0.616, 0.882 and 0.785. All these genres are defined by a strong four-to-the-floor beat with little beat variation throughout, resulting in a smaller spread of danceability averages than energy. Genres that follow the same audibly dominant and regular beat structure as those previously stated form
a large percentage of the electronic genre spectrum, making danceability less effective within research when used independent of energy.

When concerning K-means clustering, this scenario was true; 3-dimensional clustering meant a limit of 3 variables disallowed the incorporation of both energy and danceability. As explained in the previous section, K-means clustering is oriented around the 2–3-dimensional visualization of values, restricting the range of input variables per evaluation instance. Even when using principal component coefficients to condense relations between variables, this point still stands. This limitation forced the decision to use only the PCA score values of energy. With the ability to incorporate more dimensions into KNN analysis, both energy and danceability values could be used as variables. KNN analysis using both parameters should be addressed as their nature entails genre is directly affected by them.

Loudness

The loudness of a track is measured in decibels (dB). Spotify provides a technical description as ‘...the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typical range between -60 and 0 dB.’ The loudness of a song can be determined by its production and mastering process; tracks may be subjectively perceived as loud, but the calculated amplitude may not reflect this. Hence, loudness has been deemed an unreliable parameter to use in data analysis, findings within the pilot study support this.

Loudness values calculated did not seem to show any correlation. All genres but Neurofunk contain average values between -7 to -9 dB. The included genre, Neurofunk was perceived to be a lot louder than other genres at -2.941 dieback average. What makes this feature less reliable is the range in loudness values per genre suggesting a weak correlation between them.

Speechiness

The following 3 features are scaled between values 0 and 1. It should be noted that there are key intervals between these values that more definitively express the nature of the song.

Spotify states that for Speechiness, ‘Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.’ Electronic music tends toward an instrumental focus; some genres more-so than others. ‘Nu Disco’ characterizes higher speechiness values compared to other electronic genres as it stems from the vocally oriented genre, classic Disco (Echols, 2010). Additionally, the use of this parameter may be debated due to the fact any value above 0.66 should depict essentially an audiobook, according to Spotify’s definition. Theoretically all values collected for speechiness should range between 0-0.66 resulting in a smaller range of predicted values, ultimately providing less contrast between selected genres.

The pilot study concluded that the average speechiness of each genre was low. Neurofunk had the highest average at 0.173. Nu Disco had a lower average speechiness than Neurofunk, despite the selected Neurofunk tracks being distinctly instrumental compared to the Nu Disco tracks. The predicted cause for this was that the algorithm responsible for calculating speechiness mistakes some frequencies of
electronic music to be vocals, e.g., “Omnivore” by Noisia and the Upbeats is entirely instrumental yet had a speechiness of 0.264.

**Acousticness**
The fundamental composition of electronic music does not aid itself to high acoustics. The acoustics of a song are normally determined by the quality of sounds produced by physical instruments, rather than digital sounds which are often used in electronic music. Digitalized music experiences a loss in quality during compression. While not entirely noticeable to the human ear, the use of .mp3 and .wav samples in electronic music production will negatively impact acousticness. Data collected within the pilot study supports this as genres across the entire dataset presented low acoustic averages, resulting in a low potential in KNN analysis.

**Instrumentalness**
Spotify states that for Instrumentalness, ‘The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.’ This feature is a good descriptor for electronic genres of music as the act of DJ-ing relies heavily on the beat patterns and instruments used within a track. For a seamless mix, the beat pattern of two blended tracks should be similar or identical; the strength and clarity of the beat is inherited from the instrumentalness. Data collected within the pilot study concluded instrumentalness had a varying range of average values, however the range in values between tracks within genres was also vast. Calculated averages showed that Techno is highly instrumental, which is in-line with statement instrumentalness has high potential within KNN analysis.

Deep House had a range of 0.914. The reason being the track “House Every Weekend” by David Zowie, was graded 0.0 instrumentalness; potentially an anomaly caused by a failure in the instrumentalness algorithm’s calculation. This highlights the fact anomalies are possible in generated Spotify datasets. The use of a larger dataset will help to reduce unwanted skew caused by invalid data.

**Liveness**
The liveness of a track is defined as ‘the presence of an audience in the recording.’ The only way an original copy of an electronically produced song could have a presence of audience is if the producer explicitly sampled a recording of audience into the track during production. An example of this would be 2013 Drum and Bass track, “The Phantom”, by MUZZ and High Maintenance, which incorporates audience clapping and background noise despite it being an entirely electronically produced piece. The nature of DJ-ing means music is not played ‘live’, as a band might do, but more so as premade tracks digitally presented to the audience during the mix. In other words, the DJ represents a studio composed music via a turntable as an instrument (Rietveld, 2016). Following this concept, liveness data values may be considered to have low potential in KNN analysis of electronic genres.

Averages of liveness calculated from the pilot study dataset show that UK Dubstep had the highest rating at 0.324. With all other genres below this value, this does correlate with the understanding that all DJ-
oriented music pieces are essentially pre-made prior to performance. Little else can be derived from these values as liveness is shown to have weak or no relation to the other, more impactful features.

4.1.2 – Guidance Based on Pilot Study Findings
The conducted pilot study for this research project assessed the suitability of selected Spotify track features in terms of the influence they have on an AI-based algorithms ability to group concerning Spotify tracks by genre. It was concluded that Tempo, Energy and Valence features as parameters passed into a K-means clustering algorithm had significant influence on its grouping capabilities when comparing these to the associated genres of plotted datapoints. (Fellenor, 2021). This provides a starting point in forming a suitable dataset for use within a developed K-nearest neighbor algorithm; as a minimum, these parameters should be extracted from selected Spotify tracks of varying genres. A likely scenario is other available Spotify features will be passed into the KNN analysis algorithm as parameters to infer better comparison of accuracy outcomes and ultimately better the developed KNN-based systems genre-predicting capability. Further explanation of K-means and KNN, how the capability of K-means was limited and where KNN can overcome this follows.

Comparison Between K-means Clustering and K-Nearest Neighbor
In the conclusion of the pilot study, K-nearest neighbor was suggested as a better alternative for interpreting Spotify features than K-means clustering. The algorithms vary significantly in their general purpose and processes, and in order to justify the use of KNN within this study, a direct comparison between the two should first be made:

Introduction & Overview
The main purpose of K-means clustering is to group unclassified data values into clusters. It visualizes where clusters fall on an X,Y or X,Y,Z axis. K-means is a form of unsupervised learning meaning the descriptive ‘class/cluster’ value is not pre-existing and is determined by the cluster a given plotted value falls into. Only the number of centroids the K-means algorithm is required to plot is pre-defined, which determines the initial center of each cluster. In the first iteration, centroids are plotted at random coordinates. In the case of the pilot study, Euclidean distance is then used to calculate which centroid each datapoint is closest to. The centroid of each cluster is then re-calculated based on the total average of the position of all data points within the given cluster. Euclidean distance = \sqrt{((X_1 - X_2)^2 + (Y_1 - Y_2)^2)}, (Singh et al. 2013)

KNN Overview
The machine learning function in K-means occurs after data values have been plotted and is responsible for the designation of a cluster to plotted data points. This process presents a key difference between K-means and KNN; a cluster is applied to a general area of data points within K-means, however in KNN, each data object is individually assigned to a pre-defined class.

Supervised learning methods, such as KNN, require a dataset to be split into testing and training subsets. The training subset acts as a blueprint for the interpreted testing data. Supervised learning entails that data objects are designated to pre-defined classes. The KNN algorithm begins with a training phase where
the pre-determined class values are contained within the training subset. The set parameters and class values are passed into the algorithm which are used as classification guidelines when interpreting data within the testing subset during the testing phase. The test data subsets values are compared to the same parameters as that of the training data, and a class prediction can be made. Two general use-cases can be derived from KNN’s supervised learning process:

• A more pragmatic scenario where the test data subset has unknown class values and requires the assignment of a class, the KNN algorithm can be used to classify this data. The class value for each data object may be necessary in grouping data within a system, such as assigning a genre to a song in order to place it in the correct genre-based playlist.

• For research-oriented purposes, KNN can be used with datasets where the class value of every data object is known. The true class values of test subset data can be compared with the KNN algorithms class prediction to provide insight into the accuracy and capabilities of KNN’s classification ability as a machine learning algorithm. The analysis process is congruent with that of the pragmatic scenario; however, the ultimate intentions differ.

Limitations of K-means Clustering & KNN

One can insight that within K-means, clusters are highly dependent on the spread of plotted data. Data variables that lack any correlation analyzed by the K-means algorithm may present too vague clusters when visualized, reflecting a low accuracy. The spread of clustered data points visualized within the K-means analysis within the pilot study reflects this to a degree. As shown in figures 1 and 2, even when varying the number of clusters, there was significant overlap.

![Figures 1 & 2: Comparison of K-means clustering graphs](image)

Within both graphs displayed above, where K (number of clusters) = 7 and 3. Two main issues can be inferred:
An overlap of clusters is occurring suggesting overpopulation of assigned clusters, the optimal K value (Ko) is lower than K = 7 or 3. The K value in the K-means pilot study was used to reflect the number of genres relating to the plotted datapoints, at 7. The hypothesis was that each cluster should fall onto a region of plotted datapoints that are part of the same electronic genre, which was dependent on a correlation between the concerning variables. If there was an overlap, it expressed a reduction in the K-means ability to distinguish between genres. The K-means algorithm could only accurately cluster the datapoints into 3 or less clusters, suggesting it could only distinguish between more general characteristics of electronic genres.

For example, while K = 3, both subgenres of Drum and Bass, sharing a high tempo value, were clustered together presenting no distinction between them. But they were successfully grouped away from other, more drastically different genres that had a much lower tempo value, such as Tech House. Smaller differences between subgenres caused an overlap in datapoints, ultimately restricting the K-means capabilities. The interpretation of K-means silhouettes while K=7 and 3 within the pilot study reflects this concept. As seen in figures 3 and 4, K = 3 is visibly more optimal than K = 7. However, the 7 present genres are abstracted into only 3 groups, eliminating any distinction between some of these genres. This is detrimental to any system that aims to categorize data into genres of similar attributes.

For example, while K = 3, both subgenres of Drum and Bass, sharing a high tempo value, were clustered together presenting no distinction between them. But they were successfully grouped away from other, more drastically different genres that had a much lower tempo value, such as Tech House. Smaller differences between subgenres caused an overlap in datapoints, ultimately restricting the K-means capabilities. The interpretation of K-means silhouettes while K=7 and 3 within the pilot study reflects this concept. As seen in figures 3 and 4, K = 3 is visibly more optimal than K = 7. However, the 7 present genres are abstracted into only 3 groups, eliminating any distinction between some of these genres. This is detrimental to any system that aims to categorize data into genres of similar attributes.

The spread of plotted datapoints disallows for a clear line of best fit to be plotted in 2-dimensional space, expressing a lack in significant correlation between tempo and energy. This more so reflects an issue with the chosen dataset size and variable selection. The same issue inherently occurred when plotting a 3-dimensional Principal Component Analysis (PCA) plot using tempo, energy, and valence variables. PCA plots the correlations between variables onto 2- or 3-dimensional space presenting clusters of samples that share these correlations. A lack in positive or negative correlation between the 3 variables developed unclear clusters as seen in figure 5.
The case of the pilot study points out one major flaw in the use of K-means for this research; the actual genre is never directly involved in the machine learning algorithms analysis. Research points are only ever indirectly inferred from the K-means algorithms output response to Spotify features used as variables, that are just a few components of how a song’s genre is determined.

4.1.3 – Curse of Dimensionality

The curse of dimensionality dictates that within a data analysis or organization algorithm, as the number of dimensions (input variables) is increased, the number of required samples to maintain a given level of accuracy grows exponentially. Assuming all variables are binary, the number of required samples to maintain X samples per unique combination of variables would be $10 \times 2^k$, where $k$ is the number of binary variables, making $2^k$ the number of unique variable combinations for per sample.

If beyond 3 variables are involved within an instance of K-means, PCA is required to reduce dimensionality into 3 or less visualizable planes (X,Y,Z). Graphical representation of these clusters is the best method of expressing them hence K-means heavily relies on this process. As described before, only PCA scores of correlations between variables are plottable. KNN is not bound to this as the class prediction expresses all what K-means aims to present graphically. Working with beyond 3-dimensional space, or in data sampling terms 3 variables, is common within KNN. This introduces a new limitation concerning component variables of data samples that should be addressed, the curse of dimensionality.

An exponential increase in unique variable combinations means an exponential decrease in the number of closely neighbored samples in dimensional space, reducing the classification accuracy of analysis algorithms such as K-nearest neighbor. This is known as the ‘Distance Concentration’ problem within the curse of dimensionality. (Great Learning Team, 2020)
Figure 6 shows the decrease in the spread of normalized distances between data points in X dimensional space for a given scenario. As the number of dimensions increases, the standard deviation of Euclidean distance decreases and moves away from 0, indicating fewer closely neighboring data points.

This applies the scenario within this research project as the number of available input variables is determined by the number of informative track features provided by the Spotify web API; a total of 13. When implementing more input variables into the KNN algorithm during the development stage, the curse of dimensionality may manifest itself in an exhaustion of the increase of accuracy, potentially even to a point of decline, while using the same dataset size. This highlights the importance of selecting the most applicable track features when classifying into genres, as well as avoiding implementation of features that may increase the complexity of samples while hindering the classification accuracy of the KNN algorithm. There are standardized methods of feature selection and extraction that can be followed to mitigate this.

4.1.4 – Dimensionality Reduction
Dimensionality reduction offers numerous benefits to the formation of a dataset. As well as tackling the curse of dimensionality, the dataset will become inherently less complex meaning reduced processing time and required computational power. It can lower the chance of model overfitting, which occurs when a surfeit of variables causes the description of errors / noise in a model to obscure any relationship between variables in the model, hindering predictability and accuracy.

Feature Selection & Extraction
The following dimensionality reduction techniques are initially applied in section 4.1.1 and 4.3. The overall process of inspecting how each Spotify feature applies to genres of electronic music encapsulates feature
ranking in that the most influential features should be chosen, for example. Other mitigation methods are more so directly referenced. Reduction techniques fall into one of two categories, feature selection and feature extraction. Feature selection involves processes that aim to remove variables that have the least bearing on the predictability of a data model, while extraction aims to generate new variables that are the product of correlating pre-existing variables to be used instead of the originals; PCA, detailed earlier, is an example of an extraction technique. The effectiveness of each technique is dependent on the nature of the concerning research data. Techniques that are best suited to the research data within this project will be focused on. (Reddy et al. 2020)

Even when assuming all available Spotify features are used within the dataset to be created, samples cannot become overly complex relative to scenarios most reduction techniques are oriented for; datasets can easily exceed hundreds of variables whereas this project only works with a maximum of 13. Therefore, there is a low ceiling of potential for dimensionality reduction, and it should not be over-exercised.

Low Variance Filtering (Feature Selection)
Comparing the variance in the distribution of variables within a dataset enables selection of those with higher variances as they will contribute more to the predictability of a model such as KNN. A low range of variances across different samples of differing classes will not aid the model in distinguishing between them thus making the given variable less desirable.

The variables selected within K-means model of the pilot study underwent low variance filtering as the ranges of the averages of each variable per genre were calculated and used to determine which provided more distinguishability. This process can also be easily applied to the dataset created for the KNN model. Total variance of variables between samples of the same genre can be calculated as well as the variance of variables across all genres to enable greater comparison.

A variable may have a high variance within samples of the same genre, but may not have a comparatively high global variance, making it unpredictable when assigning a class to a given sample. Therefore, variables that possess a high variance across all included genres, and a low variance within samples of the same genre are most desirable and reliable in a prediction-making model such as KNN.

Forward Selection (Feature Selection)
In high-dimensional multilinear regression models forward selection can be used to determine the viability of each variable by adding them to the model build one by one. The calculated adjusted-$R^2$ value of each added variable determines whether it benefits implementation within the model. In a regression model, $R^2$ represents the goodness-of-fit; it describes the percentage of the variance in a dependent variable that is expressed collectively by the independent variables. As more variables are added, the $R^2$ value will increase closer to 100%.

The increase of $R^2$ can be misleading as variables that do not have any bearing on the correlation of a model may be responsible for this increase, resulting in the implementation of ineffective variables. This can cause invalid conclusions between variable relationships and unwarranted complexity.
Adjusted-$R^2$ accounts for the worthiness of each additional variable, meaning its value will increase or decrease relative to this. While the adjusted $R^2$ is usually computed automatically by analysis systems, the equation should be understood in order to understand the underlying process:

$$R^2_{adj} = 1 - \left( \frac{1 - R^2 \frac{n-1}{n-k-1}}{n-k-1} \right)$$

*Figure 7: General equation for adjusted $R$-squared value*

The formula in figure 7 denotes that if the pre-determined $R^2$ increases while $K$ (number of independent variables) is increased, $R^2_{adj}$ will increase proportionally. If $R^2$ does not increase with the value of $K$, $R^2_{adj}$ will decrease (Sutter, 1993).

**Feature Extraction – PCA & ICA**

This research project’s pilot study suggests there are fundamental reasons as to why exercising principal component analysis (PCA) may not be beneficial to this study. When modelling K-means clustering analysis in 3-dimensional space using tempo, energy, and valence PCA scores, a lack of linear correlation between the variables was discovered. This suggests condensing variables into their principal component products may not be effective in forming a predictable KNN model as extraction techniques rely on robust correlations between pre-existing variables.

With lack of proven correlation in mind, there are however, shared fundamental components between some of the previously unused Spotify features which express a potential link between variables. This could be investigated with PCA to test if the predictability of the KNN analysis increases. KNN analysis using PCA coefficients between selected variables could be compared against use of the raw variable values. With regards to the definitions of each track feature provided by Spotify, the following links can be inferred.

On Spotify’s track features reference page, energy is defined as ‘a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy’. In comparison, the danceability of a track is described as ‘a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity’. Tempo is a shared component across energy and danceability, therefore their PCA coefficients could be calculated and used within KNN analysis to enable comparison against the use of tempo itself. Additionally, tempo is defined to be a direct influencer of energy indicating a potentially positive correlation between these two variables; the PCA coefficients of these two variables may be a strong influencer on KNN predictability. The same cannot be assumed for danceability, however, as high danceability is not described as necessarily high tempo, but that tempo is just a component within the calculation of danceability. There is less likely to be a linear correlation between these two variables.

The pilot study revealed valence to be an asset in distinguishing between structurally similar, but emotionally differing genres. This reflects its description; the emotional gradient of a track is perceived through valence, whether it be uplifting or dark and negative. Conducting PCA on valence would be difficult and in vain as the variable is intertwined with human emotion. It is a highly qualitative variable.
To paraphrase; with reference to the definitions of these audio features, it can be inferred that the PCA coefficients of energy and danceability, as well as energy and tempo, may express linear correlations and could act as a tool in increasing the predictability of the KNN model. Additionally, calculating the PCA coefficients of tempo and danceability may aid in determining the influence tempo has on calculating danceability, and whether it could also increase KNN predictability (Talebi, 2021).

**Justification for Use of KNN as AI-based Analysis Method**

K-nearest neighbor surpasses some of the limitations that were unforeseen when using K-means clustering within the pilot study. The nature of data we are researching with suggests a combination of raw variable values and principal component coefficients may provide the greatest insight into machine learnings genre-defining capabilities. The multi-dimensional capabilities of KNN, with the assistance of dimensionality reduction, make it ideal for this study.

Furthermore, the variable interchangeability of KNN-based analysis provided by feature selection and extraction techniques are well suited to this study. A greater number of combinations of analyzed KNN variables means a larger yield of potential insightful information which may lead to discovering the most effective combination of KNN variables, that being what produces the highest predictability within the KNN based system. KNN analysis is oriented around the efficiency of interchanging its input dimensions, making it a valuable tool within this research. When working in a development environment well suited for KNN, such as MATlab, efficiency is further increased due to high data manipulation capabilities.

This is a key factor when concerning the predictability of musical electronic genres. This method of analysis will provide clear and tangible quantitative feedback data that can be used in assessing machine learnings genre predicting capabilities. KNN is a large representative of machine learning in general as it is highly applicable to varying research scenarios and is frequently used in data analysis. The fundamental outcomes of KNN analysis within this project will detail the explicit percentage accuracy of each runtime, and specifically what genres and groups of genres it struggles to predict or is effective in predicting.

**4.2 – Suitable Dataset**

There are several major factors to consider when selecting data to be included datasets for this research project; the number of genres, the number of Spotify features as parameters passed into the KNN algorithm, and how greatly the genres differ in these parameter values. Prior to selecting what genres and subgenres of electronic music should be included, the Spotify features themselves should be understood and what influence selected genres may have on these features values. In order to best define the electronic music and DJ community within this study, an all-encompassing spectrum of electronic genres should be formed to fill the dataset. This will inherently provide the greatest variance in Spotify feature values possible. It will help to eliminate bias of genres that may be witnessed from person-to-person and country-to-country etc.

**Justification for Selected Features**

The distance concentration problem described in section 4.1.3 should be considered when selecting variables that concern the KNN algorithm. A higher number of dimensions will be initially used during
Principal component and correlation coefficients calculated prior to the KNN algorithm will indicate if the number of dimensions can be reduced. Additionally, the accuracy of the KNN model while using the original feature set can be compared to a reduced set to see if variables can be eliminated without decrementing accuracy, and potentially increasing it. With reference to the descriptive nature of each feature, and their performance in the pilot study, the following features have been selected.

As explained, danceability and energy are to be held as valuable parameters due to their apparent influence on electronic genres. Values of danceability and energy scored highly and varied healthily between genres, as did tempo. The variation of tempo values expresses that the electronic music spectrum utilizes the spread of tempo values to effectively fill all potential sounds and emotions rendered throughout electronic genres. Valence proved similar as entire sub-genres can be clearly deemed darker or lighter feeling than others. The final two features that have clear influence on electronic music, and in turn, what electronic music genres seem to make use of, are speechiness and instrumentalness. It can be predicted some electronic genres generally make use of vocals more so than others, making them distinct from one another; for example, dark techno is much less vocal than nu-disco. The opposite of this therefore should be concerned within the KNN model as a relation between speechiness and instrumentalness could be discovered through PCA. The hypothesis that instrumentalness and speechiness share a negative correlation could be inferred regarding this concept. Therefore danceability, energy, tempo, valence, speechiness and instrumentalness may be an ideal initial feature set during KNN analysis, while genre is used as the classification feature.

4.2.1 – Forming a Spectrum of Electronic Genres

The increasing popularity in electronic music has prompted many to classify such genres to better organize the growing industry. There are several factors involved when forming a network of electronic genres and how they relate. Over the decades many movements in music have spawned pools of electronic genres and sub-genres, for example the Detroit Techno scene kickstarted the global adoption of Techno and varying countries followed suit in generating many new Techno genres, most notably what Germany has since output concerning Techno.

At a more musically fundamental level, the track features provided by Spotify reflect the variables that structurally define genres of music. This concept combined with time and human emotion attached to genres based on historical movements in electronic music are generally used to form a definitive network of electronic musical genres. Many networks of electronic genres have been visualized, some being more in-depth and all-encompassing than others. Set parameters must be met concerning the spectrum of electronic genres selected to be included within this study’s dataset, the most overbearing being that the entirety of the generally agreed electronic spectrum is reflected by the dataset.

Key Parent Genres of Electronic Music

It is commonly known that many genres of music inherit aspects of previous and better-established parent genres. These are known as sub-genres, and they can be rooted in multiple parent genres, drawing select features from them to form a new definitive sound. Electronic music has advanced so rapidly since its beginnings that virtually all genres that spawn nowadays are sub-genres as they can be derived from at least some pre-existing genre. This has created a great deal of ambiguity and subjectivity within the music
industry meaning many genres overlap and songs may be classed by genre differently from person to person. The importance of the founding parent electronic genres is highlighted here as they can be used to reverse engineer many modern sub-genres to form the best suited spectrum of genres to be used in this study.

The following genres represented in the network diagram shown in figure 8 form the dominant families of sub-genres that construct the spectrum of electronic music. These can be used as a starting point in selecting genres for the dataset used within KNN analysis.
This genre network diagram was published by Nat Simantov on behalf of Design Infographics in the music blog, Village.fm. It depicts the outer 10 parent genres of electronic music that the inner modern-day subgenres stem from, and notably the network of relations between them in the center.

This diagram expresses effectively the spectral fashion of genres of music as each parent genre shares qualities with those adjacent to it, meaning one can navigate it entirely while linking key factors between genres, all the way round back to the starting point. Lines that link specific subgenres across the network express a unique quality shared between the two subgenres that relates them; for example, acid house and acid trance both incorporate the distinct acid synth sound. The quality this network possesses best ensures that the entire spectrum of electronic music is included within it. This can help in selecting the most variable, unbiased and all-inclusive genre data, consequentially forming a more suitable dataset for this study.

Additional Parent Genre Dataset
A separate dataset for explicitly parent electronic genres could be formed to allow for comparison between KNN trained with sub-genre data and parent genre data. Insighted differences between the two will provide potentially unforeseen details on the usability of sub-genres or parent genres within a machine learning-based genre prediction system which could be used in future practical projects such AI-based playlist creation. The use of two datasets of differing nature will provide a strong point of reference for one another during the analysis phase. Adhering to the diagram above, a dataset using all 10 parent genres that reflect the entire electronic spectrum could be formed. This dataset would not have as great a scope as the sub-genre dataset but would enable better encapsulation of entire sub-genre groups within them.

Technical Requirements of Genres Selected for Dataset Inclusion
It is a key requirement that selected genres are included on Spotify’s own list of recognised genres. Necessary track feature data can only be pulled from Spotify tracks of these available genres; however, this is not a major issue as Spotify has a vast number of registered genres which can be viewed on EveryNoise.com.

“Every Noise at Once is an ongoing attempt at an algorithmically generated, readability-adjusted scatterplot of the musical genre-space, based on data tracked and analyzed for 5,767 genre-shaped distinctions by Spotify as of 2022-02-16.” (McDonald, 2022). The ‘Every Noise’ project developed by Spotify genre taxonomist, Glenn McDonald, can be used to compare the listed genres on the above spectrum diagram against Spotify’s database of genres. Furthermore, the ranking of each genre based on a specified parameter can be viewed on EveryNoise.com. To best reflect the demographic of electronic genres, a suitable number of the most popular sub-genres within each of the 10 parent genres could be selected using this method. Popularity being a key control variable here as it will express the most active and saturated of the genres within a pool of parent genres, making them the best representatives for data analysis.
In knowing that the most popular of each electronic genre group have pre-emptively been selected, popularity can be abstracted out of the actual KNN analysis-oriented dataset to be formed by this process. This data selection process means the greatest proportion possible of each genre group is represented, reducing the likelihood of the KNN algorithm being trained with genre data that only expresses a small region within a parent genre. Use of a lesser-known sub-genre to represent a genre group in KNN analysis could cause a negative skew and lower the genre prediction accuracy.

Another benefit of Every Noise is the project automatically creates a Spotify playlist for every genre listed. These playlists can be used to pull the necessary track feature data of each genre using the Generalized Spotify Analyzer, explained in section 4.2.5.

4.2.3 – Dataset Size Based on Collated Genres and Spotify Features
Selecting a suitable number of dataset samples is essential in any study. Too little can decrease statistical power, observations made are less reliable. A small dataset may increase the effect anomaly samples have in skewing analyzed data. This can lead to wrongly confirmed hypothesis’, known as a type 2 error in sampling. As the sample size increases, the chance of a type 2 error decreases (Harmon et al. 2007).

Congruent to the number of samples, the number of input classes should be considered. The selection of a dataset with 10 classes, and another with 40 (4 subgenres per parent genre), will ideally highlight the fact the KNN model will benefit from one of these values more so than the other. It will provide foresight as to what range of input classes is most suitable for a KNN model. The K-value is derived from the root of the total samples often within KNN analysis, the total number of samples should be large enough to equate to a healthy K-value.

To ensure statistical reliability, 50 songs of each genre will be selected. The following table simply depicts the sample sizes of both datasets:

<table>
<thead>
<tr>
<th>Dataset Type</th>
<th>No. of Genre Classes</th>
<th>Total Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Genres</td>
<td>10</td>
<td>500</td>
</tr>
<tr>
<td>Sub-genres</td>
<td>40</td>
<td>2000</td>
</tr>
</tbody>
</table>

Table 1: Dataset genre to sample size comparison

4.2.4 – Genre Selection Process
This section presents the collected necessary genre data for use in forming the Spotify track features dataset. The now defined process for genre selection will go as follows:

1. Select the 4 most popular sub-genres from the 10 parent genres listed in the electronic genre spectrum diagram. It should be noted that the popularity ranking of each genre is based on the popularity of all genres available on Spotify (5782 as of early 2022). This is updated weekly as well as the rankings.
2. Validate that each genre is recognised by Spotify using Every Noise. If the genre is listed on Every Noise, it is a valid genre.
3. List each selected genre with the associated Spotify playlist ID for use in analyzing track feature data of the playlist with GSA. Relating data is displayed in table 2. Popularity ranking is based on the volume of songs that are part of a given genre and how many audiences play this genre receives. Choosing more popular genres for use in the datasets will better reflect the audience demographic. Playlist ID is the unique identifier for the given Spotify playlist. This attribute has been used in sourcing the required data from each playlist.

<table>
<thead>
<tr>
<th>Parent Genre</th>
<th>Sub-genre</th>
<th>Popularity Ranking</th>
<th>Playlist ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>House</td>
<td>Electro House</td>
<td>#36</td>
<td>4luNnGhISzdURbFcCl2dB6</td>
</tr>
<tr>
<td>House</td>
<td>Progressive House</td>
<td>#122</td>
<td>3ApHTzZrOMqtGhnyiBSSci</td>
</tr>
<tr>
<td>House</td>
<td>Deep House</td>
<td>#242</td>
<td>4Zrv1fjwqwPcswytT3bd</td>
</tr>
<tr>
<td>House</td>
<td>Tech House</td>
<td>#520</td>
<td>7HzEKL4NF7ZKkftZFiekTr</td>
</tr>
<tr>
<td>Disco</td>
<td>Synthpop</td>
<td>#586</td>
<td>7Jwpw8KZvx324DcJx2EE</td>
</tr>
<tr>
<td>Disco</td>
<td>Nu Disco</td>
<td>#590</td>
<td>0SSAOQK7bY8ywZngyAYsp6w</td>
</tr>
<tr>
<td>Disco</td>
<td>Hi-nrg</td>
<td>#638</td>
<td>3Fb2Oreyh9819D3Mgf3tnK</td>
</tr>
<tr>
<td>Disco</td>
<td>Italian Disco</td>
<td>#927</td>
<td>3KqrijNkb21Pacyb7ihAQG</td>
</tr>
<tr>
<td>Trance</td>
<td>Uplifting Trance</td>
<td>#598</td>
<td>2bV5j3UH5k99mJywzTYKu</td>
</tr>
<tr>
<td>Trance</td>
<td>Progressive Psytrance</td>
<td>#1613</td>
<td>4rmUS4GxpHjuZPfeiSz</td>
</tr>
<tr>
<td>Trance</td>
<td>Tech Trance</td>
<td>#3495</td>
<td>2MYwwy0KqanzV0b1Lui8O</td>
</tr>
<tr>
<td>Trance</td>
<td>Acid Trance</td>
<td>#3666</td>
<td>0rSDQ760FMUFBOFSAInC8M</td>
</tr>
<tr>
<td>Techno</td>
<td>Minimal Techno</td>
<td>#529</td>
<td>68uxrFAviTCyQs0U8qr6n</td>
</tr>
<tr>
<td>Techno</td>
<td>Dark Techno</td>
<td>#1516</td>
<td>5ba2wVOCMBf5jmMSIRsGY</td>
</tr>
<tr>
<td>Techno</td>
<td>Acid Techno</td>
<td>#2725</td>
<td>10000A71QYrZ3PzCeAy</td>
</tr>
<tr>
<td>Techno</td>
<td>Detroit Techno</td>
<td>#2844</td>
<td>6eKMGlc0PX3mNLAJOJUDUB</td>
</tr>
<tr>
<td>Hardcore</td>
<td>Trancecore</td>
<td>#707</td>
<td>3GL4HNVog9DjxfVpSrP7</td>
</tr>
<tr>
<td>Hardcore</td>
<td>Hardstyle</td>
<td>#1271</td>
<td>49gROGYXsGx03Gxx0Gcmdb</td>
</tr>
<tr>
<td>Hardcore</td>
<td>Gabba</td>
<td>#1540</td>
<td>6X1FPMa9bO7P7VjluVvb1v</td>
</tr>
<tr>
<td>Hardcore</td>
<td>Frenchcore</td>
<td>#1603</td>
<td>3b10nVg1sEU0XrWvQ8Bv</td>
</tr>
<tr>
<td>Industrial</td>
<td>Electro-industrial</td>
<td>#1559</td>
<td>0ZVlhUadGf0KCEOCGlgtf</td>
</tr>
<tr>
<td>Industrial</td>
<td>Aggrotech</td>
<td>#2060</td>
<td>71010Y6I0Va3h64COeV0Inwb</td>
</tr>
<tr>
<td>Industrial</td>
<td>Power Noise</td>
<td>#4484</td>
<td>23j3NLoqFN0273rsCZH7Q</td>
</tr>
<tr>
<td>Industrial</td>
<td>Power Electronics</td>
<td>#4701</td>
<td>5kAulE1JyusA4SfPoojYG</td>
</tr>
<tr>
<td>Downtempo</td>
<td>Chillwave</td>
<td>#324</td>
<td>5pDDStz9aQULzowpukMSep</td>
</tr>
<tr>
<td>Downtempo</td>
<td>Nu Jazz</td>
<td>#414</td>
<td>26PD3jcsFPUKUDV1jgFX8</td>
</tr>
<tr>
<td>Downtempo</td>
<td>Ambient</td>
<td>#727</td>
<td>6CIyPj34GTPAoWToLNT</td>
</tr>
<tr>
<td>Downtempo</td>
<td>Chill Out</td>
<td>#1357</td>
<td>5PI1SPDdvY2lH4Qlty1</td>
</tr>
<tr>
<td>Hip-hop</td>
<td>East Coast Hip Hop</td>
<td>#117</td>
<td>61KJ6kPue41yTp9DrHwBZ</td>
</tr>
<tr>
<td>Hip-hop</td>
<td>Trip Hop</td>
<td>#404</td>
<td>2wrc23J7dQVcpPcDAged</td>
</tr>
<tr>
<td>Hip-hop</td>
<td>Turntablism</td>
<td>#784</td>
<td>5ZkLWGmHP329LayPnvpydQ</td>
</tr>
<tr>
<td>Hip-hop</td>
<td>Wonky</td>
<td>#1036</td>
<td>6GZZXXWc7QfexcM85804B7</td>
</tr>
<tr>
<td>UK Garage</td>
<td>Grime</td>
<td>#392</td>
<td>6tpwcBh10DSUksx3mRB5</td>
</tr>
</tbody>
</table>
Table 2: Table of parent & sub-genres, popularity ranking & playlist ID

<table>
<thead>
<tr>
<th>UK Garage</th>
<th>Dubstep</th>
<th>#617</th>
<th>0ZYFF9tPuPq2YNMRNhQZjt</th>
</tr>
</thead>
<tbody>
<tr>
<td>UK Garage</td>
<td>Future Garage</td>
<td>#972</td>
<td>62T9WUAB2E7Xw6PmHRSlaQ</td>
</tr>
<tr>
<td>UK Garage</td>
<td>Bassline</td>
<td>#1549</td>
<td>3Lp89wmmn5SSt3U8wWcK3</td>
</tr>
<tr>
<td>Breaks</td>
<td>Drum and Bass</td>
<td>#541</td>
<td>18b0DBiqVtRuj34OKPCgi6</td>
</tr>
<tr>
<td>Breaks</td>
<td>Breakbeat</td>
<td>#1129</td>
<td>1yckyfEZtFkkq7UPXHbLwi</td>
</tr>
<tr>
<td>Breaks</td>
<td>Liquid Funk</td>
<td>#1136</td>
<td>5mu96o6F1pWz5C98LAOnm</td>
</tr>
<tr>
<td>Breaks</td>
<td>Neurofunk</td>
<td>#1997</td>
<td>4Pbi9IXKK2ID9VnHO2wGxO</td>
</tr>
</tbody>
</table>

The 40 selected and validated sub-genres in table 2 will form the genre parameter for the sub-genre dataset. Of each Spotify sub-genre playlist, 50 tracks and their feature data will be collected from the converted .csv files and compiled into a larger .xlsx dataset containing all 40 sub-genres, therefore generating a set of 2000 tracks. Details on this process can be found in section 4.2.5. The same process applies to the parent genre dataset, where 50 tracks of the 10 parent genres will be selected to form a dataset of 500 rows of track feature values.

4.2.5 – Use of Generalized Spotify Analyzer (GSA) to Generate Track Feature Datasets

Generalized Spotify Analyzer is an open-source script created by Ole Adrian Heggli, Assistant Professor at the Centre for Music in the Brain, that allows users to pull track feature data from a Spotify playlist. This research project uses the GSA script to generate raw track feature datasets for each genre playlist selected using Every Noise. Selected fields from these datasets will then be compiled into the parent and sub-genre datasets.

**Forming .xlsx Spreadsheets from GSA-generated .pkl Files**

The developed MATLAB program is designed to read .xlsx filetypes, Excel spreadsheets. The Excel research datasheets have been formed by the conversion of the GSA script-generated pickle (.pkl) files to comma separated value (.csv) files. Both .pkl and .csv are raw data format files. The generated .csv files contain the track feature data of every song within each selected genre-focused playlist. A combined total of 50 genres, each with their own associated playlist, have been selected to form the parent genre .xlsx dataset of 10 genres and the sub-genre dataset of 40 genres. The number of tracks within each used Spotify genre-focused playlist varies, however the track feature data of only 50 tracks per playlist is required.

GSA requires access to the Spotify API to analyze tracks. An app has been set up in the Spotify developers’ dashboard where the client ID, client secret, redirect URL and username have been sourced and pasted into “spotifyConstants.py” within the GSA folder. Upon running GSA through python shell for the first time, a webpage opens prompting for authentication of the Spotify API. After authentication the process is simply taking each given playlist ID from the playlist URI and using `playlist = GSA.getInformation()` to analyze the playlist, followed by `playlistInformation = pd.read_pickle(playlist)` to store the information as a .pkl. Finally, the code within figure 9 is used to convert each .pkl to .csv.
Along with all other available Spotify track features, genre is a variable GSA creates when generating a pickle file for each playlist. Each raw generated .csv playlist datasheet is sorted to display tracks of exclusively the concerning genre at the top of the spreadsheet as Spotify tracks can be labelled with multiple genres. For example, within the Electro House playlist .csv datasheet, purely Electro House tracks will be listed above tracks labelled as Electro House and Tech House; and tracks within the Electro House playlist that are not labelled as Electro House, but a similar genre, will be listed at the bottom of the spreadsheet. By sorting each .csv in this manner, the top 50 purest tracks of each concerning genre will be selected, therefore more accurately representing the target genre within the sub-genre and parent genre datasets used in KNN analysis. This will hopefully allow for better distinction between genres, especially sub-genres, when training and testing the KNN-based genre prediction algorithm.

The screenshot of the sub-genre .xlsx dataset in figure 10 displays all variables pulled from the Spotify playlists using GSA. All variables of interest described in section 4.2.1 are included within GSA script analysis. This allows for easy analysis of differing combinations of input parameters within the KNN algorithm.

4.3 – Spotify Feature Analysis Using KNN
This section details the development of the KNN-based MATLAB program that drives most of the research data to be inferred post-development. An overview of the KNN process from a technical perspective, development components and implemented analysis tools will be given to provide context prior to in-depth explanations of each section.
**KNN Process**

In the context of this study, K-nearest neighbors is the focal point in assessing a machine learning algorithm’s capability of distinguishing between and predicting genres of electronic music. With the two now compiled datasets, the KNN algorithm can be trained and tested to determine this research question.

To begin, the developed MATlab Livescript will prompt the user to select an Excel spreadsheet, for this study either the sub-genres or parent genres dataset. Any necessary data normalization and datatype conversion is then executed once the dataset has been read into a MATlab table (detailed in section 4.3.1). Fields in the table are then randomized so each runtime generates a unique comparable analysis, and so each training and testing dataset per instance is also unique. A percentage of the dataset will be selected for use in the training dataset, and the remainder in the testing dataset. This ratio can be used as an independent variable to assess its influence on the accuracy of the KNN algorithm. A copy of the testing dataset is then created without the genre column. This dataset is used in the prediction phase of the KNN algorithm where the predictions are compared to the genre column in the original testing dataset. Proceeding this setup, the KNN model (mdl) is then trained and tested using `predict()`. Confusion matrices and percentage accuracies are then calculated post-KNN analysis. This process is carried out three times, each iteration using a different K value, meaning the number of neighboring points used in the KNN function. K acts as another independent variable in this research model, where its value changes from 10 to 20 to 30 for each iteration.

**Development Components**

To provide a summary of the functional components within the MATlab program, the Livescript can be deconstructed into 5 main groups of functions:

1. **Raw Data Formatting** – Functions responsible for ensuring data passed into the program can be used correctly by MATlab. This involves normalizing values, generating PCA coefficients and converting datatypes of columns as well as entire tables.
2. **Preliminary Data Analysis** – Use of analytical functions to generate information on the input data independent of the KNN algorithms outcome. This includes generating mean and standard deviation arrays as well as covariance and correlation coefficient matrices.
3. **KNN Data Preprocessing** – This involves randomizing the initial dataset and generating the training and testing datasets from the random permutation for use in the KNN model.
4. **KNN Classifier** – The combination of functions responsible for executing the KNN analysis. This involves creating the KNN model which stores the variables required by the MATlab `ClassificationKNN` object, where `fitcknn` is the function responsible for generating and training the model. The model is then used to predict classifications for the testing dataset.
5. **KNN-based Analysis** – The final component of the program is responsible for generating analytical tools that are dependent on the outcome of the KNN model. This includes calculating the accuracy of the three iterations of the KNN analysis (for each K value) as well as confusion matrices and Ko.

**Analytical Methods**

Different stages of the program’s execution generate informative data based on the input data. MATlab has various pre-existing objects and functions to generate these analytical methods. Analysis performed prior to the KNN function will provide insight into the nature of the input track data and corresponding
genres. This can be used to infer research points independent of KNN analysis as well as provide context and support to key points made while evaluating the outcome of KNN analysis. This includes methods stated in preliminary data analysis. Methods that are dependent on the KNN model and its variables will also be used to assess and validate the machine learning algorithm’s predictive capabilities. This includes methods stated in KNN-based analysis.

4.3.1 – Developing KNN-based Algorithm

This section details the main development stages of the KNN-based MATlab Livescript that is responsible for the analysis of the two collected datasets. The developed program functions around MATlabs FitKNN function which is responsible for the actual KNN analysis of the Spotify track variables passed through it. All the necessary data is read into the program from the collected Excel datasets. It is then formatted for use by the FitKNN model and other implemented analysis functions. Data structures and functions that provide insights into the interpreted track feature data are listed in this section and detailed from a software and functionality perspective. By understanding the process of each analytical function, greater interpretation of concerning data can be deduced. Within each function, key variables and their purpose will be detailed.

Independent & Dependent Variables

Several independent variables have been selected to increase the yield of analysis-based data. The ability to alter these parameters allows for greater flexibility in determining the optimal settings for the KNN algorithm, the predictability outcome being the dependent variable. In the MATlab development environment these independent variables manifest as control parameters instantiated as code objects or variables of varying datatypes. With guidance provided through feature selection, differing combinations of higher potential track features will be passed through the KNN algorithm as independent variables.

Additionally, K can be independently altered to deduce its optimal value, Ko. K represents the number of neighboring data points used in determining a samples classification. The data points chosen to be used in can be classifying a sample can be calculated using varying methods. Euclidean distance is used by the KNN model in this study to determine K neighboring points. It is the most widely used and is typically the default metric within data programming environments such as MATlab. The Euclidean formula calculates the most direct path between two datapoints, therefore the K most linearly close datapoints will be selected. Euclidean distance scales to N dimensions making it highly versatile within KNN where many dimensions are often apparent (Trstenjak et al. 2014). Mathematician Euclid’s equation in figure 11 expresses the Euclidean formula in Nth dimensional space where \( p \) and \( q \) are points in Euclidean N-space and \( q_i \) and \( p_i \) are Euclidean vectors from the point of origin in space.

\[
d(p, q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}
\]

*Figure 11: Euclidean distancing formula (Wikipedia, 2022)*
The optimal K value will result in the greatest predictability outcome within the KNN model. A strong starting point in selecting a K value to use the square root of the total number of input samples (Data_Size) as often v\(\sqrt{\text{Data\_Size}} = K_0\). The developed program runs 3 iterations of the KNN model, each time using a different K value. After running KNN analysis with the square root value of Data_Size, the second and third iteration can use a decreased and increased percentile of the square root of K to indicate any potential predictability increase. The program can be run multiple times with adjusted percentile values to determine what increment best suits the KNN model; this variable will be called \(K_{\text{Pcnt}}\).

The number of controllable input variables available within this study is large due to the nature of KNN and the concerning Spotify data. The focal point of this research is the effect Spotify track features have on the predictability of a KNN model, therefore only the K value and its percentage margins will act as additional independent variables. Other potential control variables such as the KNN model distancing metric may unjustifiably increase the volume of output data and shifts focus away from the actual input data. The use of altering extra independent variables could be considered after main research points have been inferred. The flowchart in figure 12 expresses the use of the concerning variables in relation to the KNN model.

**Datatypes, Normalization & Formatting for Datasets**

All Spotify track features must be passed into the KNN model as numerical values. This is a simple process as all concerning raw values are numerical in nature. The KNN model requires training and testing datasets to be formatted as a matrix, and all values within the matrix to be congruent. The training and testing datasets are matrices of double values, to allow for decimal values. From .xlsx to training and testing datasets, a series of format conversions is required.

---

**Figure 12: Flowchart of independent variables in relation to KNN model**
The genre values column is an essential parameter for the KNN as it is used to compare class predictions to the true class values, genre being the parameter the KNN model intends to classify. Genre values are of string format and so must be mapped and indexed to numerical values so they can be stored in a matrix of doubles. The developed program must be robust enough to deal with differing total class values. This study alone uses a parent genre dataset of 10 total classes compared to the sub-genre dataset of 40 total classes. Figure 13 is an extract of the genre indexing function.

```matlab
UniqueGenres = unique(FeatTbl(:,{'Genres'}));

Data_Size = height(FeatTbl);
GenresRows = height(UniqueGenres);

for i = 1 : Data_Size
    for j = 1 : GenresRows
        if isequal(FeatTbl{i,'Genres'}, UniqueGenres{j,'Genres'})
            FeatTbl{i,'Genres'} = num2cell(j);
            break;
        end
    end
end
```

*Figure 13: Code extract of genre indexing function*

The code in figure 13 is responsible for indexing the genres to their associated numerical values. `UniqueGenres` stores all the unique genre values from the dataset’s genre column in a separate matrix. The total number of data rows and unique genre rows is then determined in the proceeding 2 lines. Finally, in a nested `for` loop, the genre value in every row within the dataset is compared to the unique genre’s matrix. When the genre value is equal to the unique value, this dataset value is replaced with the numerical index value of the unique genre.

Tempo is a parameter deemed highly influential on determining the genre of a track. It is usually measured in beats per minute, electronic music usually ranging between 70-200BPM. The raw tempo values of Spotify tracks must be normalized to scale effectively with other track features that scale from 0-1, a scale factor 100 times less than of BPM. Normalization abstracts away the metric tempo is measured in and reduces the scale factor to match other variables to allow for greater informative precision in dimensional space (Pandey et al. 2017). Figure 14 contains extracted code that normalizes the tempo values.

```matlab
for i = 1 : Data_Size
    FeatMat(i,4) = FeatMat(i,4) / 100;
end
```

*Figure 14: Code extract of tempo normalization*

The now correctly formatted and normalized dataset must be split into training and testing datasets. The ratio of training samples to testing samples should be considered as it can affect the KNN model’s outcome. Common ratios used are anything between 80:20 – 60:40, however this is no single advised ratio. Higher dimensional large datasets may benefit from using higher ratios of 80:20 – 90:10. It is best
practice to use a training dataset larger than the test data, and for initial analysis within this study, a ratio of 70:30 will be used.

The code extract in figure 15 details the process of randomizing the initial samples and then assigning them to either training or testing datasets. Genre columns are then copied from these datasets into their own matrices for use in comparing predictions to true class values within the KNN model, Class_Lbl_Train/Test. This is followed by a flowchart expressing the general process of converting raw data to training and testing data as shown in figure 16.

```matlab
Data_Selected = Data_Size*0.7;
Randomized = randperm(Data_Size);

m = 1;
j = 1;
for i = 1:size(FeatMat,1)
    if m <= Data_Selected
        Training_Temp{i} = FeatMat(Randomized(i),:);
        m=m+1;
    else
        Testing_Temp{m} = FeatMat(Randomized(i),:);
        m=m+1;
        j=j+1;
    end
end
TrainingSet = cell2mat(Training_Temp');
TestingSet = cell2mat(Testing_Temp');
TestingSet2 = TestingSet(:,1:6);
Class_Lbl_Train = TrainingSet(:,7);
Class_Lbl_Test = TestingSet(:,7);
```

Figure 15: Code extract of training & testing dataset preparation
Principal Component Analysis (PCA) Functions

The MATLab PCA function can be used to identify principal component coefficients between Spotify feature variables. PCA coefficients that score a strong positive or negative value (0.5 < X < -0.5) for multiple variables may indicate a correlation between them. The implementation of these variables into the KNN model can be abstracted into the PCA coefficient ultimately reducing the dimensions of the model’s subject datasets (Shlens, 2014).

The first line of code extracted in figure 17 calls the PCA function and passes the variables of the feature matrix values into it. `PCA_Coeff_Mat` then stores the coefficient values calculated. In a MATlab coefficient matrix the columns represent the PCA function, the value of which equates the number of input variables. The rows represent the coefficient score within the given PCA function for each variable.

```matlab
[coeff, score,~,~,explained,~] = pca(FeatMat);
PCA_Coeff_Mat = coeff(:,:);
```

Figure 16: Flowchart of raw data to training & testing dataset process
Mean & Standard Deviation Arrays

These values will provide some general insight into the parent and sub-genre datasets which can be used to better describe the nature of each track feature. Mean values of each variable can be used to define the center point and these values will help reflect the nature of electronic music. Use of the mean function within MATLAB is displayed in figure 18.

\[
\text{MeanArray} = \text{mean(FeatMat(:,1:6))};
\]

\textit{Figure 18: Code extract of mean array function}

Standard deviation depicts the spread of data in relation to the mean value. Low deviation expresses greater clustering of data around the mean, or a tighter fit of datapoints, where as high deviation expresses a wider spread of data from the mean. The MATLAB function for standard deviation is \textit{std}, shown in figure 19.

\[
\text{StdDevArray} = \text{std(FeatMat(:,1:6))};
\]

\textit{Figure 19: Code extract of standard deviation array function}

Covariance & Correlation Coefficient Matrices

The data covariance and correlation coefficients represent are similar as they both describe the relationship between the variance of two variables. The difference between them manifests through their mathematical formulae. Covariance value expresses the expectation of a datapoint’s positioning dimensional space in relation to the mean value of X and Y. Shown below is the covariance formula, where \( E = \text{Expectation value and } \mu = \text{the mean value.} \)

\[
\text{Covariance (X,Y) = E((X- \mu X)(Y- \mu Y))}
\]

Assume the X value of a sample is above its mean in a scenario where there is a positive correlation between X and Y. The value of Y is likely to also be above its own mean resulting in higher value in both dimensions. They are proportional to each other; Y will also decrease as X decreases. If a negative correlation is apparent, Y becomes inversely proportional to X therefore if one is below its mean the other will likely be above. Applied to the covariance formula, this will generate a negative value as a positive is always being multiplied by a negative.

Covariance inherits the product of the units of X and Y as its own which could cause scaling issues depending on the input variables. The correlation coefficient normalizes covariance values to a scale from \(-1 \leq X \leq 1\).

\[
\text{Correlation (X,Y) = Cov(X,Y)/\sqrt{Var(X)*Var(Y)}}
\]
The MATLAB functions expressed in figure 20 make use of these formulae to calculate covariances and correlation coefficients between all input variables to form matrices for each. Interpretations of relationships between input variables can be made here during analysis.

\[
\begin{align*}
\text{Covariance Matrix} &= \text{cov}(\text{FeatMat}(:,1:6)); \\
\text{Correlation Coef Matrix} &= \text{corrcoef}(\text{FeatMat}(:,1:6));
\end{align*}
\]

Figure 20: Code extract of covariance and correlation coefficient matrices functions

**KNN Classifier Function**

The `fitcknn` function is responsible for the KNN analysis process preparation. An instance of this function is created as `Mdl` and the necessary training data is passed into it. The `predict` function is then used within a `for` loop that iterates through all test samples. This entire function is nested in a parent `for` loop that iterates 3 times where the K value is different for each \((K, K\pm10\%\)) as described previously. Predicted class values are assigned to the `KNN_Prediction` matrix as shown in figure 21.

```matlab
if Iterate == 1
    K = sqrt(Data_Size);
    K_int = int8(K);
elseif Iterate == 2
    K = sqrt(Data_Size);
    K_int = int8((K./100)*110);
else Iterate == 3
    K = sqrt(Data_Size);
    K_int = int8((K./100)*90);
end

Mdl = fitcknn(TrainingSet(:,1:6), Class_Lbl_Train(:,:), 'NumNeighbors', K_int, 'Distance', 'euclidean');

for i = 1:size(TestingSet2,1)
    Test_Exmp = TestingSet2(i,:);
    KNN_Prediction(i) = predict(Mdl, Test_Exmp);
end
```

Figure 21: Code extract of KNN classifier function
**Confusion Matrix**

Confusion matrices are highly useful in determining the distribution of class predictions and are often used in summarizing the accuracy of supervised learning algorithms. They express exactly the distribution of classifications on a class value by predicted class value matrix. Correct classifications fall on the diagonal formed by same class values through the matrix (Visa et al. 2011). When working with larger datasets, percentage values of distributions can be used for easier interpretation. The developed program uses percentage values in generated confusion matrices for each KNN iteration, shown in figure 22.

```matlab
for i = 1:40
    Index = find(Class_Lbl_Test == i);
    Total_Classed = length(Index);
    for m = 1:40
        Classification = length(find(KNN_Prediction(Index) == m));
        Confusion_Matrix(m,i) = Classification / Total_Classed * 100;
    end
end
```

*Figure 22: Code extract of confusion matrix function*

The function above loops through each genre value indexing every test sample classified as that genre and storing the total number of classed values as Total_Classed. For every genre value (i), all genres are looped through (m) and the length of indexed predictions equal to m are assigned to Classification. These values are then calculated as a percentile and stored in position (m,i) of the confusion matrix.

**Calculating Optimal K value (Ko) & Prediction Accuracy**

The optimal K-value determined for each runtime instance of the program is that which generates the highest predictability. To identify this value, the total percentage accuracy is calculated for each iteration of the KNN function. The length of total found values where KNN_Prediction equals the true class value, Class_Lbl_Test, is divided by the total class data length multiplied by 100 to determine the percentage classification accuracy as shown in figure 23.

```matlab
Avg_Crt_Class = length(find((KNN_Prediction - Class_Lbl_Test') == 0)) / length(Class_Lbl_Test) * 100;
```

*Figure 23: Code extract of average classification accuracy function*
Each iteration of the KNN function then assigns the Avg_Crt_Class value to a matrix of accuracy values with corresponding K-values. The maximum accuracy value is then selected and the matching K-value in the matrix is displayed as Ko as shown in figure 24.

\[
[m,i] = \max(Pcnt_acc_mat(:,:));
Ko = Pcnt_acc_mat(i(2));
disp ('Optimal K value:');
disp (num2str(Ko));
\]

**Figure 24: Code extract of Ko function**

4.4 – Initial Data Analysis
This section will present data generated by the analysis program and will interpret it through quantitative evaluation. Comparison of data generated using parent and sub-genre datasets will be inferred as well as relations between variables. Deduced key research points will be justified using the generated data as well as pre-existing research.

4.4.1 – Generating & Collecting Data
The use of averaging research data ensures it is more reliable, consistent patterns in data become more resolute and inferred research points are better supported. Data from multiple runtime instances can be averaged to achieve this. Besides examples provided within a specific instance of the program, or where values remain constant such as mean and standard deviation, generated data is deduced from the average of 10 runtime instances of the developed MATLAB code. This most notably affects the percentage accuracy and Ko values. All instances of matrices and tables used during analysis are either displayed by MATLAB using the disp function, or are accessed via the workspace tab.

4.4.2 – Interpreting Mean & Standard Deviation Values
Tables 3 and 4 show the mean and standard deviation values generated for the parent and sub-genre datasets.

### Mean Values

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Danceability</th>
<th>Energy</th>
<th>Valence</th>
<th>Tempo</th>
<th>Speechiness</th>
<th>Instrumentalness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent</td>
<td>0.6359</td>
<td>0.7504</td>
<td>0.4816</td>
<td>1.2591</td>
<td>0.0850</td>
<td>0.3602</td>
</tr>
<tr>
<td>Sub-genres</td>
<td>0.6169</td>
<td>0.7684</td>
<td>0.4108</td>
<td>1.2885</td>
<td>0.0927</td>
<td>0.4695</td>
</tr>
</tbody>
</table>

*Table 3: Table of dataset mean values*

### Standard Deviation Values

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Danceability</th>
<th>Energy</th>
<th>Valence</th>
<th>Tempo</th>
<th>Speechiness</th>
<th>Instrumentalness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent</td>
<td>0.1716</td>
<td>0.1863</td>
<td>0.2765</td>
<td>0.2282</td>
<td>0.0692</td>
<td>0.3924</td>
</tr>
<tr>
<td>Sub-genres</td>
<td>0.1645</td>
<td>0.2063</td>
<td>0.2730</td>
<td>0.2572</td>
<td>0.0827</td>
<td>0.3881</td>
</tr>
</tbody>
</table>

*Table 4: Table of dataset standard deviation values*

Both datasets express similar patterns within each feature. This suggests that sub-genres of electronic do not deviate significantly from their parental counterparts in their fundamental structure. Mean danceability and energy values affirm that electronic music sways towards high energy and danceability, potentially justifying the interchangeability of the terms ‘dance’ and ‘electronic’ music. The deviation of danceability and energy are highly similar at 0.17 and 0.16. These low values suggest factors that determine the concerning features vary less than others, for example valence. Concerning to the compositional nature of danceability and energy, beat strength and regularity could be responsible for the tight spread of values. This highlights the fact that genres of electronic music are comprised of set beat patterns, and within each song, they rarely switch between them and remain as one consistent beat throughout the track.

Furthermore, a highly drum-driven and consistent set of parent genres are house and disco. The drums are audibly prominent in such genres that use a four-to-the-floor beat pattern. These genres contribute to a large and arguably more popular segment of the electronic genre spectrum, which is justified by the displayed popularity rankings displayed within the table of sub-genres. The tempo of these genres, especially house, proves dominant as the mean tempo values for the parent and sub-genre datasets are 1.26 and 1.29. House music almost always is found to have a tempo of 120BPM-130BPM. The high popularity of house and disco combined with the mean tempo values suggests these are highly influential on the dataset and that a larger portion of sub-genres share similar feature values.

History of House Linked to the Fundamental Sound

This research point fits with the history of house and disco also. These are perceived as some of the founding genres of electronic music. Their deep roots favor popularity, and quantity as more is produced over a longer time. This also gives more time for the fundamentals of these genres to become better established, ultimately making tracks that fall into these groups more distinct sounding and recognizable. “Musically house music is the four on the floor, a relentless beat that keeps the mood flowing, whether fast paced, vocal, grooving or synthetic, it provides a unique feeling of repetition that gets into your mind, body and soul.” (Saunders, 2007) quoted from ‘House Music the Real Story’, reflects the concept that beat pattern and regularity are the key influencers of house and similar genres; these variables of which are described by Spotify to dictate the value of danceability and energy.

How Could This Affect Predictability?

The similarity in the beat structure of a significant portion of parent genres could make distinguishing between them a difficult task for the KNN algorithm. The KNN model can only work with danceability and energy values that abstract away the beat structure of tracks and has resulted in a small deviation of mean these values. The relation between danceability, energy and tempo may aid the KNN algorithm in classifying genres that vary in tempo, however this may be the only extent to which these features contribute. Other features may provide more distinguishability from a less structural-focused perspective, especially valence.

Valence has a significant standard deviation while mean values for both datasets are relatively centralized at 0.48 for parent genres and 0.41 for sub-genres. This deems it a relatively un-skewed and better spread
set of values that could aid predictability effectively. The opposite is apparent for speechiness which suggests some descendant vocal genres of disco may be heavily outweighed by the larger quantity of completely instrumental-focused genres.

4.4.3 – Interpreting Covariance, Correlation Coefficients & PCA Coefficients

Tables 5-8 show the covariance and correlation coefficients generated for the parent and sub-genre dataset variables.

### Parent Genres Covariance Values

<table>
<thead>
<tr>
<th>Data Feature</th>
<th>Danceability</th>
<th>Energy</th>
<th>Valence</th>
<th>Tempo</th>
<th>Speechiness</th>
<th>Instrumentalness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Danceability</td>
<td>0.0294</td>
<td>-0.0051</td>
<td>0.0217</td>
<td>-0.0012</td>
<td>0.0003</td>
<td>-0.0011</td>
</tr>
<tr>
<td>Energy</td>
<td>-0.0051</td>
<td>0.0347</td>
<td>0.0022</td>
<td>0.0105</td>
<td>0.0007</td>
<td>0.0095</td>
</tr>
<tr>
<td>Valence</td>
<td>0.0217</td>
<td>0.0022</td>
<td>0.0765</td>
<td>-0.0019</td>
<td>-0.0013</td>
<td>-0.0232</td>
</tr>
<tr>
<td>Tempo</td>
<td>-0.0012</td>
<td>0.0105</td>
<td>-0.0019</td>
<td>0.0521</td>
<td>0.0001</td>
<td>0.0191</td>
</tr>
<tr>
<td>Speechiness</td>
<td>0.0003</td>
<td>0.0007</td>
<td>-0.0013</td>
<td>0.0001</td>
<td>0.0048</td>
<td>-0.0048</td>
</tr>
<tr>
<td>Instrumentalness</td>
<td>-0.0011</td>
<td>0.0095</td>
<td>-0.0232</td>
<td>0.0191</td>
<td>-0.0048</td>
<td>0.1540</td>
</tr>
</tbody>
</table>

*Table 5: Table of parent genres covariances*

### Sub-genres Covariance Values

<table>
<thead>
<tr>
<th>Data Feature</th>
<th>Danceability</th>
<th>Energy</th>
<th>Valence</th>
<th>Tempo</th>
<th>Speechiness</th>
<th>Instrumentalness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Danceability</td>
<td>0.0271</td>
<td>0.0001</td>
<td>0.0200</td>
<td>-0.0048</td>
<td>0.0006</td>
<td>-0.0027</td>
</tr>
<tr>
<td>Energy</td>
<td>0.0001</td>
<td>0.0425</td>
<td>0.0026</td>
<td>0.0156</td>
<td>0.0023</td>
<td>-0.0112</td>
</tr>
<tr>
<td>Valence</td>
<td>0.0200</td>
<td>0.0026</td>
<td>0.0745</td>
<td>-0.0058</td>
<td>0.0008</td>
<td>-0.0315</td>
</tr>
<tr>
<td>Tempo</td>
<td>-0.0048</td>
<td>0.0156</td>
<td>-0.0058</td>
<td>0.0662</td>
<td>0.0008</td>
<td>0.0045</td>
</tr>
<tr>
<td>Speechiness</td>
<td>0.0006</td>
<td>0.0023</td>
<td>0.0008</td>
<td>0.0008</td>
<td>0.0068</td>
<td>-0.0104</td>
</tr>
<tr>
<td>Instrumentalness</td>
<td>-0.0027</td>
<td>-0.0112</td>
<td>-0.0315</td>
<td>0.0045</td>
<td>-0.0104</td>
<td>0.1506</td>
</tr>
</tbody>
</table>

*Table 6: Table of sub-genres covariances*

### Parent Genres Correlation Coefficient Values

<table>
<thead>
<tr>
<th>Data Feature</th>
<th>Danceability</th>
<th>Energy</th>
<th>Valence</th>
<th>Tempo</th>
<th>Speechiness</th>
<th>Instrumentalness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Danceability</td>
<td>1</td>
<td>-0.1593</td>
<td>0.4569</td>
<td>-0.0312</td>
<td>0.0255</td>
<td>-0.0170</td>
</tr>
<tr>
<td>Energy</td>
<td>-0.1593</td>
<td>1</td>
<td>0.0425</td>
<td>0.2459</td>
<td>0.0581</td>
<td>0.1296</td>
</tr>
<tr>
<td>Valence</td>
<td>0.4569</td>
<td>0.0425</td>
<td>1</td>
<td>-0.0303</td>
<td>-0.0661</td>
<td>-0.2137</td>
</tr>
<tr>
<td>Tempo</td>
<td>-0.0312</td>
<td>0.2459</td>
<td>-0.0303</td>
<td>1</td>
<td>0.0099</td>
<td>0.2134</td>
</tr>
<tr>
<td>Speechiness</td>
<td>0.0255</td>
<td>0.0581</td>
<td>-0.0661</td>
<td>0.0099</td>
<td>1</td>
<td>-0.1768</td>
</tr>
<tr>
<td>Instrumentalness</td>
<td>-0.0170</td>
<td>0.1296</td>
<td>-0.2137</td>
<td>0.2134</td>
<td>-0.1768</td>
<td>1</td>
</tr>
</tbody>
</table>

*Table 7: Table of parent genres correlation coefficients*
Sub-genres Correlation Coefficient Values

<table>
<thead>
<tr>
<th>Data Feature</th>
<th>Danceability</th>
<th>Energy</th>
<th>Valence</th>
<th>Tempo</th>
<th>Speechiness</th>
<th>Instrumentalness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Danceability</td>
<td>1</td>
<td>0.0029</td>
<td>0.4454</td>
<td>-0.1138</td>
<td>0.0455</td>
<td>-0.0425</td>
</tr>
<tr>
<td>Energy</td>
<td>0.0029</td>
<td>1</td>
<td>0.0467</td>
<td>0.2939</td>
<td>0.1354</td>
<td>-0.1402</td>
</tr>
<tr>
<td>Valence</td>
<td>0.4454</td>
<td>0.0467</td>
<td>1</td>
<td>-0.0822</td>
<td>0.0377</td>
<td>-0.2976</td>
</tr>
<tr>
<td>Tempo</td>
<td>-0.1138</td>
<td>0.2939</td>
<td>-0.0822</td>
<td>1</td>
<td>0.0400</td>
<td>0.0448</td>
</tr>
<tr>
<td>Speechiness</td>
<td>0.0455</td>
<td>0.1354</td>
<td>0.0377</td>
<td>0.0400</td>
<td>1</td>
<td>-0.3237</td>
</tr>
<tr>
<td>Instrumentalness</td>
<td>-0.0425</td>
<td>-0.1402</td>
<td>-0.2976</td>
<td>0.0448</td>
<td>-0.3237</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 8: Table of sub-genres correlation coefficients

At first inspection, generated covariance and correlation coefficients score relatively low in all dimensions for both datasets. This suggests a lack of correlation between any of the variables affirming they are independent of one another. Consequentially, this presents little aid as to discovering which features have greater influence over determining genre and which can be abstracted into principal components to reduce dimensionality.

**Positivity Manifested Through Electronic Dance Music**

One relationship to note is the correlation coefficient between valence and danceability. The score of 0.45 suggests a slightly positive correlation between the two. From an emotional perspective, this relationship does make sense as a highly uplifting positive sounding song may cause one to feel more inclined to dance to it. Therefore, one could expect the danceability of the track to score high if it has a high valence, as this correlation confirms. This hypothesis is routed in the psychology of humans as happiness and positivity often manifests itself through dance. Within the electronic dance music scene masses of people describe themselves experiencing a state of ‘ecstasy’ and ‘trance’. An entire group of electronic dance genres being labelled as trance; an uplifting, faster paced form of electronic music (St John, 2017).

Despite the strong human connection between positivity and dance which has been affirmed through valence and danceability feature values, the coefficient may not be strong enough to warrant the introduction of a principal component representing their relationship. Assessment of generated PCA coefficients shown in tables 9 and 10 will answer this.

Parent Genres PCA Coefficient Values

<table>
<thead>
<tr>
<th>Data Feature</th>
<th>PC-1</th>
<th>PC-2</th>
<th>PC-3</th>
<th>PC-4</th>
<th>PC-5</th>
<th>PC-6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Danceability</td>
<td>-0.0546</td>
<td>0.3691</td>
<td>-0.1565</td>
<td>-0.3535</td>
<td>0.8408</td>
<td>-0.0666</td>
</tr>
<tr>
<td>Energy</td>
<td>0.0801</td>
<td>0.0795</td>
<td>0.4280</td>
<td>0.8078</td>
<td>0.3857</td>
<td>-0.0501</td>
</tr>
<tr>
<td>Valence</td>
<td>-0.2622</td>
<td>0.8864</td>
<td>-0.0315</td>
<td>0.1278</td>
<td>-0.3541</td>
<td>0.0529</td>
</tr>
<tr>
<td>Tempo</td>
<td>0.1723</td>
<td>0.1190</td>
<td>0.86366</td>
<td>-0.4528</td>
<td>-0.0715</td>
<td>-0.0095</td>
</tr>
<tr>
<td>Speechiness</td>
<td>-0.0258</td>
<td>-0.0279</td>
<td>0.0304</td>
<td>0.0046</td>
<td>0.0969</td>
<td>0.9941</td>
</tr>
<tr>
<td>Instrumentalness</td>
<td>0.9442</td>
<td>0.2383</td>
<td>-0.2109</td>
<td>0.0293</td>
<td>-0.0668</td>
<td>0.0440</td>
</tr>
</tbody>
</table>

Table 9: Table of parent genres PCA coefficients
### Sub-genres PCA Coefficient Values

<table>
<thead>
<tr>
<th>Data Feature</th>
<th>PC-1</th>
<th>PC-2</th>
<th>PC-3</th>
<th>PC-4</th>
<th>PC-5</th>
<th>PC-6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Danceability</td>
<td>-0.07083</td>
<td>-0.24916</td>
<td>0.275724</td>
<td>0.055336</td>
<td>0.922044</td>
<td>-0.06044</td>
</tr>
<tr>
<td>Energy</td>
<td>-0.08738</td>
<td>0.351013</td>
<td>0.280407</td>
<td>0.886845</td>
<td>-0.05138</td>
<td>-0.03732</td>
</tr>
<tr>
<td>Valence</td>
<td>-0.34675</td>
<td>-0.49064</td>
<td>0.706341</td>
<td>-0.0826</td>
<td>-0.36279</td>
<td>0.04101</td>
</tr>
<tr>
<td>Tempo</td>
<td>0.051498</td>
<td>0.727609</td>
<td>0.516534</td>
<td>-0.44253</td>
<td>0.07195</td>
<td>-0.01095</td>
</tr>
<tr>
<td>Speechiness</td>
<td>-0.06418</td>
<td>0.042232</td>
<td>-0.01786</td>
<td>0.028776</td>
<td>0.075224</td>
<td>0.993626</td>
</tr>
<tr>
<td>Instrumentalness</td>
<td>0.927546</td>
<td>-0.20685</td>
<td>0.281611</td>
<td>0.083451</td>
<td>-0.06885</td>
<td>0.07656</td>
</tr>
</tbody>
</table>

*Table 10: Table of sub-genres PCA coefficients*

Each PCA coefficient above in figures 9 and 10 describes different potential relations between variables; coefficients that share large positive or negative values for variables suggest a relationship between them. For both matrices, there is 1 significantly large coefficient value for each principal component. For example, \( PC-1 \) scores very high in instrumentalness, meaning it is the focal variable and it lacks relationship with other variables as none other share a high positive coefficient.

One point of interest concerns \( PC-3 \) within both tables, where there seems to be some potentially weak relation between tempo and energy within the parent dataset, however this is not apparent within the sub-genre dataset where instead tempo and valence seem to be loosely related. These results could support the hypothesis deduced within the pilot study that the energy of a track is somewhat determined by its tempo.

Addressing danceability and valence, there is no justifiably significant relationship depicted by the parent genres dataset. Within \( PC-2 \), valence scores high at 0.88 and danceability also shares a positive value with some magnitude at 0.37, however it would be a stretch to consider using this principal component in favor of the two variables to optimize the KNN model.

#### 4.4.4 – Interpreting KNN Analysis

Analysis performed within this section will attempt to answer the main research question of this study which asks how capable a machine-learning based algorithm is at predicting genres of electronic music. To provide context to confusion matrices, the columns represent the actual genre class, and the rows represent the predicted genre. An index is provided to show which genre each column and corresponding row represent. Due to the size of the sub-genre matrices, only segments that directly relate to the inferred research point will be presented. While table 13 projects the confusion matrix for parent genres, instances of sub-genre confusion matrices in their entirety are available in the [appendix](#).
Evaluating Average KNN Accuracy
Firstly, a generalization of the KNN models performance can be assessed with collected Ko and prediction accuracy values. The two table below detail the performance of the parent and sub-genre datasets with K-values within a range of +/-20% of the root of total samples.

Parent Genres K-Values & Prediction Accuracy

<table>
<thead>
<tr>
<th>K-Value Deviation from √No. Samples</th>
<th>Iteration K-Value</th>
<th>Average Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-20%</td>
<td>18</td>
<td>42.00</td>
</tr>
<tr>
<td>-15%</td>
<td>19</td>
<td>44.16</td>
</tr>
<tr>
<td>-10%</td>
<td>20</td>
<td>42.34</td>
</tr>
<tr>
<td>-5%</td>
<td>21</td>
<td>40.00</td>
</tr>
<tr>
<td>0</td>
<td>22</td>
<td>42.50</td>
</tr>
<tr>
<td>+5%</td>
<td>23</td>
<td>39.50</td>
</tr>
<tr>
<td>+10%</td>
<td>25</td>
<td>40.33</td>
</tr>
<tr>
<td>+15%</td>
<td>26</td>
<td>41.49</td>
</tr>
<tr>
<td>+20%</td>
<td>27</td>
<td>38.33</td>
</tr>
</tbody>
</table>

Table 11: Table of parent genres K-values and prediction accuracy

Sub-genres K-Values & Prediction Accuracy

<table>
<thead>
<tr>
<th>K-Value Deviation from √No. Samples</th>
<th>Iteration K-Value</th>
<th>Average Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-20%</td>
<td>36</td>
<td>25.45</td>
</tr>
<tr>
<td>-15%</td>
<td>38</td>
<td>26.05</td>
</tr>
<tr>
<td>-10%</td>
<td>40</td>
<td>24.78</td>
</tr>
<tr>
<td>-5%</td>
<td>42</td>
<td>25.50</td>
</tr>
<tr>
<td>0</td>
<td>45</td>
<td>23.83</td>
</tr>
<tr>
<td>+5%</td>
<td>47</td>
<td>25.50</td>
</tr>
<tr>
<td>+10%</td>
<td>49</td>
<td>23.83</td>
</tr>
<tr>
<td>+15%</td>
<td>51</td>
<td>24.55</td>
</tr>
<tr>
<td>+20%</td>
<td>54</td>
<td>24.22</td>
</tr>
</tbody>
</table>

Table 12: Table of sub-genres K-values and prediction accuracy

It is highly apparent training and testing with the parent genres dataset has performed better than the sub-genres dataset. The average accuracy of all combined averages for the parent dataset is 41.18%; for the sub-genre dataset it is 24.86%. The optimal K-value for each dataset was found to be K – 15%(K). This indicates a general equation can be formed and applied datasets of any sample size when calculating Ko.

\[ Ko = \frac{(√\text{Total Samples} \times 85)}{100} \]

The parent data contains 500 samples, 50 samples of 10 genre groups therefore Ko was found to be 19, \((√500 \times 85)/100\). The dataset of sub-genres contained 2000 samples; Ko was calculated to be 38. There
does not seem to be any linear correlation between accuracy and the corresponding K-value despite both datasets sharing the same Ko formula.

Several points can be inferred from the differing predictability accuracies. There is potential that the distinct features between parental groups of electronic genres are blurred within sub-genres within differing groups. Sub-genres included in the dataset may share similar components to other sub-genres that descend from a different parental group causing difficulty for the KNN model during classification. This is known as class-overlapping. (Beckmann, 2015)

The number of classes within the sub-genre’s dataset is 40, 4 times greater than that of the parent genres dataset. The relatively large number of classes has evidently played a large role in determining accuracy. The optimal number of classes is certainly below 40 as it is known that as the number of classes within a KNN model increases, the predictability will likely decrease. This rule still applies when the frequency of each classed is balanced as it is in this study; unbalanced classes are known to cause underfitting. This is further justified by the likelihood that the subtle differences between sub-genres of same parent genre groups go unnoticed by the KNN model. The track features provided by Spotify used as variables may be too similar between these sub-genres. This leads to a key point that can be inferred from this study.

**Machine-Learning’s Inability to Identify Intangible Shared Features**

This thesis has covered the subjectivity of music and its genres. Quantifiable parameters used have enabled KNN analysis to be performed for this study, however it does not negate the qualitative attributes of the samples. Some qualities of music simply must be addressed with a humanistic, emotional approach. This relates to a general process of how some genres of music originate, and the KNN algorithm’s inability to identify these qualities further supports the subjectivity to it.

Sub-genres of music often arise through the advent of a specific sound that often skyrockets in popularity and causes an influx of new tracks to be released that share the same sound; a new genre is born. These musical qualities are hard to quantify as they form a general sound within a track. An example relevant to this study is the rise of ‘acid’ genres. Acid techno and acid trance are sub-genres both included in the research data, and both stem from different parent genres, techno and trance. Despite this, their origins and general sound closely relate. The acid sound is highly distinct and recognizable. It was coined in the late 1980’s after the short plucky stabs of acid were created on sound synthesizers at the time such as the TB-303. To this day, genres that derive from the acid sound still thrive. The sound has rooted itself into many parental forms of electronic music such as house, techno and trance (Vitos, 2014).

This nature of this attribute goes undetected throughout KNN analysis. The history behind it and its intangible quality cannot be quantified through track features that have been gathered through Spotify. This premise applies to a significant portion of electronic sub-genres. It may contribute to the significant drop in predictability accuracy when analyzing sub-genres compared to the parent genres. The parent genres more so describe a more distinct group of similarly composed genres that share concerning feature values such as tempo.
Evaluating Key Confusion Matrix Values

Figure 13 shows a confusion matrix of percentage distributions of classifications for the parent genres dataset where K is equal to the calculated optimal value, 19. The classification accuracy for this runtime instance was 43.33%.

Parent Genres Confusion Matrix (K = 19)

Columns: Actual Genre Class, Rows: Predicted Genre Class

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>59.09</td>
<td>6.25</td>
<td>5.88</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16.67</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>62.5</td>
<td>11.77</td>
<td>0</td>
<td>11.11</td>
<td>0</td>
<td>0</td>
<td>8.33</td>
<td>7.69</td>
<td>33.33</td>
</tr>
<tr>
<td>3</td>
<td>4.55</td>
<td>6.25</td>
<td>41.18</td>
<td>0</td>
<td>5.56</td>
<td>0</td>
<td>25</td>
<td>8.33</td>
<td>0</td>
<td>11.11</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>5.88</td>
<td>66.67</td>
<td>22.22</td>
<td>0</td>
<td>8.33</td>
<td>0</td>
<td>0</td>
<td>5.56</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>6.25</td>
<td>11.77</td>
<td>0</td>
<td>50</td>
<td>23.08</td>
<td>8.33</td>
<td>0</td>
<td>0</td>
<td>16.67</td>
</tr>
<tr>
<td>6</td>
<td>4.55</td>
<td>12.5</td>
<td>0</td>
<td>0</td>
<td>5.56</td>
<td>15.39</td>
<td>16.67</td>
<td>0</td>
<td>0</td>
<td>11.11</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>22.22</td>
<td>0</td>
<td>7.69</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>18.18</td>
<td>6.25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>23.08</td>
<td>8.33</td>
<td>33.33</td>
<td>7.69</td>
<td>5.56</td>
</tr>
<tr>
<td>9</td>
<td>13.64</td>
<td>0</td>
<td>17.65</td>
<td>11.11</td>
<td>5.56</td>
<td>15.39</td>
<td>25</td>
<td>16.67</td>
<td>84.62</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>5.88</td>
<td>0</td>
<td>0</td>
<td>15.39</td>
<td>8.33</td>
<td>16.67</td>
<td>0</td>
<td>16.67</td>
</tr>
</tbody>
</table>

Table 13: Confusion matrix of parent genre predictions

It is apparent all classes experience at least some distribution of incorrect classifications. Breaks, disco, hardcore, hip-hop and trance express classifications that exceed the average, ranging from 59.09% to 84.62%. The remaining genres suffer lower accuracies that range from 0% to 50%. Notably, house, industrial and UK-garage contain a higher conglomerate of classifications within the incorrect genre. For example, house contains 23.08% of its classifications in hip-hop and trance.

To assess whether the randomization of samples has caused, or whether these genres are fundamentally detrimental within the KNN model, average accuracies of 10 instances of the KNN model were collected using two altered datasets: one that does not include the 3 lowest scoring genres in the confusion matrix.
above, and one that does not include the 3 highest scoring. By running the program with both altered version, potential that simply a reduced number of classes benefitting the KNN model will be addressed. If only the dataset that has removed the 3 worst genres from above presents an increase in accuracy, it may suggest the feature-based composition of these genres make them difficult to correctly classify. Table 13 presents these calculated values in relation to the original parent datasets accuracy.

<table>
<thead>
<tr>
<th>Original Dataset</th>
<th>Low-scoring Genres Removed (A1)</th>
<th>High-scoring Genres Removed (A2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>44.16%</td>
<td>54.80%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>40.13%</td>
</tr>
</tbody>
</table>

Table 14: Table of reduced dataset prediction accuracies

It should be noted the total number of predictions made has now changed for the altered datasets. The total number of samples in the original parent dataset was 500, 30% of these were used as test data therefore 150 predictions were made. There are 350 samples in the altered datasets meaning 105 predictions have been made.

A 10.64% increase in the average prediction accuracy has been witnessed using the dataset where the 3 lowest scoring genres from the previous instance have been removed. The accuracy of the dataset which removed the 3 highest scoring genres has an average accuracy 4.03% lower. This suggests house, industrial and UK-garage are more difficult to classify based on the chosen input variables, and that it is not the random permutation within the original runtime instance that caused the low scores. Additionally, the increase in accuracy for A1 deviates far greater from the original average than A2 negatively deviates. This could suggest a lower total class value, 7 as opposed to 10, benefits the KNN model. This is in-line with pre-existing research and further justified the significantly low accuracy for the dataset of sub-genres.

**Sub-genres Confusion Matrix**

The confusion matrix referred two in this section can be found in the appendix.

Results observed from the confusion matrix generated using the sub-genres dataset provides some key information. Their range of accuracies between genres deviates massively from 0% - 85.71%. There seems to be no specific group of sub-genres that score better than others as high accuracy values were found in genres that range from ambient to dark. However, there is a noticeably large separation between the high scoring accuracies and the low. For Ko, 38, the average percentage accuracy of the KNN algorithm was 27%. 23 sub-genres scored lower than the average percentage accuracy, 8 of these scoring 0%, heavily negatively skewing the average accuracy. The spread of accuracies for the above average scoring sub-genres is more evenly spread beyond the 27%. The large spread of accuracies further suggests some genres may be difficult to distinguish between due to their definitive features that are disconnected to the selected variables passed into the KNN algorithm. Interestingly, acid sub-genres that were detailed earlier scored low accuracies which supports the hypothesis made deduced using the selected variables for KNN analysis.
5 – Conclusion

This section condensed key research points inferred and assesses the success of the research project.

5.1 – Summary of Key Research Points Inferred

Referring to the main research question of this thesis, the machine-learning algorithm applied to electronic genres of music seems to have some promise in classifying genres. While using the detailed Spotify track features, it seemed relatively capable at distinguishing between larger groups of genres such as those within the parent dataset. The KNN model scored too low concerning sub-genres to justify any practical use. Potential causes for this are explained in the following key points.

The KNN algorithm seemed to fare better with the parent genres dataset. It was determined that two major factors played a role in this. The lower number of possible classes meant there was statistically less margin for error, consequently boosting the average prediction accuracy of the KNN model. Additionally, there is likely greater distinction between parent genre groups compared to sub-genres, as the sub-genres confusion matrix expresses a lack in high predictability for any one group of sub-genres. Class-overlapping between certain sub-genres could have severely impacted the average accuracy. This leads to the next research point.

The quirks of sub-genres may have blurred the lines between the parent groups they stem from, ultimately reducing accuracy. Intangible attributes that cannot be quantified through Spotify features or numerical scales may be overly dominant in determining what genre a song is, reducing the impact variables assessed within this project have on a given song.

The optimal K-value seemed to be 15% below the root of the total analyzed samples. From this, a formula was developed to describe this trend discovered within this thesis.

The KNN model struggled significantly to classify genres within house, industrial and UK-garage. This contrasted the hypothesis that the strong danceability of house, due to its strong beat pattern, would make it easier to distinguish. This was further justified when these genres were removed from the dataset and the accuracy increased significantly.

Upon reflection, a large component of house is the inclusion of claps and snares to elevate the overall sound above the bass. This is not as often seen in techno, which shares a similar tempo range and beat pattern, but does not include these instruments so often; however, techno seemed to be more predictable. This suggests these attributes have little bearing on the concerning track features. The inclusion of speechiness in the KNN model may have been justifiable due to the high predictability score for disco, which was determined to be the most vocal set of genres within the pilot study.

5.2 – Research Applied to Future Work

This research project has highlighted the capabilities and drawbacks of machine-learning techniques applied to a highly subjective and qualitative field of research. Quantitative evaluation is highly beneficial, however, when attempting to develop an analysis system that can only comprehend the numerical
aspects of digital music. DJ software cannot consider the ‘acid’ traits of a group of songs, or the history of its origins; it can only deconstruct the songs with tangible scalable values, such as those used here. This concept may hopefully open individuals minds as to how musical analysis could be further developed and could provide a foundation for others to work from.

5.3 – Research Project Reflection

It is apparent outcomes of KNN analysis within this project did not render perfect results. This was to be expected however as this is has been an attempt at gaining some traction on how electronic genres of music can be made more quantifiable and scalable. The project has certainly raised many research points and has presented many improvements that could be made within future iterations and relating topics.

For a start, greater variation of input variables could have been assessed. Iteration of testing could have included a different KNN distancing metric or training to test data ratio to squeeze more accuracy out of the program. Additionally, testing with a greater variation of classes may have provided more insight. To provide a stronger point of reference, analysis of non-electronic genres could have been examined. This whole process expresses that a multitude of control variables could be used within this type of research, it is important to select only what can be handled within reason. The use of too many variables causes overcomplexity and is highly time-consuming.

To conclude, this project has been successful in generating some valuable insight into the nature of electronic genres of music and how well machine-learning based algorithms may be add recognizing and predicting them. From a pragmatic perspective, the values outputted by the KNN algorithm are arguably not necessarily usable, but there is still plenty of potential to increase the efficiency of these algorithms.

6 – List of References


Shawhin Talebi, 2021, *Independent Components Analysis (ICA)*. Available at: https://towardsdatascience.com/independent-component-analysis-ica-a3eba0ccce35


Wikipedia, Edited 28th October 2022, Euclidean distance.


Open-source Code and Websites Used

**Spotify Research API** – https://developer.spotify.com/documentation/web-api/

**Every Noise Spotify Genre Database** by Glenn McDonald – https://everynoise.com/engenremap.html

**GSA open-source script** by Ole Adrian Heggli – https://github.com/OleAd/GeneralizedSpotifyAnalyser

**Developed open-source KNN genre prediction MATlab Livescript** by Ned Fellenor – https://github.com/NFellenor/MATlab_Musical_Genre_KNN_Analysis_Livescript
## 7 – Appendices

*Extract of Sub-genre Confusion Matrix (Where K = Ko)*

![Confusion Matrix Table](image)

*Figure. Columns 1-10, rows 1-20 of Sub-genres confusion matrix*
**Figure. Columns 11-20, rows 1-20 of Sub-genres confusion matrix**

<table>
<thead>
<tr>
<th></th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
<th>27</th>
<th>28</th>
<th>29</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5.5556</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>5.0990</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5.6667</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>18.7500</td>
<td>9.0909</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9.0909</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6.2500</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>27.2727</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>13.3333</td>
<td>6.2500</td>
<td>0</td>
<td>5.5556</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>9.0909</td>
<td>0</td>
<td>0</td>
<td>13.3333</td>
<td>0</td>
<td>0</td>
<td>12.5000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5.5556</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0</td>
<td>9.0909</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>13.3333</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7.1429</td>
<td>0</td>
<td>6.6667</td>
<td>0</td>
<td>0</td>
<td>9.0909</td>
<td>5.5556</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6.6667</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>13.3333</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5.5556</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

**Figure. Columns 21-30, rows 1-20 of Sub-genres confusion matrix**

<table>
<thead>
<tr>
<th></th>
<th>31</th>
<th>32</th>
<th>33</th>
<th>34</th>
<th>35</th>
<th>36</th>
<th>37</th>
<th>38</th>
<th>39</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>17.5471</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6.6667</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12.5000</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5.5556</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6.6667</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5.5556</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>11.1111</td>
<td>0</td>
<td>1.7619</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11.1111</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>14.2857</td>
<td>5.8824</td>
<td>0</td>
<td>9.2528</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>7.1429</td>
<td>0</td>
<td>5.5556</td>
<td>14.2857</td>
<td>0</td>
<td>0</td>
<td>5.5556</td>
<td>0</td>
<td>6.6667</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0</td>
<td>11.1111</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11.1111</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
<td>14.2857</td>
<td>0</td>
<td>4.7619</td>
<td>13.3333</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6.6667</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11.1111</td>
<td>25</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>14.2857</td>
<td>0</td>
<td>22.2222</td>
<td>4.7619</td>
<td>0</td>
<td>6.6667</td>
<td>5.5556</td>
<td>0</td>
<td>6.6667</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>7.1429</td>
<td>5.8824</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>7.1429</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Figure. Columns 31-40, rows 1-20 of Sub-genres confusion matrix**

56
Figure. Columns 1-10, rows 21-40 of Sub-genres confusion matrix

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>0</th>
<th>11.111</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>10</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>0</td>
<td>0</td>
<td>11.111</td>
<td>0</td>
<td>31.5789</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16.6667</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5.2632</td>
<td>9.5238</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8.3333</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>6.2500</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>25</td>
<td>6.6667</td>
<td>12.5000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>15.3846</td>
<td>0</td>
<td>0</td>
<td>8.3333</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>0</td>
<td>6.2500</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9.5238</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>27</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>28</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4.3478</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>6.2500</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4.7619</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>31</td>
<td>13.3333</td>
<td>12.5000</td>
<td>0</td>
<td>0</td>
<td>5.2632</td>
<td>9.5238</td>
<td>0</td>
<td>0</td>
<td>8.3333</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>33</td>
<td>6.6667</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>34</td>
<td>20</td>
<td>6.2500</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>35</td>
<td>0</td>
<td>22.2222</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>36</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>37</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>38</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>39</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

| 20  | 0   | 0   | 5.5556 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 47.0968 |
| 21  | 0   | 11.111 | 11.111 | 0   | 8.3333 | 0   | 0   | 6.2500 | 35.7143 | 11.7647 |
| 22  | 0   | 0   | 0   | 0   | 0   | 0   | 16.6667 | 0   | 0   | 0   | 0   |
| 23  | 0   | 0   | 0   | 7.1429 | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| 24  | 0   | 22.2222 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| 25  | 0   | 0   | 0   | 0   | 0   | 0   | 8.3333 | 0   | 0   | 12.5000 | 0   |
| 26  | 0   | 22.2222 | 5.5556 | 0   | 0   | 0   | 5.5556 | 0   | 7.1429 | 0   |
| 27  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| 28  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| 29  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| 30  | 0   | 0   | 0   | 5.5556 | 0   | 0   | 0   | 0   | 0   | 0   |
| 31  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| 32  | 9.0909 | 0   | 5.5556 | 0   | 0   | 0   | 0   | 0   | 7.1429 | 0   |
| 33  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| 34  | 9.0909 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| 35  | 9.0909 | 0   | 5.5556 | 0   | 0   | 16.6667 | 0   | 12.5000 | 0   |
| 36  | 0   | 0   | 0   | 0   | 0   | 8.3333 | 0   | 0   | 0   | 0   |
| 37  | 0   | 0   | 0   | 7.1429 | 0   | 0   | 0   | 0   | 0   | 0   |
| 38  | 18.1818 | 5.5556 | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| 39  | 0   | 0   | 11.111 | 0   | 0   | 8.3333 | 0   | 0   | 0   | 0   | 0   |
| 40  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
**Figure. Columns 11-20, rows 21-40 of Sub-genres confusion matrix**

```
21  80  0  0  214286  0  0  0  0  9.0895  0
22  0  27.2727  36.3636  0  0  0  0  0  9.0895  0
23  0  27.2727  54.5455  0  0  0  13.3333  18.7500  0  0
24  0  0  0  28.5714  0  20  0  6.2500  0  5.5556
25  0  0  0  0  73.3333  0  13.3333  6.2500  0  0
26  0  0  0  25.7143  0  60  0  0  0  0
27  0  0  0  0  6.6667  0  13.3333  6.2500  0  0
28  0  0  0  0  0  0  6.6667  12.5000  0  0
29  0  0  0  0  0  0  0  45.4545  0  0
30  0  0  0  0  0  0  0  0  0  0
31  0  0  0  0  0  0  0  0  0  5.5556
32  0  0  0  0  6.6667  0  0  0  0  5.5556
33  0  0  0  0  0  0  0  0  0  0
34  0  0  0  0  0  0  0  0  0  0
35  20  0  0  0  0  0  0  0  0  22.2222
36  0  0  0  0  0  0  0  0  0  0
37  0  0  0  0  0  0  0  0  0  0
38  0  0  0  7.1429  0  0  6.6667  6.2500  0  0
39  0  0  0  0  0  0  6.6667  0  0  11.1111
40  0  0  0  0  0  0  0  0  0  0
```

**Figure. Columns 21-30, rows 21-40 of Sub-genres confusion matrix**

```
21  0  0  0  0  0  66.6667  16.6667  6.2500  6.6667  0
22  0  0  5.5556  0  0  0  0  0  0  0
23  7.1429  0  22.2222  4.7619  0  0  5.5556  0  0  0
24  0  0  0  0  0  0  0  6.2500  0  5.5556
25  0  5.8824  0  38.0952  0  0  11.1111  12.5000  0  0
26  0  0  0  0  0  0  0  0  0  0
27  0  0  0  9.5238  6.6667  0  0  6.2500  0  6.6667
28  0  0  5.5556  0  0  0  0  0  0  0
29  0  0  0  0  0  0  0  0  0  0
30  0  0  0  0  0  0  0  0  0  0
31  14.2857  0  0  0  0  0  0  0  0  0
32  0  41.1765  0  0  0  0  0  0  0  6.6667
33  0  0  11.1111  0  0  0  0  0  0  0
34  7.1429  0  0  9.5238  0  0  0  0  0  0
35  0  0  0  0  60  5.5556  0  0  53.3333  0
36  7.1429  0  0  0  0  6.6667  0  0  0  0
37  0  0  0  0  0  0  0  0  0  0
38  0  5.8824  0  0  0  0  0  0  18.7500  0  6.6667
39  0  5.8824  0  0  13.3333  0  0  0  0  5.5556
40  0  0  0  0  0  0  0  0  0  0
```

**Figure. Columns 31-40, rows 21-40 of Sub-genres confusion matrix**

```