ATHABASCA UNIVERSITY

A LEARNING ANALYTIC FOR MUSIC PROGRESS

BY

JOEL BURROWS

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION SYSTEMS

FACULTY OF SCIENCE AND TECHNOLOTY, SCHOOL OF COMPUTING AND INFORMATION SYSTEMS

ATHABASCA UNIVERSITY

AUGUST, 2018

Athabasca University
FACULTY OF GRADUATE STUDIES

The future of learning.

Approval of Thesis

The undersigned certify that they have read the thesis entitled

A LEARNING ANALYTIC FOR MUSIC PROGRESS

Submitted by

Joel Burrows

In partial fulfillment of the requirements for the degree of

Master of Science in Information Systems

The thesis examination committee certifies that the thesis
and the oral examination is approved

Co-Supervisors:
Dr. Vivekanandan Kumar
Athabasca University

Dr. M. Ali Akber Dewan
Athabasca University

Committee Member:
Dr. Kinshuk
University of North Texas

External Examiner:
Dr. Mickey Vallee
Athabasca University

August 21, 2018

1 University Drive, Athabasca, AB, T9S 3A3 Canada
P: 780.675-6821 | Toll-free (CAN/U.S.) 1.800.788.9041 (ext 6821)
fgs@athabascau.ca | fgs.athabascau.ca | athabascau.ca

Abstract

The objective ear is an application that, given a pair of performances of a piece of music, judges the amount of progress made between the two performances. The application has two components: an evaluator and a classifier. The evaluator component analyzes each performance to generate a vector of metrics. These vectors are subtracted from each other to give a vector of differences. The difference vector is used as input to a decision tree, a machine learning classifier, which assigns a level of progress to the pair of performances. Testing of the classifier shows that the application provides accurate assessments and could be used in music education environments to aid students in assessing their progress, and to provide useful data on how music students progress.

Table of Contents

List of Tables

List of Figures

Chapter I - INTRODUCTION

Assessment is an important part of the learning experience. It provides a student with an indication of progress, information that is critical to students, teachers, and administrators when preparing and planning for education. Assessment itself can be error-prone. Assessment cannot be easily generalized because it must be appropriate for the subject it assesses. For example, to assess a student's knowledge of medical terminology, a series of multiple-choice assessments might be sufficient. To assess a student's skill in music, no simple assessment exists, but instead an assessor must make error-prone judgements about the quality of a performance. This research investigates using machine-learning techniques to produce an application which gives objective and accurate assessments of a student's progress with musical tasks.

Statement of the Problem

Music may be a diverse field, rich with history and theory, but music education primarily focuses on improving a student's skill performing music. A musician is not a person who knows a lot about music, but to be a person who can skillfully perform music. Thus, skill assessment is a necessary part of music education.

Music lessons usually follow a pattern. At any given time, a student is working on a limited number of pieces and technical exercises. During the lesson, the teacher listens to the student's attempts at these pieces and exercises, suggests improvements, and demonstrates skills. As a student masters pieces and exercises, the teacher directs the student to new work to master. The practice period between lessons is critical for the student to make progress with these pieces and exercises by practicing the material. At the next lesson, the teacher must recall the student's previous skill at a piece or exercise and assess the student's progress. The teacher must either recall the previous performance, or have taken good notes about problems with the previous performance. Students, during the practice period, must evaluate their progress by listening as they practice. Performing and critically evaluating a performance both require careful attention and focus, and doing both activities at the same time is difficult. Thus, music assessment activities are faulty, and can lead to improper judgments of progress.

Even with good notes, a teacher's assessment can be faulty. A teacher's training may prepare him or her for detecting specific kinds of errors, such as improper rhythms; but other errors, such as pitch errors, may be overlooked. Different types of music teachers are faulty at different aspects of assessment, and even aural training does not prepare a teacher for all types of errors (Stambaugh, 2016). Several factors can affect a teacher's ability to detect pitch errors such as the music's tonality, the thickness of the musical texture, and the size of the interval when a student makes a pitch error. Listeners with perfect pitch are far superior at detecting pitch errors, but limiting the pool of music teachers to those few people with perfect pitch is impractical (Groulx, 2013).

Students show great variety in their ability to judge their own performances. Voice students seem to have an advantage, and even young voice students can accurately judge their performance when sight-singing (Darrow, 2006). Instrumentalists make errors in judgment based on their level of skill. Strong students evaluate themselves overly harshly, and weaker students are erroneously optimistic of their performances (Hewitt, 2015).

Russell points out that teachers and students need to "attain higher levels of objectivity and accuracy in measuring performance," but "the reliability of music performance assessments is moderate at best." An objective assessment can help teachers better address their students' weaknesses, and it also makes music education in schools more attractive to those who demand "predictable educational outcomes" (Russell, 2015).

Significance of the Problem

Students and teachers both benefit from the development of a tool that acts as an objective ear. Having a tool that provides immediate feedback rather than just at the end of a course is extremely valuable because it alerts the teacher and the student of problems early enough to correct them. Using this tool, teachers can get an objective assessment of progress between lessons and better plan for and direct the student. Access to this progress data, especially over time, allows a teacher to experiment with different approaches for a student, and gives reliable data in personalizing an approach to music education for that student. As well, for the student, it provides access to assessment during the practice period between lessons. For some students, this can be motivating, driving the student to achieve the maximum level of progress each week.

This tool provides strong new possibilities to researchers. Providing assessment data allows researchers to develop far more accurate theories of music education. These theories could be large-scale theories that provide generalizations across all students. For example, a learning management system that tracks student practice time could provide the data to accurately model the impact of practice time on progress. Generalizations such that "45 minutes of practice time a day is sufficient for an intermediate student to make strong progress" provide useful heuristics for students.

This research works with skill-based assessments, and may provide insights that can be generalized beyond music-based learning analytics. Some skills can be difficult to assess objectively, although not all skills fall into this category. For example, arithmetic skills can often be assessed objectively, as can archery skills. What each of these easily-assessed skills has in common is the skill’s teleological nature and that the skill’s end result can easily be measured. For other more process-oriented skills, such as gymnastics or music, the entire process must be evaluated. A performance of a piece of music is not evaluated by how well the performer plays the last note, but every note in the performance is evaluated, vastly increasing the data that must be measured and evaluated. Research into process-oriented assessment can create techniques to refine this data into a form that benefits the learning analytics community.

Chapter II – LITERATURE REVIEW

The literature review covers three different areas of research. First, music assessment is reviewed. Because assessment problems are the source of the problem this research addresses, it is important to understand fully the problems of assessment. Next, automated music analysis is reviewed. The field of automated music analysis investigates algorithms for extracting properties from music. Because the objective ear is structured as a machine learning classifier, it needs features for which the classifier can be trained to perform the classification. These features correspond to various metrics extracted from performances, and automated music analysis provides algorithms used to perform these extractions. Finally, the objective ear is a tool that can be used within a learning analytic environment, so the field of learning analytics is reviewed to place the objective ear within its proper context within the field of learning analytics.

Music Assessment

A music educator needs to be able to objectively assess music skill. Students deserve fair grades, but assessment in music is more important than a grade at the end of the learning process: assessment is critical to the learning process itself, providing the feedback necessary to guide a student's learning plan. Is learning to play music any guarantee of being able to assess it? Learning music does provide many benefits beyond just improving musical performance skills (Jaschke, Honing and Herder, 2018). Stambaugh, who focuses on ensemble teaching, states that a “music teacher’s ability to detect errors in ensemble performance is a critical skill for effective rehearsing and teaching" (Stambaugh, 2016). Hewitt, focusing on students, states that students “need to monitor their progress while practicing and then make judgments concerning the success - or lack of success - of their efforts. This information then is used to readjust goals and practice plans” (Hewitt, 2015). Even though assessment is important, Russel states that "the reliability of music performance assessments is moderate at best" (Russel, 2015). Russel also makes the point that an objective assessment gives education administrators more confidence in a music program of study.

The research community has focused on understanding how music assessment is difficult. Their research has determined many factors that affect assessment quality. Stambaugh looks at the kind of training and experience a music teacher has. "Band majors were most successful at locating rhythm errors, while choral majors were most successful at hearing pitch errors" (Stambaugh, 2016). On the other hand, she finds that the ability to detect errors is intransient across factors such as age, gender, and years of experience. This is a troublesome finding because a broad education for music educators is unrealistic since specific training for preferred instruments usually dominates a music teacher's education. Groulx, limiting his research to pitch errors, finds several conditions that affect error detection. "Conditions that contributed to more successful error detection included errors involving larger interval deviations from the original pitch, greater alteration of melodic contour, thinner textures, errors that deviated from the tonal framework of the piece, and live performance on familiar acoustic instruments" (Groulx, 2016). He also discovers that those with perfect pitch are far superior at detecting pitch errors. Darrow finds that young singers can accurately evaluate their sight-singing abilities, although it would be interesting to compare self-evaluation with singing to self-evaluation with musical instruments (Darrow, 2006).

Music assessment skills are correlated with musical ability. Hewitt researches self-evaluation with students and finds that "higher-performing students in the present study underrated their performance ability while lower-performing students overrated their achievement" (Hewitt, 2016). Wapnick and Darrow investigate the difference between assessing a full, long piece of music, and assessing excerpts individually, and they discover differences. They also find that, while listeners may agree on which performances are better or worse, when comparing performances, they differ greatly in the actual evaluation (Wapnick & Darrow, 2013).

Although these research activities provide a greater understanding of the various factors that impact music assessment, few give direct guidance as to how to improve assessment. Hewitt shows that improving a person's musical ability makes that person a better assessor (Hewitt 2016). Russel provides an assessment model. This assessment model attempts to capture all the various aspects of assessment, making them explicit. This approach provides a more stable assessment.

Automated Music Analysis

One method of assessing a performance is to compare a performance to an ideal. This approach is used by Bora, Tufan, and Semih to create a tool that "is meant to supply information about the progress of the student." That is, the tool is meant more to help with the learning progress of the student than to provide an evaluation at the end of a learning program. They compare pitch and rhythm against the ideal performance (Bora, Turan, & Semih, 2000). This approach has already been commercialized and is available in various music products.

A related area of research is score matching. The goal of score matching is to map each note of a performance to a note on the score. This area has also been researched sufficiently for commercial products to be common. The area is not without challenge though, the greatest problem being that a score does not specify all aspects of a performance. Gingras and McAdams state that "automated score-performance matching is a complex problem due to the use of expressive timing by performers and the presence of notes that are unspecified in the score, such as performance errors and ornaments" (Gingras & McAdams, 2011). Heijink, Windsor, and Desain state that "matching is a complex task. It is difficult because performers make errors (score notes may be missing in the performance, or performance notes may be missing in the score), performers make use of expressive timing, and scores are frequently underspecified" (Heijink, Windsor, & Desain, 2000). This problem of underspecification is the greatest weakness of any assessment approach that is based on a specific score. Although a score may specify the notes to play and when to play them, the actual performance may introduce major differences in both aspects.

To assess the performances, they need to be analyzed musically. Doing so builds upon research done in the field of automated music analysis. Using research in this field, software algorithms can be implemented to detect various kinds of music errors. The field of music analysis is itself undergoing a paradigm shift, and these software algorithms often conform to the new paradigm. In 1983, Fred Lerdahl and Ray Jackendoff published *A Generative Theory of Tonal Music*. Since then, their book’s approach has been increasingly adopted by the research community. Part of its appeal to the automated music analysis community is that it states clear rules for determining several attributes of music along with preference rules that can be applied to ambiguous situations. This approach organizes the analysis in a hierarchical form that can be easily created and manipulated by software processes (Lerdahl, 1983). Hamanaka, Hirata, and Tojo successfully implemented the rules specified by the theory for three of the four major subdivisions of the theory (Hanamaka, Hirata, & Tojo, 2006). Their work showed that although the theory may be laid out in clear rules, effort was still necessary to implement those rules on a computer. Concepts that appear simple on paper can be much more complex when specified formally to the extent that they can be programmed on a computer. One section that they implemented is the grouping rules. These are useful because they can be used to determine the overall structure of a piece of music. Music from the Classical era conforms to hierarchical grouping concepts. For example, a piece of music in sonata form has a clear mid-point between the end of the exposition and the start of the development. The exposition itself can be divided into the two major themes it introduces, and each these can be broken down into multiple phrases, short sections of music that express a musical idea. Understanding this grouping is important when evaluating a performance, because certain aspects of the performance vary depending on the grouping. This is particularly important when it comes to timing, as performers in the Classical era frequently slow down the tempo at the end of a grouping; and the higher the group in the hierarchy, the more significant the slowdown.

Music theorists perform many operations on a piece of music when analyzing it, but the most common analysis is a harmonic analysis. A harmonic analysis of a piece of music maps the piece of music to an underlying harmonic structure, and identifies the role of each note in that harmonic structure. Because of its dominance, automated harmonic analysis has been an active area of research for decades with many different approaches applied to the problem. For example, Pardo and Birmingham focus on segmentation, breaking the music into groups. Once segmented, harmonic analysis can be more accurately applied (Pardo & Birmingham, 2002). Segmentation provides many benefits beyond just better harmonic analysis because many aspects of music are affected by the context given by a grouping structure. Bas De Has, Magalhaes, Wiering and Veltkamp use functional programming to perform music analysis and exploit the capabilities of functional programming in performing the analysis, but they discovered that a grouping structure would have made their research stronger (Bas De Has, Magalhaes, Wiering & Veltkamp, 2013). Raphael and Stoddard take a probabilistic approach to harmonic analysis, and use a quasi-hidden Markov model approach to determining the harmonic progressions of a piece of music (Raphael & Stoddard, 2004). This approach works because the harmonic structure of a piece of music follows predictable patterns, making it possible to determine likely harmonic progressions.

These research activities hint at approaches to pitch error detection that can be taken without necessarily doing a full harmonic analysis first. Pearce and Wiggins use an nGram algorithm on monophonic melodies to predict the next note in a melody (Pearce & Wiggins, 2004). Their model could be extended to work with polyphonic music, especially if the voices in a piece of music are first separated. Chordia, Sastry and Senturk used variable-length hidden Markov models to predict patterns in tabla music (Chordia, Sastry, & Senturk, 2011). This approach could be used to create a predictive model for pitches and used to detect pitch errors. Although these researchers take very different approaches, they have shown success using various probabilistic modeling techniques, giving confidence that a predictive pitch model can be created to determine pitch errors in a performance. Unfortunately, the emphasis of research is on analyzing existing pieces of music rather than detecting probable errors. Thus, although many of these activities point towards a probabilistic model for detecting pitch errors, none of these researchers provide an actual algorithm that can be directly used.

Pitch and rhythm are the two main components of music. Rhythm analysis emphasizes the tempo of a piece of music, the metre of a piece of music, and repeating rhythmic patterns. Gouyan and Dixon reviewed many different approaches to automated rhythm description (Gouyan & Dixon, 2005). The basic requirement for error detection is a system that can reliably determine the pulse of a piece of music across different performances of the same piece. A common problem with pulse detection is determining the pulse level. For example, one system may assign the pulse to the quarter notes whereas another system may assign it to the half notes. This does not matter for error detection if the pulse is assigned reliably. Cemgil and Kappen apply Kalman filtering to determine the tempo (Cemgil & Kappen, 2000). Although this approach can detect the pulse level, it is a highly complex approach to finding the tempo. Dixon instead uses clustering techniques to find the most common note value and uses this to determine the pulse (Dixon, 2001). Although Dixon admits that his algorithm has trouble finding the correct level of the pulse, his algorithm should still be reliable across multiple performances of the same piece of music. This is the approach that is indeed used in this research to find the tempo. Identifying likely rhythm errors has not been an area of research in automated music analysis.

Pitch and rhythm are the more technical aspects of a musical performance. The artistry in a musical performance comes from the interpretation of a piece of music. Interpretation can affect the pitch by adding ornamentation to notes. For example, rather than playing a single note, a musician may trill it by quickly alternating between the note's pitch and an adjacent pitch for the duration of the note. Interpretation can affect rhythm by using expressive timing. Expressive timing changes the duration of notes. For example, at the end of a piece of music, a musician may slow down, playing notes for a longer duration. Interpretation strongly affects the dynamics of a piece of music. Dynamics refers to how loudly or softly a note is played, and although dynamics are a key aspect of interpretation, Widmer finds that dynamics follow regular rules, and errors in dynamics can be evaluated. He applies machine learning techniques to a corpus of performances, and creates several rules of how dynamics are applied (Widmer, 2002). This is not surprising because, although dynamics are a part of interpretation, most changes in dynamics are slight and predictable when a competent musician shapes a line of music. Widmer’s rules can be applied to a performance to detect when unexpected dynamic changes occur. This same research also identifies rules in applying expressive timing. Thus, these rules make possible the detection of interpretation errors.

These approaches have something in common: they isolate a single aspect of music and analyze it. Temperley unifies several different aspects of music: pitch, rhythm, and grouping; and integrates them within a probabilistic model (Temperley, 2009). None of these aspects of music exists in isolation, and by combining these analyses, Temperley can provide a stronger model. Because automated music analysis has been such an active research area for so long, many aspects of it have adequate solutions, and much of the current research is approaching more sophisticated approaches, such as combining multiple analyses.

Learning Analytics

What is learning analytics? "Learning Analytics is a sprawling term, at times referring to complex predictive models and at other times to routine tasks such as classroom allocation and energy conservation (Siemens, 2012.)" Although "the term ‘analytics’ has been used in many different ways in recent years and has become part of the buzzword jargon that sometimes seeps into new technology applications and products (Picciano, 2012,)" we can define analytics as "the science of examining data to draw conclusions and, when used in decision making, to present paths or courses of action (Picciano, 2012.)" Learning analytics extends "the potential of analytics to the level of individual learning, by selecting, capturing and interpreting data on teaching and learning activities, with the goal of improving teaching and learning outcomes (Macfadyen and Dawson, 2012.)"

Experienced teachers have hunches that often prove to be true, but hunches can be an error-prone method of making decisions and are difficult to justify. Learning analytics can provide justifications for these hunches, giving educators confidence to act in ways the improve their student's success (Dietz-Uhler and Hurn, 2013.) Understanding what actions can improve learning is critical. "Be it the assessment marks distribution in a classroom context or the mined pattern of best practices in an apprenticeship context, analysis and discovery have always addressed the elusive causal question about the need to best serve learners’ learning efficiency and the need to make informed choices on a learning context’s instructional effectiveness (Ebner, Kinshuk, Wohlhart, Taraghi, and Kumar, 2015)." The key mechanism in learning analytics is the discovery of patterns in data that provide insight that can lead to action. "Learning Analytics has the potential for new insights into learning processes by making hitherto invisible patterns in the educational data visible to researchers and end users, and to enable development of new instruments for everyday educational practice (Greller and Drachsler, 2012)."

A variety of applications derive from learning analytics. Learning analytics "includes techniques such as predictive modeling, building learner profiles, personalized and adaptive learning, optimizing learner success, early interventions, social network analysis, concept analysis, and sentiment analysis (Siemens, 2012.)" A key application is in improving the delivery of curriculum. "Analytics can add distinct value to teaching and learning practice by providing greater insight into the student learning process to identify the impact of curriculum and learning strategies, while at the same time facilitating individual learner progress (Macfadyen, Dawson, Pardo and Gasevic, 2014.)" Thus, learning analytics can be applied to the data gathered during the learning process to understand how learning can be improved. Even imperfect tools can still provide value to educators. For example, automated essay scoring systems to evaluate an essay based on argumentation and rhetoric, but even providing a shallow assessment of text production skills can provide value to an educator (Kumar, Fraser, and Boulanger, 2017.)

This approach can be formalized in the evaluation of learning designs. "A learning design captures the pedagogical intent of a unit of study. Learning designs provide a broad picture of a series of planned pedagogical actions rather than detailed accounts of a particular instructional event (Lockyer, Heathcote, and Dawson, 2013.)" A learning design may make strong claims, but "learning analytics allow us to test those assumptions with actual student interaction data in lieu of self-report measures such as post hoc surveys (Lockyer and Dawson, 2011.)" As a next step, the combination of learning design and learning analytics can be used to provide "pedagogical recommender systems" that can recommend a suitable learning design for a particular context (Lockyer, Heathcote, and Dawson, 2013.)

Another application of learning analytics is personalized learning. "It is expected that personalised learning has the potential to reduce delivery costs while at the same time creating more effective learning experiences, accelerating competence development, and increasing collaboration between learners (Greller and Drachsler, 2012.)" Associated with personalized learning are concepts like self-reflection and motivation (Santos, Govaerts, Verbert, and Duval, 2012.) This area has challenges. "Awareness and reflection support for students are consequently highly important aims of learning analytics. The existence and impact of these aims, however, are hard to measure due to the lack of standards that the student support of Learning Analytics tools can be measured against (Scheffel, Drachsler, Stoyanov, and Specht, 2014.)"

A related application of learning analytics is the creation of learning models of students which can then be used to personalize learning. "Student models represent information about a student’s characteristics or state, such as the student’s current knowledge, motivation, meta-cognition, and attitudes. Modeling student individual differences in these areas enables software to respond to those individual differences, significantly improving student learning (Baker and Yacef, 2009.)" This model is not static but "the learners' profile must be adapted according to the evolution of the observed learning behaviour (Taraghi, Saranti, Ebner, Muller, and Grosmann, 2015.)" These models could be useful in approaches the student takes to other courses, or in informing the student of how best he or she learns (Bull and Kay, 2013.) A smart learning environment can perform competence analysis on a variety of types of evidence, providing guidance to students in their preparation for future activities (Kumar, Boulanger, Seanosky, Kinshuk, Panneerselvam and Somasundaram, 2014.)

One application that has proven its success is the identification of students who are at risk of failing or withdrawing. Prediction is not perfect. "Whereas the holy grail of predictive models in higher education would likely be one that could predict graduation at the time a student applies for admission, the reality is that the elapsed time between the start and end of college is years long, creating the opportunity for a multitude of factors to interfere with a student's progress (Barber and Sharkey, 2012.)" Yet predictions can often be accurate, because "a student enrolled in a course will often display signs of course failure before either formally withdrawing or disappearing altogether (Barber and Sharkey, 2012.)" An implementation of this kind of prediction is Course Signals at Purdue University in which students at risk are identified, giving academic advisors the opportunity to intervene (Arnold and Pistilli, 2012.) Using this approach, "learning analytics can provide insights into what is happening with the learner in nearly real-time. Armed with this information, faculty can make suggestions to students that will help them succeed (Dietz-Uhler and Hurn, 2013.)"

Assessment is an integral part of both education and learning analytics. "Assessment practice will continue to be intricately intertwined both with learning and with program accreditation and accountability measures (Macfadyen, Dawson, Pardo and Gasevic, 2014.)" Assessment data is of interest to many different stakeholders because of the multitude of ways it can be used. "While the majority of education practitioners lean towards assessment as a process for improving student learning, assessment nonetheless remains firmly positioned as an important tool for determining accountability and demonstrating quality (Macfadyen, Dawson, Pardo and Gasevic, 2014.)" Furthermore, assessment data itself can be motivational for students. Gathering this data can be challenging. "While measurement and assessment of learning is a major objective of learning analytics, it is also a demanding experience for many teachers due to the heavy workload and time-consuming nature of the assessment activities (Goggins, Xing, Chen, Chen, and Wadholm, 2015.)" Skills, in particular, are difficult to assess. "One of the difficulties is that current assessment instruments are based on products (an exam, a project, a portfolio), and not on processes (the actual cognitive and intellectual development while performing a learning activity), due to the intrinsic difficulties in capturing detailed process data for large numbers of students (Blikstein, 2011.)"

As the field of learning analytics has developed, so have theoretical frameworks for analyzing, understanding, and developing learning analytics tools. Greller and Drachsler identify six dimensions in their learning analytics framework: "stakeholders, objectives, data, instruments, external constraints, and internal limitations (Greller and Drachsler, 2012.)" They identify that "the main stakeholder groups of Learning Analytics in formal learning situations are learners, teachers, and educational institutions. These may be expanded or substituted by other stakeholder groups, such as researchers, service providers, or governmental agencies (Greller and Drachsler, 2012.)" Different stakeholders "require analytics work on different scales and at different granularities. The choice of target audience therefore affects how researchers conceptualise problems, capture data, report findings, act on their findings and refine their models (Ferguson, 2012.)" Also of note in Greller and Drachsler's model is the instruments dimension, under which they "also subsume conceptual instruments such as theoretical constructs, algorithms, or weightings, by which we mean different ways of approaching data. These ways in the broadest sense 'translate' raw data into information (Greller and Drachsler, 2012.)" Siemens provides a simpler framework of learning analytics as "two overlapping components: techniques and applications. Techniques involve the specific algorithms and models for conducting analytics. Applications involve the ways in which techniques are used to impact and improve teaching and learning (Siemens, 2013.)"

It must be noted that learning analytics does not simply analyze data, but it also seeks to

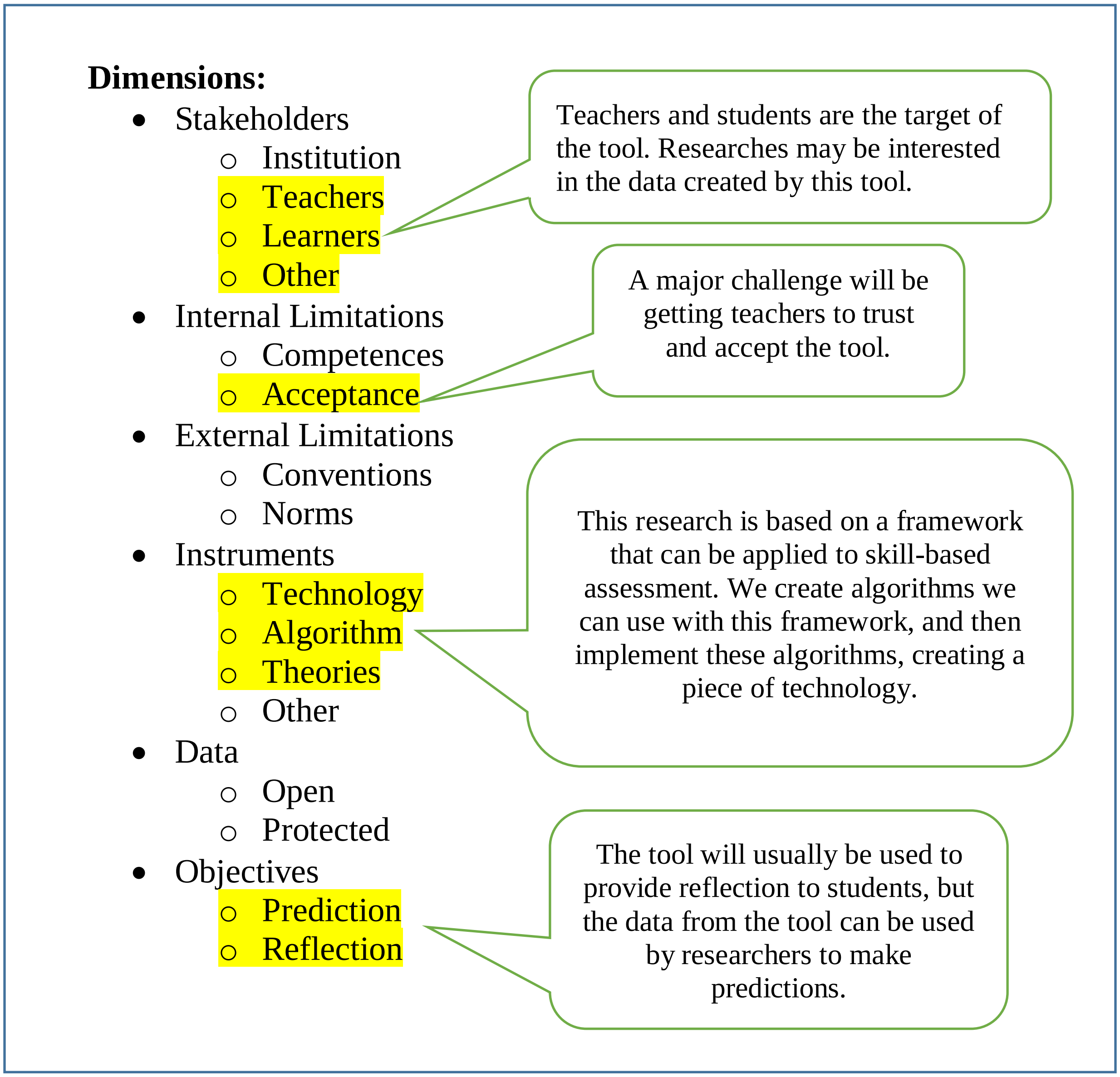

Figure 1 Placement of the tool within Greller and Drachsler's framework

include action and close the loop. "The benefits of analytics accrue only to the extent to which they inform action; that is, we do not simply want academics and students to be able to comprehend available data, but to be able to take effective learning actions based on that information (Dix and Leavesley, 2015.)" Without action, learning analytics is futile. Therefore, "tools should not only be usable, but also useful in the context of the goals we want to achieve (Dyckhoff, Lukarov, Muslim, Chatti, and Schroeder, 2013.)" To ensure that learning analytics result in action, "it is important to start with the questions, so as not to be limited by the data available. A myopic focus on the data can lead to answering questions for which we have easy answers but which do not matter (Olmos and Corrin, 2012.)"

Learning analytics must be sensitive to ethical issues around the gathering and use of student's data. Data privacy will be a major concern. "It may be threatening to some students and faculty to know that someone can 'watch' and track all that they do (Dietz-Uhler and Hurn, 2013.)" How the data gets used is also a major concern. "Data can easily be abused as supporting evidence for exercising inappropriate pressures on data subjects to change otherwise perfectly acceptable or explainable performance behaviour (Greller and Drachsler, 2012.)" It may be easier to justify gathering private data from students if they feel that the analytics on that data truly is meant to address their needs rather than just the education institution's needs. "Such a perspective has the potential to extend criteria for learning success beyond grades and persistence to include motivation, confidence, enjoyment, satisfaction and meeting career goals (Ferguson, 2012.)" It is better for an institution to use analytics to improve student success rather than maximizing profits. Even well-intentioned uses of learning analytics must be carefully applied. A student may recognize and resent efforts by an institution to nudge him or her towards a particular path to success, especially because that path "may be in conflict with his or her own preferences or study goal (Slade and Prinsloo, 2013.)"

Ethical issues are not the only challenge for learning analytics. Acceptance is a major problem. Providing learning analytics to educators is not enough. Educators must use them. "Faculty need to be involved in order for learning analytics to have its greatest impact (Dietz-Uhler and Hurn, 2013.)" Resistance may simply be a fear of change. "Although social systems such as educational institutions do evolve and change over time, they are inherently resistant to change and designed to neutralize the impact of attempts to bring about change (Macfadyen and Dawson, 2012)." Faculty are not guaranteed to act rationally. They may resist that which makes them uncomfortable, especially if they think their students will be superior to them at new technology. They may also be complacent in accepting methods which have worked in the past (Spector, 2013.) To overcome resistance, "successful institutional adoption demands comprehensive development and implementation of policies to address Learning Analytics challenges of learning design, leadership, institutional culture, data access and security, data privacy and ethical dilemmas, technology infrastructure, and a demonstrable gap in institutional Learning Analytics skills and capacity (Macfadyen, Dawson, Pardo and Gasevic, 2014.)" At the same time, learning analytics must not be sold as a silver bullet solution to all educational woes (Ferguson, 2012.) There are limits to learning analytics. "The learning process is creative, requiring the generation of new ideas, approaches, and concepts. Analytics, in contrast, is about identifying and revealing what already exists (Siemens, 2013.)"

Chapter III – THEORETICAL FRAMEWORK

The *objective ear* is an application suitable for assessing progress on a music task. An analysis of different types of skills will show how music is a kind of performance skill that is difficult to assess, and that assessing progress is a useful approach.

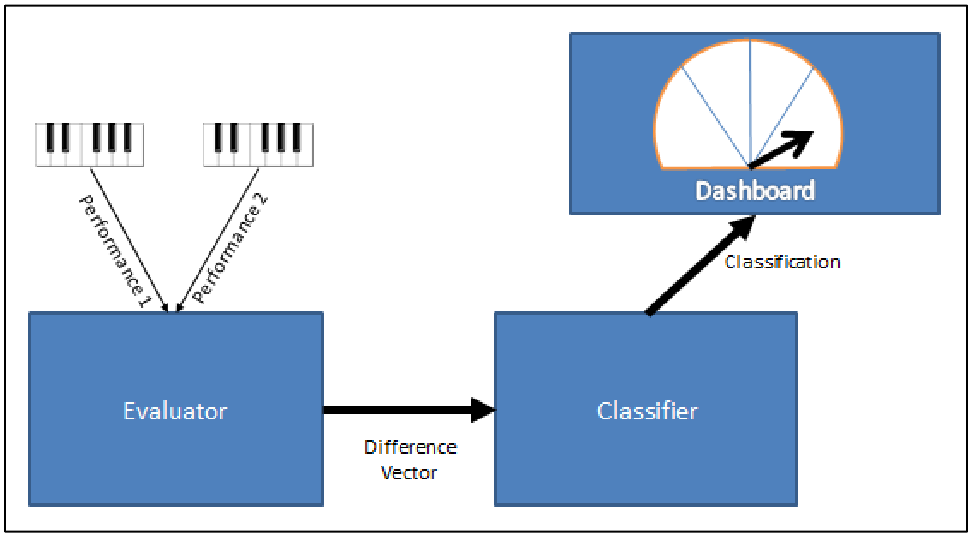

Figure 2 Major Components

The *objective ear* itself is a Java application. It is designed such that it can easily be integrated into a learning management system's dashboard. It has two main components: an Evaluator and a Classifier. The overall application acts as a machine-learning classifier, taking as input two performances of a piece of music in MIDI format and returning a classification of the progress made between the two performances. Classification works on a vector of features, where each feature is an aspect of the underlying data that has been extracted and quantified. Most of the *objective ear's* complexity is in the creation of this vector of features from the MIDI data input. This is done by the Evaluator component.

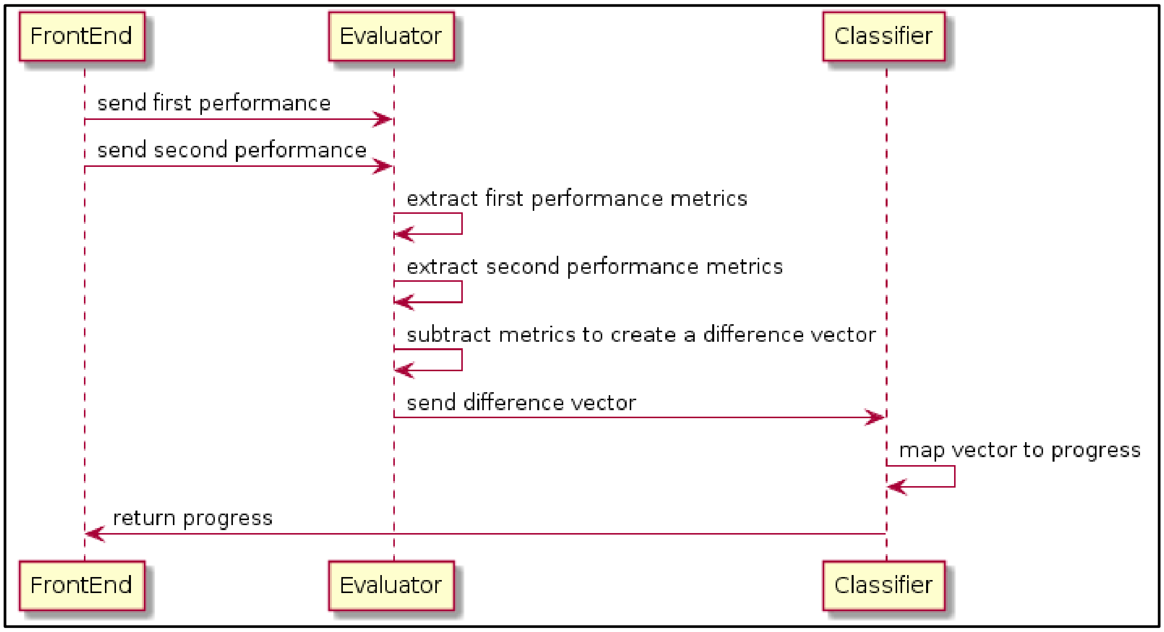

Figure 3 Overall Sequence

The Evaluator component takes the MIDI performances as input, and for each performance it performs several analyses, each analysis producing a metric represented as a real number. For each performance, these real numbers are gathered into a vector of metrics. The vector from the second performance is subtracted from the first, resulting in a vector of differences. This difference vector acts as the vector of features for the Classifier component. These vectors all have seven elements, corresponding to the seven metrics derived from each performance.

The field of machine learning provides a variety of classifiers. A classifier acts like a function in that it maps a list of features to a value. A decision tree algorithm was chosen to perform the classification. Decision trees produce output in the form of a large if statement that can be understood easily. Thus, considering teachers uncertain about a machine learning tool, having easily understood output is a major advantage.

The following pseudocode gives the overall algorithm:

```
JudgePerformances(first, second)
  firstJSON = midi2json(first)
  firstMetrics[] = evaluate(firstJSON)
  secondJSON = midi2json(second)
```

```
  secondMetrics[] = evaluate(secondJSON)
  differenceVector = subtract(secondMetrics, firstMetrics)
  return decisionTreeClassify(differenceVector)
```

The MIDI format was chosen as the input format. This limits the objective ear to instruments that are capable of producing MIDI output. For this research, output from a keyboard was preferred and should be more complicated than many other instruments because of the polyphonic nature of the keyboard. A hypothetical tool that converted a performance to MIDI output could act as a front-end to the objective ear, making it accessible to more instruments and to the human voice.

The MIDI files are transformed into a JSON format, and input it into the Evaluator component. This step was added because it makes it easier to integrate the objective ear into a system in which a server hosts the Evaluator and Classifier run on a server whereas a client processes the performances and displays the result.

The Evaluator limits itself to music from the Classical era, a period that extends from the late-18th to the early-19th centuries. This era roughly begins with Haydn’s career as he led the transition from Baroque music to Classical and ends with Beethoven as he transitioned Classical music to the Romantic era. This Classical period is ideal for a probabilistic approach. The Baroque era may have had more rigid rhythmic patterns, but its emphasis on counterpoint and sequences makes it more difficult to detect pitch errors without being extremely familiar with the specific piece being performed. The Romantic era weakened tonality greatly, preparing music for the atonal experiments that were to follow. Thus, tonality was strongest during the Classical period, and a strong tonality greatly aids a probabilistic approach to error detection because pitches align much more closely to a fixed pattern of notes. Also, the Classical period heavily favored scale passages and arpeggios within its works, meaning that a probabilistic tool trained for music from the Classical period is also well suited for most technical exercises. The objective ear is unsuitable for contemporary classical music. After Debussy, classical music branches into conflicting approaches, and many composers make explicit attempts to avoid any kind of regularity within their music, fleeing from concepts like melody and harmony. It is impossible to evaluate a performance of this kind of music without reference to the score, so an objective ear would never be able to provide a successful assessment of progress.

The interface between the Evaluator and Classifier is a difference vector. This difference vector is an ordered list of real numbers. This vector reflects the metrics extracted from the music by the Evaluator, and is used as the feature list input into the Classifier to get a classification of progress.

We have four possible classifications of progress:

- worse,
- the same,
- better,
- much better.

These categories were chosen to increase the validity of the dataset. The dataset required human listeners to listen to pairs of performances and give a judgement. Fewer categories results in easier judgements.

This classification can be returned to students' and teachers' learning management system. Typically, this will be a learning environment that features a dashboard that can report the classification of progress.

Assessment

An assessment is an evaluation performed on a student. This can be a formal evaluation, such as an exam in which the assessment is reported as a grade, or a less formal assessment in which a teacher points out a student's mistake or suggests how better perform a task. Assessment thus has a wide scope.

Assessment can be trivial. Consider a course that is knowledge-based, as opposed to skill-based, and that has many students. The assessment takes a sample of the knowledge base covered in the course, and creates multiple-choice questions that probe each student's understanding of this sample. The challenge may be in creating the appropriate exam, but grading the exam is trivial because each question has a single possible correct answer.

Assessing knowledge can be more complicated. Rather than a multiple-choice question, we may ask the student a question and expect the student to write an answer. For example, in a multiple-choice exam, we might have the following question:

Which of the following is an object-oriented language:

- *Fortran*
- *Java*
- *Lisp*

Either the student circles Java and gets the answer correct, or the student gets the question wrong. Consider instead the following question:

Give an example of an object-oriented language.

To evaluate an answer, we must have knowledge of all computer languages and know if they can be classified as object-oriented or not. An even more challenging knowledge question might be:

What are the properties of an object-oriented language?

This question demands a long response which the evaluator must read and extract relevant facts. Then the evaluator must determine a grade for the question depending on the facts extracted from the answer.

If we assess a skill rather than knowledge, then the assessment process may be more difficult. For example, consider the following assessment task:

Implement a routine to sort integers using Java.

To assess this task, the evaluator only needs to confirm that the program is written in Java and, after running the program on some test sets of data, makes sure that the output is correct. If the output is wrong, the evaluator's job becomes more difficult because the evaluator may still judge that the program has some merit and needs to determine partial grades.

Consider that instead of evaluating just the Java program written by the student, we instead closely monitor everything that the student does while creating the program and evaluate the student's actual performance of the task. How quickly does the student write the code?

How many mistakes does the student make during the implementation? How quickly and effectively does the student debug the program?

Assessing a performance is difficult! Consider the judging controversies that plague figure skating and boxing. Why is this kind of assessment so difficult? First, a performance consists of a huge number of tasks, each which must be evaluated. Secondly, a performance occurs in time which can be an important dimension in the assessment because an action that is appropriate in one time context may be inappropriate at another. For some types of performances, an absolute sense of time is critical, whereas for other types of performances, only the order that the tasks are performed matters. A pianist plays a note or a dancer takes a step – sometimes it is a correct action and sometimes it is not. Thus, to assess a performance, we must evaluate many actions by their time-related context.

How can we perform such an assessment? Consider a simplification: assess progress rather than raw skill. Why is this a simplification? A problem with assessing a performance task is that we need something to measure it against. In fact, we need a minimum standard to use as comparison. Performance tasks typically contain aspects that are measured as continuous values rather than discrete values, even if during the recording or analysis process we must discretize these continuous aspects. No musical note has ever been played at a precisely accurate rhythm because, since it is a continuous variable in time, we can increase the level of precision until we detect an error. Likewise, no musical note has ever been played at a perfectly accurate pitch. Perfect performances are impossible, and it is only the limits of the human brain itself that allows the illusion of perfection. Creating minimum standards is a challenge, especially considering the number of possible tasks that can be assessed.

If we assess progress rather than raw skill, we can use an earlier performance of a task for comparison. Thus, the students themselves provide their own baselines. Furthermore, as a day-to-day assessment, progress assessment is superior to raw skill-assessment because it allows a student to focus on improving a specific performance task. For most performance tasks, skill is gained by mastering increasingly difficult tasks. Assessing a student's progress on these tasks provides valuable information to a student and a teacher. If a student masters every task between lessons, the student is probably ready for much more difficult tasks. If a student cannot make progress on a task, it may be an indication that a student's current tasks are too challenging. With a skilled teacher, accurate progress assessment data can be used to highly tune a student's learning plan.

The Objective Ear

The two major components are the Evaluator and the Classifier.

The Evaluator Component

The Evaluator first transforms the music data into a sequence of *NoteInfo* objects. Each *NoteInfo* object contains the note's pitch, timestamp, and velocity. A note's velocity is the strength at which it is played. Typically, a higher velocity results in a louder note. The main list of notes contains all the notes sorted by timestamp. The *NoteInfo* objects are also stored in a second data structure. To create the second data structure, the notes are separated into voices. Because some metrics evaluate the voices individually, it is important to first perform this voice separation. This second data structure stores the notes for each voice separately, making it easier for the Evaluator to analyze each particular voice.

```
NoteInfo {
     Note note; // the pitch
     int duration;
     int velocity; // amplitude
     int differential; // inter-onset interval
     long timestamp; // onset time
}
```

The *NoteInfo* objects are very small, and a performance typically has a note count in the thousands. Thus, storing these performances as sequences of *NoteInfo* objects causes no memory issues.

The Evaluator is a framework with plugins, every metric acting as a plugin. This simplifies adding and removing metrics, and should make it easy to create new metrics in the future as necessary. Each plugin implements a *Metric* interface that has methods to perform the appropriate evaluation to get the metric and to verify that an evaluation for a metric can be performed. This verification function is necessary because some metrics depend upon others. For example, to do a rhythm analysis, the Evaluator must know the tempo of the performance. The various *Metrics* are registered in a *MetricConfig* object. As part of the registration, each metric is given an index. This index maps the metric to its position in a vector of metrics for a performance.

```
Interface Metric {
     boolean canRun();
     double getMetric();
     int getIndex();
}
```

For each performance, the Evaluator iterates over each *Metric* registered in the *MetricConfig*, resulting in a pair of vectors. It then subtracts the first vector from the second giving a difference vector, the input to the Classifier component. We will discuss each metric in turn, as well as foundational algorithms needed to calculate metrics. For many of these metrics, their calculation is non-trivial and requires sophisticated algorithms.

Table 1 Metrics

Metric	**Description**
Tempo	Determines the number of beats per minute.
Ornament	The rate of ornaments in a piece, weighted by complexity.
Pitch	The rate of pitch errors.
Rhythm	The rate of rhythm errors.
FlexRhythm	The rate of rhythm errors, allowing for a more expressionistic performance.
Dynamics	The rate of dynamics errors.
Error Groups	The number of errors when multiple errors in a short period of time are combined into a single larger error.

Voice Separation

Before the Evaluator calculates the metrics, the voices are separated. Several of the metric calculations require separated voices. For example, an ornament is always performed in a single voice. It is possible to have multiple ornaments performed simultaneously, but each is in its own voice. Thus, it is necessary to first separate the voices.

Because of how the separated voices are used, an original algorithm for voice separation was created. Most voice separation algorithms have requirements that insist that a voice be consistent across the entire separation. For example, in four part writing, the soprano should always be in the same voice. This was not considered necessary in the application. None of the metrics expect this kind of consistency within a voice, and are more interested in the sequence of voices within smaller sections. As well, many voice separation algorithms search for endings for each voice. That is, rather than a set of voices in which each voice can be active for the entire piece of music, a voice may only be relevant for a few notes, and then ends. In the case of the *objective ear*, as many voices as necessary are created and remain active for the entire piece of music, because the algorithms that use separated voices implicitly assume that a long delay between two notes in a voice greatly reduces the algorithm's ability to predict anything about the next note.

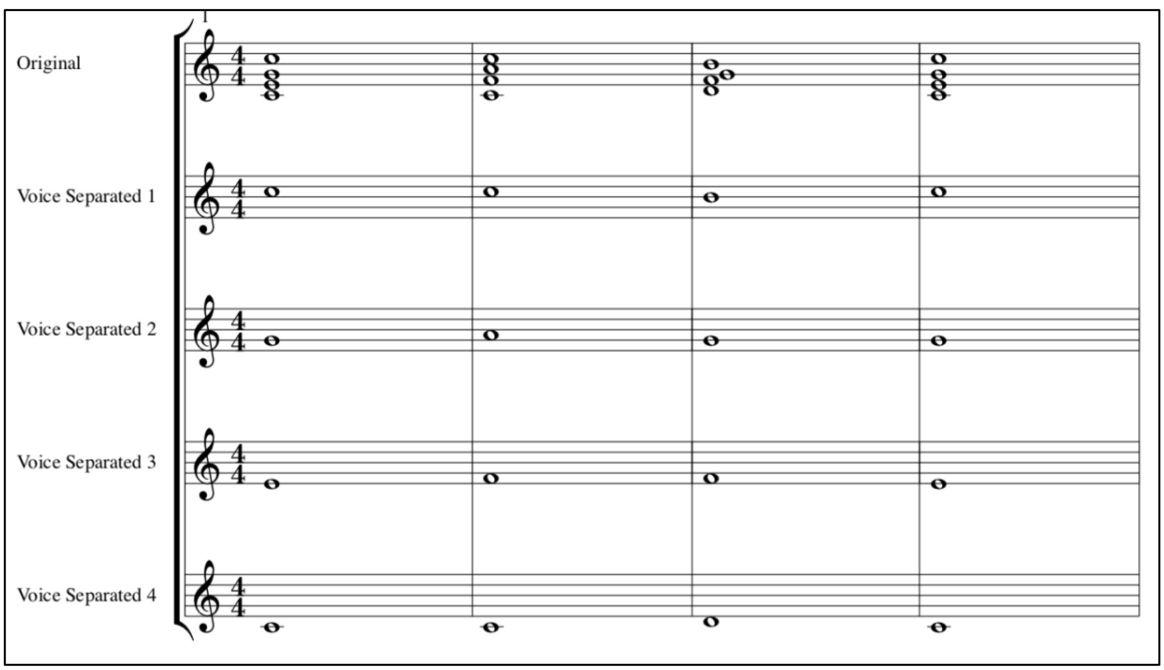

Figure 4 Voice Separation

Our simplified algorithm is a greedy search algorithm that iterates through the list of notes. For each note, a function evaluates the cost of adding the note to each existing voice, and then adds the note to the voice with the smallest cost. The cost function works by comparing a potential note, n_P, to the last note in a voice, n_L. It calculates the onset time between them:

$$t_V = n_L.onset - n_P.onset$$

and the difference in pitch:

$$f_V = n_L.pitch - n_P.pitch$$

It then creates a weighted sum:

$$cost_v = t_V \times w_{tV} + f_V \times w_{fV}$$

The weights, w_{tV} and w_{fV}, are parameters that can be tuned to get the best result.

```
while notes not empty
     currentNotes = extractAllCurrentNotes()
     costList = calculateCosts(currentNotes, voices)
     while costList not empty
          {cost, note, voice} = costList.smallest ()
          voice.add(note)
          currentNotes.remove(note)
          costList.removeNote(note)
          costList.removeVoice(voice)
     if (currentNotes not empty)
          for each note in currentNotes
          newVoice = new voices.createNewVoice()
          newVoice.add(note)
```

The algorithm is complicated by the need to handle simultaneous notes. At each step in the algorithm, the algorithm selects all the notes with a simultaneous onset time, creating a set of simultaneous notes. For each note in the set of simultaneous notes, the cost function is calculated against every open voice. This cost function is sorted by size, and the algorithm selects the smallest-costing note that has not been assigned, and assign it to the corresponding voice. If the algorithm run out of voices but there are still notes to assign, this implies that the algorithm needs to create new voices. Each remaining note in the set is assigned to a new voice. The number of cost functions evaluated at each step is of size $|set_s||set_v|$ where $|set_s|$ is the size of the set of simultaneous notes, and $|set_v|$ is the set of open voices. This set will never grow too large. A performer is always limited in the number of simultaneous notes that he or she can play. Keyboard instruments have the largest possible range of 10 notes, although in practice this number is much smaller and rarely grows larger than 5. (Although Chopin has written music that requires 11 note chords, chords of more than 10 notes in keyboard music are extremely rare.) Thus, the algorithm has an upper bound for both $|set_s|$ and $|set_v|$, meaning that $|set_s||set_v|$ will always be less than 100, and the voice separation algorithm runs efficiently.

At the end of the iteration through the notes, the algorithm will have a set of voices, each voice being a list of notes within that voice. This data structure is stored in a common data store where it can be accessed by all the metrics.

Tempo

The tempo of a piece of music is not only a metric, but it is a key parameter for many other metrics. Think of the tempo as how quickly a piece of music is performed. Typically, in music literature, it is represented as the number of beats per minute. The beat is mapped to a note value, such as a quarter note, or the eighth note of a triplet. The Evaluator defines the tempo as the duration of the most common inter-onset interval between notes. This is the time between the start of one note and the start of the next note. This approach would be impractical for a performer, but it greatly simplifies the analysis of a performance. For example, consider the second etude from Chopin's *Trois Nouvelles Etudes*. For most of its duration, this piece features cross-rhythms with the left hand playing pairs of eighth notes to the right hands eighth-note triplets. The most common inter-onset interval is a sixth of a quarter note, which would be a very bizarre notation even though it works very effectively in the rhythmic analysis of the piece of music.

A clustering approach to finding the tempo (Dixon, 2001) works effectively for the purposes of finding the tempo. Dixon admits that this method is weak at finding the correct level of the pulse, but this is unnecessary information for the Evaluator's metrics. The clustering approach is effective because the level stays stable across performances. That is, even though the most common inter-onset interval will be different between two performances, this most-common interval will be for the same pairs of notes between the two performances except in cases of extreme error.

Clustering is done by implementing a simple agglomerative hierarchical clustering algorithm. The only parameter for the algorithm is a width parameter that is compared to inter-onset intervals. The first step in clustering is to go through the list of notes, and create a corresponding list of inter-onset intervals. This is simply a list of integers. As a first pass, each value in the list is compared to the center of each existing cluster. The center is calculated as the mean value of each value in the cluster. If the $|mean_{cluster} – interval| <$ width, then the algorithm adds the interval to the cluster. If no appropriate cluster is found, the algorithm creates a new one containing only this interval. If the inter-onset interval is zero or less than a threshold value which makes it effectively zero, then the algorithm ignores the value. Otherwise, the largest cluster would usually be the zero cluster, which provides no value. After the first pass, the algorithm compares each cluster to each other cluster to see if the absolute value of the difference of their means is less than the width: $|mean_{clusterA} - mean_{clusterB}| <$ width. If so, the two clusters merge. The algorithm repeatedly go through the list of clusters performing merges until it traverses through the entire list of clusters without finding a pair to merge. This merging is necessary because every time the algorithm adds a value to a cluster, its center may change, meaning that a pair of clusters that were not mergeable in one pass may become mergeable in the next. This merging cannot continue indefinitely, and the worst-case scenario for the number of iterations is n^2 where n is the number of inter-onset intervals, although in practice, the number of merges will be much less than this.

At the end of the merging, the algorithm selects the cluster with the most intervals, and takes its mean as the tempo. That is, $tempo = mean_{clusterMax}$ where clusterMax is the cluster with the most elements.

Ornamentation

An ornament is a flourish, often optional, that a performer adds to a piece of music during a performance. Sometimes ornaments are notated on the score, but performers will often add ornaments to suit their own taste. A performer’s use of ornaments acts as a useful metric because as skill on a music task increases, a performer may add more ornaments or will play them faster.

To use ornaments as a metric, we must be able to detect them, and in doing so, we quickly discover an existential quandary for ornaments. With a score, we can easily spot ornaments, but just by listening to a performance, we cannot necessary state that an ornament is an ornament. An ornament is typically notated as a symbol over or under a note indicating the kind of ornament to perform centered on that note. A composer can also write out an ornament fully so that it appears integrated into the score, even though it still sounds like an ornament. The solution to this is to not care! If a set of notes sounds like an ornament, it gets treated as an ornament.

Ornaments pose a problem for those who are trying to implement a score-matching program because the ornaments need to be removed before making the comparison to the score. Thus, research into score matching has determined rules that can be used to detect ornaments (Gingras & McAdams, 2011). There are two properties of most ornaments that the algorithm for ornament detection uses. First, most ornaments are played very quickly, especially within the context of their surrounding notes. Second, most notes within an ornament limit themselves to a narrow range of pitches. For example, a common ornament is a trill which is performed by alternating very quickly between a pair of notes only a semitone apart.

The ornament detection algorithm iterates through separated voices. The reason is that ornaments are always performed within a single voice. For example, while a pianist is

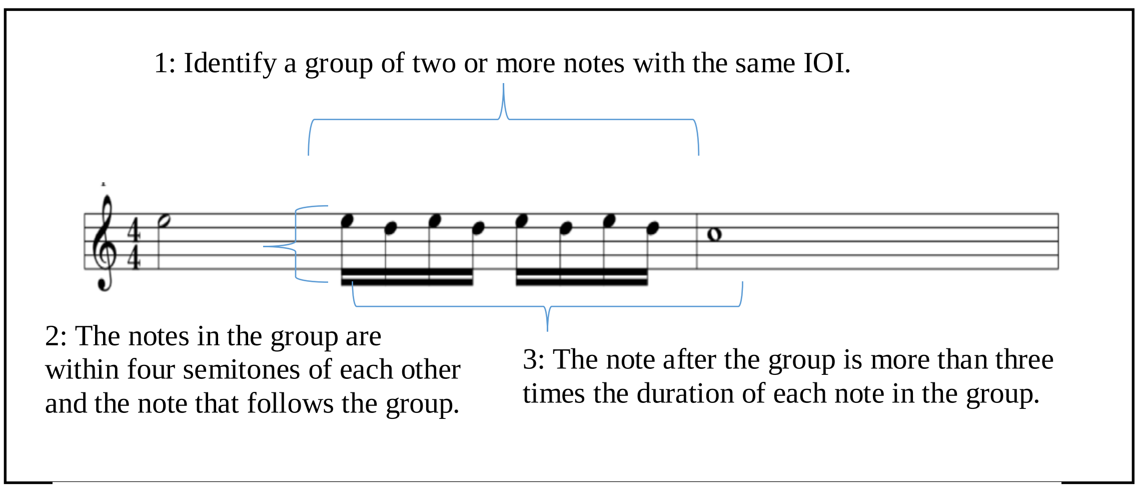

Figure 5 Identifying an ornament

performing a trill with his or her right hand, the left hand can play several notes in the bass. Without voice separation, the algorithm would unnecessarily subdivide the trill by notes from the other hand. An extreme example is the third movement of Beethoven's *Waldstein Sonata*, Op. 53. The performer is expected to play an extended trill with the fourth and fifth fingers of the right hand while the rest of the fingers in each hand perform multiple other voice lines. Without voice separation, the algorithm would not be able to effectively identify and analyze the trill.

The algorithm iterates through each voice. It first searches for ornaments based purely by rhythm. It looks for a sequence of notes whose inter-onset interval is less than a threshold specified as a parameter that is defined relative to the tempo. When it finds such a group, it extracts the notes, and then evaluate them further to determine if they are an ornament by determining if the range of the pitches is within four semitones.

The algorithm returns as a metric the rate of ornaments performed: ${}^{n_{ornaments}}/_{n_{notes}}$.

Pitch

The field of automated music analysis has heavily researched pitch analysis, but almost always with the purpose of understanding the underlying harmonic structure of a piece of music. Understanding the harmonic structure is the most common task of a music theorist when presented with a piece of music. Less research effort has been put into detecting pitch errors. Many researchers have used an nGram approach to various aspects of music when creating predictive models. This approach has been successful because much of music is structured in such a way that it can be modelled by a hidden Markov model: at every point of time in a piece of music, the probability of what notes will be played next can be determined based on which notes have been played before. Thus, an nGram approach was chosen to predict pitch errors.

An nGram is a sequence of *n* data points which captures *n* events that are performed in sequence. By examining a corpus of data, the probability of each nGram can be determined by counting the number of times it occurs within the corpus. With a set of nGrams probabilities, a threshold value can be chosen such that every nGram whose probability is less than this threshold is deemed unlikely and thus an error.

A key prerequisite to this nGram analysis is creating an appropriate alphabet for the nGrams. The notes themselves were not used as raw data points in the nGram for two reasons. First, the variety in each raw note is huge and would result in a massive alphabet; and since the size of the set of all possible nGrams is the size of the alphabet to the power of *n*, the resulting data would be too large to efficiently analyze. Second, the raw note is not normalized, so the timing information is meaningless. For example, consider a note of a particular duration. In one performance, that note may be considered a long note, whereas in another it may be considered a short note, all depending on the context of the other notes.

To create the alphabet, the pitch algorithm switches the nGram from representing a single note to instead representing the transition between two notes. That is, rather than representing something like a middle C quarter note, the algorithm instead represents a transition up a fifth (such as from C to G) with the notes played in quick succession. This transition is normalized on two dimensions. First, the pitch is reduced to an interval within an octave. This is done by performing a modulus operation on the pitch. Second, the duration of the note is categorized as being zero, short, medium or long. A *start* element in the alphabet accounts for the first note where there is no interval or IOI. This classification is done relative to the tempo of the piece of music, and the thresholds for various categories are tuneable parameters that act as weights on the tempo.

```
start,            eleven_zero,      eleven_short,
zero_zero,        zero_short,       zero_medium,
one_zero,         one_short,        one_medium,
two_zero,         two_short,        two_medium,
three_zero,       three_short,      three_medium,
four_zero,        four_short,       four_medium,
five_zero,        five_short,       five_medium,
six_zero,         six_short,        six_medium,
seven_zero,       seven_short,      seven_medium,
eight_zero,       eight_short,      eight_medium,
nine_zero,        nine_short,       nine_medium,
ten_zero,         ten_short,        ten_medium,
```

eleven_medium,
zero_long,
one_long,
two_long,
three_long,
four_long,
five_long,
six_long,
seven_long,
eight_long,
nine_long,
ten_long,
eleven_long;

When assembling a list of nGrams using this alphabet, special consideration was given for zero-duration members of the alphabet. Whenever one or more zero duration transitions were discovered in a sequence, a combinatorial expansion was allowed to occur on these transitions to account for the fact that any sequence of transitions within a set of zero-duration transitions is valid. For instance, if a performer plays a C, E, and G at the same time, the algorithm considers CEG, CGE, ECG, EGC, GCE, and GEC to all be valid orderings of the notes when converting them to nGrams.

The *n* in nGrams can be any integer greater than zero. The size of set of all possible nGrams is the size of the alphabet to the power of *n*. Thus, the set size grows exponentially as *n* increases. As *n* decreases, history is lost as the algorithm calculates probabilities based on fewer previous events. As *n* increases, the probability of a nGram is reduced and the algorithm needs significantly larger training sets. An attempt at using 4-Grams failed because the set of nGrams consumed a large amount of memory. 3-Grams were used as they were contained the most history yet were restrictive enough to prevent performance issues or nGrams of too low a probability.

The alphabet was trained using a large corpus of MIDI data files of music from the Classical era. The result of this training was a large list of tuples, each tuple being a 3-Gram, stored along with its rate of occurrence in the corpus. This list was analyzed to determine a reasonable threshold. Every 3-Gram whose rate was greater than this threshold was considered a probable 3-Gram. Thus, the final output of the training was a list of probable 3-Grams.

To generate the pitch metric, the Evaluator converts a performance's MIDI data into a set of 3-Grams. If a 3-Gram is not in the list of probable 3-Grams, then the final note from the last transition in the 3-Gram set is considered an error.

The algorithm returns as the metric the rate of pitch errors. This is the number of error notes divided by the number of notes in the performance: ${n_{pitchErrors}}/{n_{notes}}$.

Rhythm

The Evaluator provides two different rhythm metrics. The first is a simple, naïve metric whereas the second metric is more representative of performance practice. The naïve metric always detects more errors, but many of them are false positives.

To understand how the rhythm metrics work, first understand some basic properties of rhythm in the Classical era. The rhythm is directly related to the tempo. If the tempo of a piece is 60 quarter notes per minute, then a common inter-onset interval should be one second between two quarter notes. This beat is subdivided or combined in a very regular way. Virtually every note is related to the beat of the tempo by a factor of powers of two or three. For example, a regular eighth note is half the duration of a quarter note, and a triplet eighth note is a third the duration. If the tempo of a piece of music is known, the expected inter-onset interval of every note can be determined by applying these factors. The set

$$F_T: \left\{x: 2^i 3^j, -5 \leq i \leq 5, -5 \leq j \leq 5\right\}$$

contains a list of valid tempo factors. Using these factors, the set

$$R: \{tempo * x: x \in F\}$$

contains a set of valid values for a performance. That is, the rhythm algorithm multiplies each element of F by the tempo to create a set of factors relative to the tempo. Every inter-onset interval should be close to one of these values.

For the naïve metric, the Evaluator iterates over the notes in a performance, and finds the element of *R* closest to the note's inter-onset interval. It then determines how close the note is to the closest interval. It does not use the raw difference but rather determines by what percent it differs from the interval. A tunable parameter is the percent that an interval can differ from the ideal without being considered an error. Any note whose inter-onset interval is not tolerably close to an ideal is counted as an error, and the metric is returned as the rate of rhythm errors, that is, the number of errors detected divided by the total number of notes in the performance: ${n_{rhythmErrors}}/{n_{notes}}$.

It should be noted that occasionally, a composer in the Classical era will subdivide a beat by five, seven, eleven, thirteen, or some other number that is not divisible by two or three. This creates a dramatic effect, but is used very sparingly and usually only in very advanced pieces of music. Thus, the number of notes that will be erroneously flagged as errors is very low and will have very little effect on the overall error rate.

What the naïve approach fails to account for is that performers will occasionally slow the performance and this occurs in a regular way. A piece of music from the Classical era is highly structured. Short music phrases, the musical equivalent of a sentence, can usually be identified, and groups of phrases will make a section that is often repeated, sometimes with a variation. Different sections are often contrasted, and commonly, a piece of music can be subdivided into two major sections, such as in sonata form between the exposition and the development. The performer will often slow the tempo relative to the importance of the subdivision. Thus, during a major subdivision, the tempo may slow down much more noticeably than at the end of a smaller section.

The second metric accounts for this flexible rhythm. It identifies a hierarchy of groupings, and then allows a note's inter-onset interval to deviate more significantly from the ideal depending on how close the note is to the end of a grouping, and the significance of that grouping.

Grouping rules are an important component of Lerdahl and Jackendoff's *A Generative Theory of Tonal Music*. These rules have been implemented as a computer algorithm (Hamanaka, Hirata, & Tojo, 2006). This algorithm was implemented to determine various rules that can be used for grouping. Four of the grouping rules are implemented by analyzing sequences of notes in voices:

- GPR2a recognizes slurs and rests. The algorithm analyzes four notes in a voice in succession, n_1, n_2, n_3, and n_4; and examines the offset-to-onset interval (OOI) between the notes, r_1, r_2, and r_3. If $r_1<r_2$ and $r_3<r_2$, then consider this rule activated.

- GPR2b recognizes attack points. The algorithm analyzes four notes in sequence, n_1, n_2, n_3, and n_4; and examines the inter-offset interval (IOI) between the notes, i_1, i_2, and i_3. If $i_1 < i_2$ and $i_3 < i_2$, then the rule is activated.
- GPR3a recognizes pitch leaps. Analyzing four notes in sequence, n_1, n_2, n_3, and n_4; the algorithm examines the absolute difference in pitch intervals, $|p_1|$, $|p_2|$, and $|p_3|$. If $|p_1|<|p_2|$ and $|p_3|<|p_2|$ then the rule is activated.
- GPR4 combines parameters from above, along with dynamics differences between four notes: d_1, d_2, and d_3. Let $R=r_2/(r_1+r_2+r_3)$, $I=i_2/(i_1+i_2+i_3)$, $P=|p_2|/(|p_1|+|p_2|+|p_3|)$, and $D=d_2/(d_1+d_2+d_3)$. GPR4 is activated if max(R,I,P,D) is greater than a threshold, a parameter that can be set.

Calculate a threshold depending on how many rules are triggers. If *n* rules are triggered, the value of the trigger *Tr* is:

$$Tr = \begin{cases} 0: n = 0 \\ 0.5: n = 1 \\ 0.75: n = 2 \\ 0.88: n = 3 \end{cases}.$$

The algorithm also determines the time that the piece ends, and the time that the note is played, and calculate *X*, the point the note is played, normalized to the range [0,1]. For example, a note played two-thirds of the way through would have the value 0.66. Every note has a split strength calculated as *Tr*X*.

The algorithm sets a limit to the number of levels for which to consider splits. This is a configurable parameter. It starts at the highest split level, the most major group, and works down recursively to the least significant groups. It searches through the list of notes for the note with the strongest split strength, and splits the notes there. The algorithm then splits the next level down for the group of notes that precede the split and the group of notes that follows the split.

Finally, the algorithm applies a time multiplier to every note. Every note has a default time multiplier of 1. It iterates through every note, and if that note is within five notes of a split, it calculates the time multiplier. The algorithm calculates how many notes it is from the split, a value from 0 to 4, and use this to index the array [1.0, 0.8, 0.6, 0.4, 0.2] to get a value A. Let the split level be S. The algorithm sets the time multiplier to 1.0 + A/S. Thus, the closer to the split, the larger the time multiplier. Likewise, the more significant the split, the larger the time multiplier.

Once every note has a time multiplier, the algorithm proceeds as per the naïve rhythm metric, except that it now divides the error by the note's time multiplier before checking if the error exceeds the threshold. This means that as the algorithm approaches the end of a group, the calculated error will be reduced, making it less likely to exceed the threshold. As with the naïve metric, the algorithm returns an error rate – the number of errors detected divided by the total number of notes: $n_{flexRhythmErrors}/n_{notes}$.

Dynamics

Dynamics refers to how loudly or softly a performer plays a note. We usually think of dynamics in the large, great booming pieces of music that shake the windows; or quiet, intimate music that insists we lean in to hear it. Dynamics plays a far more significant role. During the performance of a piece of music, a performer must shape a phrase, and much of this shaping is done with dynamics. Without shaping, the performance sounds amateurish, wooden, and robotic. Musicians shape phrases is predictable ways, making it possible to assess dynamics and to identify dynamics errors.

Within the data, dynamics can be assessed using the velocity data for a note in the MIDI file format. The velocity refers to how quickly the note is attacked. This corresponds to several physical movements with music instruments. For a keyboard, it refers to how hard the key is pressed. For a violin, it refers to the strength of the bowing. For wind and brass instruments, it refers to the force of air blown into the instrument. In all cases, a stronger velocity means louder dynamics.

Gerhard Widmer's paper, "Machine Learning: A Few Simple, Robust Local Expression Principles" (Widmer, 2002) provides many valuable rules for identifying dynamics errors. Widmer uses learning algorithms on a large corpus of performance data to create these rules, and discovers that, although dynamics is considered part of interpretation, the artistry of performing, dynamics show very regular patterns than can be detected by a learning algorithm. Rather than recreating Widmer's work, the dynamics algorithm directly use the rules that he discovers.

It implements three of his rule:

- DL1: When the interval between two notes is greater than a perfect fourth upwards, the second note is played more loudly than the first note. This can be calculated by comparing the two notes for pitch and velocity.
- DL2: If an interval between two notes is greater than a third, and the next note is any interval down rather than up, the middle note is louder. This can be calculated by comparing three notes for pitch and velocity.
- DS3: If a note's duration is less than a fifth of the duration of the previous note, then it is played more quietly. This can be done by comparing two notes for duration and velocity.

Rather than looking at the raw list of notes, the algorithm uses the voices from voice splitting since these rules apply to notes within a voice. It applies these rules for each note in a voice, and if a note fails a rule, it marks that note as an error. If a note fails multiple rules, the algorithm still only flags it as being a single error. To generate the metric, the algorithm returns the error rate, the number of errors divided by the number of notes: ${n_{dynamicsErrors}}/{n_{notes}}$.

Error Groups

The error groups metric applies an intuitive observation about performance errors. If a performer plays a note with the wrong pitch, rhythm, and dynamics, a listener does not hear it as three separate errors, but instead hears an single error. The evaluator, as it processes metrics, flags each note with the errors attributed to that note. Thus, the errors groups algorithm analyzes all the errors in a piece of music, and identifies how many errors the listener is likely to hear, rather than the absolute number of errors.

The algorithm first extracts from the list of notes a sublist of erroneous notes. For each note, it extracts the note's onset time into a new list. Thus, for the analysis, the algorithm has a list of integers representing the onset times of erroneous notes.

The algorithm then clusters these notes. The clustering algorithm is the same as for the tempo, except that it has a much wider threshold value. This threshold value is a configurable parameter. When the algorithm has clustered the notes, within every cluster is a set of notes that will all contribute to what the listener hears as a single error. Thus, the number of clusters can be equated to the number of actual errors heard. The algorithm normalizes this by the number of notes in the piece to get an error rate: ${}^{n_{errorGroups}}/_{n_{notes}}$.

The Classifier Component

A machine-learning classifier is an algorithm that acts as a function in that it maps a set of input values, in this case called features, to an output value. What makes it special is that internally, rather than explicitly giving rules for the mapping, the classifier learns from a dataset which inputs to map to which outputs. This has the drawback that the classifier may not always give the correct response, but by learning from a dataset, a classifier can discover relationships between features that might not be obvious to the human coding the function, and can provide a mapping that a human might struggle to comprehend, let alone implement.

Several different types of classifiers exist, each with different properties. For the objective ear, a decision tree was chosen. A decision tree is not as powerful as some other classifiers, but it has the property that once trained, it can display the rules it uses to make a classification in a human-readable form. Many classifiers, such as support vector machines or neural networks, are not able to do so at all or without a great deal of effort. Being transparent in what rules are used to make a decision may provide skeptical music teachers more trust in the tool.

A decision tree is a series of nested if-then-else statements in which every if statement compares one of the features to a value. Thus, a decision tree is extremely efficient at runtime with both time and memory. Although executing a classification with a decision tree is simple, training it is far more complicated.

For training, a decision tree needs a training set. This training set contains features, matched with classifications. The high-level algorithm for performing the training is simple. It chooses a split point for a feature. This is just a value that a feature can take. It then divides the training set into those elements whose value for this feature is less than the split point, and those whose value is greater. It then recursively subdivides each subset, and continue until all the classifications for each subset are the same. Consider a trivial example. The input consists of a single feature, a person's age. The classification is a boolean value that flags if the person is old enough to drive. The algorithm choses a split point of 16. All elements with an age less than 16 are placed in one set, and all elements with an age of 16 or over are in the other set. This results in two sets: one with all false values and the other with all true values. This, of course, is a trivial example that would not require training a decision tree, but with more classifications and more features, we can quickly get into a situation where we need to train a classifier to understand how to best create decision rules.

The algorithm applies information theory to find a split point at each level. In particular, it uses information gain to find which splitting point of which feature gives us the most information. In other words, which splitting point most effectively divides the set to isolate classifications in one subset or another. Information gain is the traditional method used in the field of machine learning to find a split point.

To calculate information gain, the algorithm examines the data, D, within the current partition. It lets m = 4, the number of categories for classification, and p_i be the probability that a data element in D has the classification i where i is one of the classifications. It calculates Info(D), the information it expects to classify D, as:

$$Info(D) = -\sum_{i=1}^{m} p_i log_2(p_i).$$

It then uses Info(D) to evaluate every possible split within D to determine which split gives us the most information. For every metric, the algorithm sorts the values, and then choose every midpoint between adjacent values as a splitting point. This splits the data into two subsets, D_L and D_R. It calculates the information $Info_A(D)$ at the split point as:

$$Info_A(D) = \frac{|D_L|}{|D|} \times Info(D_L) + \frac{|D_R|}{|D|} \times Info(D_R).$$

Then, to calculate the information gain at a split point, the algorithm determines:

$$Gain(A) = Info(D) - Info_A(D).$$

It compares the gains for every possible splitting point of every attribute, and determines the best split point (Han, 2012.) This is a computationally expensive operation because of the number of possible split points to evaluate, but this is only done during the training phase, so the excessive processing time is acceptable.

As will be discussed below, a dataset was prepared which paired pairs of performances with a judgement of progress from human listeners. The Evaluator was run on all the performances, converting them into a set of difference vectors paired with a classification of progress. These difference vectors were used as the feature data for training. The output from the training was a tree.dat file containing information about the split points and the corresponding classification for leaves.

Table 2 Fields of tree.dat records

Field	**Description**
Id	Identifies the record. Used with the Parent field to determine the hierarchy of the tree.
Parent	The parent of this record in the tree's hierarchy.
Field	Which metric to split on.
Split Point	What value to split on.
Side	If this is the left or right subnode of the parent. Since the order matters, and the tree is built from the bottom up (i.e. nodes have references to parents rather than children), this is necessary.
Classification	For leaf nodes, what is the classification returned if this node is reached.

During execution, the tree.dat file is loaded and converted into the corresponding decision tree rules. After the Evaluator prepares a difference vector for a pair of performances, it passes it to the classifier which maps it to a classification.

Summary of parameters

The objective ear has several tunable parameters that have been discussed in the descriptions of the various metrics. The following table summarizes these parameters and their settings:

Table 3 Tunable Parameters

Algorithm	Parameter	Value	Description
Voice Separation	Duration Weight	0.0	How much emphasis is the duration given in the cost function when deciding which voice to add a note to. The duration refers to the rest period, the period of time between when one note ends and the next note begins. In practice, the rest period was not useful in calculating the cost, and so it was set to zero.
Voice Separation	Pitch Weight	0.001	How much emphasis is the pitch given in the cost function when deciding which voice to add a note to. The pitch difference is calculated as the absolute value between two pitches, after mapping the pitches to the natural numbers. This weight is multiplied to the pitch difference when calculating the cost function used by the voice separation algorithm's greedy search.
Tempo	Cluster Width	5.0	How much can the IOI of notes differ from each other yet remain in the same cluster. If this is too wide, notes with different note values might be considered as the same note. If this is too narrow, notes as the same note value may be considered different note values. A value of 5.0 corresponds to 145 milliseconds.
Ornaments	Tempo Threshold	4.0	How quickly must notes be performed to be considered part of an ornament. The tempo is divided by this factor. Thus, the threshold is a quarter of a beat.
Ornaments	Pitch Range	5.0	For a group of notes, how narrow must their pitch range be to be considered an ornament. This corresponds to 5 semitones. In other words, all pitches must be within a perfect fourth interval.
Pitch	Short Cut-off	3.5	What is the IOI value, relative to the tempo, used to differentiate between short and medium duration notes. The tempo is divided by this value. Thus, any note whose IOI is shorter than tempo/3.5 is considered a short note.

Pitch	Long Cut-off	3.5	What is the IOI value, relative to the tempo, used to differentiate between medium and long duration notes. The tempo is multiplied by this value. Thus, any note whose IOI is longer than tempo * 3.5 is considered a long node.
Pitch	Probable Cut-off	5.0	The probability value used to differentiate between probable and improbable pitches. The value of 5 means that, when analyzing the nGrams from the repository, any nGram that appears more than 5 times in the repository is considered probable. This is an absolute value, which means that it needs to be re-tuned if the size of the repository is changed.
Rhythm	Tolerance	0.05	How close must a note's IOI be from an expected IOI before determining that the note's rhythm is in error. This value corresponds to a 5% difference. If the IOI deviates from its closest expected value by more than 5%, it is considered an error.
Flex Rhythm	Number of Levels	5	The number of subdivisions performed when grouping. Each subdivision is binary. Thus, the output of the grouping algorithm will be a binary tree of depth 5 with the notes distributed across all of the nodes at each level.
Error Group	Cluster Width	100	How close in time can errors occur for them to be considered part of the same error group. This setting of 100 works out to a bit under three seconds. In other words, if two errors occur within three seconds of each other, they are grouped together.

Chapter IV – RESEARCH DESIGN

The previous section describes in detail the *objective ear*, but it provides no reason to expect that the *objective ear* works. This research design section uses research methods to evaluate the performance of the *objective ear* and determine its effectiveness. Because the *objective ear* is, in general, a machine learning classifier, techniques standard to the field of machine learning are used to evaluate the *objective ear*.

Purpose

The purpose of the research methodology is to determine if the *objective ear* can accurately determine if a student is making progress on a music task. It focusses the evaluation on the Classifier component rather than the Evaluator. This is sufficient because the Evaluator is implicitly evaluated when the Classifier is evaluated. The Evaluator is responsible for turning a pair of performances into the set of features that is passed as input into the Classifier. If the Evaluator is ineffective at extracting useful features from the performances, the Classifier will fail because it cannot find meaningful patterns in the data to use to map the features to the classifications.

Applying this research methodology determines if the *objective ear* is accurate. If the tool is insufficiently accurate, its use as an application to aid in music education will be limited.

Research Question

A research methodology determines how we answer a research question. This research proposes a tool for evaluating musical progress, and the research question asks if this tool works:

> *Research Question: Does the tool accurately classify the progress made between two performances of a piece of music.*

This research question presents a clear hypothesis that can be tested once the terms are clearly defined.

The tool is *accurate* if the classifications suggested by the objective ear match the classifications assigned by human listeners. The field of machine learning gives a clear, mathematical definition of accuracy that can be used.

The tool *classifies the progress* if it assigns a pair of performances a classification from the set of classifications:

- *Worse,*
- *The same,*
- *Better, and*
- *Much better.*

The two performances must be of the same piece of music. These performances must be expressed in the MIDI data format. As well, these performances must be of music from the Classical era.

These definitions reveal the assumptions behind the research question. These assumptions have been integrated into the *objective ear* and into the data set used to train and test the tool.

Research Methods

The research question is answered by using traditional machine learning approaches to evaluating a classifier. The data set is partitioned into a training set and a test set of data. After the tool is trained using the training set of data the test set is used to evaluate it.

Testing starts with an Evaluator and a Classifier that are ready for training. One-third of the data set is reserved for validation, and the remaining two-thirds for training. This partition is done completely randomly. Training the Classifier produces a model to evaluate.

The trained Classifier is used to classify every record in the test set. Each classification maps the record to one of the four progress indicators: degradation, stagnation, minor improvement, or major improvement. Compare this classification to the record’s classification by human experts, a 4x4 confusion matrix can be created:

Table 4 Confusion Matrix

	Predicted Degradation	**Predicated Stagnation**	**Predicated Minor Improvement**	**Predicted Major Improvement**
Actual Degradation	True Degradation	False Stagnation	False Minor Improvement	False Major Improvement
Actual Stagnation	False Degradation	True Stagnation	False Minor Improvement	False Major Improvement
Actual Minor Improvement	False Degradation	False Stagnation	True Minor Improvement	False Major Improvement
Actual Major Improvement	False Degradation	False Stagnation	False Minor Improvement	True Major Improvement

Thus, we can get the following measures:

- True Degradations (TD): The number of degradations accurately predicted.
- False Degradations (FD): The number of records incorrectly predicted to be degradations.
- Actual Degradations (AD): The total number of actual degradations.
- True Stagnations (TS): The number of stagnations accurately predicted.
- False Stagnations (FS): The number of records incorrectly predicted to be stagnations.
- Actual Stagnations (AS): The total number of actual stagnations.
- True Minor Improvements (TmI): The number of records accurately predicted as minor improvement.
- False Minor Improvements (FmI): The number of records incorrectly predicted to be minor improvements.
- Actual Minor Improvements (AmI): The total number of actual minor improvements.
- True Major Improvements (TMI): The number of records accurately predicted to be major improvements.

- False Major Improvements (FMI): The number of records incorrectly predicted to be major improvements.
- Actual Major Improvements (AMI): The total number of actual major improvements.

Let n be the total number of records. Thus,

$$n = TP + TS + TmI + TMI + FP + FS + FmI + FMI \qquad \text{(x)}$$

$$n = AP + AS + AmI + AMI. \qquad \text{(x)}$$

Using these measures, several further measures can be calculated that can be used to evaluate the model. First of all, the accuracy gives us the percentage of records that were accurately classified:

$$accuracy = \frac{TP+TS+TmI+TMI}{n}$$

The error rate, the percentage of records that were misclassified, can be calculated as:

$$errorRate = 1 - accuracy = \frac{FP+FS+FmI+FMI}{n}$$

For each of the four categories, the recall and precision can be calculated. The recall is the percentage of records for a classification that are correctly identified. For a classification x, the recall is:

$$recall(x) = \frac{Tx}{Ax} \qquad \text{(x)}$$

The precision is the percentage of records classified in a particular category that were correctly classified:

$$precision(x) = \frac{Tx}{Tx+Fx} \qquad \text{(x)}$$

For the test on the validation set, the accuracy of the classier can be estimated. First calculate the standard error:

$$SE = \sqrt{\frac{accuracy(1-accuracy)}{n}} \qquad \text{(x)}$$

For a 95% confidence interval, the z-value of 1.96. Thus, the confidence interval is:

$$[accuracy - 1.96SE, accuracy + 1.96SE] \qquad \text{(x)}$$

If the lower bound of this confidence interval is above 0.5, then there is a 95% certainty that the classifier is doing better than randomly assigning records to classes.

Data

The dataset is critical to this research. The dataset is required for both training the classifier, and testing the overall system. No appropriate dataset existed so it had to be created. Preparing the dataset is a major part of this research.

The items of the dataset must be pairs of performances in MIDI format and a judgement of the progress between those performances. In other words, the dataset item is a record of the form:

```
record {
  MIDI: performanceOne
  MIDI: performanceTwo
  ProgressEnum: progress
}
```

Where the progress enumeration is of the form:

```
ProgressEnum {
  worse,
  same,
  better,
  muchBetter
}
```

This dataset was created in multiple steps. Performances were recorded. Then a data gathering tool was created. Next, human participants used the tool to record judgements or progress. The data from the participants was pooled and cleaned and the training and test sets created.

Recording the performances itself took several steps. First, the music was selected. The music chosen was all piano music of various difficultly levels from the Classical era. From the chosen music, excerpts to perform were identified. These excerpts were selected with the hope they would take between twenty and thirty seconds to perform. This turned out to be true of most of the excerpts. Each piece of music was played twice. During the first performance, the tempo was semi-randomly chosen: for some pieces, they were immediately performed at tempo whereas for other pieces they were initially played more slowly and carefully. Next a ten-sided die was rolled to determine how many times the piece could be practiced before the second recording. This provided variety in the amount of progress made. Then the second recording was made. In total, 227 pairs of performances were recorded.

A simple Java application was created for volunteers to use when judging performances. Using this tool, you first choose to create or open a judgement file. The tool then plays pairs of performances and demands a judgement. The possible judgements match the classifier's possible classifications. Volunteers used this tool to listen to pairs of performances, and provide judgements. Once complete, the volunteers returned a data file containing records that identified the performances that were evaluated and the volunteers' judgements.

These data files were combined into a master file. The master file is a list of records, each record being a performance identifier, and a list of judgements from volunteers. The judgements were reviewed to identify problematic records. A record would have been considered problematic if there had been too much disagreement by the volunteers over the judgement. No problematic records were found, which is encouraging because it suggests that the volunteers all had similar attitudes towards their judgements.

Further analysis was performed on the data to determine its quality. The concern was with records with inconsistent judgements. The data analysis provides insight into the overall consistency of the data. For every record, the variance and standard deviation of the set of human judgements was determined, and then an overall mean of the variance and standard deviation was derived. This gave the following results:

Table 5 Dataset statistics

Attribute	Value
Mean of the Variances	0.116
Mean of the Standard Deviations	0.183

The decisions were highly consistent. Consider that the classifications were transformed into integers, and it is apparent that a data item needs to be several standard deviations from the mean before it results in a different classification. This data analysis gives us confidence in the consistency of the data.

The next step was to convert the performances into features. This reused the Evaluator component to convert each performance into metrics, and subtract them from each other to provide a difference vector, the input to the Classifier. At this point, a data file was created containing the difference vectors and their corresponding judgement. This can be considered the raw data file for the dataset. It contains all the data in an appropriate format, but it still needs to be cleaned and to be broken into training and test sets.

As part of cleaning the data, the dataset's items were balanced across the various categories. Doing so avoids a situation where and imbalanced dataset might only have two data items with the *same* classification and two-hundred with the *better* classification. In cleaning the data, several properties of the data were used. First, given a performance that is classified as *better* or *much better*, by reversing the signs of all the features, the order of performances is reversed, meaning that the result will be *worse*. As well, if two performances are identical, their feature set will consist solely of zeros. Another technique that can be used is to repeat items. This can boost the number of items in an underrepresented category, but it only emphasizes existing data rather than introducing new data (Han, 2012.)

When the data was analyzed, the *better* category contained the most items. The other categories were boosted to provide the same number of data items. The *worse* category was boosted by randomly choosing items from the *better* and *much better* categories and reversing the signs in the feature. The *same* category was boosted by randomly choosing between repeating an existing random item from the category, or introducing a feature vector of all zeroes. The *much better* category was boosted by repeating random items from the same category.

The data was then broken into a training set and a test set. One third of the records were randomly chosen to be in the test set, and the remaining were left for the training set.

Ethical Considerations

A Research Ethics Board application was made for this research because of the human volunteers involved in creating the dataset. The volunteers were exposed to pairs of performances, and recorded judgements of those performances. The data gathering was set up in such a way that no volunteer was coerced into volunteering, and all volunteers had the option of exiting the research at any time. The data was gathered and combined, the combined data being completely anonymous. An analysis of the data showed that most participants agreed on the judgements, making it difficult to de-anonymize the data. Appendix A contains the details of the Research Ethics Board application, and examples of the forms used.

CHAPTER V - RESULTS

The data items in the data set contain a difference vector paired with a judgement derived from the human participant's judgements. Each data item was passed into the classifier and the result was recorded as the actual result. These actual results were compared to the expected results, the human judgements, and recorded in the confusion matrix.

From the test data, the following confusion matrix was generated.

Table 6 Confusion matrix

expected / actual	**worse**	**same**	**better**	**much better**
worse	14	0	1	0
same	0	13	1	2
better	3	5	13	3
much better	0	2	1	7

Of the 65 data items, 47 were correct and 18 were erroneous, giving an accuracy of:

$$accuracy = \frac{47}{65} = 0.723,$$

and an error rate of:

$$errorrate = \frac{18}{65} = 0.277$$

The standard error is:

$$SE = \sqrt{\frac{accuracy \times errorrate}{n}} = \sqrt{\frac{0.723 \times 0.277}{65}} = 0.056.$$

The 95% confidence interval for the accuracy is:

$$[0.614{,}0.812].$$

The precision and recall are:

Table 7 Precision and Recall - Four Categories

	worse	**same**	**better**	**much better**
precision	0.933	0.813	0.542	0.700

recall	0.824	0.65	0.813	0.583

The precision and recall are of less importance than the accuracy because the categories are not unbalanced.

A second approach was taken to assess the tool simply on progress. In this approach, the *better* and *much better* categories were combined into a single category. This was done based on the judged data in the dataset. All the steps, including the data cleaning, were re-done using this changed dataset. Because the data were normalized based on the category with the largest number of elements, this resulted in a larger overall dataset.

The confusion matrix was generated for the test data.

Table 8 Confusion matrix, reduced categories

expected/actual	worse	same	better
worse	22	1	4
same	1	28	6
better	2	9	34

Of the 107 data items in the test set, 84 judgements were correct and 23 were erroneous, giving an accuracy of:

$$accuracy = \frac{84}{107} = 0.785,$$

and an error rate of:

$$errorrate = \frac{23}{107} = 0.215.$$

The standard error is:

$$SE = \sqrt{\frac{0.785 \times 0.215}{107}} = 0.040.$$

The 95% confidence interval is:

$$[0.707, 0.863].$$

The precision and recall are:

Table 9 Precision and Recall - Three Categories

	worse	**same**	**better**
precision	0.815	0.800	0.756
recall	0.880	0.737	0.773

The Weka toolset (Eibe, 2016) is an application that provides several different machine learning algorithms, and provides an easy way to quickly import a dataset and try it on several different machine learning algorithms. Using this modified dataset with only three categories, several classifiers in the Weka toolset were trained and tested for comparison.

Table 10 Weka results

Algorithm	**Accuracy**	**Parameters**
J.48 Decision Tree	0.819	Confidence Factor (pruning): 0.25; Minimum number of instances per leaf: 2
Random Forest	0.842	Max Depth: unlimited; minVarianceProp: 0.001; minNum: 1.0; seed: 1
Multilayer Perceptron	0.853	Learning Rate: 0.3; Momentum: 0.2; Training time: 500; Validation Threshold: 20
Decision Table	0.864	Cross-Validation folds: 1; Best First search method

Chapter VI - DISCUSSION

The analysis of the results shows a 95% confidence interval of [0.614, 0.812] for the accuracy when the classifier maps the pair of performances to one of four categories. When the *better* and *much better* categories are merged into a single category, the classifier instead gives a basic judgement of whether progress is detected or not, rather than trying to assess the amount of progress. In this case, the 95% confidence interval is [0.707, 0.863]. In both cases, the analysis of the results clearly shows that the tool can assess progress. We can consider that the hypothesis of the research question has been tested. The tool works.

The research in this thesis focuses on the viability of an *objective ear*, but now that we know such a tool is viable, how can we use it? The most likely application of the *objective ear* is to embed it into a learning management system (LMS) and to include its classifications as part of the LMS's dashboard. This integration make sense because the LMS will likely organize and store data by musical task, and the tool must be applied to performances of the same task. This list of tasks can easily be extended to store past performances so that the *objective ear* has data to use. The LMS can compress the data if, instead of storing recordings of past performances, it stores the associated vectors of metrics generated by the Evaluator component.

Such an LMS with an integrated *objective ear* would be central to the lesson-practice cycle. Music lessons are typically weekly. During the lesson, the teacher assesses the student's progress on assigned tasks, provides guidance to the student, and prepares a list of tasks for the student to practice for the next lesson. Although the list of tasks changes from week to week, this change is slow, and much of the list overlaps week-to-week. A student will typically practice a task for several weeks.

At each lesson, when the student performs each task that he or she has been practicing, the tool can assess the progress relative to the previous week. This progress gives the teacher immediate feedback that can be used to help the teacher overcome problems with memory, and give an overall assessment. A teacher frequently provides a student with specific goals to accomplish during the practice period. Perhaps a problem area needs to be addressed, or the overall tempo needs to be increased. The teacher, while focusing on these goals, may lose track of the overall progress of a piece. For example, a student may continue to struggle with a specific section within the performance, but overall, the performance may be much improved. By over-focusing on one specific section, the teacher may miss the overall progress a student has made.

The LMS also provides useful historical data that the teacher can use to judge larger trends in a student's progress. For example, a teacher may notice that a student has stagnated on a piece because no progress has been made in that task for several weeks. The teacher may also notice that the student had a bad week and failed to progress in any task. The teacher may also be able to derive trends for the student. Perhaps the student always struggles with Beethoven's compositions, but quickly masters Mozart's. Although the teacher may feel he or she intuitively knows this, the LMS with data from the *objective ear* would provide the teacher with firm evidence. This kind of information could be important when a teacher is suggesting material the student learn for a recital or audition.

The student could also use the *objective ear* as part of a practice regimen. For example, a student could maintain daily recordings of performances, and attempt to make progress each day, thus ensuring that he or she will have progress to show at each week's lesson. Thus, the tool becomes an integral part of the student's motivation during practice sessions.

The LMS with an *objective ear* provides valuable data to researcher who wish to better understand music education. Progress data is highly prized by such researchers because it is so much finer-grained than the kind of overall assessment data that appears at the end of an education program, such as the final grade of a course. Progress data can be paired with many other factors in creating models of music education. Some factors could be properties of the student such as the student's age, years of study, gender, or family income. Others could be data extracted from the LMS itself such as the number of concurrent musical tasks, the overall duration of all the tasks, the amount of time a student spends practicing in a week, or the distribution of practice time over the week. Some of these aspects may be highly personal, and could be used by the LMS to provide the teacher with guidance on how the student learns best. Others could be more general, and provide heuristics that can be applied in general to all music students. The application of the data is limited only by the imagination of the researchers.

The *objective ear* may be aimed at the field of music education, but it implements a general framework that can be applied to the education of many different skills. It is easy to foresee how skills like dancing or figure skating could apply this framework. Most sporting activities are skill-based. The key aspect of this framework is the comparison of performances to get progress. Most of the work in implementing this framework comes from understanding the domain well enough to know which features to extract, and knowing how to properly normalize the data so that the comparisons between performances are valid.

There are several ways in which this research can be expanded and improved. The most obvious next step is to push the *objective ear* out to teachers and students and evaluate how well it works in practice. Much of this effort involves determining how to best integrate the tool into an LMS, and advertise its use to students and teachers. Having students and teachers use the tool for several months would give valuable feedback to the tool, but it would also provide vastly more data. Along with providing the tool's assessments, we could allow teachers to review pairs of performances and give their own assessment. That data could be used to improve the tool.

This new data would help the tool overcome a major deficiency – it needs more data for training. The tool was trained using a data set of around two hundred data items. Increasing the size of the dataset by an order of magnitude or several orders of magnitude could dramatically improve the tool's accuracy. Machine learning algorithms typically improve dramatically based on the number of data items, and although the dataset used by this research was large enough to show that the metrics gathered were sufficient to assess progress, improving the accuracy should be a long-term goal of this research.

Another long-term goal should be to train the tool with more genres of music, making it suitable for music outside of the Classical era. Further research is necessary to determine how to best do this, because multiple approaches present themselves. The easiest approach is to add data items for various genres to the dataset, and then retrain the tool with the new dataset. This approach is risky because the differences between genres may be enough that the overall accuracy of the tool suffers. That is, the tool may not be able to effectively classify different genres with the same tool. In such a case, we may instead partition the dataset into a dataset for each genre, and train a version of the *objective ear* for each genre. Thus, we might end up with a *baroque ear*, *classical ear*, and *romantic ear*. It may be worthwhile to go even more fine-grained and create a *brahms ear* or a *chopin ear*. The literature review did not find any research on music assessment by genre, but such research would be useful because it would provide useful guidance on the success of a general *objective ear*. Research might show that a teacher whose training has been solely with Jazz music may still be effective at assessing Baroque performance, or it may show that such skills are not easily transferrable. When expanding the tool to multiple genres, it is important to remember that the pitch metric uses a learning algorithm to determine likely pitches. It has been trained based on the genre of music. If the *objective ear* is expanded to assess multiple genres, the pitch metric also needs to be retrained for those genres.

The analysis of the results showed that the classifier can be improved. Several examples of alternative classifiers were given with better accuracy than the *objective ear's* decision tree. Even Weka's J.48 algorithm was superior and the *objective ear's* classifier is based on the same algorithm. The difference is that the Weka algorithm implements pruning on the decision tree. This step was skipped because the dataset of small enough that pruning seemed unnecessary. This assumption may have been wrong, as Weka's pruning proved to be more effective. Other algorithms, which discard the self-imposed restriction that the rules generated by the classifier be human-understandable, could be much more effective. Some approaches to machine learning may not be so effective. For example, some approaches which use autoencoding may be less effective, because this progress-assessment framework depends on stable features being extracted and compared.

The *objective ear* will only be used it is easy to students and teachers to use it. This ease of use depends on how effectively it can be integrated into a learning management system, but it also depends on how easily a student can record his or her performance. What can we do to make this recording easier? The MIDI data format is ubiquitous with electronic musical instruments, and can be easily attached to a computer, making it easy to connect instruments with a computer hosting a LMS. A wireless connection would be better, but because of the precise timing involved, MIDI connections have mostly stayed wired. A better approach would be to implement an automated music transcription application to act as a front-end to the objective ear. Such an application would use raw sound waves as input, and convert those sound waves to a computer data format such as MIDI. This would have the benefit of allowing the tool to run on devices which do not have the ports necessary for a MIDI connection such as a smartphone. Automated music transcription is an active field of research, and it may soon be feasible to consider such a front-end.

REFERENCES

Arnold, K. E. & Pistilli, M. D. "Course Signals at Purdue: Using Learning Analytics to Increase Student Success." Proceedings of the 2nd International Conference on Learning Analytics and Knowledge, pp. 267-270, 2012.

Baker, R. & Yacef, K. "The state of educational data mining in 2009: A review and future visions." Journal of Educational Data Mining, vol. 1, no. 1, pp. 3-17, 2009.

Barber, R. & Sharkey, M. "Course Correction: Using Analytics to Predict Course Success." Proceedings of the 2nd International Conference on Learning Analytics and Knowledge, pp. 259-262, 2012.

Bas De Has, W., Jose Pedro Magalhaes, Frans Wiering, and Remco C. Veltkamp. “Automatic Functional Harmonic Analysis.” Computer Music Journal, vol. 37, no. 4, pp. 37-53, 2013.

Blikstein, P. "Using learning analytics to assess students’ behavior in open-ended programming tasks." Proceedings of the 1st international conference on learning analytics and knowledge, pp. 110-116, 2011.

Bora, Uzay, Selmin Tufan, and Gilgen Semih. “A Tool for Comparison of Piano Performances.” Journal of New Music Research, vol. 29, no. 1, pp. 85-99, 2000.

Bull, S. & Kay, J. "Open learner models." Advances in Intelligent Tutoring Systems, pp. 301-322. Springer. 2013,

Burrows, J., and V.S. Kumar. “The Objective Ear: Assess the Progress of a Music Task.” ISCLE, 2018.

Burrows, J., Kumar, V.S., Dewan, A., and Kinshuk. “Assessing a Music Student’s Progress.” ICALT, 2018.

Cemgil, Ali Taylan, and Bert Kappen. “Monte Carlo Methods for Tempo Tracking and Rhythm Quantization.” Journal of Artificial Intelligence Research, vol. 18, pp. 45-81, 2003.

Chordia, Parag, Avinash Sastry and Sertan Senturk. “Predictive Tabla Modelling Using Variable-length Markov and Hidden Markov Models.” Journal of New Music Research, vol. 40, no. 2, pp. 105-118, 2011.

Darrow, Alice-Ann. “Examining the validity of self-report: middle-level singers’ ability to predict and assess their sight-singing skills.” International Journal of Music Education, vol. 24, no. 1, pp. 21-29, 2006.

Dietz-Uhler, Beth and Janet E. Hurn. "Using Learning Analytics to Predict (and Improve) Student Success: A Faculty Perspective." Journal of Interactive Online Learning, vol. 12, no. 1, 2013.

Dix, Alan and Justin Leavesley. "Learning Analyatics for the Academic: An Action Perspective." Journal of Universal Computer Science, vol. 21, no. 1, pp. 48-65, 2015.

Dixon, Simon. “Automatic Extraction of Tempo and Beat from Expressive Performances.” Journal of New Music Research, vol. 30, no. 1, pp. 39-58, 2001.

Dyckhoff, A. L., V. Lukarov, A. Muslim, M. A. Chatti, and U. Schroeder. "Supporting Action Research with Learning Analytics." Proceedings from LAK'13, 2013.

Ebner, Martin, Kinshuk, David Wohlhart, Behnam Taraghi, and Vive Kumar. "Learning Analytics, J.UCS Special Issue." Journal of Universal Computer Science, vol. 21, no. 1, pp. 1-6, 2015.

Ferguson, R. "Learning analytics: drivers, developments and challenges." International Journal of Technology Enhanced Learning, vol. 4, no. 5/6, pp. 304-317, 2012.

Gingras, Bruno and Stephen McAdams. "Improved Score-performance Matching Using Both Structure and Temporal Information from MIDI Recordings." Journal of New Music Research, vol. 41, no. 1, pp. 43-57, 2011.

Goggins, Sean, Wanli Xing, Xin Chen, Bodong Chen, and Bob Wadholm. "Learning Analytics at 'Small' Scale: Exploring a Complexity-Grounded Model for Assessment Automation." Journal of Universal Computer Science, vol. 21, no. 1, pp. 66-92, 2015.

Gouyan, Francis and Simon Dixon. “A Review of Automatic Rhythm Description Systems.” Computer Music Journal, vol. 29, no. 1, pp. 34-54, 2005.

Greller, Wolfgang and Hendrik Drachsler, “Translating Learning into Numbers: A Generic Framework for Learning Analytics.” Educational Technology and Society, vol. 15, no. 3, pp. 42-57, 2012.

Groulx, Timothy. “The Influence of Tonal and Atonal Contexts on Error Detection Accuracy.” Journal of Research in Music Education, vol. 61, no. 2, pp. 233-243, 2013.

Hamanaka, Masatoshi, Keiji Hirata, and Satoshi Tojo. “Implementing ‘A Generative Theory of Tonal Music’.” Journal of New Music Research, vol. 35, no. 4, pp. 249-277, 2006.

Han, Jiawei, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. 3rd edition. Morgan Kaufmann Publishers, Waltham, Massachusetts, 2012.

Heijink, Hank, Luke Windsor, and Peter Desain. “Data processing in music performance research: Using structural information to improve score-performance matching.” Behavior Research Methods, Instruments, & Computers, vol. 32, no. 4, pp. 546-554, 2000.

Hewitt, Michael. “Self-Efficacy, Self-Evaluation, and Music Performance of Secondary-Level Band Students.” Journal of Research in Music Education, vol. 63, no. 3, pp. 298-313, 2015.

Jaschke, Artur C., Honing Henkjan, and Eric J. A. Scherder. “Longitudinal Analysis of Music Education on Executive Functions in Primary School Children.” Frontiers in Neuroscience, vol. 12, pp. 103, 2018.

Kumar, V., Boulanger, D., Seanosky, J., Kinshuk, Panneerselvam, K., and T.S. Somasundaram. “Competence analytics.” Journal of computers in education, vol. 1, no. 4, 2014.

Kumar, V.S., Fraser, S.N., and D. Boulanger. “Discovering the predictive power of five baseline writing competences.” Journal of Writing Analytics, vol. 1, no. 1, 2017.

Lockyer, L. & Dawson, S. "Learning Designs and Learning Analytics." Proceedings of the 1st International Conference on Learning Analytics and Knowledge, pp. 153-156, 2011.

Lockyer, L., Heathcote, E., and Dawson, S. "Informing pedagogical action: Aligning learning analytics with learning design." American Behavioral Scientist, vol. 57, no. 10, 2013

Macfadyen, Leah P., and Shane Dawson. “Numbers Are Not Enough. Why e-Learning Analytics Failed to Inform an Institutional Strategic Plan.” Educational Technology & Society, vol. 15, no. 3, 2012.

Macfadyen, Leah P., Shane Dawson, Abelardo Pardo, and Dragan Gasevic. “Embracing Big Data in Complex Educational Systems: The Learning Analytics Imperative and the Policy Challenge.” Research and Practice in Assessment, vol. 9, pp. 17-28, 2014.

Olmos, Martin and Linda Corrin. "Learning analytics: a case study of the process of design of visualizations." Journal of Asynchronous Learning Networks, vol. 16, no. 3, pp. 39-49, 2012.

Pardo, Bryan and William P. Birmingham. “Algorithms for Chordal Analysis.” Computer Music Journal, vol. 26, no. 2, pp. 27-49, 2002.

Park, Y., & Jo, Il-Hyun "Development of the Learning Analytics Dashboard to Support Students’ Learning Performance." Journal of Universal Computer Science, vol. 21 no. 1, pp. 110-133, 2015.

Pearce, Marcus and Geraint Wiggins. “Improved Methods for Statistical Modelling of Monophonic Music.” Journal of New Music Research, vol. 33, no. 4, pp. 367-385.

Picciano, A. G. "The Evolution of Big Data and Learning Analytics in American Higher Education." Journal of Asynchronous Learning Networks, vol. 16, no. 3, pp. 9-20, 2012.

Raphael, Christopher and Joshua Stoddard. “Functional Harmonic Analysis Using Probabilistic Models.” Computer Music Journal, vol. 28, no. 3, pp. 45-52, 2004.

Russel, Brian. “An empirical study of a solo performance assessment model.” International Journal of Music Education, vol 33, no. 3, pp. 359-371, 2015.

Santos, Jose Luis, Sten Govaerts, Katrien Verbert, and Eric Duval. "Goal-oriented visualizations of activity tracking: a case study with engineering students." Proceedings from LAK'12, 2012.

Scheffel, Maren, Hendrik Drachsler, Slavi Stoyanov, and Martin Specht. "Quality Indicators for Learning Analytics." Educational Technology & Society, vol. 17, no. 4, pp. 117-132, 2014.

Siemens, George. “Learning Analytics: Envisioning a Research Discipline and a Domain of Practice.” Proceedings from LAK’12, 2012.

Siemens, George. “Learning Analytics: The Emergence of a Discipline.” American Behavioral Scientist, vol. 57, no. 10, pp. 1380-1400, 2013.

Slade, S. & Prinsloo, P. "Learning Analytics: Ethical Issues and Dilemmas." American Behavioral Scientist, vol. 57, no. 10, pp. 1510-1529, 2013.

Spector, J. Michael. "Emerging Educational Technologies and Research Directions." Educational Technology & Society, vol. 16, no. 2, pp. 21-30, 2013.

Stambaugh, Laura. “Differences in Error Detection Skills by Band and Choral Preservice Teachers.” Journal of Music Teacher Education, vol. 25, no. 2, pp. 25-36, 2016.

Taraghi, Behnam, Anna Saranti, Martin Ebner, Vinzent Muller, and Arndt Grosmann. "Towards a Learning-Aware Application Guided by Hierarchical Classification of Learner Profiles." Journal of Universal Computer Science, vol. 21, no. 1, pp. 93-109, 2015.

Temperley, “An Evaluation System for Metrical Models.” Computer Music Journal, vol. 28, no. 3, pp. 28-44, 2004.

Temperley, David. “A Unified Probabilistic Model for Polyphonic Music Analysis.” Journal for New Music Research, vol. 38, no. 1, pp. 3-18, 2009.

Wapnick, Joel and Alice-Ann Darrow. “Sectional Versus Intact Evaluations of Four Versions of a Chopin Etude.” Journal of Research in Music Education, vol. 60, no. 4, pp. 462-474, 2013.

Widmer, Gerhard. “Machine Discoveries: A Few Simple, Robust Local Expression Principles.” Journal of New Music Research, vol. 31, no. 1, pp. 37-50, 2002.

APPENDIX A: ETHICS

The following document is the information letter that was sent to volunteers who helped with the data gathering.

INFORMATION LETTER
Learning Analytics for Music Progress

[Date]

Principal Investigator (Researcher):
Joel Burrows
joelburrows@outlook.com

Supervisor:
Dr. Kinshuk
kinshuk@athabascau.ca

Dr. Vive Kumar
vive@athabascau.ca

You are invited to take part in a research project entitled 'Learning Analytics for Music Progress'.

This letter provides an introduction to the project and an invitation to participate in the project. Should you choose to participate, this letter will be followed by an informed consent letter that contains the same information as this letter along with a form for you to fill out. Please contact the principal investigator, Joel Burrows, if you have any questions about the project or would like more information. Participation in this project is completely voluntary, and you can choose to stop participating at any time in this project up until when the data you provide is anonymised and pooled with other participants' data.

Introduction
My name is Joel Burrows and I am a Masters of Information Science student at Athabasca University. As a requirement to complete my degree, I am conducting a research project about learning analytics for music students who are learning a piece of music or a technical exercise. I am conducting this project under the supervision of Dr. Kinshuk and Dr. Vive Kumar.

Why are you being asked to take part in this research project?
You are being invited to participate in this project because you can make basic judgements about the quality of a music performance for music that conforms to Western tonality. In other words, you must be able to hear, and you must have listened to a great deal of music. The term Western tonality sounds very specific, but in fact, this encompasses a large amount of music, including background and incidental music that you may regularly listen to. For example, if you regularly watch television or movies

from the Americas or from Europe, you have listened to sufficient Western tonal music for this study.

What is the purpose of this research project?
The purpose of this study is to develop a software tool that, given a pair of performances of a piece of music or a technical exercise, determines the amount of progress a student made. The tool uses a machine learning technique called supervised learning to determine how to map performances to a level or progress. The data set used for supervised learning must include pairs of performances that have been classified by human experts. As a participant in this study, you will be one of the human experts that helps create this supervised learning data set.

What will you be asked to do?
You will be sent a software application that includes a simple user interface and a collection of music files. The application will randomly choose a pair of performances of the same music excerpt. These excerpts are extremely short, usually between five and thirty seconds in duration. The user interface provides buttons that play the two music files. It will present you with a set of choices: worse, same, better, and much better. You will select which choice best characterizes the progress made between the two performances, and hit the next button. The tool will track your decisions in a file which you will send back to the researchers. You will have two months to create this file. There is no set limit on how many excerpts you must listen to. You may listen to only a small portion of the performances, or you may listen to some pairs multiple times. You choose how much you can contribute. After the data gathering period has completed, you will send your data file by email, or alternatively, using a USB drive. The results for all participants will be combined in the final data set.
At any time, you can contact Joel Burrows to ask questions about the data capture.

What are the risks and benefits?
This study lacks any serious risks to participants. Participating in this study will not necessarily be pleasant. The music performances you will listen to will be of the low quality of a student learning to play a piece of music. Listening to many of these performances may become tedious or frustrating. Thus, the tool for gathering data has been developed such that the participant is in complete control and can stop participating any time he or she chooses. The data can be gathered over as many sessions as the participant wants. Every session will add data to the data file that is returned to the researches. There are no direct benefits to the participants of this study.

Do you have to take part in this project?
As stated earlier in this letter, involvement in this project is entirely voluntary. If you do not have the time to participate, or do not thing you are qualified to participate, you can immediately stop your participation. After returning your results file, this data will be combined anonymously with data from other participants. As a result,
after anonymising your results, you cannot later request those results be removed.

How will your privacy and confidentiality be protected?
The ethical duty of confidentiality includes safeguarding participants' identities, personal information, and data from unauthorized access, use or disclosure. The results file that you return to researchers contains no information other than your judgements of performances. Absolutely no personal data will be included in this file, and the data will

be stored in a completely anonymous way. Likewise, when the data from multiple participants is combined, this final data set will have no means of tracking individual judgements to particular files. There are no limits to this confidentiality. In other words, nothing in this data will be reportable information that must be passed on to legal or professional body.

Page Break

How will my anonymity be protected?

Anonymity refers to protecting participants' identifying characteristics, such as name or description of physical appearance. Absolutely no identifying characteristics will be gathered for a participant other than, perhaps, basic music skill. The data will be stored in a completely anonymous way that prevents tracking data files to participants. If the participant sends an email with the results file, that email will be destroyed immediately after the results file is extracted. If the participant does not want to disclose and email address, a USB drive can be used to pass data to and from the participant.

Every reasonable effort will be made to ensure your anonymity; you will not be identified in publications without your explicit permission.

How will the data collected be stored?

The individual results files will be destroyed within six months of them being returned. The data will be incorporated into a data set that will include all of the pairs of performances and a metadata file that gives the judgment for each pair. This judgment includes the number of human judgements, the mean judgment, and the standard deviation of the judgments. From this data set, it is impossible to learn anything about individual participants. This data set will be kept indefinitely, and may be shared with other researchers who are interested in using this data in their own research.

Who will receive the results of the research project?

The existence of the research will be listed in an abstract posted online at the Athabasca University Library's Digital Thesis and Project Room and the final research paper will be publicly available. The identity of the participants will not be disclosed in any way in this research paper.

Who can you contact for more information or to indicate your interest in participating in the research project?

Thank you for considering this invitation. If you have any questions or would like more information, please contact me, Joel Burrows by e-mail: joelburrows@outlook.com.

If you are ready to participate in this project, please contact Joel Burrows at the above email and return to him a signed consent form.

Thank you.

Joel Burrows

The following is the consent form that all participants filled out and signed:

LETTER OF INFORMATION / INFORMED CONSENT FORM

Learning Analytics for Music Progress

[Date]

Principal Investigator (Researcher):
Joel Burrows
joelburrows@outlook.com

Supervisor:
Dr. Kinshuk
kinshuk@athabascau.ca

Dr. Vive Kumar
vive@athabascau.ca

You are invited to take part in a research project entitled 'Learning Analytics for Music Progress'.

This form is part of the process of informed consent. The information presented should give you the basic idea of what this research is about and what your participation will involve, should you choose to participate. It also describes your right to withdraw from the project. In order to decide whether you wish to participate in this research project, you should understand enough about its risks, benefits and what it requires of you to be able to make an informed decision. This is the informed consent process. Take time to read this carefully as it is important that you understand the information given to you. Please contact the principal investigator, Joel Burrows, if you have any questions about the project or would like more information before you consent to participate.

It is entirely up to you whether or not you take part in this research. If you choose not to take part, or if you decide to withdraw from the research once it has started, there will be no negative consequences for you now, or in the future.

Introduction
My name is Joel Burrows and I am a Masters of Information Science student at Athabasca University. As a requirement to complete my degree, I am conducting a research project about learning analytics for music students who are learning a piece of music or a technical exercise. I am conducting this project under the supervision of Dr. Kinshuk and Dr. Vive Kumar.

Why are you being asked to take part in this research project?
You are being invited to participate in this project because you can make basic judgements about the quality of a music performance for music that conforms to Western tonality. In other words, you must be able to hear, and you must have listened to a great deal of music. The term Western tonality sounds very specific, but in fact, this encompasses a large amount of music, including background and incidental music that you may regularly listen to. For example, if you regularly watch television or movies from the Americas or from Europe, you have listened to sufficient Western tonal music for this study.

What is the purpose of this research project?
The purpose of this study is to develop a software tool that, given a pair of performances of a piece of music or a technical exercise, determines the amount of progress a student made. The tool uses a machine learning technique called supervised learning to

determine how to map performances to a level or progress. The data set used for supervised learning must include pairs of performances that have been classified by human experts. As a participant in this study, you will be one of the human experts that helps create this supervised learning data set.

What will you be asked to do?
You will be sent a software application that includes a simple user interface and a collection of music files. The application will randomly choose a pair of performances of the same music excerpt. These excerpts are extremely short, usually between five and thirty seconds in duration. The user interface provides buttons that play the two music files. It will present you with a set of choices: worse, same, better, and much better. You will select which choice best characterizes the progress made between the two performances, and hit the next button. The tool will track your decisions in a file which you will send back to the researchers. You will have two months to create this file. There is no set limit on how many excerpts you must listen to. You may listen to only a small portion of the performances, or you may listen to some pairs multiple times. You choose how much you can contribute. After the data gathering period has completed, you will send your data file by email, or alternatively, using a USB drive. The results for all participants will be combined in the final data set.
At any time, you can contact Joel Burrows to ask questions about the data capture.

What are the risks and benefits?
This study lacks any serious risks to participants. Participating in this study will not necessarily be pleasant. The music performances you will listen to will be of the low quality of a student learning to play a piece of music. Listening to many of these performances may become tedious or frustrating. Thus, the tool for gathering data has been developed such that the participant is in complete control and can stop participating any time he or she chooses. The data can be gathered over as many sessions as the participant wants. Every session will add data to the data file that is returned to the researches. There are no direct benefits to the participants of this study.

Do you have to take part in this project?
As stated earlier in this letter, involvement in this project is entirely voluntary. If you do not have the time to participate, or do not thing you are qualified to participate, you can immediately stop your participation. After returning your results file, this data will be combined anonymously with data from other participants. As a result, after returning your results, you cannot later request those results be removed.

How will your privacy and confidentiality be protected?
The ethical duty of confidentiality includes safeguarding participants' identities, personal information, and data from unauthorized access, use or disclosure. The results file that you return to researchers contains no information other than your judgements of performances. Absolutely no personal data will be included in this file, and the data will be stored in a completely anonymous way. Likewise, when the data from multiple participants is combined, this final data set will have no means of tracking individual judgements to particular files. There are no limits to this confidentiality. In other words, nothing in this data will be reportable information that must be passed on to legal or professional body.

Page Break

How will my anonymity be protected?

Anonymity refers to protecting participants' identifying characteristics, such as name or description of physical appearance. Absolutely no identifying characteristics will be gathered for a participant other than, perhaps, basic music skill. The data will be stored in a completely anonymous way that prevents tracking data files to participants. If the participant sends an email with the results file, that email will be destroyed immediately after the results file is extracted. If the participant does not want to disclose and email address, a USB drive can be used to pass data to and from the participant.
Every reasonable effort will be made to ensure your anonymity; you will not be identified in publications without your explicit permission.

How will the data collected be stored?
The individual results files will be destroyed within six months of them being returned. The data will be incorporated into a data set that will include all of the pairs of performances and a metadata file that gives the judgment for each pair. This judgment includes the number of human judgements, the mean judgment, and the standard deviation of the judgments. From this data set, it is impossible to learn anything about individual participants. This data set will be kept indefinitely, and may be shared with other researchers who are interested in using this data in their own research.

Who will receive the results of the research project?
The existence of the research will be listed in an abstract posted online at the Athabasca University Library's Digital Thesis and Project Room and the final research paper will be publicly available. The identity of the participants will not be disclosed in any way in this research paper.

Who can you contact for more information or to indicate your interest in participating in the research project?
Thank you for considering this invitation. If you have any questions or would like more information, please contact me, Joel Burrows by e-mail: joelburrows@outlook.com.
If you are ready to participate in this project, please contact Joel Burrows at the above email and return to him a signed consent form.
Thank you.
Joel Burrows

This project has been reviewed by the Athabasca University Research Ethics Board. Should you have any comments or concerns regarding your treatment as a participant in this project, please contact the Research Ethics Office by e-mail at rebsec@athabascau.ca or by telephone at 1-800-788-9041, ext. 6718.

Page Break

Informed Consent:
Your signature on this form means that:

- You have read the information about the research project.
- You have been able to ask questions about this project.
- You are satisfied with the answers to any questions you may have had.
- You understand what the research project is about and what you will be asked to do.
- You understand that you are free to withdraw your participation in the research project without having to give a reason, and that doing so will not affect you now, or in the future.

- You understand that if you choose to end your participation **during** data collection, any data collected from you up to that point will be destroyed.
- You understand that your data is being collected anonymously, and therefore cannot be removed once the data collection has ended.

	YES	NO
I allow data collected from me to be archived in the Music Progress Data Set.	◯	◯

Your signature confirms:

- You have read what this research project is about and understood the risks and benefits. You have had time to think about participating in the project and had the opportunity to ask questions and have those questions answered to your satisfaction.

- You understand that participating in the project is entirely voluntary and that you may end your participation at any time without any penalty or negative consequences.

- You have been given a copy of this Informed Consent form for your records; and

- You agree to participate in this research project.

____

Signature of ParticipantDate

Principal Investigator's Signature:

I have explained this project to the best of my ability. I invited questions and responded to any that were asked. I believe that the participant fully understands what is involved in participating in the research project, any potential risks and that he or she has freely chosen to participate.

____

Signature of Principal InvestigatorDate

APPENDIX B: DECISION TREE RULES

```
if ornament < -1.5
--if errorGroup < -5.994314339497421E-4
----if ornament < -3.5
------if errorGroup < -0.0015198255660195704
--------if pitch < 26.5
----------if pitch < -0.5
------------ : muchBetter
----------else (pitch > -0.5)
------------>  : better
--------else (pitch > 26.5)
---------->  : worse
------else (errorGroup > -0.0015198255660195704)
--------> if tempo < -2.534302503052502
--------> --if tempo < -334.27575757575755
--------> ---- : muchBetter
--------> --else (tempo > -334.27575757575755)
--------> ---->  : worse
--------> else (tempo > -2.534302503052502)
--------> --> if errorGroup < -0.0013435720634320204
--------> --> -- : worse
--------> --> else (errorGroup > -0.0013435720634320204)
--------> --> --> if rhythm < 5.5
--------> --> --> -- : same
--------> --> --> else (rhythm > 5.5)
--------> --> --> --> if tempo < -0.492436974789916
--------> --> --> --> --if tempo < -1.2919180492709899
--------> --> --> --> ---- : better
--------> --> --> --> --else (tempo > -1.2919180492709899)
--------> --> --> --> ---->  : same
--------> --> --> --> else (tempo > -0.492436974789916)
--------> --> --> --> -->  : better
----else (ornament > -3.5)
------>  : muchBetter
--else (errorGroup > -5.994314339497421E-4)
----> if pitch < 8.5
----> --if tempo < -0.32684178743961345
----> ----if rhythm < 3.5
----> ------if rhythm < 1.5
----> --------if tempo < -263.26863779495363
----> ---------- : muchBetter
----> --------else (tempo > -263.26863779495363)
----> ----------> if tempo < -7.592436974789916
----> ----------> --if pitch < -11.5
----> ----------> ---- : better
----> ----------> --else (pitch > -11.5)
```

```
----> ----------> ---->  : same
----> ----------> else (tempo > -7.592436974789916)
----> ----------> --> if errorGroup < 2.989577380853583E-4
----> ----------> --> --if pitch < 4.0
----> ----------> --> ---- : better
----> ----------> --> --else (pitch > 4.0)
----> ----------> --> ---->  : same
----> ----------> --> else (errorGroup >
2.989577380853583E-4)
----> ----------> --> -->  : muchBetter
----> ------else (rhythm > 1.5)
----> -------->  : muchBetter
----> ----else (rhythm > 3.5)
----> ------> if rhythm < 5.5
----> ------> --if tempo < -6.302849002848985
----> ------> ----if tempo < -61.299235949235936
----> ------> ------ : same
----> ------> ----else (tempo > -61.299235949235936)
----> ------> ------>  : better
----> ------> --else (tempo > -6.302849002848985)
----> ------> ---->  : worse
----> ------> else (rhythm > 5.5)
----> ------> --> if errorGroup < -4.2998524711089156E-4
----> ------> --> -- : same
----> ------> --> else (errorGroup > -4.2998524711089156E-
4)
----> ------> --> --> if tempo < -2.6142857142857148
----> ------> --> --> -- : muchBetter
----> ------> --> --> else (tempo > -2.6142857142857148)
----> ------> --> --> --> if pitch < -2.0
----> ------> --> --> --> -- : better
----> ------> --> --> --> else (pitch > -2.0)
----> ------> --> --> --> -->  : same
----> --else (tempo > -0.32684178743961345)
----> ----> if pitch < 0.5
----> ----> --if errorGroup < -7.527853056308285E-5
----> ----> ----if errorGroup < -2.056331826068656E-4
----> ----> ------if ornament < -5.5
----> ----> -------- : muchBetter
----> ----> ------else (ornament > -5.5)
----> ----> -------->  : better
----> ----> ----else (errorGroup > -2.056331826068656E-4)
----> ----> ------>  : muchBetter
----> ----> --else (errorGroup > -7.527853056308285E-5)
----> ----> ----> if tempo < 2.6858359032272077
----> ----> ----> --if tempo < 1.95776397515528
----> ----> ----> ----if ornament < -3.5
```

```
----> ----> ----> ------ : better
----> ----> ----> ----else (ornament > -3.5)
----> ----> ----> ------>  : same
----> ----> ----> --else (tempo > 1.95776397515528)
----> ----> ----> ---->  : muchBetter
----> ----> ----> else (tempo > 2.6858359032272077)
----> ----> ----> -->  : same
----> ----> else (pitch > 0.5)
----> ----> --> if errorGroup < 1.13071008593397E-4
----> ----> --> -- : better
----> ----> --> else (errorGroup > 1.13071008593397E-4)
----> ----> --> --> if tempo < 39.769268925518936
----> ----> --> --> -- : better
----> ----> --> --> else (tempo > 39.769268925518936)
----> ----> --> --> -->  : muchBetter
----> else (pitch > 8.5)
----> --> if errorGroup < -4.835626226151234E-4
----> --> -- : muchBetter
----> --> else (errorGroup > -4.835626226151234E-4)
----> --> --> if tempo < -3.3810307017543857
----> --> --> -- : muchBetter
----> --> --> else (tempo > -3.3810307017543857)
----> --> --> --> if tempo < 6.841348265261308
----> --> --> --> --if rhythm < 38.0
----> --> --> --> ---- : better
----> --> --> --> --else (rhythm > 38.0)
----> --> --> --> ---->  : muchBetter
----> --> --> --> else (tempo > 6.841348265261308)
----> --> --> --> -->  : muchBetter
else (ornament > -1.5)
-->  : same
```

APPENDIX C: TABLE OF SYMBOLS

Table 11 Symbols

Symbol	**Description**
A	The index into a flex-time array.
AD	The number of actual degradations.
AmI	The number of actual minor improvements.
AMI	The number of actual major improvements.
AS	The number of actual stagnations.
$cost_V$	The cost of adding n_P to n_L's voice.
D	the set of data in a partition while training a tree.
f_V	The pitch difference between n_P and n_L.
FD	The number of misclassified degradations.
FmI	The number of misclassified minor improvements.
FMI	The number of misclassified major improvements.
FS	The number of misclassified stagnations.
F_T	Set of tempo factors.
i_i	The inter-offset interval of a note.
m	The number of categories for classification.
n	The number of items in the test set.
n_i	A note.
n_P	Potential addition to a voice.
n_L	The last note in voice.
p_i	The pitch of a note.
r_i	The offset-to-offset interval of a note.
R	Set of expected IOIs.
set_S	A set of simultaneous notes.
set_V	A set of voices.
S	The split level of a group.
t_V	The inter-onset interval between n_P and n_L.
T	Tempo
TD	The number of degradations classified correctly.

TmI	The number of minor improvements classified correctly.
TMI	The number of major improvements classified correctly.
TS	The number of stagnations classified correctly.
T_r	The trigger for grouping.
w_{fV}	Weighting factor against f_V.
w_{tV}	Weighting factor against t_V.
X	The point in time during a performance that a note is played, normalized to [0,1] where 0 is the beginning of the performance and 1 is the end of the performance.

APPENDIX D: PUBLICATIONS

Current Publication:
Burrows J., Kumar V. (2018) *The Objective Ear*: Assessing the Progress of a Music Task. In: Chang M. et al. (eds) Challenges and Solutions in Smart Learning. Lecture Notes in Educational Technology. Springer, Singapore. DOI: https://doi.org/10.1007/978-981-10-8743-1_15

Upcoming Publication:
ICALT 2018 in July 2018
Paper submitted to the journal: Smart Learning Environments

APPENDIX E: ETHICS APPROVAL

Athabasca University
RESEARCH CENTRE

CERTIFICATION OF ETHICAL APPROVAL

The Athabasca University Research Ethics Board (AUREB) has reviewed and approved the research project noted below. The AUREB is constituted and operates in accordance with the current version of the Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans (TCPS) and Athabasca University Policy and Procedures.

Ethics File No.: **22286**

Principal Investigator:
Mr. Joel Burrows, Graduate Student
Faculty of Science & Technology\School of Computing & Information Systems

Supervisor:
Dr. Kinshuk Kinshuk (Co-Supervisor)
Dr. Vivekanandan (Vivek) Kumar (Co-Supervisor)

Project Title:
Learning Analytics for Music Progress

Effective Date: August 12, 2016 **Expiry Date**: August 11, 2017

Restrictions:

Any modification or amendment to the approved research must be submitted to the AUREB for approval.

Ethical approval is valid *for a period of one year*. An annual request for renewal must be submitted and approved by the above expiry date if a project is ongoing beyond one year.

A Project Completion (Final) Report must be submitted when the research is complete *(i.e. all participant contact and data collection is concluded, no follow-up with participants is anticipated and findings have been made available/provided to participants (if applicable))* or the research is terminated.

Approved by: **Date:** August 12, 2016

Ali Akber-Dewan, Chair
School of Computing & Information Systems, Departmental Ethics Review Committee

---

Athabasca University Research Ethics Board
University Research Services, Research Centre
1 University Drive, Athabasca AB Canada T9S 3A3
E-mail rebsec@athabascau.ca
Telephone: 780.675.6718

APPENDIX F: ETHICS RENEWAL

Athabasca University
RESEARCH ETHICS BOARD

The future of learning.

CERTIFICATION OF ETHICAL APPROVAL - RENEWAL

The Athabasca University Research Ethics Board (AUREB) has reviewed and approved the research project noted below. The AUREB is constituted and operates in accordance with the current version of the *Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans (TCPS)* and Athabasca University Policy and Procedures.

Ethics File No.: 22286

Principal Investigator: Joel Burrows, Graduate Student, School of Computing & Inforrmation Systems

Supervisor (if applicable): Vive Kumar, Professor, Faculty of Science & Technology

Project Title: 'Learning Analytics for Music Progress'

Effective Date: July 26, 2017 **Expiry Date**: July 31, 2018

Restrictions:

- Any modification or amendment to the approved research must be submitted to the AUREB for approval.
- Ethical approval is *valid for a period of one year*. An annual request for renewal must be submitted and approved by the above expiry date if a project is ongoing beyond one year.
- A Project Completion (Final) Report must be submitted when the research is complete *(i.e. all participant contact and data collection is concluded, no follow-up with participants is anticipated and findings have been made available/provided to participants (if applicable))* or the research is terminated.

Approved by: **Date: July 26, 2017**

Joy Fraser, Chair
Athabasca University Research Ethics Board

Athabasca University Research Ethics Board
University Research Services, Research Centre
1 University Drive, Athabasca AB Canada T9S 3A3
E-mail: rebsec@athabascau.ca
Telephone: 780.675.6718