Abstract: Musical keys and their modes have intrigued musicians and scientists alike. While a trained musician can easily identify the key a piece is in, computers have a bit of a harder time understanding how they can come to be. What pops up is an interesting question: what makes a key? There's more to it than just a letter number. Musical keys can be described by mode and key, with each mode adding a different tonality, while each key adding a different tonic. I plan to take the time for this final project to discuss implementations of Artificial Intelligence algorithms in finding the key of a musical piece.Three classification algorithms were used, and their overall performance with predicting keys will be discussed

I. INTRODUCTION

The three classification algorithms used to analyze the dataset were Neural Net Classification, Softmax Regression, and Naive Bayes Classification. I was able to obtain high accuracies (94%, 91%, 87%) for all models except Naive Bayes Classification. The dataset had to be modified via exploratory data analysis and feature selecting.

II. BACKGROUND OF MUSIC THEORY

A musical scale is a set of notes within an octave arranged in sequential order by their pitch. The descending and ascending interval relationships between the notes define each scale. Most music is written in different modes depending on what the first degree of the scale is. For the purpose of these experiments, I will only be discussing the major scale. The principal major scale, C Major, can be seen in a piano on Fig 1. In a piano, the white notes represent whole notes, while the black notes represent accidental notes. These accidental notes can be described as being a "half-step" from the previous white note. An accidental (# or b) is usually added to the note either before or after to describe this half step. However, one can notice that there is no black note between E and F and B and C. This is intentional, as those notes are actually a half step apart instead of a whole step apart. As such, one can define any Major scale using these intervals of whole steps and half steps.

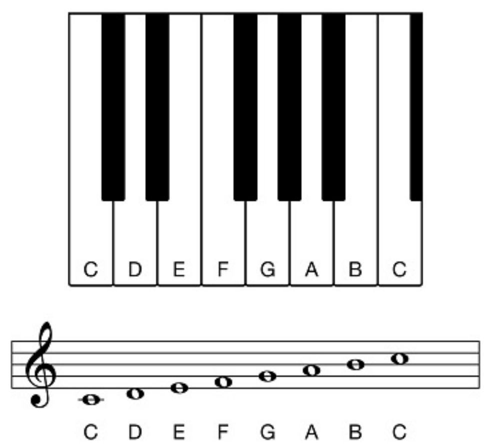

Figure 1: C Major Scale

For example, if one were to ask what the D Major Scale would be, a simple formula of applying half steps to the 3rd-4th notes and 7th-8th notes can be used. As such, the D major scale can be described as:

D E F# G A B C# D

Where the half step falls between the F# and G, and C# and D.

So, how can a musician usually tell if a piece is using a certain scale over the other?

Well, if a piece is written in a certain major scale, the most important note is usually the tonic, or the first degree of the scale. This note is usually emphasised the most throughout the piece. Another note that is important to determining the key of a piece is the dominant note, or the fifth degree of the scale. For C, for example, this would be G. The fifth degree of the scale (dominant) should also be of importance. So how does one measure emphasis? One way is to see whether it falls on the strongest beat of a piece (i.e. the beginning of a measure), another way is to see how often it is played and the duration. The other notes of importance can also be considered, as well intervals that are common in the major.

III. THE DATASET

The dataset I used in this project is DCM's Lab The Annotated Mozart Sonatas: Score, Harmony, and Cadence Database. This database contained detailed information on 18 piano sonatas composed by Mozart Amadeus Wolfgang, a famous late 1700's music composer. It's features involved notes values, beats, duration, measure numbers, global key, local key, cadence, chord, and the like. The dataset essentially represented all the information a knowledgeable musician could gleam from looking at the written piece.

A. *Exploratory Data Analysis*

As noted from Section I, the key elements in determining the key of a musical piece are: the notes played, what notes fall on the strongest beat, and duration. Taking this into account, I structured a new dataframe with the features being: times_played, down_beat, duration, and the global key of the piece. A multi index was used to categorize the data, with the first index being a label for the data, and the second index being the list of possible notes. As such, each feature was mapped to the specific note of a specific piece.

This brought an issue though, as the number of pieces was very small (82), while the number of features was relatively large (four). To combat this I collected the data in intervals of five measures, which gave me a good size data set. However, another issue arose as well. The dataset was clearly unbalanced, as seen in Figure 2.

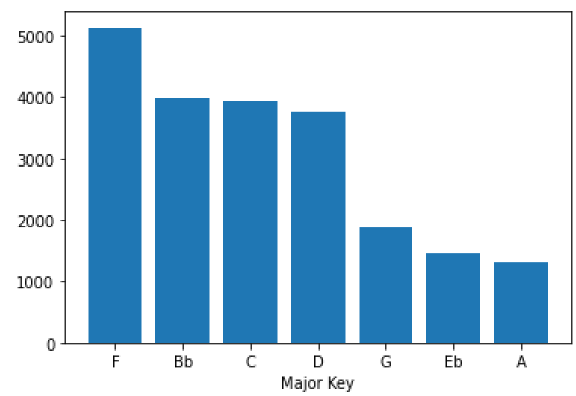

Fig 2: Unbalanced Data for Major Keys in Mozart Sonatas

To balance the data, and as this is an exploratory analysis instead of an in depth one, I took out the major keys of G, Eb, and A. This allowed the data to be more balanced. Unbalanced data can cause issues when training models, as the models can assign the same level of weights to each class even though the distribution of the classes are different.

An interesting thing occurred when I compiled the data: density can be seen to occur around certain notes. An example of this is the first few measures of Piano Sonata

No. 1 in C Major. As such, this data shows promising results for the models.

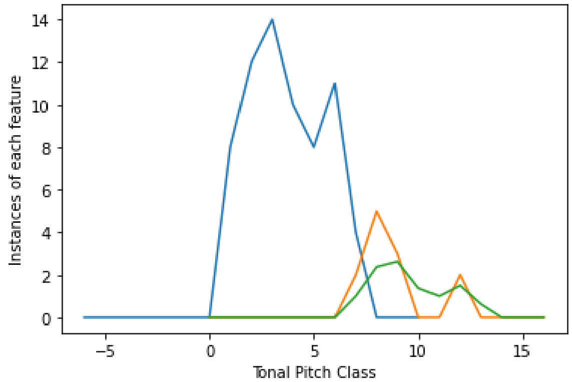

Fig 3: Density comparison of features and the notes.

III. NEURAL NETWORK CLASSIFICATION

The first model I used to train and fit my data was a neural network. This neural network works by using a network of functions to translate the data into predictive classification models. Only four layers were used in my neural network. One was to flatten the 2d data into a one dimensional array, two layers were fully connected layers, one layer was an Activation layer using relu, and another layer was a softmax layer that applied logistic regression to the model. The data was normalized for training and testing to allow for equal weight for each feature. Overall the model was highly accurate in predicting the key of a piece.

Table 1: Metrics for Neural Network

	Accuracy	Precision	Recall
F major	90.7%	80.4%	85.06%
C major	95.46%	87.5%	90.32%
D major	97.95%	91.66%	95.65%
Bb major	94.3%	90.6%	75%
Average	94.50%	81.54%	86.51%

IV. SOFTMAX REGRESSION

Softmax regression works by using statistical probability via logistic functions to predict multi classed data. It works much like logistic regression, except for a key difference in that it takes the sum of the standard exponential function of the output vector. Normal Logistic regression works mainly for binary classification, and as such only one logistic function is needed. The data was normalized for training and testing to allow for equal weight for each feature. This model behaved worse than the neural network classification model. The accuracy of the model is still very high at 91.43%. However, the average precision and recall scores should not be ignored. Overall, this model does show itself to be promising. Further research is required.

Table 2: Metrics of Softmax Regression

	Accuracy	Precision	Recall
F major	85.93%	72.73%	73.56%
C major	91.904%	77.61%	83.87%
D major	95.90%	84.00%	91.30%
Bb major	91.97%	83.33%	67.31%
Average	91.43%	79.17%	79.01%

V. NAIVE BAYES CLASSIFIER

The Naive Bayes model was the last model I used to train and fit my data. This model is based on Gaussian distributions, and the probabilities of the x feature given y. I was excited to test my model on this, as Gaussian distributions are based on density. As seen in my data analysis section, the density of features were higher at certain notes. Before I trained my model, the data was flattened.

Table 3: Metrics of Naive Bayes Classifier

	Accuracy	Precision	Recall
F major	85.31	75.00%	53.42%
C major	88.67%	62.34%	90.57%
D major	92.98%	92.98%	92.98%
Bb major	83.61%	83.61%	79.69%
Average	87.64%	78.48%	79.17%

The model turned out to be the most inaccurate out of all the models analyzed. I believe it is because I flattened the feature set, increasing the number of features. Each feature was equally weighted, even though certain notes that were never played should not have been weighted the same as other notes that were played quite a lot. However, this model does show promise. With other data sets with too many features, Naive Bayes usually fails to be accurate. An example of this is the MNIST handwritten data analyzed in class. Using the same naive bayes algorithm showed about a 50% accuracy. I am confident that with more feature scaling the average accuracy, precision, and recall can increase.

VI. CONCLUSION

After analyzing and feature-selecting DCM's Lab The Annotated Mozart Sonatas: Score, Harmony, and Cadence Database, I was able to train three models to obtain high accuracy scores for predicting the musical key of a piece. Overall the Neural Net Classifier had the highest accuracy, precision and recall scores. The Softmax Regression model came second, while the Naive Bayes Classifier came third. Using the specific feature selection (quantifying notes played, notes played on the strongest beat, and duration of each note) I hypothesized that using specific feature selection would be beneficial and the hypothesis ended up being backed up by the models. That specific feature selection was quantifying notes played, quantifying notes played on the strongest beat, and duration of each note. One can summarize from this

preliminary research that further research
into these factors could help future music
key prediction models