When considering audio information we are often working with complex data sets. Audio tends to include a significant amount of signal noise. This noise refers to unwanted information and ultimately unnecessary features in the data set. This is a crucial concept to grasp, as once you have decided what your production goal is with the data set you may then start to process your audio data accordingly.

Sound is defined as the audible perception of variation in air pressure. When capturing a sound source we end up with a series of complex waveforms which may be defined in terms of frequency and loudness. When viewing a real-world captured waveform in two dimensions, plotting frequency (Hz) against loudness (dB), depending on the source of the information, we almost never will be viewing a waveform consisting of a mathematically simple type, such as a sine wave. This is one instance in which we encounter auditory complexity in our analysis.

To simplify a harmonically rich and layered source we may implement what is known as a Fourier transform. A Fourier transform is a mathematical model which breaks down functions measuring space and or time into functions measuring temporal frequency and loudness. In practice, we may take a signal including multiple individuals singing and or speaking simultaneously and isolate each of them so that we may remove the unwanted information. The model is defined as follows:

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)\ e^{-2\pi i x \xi}\,dx$$

Or in the case of the variable x representing time:

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi)\ e^{2\pi i x \xi}\,d\xi$$

In integral calculus we can look at this equation and begin to gather an understanding of the inner workings of the process. There is an implementation of Euler's formula, which allows for us to measure amplitude for individual waveforms, and the integral allows us to view information over time.

However, through the utilization of Python and libraries such as Librosa, we do not need to apply this model by hand. Librosa was developed specifically for music and audio analysis. It includes methods such as $stft()$, or short-time Fourier transform, which will perform the above described process. This is only one method that Librosa defines for us, there are many more, from time-domain processing, signal generation to pitch and tuning. This library is potentially crucial in any audio data analysis. I have used Librosa when viewing spectrograms, which display a heat-map of the frequency spectrum as it correlates to time. I have also used Librosa when analyzing pitch information, which allows analysts to determine repetition, actual note performance information in music through the use of a piano roll, and more.

One project that I worked on, which is simply a piece in a larger project that I have ongoing, was a song genre classifier. I achieved this through the use of the Spotify Developer API. A number of audio features had to be taken into account to actually classify by genre. One of the features that tended to be obviously determinant in genre classification was 'instrumentalness.' Instrumentalness could be determined through the use of a Fourier transform where we break down the songs waveform, determine harmonic and timbre and thus instrumental content, search for the use of a human voice and then categorize based off our returned value ranging from 0.0 to 1.0.

Waveform complexity is not the only barrier when analyzing audio data. Depending on the desired outcome, the quality of the data can be crucial. When considering music, the standard sampling rate is 44.1kHz and the standard bit depth will rest at 16 or 24 bit. Often when using audio data however, we will not have access to audio of this quality. This means that if you were to perform a high level analysis of a waveform as complex as a piece of music

you may find yourself limited in what you can actually process. Speech and ambient audio data sets will generally not present the same kinds of challenges.

Processing audio and music data when taking a machine learning approach includes an extensive effort towards feature extraction. Feature extraction will allow access to spectral centroid, spectral rolloff, spectral bandwidth, zero-crossing rate, Mel-frequency cepstral coefficients, chroma features and others. These can be useful when determining 'center-of-mass' of an audio signal at an instance in time which can be used for analyzing timbre, or converting the scale of frequency representation into a logarithmic format which more closely represents the auditory perception of humans so that we may perform more effective analyses towards classifications, mood recognition and instrument classifications.

Presenting the performance of any analysis is determinant on the context. In the instance of my aforementioned genre classifier, I presented the performance of the machine learning models and confusion matrix inherent therein. A confusion matrix is an effective way of summarizing the performance of a model. The confusion matrix simply states, how often was x classified as $(a_1, a_2, a_3, \ldots, a_n)$. Through business intelligence tools, such as Tableau, we can visualize and present this information in an easy to consume format.