**Extract, Transform and Load Analysis**

**Project Overview**

This module focused on extracting and cleaning multiple data files. After transforming, the data was loaded into a database created on SQL. This process was done with analysis using scatter plots and steps needed to understand the concepts. For the challenge, however, it was important to create a pipeline which did the extracting, transforming, and loading steps with more automation with the form of a callable function.

**Considerations and Assumptions**

The transformation steps followed a specific process which ensured all necessary variables and columns were considered. This was done through lamda functions and regular Pandas functions such as dropna, rename, etc. When the data was loaded into SQL, the changes were retained as a table with all the three data files condensed into one.

This process was specific to the data types and files on hand; thus, it is important to consider the integrity of the data when using the pipeline many times throughout the future. It is important to note that if data changes do occur, that the steps may not be relevant and may cause errors. This is where the try/except blocks can be used to recover from an entire function crashing. They can provide a roadblock if large data files did encounter changes.

The Wikipedia data, as it is a JSON, file will stay intact unless major changes occur. The Kaggle metadata, however, is large and has many different data types and columns. Since this data is from a website, if the website chooses to add more languages, for example, that we did not see before, this will cause the pipeline to fail at those parts. Similarly, the website can choose to label variables differently from now, so the same problem can occur. Even though, the pipeline we created is a powerful tool, it may not respond well to major data changes unless many try/except blocks are placed in. It can be assumed that most of the data will stay intact, but the uncertainty of the rest will affect the relevancy of the pipeline function that was created.

**Conclusion**

In conclusion, ETL is a strong tool when working with multiple datasets from multiple sources. It is a lengthy and specific process that will create an end-product that is suitable for further data analysis. However, it is important to consider that when data is being sourced from oscillating sources, the work done, will also have to be changed to fit the data. This understanding will help prepare for errors better and can even be used to find the large relevancy issues over time. This tool will help efficiency for companies and projects that need to constantly gather data for their day-to-day business. For example, Spotify, music company, must source data about listeners and music to help create member weekly playlists. That is not an easy task, but the ETL tool can make it ETL tool is vast and can make a large impact on data analysis for a company now, and in the future.