**Capstone Technical Report:**

**Predicting Popularity on Spotify and YouTube**

**By Andrew Vieira**

**Project Statement:**

What qualities make some songs more popular than others? Can one use this information to predict the number of plays (streams or views) a song achieves on Spotify or YouTube?

**Audience:**

While anyone who is interested in music, particularly on Spotify and YouTube, would find the results interesting, the parties who would benefit the most are the ones who profit from popular music, such as musicians, songwriters, and recording companies.

**Analytical Questions:**

The datasets I used for this project include potential ways to measure the popularity of a song, such as the number of streams on Spotify and views and likes on YouTube. They also contain certain qualities of songs: danceability, energy, key, loudness, speechiness, acousticness, instrumentalness, liveness, valence, and tempo. Their definitions are included in the Data Dictionary. These metrics have all been quantified in some way. Some of them use standard measurements; loudness is measured in decibels, tempo is in beats per minute, and key is a standard major scale, mapped to an integer. The other metrics, such as danceability and energy, were rated on a scale of 0 to 1, rounded to three significant digits. Although those metrics can be considered subjective, it is possible, but not verified, that a computer algorithm determined the values.

My goal was to determine a correlation between the qualities of a song and its popularity. Is it possible to create a formula or an algorithm that predicts the number of Spotify streams or YouTube views a song will get, based on its quality metrics? Which factors affect popularity? For example, does each genre require a different formula, or could one work for all of them?

**Criteria for Success:**

* Although I do not expect every metric to affect popularity, I do expect at least one of them to affect the number of streams or views a song has. If multiple factors affect popularity, they might work together in some way. (For example, streams increase when two metrics increase simultaneously.) Success means I find which metrics correlate with popularity.
* It is easy to declare one music genre the most popular, but not everybody likes the same kind of music. I believe the results would be different for each genre. (For example, I expect heavy metal to have high energy and classical to have low energy.) I would find it unusual if all genres produced the same formulas.
* Any formulas or algorithms I generate should have a high degree of accuracy. A good strategy is to use 75% of the rows as training data, and the other 25% as testing data. A successful algorithm should maintain that accuracy if new songs were added to the table.

**Datasets:**

The main dataset, Spotify\_Youtube.csv, was downloaded from the Kaggle page *Spotify and Youtube*. (<https://www.kaggle.com/datasets/salvatorerastelli/spotify-and-youtube>) It was created by Salvatore Rastelli, Marco Guarisco, and Marco Sallustio. After cleaning, it contains 20,716 rows and 29 columns about 18,862 unique songs. Songs with multiple artists have a row for each artist. I am unsure how this information was collected, but after visiting a few of the listed URLs, I can confirm that these Spotify listings and YouTube videos are real. Along with basic statistics, such as the number of Streams on Spotify and views on YouTube, the creators also included the previously mentioned qualities about the songs themselves.

A second dataset, Final database.csv, was downloaded from another Kaggle page, called *Spotify HUGE database - daily charts over 3 years.* (<https://www.kaggle.com/datasets/pepepython/spotify-huge-database-daily-charts-over-3-years?select=Final+database.csv>). I renamed it Final\_database. After cleaning, the set, whose creator is simply named Pepe, has 170,628 rows and 151 columns about 62,420 unique songs. Some songs are in both datasets, but each set has songs that are not in the other. Some of the columns, such as title, artist, and album are the same in this table as in the first one. Even the quality metrics are in this set, with the same values for songs in both tables. Most likely, the creators of these sets did not calculate these statistics and obtained them from the same source.

Final\_database does not have YouTube data. Instead, it shows the popularity of songs based on their positions on Spotify Daily Top 200 chart, aggregated from 2017 to 2020. If the song charted in multiple countries, it has a row for each of those countries, each with the respective popularity score. It also lists additional useful facts, such as genre and release date, as well as two extra quality metrics: mode and time signature. For analysis involving genres, I joined the two tables so that 11,695 songs in the main dataset have genres included. To prevent the table from being unmanageably large, songs from that were not originally in Spotify\_Youtube were not added from Final\_database.

The Excel file DataDictionary\_DataHandlingSummary.xlsx contains the data dictionary and data handling summary. Please see this file for more details about the data fields, data cleaning techniques, and treatment of missing or problematic data.

**Risks/Assumptions:**

* Missing values limit the analysis I can perform on certain songs. For example, some songs do not have information about a YouTube video, possibly because they don’t have official videos. These were analyzed for popularity on Spotify but excluded from YouTube analysis. Similarly, songs with videos but no count of Spotify streams could only be part of YouTube tests. A few songs that had both kinds of information missing were excluded from all tests. Also, I could not determine the genres of all songs; these songs were excluded from genre-based analysis.
* The three measures used to measure popularity are the number of streams on Spotify, the number of views on YouTube, and the number of likes on YouTube. Only the following eleven metrics were used as inputs for modeling: danceability, energy, key, loudness, speechiness, acousticness, instrumentalness, liveness, valence, tempo, and duration (length of song in milliseconds).
* Popularity is not the only factor that determines streams and views. A more recently released song did not have the time to gain as many streams as an older song. A better measure than absolute streams/views is the rate of streams/views over the time the song was on each platform. Unfortunately, upload dates are not in either table, and finding that information for a sufficient number of songs is impractical.

**Analysis Tools:**

* **Excel (Microsoft Office 365):** Used to clean datasets:
  + Detected and deleted blank values
  + Detected and deleted duplicate rows
  + Added new columns and determined values to insert into them
  + Removed unnecessary columns
  + Wrote data dictionary and data handling summary
* **Jupyter Notebook 6.5.2:** Used to create models:
  + Launched from Anaconda Navigator 2.4.0
  + Wrote code in Python
  + Imported from pandas, NumPy, Seaborn, matplotlib, and scikit-learn
  + Inspected datasets and created charts to find patterns
  + Constructed regression models and calculated their accuracies
* **Tableau Desktop 2023.1.0:** Used to graph data
  + Histograms to show distributions of songs by numbers of streams/view/likes
  + Scatterplots to compare the number of plays of each song vs each song quality metric
  + Scatterplots to compare two quality metrics of each song
  + Parameters to specify the input and output metrics for the above charts, as well as narrow the data down by genre

**Models:**

* **For all models:**
  + 75% of the songs, chosen at random, were used for training. The other 25% were for testing.
  + The baseline model, the standard to which the predicted models are compared, is simply guessing the average (mean) value for each prediction.
  + Accuracy is measured with the root mean squared area (RMSE).
  + Although different types of regression were checked, the best was always a linear regression model.
* **Model for predicting Spotify streams:**
  + Average number of streams for test songs = 131,186,291
  + RMSE for baseline = 232,646,553
  + Key column was replaced with binary-value dummy columns
  + Outliers removed:
    - Number of streams greater than 1.5 billion
    - Tempo equal to 0 or greater than 230 bpm
    - 109 songs (0.60% of original dataset) removed
  + Model: Streams = 258144754 + 46678395\*Danceability - 91527675\*Energy + 5140060\*Loudness - 86702602\*Speechiness -77627621\*Acousticness - 43720965\*Instrumentalness - 36207394\*Liveness - 48573063\*Valence - 26000\*Tempo + 29.39\*Duration\_ms + 1806598\*Key\_1 - 10359527\*Key\_2 + 2625384\*Key\_3 - 3298022\*Key\_4 + 7309562\*Key\_5 - 476349\*Key\_6 - 14685578\*Key\_7 + 5825261\*Key\_8 - 14443331\*Key\_9 - 8621322\*Key\_10 - 534147\*Key\_11
  + RMSE for this model = 197,570,949, a 15% improvement over baseline RMSE
* **Model for predicting YouTube views:**
  + Average number of views for test songs = 87,437,106
  + RMSE for baseline = 228,587,416
  + Outliers removed:
    - Number of views greater than 2 billion
    - Tempo equal to 0 or greater than 230 bpm
    - Duration greater than 1 million ms (16 minutes and 40 seconds)
    - 87 songs (0.47% of original dataset) removed
  + Columns for key, valence, and tempo were excluded from model
  + Model: Views = 85289684 + 62620690\*Danceability - 52870681\*Energy + 6014042\*Loudness - 47042349\*Speechiness - 15025558\*Acousticness - 11792246\*Instrumentalness + 3002845\*Liveness + 200.46\*Duration\_ms
  + RMSE for this model = 181,045,436, a 21% improvement over baseline RMSE
* **Model for predicting YouTube likes:**
  + Average number of likes for test songs = 623,573
  + RMSE for baseline = 1,751,936
  + Key column was replaced with binary-value dummy columns
  + Outliers removed:
    - Number of likes greater than 20 million
    - Danceability equal to 0
    - Loudness less than -40 dB
    - Tempo equal to 0 or greater than 230 bpm
    - Duration greater than 1 million ms (16 minutes and 40 seconds)
    - 61 songs (0.33% of original dataset) removed
  + Model: Likes = 747926 + 812171\*Danceability - 423204\*Energy + 48601\*Loudness - 151361\*Speechiness - 140015\*Acousticness - 10726\*Instrumentalness - 33322\*Liveness - 371075\*Valence + 566\*Tempo + 0.71\*Duration\_ms + 35962\*Key\_1 - 55237\*Key\_2 + 29870\*Key\_3 + 38565\*Key\_4 + 106844\*Key\_5 + 49626\*Key\_6 - 61126\*Key\_7 + 121332\*Key\_8 - 23064\*Key\_9 + 52703\*Key\_10 + 22334\*Key\_11
  + RMSE for this model = 1,318,429, a 25% improvement over baseline RMSE

**Result of Models:**

The best models I created are not great. They are better than the baseline of simply guessing the average, and I would prefer these models to the baseline. However, the RMSEs are even larger than the respective averages of streams, views, or likes. In practice, I would not use these models to predict how often a song is played based on the qualities used for this project.

**Observations:**

My next strategy was to inspect each attribute individually, instead of all at once. Perhaps a single quality might reveal something. In Python, I charted each metric in a boxplot to see the distribution of songs with each value of that metric. In both Python and Tableau, I charted each metric in a scatterplot against plays to show which metric values produce the highest number of plays.

Note: The scatterplots are in the Jupyter notebooks and the Tableau file.

* **Examples that were not helpful:**
  + Songs with both low and high Spotify streams appear throughout the entire range of valence values (0.0 to 1.0).
  + Except for a few outliers, the tempo of most songs is between 50 and 210 bpm. However, the range of streams also varies greatly.
  + In both cases, the number of streams is not related to the song attribute.
* **Examples that were more helpful:**
  + Songs with high Spotify streams are concentrated where loudness is greater than -10 dB.
  + Songs with high Spotify streams are concentrated where liveness is less than 0.15.
  + Most songs with high Spotify streams have an instrumentalness of 0.0 (or at least less than 0.02).
  + In these cases, songs with high numbers of streams were in a narrow range of a certain attribute.
  + However, those ranges also contain higher concentrations of songs with lower numbers of streams, which effectively cancel out the songs with more plays. Therefore, the models will predict a low number of plays for any song, regardless of its qualities.
* **Good conclusion:** Highly played songs share certain qualities. For example, most popular songs are above a certain loudness and below a certain liveness and a certain instrumentalness.
* **Bad conclusion:** All songs with these qualities will be played a lot. They have a higher chance of a high number of plays, but the chance of low plays is still greater.

**Genre Analysis:**

Two linear regression models were created from songs with known genres.

* RMSE of model that excludes genre = 234,341,760
* RMSE of model that excludes genre = 231,250,408, an improvement of 1.32%

Linear regression models were also made for each set of songs of each genre.

* Compared to the model with all genres, the RMSEs of the genre-specific models range from a 68% improvement (for Original Pilipino Music) to a 98% increase (reggaeton).
* Some genre sets had fewer than 100 songs (less than 1% of the total songs). The RMSEs for these sets might not be accurate.

**Conclusions:**

Unfortunately, I could not accurately predict how many times a song would be played based on its qualities. I determined that most popular songs have certain qualities, but not all songs with these qualities are guaranteed to be popular. Maybe there exists a model that predicts how plays a song will achieve. It might not be worth trying to find, but it would give me an excuse to continue playing with the data. I might even discover something else, such as how the quality metrics relate to each other.

More importantly, songs should not be about sales and popularity. Musicians still need to make money, but their best songs should be about expressing feelings or voicing opinions. Songs can even be just for fun.

This project has taught me things beyond a hypothetical business situation. Through experience, I learned about the process of data analysis. I went through the stages of data analytics workflow: extract, frame, wrangle/prepare, analyze, interpret, and communicate. I used several data analysis tools, including Excel, Python, and Tableau. I learned the need to interpret results carefully. For example, I could conclude that popular songs have certain special qualities, but I could not conclude that a song with these qualities is popular. I originally hoped to discover the latter statement, but even with careful analysis, I might not always find my intended result. I might discover something completely different—or nothing at all. But that two things might not be related, which happened in this project, is still a valid conclusion.

Data analysis plays a vital role in business, but it can do so more, from saving lives to satisfying human curiosity. This project began with financial intentions but became fun way to learn about music.