**Project 3**

Since its founding in 2006 Spotify has become one of the largest audio streaming platforms in the world. With millions of artists and hundreds of millions of songs Spotify has become an essential part of the music industry. Because of this, the ability to predict what songs are most likely to go viral on Spotify, and potentially even create them yourself, is an interesting, and potentially lucrative prospect.

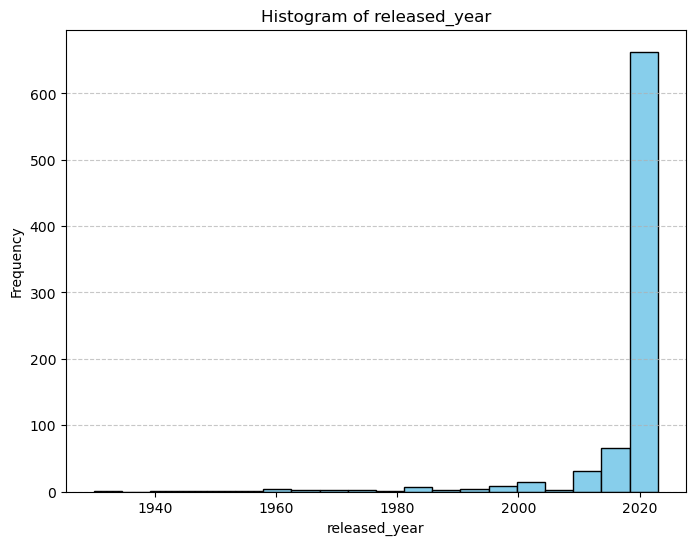
The dataset being used for this project takes the amount of streams for different songs on Spotify, along with many factors about those songs (https://www.kaggle.com/datasets/abdulszz/spotify-most-streamed-songs). The factors included in this dataset are track\_name, artist(s)\_name, artist\_count, released\_year, released\_month, released\_day, in\_spotify\_playlists, in\_spotify\_charts, streams, in\_apple\_playlists, in\_apple\_charts, in\_deezer\_playlists, in\_deezer\_charts, in \_shazam\_charts, bpm (beats per minute), key, mode (major or minor), danceability\_%, valence\_% (the positivity of the song), energy\_%, acoustiness\_%, instrumentalness\_%, liveness\_% (how much of the song is a live performance), speechiness\_% (the amount of spoken words in the song), and cover\_url (a link to the cover of the song). The columns related to playlists and charts show the quantity of playlists and charts which that song appears in. The columns appended with “\_%” measure how much of that attribute is present in the song, on a scale of 0 to 100.

Linear Regression is the attempt to draw the best possible straight line through every graphed datapoint. This line, hopefully, is able to explain the average datapoint accurately. The formula for Linear Regression is y = mx + b. In this equation y represents the predicted value and x the predicted value. The b in the equation represents the intercept, which is the y-value when the created line crosses x = 0. Finally, the m-value is the slope of the equation, which is how much the y-value changes for every increase in the x-value. For example, if the found trend line is y = 6x - 30, and the x-value is known to be 7 then the equation would be y = 6(7) - 30. When simplifying the equation would be y = 42 - 30, so in this example y equals 12.

In order to get a full understanding of the individual columns within the data a function was created which looped through every column, creating a histogram for any numeric ones. Many of the columns had typical outputs, though some columns had notable skews. For instance, the “released\_year” column showed a significant skew toward newer movies, as can be seen in Figure 1, likely showing the bias of the Spotify users. This does not mean that this data is

**Figure 1**

*Histogram of released\_year*



unusable, but it should be remembered in the future as a possible datapoint to be removed due to its skew. A created correlation matrix showed that many of the columns had little correlation with each other. Interestingly, the highest correlation was acousticness to energy, where there was a -0.55 correlation. Also notable was valence’s 0.39 correlation with danceability, and energy’s 0.35 correlation with valence.

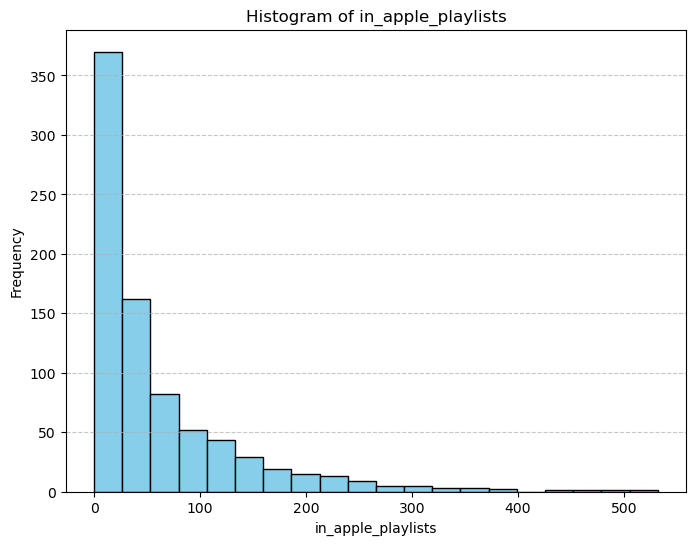
The first step in data preprocessing was to check for null values. In total, only two columns had any null values, with in\_shazam\_charts having 50 nulls, and key having 95 nulls. Removing these nulls left a much more manageable dataset. The next check of the dataset was to check the variable types of each column, which showed something odd. The column “streams” was an object type, even though it should be a numeric type. When sorting the column to check for outliers the problem was spotted. For some reason, one of the values was “BPM110KeyAModeMajorDanceability53Valence75Energy69Acousticness7Instrumentalness0Liveness17Speechiness3”, which was presumably an error. Interestingly, two other columns also were errantly object types, specifically “in\_deezer\_playlists” and “in\_shazam\_playlists”, however they had no discernable data error. In order to rectify these problems any non-numeric data points within these three columns were transformed into nulls, and nulls dropped from the dataset. It was chosen that for the first attempt at modeling the columns related to charts and playlist totals would be excluded from the dataset, in order to simulate a more realistic dataset which an average person could create. “track\_name” and “cover\_url” were also dropped due to low predicting ability. Additionally, all columns were renamed so that they would be more easily readable. In order to more easily use the month column the months column was changed from being categorical to being one-hot encoded. Finally, the “artist” column was split, and one-hot encoded, so that the presence or lack of every artist was its own column, though artists with less than 10 songs in the dataset were not included.

A Linear Regression was created to be used on this dataset, with a test size of 0.2 and a random state of 42. This meant that eighty percent of the dataset was used to train the model, and twenty percent to test the prediction of the model. The results of the model were not fantastic. The Mean Squared Error (MSE) of the model was found to be 337166521978547968 and the R-Squared value to be 0.098. This means that the model was not doing a very good job predicting the streams of songs. Such a high MSE value shows that there is a ton of variance between the predicted values and the actual ones. Similarly, an R-Squared value of 0.098 means that the model was able to predict less than ten percent of the variance within the data. In order to further investigate the data a correlation model was made specifically to test the correlation with “streams”. This correlation model showed that only two columns had greater than 0.2 correlation with “streams”, those being year with -0.2 correlation and “artist Ed Sheeran” with a correlation of -0.21. What this showed was that more effective variables for predicting were needed.

In an attempt to improve the model the previously removed charts and playlist data was reintroduced into the dataset. While this data might be difficult for a random user to obtain, clearly it does exist, and the more accessible data did not have enough impactful predicting power. When investigating these new columns they were shown to generally skew left, with the top songs taking a large portion of the total streams or playlists. An example of this skew can be seen in Figure 2, with the in\_apple\_playlists column. These skews may prove to lessen the predicting power of these variables. The correlation matrix showed that the newly introduced columns did not tend to have a high correlation with the existing columns, though they were highly correlated with each other.

**Figure 1**

*Histogram of in\_apple\_playlists*



The preprocessing steps needed for the second model were the same as the first, with nulls being dropped, “months” being remapped to be categorical, and “artists” being split and one-hot encoded. With these steps completed the model could be created, using the same settings as before. This second model had notably better metrics, with a MSE of 149635997685452608 and an R-Squared value of 0.145. So, while these metrics were certainly an improvement, the model was still unable to predict even fifteen percent of the variance within the dataset.

The final test for the dataset was to remove the many columns which were highly uncorrelated with “streams”. The hope was that by removing the unhelpful data, which could potentially be bloating the model and causing overfitting, the model would actually improve. Therefore, a new dataframe was created which only included columns that had an absolute correlation with “streams” of 0.1 or higher. These columns ended up being "acousticness\_%", "bpm", "danceability\_%", "energy\_%", "in\_shazam\_charts", "instrumentalness\_%", "liveness\_%", "released\_day", "released\_month", "speechiness\_%", "streams", "valence\_%". The preprocessing for this dataframe was limited. Month was remapped to be categorical, 4 nulls were dropped from the “in\_shazam\_charts” column, and Month was one-hot encoded.

With the preprocessing complete a model could be created. Once again, twenty percent of the dataset was used for testing and eighty percent for training. The final metrics for this model were very disappointing. Contrary to my theory, removing the seemingly superfluous data considerably harmed the data resulting, in a MSE of 346012191136056832 and an R-Squared value of -0.044. These metrics showed that the model was fundamentally flawed, possibly due to a lack of quality data, or else biases in the data.

When testing all three models model two was clearly the best. Unfortunately a model which effectively predicted song popularity was not able to be created. Likely this is due to the difficulty in creating, or in this case predicting, a hit song. Ultimately, there is not a mathematical formula for creating the perfect song. As with any art form, taste is subjective, so while some things may generally appeal to a crowd, attempting to artificially manufacture the perfect song will likely always end in failure. Therefore, perhaps the most positive impact this study could have would be to remind artists not to attempt to engineer songs based on their likelihood for popularity, but instead based on their own desire to make creative music. (Or, if they really want tips based on the data they should create music which is not positive, not filled with spoken word sections, and sung by Bad Bunny) A potential negative aspect to this study would be if aspiring artists became discouraged by the daunting task of creating a popular song, and how random it may seem to be.

In conclusion, what makes music popular is extremely varied and unpredictable. While I had hoped that there would be some sort of datapoint which could predict popular songs, that datapoint does not seem to exist, at least in this datapoint. If I were to continue this study I would likely look to expand the dataset with new data, such as the length or tempo of the song. Additionally, more objective data points could potentially be useful, as the more objective data points surrounding playlists and charts seemed to increase the predicting power of the model. However, I do ultimately suspect that a very effective model cannot actually be created for this sort of study, due to the subjectivity in music taste and luck required in getting a hit song.

**References**

https://www.kaggle.com/datasets/abdulszz/spotify-most-streamed-songs

https://chatgpt.com/

Note: ChatGPT was used in this assignment for specific code segments. Specifically, to assist in creating the models and graphics

**Link to Code**

https://github.com/bigbadraj/Project-3---Spotify-Modeling