PAC Report

Harrison Dvoor

**Introduction**

Upon first view of the data, I didn’t think this project was going to be too difficult. I figured I could merely explore the raw dataset and use some of the correlation techniques we’ve learned in class to analyze what variables had strong correlations with the rating variable and subsequently use them in my model as predictors.

However, I was terribly mistaken as there was a lot more that went into this project. To be successful in this competition required not only our modeling skills that we learned in class, but our data cleaning/wrangling techniques as well as our data plotting/charting skills to visualize some of our findings and analysis.

**Data Cleaning and Wrangling Routines**

The data cleaning and wrangling routines ultimately consumed the majority of my time and effort on this project. For my first model’s submission, I attempted to not clean or transform any of the provided dataset, and my RMSE was extremely large. Thus, I knew the data had to be tidied in order to score better.

The first major issue I noticed with the original dataset is that the genre column contained a list of strings, with multiple genres in a list in some cases. This is not a format in which you can effectively analyze and predict what genres resulted in higher song ratings. That being said, I determined that I wanted my dataset to include a column for each unique genre that is listed in any of the rows of the original dataset, with each row containing a 0 or 1 depending on whether or not the song falls into that genre. For example, Taylor Swift’s “Ready For It?” will list 1’s for the pop and post-teen pop genres, with zeros for all other genres.

The first step in creating this was to remove the brackets ( [] ) and quotes ( ‘’ ) from each value in the genre column. I used the *gsub* function with regular expressions to do so. Now that special characters have been removed from the genre column, I used *separate\_rows* to separate each genre in each row such that each row contains a unique combination of ID and genre. For example, id = 48848 and genre = pop in one row, and id = 48848 and genre = post-teen pop in another row. My next step was to create a new column that shows whether or not the genre was present. This column will be utilized when I perform the pivot. Before conducting the pivot wider, I needed to ensure that I classify the rows with blank values for genre. For any of these rows, I set the value of genre equal to “NoGenre”.

Next, I conducted the *pivot\_wider*. I used the cleaned genre column as the names for the pivot, and the column showing whether or not the specific genre was present as the values for the pivot. I also formatted the pivot to fill any empty values as 0. This resulted in a new table with the same number of rows (19,485), but now there were over 1,000 columns as each unique genre listed in the original genre column now has its own column. This was an overwhelming amount of columns that were created, making it hard to determine correlation between rating and 1,000 different genres. One step I took was to count the genre frequencies to see which genres were the most popular. I felt that doing this would help narrow down which of the genres I wanted to include in my predictive model. I grouped the dataset by genre and counted the frequency of each genre, arranging the count in descending order.

I now believed that the data was ready to be analyzed, but I soon realized there were a few more data cleansing techniques that I needed to follow. First, I noticed that while the genre names were stripped of the special characters, there were still spaces between the genres with multiple words (i.e. adult standards). Thus, I needed to use *gsub* function once more on the genre column, this time removing any whitespace in the string. Additionally, I noticed that the *track\_explicit* column was of the logical type (TRUE/FALSE). In order to effectively implement this variable into my model, I transformed it into an integer/double type, made up of 1’s for TRUE values and 0’s for FALSE values.

It's also worth noting that I needed to apply all of the above data cleansing and transformation techniques to the Scoring dataset as well so that I could properly submit my model for grading.

**Charting and Plotting Routines**

Throughout the competition, I used a few different charts and plots to visualize the data at hand to help support some of the decisions I made. I regret not using more, but I felt that the ones that I did use were very effective in leading me in the right direction regarding which variables in the dataset to use as predictors.

First, I created a correlation plot with the variables in the original dataset. I created this plot prior to cleaning or transforming any of the data as I wanted to get an idea of which variables have strong and/or weak relationships with one another. As you can see in Figure 1 (see appendix), most of the relationships between these variables are relatively weak, with a few exceptions: *acousticness* and *energy* have a strong inverse relationship, and *loudness* and *energy* have a strong positive relationship. Ultimately, I knew that the data needed to be both cleaned and transformed in order to get a better idea of which variables can be utilized as effective predictors for my model.

I also decided to examine the relevance and redundancy of all the original predictors together. The reason for this would be that what is true in pair-wise relationships is not always the same when the predictors are all considered together. To see if this is the case for the variables in the original dataset, I wanted to test the statistical significance of regression coefficients and variance inflating factors (VIF). As you can see in Figure 2 (Appendix), there are no VIF’s greater than 5. Thus, since 5<VIF<10 warrants examination for multicollinearity, and VIF>10 indicates serious multicollinearity, I proceeded without any threat of multicollinearity.

The last data visualization that I created came directly from one of the lectures. This was for forward selection. I conducted forward selection using a handful of the variables from the original dataset, in addition to the top 25 genres based on frequency. As shown in Figure 3 (see Appendix), variables are being added successively until margin improvement in AIC is no longer significant. While I was able to successfully run the forward selection, the model did not perform as well as I had hoped it would. Therefore, I decided to abandon this selection method.

**Model Analysis Routines**

My model analysis routine was not overly complicated. I merely attempted using several different types of models that we had covered in class, including linear regression, generalized linear regression, random forest models and ranger models.

To prepare to run my model, the first step was to set the seed and then split the data into train and test. I split it with *p=0.7* and *groups=100*.

The first submission I made was without any data cleansing or transformation. I just wanted to get a feel for what the RMSE would look like, and thus hand-picked the variables to include using my intuition. In this model, I included *danceability, energy, acousticness, liveness,* and *tempo*. This model did not perform well, yielding a test RMSE of 16.03. I didn’t expect this model to perform very well as I put no effort into cleaning the data and minimal effort into analyzing what predictors to include. I also tried generalized linear regression, which didn’t perform any better. I expected this as well, but given the amount of submission attempts I had, I figured it was worth a shot.

After cleaning the data using all of the techniques mentioned in the “Data Cleaning and Wrangling Routines” section, I now tried to run a random forest model. One question I had a little difficulty answering was how many of the genre dummy variables I should include in my final model. I decided to test different amounts in my models. In each model I ran, I included some of the variables from the original dataset, including *track\_duration, track\_explicit, danceability, energy, loudness, acousticness, instrumentalness, tempo*, and *time\_signature*. I then used the analysis from my technique of counting the frequency of each genre in the dataset. I tried using the top 10 in random forest models, top 25, top 50, top 100, top 200 and top 300. I’m not entirely sure why, but the best RMSE was yielded from the random forest model with the variables mentioned above, in addition to the top 200 genre dummy variables. This yielded a test RMSE of 14.70648. While this was significantly better than the linear regression submissions, I knew that there was room for improvement.

I then decided to run a ranger model. Again, I decided to use *track\_duration, track\_explicit, danceability, energy, loudness, acousticness, instrumentalness, tempo*, and *time\_signature* from the original dataset, in addition to the top 10, top 25, top 50, top 100, top 200, and top 300 genres based on frequency. I decided to mess around with the number of trees in the model and found that 2000 yielded my best test RMSE. Once again, the top 200 genres yielded the best RMSE, which was 10.85389 for the train data, 14.58802 for the test data, and 14.51275 for the scoring data. This model was my best scoring model overall.

**Conclusion**

All things considered, I thought that this competition was a really good learning experience for me. This was the first time I’ve ever participated in a coding competition, and I thought it was super motivating to see how other classmates are scoring on the leaderboard as it pushed me to keep trying to enhance my model and improve my score.

My main takeaway from this project was that data cleansing is without a doubt the most important part of data modeling. If you’re data is not clean, there is no chance that your model will be as effective. I also learned that it’s ok to clean your data in iterations - make one change, see how it affects your model and go back and continue to clean and transform the data until you like what you see.

I also realized that I probably should’ve tried using more types of models (i.e. xgboost) instead of continuously trying to enhance my random forest and ranger models. Furthermore, I noticed towards the end of the competition that I never considered the *performer* variable. This was foolish on my end as popular performers more than likely have songs with higher ratings. I definitely should’ve transformed this variable in a way that made it an effective predictor of rating, as I did with the *genre* variable.

Overall, this project was a lot of fun and really taught me a lot about data modeling. I’m hoping that this will help prepare me for Frameworks and Methods II as well as any other future coding/data modeling courses.

**Appendix**

*Figure 1*

Chart

Description automatically generated

*Figure 2*

Chart, funnel chart

Description automatically generated

*Figure 3*

Chart

Description automatically generated