RIDGE REGRESSION FOR PREDICTING TRACK POPULARITY

DARYA SAVITSKAYA

ABSTRACT. This project employs ridge regression to predict the popularity of tracks on Spotify, with a dataset containing both numerical and categorical features. Initially, a model relying solely on numerical features is developed, followed by a more complex model that includes encoded categorical variables. The performance of both models is evaluated, revealing that the inclusion of categorical data slightly improves the variance explained in track popularity. The results underscore the potential of ridge regression in managing multicollinearity and feature selection, with the enhanced model showing a modest improvement in predictive accuracy.

Contents

1. Introduction

, The goal of this project is to use ridge regression—a statistical technique that fine-tunes predictions by balancing detail with simplicity—to understand what drives a track's popularity on Spotify. Ridge regression is particularly useful when we're juggling many variables that could influence a song's success. It helps us avoid being misled by the noise in the data. This method adjusts for variables that move together, like the volume and energy of a track, ensuring that our predictions don't rely too heavily on redundant information.

The path of this project is as follows: first, to apply ridge regression using only the numeric features of the tracks, and second, to weave in the categorical features and assess their impact on our predictions. By comparing these two approaches, we seek to illuminate the value added by each type of data.

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. We understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

2. Data Description

2.1. **Dataset Overview.** Spotify dataset contains information on 114000 songs and their various characteristics. Initially, there are 20 variables, some are straight forward pieces of information, such as Track ID, Artists, Album name, Track name, Duration etc., while others required calculation, in some cases by an algorithm, such as, among others, danceability, energy, valence and, the variable of interest for this paper, popularity, that was calculated by an algorithm and takes into account the total number of plays the track has had and how recent those plays are. Full list can be found at the Table 1. The number of null values is clearly insignificant.

Table 1. Variables Information

Variable Name	**Non-null Count**	**Data Type**
track_id	114000	object
artists	113999	object
album_name	113999	object
track_name	113999	object
popularity	114000	int64
duration_ms	114000	int64
explicit	114000	bool
danceability	114000	float64
energy	114000	float64
key	114000	int64
loudness	114000	float64
mode	114000	int64
speechiness	114000	float64
acousticness	114000	float64
instrumentalness	114000	float64
liveness	114000	float64
valence	114000	float64
tempo	114000	float64
time_signature	114000	int64
track_genre	114000	object

The whole dataset contained duplicates which make the data biased towards repeated songs and make data encoding difficult in the future. It was decided to remove duplicates from the whole dataset before train and validation split in order to make the whole dataset more clean. As a result, 450 rows were removed.

The dataset was divided into test and training set (0.2), training set was devided into training and validation with the same proportion.

2.2. **Correlation Analysis and Multicollinearity check.** In the figure below we can see the correlations between independent variables that are higher then 0.7.

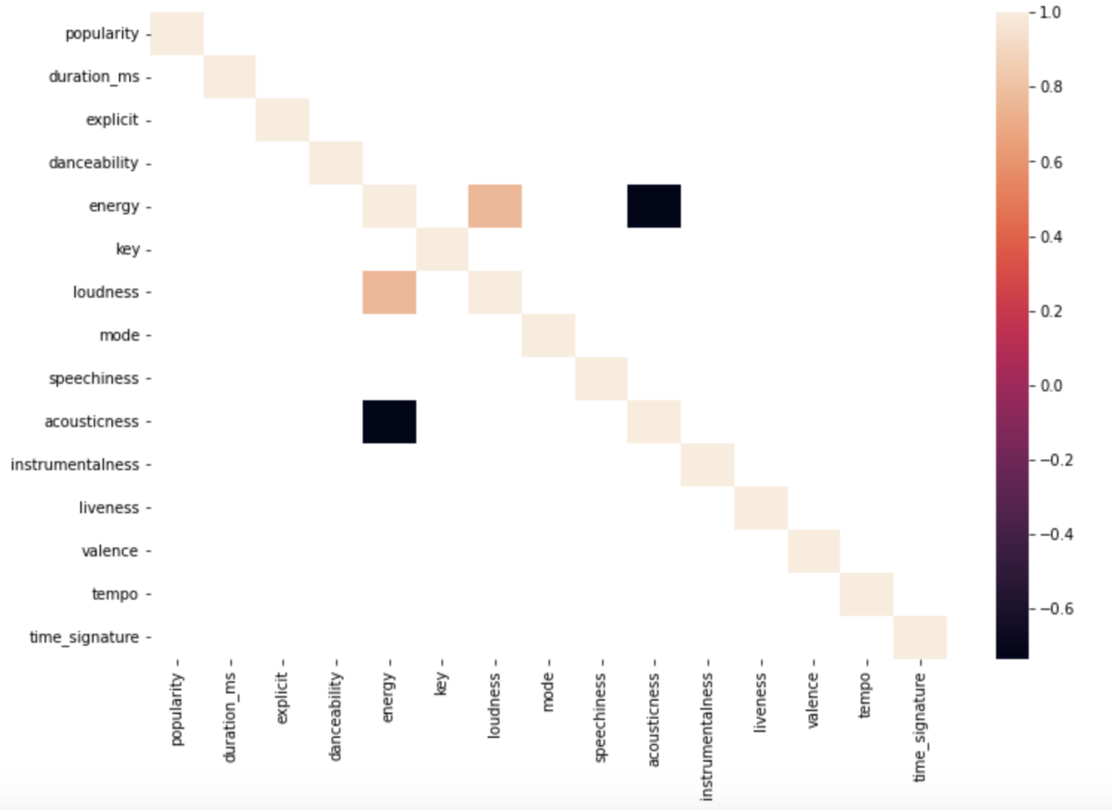

Figure 1. Correlation

It can be seen that energy is positively correlated with loudness and negatively correlated with acousticness, which makes perfect sense as loud music tends to be more energetic while acoustic music is usually more mellow and slow. Indeed, the VIF factor for energy is 4.2, the highest above all variables, followed by 3.2 (loudness) and 2.4 (acousticness). Still, IVF below five are generally considered a sign of moderate multicollinearity, given that ridge regression used in this model is also helpful against correlated variables, no further measures had been taken.

2.3. **Outliers Check.** I have decided to use z score method to locate the outliers. i calculated the z-score for every variable, picked the maximum value for every row and then eliminated the rows with maximum absolute value of the z-score above 3.5. I ended up shrinking the dataset from 57432 rows to 54076, eliminating 3356 rows.

2.4. **Data Preprocessing.** While many variables have a standard scale from 0 to 1 (danceability, speechiness, acousticness, instrumentalness, liveness, valence), some stand out, like tempo with range from 0 to 243 and loudness with range from

-50 to 5. As in ridge regression the penalty term is affected by the range of the variables, the variables were standardized.

2.5. **Handling Categorical Features.** To widen variable selection, it is possible to include some categorical variables, such as track_id, artists, album_name, track_name, track_genre and explicit as well, which is a binomial variable and was not used in purely numerical analysis. The number of non unique values was calculated in order to pick the best method of encoding.

Table 2. Number of unique values of categorical variables

Variable Name	**Unique count**
artists	27840
album_name	40600
track_name	62077
track_genre	114

It can be seen that track_genre has the lowest number of unique values. It might be a valid hypothesis that track genre affects the chances of a song to be popular (taking in account also that one genre is called 'pop', i.e. popular). Therefore, I chose this variable as one to use one-hot encoding on. I have noticed a particularity about this dataset that required some action: if a track has more then one genre, then it will also have more then one row in the dataset - new row for every genre. This could bias the data, so I decided to deal with this in the following way: 1. To keep the data consistent I performed encoding of the categorical variable early in the analysis 2. I grouped the rows by all the variables that were not dummy variables with the goal of summing up the dummy variables such that every song has one row but possible more than one dummy variable. I did this even before running the model on only numerical dataset in order to avoid complication in removing duplicates and keep numerical and full datasets consistent with each other.

To deal with possible multicollinearity within dummy variables, I hypothesized that some genres will be less frequent than the other and could be grouped into the 'other' variable - that, however, surprisingly was not the case, as the range of the unique count of the genre was [776, 832]. Therefore, I have used chi-square test to select 20 dummy variables most influencial on the popularity variable.

While dealing with other variables, it was decided to not use track name or album name, as these variables were hypothesised to not add significant information and be more closer to the identifier kind of variable such as, for example, track id. To deal with artists name, I decided to use frequency encoding: each song was given a value from 0 to 1, calculated as the number of this particular artists divided by the total number of artists. Th hypothesis is that more successful artists are likely to have more songs, rather than less successful ones.

2.6. **Target Variable.** Popularity variable is a positive variable that is not normally distributed. With a range of [0, 100] where 100 is the most popular, it has mean equal to 33.2 and standard deviation equal to 20.58.

3. Implementation

3.1. Theoretical comments.

3.1.1. *Ridge regression.* To understand better what are the advantages of ridge regression and what are its mechanics, suppose, we are starting with least squares regression in 2D - ultimately we end up with a line that results in the minimum sum of squared residuals. The line is supposed to reflect the relationship between two variables because it tries to minimise the distance between itself and every point it is given for the calculation. Ridge regression introduces a small amount of bias in order to prevent overfitting and minimise test error. To do that, the model introduces a penalty - lambda multiplied by squared slope (the difference between two neighbouring predicted points), that makes independent variable a bit less sensitive to the dependent. This reduced sensitivity to the training data makes this model a good choice for data with some problems: data that suffers from some multicollinearity benefits from the added penalty as it makes the coefficients affected less by the correlated variables that may cause high variance; high-dimensional data is a great call for ridge regression that shrinks coefficients of the less influencial variables performing a sort of feature selection process; some scale differences can also be handled by ridge regression penalty shrinkage - this, however, can be easily combatted by standartization. The data we are working with here suffers from some mild multicollinearity, as well as could be considered high-dimentional, especially if categorical variables are one-hot encoded and feature selection is not performed. Theoretically, the model that includes categorical variables should perform better as a result of model's selection of the right variables

3.1.2. *Cross Validation.* To fully appreciate the role and benefits of 5-fold cross-validation, imagine we're conducting an experiment where our aim is not only to gauge the current performance of a predictive model, like ridge regression, but also to estimate how well it would perform on unseen data. This is where cross-validation shines as an assessment tool, especially 5-fold cross-validation. Think of 5-fold cross-validation as a method of reliability testing. Instead of relying on a single split of training and testing data, we methodically shuffle and partition the entire dataset into five distinct 'folds'. In each round, one fold is reserved as the test set while the remaining four folds are combined to form the training set. This process repeats five times, cycling through each fold as the test set exactly once. The model is trained and evaluated in each iteration, yielding five separate performance scores. Additionally, cross validation does not require validation set, as it simulates the unseen data using only training data. In my analysis, validation set was only used to test the model performance after picking the best performing parameter.

4. Data Analysis

4.1. **Model specification 1: numerical variables.** First, the model was run using only numerical variables: 'duration_ms', 'danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo', 'time_signature'. Taking the base formula for ridge regression, the following formula was used.

```
def ridge_r(X,y,lambda_p):
    n = X.shape[1]
    I = np.eye(n)
    I[0,0] = 0
    X_np = X.to_numpy()
    y_np = y.to_numpy()


    beta = np.linalg.inv(X_np.T @ X_np + lambda_p * I) @ X_np.T @ y_np
    return beta
```

Lambda was chosen using the 5 fold cross validation, the risk measure I looked at was RMSE as I find it very nicely interpretable. From the list of lambdas equal to [0, 1 , 10 ,100, 1000], the model performed best with 100, but the differences were minimal. Using lambda equal to 100, the results were as follows: on average, the predictions made by the model have an error of approximately 20 units, which is close to the standard deviation of the target variable, which shows that the model does not outperform a baseline model, the proportion of the explained variance by the independent variables was even worse - around 2%. In the scatter plot, plotting validation popularity and predicted popularity, we can see that the model does the worst for the biggest values of the target variable, closer to 10, because the range of predicted values [0, 45] is small compared to the range of the target variable [0, 100].

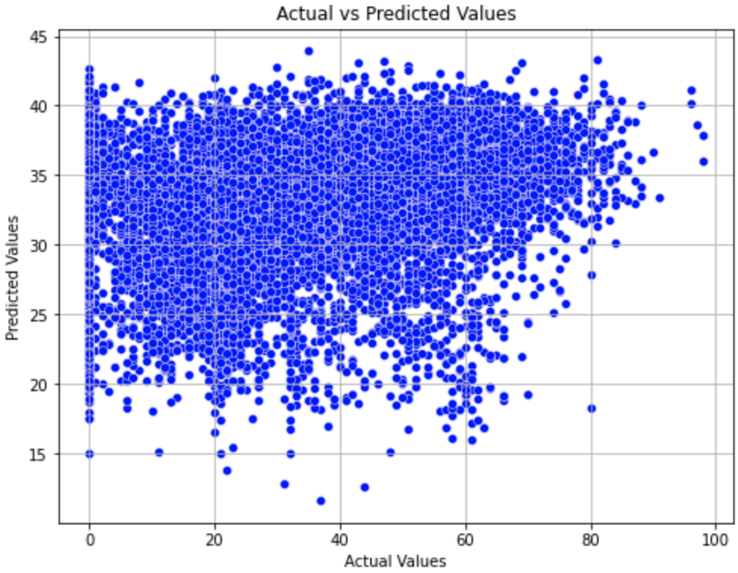

Figure 2. Scatter Plot

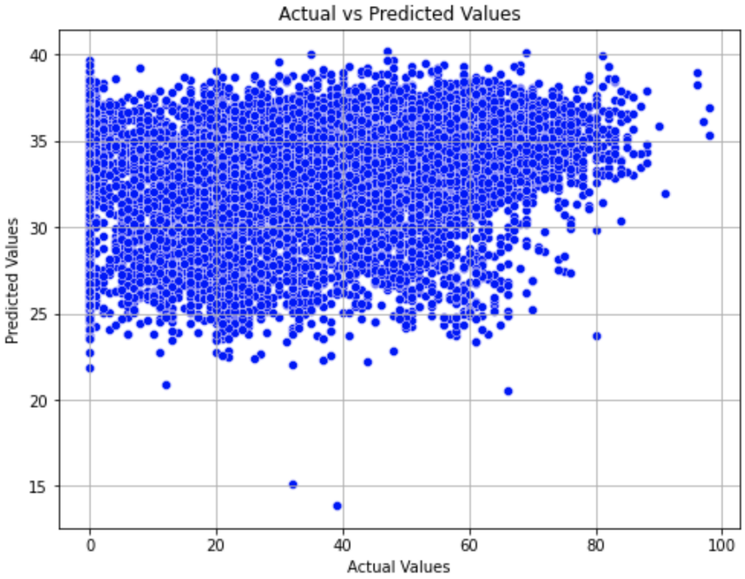

Figure 3. Scatter Plot Including outliers

An approach was made to train a model using the data with all points, including previously discarded outliers and a high lambda = 10000. Surprisingly, the range has decreased and the smaller values of target variable were predicted less accurately.

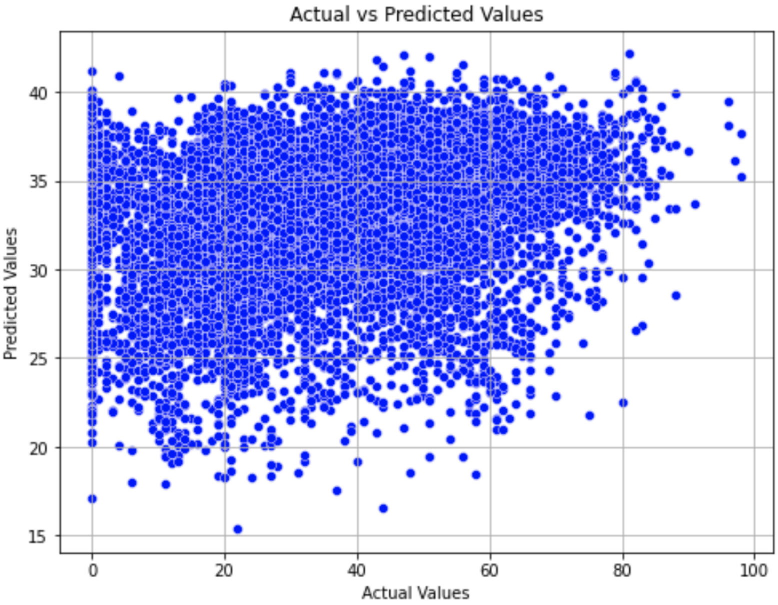

FIGURE 4. Scatter Plot, spec 2

4.2. **Model specification 2: numerical and categorical variables.** This time additional variables are included: 20 dummy variables for genres (picked by the test described previously) and a new variable describing artists frequency. Indeed, this model specification introduces more noise. The results of the cross validation show that out of the following parameters [1,10,100,1000,10000], the best performing is 10000, but the differences are also small. The percentage of the explained variance rises, but ever so slightly - now it's 4%. From a short comparison of the residual histograms for both models we see make some conclusions. Both histograms appear to have their peaks around zero, suggesting that both models, on average, make unbiased predictions. The first graph shows a slight right-skewness, indicating a tendency to under predict, while the second one also exhibits right-skewness, it seems slightly less pronounced than the first one. This might suggest that the second model is a bit more accurate in its predictions or that the additional variables help to explain some of the variance in the track popularity. Additionally, the second histogram is flatter with less steep tails, suggesting a distribution that more closely approximates normality, indicating that the extended model may be capturing more nuances in the data. Based on these observations, the model with additional variables appears to provide a better fit to the data, as evidenced by its higher R^2 value, less skewness, and a tighter spread of residuals, suggesting fewer and less extreme outliers. It's also worth noting that while both models show right-skewness, implying that both tend to underpredict to some degree, the effect is less in the extended model.

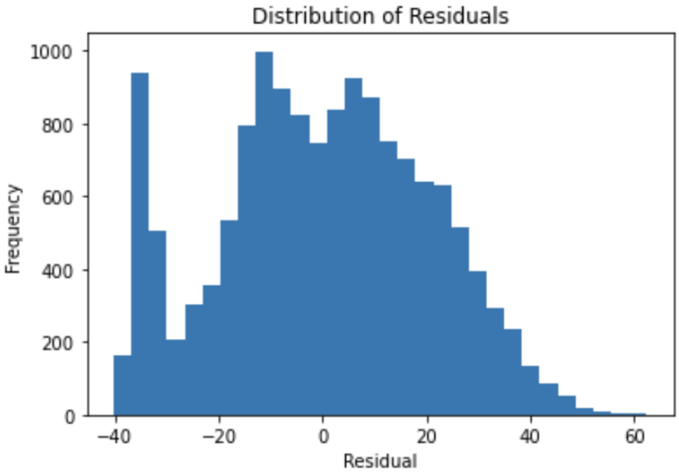

Figure 5. Distribution of the residuals, spec 1

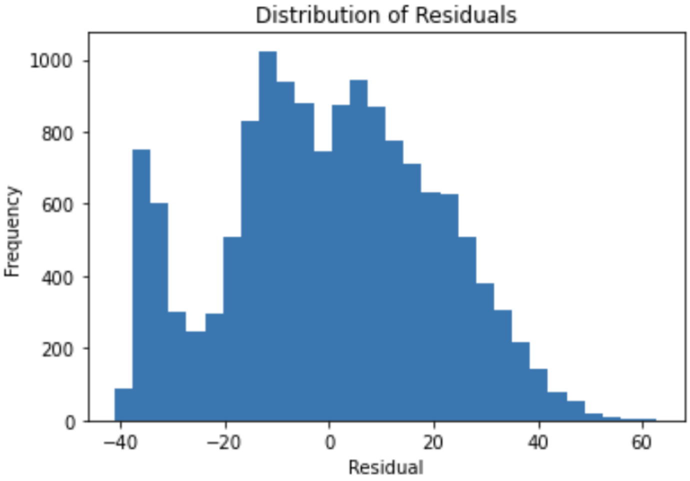

Figure 6. Distribution of the residuals, spec 2

5. Conclusion

This project set out to determine how well we could predict the popularity of Spotify tracks using a mathematical method known as ridge regression. We looked at two different models: the first used just basic numerical data about the tracks, while the second also considered the genres of the songs and how often the artists showed up on Spotify. Findings indicate that when we add these extra bits about genres and artists, the model does a slightly better job of explaining what makes a track popular—up by 1% compared to the simpler model. However, both models tended to underestimate the popularity scores, particularly for the most popular tracks, hinting that there are more pieces to this puzzle we haven't found yet.