Machine Learning Project

Ridge Regression

Spotify Music Data Set

Araz Asgharieh Ahari

2023 – 2024

Introduction

Predicting the popularity of music tracks can offer valuable insights for artists, producers, and streaming services. Spotify, one of the leading music streaming platforms, provides various metrics that can help gauge the potential success of a song. By leveraging machine learning techniques, we can analyze these metrics to predict a song's popularity.

Ridge Regression, a linear algorithm, is used to predict the target variable in the dataset, which, in this case, is the popularity of songs. The dataset comprises both numerical variables related to the technical aspects of songs and categorical variables.

After an initial phase of data preprocessing and exploration, the algorithm was tested on the dataset using only the numerical variables. Following this, the algorithm was applied to the entire dataset, incorporating encoding techniques for the categorical variables.

What is Ridge Regression?

Ridge Regression is a type of linear regression that adds a penalty to the model's complexity to prevent overfitting. In standard linear regression, we simply try to find the best fit line that minimizes the difference between the predicted and actual values. However, this can sometimes lead to overfitting, especially when the model becomes too complex or when the independent variables are highly correlated.

To address this, Ridge Regression includes a penalty term that discourages the model from having overly large coefficients. This penalty helps to keep the model simpler and more generalizable to new data. The key idea is to strike a balance between fitting the training data well and keeping the model simple to ensure it performs well on unseen data.

Key Points

- **Regularization Parameter**: Ridge Regression introduces a parameter that controls the strength of the penalty on the coefficients. A higher value means a stronger penalty, which can prevent the model from overfitting but may also make it less flexible.
- **Bias-Variance Trade-off**: The penalty helps to manage the trade-off between bias (error due to overly simplistic models) and variance (error due to overly complex models).

- **Handling Multicollinearity**: Ridge Regression is particularly effective when independent variables are highly correlated. It helps to stabilize the estimates of the coefficients.

Advantages

- **Prevents Overfitting**: By penalizing large coefficients, Ridge Regression reduces the risk of overfitting the training data.
- **Handles Multicollinearity**: Provides more reliable estimates when independent variables are highly correlated.
- **Improves Predictive Performance**: Often leads to better predictions by balancing model complexity and fit.

Disadvantages

- **Interpretability**: The coefficients in Ridge Regression are biased, which can make them harder to interpret compared to standard linear regression.
- **Choosing the Penalty Parameter**: The performance of Ridge Regression heavily depends on selecting the right penalty parameter, which often requires cross-validation.

About Dataset

Content

This is a dataset of Spotify tracks over a range of **125** different genres. Each track has some audio features associated with it. The data is in CSV format which is tabular and can be loaded quickly.

Usage

The dataset can be used for:

- Building a **Recommendation System** based on some user input or preference
- **Classification** purposes based on audio features and available genres
- Any other application that you can think of. Feel free to discuss!

Column Description

- **track_id**: The Spotify ID for the track

- **artists**: The artists' names who performed the track. If there is more than one artist, they are separated by a ;
- **album_name**: The album name in which the track appears
- **track_name**: Name of the track
- **popularity**: **The popularity of a track is a value between 0 and 100, with 100 being the most popular**. The popularity is calculated by algorithm and is based, in the most part, on the total number of plays the track has had and how recent those plays are. Generally speaking, songs that are being played a lot now will have a higher popularity than songs that were played a lot in the past. Duplicate tracks (e.g. the same track from a single and an album) are rated independently. Artist and album popularity is derived mathematically from track popularity.
- **duration_ms**: The track length in milliseconds
- **explicit**: Whether or not the track has explicit lyrics (true = yes it does; false = no it does not OR unknown)
- **danceability**: Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable
- **energy**: Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale
- **key**: The key the track is in. Integers map to pitches using standard Pitch Class notation. E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on. If no key was detected, the value is -1
- **loudness**: The overall loudness of a track in decibels (dB)
- **mode**: Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0
- **speechiness**: Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks
- **acousticness**: A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic
- **instrumentalness**: Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal". The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content

- **liveness**: Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live
- **valence**: A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry)
- **tempo**: The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration
- **time_signature**: An estimated time signature. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure). The time signature ranges from 3 to 7 indicating time signatures of 3/4, to 7/4.
- **track_genre**: The genre in which the track belongs

Data preprocessing

The first step was to check for any NaN values. Fortunately, there was only one, belonging to an observation lacking both the name and artist information, which could be simply removed. Furthermore, the ”Unnamed: 0” column, a repetition of the dataset indices, was also removed.

Following that, while examining the statistics of the variables, it was ob- served that the unique number of tracks (’track id’) did not match the number of observations. Specifically, there were only 89,740 unique tracks out of 113,999.

Delving deeper into this issue revealed that some tracks were associated with multiple genres. This was confirmed by identifying duplicated values of ’track id’, which had the same values for all features except for the ’track genre’ column. The presence of multiple genres for a single track could lead to biasesin the algorithm’s results if these tracks were randomly assigned to both the training and test sets.

One possible solution was to create a Dataframe with only ’track id’ and ’track genre’ one-hot encoded, then group by ’track id’ and sum the dummy variables for each track. Subsequently, another Dataframe was created with all the variables, except for ’track genre’, and the duplicated ’track id’ rows were dropped.

Finally, the two Dataframes were merged to obtain a clean dataset with unique track values and one column one-hot encoded for each track.

Categorical Variables

Encoding categorical variables is an important step before running a machine learning algorithm. In the case of the Spotify dataset, there are 5 categorical variables, which have been encoded as follows:

Binary encoding for ’explicit’ (0 = False; 1 = True). This type of encoding was chosen because it’s a boolean variable that takes on only two values: False and True.

One-Hot encoding for ’track genre’, as anticipated in subsection 2.2. Since there are few unique values for this variable, One-Hot encoding is appro- priate: it creates a column for each unique value, where 0 indicates” the track does not have that genre” and 1 indicates” the track has that genre”.

Target and Leave-One-Out encoding for ’artists’, ’album name’, and ’track name’.

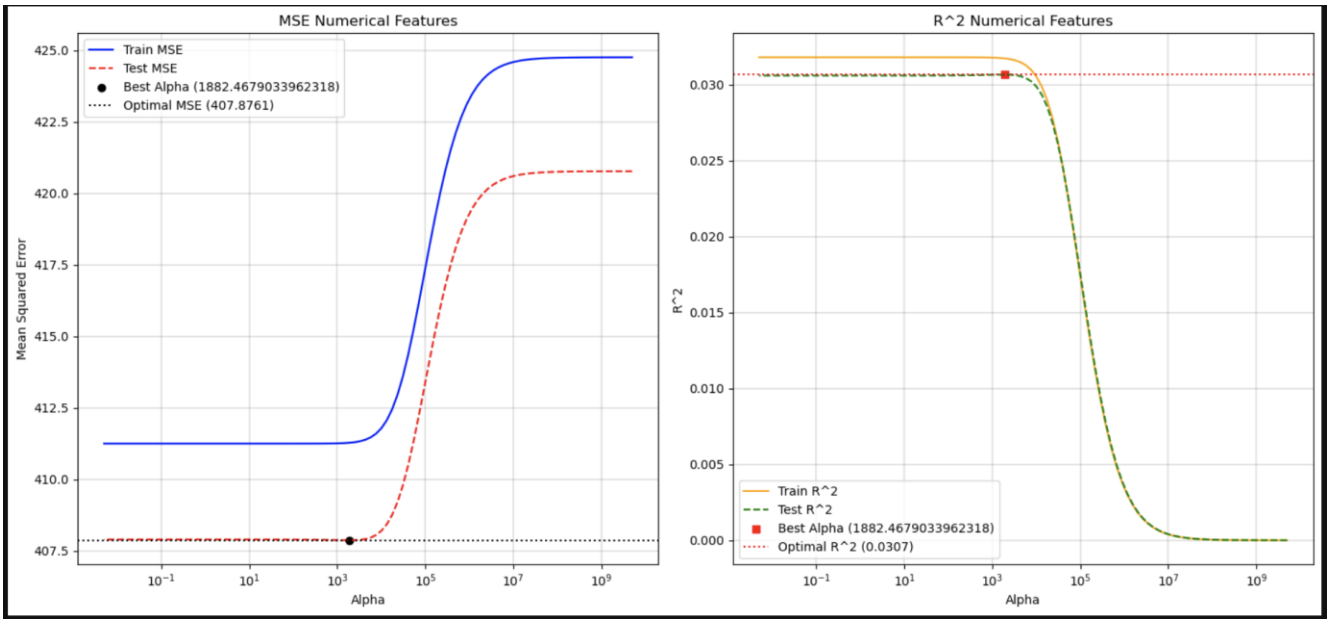
First Analysis: numerical

Training MSE = 411.255

Test MSE = 407.903

Training R*2 = 0.031

Test R*2 = 0.030



The small difference between the training and test error suggests that over-fitting was avoided. However, the R2 values are quite low, indicating that the numerical variables explain only 3.1% of the variance in the target variable.

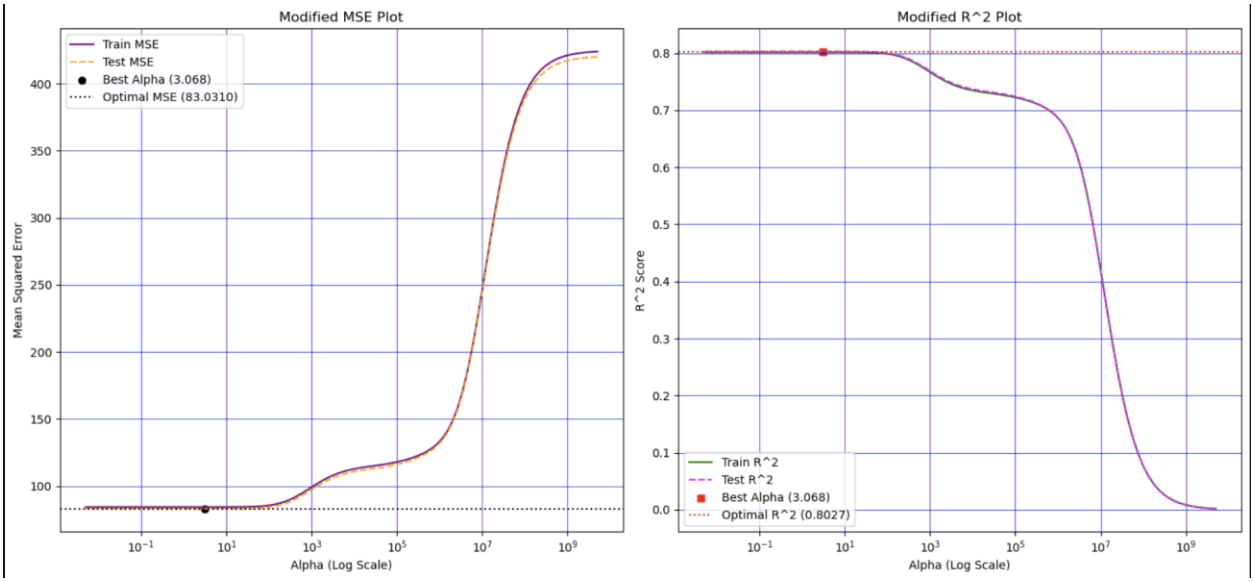
Second Analysis: Categorical included

Training MSE = 84.306

Test MSE = 83.032

Training R*2 = 0.801

Test R*2 = 0.802



In this case, overfitting was also avoided, and the model showed significant improvement compared to the numerical dataset, both in terms of error and goodness of fit. The model now explains approximately 80% of the variability.

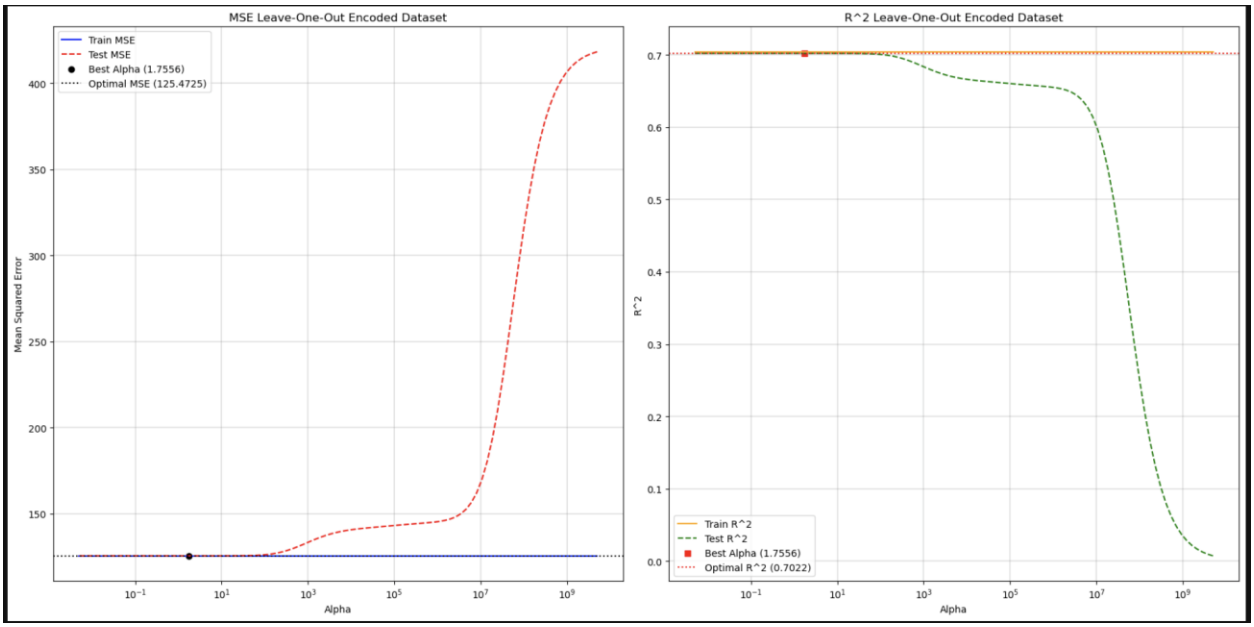
Third Analysis: Leave one out

Training MSE = 125.472

Test MSE = 125.493

Training R*2 = 0.704

Test R*2 = 0.702

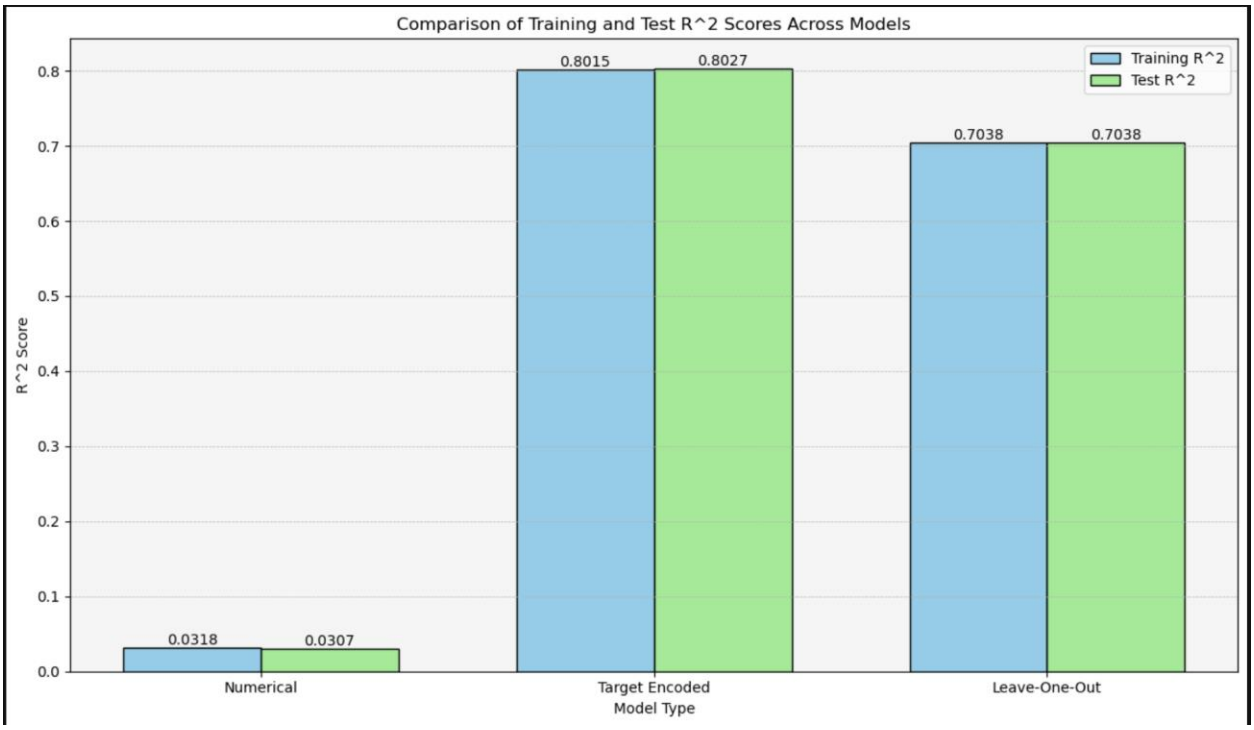


The model appears to perform good with approximately 70% of variance. However, it does not achieve the results of the Target encoded dataset.

Conclusion:

In conclusion, it’s evident that using only the numerical variables of the ’Spotify Tracks Dataset’ to predict the popularity of tracks is insufficient.

The model shows significantly better performance when all variables are utilized. Among the different encoding types tested, Target encoding achieved better results than the Leave one out encoding.


Comparison of Training and Test R^2 Scores Across Models
Training R^2
Test R^2
R^2 Score
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.0318
0.0307
0.8015
0.8027
0.7038
0.7038
Numerical
Target Encoded
Leave-One-Out
Model Type