Ridge Regression: Predicting Spotify Track Popularity

Thomas Mayr (12949A)

May 2024

1 Introduction

This project serves as a demonstration of ridge regression, cross validation techniques and hyperparameter tuning in machine learning tasks using Python. More specifically, the goal was to predict the popularity of Spotify tracks by incorporating information about the track such as sonic characteristics like tempo, energy, loudness and many more into a ridge regression model, experimenting with hyperparameters and optimizing the model accordingly. Two main specifications of this model are tested, one with only numerical variables, in essence using only the aforementioned sonic features, and one that includes categorical variables as well such as information about the genre of the track. The performances of each model are carefully evaluated and compared with one-another.

The structure of this project is as follows. First, the theoretical approach is briefly explained, then I proceed by talking about the steps taken to clean and properly prepare the data in the data preprocessing part, which most importantly includes a discussion of appropriately handling the categorical variables. After a very brief descriptive analysis, the main part of introducing different model specifications and comparing model performance follows. Lastly, a discussion about the overall results concludes.

2 Theory

The technique used for predicting track popularity is ridge regression. Ridge regression addresses the issue of unstable coefficients, sensitive to small changes in the data, due to multicollinearity, i.e. independent variables that are highly correlated with each other, by adding a regularization term to the traditional linear regression objective function. This term penalizes the size of the coefficients, forcing them to be smaller and thus reducing the impact of multicollinearity, introducing bias in the process. The regularization term is controlled by a hyperparameter α (often also denoted as λ), which determines the strength of regularization applied.

For the technical implementation the closed form solution, rather than the gradient descent approach, was used since computational efficiency was assumed to be higher this way due to the relatively manageable size of the dataset. The closed form solution is as follows:

$$\hat{\beta} = (X'X + \alpha I)^{-1} X'Y$$

The metrics used for comparing model performance were:

- **Root Mean Squared Error:**

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y_i})^2}$$

- $\mathbf{R^2}$:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$

with y_i being the actual observation and $\hat{y_i}$ the prediction.

To compute risk estimates **5-fold cross validation** and **nested cross validation** were used. Both were implemented from scratch.

5-fold cross-validation is a technique used to assess the performance of a machine learning model. It involves splitting the dataset into five equal-sized folds. In each iteration, one fold is held out as the validation set, while the remaining four folds are used for training. This process is repeated five times, with each fold being used once as the validation set. The model's performance metrics, in our case root mean squared errors, are then averaged across the five iterations to obtain an overall estimate of the model's performance. 5-fold cross-validation helps to provide a more reliable estimate of a model's performance compared to a single train-test split, as it utilizes multiple validation sets and reduces the variability in the evaluation process.

Nested cross-validation is an extension of cross-validation used for model selection and hyperparameter tuning. It involves performing an outer cross-validation loop and an inner cross-validation loop. In the outer loop, the dataset is split into multiple folds, using 5-fold cross-validation in this case. For each fold in the outer loop, a model is trained on the training data and evaluated on the validation data. Then, in the inner loop, another round of cross-validation is performed within each fold of the outer loop to select the best hyperparameters or model configuration. This inner loop helps to prevent overfitting to the validation set in the outer loop and provides a more accurate estimate of the model's performance. The final risk estimate is obtained by averaging the errors of the outer loop.

3 Data Preprocessing

The original data set consists of 114000 observations and 21 variables. The variable `Unnamed:0` was quickly identified as a index that provided no analytical use therefore leaving 20 variables that were actually considered potentially viable. These variables include meta-info about the tracks such as performing artists, genre, the name of the album the track appears on, the track name and a range of variables that mainly aim to quantify and describe the sound of the track such as energy, tempo or valence. Of these, the following features were identified as categorical (or maybe more accurately as non-numerical) from a conceptual perspective:

- `track_id`
- `artists`
- `album_name`
- `track_name`
- `explicit`
- `key`
- `mode`
- `track_genre`

The variable `track_id`, that simply denotes the Spotify track ID, cannot give us any information when it comes to predicting track popularity and was therefore deemed not viable for any model and discarded from the analysis. Nonetheless `track_id` was used to find duplicate tracks in the data set of which there were quite a few, 24259 to be exact. Duplicate tracks can occur for instance due to a track being released as a single but also being included on an album or a compilation. To avoid biased results, only the first occurrence in the data of such tracks was kept and all duplicates were dropped, leaving us with 89741 observations.

Moving on to the other features, the variables `key` and `mode` were already pre-encoded into a numerical format and were left as is. `key` maps integers to pitches in a ordinal format and `mode` gives information about the modality (major or minor) of a track, major is represented by 1 and minor is 0.

All other variables (`popularity`, `duration_ms`, `danceability`, `energy`, `loudness`, `speechiness`, `acousticness`, `instrumentalness`, `liveness`, `valence`, `tempo`, `time_signature`) were of numerical type and therefore did not receive any special consideration (other than rescaling) in the preprocessing process. The variable `popularity` represents the dependent variable.

To get a better understanding of relative feature importance, rescaling was applied to all numerical variables. A standard min-max-rescaling was chosen that results in all variables ranging from 0 to 1.

When it comes to the variable `artists`, which indicates the performing artists on the track, it should definitely be able to give information about track popularity but since there are 31437 unique artists in the data (although this includes collaborations) one-hot encoding would be extremely memory-intensive. Another option was considered: calculating the artist popularity by taking the mean popularity of all tracks of an artist and including it in the model. A quick analysis showed that this would greatly improve the predictive power of the model but since about 2/3 of all artists included in the data only have one single track, artist popularity is often equal to track popularity, therefore this approach was discarded. A similar approach would be to put artists into popularity categories like low, medium and high popularity and using this measure in our model, but since this is basically just a more fuzzy version of the former approach, still including lots of information from our target variable, this was also not used and the variable `artists` was ultimately discarded. For similar reasons the variable `album_name` was also discarded.

For the variable `track_name` using the length of the track name as a feature in the model was considered, the thought process being that listeners find shorter, simple track names more appealing than long, complicated names, but since no correlation was found between popularity and track name length, this idea, and with it the variable `track_name`, was also removed.

The variable `explicit` is a true-false boolean and was simple transformed to integer (1 being explicit) for technical reasons.

For the variable `track_genre` one-hot encoding was chosen since the number of genres is high but still manageable with 113 unique genres.

The decision to leave out the variables `artists`, `album_name` and `track_name` was taken with careful consideration and after extensive testing. Ultimately no sensible way was found to incorporate the information given by these features without risking overfitting or doing overly extensive target encoding that might create a model that performs extremely well in predicting track popularity but the results of which don't really have any interesting interpretation since its performance is heavily reliant on the information of the target variable itself.

Looking at the correlation matrix we can see that correlations between all variables and popularity are quite low, the highest in absolute terms being a negative correlation of 0.13 for the variable `instrumentalness`. The correlations between the predictor variables with one-another are generally substantially higher, with the highest being 0.76 of `loudness` with `energy`. This is somewhat supportive of the choice to use ridge regression since it should help with issues related to multicollinearity and unstable models.

4 Model 1: Numerical Features

For the model with only the numerical features, boolean variables and pre-encoded features such as `key` were excluded. Ultimately this results in a model that attempts to predict track popularity purely based on sonic features. As seen before though, these features exhibit a very low Pearson correlation coefficient,

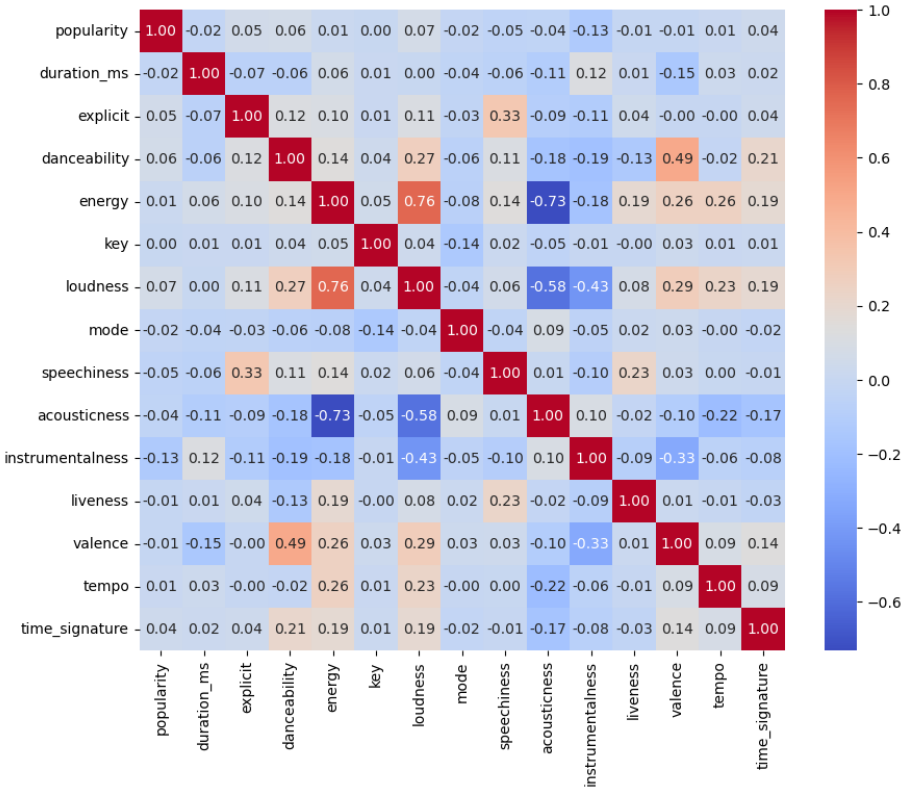


Figure 1: Correlations

suggesting that there is no strong (linear) relationship between sonic features and track popularity itself. Ridge regression results seem to confirm this.

The model was trained with a wide range of alphas $[10^{-5}, 10^5]$. For each alpha 5-fold cross-validation was applied to get appropriate risk estimates. A base model with $\alpha = 0$ was also evaluated, which just reduces to regular linear regression without any penalty on the coefficients. The results for cross-validated R^2-scores and root mean squared errors can be seen in the figures below. What is immediately noticeably is that performance on training and test set is extremely similar (in the images displayed the lines for train and test scores overlap). Though this should be somewhat expected since due to the cross-validation process the training and test errors are averages over multiple folds of the data thus ensuring are more realistic estimate of the models performance on unseen data usually bringing training and test errors closer together.

Cross-validation implies that setting the hyperparameter to $\alpha = 0.1$ achieves the best performance both in terms of root mean squared error (RMSE) and R^2. To be exact: $RMSE = 0.203$ and $R^2 = 0.0309$. For comparison, using the average of the training set track popularity for calculating the (cross-validated) RMSE yields $RMSE = 0.206$ and a model with $\alpha = 0$ which is equal to ordinary least squares regression yields a RMSE of $RMSE = 0.203$ as well. These results suggest that the predictive power of the features in this model is quite low and introducing a regularization penalty barely offers any improvement here.

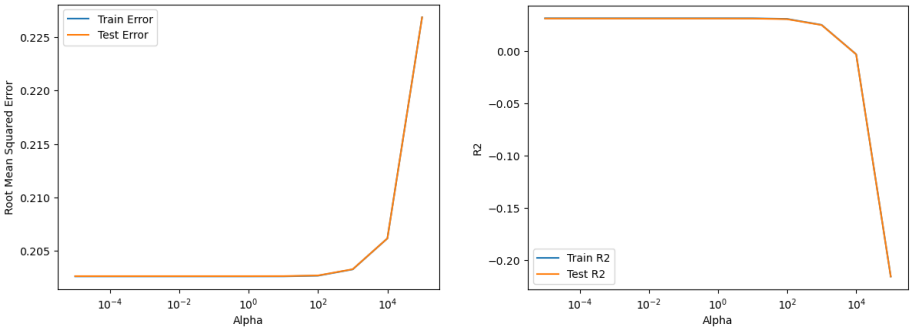
The risk estimate achieved with nested cross-validation was also $RMSE = 0.203$.

It is therefore not surprising that the weight of the feature coefficients are also quite low. The data implies that shorter tracks, that rely less on pure instrumentals and speech-like vocals are more popular. Furthermore, danceability also seems to positively affect popularity, while valence, which measures the musical positiveness of a track, seems to have a negative effect, suggesting that more negative tracks (e.g. sad, depressed, angry) actually are more popular. It must be said though that these results should be interpretated very carefully since the effects are so close to 0 and furthermore the dataset provided no information on how the tracks were sampled.

5 Model 2: Numerical and Categorical Features

For the second model all numerical features and the categorical features `key`, `mode` and `track_genre` (one-hot encoded) were included.

Again, for this model a wide range of alphas was used $[10^{-5}, 10^5]$ and again for each alpha 5-fold cross-validation was applied to get accurate risk estimates. Looking at the results for cross-validated R^2-scores and root mean squared errors it can be seen that the performance generally improved compared to the model with only numeric features, with the best hyperparameter value being $\alpha = 1$. The corresponding scores were: $RMSE = 0.169$ and $R^2 = 0.328$. When we compare this again to the score for the model that simply uses mean popularity as a predictor with a $RMSE = 0.206$ we can see a definite performance



(a) Root Mean Squared Error (b) R^2

Figure 2: Model 1 Performance

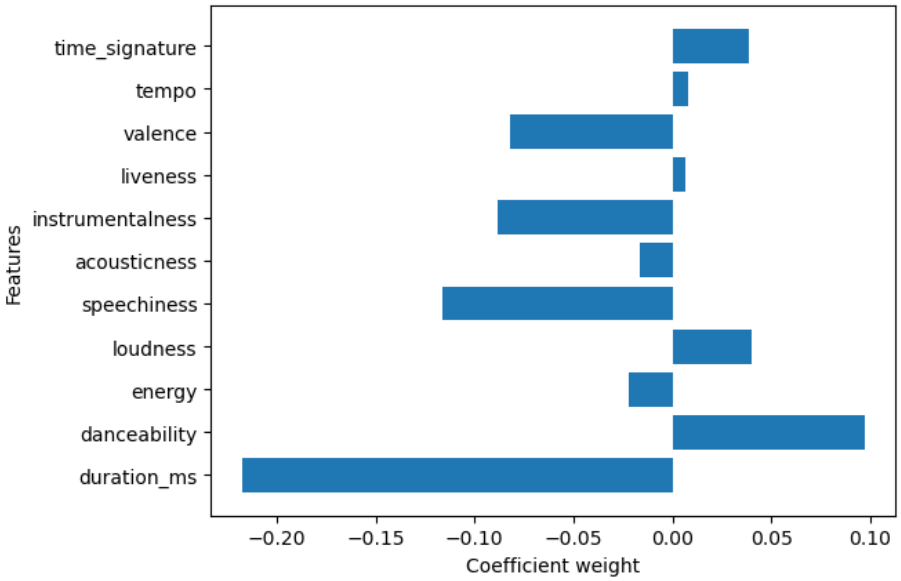
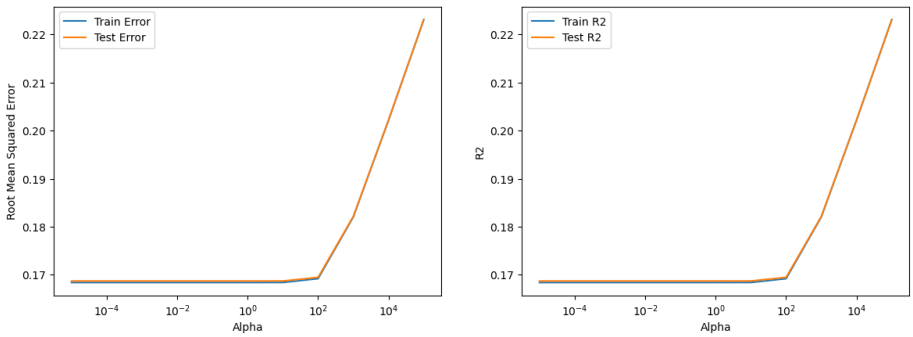
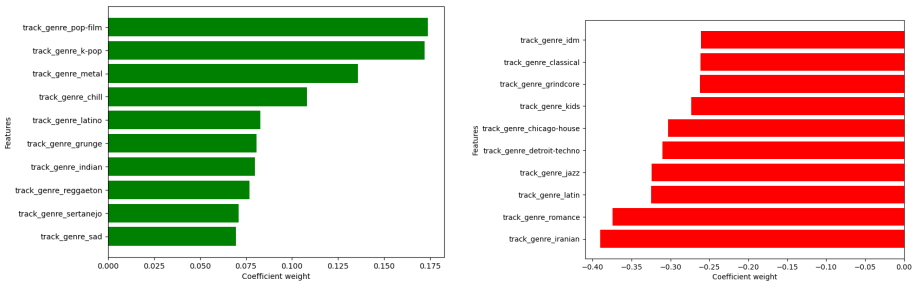


Figure 3: Model 1 Coefficient Weights



(a) Root Mean Squared Error (b) R^2

Figure 4: Model 2 Performance



(a) Top 10 Features (b) Bottom 10 Features

Figure 5: Model 2 Coefficent Weights Sorted

improvement. The risk estimate achieved with nested cross-validation was also $RMSE = 0.169$. The result suggests that information about a track belonging to a specific genre carries a relatively big amount of information about popularity, maybe unsurprisingly since it seems to be expected that certain genres are just generally more appealing than others, especially in comparison to more obscure types of music. The OLS model without any penalty on the coefficients achieves almost the exact same performance as the ridge regression model. The ridge regression therefore does not seem to offer any particular advantages for this specific model configuration.

When we look at the coefficient weights we can see that the features with the highest postive and negative effects all are variables related to a genre. Looking at these genres there is no obvious, discernable pattern for their influence. For example, one might expect genres that are generally more popular like pop or rock music to have the largest coefficients but instead it is "pop-film", I will therefore not attempt to further interpret these results. What is noticeable though is that the effects of all other variables get extremely close to 0, including the other features not used in model 1 `explicit`, `key` and `mode`.

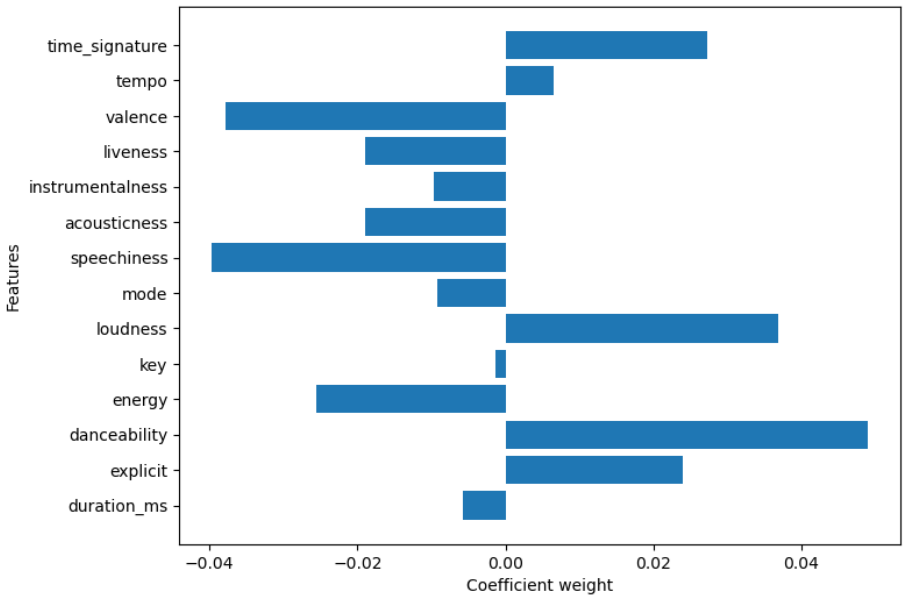


Figure 6: Model 2 Coefficient Weights Non-Genre Features

6 Conclusion

Overall, when we look at the performance of both models we can conclude that model 1, which relies almost exclusively on sonic features with a wide variety of variables that try to capture how a track sounds and feels to the listener, performed quite poorly. This suggests that either the data is insufficient, the variables measure their target inadequately or simply that sound and feel is not what primarly contributes to a tracks overall popularity. In contrast, model 2 that incorporated categorical variables, most importantly the genre of the track, offered some performance improvement.

For both models the ridge regression model with its penalty-inducing hyper-parameter did not offer any noticeable improvement when it comes to preventing overfitting, dealing with unseen data and improving performance metrics.

A major challenge in the data preprocessing was how to deal with some of the categorical variables. Ultimately, for some of them no satisfactory method was found to incorporate them in the model.

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. I 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.