RIDGE REGRESSION
ON SPOTIFY TRACKS DATASET

Data Science for Economics
-
Machine Learning and Statistical Learning

Author
Davide Matta

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/us or any other person for assessment on this or any other course of study.

Contents

1 The project

1.1 Spotify Tracks Dataset

The project is based on the Spotify Tracks Dataset, available on Kaggle. This dataset provides data for over 100.000 songs, with both categorical and numerical features. Categorical features include the artists, album and genres of the song, while numerical features mostly refer to the song's characteristics, such as danceability, acousticness or the duration of the song, and finally two binary feature are present as well, indicating whether the song's lyrics are explicit or not and whether the song's modality is major or minor. The target variable is the song popularity, an index from 0 to 100 based on the total number of plays and their recency. Additional info on the data can be found in the Kaggle page.

1.2 Methodology

The project requires to use Ridge regression to predict the track popularity, first with just the numerical features and then with all the features. In order to do that, data have been first appropriately transformed, and 5-fold cross validation has been used for computing the risk estimates. Finally, alternative algorithms have been tested to see how they compare with Ridge regression. Each step will be explored further later. More about the code in the final section.

2 Data cleaning and transformation

While exploring the data, some issues emerged, and in addition it was necessary to find a way to address the categorical features. These aspects will be discussed in this section.

2.1 Duplicate tracks

This was probably the most relevant issue: some tracks appear multiple times, sometimes with the same id and others no, but always with the same technical characteristics, which indicates they are the same song. This happens because some songs are present both as single tracks and as songs included in albums or compilations.

This would not be a problem by itself, but it is because popularity is computed separately, and this leads to situations where the same exact song with its identical numerical features in all records, has in some occurrences a high popularity and in others a really low one. It is easy to see how this would create problems to any learning algorithm, in particular when the analysis is restricted to numerical features.

In addition, the situation is complicated by the fact that, in some cases, despite it is evident that two tracks are the same, some measures are slightly different, so the problem cannot be solved just removing identical duplicates.

Here are the actions taken:

- All numerical columns have been rounded to two decimal places to reduces cases like those described before. Since most columns are indexes between 0 and 100 this does not create problems.

- Among the tracks that shared the same technical features, genre included, only the one with highest popularity was kept, as we assume this is "the original". In fact, in all examples checked the most popular track was always the song extracted from the album, while songs from compilations have really low popularity in comparison.

- However, this action was not totally resolutive, as there are still some duplicates that only differs for one or two column (let's say, for example danceability 22.43 instead of 22.39, with all the other columns identical). For this reason, there was no choice other than removing all the low popularity tracks, meaning their popularity is lower than 10. This was not an easy choice because it led to removing proper tracks as well, but it also ensured that the analysis on what remains, that is the vast majority of the data, is reliable. The obtained predictors will just not work when songs are particularly unpopular, as in the training set we do not have anymore this kind of songs.

2.2 Categorical features

Another issue was how to relate to categorical features (artists, album and genre). One-hot encoding is not practicable since we have too many occurrences, then the choice is target encoding. In other words, the genre 'pop', for instance, is replaced by the average popularity among the songs that belongs to the pop genre. However, further adjustments were necessary for all the three column

2.2.1 Multi-genre songs

Some tracks are exactly the same, including the id, but differ in their genre. These are the multi-genre songs, that are labelled with more than one genre (for instance, both pop and dance). For these songs only the genre that is the most frequent in the dataset has been kept.

2.2.2 Multi-artist songs

In many tracks, the 'artists' column contains more than one artist. Considering each combination of artists as an artist itself may be appealing, but actually most of them would be rare or even appear only once, then information derived from them could be misleading and absorb effects that actually came from other variables. For this reason, among the artists in a track, only the most popular one has been kept, assuming he/she is the main driver for the song's popularity. Artists' popularity has been computed as the average popularity of the songs where they are present.

2.2.3 Homonymous albums

Some albums may have the same title despite they are different projects from different artists. Luckily, the first artist in the artists column seems to be always the album artist, then it was possible to use the couple album title - first artist to identify unique albums.

The distribution of the target variable popularity after the data cleaning and transformation operations is shown below in Figure 1.

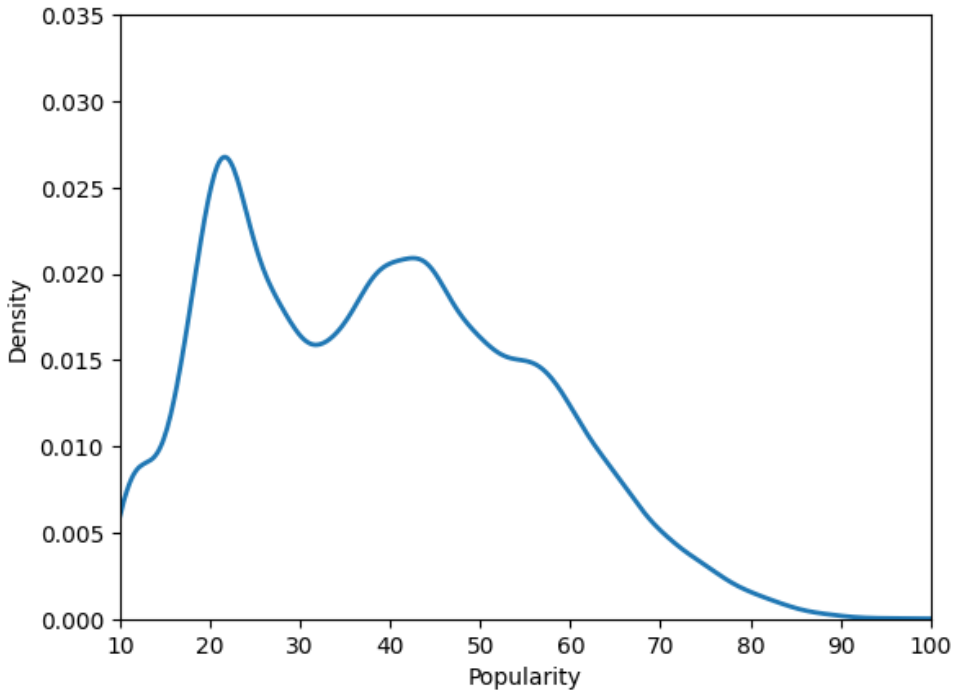

Figure 1: Density Plot of Popularity

3 Ridge Regressions

As previously stated, the project requires to implement Ridge regression, first only on numerical features, then on both numerical and categorical features.

3.1 Basics of Ridge Regression and Cross Validation

3.1.1 Ridge Regression

Ridge regression is an improved version of the standard linear regression model, and it is used to address the problems of multicollinearity and overfitting.

The model achieves its goal through the introduction of a regularization term into the model. This regularizer adds a constraint to the optimization problem, making sure that the model coefficients remain small or even close to zero, but still allows them to have some impact on the model. The extent of the regularization is controlled by the hyperparameter alpha, which determines the trade-off between bias and variance error.

The close form of Ridge regression is:

$$w = (S^T S + \alpha I)^{-1} S^T y$$

where α is the regularizer, S is the training example matrix, y is the vector of labels and w is the vector of weights

3.1.2 Cross Validation

In order to compute the risk estimates, 5-fold cross validation has been used. This method splits the sample in 5 folds and train the model 5 times, where in each of them one of the folds serves as the test set and the other four as the training set. By doing so, we can rely on more robust estimates on the model's performances.

3.2 Ridge Regression on numerical features

As a first model, a Ridge regression on only the numerical features, included the binary 'Explicit' and 'Mode', is run.

The regressions' coefficients are shown in Table 1. Some of them are close to zero: duration is precisely zero with this level of approximation (yet not zero exactly), but the values of this column are extremely large, so it is normal that the coefficient is small as the target variable is smaller; this extends to tempo as well, but to a much lower extent.

Another notable aspect is that the value of the hyperparameter alpha do not influence so much the coefficients. In fact, values are pretty similar, except when alpha is 100: in this case there is some difference but it does not seem particularly relevant.

In Figure 2 is reported the performance of the model. The chosen metric is RMSE (Root Mean Square Error). Since the target variable is numeric it does not make sense to use metrics like accuracy, so we rely on the mean square error of prediction, and we take the root in order to have it expressed in the same unit as the target variable. RMSE has been preferred to MAE (Mean Absolute Error) because with RMSE larger errors are more penalized.

Table 1: Ridge coefficients - numerical features only

Alpha	0.01	0.1	1	10	100
Intercept	47.90	47.90	47.86	47.63	37.39
Duration (ms)	-0.00	-0.00	-0.00	-0.00	-0.00
Explicit	4.72	4.72	4.72	4.69	4.43
Danceability	11.63	11.63	11.50	11.59	11.99
Energy	-8.59	-8.59	-8.23	-7.86	-5.21
Key	0.03	0.03	0.03	0.03	0.05
Loudness	0.14	0.14	0.14	0.12	-0.02
Mode	-1.00	-1.00	-1.00	-0.96	-0.73
Speechiness	-19.35	-19.35	-19.55	-19.34	-17.39
Acousticness	-2.97	-2.97	-3.05	-2.90	-1.83
Instrumentalness	-10.34	-10.34	-10.26	-10.26	-10.20
Liveness	-2.88	-2.88	-2.92	-2.94	-3.06
Valence	-6.66	-6.66	-6.58	-6.57	-6.36
Tempo	-0.01	-0.01	-0.01	-0.01	-0.00
Time Signature	0.47	0.47	0.41	0.55	1.67

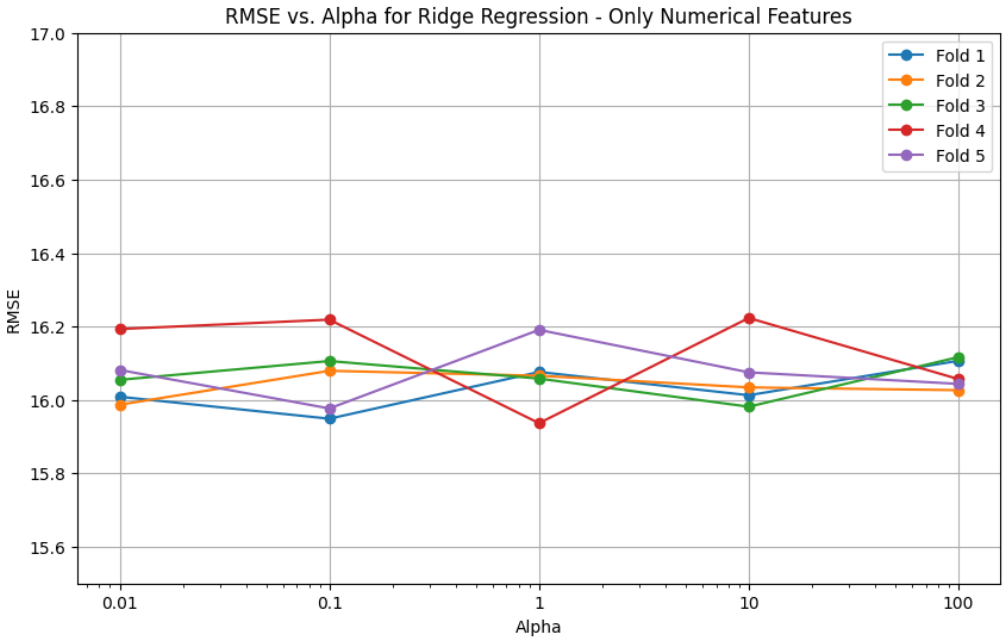

Figure 2: Performance of Ridge regression on numerical features

As the graph shows, RMSR is around 16 in all 5 folds with all the levels of alpha. Considering that popularity is an index from 0 (10 for us) to 100, this means the model gets wrong by 16 points on average in predicting a song popularity. This is not satisfactory, but it is at least decent and indicates that good performances could probably be achieved including the categorical features, and this is in fact the next step.

3.3 Ridge Regression on all features

This second model is an improved version of the previous one, and include the three categorical features: artist, album and genre. The new coefficients are shown in Table 2.

Table 2: Ridge coefficients - all features

Alpha	0.01	0.1	1	10	100
Intercept	-0.37	-0.30	-0.34	-0.37	-0.30
Duration (ms)	0.00	0.00	0.00	0.00	0.00
Explicit	0.13	0.16	0.17	0.15	0.15
Danceability	0.12	0.07	0.06	0.05	0.01
Energy	0.25	0.17	0.15	0.19	0.22
Key	0.00	-0.00	-0.00	0.00	0.00
Loudness	0.00	0.00	0.01	0.00	0.00
Mode	0.02	0.02	-0.00	0.02	0.01
Speechiness	-0.30	-0.32	-0.29	-0.26	-0.26
Acousticness	0.19	0.16	0.13	0.15	0.17
Instrumentalness	-0.03	-0.08	-0.02	-0.06	-0.12
Liveness	-0.13	-0.03	-0.07	-0.09	-0.13
Valence	0.14	0.11	0.20	0.18	0.14
Tempo	-0.00	-0.00	0.00	-0.00	0.00
Time Signature	-0.01	0.01	0.01	0.00	0.00
Album Popularity	0.98	0.98	0.98	0.98	0.98
Artist Popularity	0.02	0.02	0.02	0.02	0.02
Genre Popularity	-0.00	-0.00	-0.00	-0.00	-0.00

As we can see, here the track popularity is highly dependent on the album popularity. This makes sense, but the relation might be inflated by the fact that in many cases we do not have all the songs from a particular album, and in some of them we actually only have one song, so that the album popularity is just the popularity of that one track. If we had the whole album the relation would probably be smaller but we cannot prove to what extent with this data.

It is worth noting that introducing these features reduced the magnitude of the coefficients of the other variables because of the strong correlation between the track popularity and album in particular, and also the new ones are smaller but mainly because the new variables are in the same scale of the target variable.

There is also a larger number of feature with coefficient zero or close to zero, as it expectable from the Ridge model, but what explained for the other model remains true. Among the 0-1 indexes, energy and speechiness are the most relevant for the model, influencing the popularity positively the former, negatively the latter.

RMSE for this model is shown in Figure 3. Introducing the categorical features dramatically reduced the RMSE in all folds and for all levels of alpha, as it now lies around 3.3 and

3.6. This is a five-times reduction and it is a remarkable result for such task. Of course, we need to take into account that with more complete data the predictive power of the album popularity would probably decrease, but in these conditions results are good.

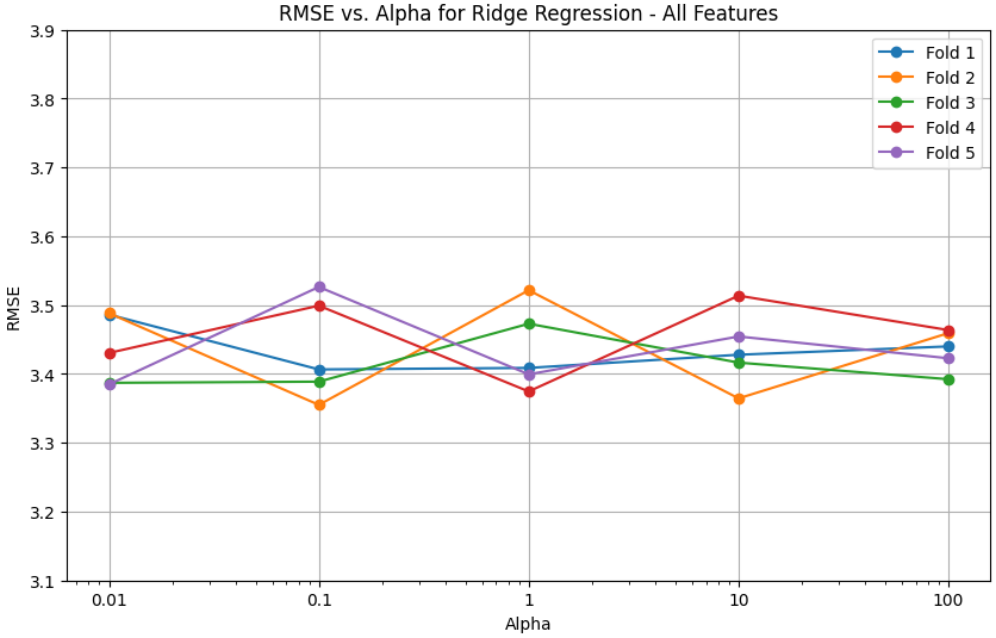

Figure 3: Performance of Ridge regression on all features

Since predictions are already highly precise, it does not make sense to experiment more with this model, for example adding polynomial terms or using the kernel trick.

However, it might be interesting to compare Ridge regression results with the output of other algorithms: this analysis is performed in the next section.

4 Other Models

In order to compare the model with other algorithms, Lasso - another kind of regularization - and the standard linear regression model have been applied to the same data. These algorithms have directly been feeded with all available features.

4.1 Lasso Regression

The main difference between Ridge and Lasso is that with Lasso coefficients can be exactly zero, thus the algorithm serves as a feature selection method as well and could do a better job in addressing multicollinearity.

RMSE for Lasso regression with all variables are reported in Figure 4.

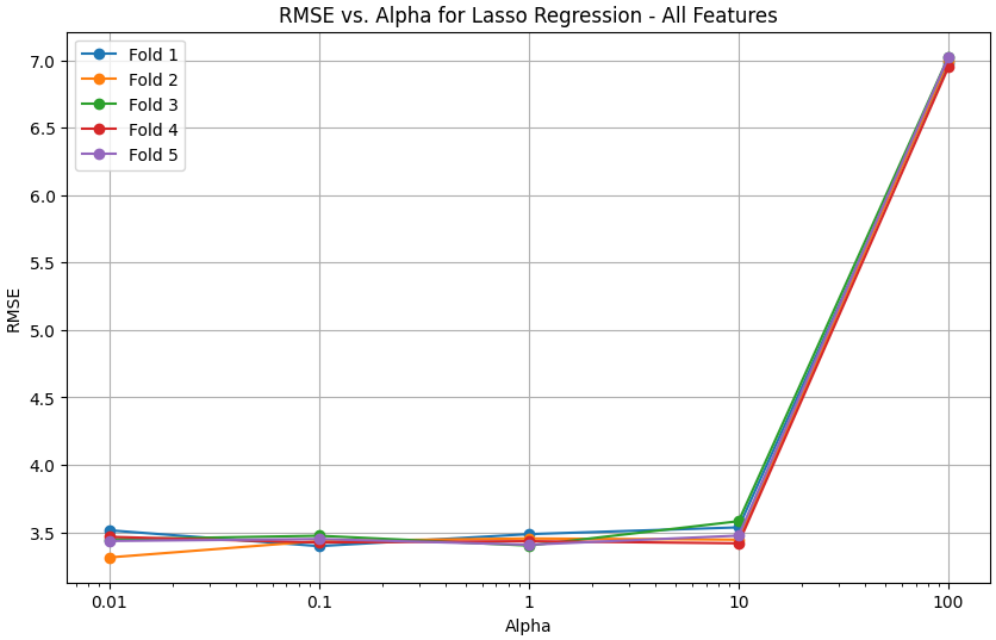

Figure 4: Performance of Lasso regression on all features

As the graph shows, Lasso performs about the same as Ridge, except when alpha is 100, where Ridge did a better job in comparison. So, unless we want to simplify the model removing some variables, there is no reason to prefer Lasso to Ridge in this application.

4.2 Linear Regression

Other than wondering which regularization method does better, we may check if we need regularization at all. For this purpose it is worth trying to estimate a standard OLS model, with all the variables in this case as well.

Results are reported in Figure 5. The graph is different from the others since we do not have a regularizer in standard OLS, so it is only possible to show the RMSE for each fold. Results range from 3.3 to 3.5, and are again similar to what Ridge achieved.

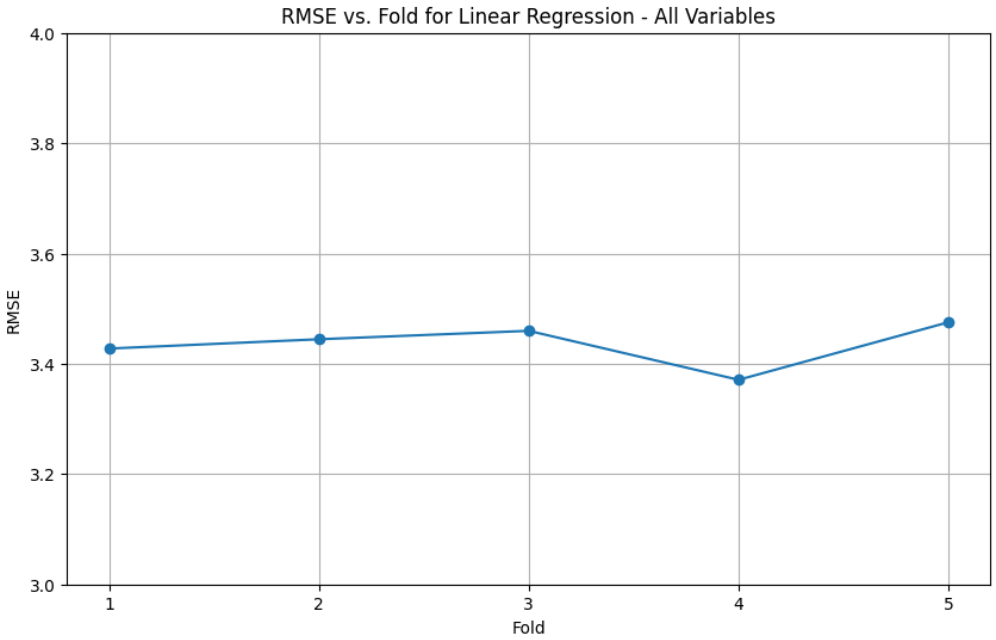

Figure 5: Performance of Linear regression on all features

4.3 Ridge vs Lasso vs OLS

To sum up, the three models did a similar job, except Lasso when alpha was particularly large. Of course, what specified before about the album popularity remains true with the other two models as well. For the needs of this task, the standard linear regression, which is the simpler model, is largely sufficient, but in other settings there could be reasons to prefer one of the other models. In particular, with more complete data for albums the regularization parameter could assume more relevance.

5 Conclusion

Finally we can summarize the main results of the project:

- Ridge regression on numerical features performed poorly.

  The model did not go any further than a RMSE = 16, so these variables are not sufficient to predict accurately the popularity.

- Adding categorical features improved significantly the predictive power of the model.

  Introducing Artist, Album and Genre, through target encoding, helped the model to improve and obtain a five-time RMSE reduction to something around 3.4.

- However, these results must be dealt with caution.

  In particular, we do not have all the songs for most albums, and since Album was the most relevant feature this is a sign that the model could not work with more complete data (that is, with a more accurate evaluation of the album popularity).

- The value assigned to the hyperparameter alpha is not influential in this application.

  Five different values for alpha have been tested but the change in results is negligible.

- Lasso or Linear regression had similar performances.

  None of these models provided better results (neither worse), and the choice among different models would only depend on the purpose of the work.

6 Code

The code can be found on Github. Here is a description of the files in the repository.

- project_report.pdf: this file.
- data: this folder contains the original dataset plus the elaborations on it.
- data_cleaning.ipynb: in this notebook are operated data cleaning and elaboration tasks.
- ridge.py: this is the script for the Ridge regression.
- model.ipynb: this notebook has been used to run the model and generate metrics and graphs.
- other_models.ipynb: in this model other techniques (Lasso and OLS) are performed for comparison purposes.