# Machine Learning Project: Ridge Regression

Francesco Scateni

A.Y. 2023-2024

## Contents

# 1 Introduction

This project focuses on the implementation of the Ridge Regression algorithm, which has been developed from scratch using the Python language. The dataset used for this project is the 'Spotify Tracks Dataset'.

Ridge Regression is a linear algorithm that predicts a chosen variable in the dataset, also known as the target variable. In this case, the goal is to predict the popularity of songs.

The dataset includes both numerical variables related to technical aspects of the songs (such as 'acousticness' and 'tempo') and categorical variables (such as 'artists' and 'album_name').

Therefore, after a phase of data preprocessing and exploration, the algorithm was first tested on the dataset, considering only the numerical variables. Subsequently, it was applied to the entire dataset, incorporating encoding techniques for the categorical variables.

Then, two very popular metrics were used to assess the algorithm: the Mean Squared Error (MSE) to calculate the error estimate, and the coefficient of determination ($R^2$) to estimate the accuracy.

Moreover, for each model created, different values of the $\alpha$ parameter were tested using a simple for loop. Then, the risk estimate was calculated by finding the best alpha value using 5-fold cross-validation.

Finally, a performance comparison between the different models is shown through the use of explanatory plots.

# 2 The Dataset

## 2.1 Description

The 'Spotify Tracks Dataset' is composed of 114,000 rows and 21 columns. Each row represents a song, serving as the collected observations, while the columns represent the characteristics of each song.

Not all columns were significant, so the dataset needed to be explored before running the algorithm.

## 2.2 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 observed 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 biases in 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 in order to obtain a clean dataset with unique track values and one column one-hot encoded for each track.

#### 2.2.1 Data Exploration

Some numerical variables appear to have outlier values and high skewness in absolute terms, indicating a non-normal distribution.

However, an effective approach to avoid removing these values and losing information is to standardize the variables, as discussed in section 2.4.

Furthermore, there are no significant multicollinearity issues. Specifically, the only variables exhibiting relatively high correlation are 'acousticness' with 'energy', showing a negative coefficient of -0.73, and 'energy' with 'loudness', displaying a positive relationship of 0.76.

However, considering the nature of these characteristics, such correlations are reasonable.

### 2.3 Categorical Variables Encoding

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 appropriate: 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, Catboost, and Leave-One-Out encoding for 'artists', 'album_name', and 'track_name'.

  These three variables have too many unique values and are not ordered, making it very difficult to apply other types of encoding.

In the final section, the performance of the algorithm with respect to these encoding types will be compared.

### 2.4 Standardization and Dataset Splitting

The final step before proceeding with the experiment involves standardizing the numerical variables using the Z-score formula:

$$Z = \frac{X_i - \mu}{\sigma}$$

This approach was chosen for the reasons discussed in section 2.2.1, and also because not all variables are on the same scale.

With standardization, most values fall within the range [-1, 1], helping to prevent overfitting and ensuring that variables with larger scales do not dominate the model.

Additionally, the dataset was split into two parts: training (X_train, y_train) and test (X_test, y_test) sets, with a ratio of 80% and 20%, respectively.

This splitting was performed with a random seed equal to 0 to ensure reproducibility of the results and before the standardization to avoid data leakage between training and test set.

Subsequently, only the numerical variables were extracted from X_train and X_test, resulting in X_train_num and X_test_num.

Finally, the standardization and dataset splitting processes were repeated for the other encoding types at the end, using the same random seed.

## 3 Ridge Regression

### 3.1 Theory

The Ridge Regression algorithm is a linear predictor. However, unlike classic Linear Regression, it introduces a hyperparameter alpha ($\alpha$). This hyperparameter allows the algorithm to shrink the coefficient estimates towards zero, aiming to reduce variance and ultimately improve risk.

Instead of using the Linear Regression formula:

$$\boldsymbol{w}_S = \arg\min_{\boldsymbol{w}\in\mathbb{R}} \|S\boldsymbol{w} - \boldsymbol{y}\|^2$$

to calculate the vector of weight estimates $\boldsymbol{w}_S$, Ridge Regression utilizes:

$$\boldsymbol{w}_{S,\alpha} = \arg\min_{\boldsymbol{w}\in\mathbb{R}} \|S\boldsymbol{w} - \boldsymbol{y}\|^2 + \alpha\|\boldsymbol{w}\|^2$$

where $\alpha > 0$ is the regularization parameter. When $\alpha \to 0$, we have the regular Linear Regression, when $\alpha \to \infty$ the solution of $\boldsymbol{w}_{S,\alpha}$ becomes the zero vector.

In matrix notation, the closed-form formula for Ridge Regression to calculate the weights $\boldsymbol{w}_{S,\alpha}$ is:

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

where $I$ is the identity matrix, $S$ is the design $m \times d$ matrix, $S^T$ is its transpose, and $\boldsymbol{y}$ is the vector of target feature values.

### 3.2 Implementation

To implement the algorithm, a RidgeRegression class was created with the following components:

- A __init__ constructor, which includes a default value for alpha ($\alpha = 1.0$).
- A fit() method, which is the core of the algorithm. It computes the vector of weights using the closed-form formula mentioned in the previous section.
- A predict() method, which makes predictions on both the training and test sets.
- An mse() method to compute the Mean Squared Error. It can be used to calculate both training and test MSE. It employs the formula: $\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$.
- An r_squared() method to compute the $R^2$. It utilizes the formula: $R^2 = 1 - \frac{RSS}{TSS}$, where RSS is the Residual Sum of Squares and TSS is the Total Sum of Squares.
- A cross_validation() method, which utilizes the Kfold() method from sklearn to compute the risk estimate and find the best alpha value across a list of values. The default value is 5.

## 4 Experiment Results

In each experiment, the following steps were performed:

1. Fit the model on the training set.
2. Make predictions on the training and test sets.
3. Compute the training and test MSE and $R^2$.
4. Test the performance of the model with different alpha values. A list of 100 values between $10^{-2}$ and $10^{10}$ scaled by 0.5 was used. Then, a graph was plotted to show how the performance changes with alpha.
5. Find the best alpha value and compute the risk estimate with 5-fold cross_validation method() (using the same list of values).

These steps were repeated for each experiment to evaluate the performance of the Ridge Regression model under different conditions.

## 4.1 Numerical Dataset

The first experiment was conducted on the Numerical dataset. Here are the results: Training MSE = 408.0631; Test MSE = 418.9764; Training $R^2$ = 0.0312; Test $R^2$ = 0.0318.

The small difference between the training and test error suggests that overfitting was avoided. However, the $R^2$ values are quite low, indicating that the numerical variables explain only 3.1% of the variance in the target variable.

Furthermore, as shown in Figure 1, the model performs worse for very high values of alpha (from approximately $10^4$ onwards).

Finally, the 5-fold cross-validation calculated an optimal value of alpha = 201.85 with a test risk estimate of 408.1988.

This suggests an improvement over the error computed using the default alpha value, but performance can still be enhanced by including all variables.

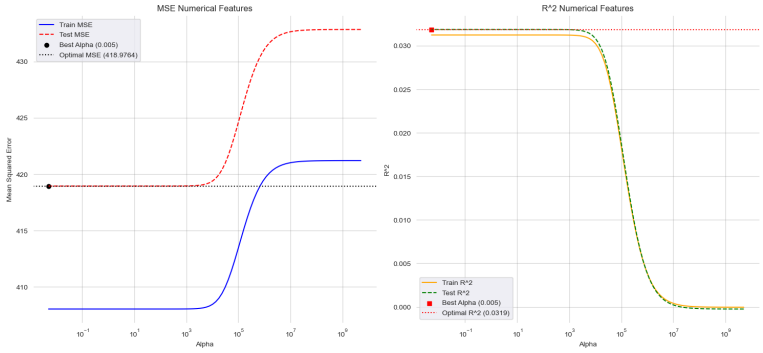


Figure 1: The left side shows the variation of training and test MSE with different values of alpha for the Numerical dataset; the right side displays the corresponding trend for $R^2$.

## 4.2 Encoded Datasets

### 4.2.1 Target Encoding

The model was then implemented on the Target encoded dataset. The results are as follows: Training MSE = 84.6608; Test MSE = 88.8601; Training $R^2$ = 0.7990; Test $R^2$ = 0.7946.

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 79.46% of the variability.

Moreover, Figure 2 illustrates the trend of alpha as MSE and $R^2$ vary. Similar to the previous dataset, for large values of alpha, the performance worsens. However, there is a difference: in this case, there is a slight worsening for values between $10^3$ and $10^5$, with a clear peak for values above $10^7$.

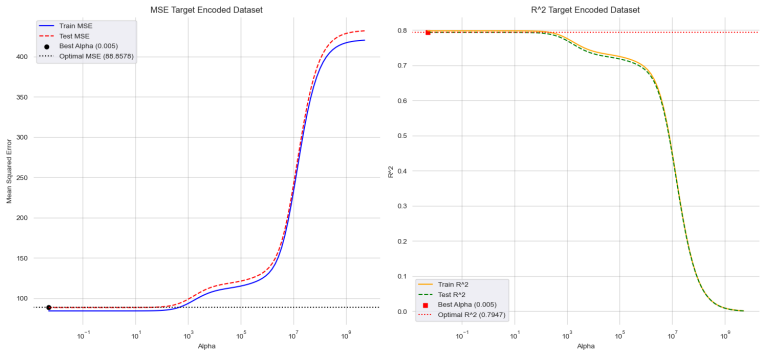


Figure 2: The left side shows the variation of training and test MSE with different values of alpha for the Target encoded dataset; the right side displays the corresponding trend for $R^2$.

Finally, the 5-fold CV suggests an optimal value of alpha = 1.3280 with an estimated test MSE = 85.0465. Once again, cross-validation provided a slightly better estimate.

### 4.2.2 Catboost Encoding

As for the Catboost encoded dataset, the following results were obtained: Training MSE = 149.0459; Test MSE = 155.2973; Training $R^2$ = 0.6461; Test $R^2$ = 0.6411.

Once again, there is a notable improvement compared to the numerical dataset; however, the model does not perform better than the Target dataset.

Additionally, the changes in alpha appear to be very similar, as shown in Figure 3.

The risk estimated by the 5-fold CV is: MSE = 149.7125 with an optimal alpha = 1.3280. Therefore, the alpha value is exactly the same as the model with Target encoding, and once again, the estimate for the MSE is lower.

### 4.2.3 Leave-One-Out Encoding

Lastly, the Ridge Regression algorithm was tested on the Leave-One-Out encoded dataset: Training MSE = 125.1255; Test MSE = 131.0191; Training $R^2$ = 0.7029; Test $R^2$ = 0.6972.

The model appears to perform better than Catboost, with approximately 69.72% of variance explained compared to the previous 64.11%. However, it does not achieve the results of the Target encoded dataset. Similar to the other two encoded datasets, the model performs better for low alpha values.

Finally, the 5-fold CV has estimated an optimal alpha value of 3.0679 with a risk estimate of 125.6072, once again showing a clear improvement.

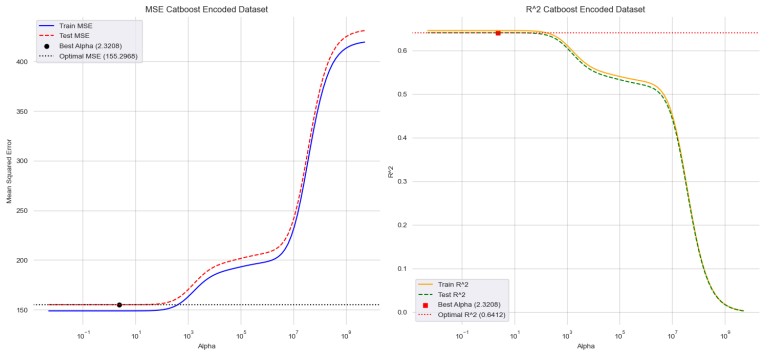


Figure 3: The left side shows the variation of training and test MSE with different values of alpha for the Catboost encoded dataset; the right side displays the corresponding trend for $R^2$.

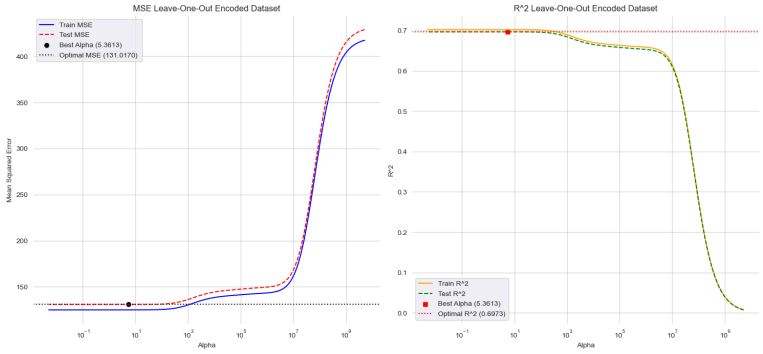


Figure 4: The left side shows the variation of training and test MSE with different values of alpha for the Leave-One-Out encoded dataset; the right side displays the corresponding trend for $R^2$.

# 5 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 the best results, followed by Leave-One-Out and Catboost encoding, in that order.

The figures below provide a summary of the performance of the different models.

Figure 5 illustrates the training and test MSE values across the different datasets (calculated without cross-validation support), while Figure 6 displays the different goodness-of-fit measures ($R^2$).

Finally, Figure 7 presents the different risk estimates calculated with 5-fold cross-validation, along with the corresponding optimal alpha value.

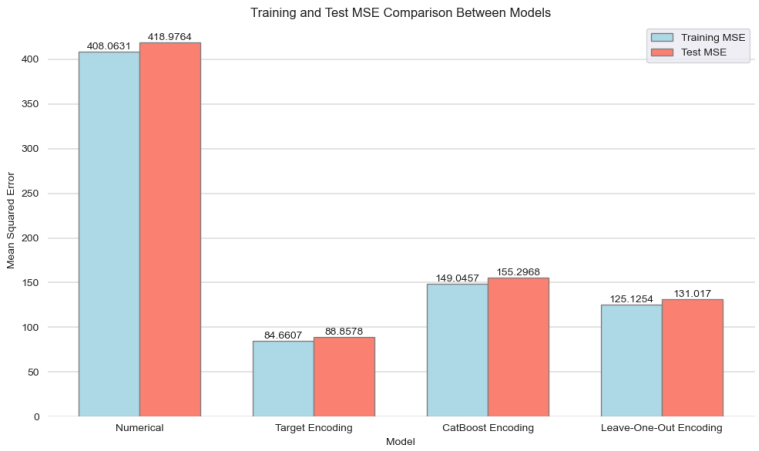


Figure 5: The plot shows the performance of each model in terms of training and test MSE.

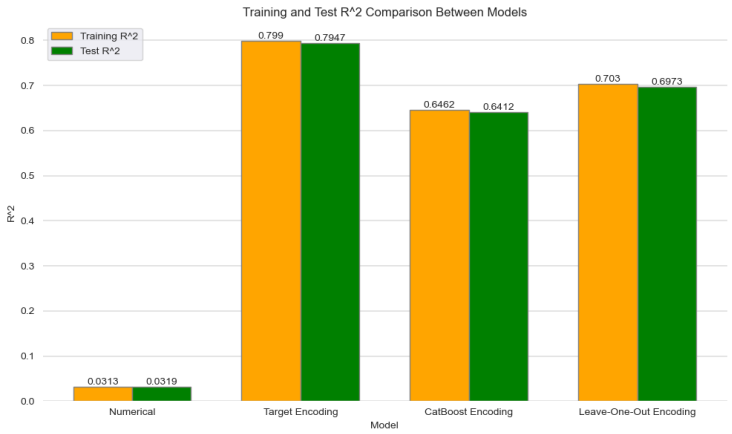


Figure 6: The plot shows the performance of each model in terms of training and test $R^2$.

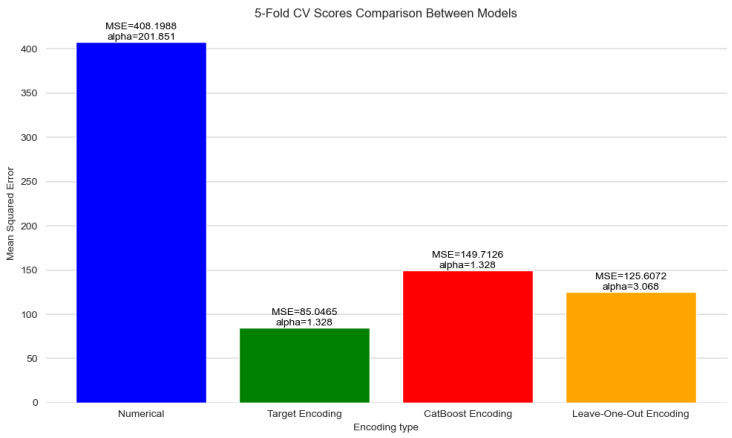


Figure 7: The plot shows the difference between the 5-fold CV scores for each model, including both alpha and MSE.

# 6 Final Declaration

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.