Optimizing Predictive Modeling for Song Popularity with Ridge Regression: A Comprehensive Study

Martín do Río Rico[1]

[1]Department of Computer Science, Università degli Studi di Milano, Milano, Italy.
[2]Department of Computer Science, Universidade de A Coruña, A Coruña, Spain.
[*]Address correspondence to: martin.doriorico@studenti.unimi.it

Abstract

This report conducts a thorough study of a dataset to develop an efficient predictive model for song popularity. After data acquisition and exploratory analysis, numerical and categorical features were processed, including reclassification of some columns. Ridge regression with 5-fold cross-validation assessed the impact of preprocessing techniques on model performance, while grid search optimized the regularization parameter. Challenges during cross-validation were addressed and resolved to enhance the accuracy of the predictive model.

Introduction

This report focuses on a comprehensive study of a data set with the primary goal of analyzing both numerical and categorical features to develop an efficient predictive model for song popularity.

The exploration of the data set included a detailed analysis of various aspects, such as the distribution of numerical variables, the cardinality of categorical variables, the presence of potential outliers, and the intricate relationships between independent and dependent features. The study aims to gain valuable insights into the data's characteristics and uncover patterns that can aid in developing a robust predictive model.

To optimize the predictive model, ridge regression was employed, and the key training parameter, the regularization parameter, was fine-tuned using 5-fold cross-validation. Cross-validation is a powerful technique to estimate model performance and prevent overfitting. The study aims to identify the optimal value for the regularization parameter that leads to the best generalization performance for the ridge regression model.

By studying both numerical and categorical features and employing cross-validation to optimize the training parameter, this report strives to provide valuable insights into the dataset and enhance the accuracy and effectiveness of the predictive model for song popularity. The findings will contribute to informed decisions in subsequent model development and foster advancements in the field of machine learning applications.

Study of the Dataset

The initial phase of this empirical project involved the acquisition of the data set, followed by meticulous measures to ensure its integrity. Data completeness and uniqueness were achieved through a thorough process, which involved removing duplicate rows and eliminating entries containing missing values (NaN).

Subsequently, an in-depth analysis of the data set was conducted, drawing inspiration from a similar project associated with the database. The primary objectives of this exploratory data analysis were to gain insights into several key aspects of the data, including the distribution of numerical variables, the cardinality of categorical variables, the presence of potential outliers, and the intricate relationships between independent and dependent features.

The dataset contained a total of 15 numerical columns. Upon closer inspection, it was evident that some of these columns could be more accurately classified as categorical variables. The numerical categorical features or discrete numerical features include:

- 'explicit': This binary variable indicates whether a track has explicit lyrics (1 for true and 0 for false).
- 'key': Represents the key of the track using standard Pitch Class notation.

- 'mode': Represents the modality (major or minor) of a track, denoted by 1 for major and 0 for minor.
- 'time_signature': An estimated time signature that specifies the number of beats in each bar (or measure), ranging from 3 to 7.

With the data set categorized accordingly, further investigation was conducted on the distribution of the numerical features.

Further delving into the numerical features, a meticulous examination of their distribution revealed intriguing patterns. 'Danceability,' 'Valence,' and 'Tempo' exhibited nearly normal distributions, while 'Loudness' displayed a distinct left-skewed behavior, as can be appreciated in Figure 1.

The remaining numerical features demonstrated right-skewed distributions, signifying variations in their data distribution patterns.

Before initiating any corrective measures or transformations, an investigation into the correlation between the numerical features and the target variable 'popularity' was conducted. The outcome indicated that none of the continuous features exhibited substantial correlations with the target variable, suggesting the need for careful feature selection and engineering to improve model performance.

Additionally, the data set was subjected to outlier analysis, revealing noteworthy observations. Notably, 'Energy,' 'Acousticness,' and 'Valence' showcased a considerable number of outliers, while several other features also contained scattered outlier values.

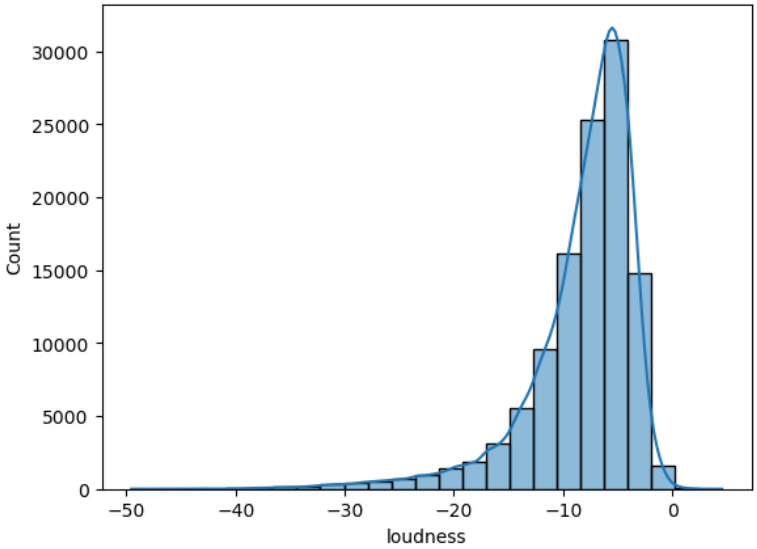
The comprehensive analysis of the data set, as presented in this report, provides a robust foundation for the subsequent stages of this empirical project. The acquired insights will significantly influence data transformation decisions and guide the development of effective predictive models.

Numerical Features Preprocessing

The preprocessing of numerical values is a critical step aimed at enhancing the predictive performance of regression algorithms. Two crucial steps were undertaken in this phase:

Skewness Removal: Skewness in continuous numerical features can adversely affect the predictive accuracy of regression models. To mitigate this issue, various transformations were applied to address the skewness present in the dataset's continuous columns. The objective was to obtain improved predictions for the target variable, 'popularity.' Although limited correlation was found between continuous features and 'popularity,' the decision to reduce skewness was made to assess its impact on model performance.

Standardization: Ensuring uniformity and comparability across variables is essential for regression models. Therefore, the values of numerical features were standardized to a common range. This step is crucial for facilitating unbiased parameter estimation and fostering convergence during optimization.



The data set exhibits left-skewness, which means most data points are concentrated on the right side of the graph. The left tail is longer and has some extreme values, causing the mean to be pulled to the left. The median is closer to the third quartile than the first quartile. The mean is less than the median. The skewness value for this left-skewed feature is -2.0133133823721505.

Figure 1: Skewness of the 'loudness' feature.

Addressing skewness in continuous data is imperative for the effective performance of regression models. An analysis revealed that several continuous columns in the data set exhibited skewness. Hence, transformations were applied to these features before proceeding with model development. Figure 2 shows a visual representation of the transformations.

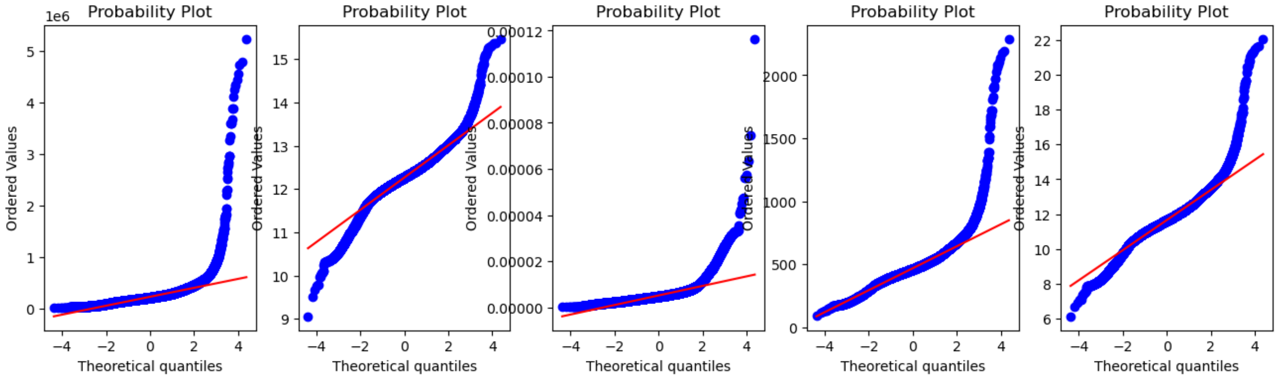
Four different transformations, namely logarithmic, reciprocal, square-root, and exponential, were applied to the skewed continuous features. The results of these transformations are summarized below:

- **Duration:** Both logarithmic and exponential transformations demonstrated promising results, with the logarithmic transformation slightly outperforming the exponential one.
- **Loudness:** Due to limitations, several transformations were inapplicable, and the remaining ones significantly underperformed. Addressing skewness in the 'Loudness' feature proved challenging.
- **Speechiness:** An exponential transformation yielded satisfactory results for the 'Speechiness' feature.
- **Acousticness:** The square-root transformation proved effective in reducing skewness in the 'Acousticness' feature.
- **Instrumentalness:** An exponential transformation was successful in mitigating skewness in the 'Instrumentalness' feature.
- **Liveness:** Similar to 'Instrumentalness,' an exponential transformation yielded favorable outcomes for the 'Liveness' feature.

For the remaining features, the applied transformations did not yield substantial improvements. As a result, these features were kept unchanged in the preprocessing phase.

Due to constraints against using external libraries to aid algorithm implementation, a custom scaling method was employed instead of Sklearn's Standard Scaler. Despite these challenges, a thorough study of the dataset guided data-driven decisions during the code implementation.

With the numerical features preprocessed, the data is now ready for subsequent stages, including feature selection and regression model training. These steps will enable us to make robust predictions for the target variable, 'popularity,' and derive meaningful insights from the regression analysis.



The dataset contains a set of `duration_ms` values, and the original skewness of these values is 10.81, indicating a significant right skew, with a longer tail on the right side. To mitigate this skewness, we applied several transformations to the data:

Logarithmic Transformation: Applying the logarithmic transformation resulted in a skewness of -0.32. The data became less skewed, with the right tail becoming shorter, and the distribution is now closer to a normal distribution.

Reciprocal Transformation: The skewness after the reciprocal transformation is 5.06. The right skew has decreased, but the distribution still shows some right-leaning characteristics.

Square Root Transformation: After the square root transformation, the skewness is 1.79. The distribution shows a reduction in right skewness compared to the original data.

Exponential Transformation: Applying the exponential transformation resulted in a skewness of 0.33. The data became slightly right-skewed, but the skewness is significantly reduced compared to the original data.

Figure 2: Skewness transformation of the 'duration' feature.

Evaluation of Numerical Features Regression

To ensure a comprehensive evaluation of the predictive performance of our model, we employed an 80-20 split between training and test data. Additionally, drawing inspiration from a previous project, we aimed not only to develop an efficient predictive model for song popularity but also to conduct an in-depth analysis of the effects of various preprocessing techniques on the overall project.

To facilitate a thorough comparison, we trained multiple models using distinct datasets, each processed with different preprocessing techniques:

1. **Unscaled Dataset**: The data was left in its original form without any scaling. While unscaled data is commonly considered disadvantageous for regression models due to varying feature scales, we were intrigued to explore its performance in our specific context.

2. **Standard Scaler**: Sklearn's Standard Scaler was utilized to scale the data, bringing all features to a common scale with zero mean and unit variance. This is a widely used technique in machine learning to improve convergence and performance.

3. **Manual Scaling**: In an effort to overcome constraints on using external libraries, we implemented a custom scaling method. This manual scaling aimed to achieve standardization and ensure comparability between numerical features.

4. **Skewness Correction**: Skewness is a common issue in continuous numerical features that can adversely impact model performance. To tackle this, we performed data scaling without skewness correction, aiming to assess its effect on model predictions.

Through this diverse preprocessing approach, we conducted a sample analysis of the results obtained. Notably, we observed that increasing the strength of the regularization parameter consistently led to a decline in model performance (Figure 4), with a significant drop for larger values, indicating potential heavy underfitting. To better optimize training parameters and ensure a well-generalized model, cross-validation will be employed to gain deeper insights.

An equally critical aspect of interest was the impact of the different preprocessing techniques applied to the data. Here are the key findings:

- **Best Performing Sets**: The manually and automatically scaled sets exhibited similar and impressive performances, with negligible differences, as shown in Table 1. This indicates that both custom scaling and Sklearn's Standard Scaler effectively prepared the data for modeling.

- **Unscaled Set**: Surprisingly, the unscaled set performed relatively well, showing similarity to the manually and automatically scaled sets in terms of predictive performance. However, it displayed erratic behavior as the regularization parameter increased, suggesting that scaling might still be beneficial for certain model configurations. This and an graphic representation of the results can be found in Figure 5.

- **Skewness Correction Set**: The set without skewness correction performed slightly worse than the manually and automatically scaled sets, but it followed a similar performance curve and evolution. This highlights the importance of addressing skewness in continuous features to optimize model predictions.

	Model	Lambda	MSE	ABMSE	R2_score
Lambda value: 0.01	Manual scale set	0.01	490.924803	18.473862	0.024508
	Automatic scale set	0.01	490.924803	18.473862	0.024508
Lambda value: 0.1	Manual scale set	0.1	490.924805	18.473862	0.024508
	Automatic scale set	0.1	490.924805	18.473862	0.024508
Lambda value: 1.0	Manual scale set	1.0	490.924826	18.473869	0.024508
	Automatic scale set	1.0	490.924826	18.473869	0.024508
Lambda value: 10.0	Manual scale set	10.0	490.925033	18.473937	0.024507
	Automatic scale set	10.0	490.925033	18.473937	0.024507
Lambda value: 100.0	Manual scale set	100.0	490.927148	18.474620	0.024503
	Automatic scale set	100.0	490.927147	18.474619	0.024503
Lambda value: 1000.0	Manual scale set	1000.0	490.952244	18.481364	0.024453
	Automatic scale set	1000.0	490.952227	18.481357	0.024453

Table 1: Comparison between method of scaling for different Lambda values

- **Removal of Correlated Features Set**: Unfortunately, the set where highly correlated features were removed performed the worst, just slightly worse than the skewness-corrected set, and displayed a similar structure and evolution in performance. This suggests that careful feature selection is crucial, as removing correlated features can negatively impact model accuracy.

To ensure the robustness of the findings, we adopted a ”double-check” approach, using double the number of values for the regularization parameter and employing different train-test splits. This verification process aimed to ensure the model's independence from element distribution and assess the impact of removing correlated features on performance.

The extensive evaluation of various model training techniques and preprocessing methodologies has provided invaluable insights into their effects on the predictive performance of the algorithm. These findings will significantly contribute to the achievement of our project goals and lead to informed decisions in further model development.

Categorical Features Preprocessing

In addition to the preprocessing applied to numerical features, the categorical features in the dataset also required careful handling. These categorical features, which depict classes or characteristics and possess only a limited number of unique values, were subjected to specific transformations.

Initially, the focus was directed toward those categorical features that were previously identified as discrete numerical values. For such features, numerical representations were used to depict their classes or characteristics. The first step in preprocessing involved formalizing the transformation of the 'explicit' feature, which was originally a boolean variable. It was converted into a binary numerical value to facilitate further processing. Additionally, the 'mode' feature, being binary as well, remained unchanged, as it represented the only other binary feature at this stage.

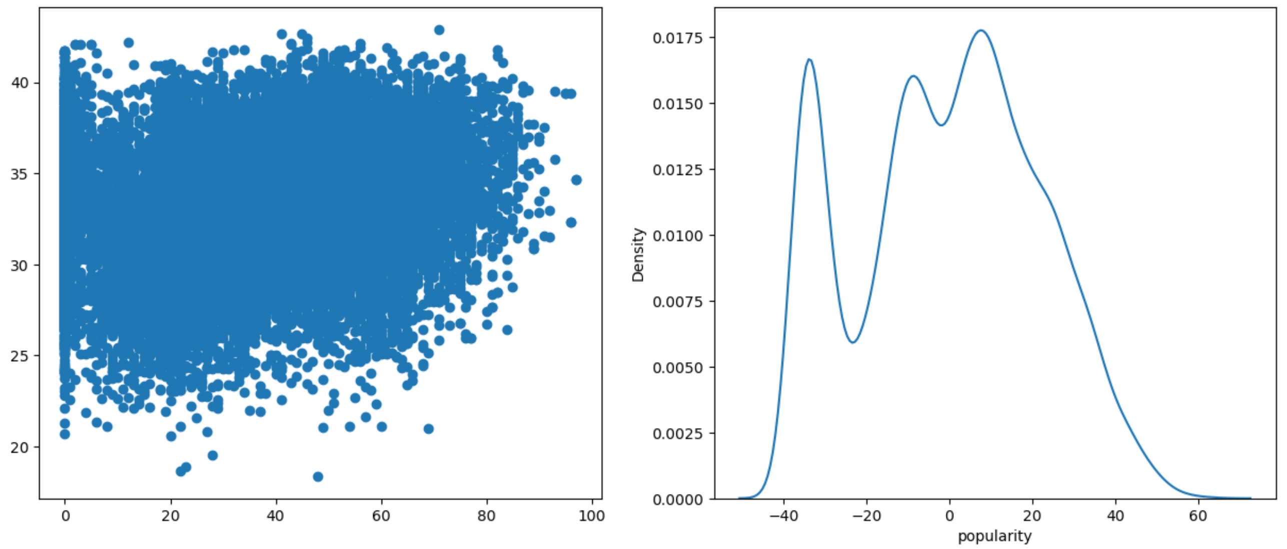


Figure 3: Representation of the numerical model.

Subsequently, the remaining discrete numerical values underwent scaling to ensure uniformity in their representation. Moving on to true categorical values, we conducted an analysis to determine the number of distinct unique values in each category. The objective was to identify optimal encoding techniques.

The analysis revealed the following number of unique values in each category:

- artists: 31,437
- album_name: 46,589
- track_name: 73,608
- track_genre: 114

Given the substantial number of unique entries in these categorical columns, we decided to use BaseN encoding as it appeared to be the most suitable approach. Attempting to use One Hot encoding faced challenges due to the extensive variety of values present in these categorical features. Hence, alongside BaseN encoding, we explored both Label encoding and Target encoding as alternative methods.

BaseN encoding is a method used to transform categorical data into numerical representations. It shares some conceptual similarities with One Hot encoding, but it operates differently. In BaseN encoding, the categorical data is divided into fixed-size bit chunks. Each chunk is then converted into its corresponding base-N representation. The value of N is determined by the number of unique categories in the categorical feature. Each symbol in the encoded data represents a value from 0 to N-1, effectively creating a sequence of numerical values that represent the original categories.

Label encoding, on the other hand, is a simpler approach. In Label encoding, each distinct category in the categorical feature is assigned a unique integer value. These integer values range from 0

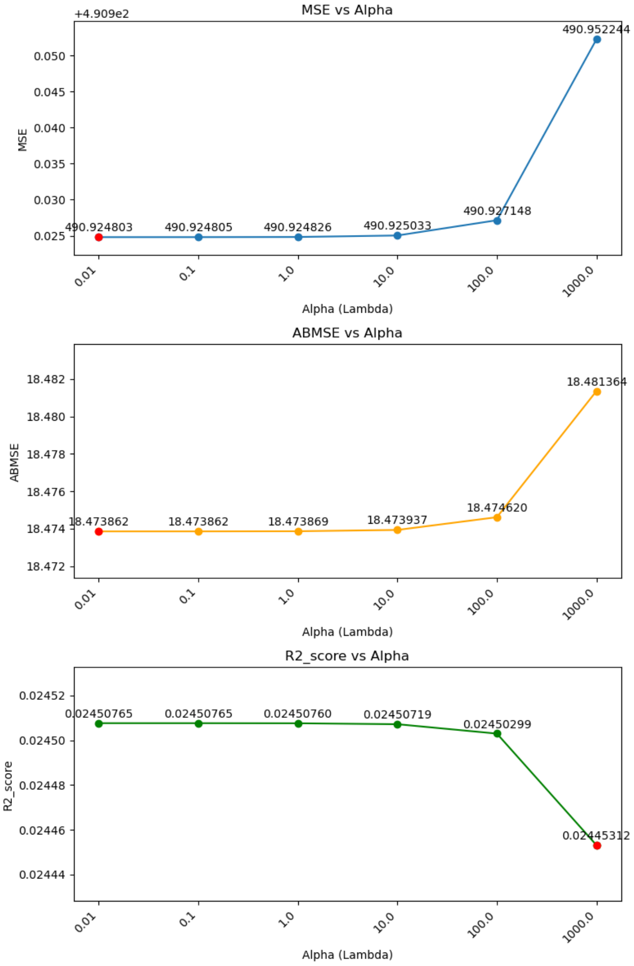
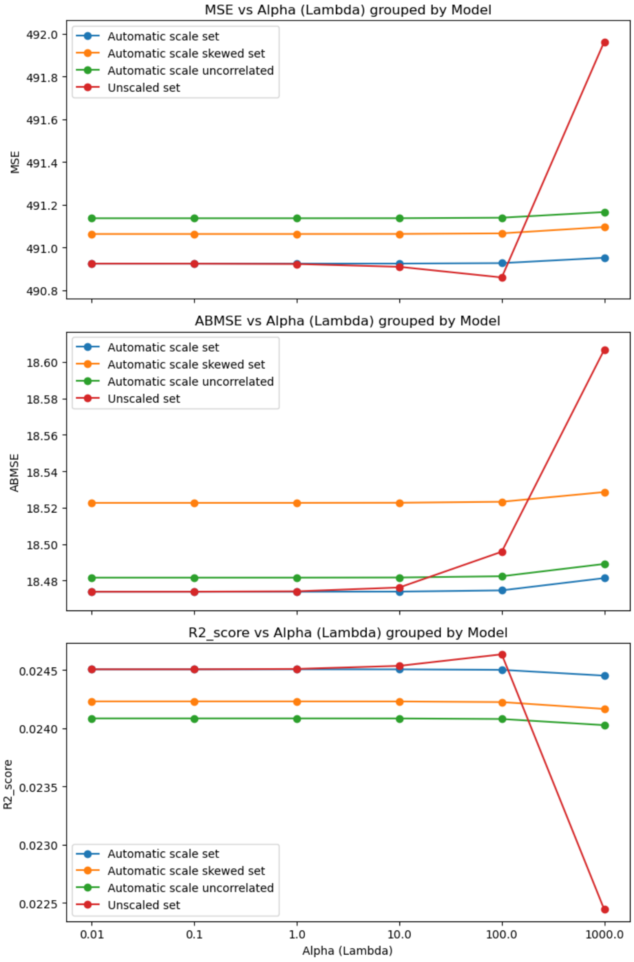


Figure 4: Visualization of the algorithm's strictly numerical result scores.

to N-1, where N represents the total number of unique categories. The encoded data now consists of integer labels instead of the original categorical values.

Target encoding, as a unique method, replaces each category of the categorical variable with the mean value of the target variable (e.g., popularity). This technique directly captures the statistical relationship between the categorical feature and the target. For instance, if a category appears more frequently and is associated with higher target values (e.g., high popularity), then its target-encoded value will reflect this tendency.

As a final refinement step, the values resulting from Label and Target encoding underwent scaling. This capitalized on their newfound numerical attributes, enhancing their efficacy in subsequent analysis and modeling phases.



The manually scaled model is omitted as it closely resembles the automatically scaled model, with only negligible differences apparent from an extremely close distance.

Figure 5: Performance comparison of various techniques on the algorithm.

Evaluation of Categorical Features Regression

Similar to the preprocessing approach used for the numerical features in the Ridge Regression analysis, the categorical features in the dataset were processed based on different encoding methods. The datasets were divided accordingly, and the fit and prediction processes were performed twice for each dataset – once with the complete set of features and once after removing correlated features.

The results of the study indicated that BaseN encoding outperformed strictly numerical encoding, showing conceptual alignment with One Hot encoding. Despite this advantage, the overall curve and behavior of the model remained consistent, which is a promising sign of its reliability.

Interestingly, the increase in the regularization parameter showed a direct proportionality to the error. However, some notable discrepancies in the results warranted further investigation into the intricacies of the encoding process.

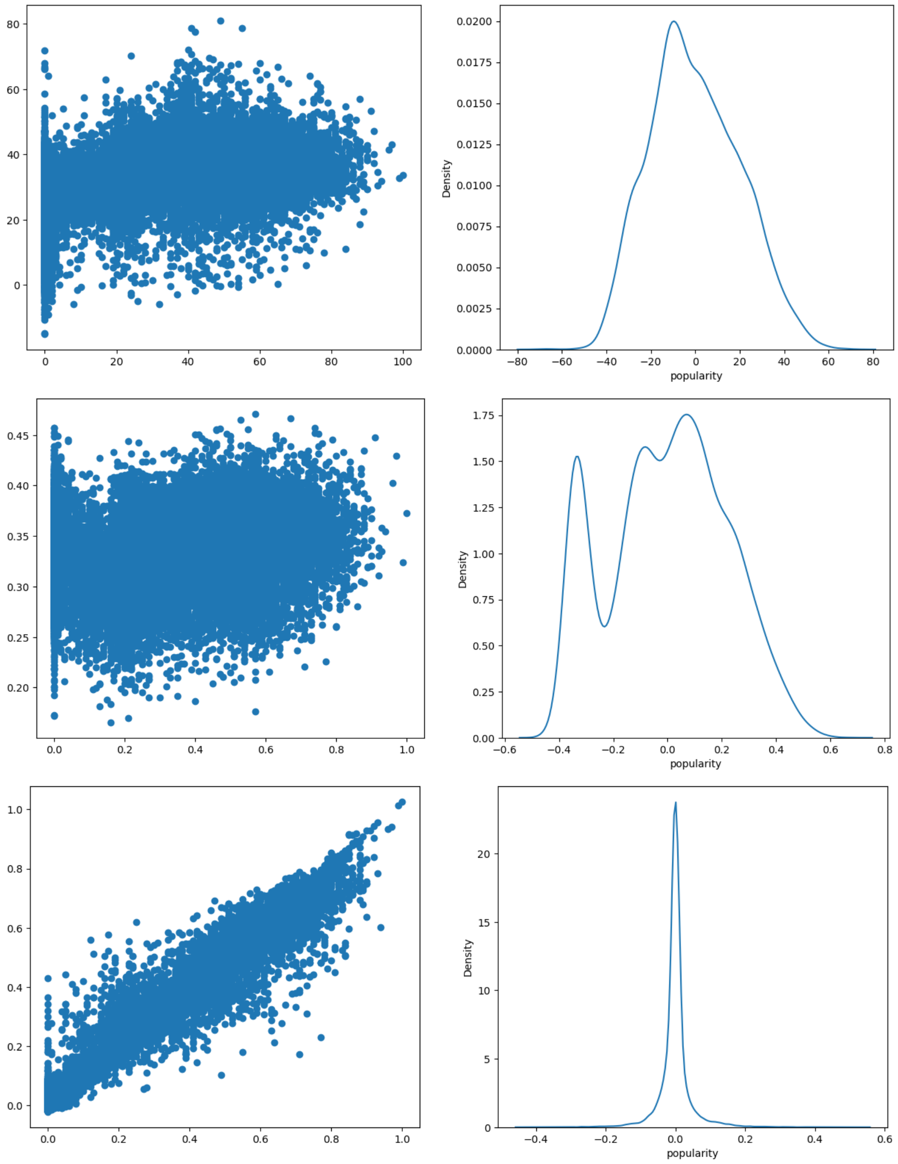


Figure 6: Representation of the different categorical models.

BaseN encoding offers compelling advantages, including reduced dimensionality for handling high-dimensional datasets and the preservation of non-ordinal representations. Its flexibility in selecting bases also allows for human-readable encoding schemes, enhancing model interpretability. However, it is crucial to consider its limitations, such as sensitivity to scaling and potential representation bias, which could impact its performance.

Unsurprisingly, in this study, the worst-performing technique was Label encoding, having a performance achievable with a good distribution of the training and validation set of the strictly numerical set. The results can be found in Table 2.

A significant and intriguing finding emerged when comparing BaseN encoding to Label and Target encoding. Both BaseN and Label encoding, while having big differences in performance, were close to the original results of the numerical set, with BaseN being a clear improvement. Yet, Target encoding stood out by showcasing the lowest Mean Squared Error and Average Bias-corrected Mean Squared Error, indicating its remarkable predictive capabilities, way above the other two encoding methods.

Lambda value	Model	Lambda	MSE	ABMSE	R2_score
0.01	Base-N encoding	0.01	0.041907	0.168533	0.167277
	Target encoding	0.01	0.002039	0.024356	0.959487
	Label encoding	0.01	0.048859	0.184199	0.029150
0.1	Base-N encoding	0.1	0.041907	0.168534	0.167278
	Target encoding	0.1	0.002039	0.024357	0.959487
	Label encoding	0.1	0.048859	0.184199	0.029150
1.0	Base-N encoding	1.0	0.041907	0.168537	0.167281
	Target encoding	1.0	0.002039	0.024363	0.959484
	Label encoding	1.0	0.048859	0.184202	0.029153
10.0	Base-N encoding	10.0	0.041906	0.168576	0.167311
	Target encoding	10.0	0.002041	0.024426	0.959445
	Label encoding	10.0	0.048857	0.184228	0.029180
100.0	Base-N encoding	100.0	0.041911	0.168945	0.167216
	Target encoding	100.0	0.002104	0.025171	0.958186
	Label encoding	100.0	0.048855	0.184466	0.029226
1000.0	Base-N encoding	1000.0	0.042528	0.172065	0.154957
	Target encoding	1000.0	0.003258	0.032875	0.935253
	Label encoding	1000.0	0.049021	0.185807	0.025925

Table 2: Comparison of different encoding methods performance.

While Label encoding is praised for its simplicity, memory efficiency, and preservation of ordinal information, Target encoding retains the categorical nature of the data, can handle high cardinality data, and captures information from the target variable, making it a powerful option for transforming categorical features.

The differences between encoding methods became evident when examining the R-squared Score. Target encoding, by generating labels based on the target feature, established a high correlation between all categorical features and the target. As a result, the scores were close to one, indicating a better model fit to the data and approaching a normal distribution. On the other hand, the Label encoding method generated entirely arbitrary values, resulting in scores close to zero (approximately around 0.02), potentially due to the introduced ordinality bias that renders it unsuitable for nominal data, yet very close to the original results. BaseN, in this section, shows a clear and strong improvement, yet not remotely close to Target encoding, but performing much better than its Label counterpart (with values around 0.1).

While Target encoding offers numerous advantages, it also carries certain risks, such as data leakage, potential overfitting, and possible loss of information for rare categories. Therefore, a careful evaluation and consideration of the specific data and use case are essential when selecting the appropriate encoding method for machine learning applications.

5-Fold Cross Validation for Ridge Regression Optimization

In this section, we present the results obtained from applying 5-fold cross-validation to optimize the regularization parameter in the context of ridge regression. The previous runs of the algorithm hinted at a significant correlation between the performance of the dataset and the proximity of the regularization parameter to zero. As observed, the ridge regression demonstrated improved performance as the regularization parameter approached zero, revealing a clear monotonic increasing relationship. However, to ensure the reliability and generalizability of our findings, we decided to conduct a comprehensive 5-fold cross-validation.

Cross-validation is a widely recognized technique used to estimate the performance of a machine-learning model and mitigate overfitting. In our study, we employed 5-fold cross-validation, dividing the dataset into five subsets, or "folds," where four folds were used for training the ridge regression model, and the remaining fold was used for validation. This process was iterated five times, with each fold serving as the validation set once. The final performance metric was then computed as the average of the results obtained from each iteration.

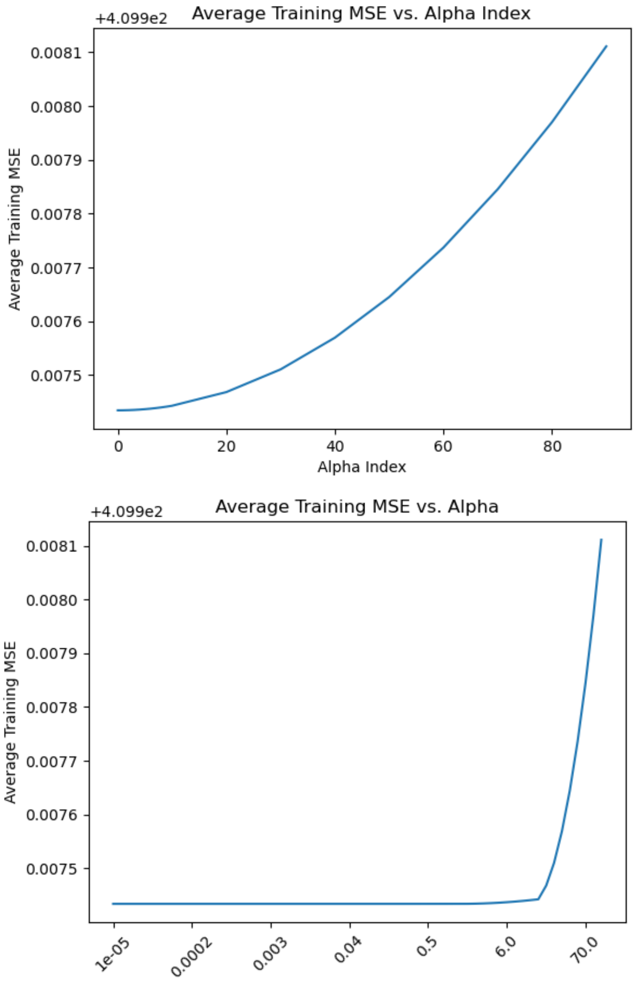
The outcomes of the cross-validation align with our initial observations: the ridge regression algorithm consistently performed better with lower values of the regularization parameter (Figure ??), to be specific all methods of encoding coincide that the optimal value of Lambda is 1e-05, the lowest values available. This finding reinforces the significance of regularization in preventing overfitting and improving the generalization capability of the model.

However, it is worth noting that we encountered some unexpected challenges during the execution of cross-validation, which warrants further discussion in a subsequent section. These challenges could provide valuable insights for future researchers attempting to optimize ridge regression through cross-validation.

Moving on to the optimization of training parameters, our focus was primarily on the regularization parameter since it was the sole scalable parameter in this context. To find the optimal value for this parameter, we utilized the grid search technique, a widely adopted and effective method for hyperparameter tuning. Grid search involves systematically evaluating the model's performance across a predefined range of hyperparameter values.

While grid search can become computationally expensive when dealing with larger hyperparameter grids, we benefited from the simplicity and efficiency of this method, given that we had only one parameter to optimize. Moreover, since there was only one parameter, the computational burden associated with grid search was considerably mitigated, allowing us to efficiently explore the parameter space and make informed decisions regarding the regularization strength.

Throughout the project, we also relied on data visualization techniques, such as plotting graphs, to gain deeper insights into the behavior of the ridge regression model. These visualizations proved invaluable in understanding the relationship between the regularization parameter and the performance of the algorithm, aiding us in making informed choices during the optimization process.



The first graph displays points separated by distance, while the second graph presents the values of lambda (alpha) as labels, offering two distinct perspectives of performance evolution. Each value is obtained by multiplying 0.00001, 0.0001, 0.001, 0.01, 0.1, 1, and 10 by all numbers from 1 to 9 (inclusive).

Figure 7: Visualization of the performance of various training parameters.

Cross-validation Challenges and Resolution

During the course of this project, an unexpected challenge arose while performing cross-validation, particularly when applying different encoding techniques to the dataset. Prior to this stage, all computations proceeded without encountering any abnormal values. However, upon conducting cross-validation, certain folds of the BaseN encoding set resulted in negative R-squared scores, while nearly all folds of the Label encoding exhibited similar anomalies (Figure 8). Surprisingly, the Target encoding remained unaffected by these issues.

To identify the root cause of these discrepancies, a thorough investigation of the project's progress

Metric	Train	Test
Mean Squared Error (MSE)	0.04099074339547423	0.04102385598424115
Mean Absolute Error (MAE)	0.16645544123070782	0.16652539955629625
R-Squared (R2)	0.1745137722669562	0.1738225792869393
Root Mean Squared Error (RMSE)	0.20246163018590893	0.2025422159618427

Metric	Train	Test
Mean Squared Error (MSE)	0.04821004910991263	0.04823098882311951
Mean Absolute Error (MAE)	0.18256516081374702	0.18260720908307554
R-Squared (R2)	0.029129350883897388	0.02868710922606179
Root Mean Squared Error (RMSE)	0.21956783118083822	0.21961494207521876

Metric	Train	Test
Mean Squared Error (MSE)	0.002019961856836152	0.0020208635714078194
Mean Absolute Error (MAE)	0.024260573337500417	0.02426523597337773
R-Squared (R2)	0.9593211713301535	0.9593001086447591
Root Mean Squared Error (RMSE)	0.04494386681680768	0.044952137705841194

The first table corresponds to Base-N, more specifically Base-10 encoding. The second table corresponds to Label encoding. The third table corresponds to Target encoding.

Table 3: Performance Metrics of Base-N encoding

and code execution was undertaken. The first hypothesis considered was the possibility of an error during the adaptation of the code to the cross-validation estimator. However, a comparison of the weight matrices (thetas) between the adapted code for Ridge Regression and the original algorithm revealed identical results, ruling out any issues related to the code adaptation.

The next hypothesis centered on the discrepancy arising from the usage of a different function to measure the R-squared score within the cross-validation function. To validate this assumption, the index of the folds was independently generated, and the fit and prediction were performed manually. The R-squared error was then calculated using a standalone function, and the results proved to be consistent with those obtained during the cross-validation, ruling out the second hypothesis as well.

The breakthrough in understanding the root cause of the problem emerged during the investigation of the fold indexes used by the cross-validation function. It was observed that the function tended to group or cluster large numbers of rows together. At this point, a crucial distinction was made, as the only encoding technique unaffected by the issue was Target encoding. In contrast, Label encoding, and to a lesser extent BaseN encoding, exhibited a strong correlation between the values used to replace categorical features and the index or row number.

In the case of Label encoding, arbitrary values are generated for unique categorical values, and the proximity of these values to the beginning of the dataset resulted in values closer to 1. As BaseN encoding also introduced some level of proximity-related values, it occasionally encountered issues when a fold encapsulated clusters of similar values, leading to discrepancies between the training and

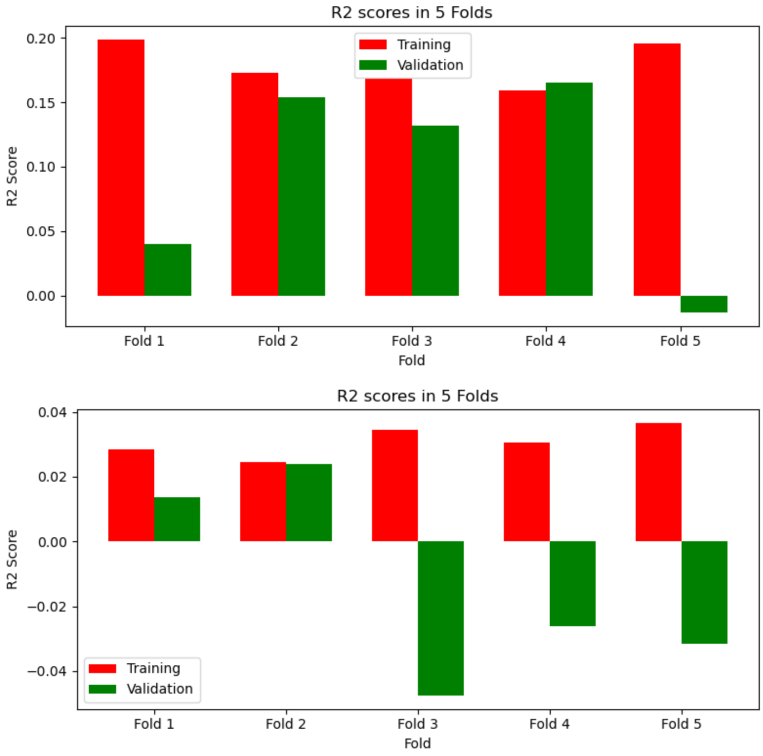


Figure 8: Anomalous Cross-Validation Results with BaseN and Label Encoding.

validation sets.

Fortunately, the nature of Target encoding, which derives values in relation to the target feature, rendered it impervious to the problem observed in the other encoding techniques.

To address and resolve this issue, a simple yet effective solution was implemented. Prior to conducting cross-validation, the database was shuffled, thereby preventing any complete encapsulation of clusters within either the training or validation sets. This approach ensured a more representative distribution of data across folds and allowed for a fair and unbiased evaluation of the model's performance during the cross-validation process, the result of this method can be appreciated on Table 3.

Conclusion

In conclusion, the comprehensive study of the dataset and the application of various preprocessing techniques have shed light on the optimal approach to building an efficient predictive model for song popularity. The analysis of numerical and categorical features provided valuable insights into the dataset's characteristics and revealed the importance of careful feature selection and engineering to improve model performance.

By implementing ridge regression with 5-fold cross-validation, we were able to optimize the regularization parameter and develop a well-generalized model. The results demonstrated that lower values of the regularization parameter consistently led to improved model performance, reinforcing the importance of regularization in preventing overfitting.

The study also investigated the impact of different encoding techniques on model performance. Target encoding emerged as the most powerful method, offering superior predictive capabilities compared to BaseN and Label encoding. The challenges encountered during cross-validation were identified and resolved, enhancing the reliability of the results.

In conclusion, this report presents a robust foundation for developing an efficient predictive model for song popularity. The findings and methodologies will contribute to advancements in machine learning applications and aid in informed decisions for subsequent model development.