Spotify Tracks Popularity

Oumaima El Menni

September 2023

1 Introduction

In this project, we delve into the realm of Spotify track popularity with an objective lens. Our primary aim is to explore the intricate web of factors influencing a track's success through the application of machine learning techniques, specifically Ridge Regression.

This endeavor is rooted in the pursuit of data-driven insights. We analyze a diverse set of features, including audio attributes and lyrical characteristics, to predict and understand track popularity trends on Spotify. Beyond its relevance in the music industry, our study holds potential implications for recommendation systems and audience engagement on streaming platforms.

2 Dataset

Our dataset contains 20 numerical and categorical variables. The numerical variables have different ranges that we summarize in the following table:

Variable	Range
popularity	0 - 100
duration-ms	0 - 5237295
danceability	0.0 - 0.985
energy	0.0 - 1.0
key	0 - 11
loudness	-49.531 - 4.532
mode	0 - 1
speechiness	0.0 - 0.965
acousticness	0.0 - 0.996
instrumentalness	0.0 - 1.0
liveness	0.0 - 1.0
valence	0.0 - 0.995
tempo	0.0 - 243.372
time-signature	0.0 - 5

We have two categorical variables: 'explicit' and 'track-genre'. The first variable is binary: it takes the value 'True' if the track has explicit content, and 'False' otherwise. The variable 'track-genre' takes **114** different values.

2.1 Features density

In this section, we want to see the distribution of the different variables:

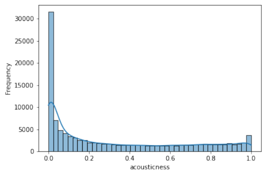


Figure 1: (a) Acousticness Density

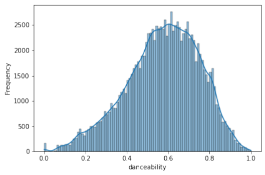


Figure 2: (b) Danceability Density

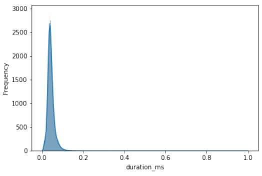


Figure 3: (c) Duration Density

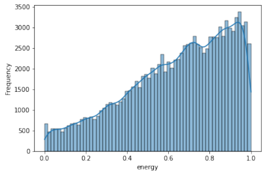


Figure 4: (d) Energy Density

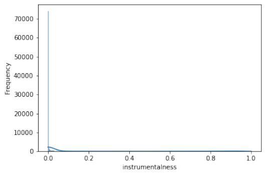


Figure 5: (e) Instrumentalness Density

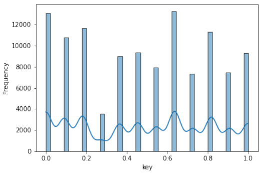


Figure 6: (f) Key Density

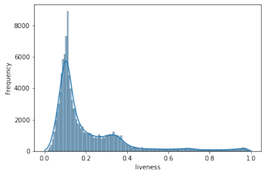


Figure 7: (g) Liveness Density

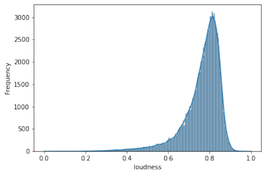


Figure 8: (h) Loudness Density

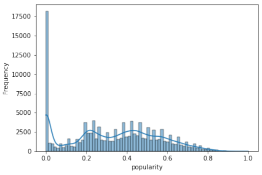


Figure 9: (i) Popularity Density

The variables exhibit different distribution shapes, which is understandable since diverse underlying factors affect each variable.

2.2 Bivariate analysis

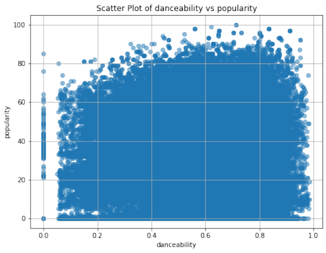


Figure 10: *
(a) danceability vs popularity

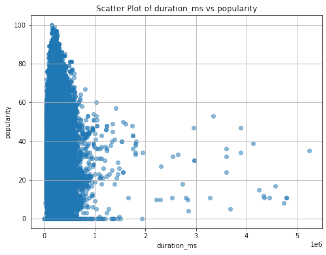


Figure 11: *
(b) duration-ms vs popularity

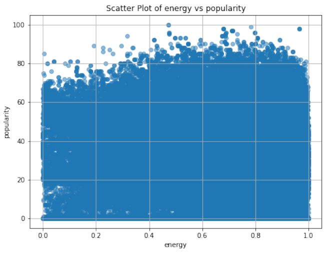


Figure 12: *
(c) energy vs popularity

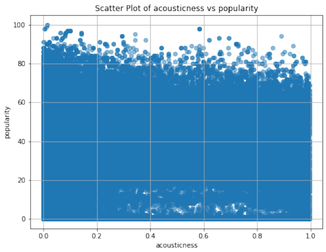


Figure 13: *
(d) acousticness vs popularity

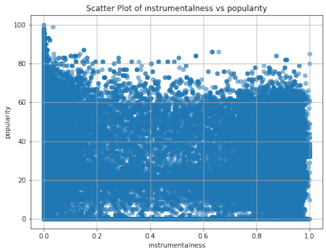


Figure 14: *
(e) instrumentalness vs popularity

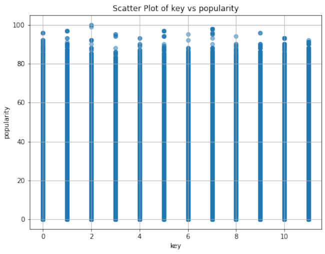
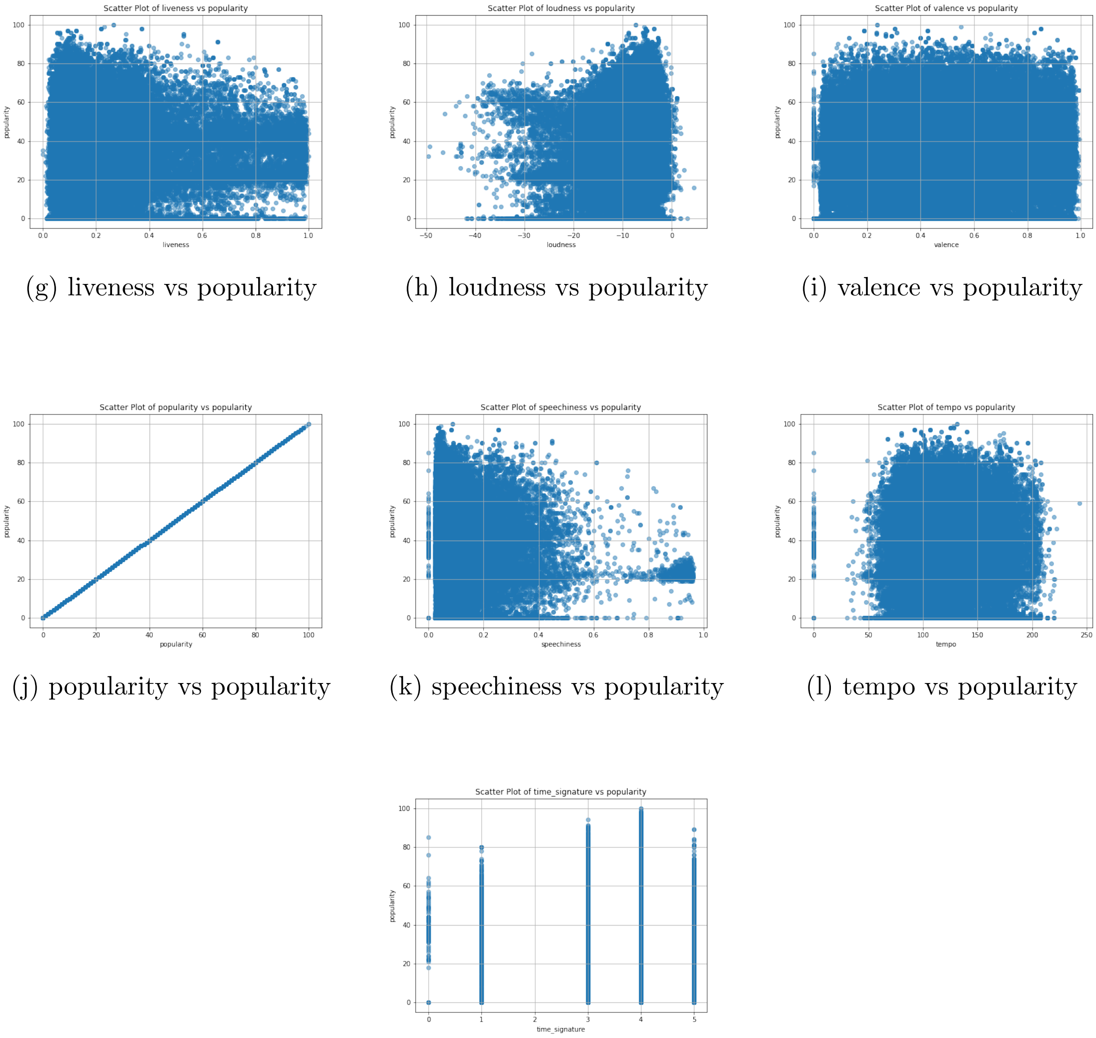


Figure 15: *
(f) key vs popularity



(g) liveness vs popularity

(h) loudness vs popularity

(i) valence vs popularity

(j) popularity vs popularity

(k) speechiness vs popularity

(l) tempo vs popularity

(m) time signature vs popularity

Figure 16: Bivariate analysis

Comments on the scatter plots:

- There is a negative correlation between popularity and duration.
- There is a positive correlation between the loudness of the track and its popularity.
- We notice a negative correlation between speechiness and popularity.

However, these correlations remain low.The correlation heatmap confirms this:

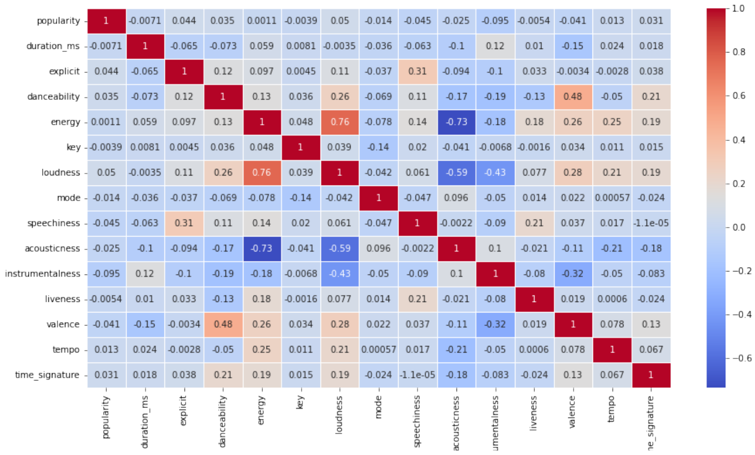


Figure 17: Correlation Heatmap for Numerical variables

3 Data preprocessing

In terms of data preprocessing, we will apply two types of normalization: z-score normalization and Min-Max scaling. The objective is to investigate whether these transformations enhance the results of ridge regression.

Z-Score Normalization: Z-score normalization transforms data by centering it around zero and scaling it to have a standard deviation of one. The formula used is:

$$z = \frac{x - \mu}{\sigma}$$

where x is an individual data point, μ is the mean, and σ is the standard deviation. In ridge regression, this ensures all features contribute equally, leading to more balanced regularization and potentially improved model performance.

Min-Max Scaling: Min-Max scaling rescales data to a fixed range. In our case, it's $[0, 1]$. The formula is:

$$x' = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}$$

where x is an individual data point, x_{min} is the minimum, and x_{max} is the maximum value of the feature. This normalization ensures comparable feature scales in ridge regression, improving model convergence and enhancing predictive accuracy.

4 Ridge Regression

To analyze the relationship between the popularity of a track, and our variables, we are going to use the Ridge Regression Algorithm. The Ridge Regression is a linear regression algorithm that helps mitigate multicollinearity and reduce the risk of overfitting in a model.

In Ridge Regression, we aim to minimize the following objective function:

$$J(\mathbf{w}) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i\mathbf{w})^2 + \frac{\alpha}{2}\|\mathbf{w}\|^2$$

where:

- $\mathbf{w}$ is the vector of coefficients (weights) we want to learn.
- y_i is the true target value for the i-th sample.
- $\mathbf{x}_i$ is the feature vector for the i-th sample.
- n is the number of samples.
- α is the regularization parameter that controls the amount of shrinkage applied to the coefficients.
- $\|\mathbf{w}\|^2$ is the squared Euclidean norm of the coefficient vector, which acts as the regularization term.

The solution to this optimization problem is given by:

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

where:

- $\mathbf{X}$ is the matrix of input features.
- $\mathbf{y}$ is the vector of target values.
- $\mathbf{I}$ is the identity matrix of appropriate dimensions.

We implement the algorithm in Python as follows:

```
class RidgeRegression(BaseEstimator):
    def __init__(self, alpha=1.0):
        self.alpha = alpha
        self.coef_ = None

    def fit(self, X, y):
        n_samples, n_features = X.shape

        # Adding regularization term to the loss function
        XtX = X.T @ X
        reg_term = self.alpha * np.eye(n_features)
        self.coef_ = np.linalg.inv(XtX + reg_term) @ X.T @ y

        return self

    def predict(self, X):
        if self.coef_ is None:
            raise ValueError("Model␣has␣not␣been␣fitted␣yet.")

        if X.shape[1] != self.coef_.shape[0]:
            raise ValueError("Number␣of␣features␣in␣X␣does␣not␣match␣the␣number␣
               of␣coefficients.")
        # Ensuring that X is a NumPy array
        X = np.array(X)

        y_pred = X @ self.coef_
        return y_pred
```

We will experiment the impact of giving different values to the regularization parameter **alpha** on the error metric: R$^{\mathbf{2}}$.

R$^{\mathbf{2}}$ (R-squared): It provides an indication of how well the regression model fits the observed data compared to a simple horizontal line (the mean of the target variable). It measures the proportion of the variance in the dependent variable that is explained by the independent variables in the model.

5 Procedure

In this analysis, we initiated by splitting the data into training, validation, and test sets for each dataframe using a custom function, ensuring distinct datasets for model training, hyperparameter tuning, and final evaluation. Subsequently, we employed cross-validation on the training set to optimize the regularization parameter (alpha) for Ridge Regression, utilizing a grid search strategy to identify the optimal alpha value. During cross-validation, the dataset was segmented into 5 folds, and the model's performance was assessed on each fold to approximate its generalization capability. After determining the best alpha value for each dataframe, we trained the final Ridge Regression model on the combined training and validation sets using the optimal alpha. We then calculated the R$^{\mathbf{2}}$ scores for the test set to evaluate the model's performance on unseen data. Additionally, we performed direct fitting without cross-validation to compare its results with those obtained from the cross-validation approach. Finally, we organized the results, including the best alpha values and test set R$^{\mathbf{2}}$ scores, into dataframes for each approach, providing insights into the model's performance on unseen data and facilitating further analysis and interpretation.

This procedure was done for the dataset with only numerical variables, and the dataset with categorical variables included.

5.1 Results For Numerical Variables

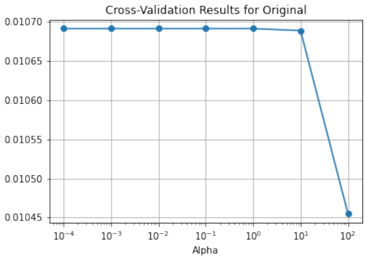
5.1.1 Cross Validation

Following are the results for the cross-validation

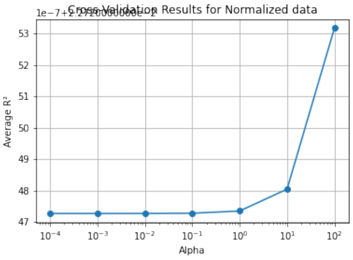
DataFrame	Alpha	Mean Test R²
Original	0.0001	0.010691
Original	0.001	0.010691
Original	0.01	0.010691
Original	0.1	0.010691
Original	1	0.010691
Original	10	0.010689
Original	100	0.010455
Normalized data	0.0001	0.022725
Normalized data	0.001	0.022725
Normalized data	0.01	0.022725
Normalized data	0.1	0.022725
Normalized data	1	0.022725
Normalized data	10	0.022725
Normalized data	100	0.022725
Scaled data	0.0001	0.016960
Scaled data	0.001	0.016960
Scaled data	0.01	0.016960
Scaled data	0.1	0.016960
Scaled data	1	0.016961
Scaled data	10	0.016963
Scaled data	100	0.016628

Table 2: Results for different alpha values

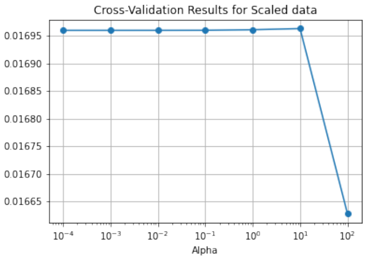
To better visualize the results:



(a) CV Results for Original Data



(b) CV Results for Normalized Data



(c) CV Results for Scaled Data

The best alphas based on the Cross Validation are:

DataFrame	Best Alpha	Test R²
Original	1	0.010691
Normalized data	100	0.022725
Scaled data	10	0.016963

Table 3: Results for different dataframes

Despite these optimal alpha values, the Mean Test R² values remained relatively low across all datasets. This suggests that the Ridge Regression model, trained solely on numerical variables, struggled to capture the underlying patterns in the data effectively.

5.1.2 Final model Training and Evaluation

DataFrame	Best Alpha	Test R^2
Original	1	0.009514
Normalized data	100	0.020683
Scaled data	10	0.015324

Table 4: Results for different dataframes

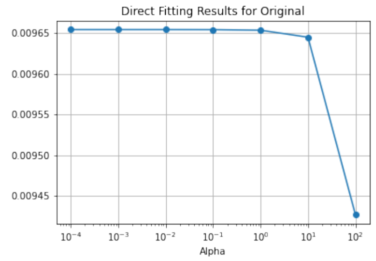
Interestingly, although the models showed slightly lower R^2 scores on the test sets compared to cross-validation, the trend remained consistent. This suggests that the models perform reasonably well on new data, even though their predictive power is modest.Also, we notice a slight better performance on Normalized and scaled datasets.

5.1.3 Direct Fitting Without Cross-Validation

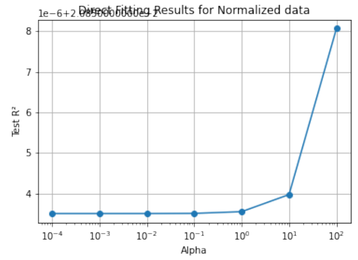
DataFrame	Alpha	Mean Test R^2
Original	0.0001	0.009655
Original	0.0010	0.009655
Original	0.0100	0.009655
Original	0.1000	0.009655
Original	1.0000	0.009654
Original	10.0000	0.009645
Original	100.0000	0.009427
Normalized data	0.0001	0.020854
Normalized data	0.0010	0.020854
Normalized data	0.0100	0.020854
Normalized data	0.1000	0.020854
Normalized data	1.0000	0.020854
Normalized data	10.0000	0.020854
Normalized data	100.0000	0.020858
Scaled data	0.0001	0.015449
Scaled data	0.0010	0.015449
Scaled data	0.0100	0.015449
Scaled data	0.1000	0.015449
Scaled data	1.0000	0.015450
Scaled data	10.0000	0.015460
Scaled data	100.0000	0.015355

Table 5: Results for different alpha values

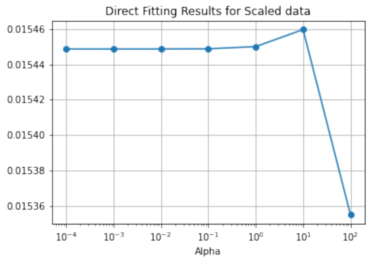
The best alphas based on the Direct Fitting are:



(a) CV Results for Original Data



(b) CV Results for Normalized Data



(c) CV Results for Scaled Data

DataFrame	**Best Alpha**	**Test R²**
Original	0.0001	0.009655
Normalized data	100	0.020858
Scaled data	10	0.015460

Table 6: Results for different dataframes

For the direct fitting results, we observe slightly different best alpha values compared to cross-validation and final model training and evaluation, notably for the original and scaled datasets. However, for the normalized dataset, the best alpha value aligned with those found in cross-validation and final model assessments, indicating consistency across various evaluation approaches. Despite these differences, the models generally demonstrated similar predictive accuracy, as reflected in the R^2 scores on the test sets, suggesting overall stability in performance across different methodologies. Additionally, the model still performs better on normalized and scaled dataframes than the original one, highlighting the importance of data preprocessing in enhancing model performance.

5.2 Results After Including Categorical variables

In this section, we'll be adding two categorical variables: `track_genre` and `explicit`.

The two variables were encoded using different techniques to prepare them for our model. `explicit`, denoting whether a track contains explicit content, was encoded using `LabelEncoder`, assigning binary values (0 or 1) to represent the absence or presence of explicit content, respectively. Meanwhile, `track_genre`, indicating the genre of the track, was encoded using `TargetEncoder`, which replaced each category label with the mean popularity of tracks within that genre. `TargetEncoder`'s smoothing parameter was set to 10 to handle categories with limited observations, preventing overfitting to rare categories while capturing the relationship between genre and popularity.

After encoding the categorical variables, the dataset has undergone the same preprocessing methods applied to the numerical variables. We applied both z-score and Min Max scaling to compare the results of the original dataset and these transformed datasets.

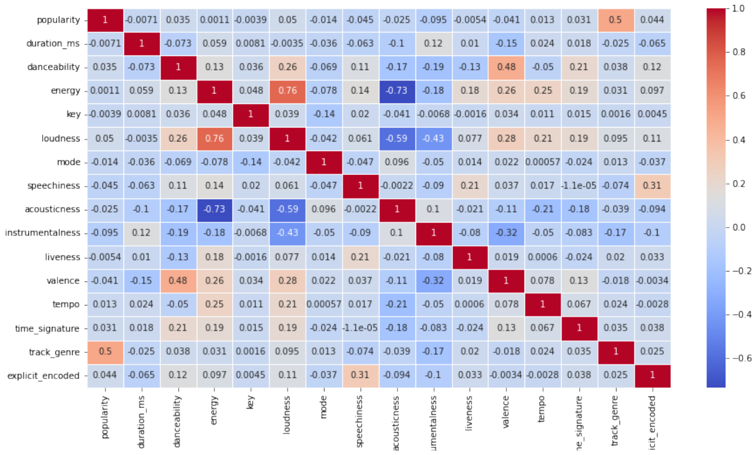


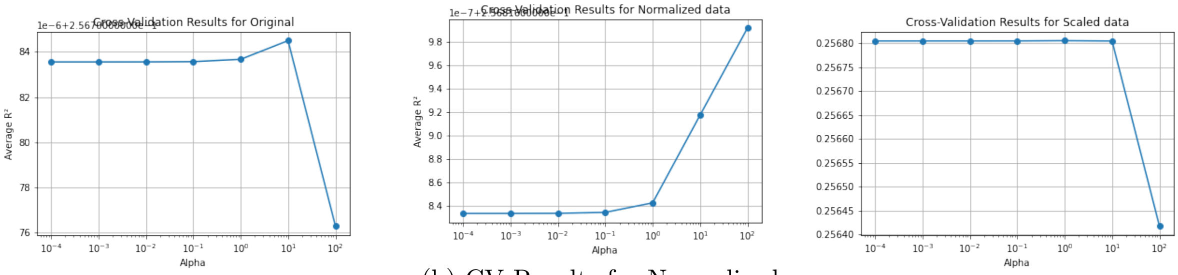
Figure 18: Correlation Heatmap for Numerical Variables After Including Categorical Variables

Adding the two categorical variables, we notice a relatively high correlation between popularity and track_genre, we expect it to enhance the regression results.

5.2.1 Cross Validation

DataFrame	**Alpha**	**Mean Test R²**
Original	0.0001	0.256784
Original	0.001	0.256784
Original	0.01	0.256784
Original	0.1	0.256784
Original	1	0.256784
Original	10	0.256784
Original	100	0.256776
Normalized data	0.0001	0.256817
Normalized data	0.001	0.256817
Normalized data	0.01	0.256817
Normalized data	0.1	0.256817
Normalized data	1	0.256817
Normalized data	10	0.256817
Normalized data	100	0.256817
Scaled data	0.0001	0.256805
Scaled data	0.001	0.256805
Scaled data	0.01	0.256805
Scaled data	0.1	0.256805
Scaled data	1	0.256806
Scaled data	10	0.256805
Scaled data	100	0.256418

Table 7: Results for different alpha values



(a) CV Results for Original Data

(b) CV Results for Normalized Data

(c) CV Results for Scaled Data

The best alphas based on the Cross Validation are:

DataFrame	**Best Alpha**	**Test R²**
Original	10	0.256784
Normalized data	100	0.256817
Scaled data	1	0.256806

Table 8: Results for different dataframes

The inclusion of categorical variables led to a substantial enhancement in model performance, as evidenced by notably higher R^2 scores compared to the previous analysis focused solely on numerical variables. Also, the enhancement of R^2 scores across preprocessed dataframes is consistent.

5.2.2 Final model Training and Evaluation

DataFrame	**Best Alpha**	**Test R²**
Original	10	0.259830
Normalized data	100	0.259773
Scaled data	1	0.259771

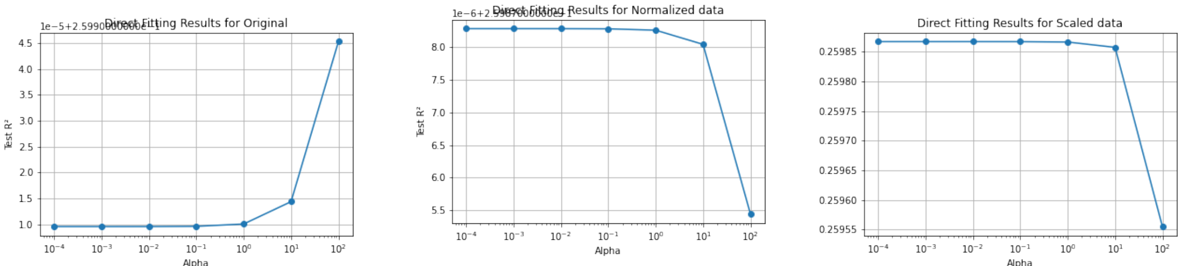
Table 9: Results for different dataframes

The Test R^2 values slightly increased for all dataframes compared to the cross-validation results, suggesting a small improvement in model performance when combining the training and validation sets for the final model fitting. Notably, the normalized and scaled dataframes still yielded higher performance compared to the original dataframe, underscoring the effectiveness of data normalization and scaling in improving the model's predictive accuracy. This consistency in alpha values across both evaluation stages also reinforces the reliability of the cross-validation process in selecting appropriate hyperparameters.

5.2.3 Direct Fitting Without Cross-Validation

DataFrame	Alpha	Mean Test R²
Original	0.0001	0.259910
Original	0.0010	0.259910
Original	0.0100	0.259910
Original	0.1000	0.259910
Original	1.0000	0.259910
Original	10.0000	0.259914
Original	100.0000	0.259945
Normalized data	0.0001	0.259878
Normalized data	0.0010	0.259878
Normalized data	0.0100	0.259878
Normalized data	0.1000	0.259878
Normalized data	1.0000	0.259878
Normalized data	10.0000	0.259878
Normalized data	100.0000	0.259875
Scaled data	0.0001	0.259867
Scaled data	0.0010	0.259867
Scaled data	0.0100	0.259867
Scaled data	0.1000	0.259867
Scaled data	1.0000	0.259866
Scaled data	10.0000	0.259857
Scaled data	100.0000	0.259555

Table 10: Results for different alpha values



(a) CV Results for Original Data

(b) CV Results for Normalized Data

(c) CV Results for Scaled Data

The best alphas based on the Direct Fitting are:

DataFrame	Best Alpha	Test R²
Original	100	0.259945
Normalized data	0.0001	0.259878
Scaled data	0.0001	0.259867

Table 11: Results for different dataframes

The direct fitting results show that the models performed consistently well across different datasets. Even though the best alpha values weren't exactly the same as those from cross-validation, the modelsstill predicted test data quite accurately. This means that our model, even when directly fitted, did a good job of capturing the patterns in the data, giving us reliable predictions. (We can consider the results reliable because they consistently produced high R^2 values across different datasets.)

6 Summary

In this project, we evaluated the performance of Ridge Regression models using two distinct approaches: one with only numerical variables and another incorporating categorical variables. The inclusion of categorical variables was achieved through appropriate encoding techniques, enhancing the model's predictive power.

Cross-Validation Results

- For the model using only numerical variables, the best alphas were 1, 100, and 10 for the original, normalized, and scaled datasets, respectively, with relatively low Mean Test R^2 values.
- When categorical variables were included, the best alphas shifted to 10, 100, and 1 for the original, normalized, and scaled datasets, respectively, with significantly higher Mean Test R^2 values, indicating better model performance.

Final Model Training and Evaluation

- Using only numerical variables, the Test R^2 values for models trained on the combined training and validation sets were slightly lower than those obtained through cross-validation.
- With the inclusion of categorical variables, the Test R^2 values were substantially higher, reflecting improved model accuracy compared to using only numerical variables.

Direct Fitting Without Cross-Validation

- Direct fitting results showed greater variation in the best alphas, especially for the original and scaled datasets, highlighting the potential overfitting issue when cross-validation is not employed.
- The Test R^2 values for models with categorical variables were consistently higher, further confirming the benefit of including these features.

7 Conclusion

The project demonstrated that incorporating categorical variables significantly improves the Ridge Regression model's performance, as evidenced by higher R^2 values across all datasets and methods. Cross-validation proved to be a reliable method for selecting hyperparameters, ensuring model stability and generalizability. In contrast, direct fitting without cross-validation resulted in varied hyperparameter selections, emphasizing the risk of overfitting. Overall, the inclusion of categorical data and proper cross-validation techniques are crucial for building robust predictive models.

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.