Regression Analysis and Time Series Models (MA60280) Term Project Report

A study on the use of regression modelling techniques applied for predicting the popularity of songs.

Prabhav Patil - 20MA20042

Samarth Somani - 20MA20049

Mangalik Mitra - 20MA20070

Sandeep Mishra - 20MA20071

Rohan Das - 20MA20077

Anubhab Mandal - 20MA20080

Project Report presented for course evaluation

Spring Semester, 2024

Indian Institute of Technology, Kharagpur

Date of Submission: 15th April, 2024

Abstract

This project delves into the dynamics of music popularity, aiming to uncover the underlying determinants of song success through regression modelling. Understanding the factors that contribute to the popularity of songs within the context of a digital music platform like Spotify presents an intriguing avenue for exploration.

Exploratory data analysis techniques are employed to gain a deeper insight into the underlying patterns of the dataset, and to observe any variability of statistical properties of variables based on whether the songs are popular or not.

Using a Kaggle dataset comprising over 100,000 songs, a methodical approach integrating regression modelling techniques is employed to find out the multifaceted relationship between song attributes and popularity scores.

We apply multiple linear regression (MLR) using all the song features with the popularity score as the dependent variable. Subsequent methodological refinements, including Principal Component Regression (PCR) and MLR on reduced features, are applied to handle multicollinearity issues.

Due to the failing nature of the linear regression models on the full dataset, an undersampling procedure is performed to balance the dataset, emphasising more on the relevant popularity scores. MLR is then, finally, trained on data which was extracted based on different popularity cutoffs, which gives us comparable results.

To estimate the error analysis of the final model, the Jackknife method is used to extract the properties of the outliers that the data can potentially possess.

Finally, we also apply Logistic Regression (Logit Model) to predict whether a song is ‘Popular’ or ‘Non Popular’.

The process of setting the cutoff, sampling, running the logistic regression, and then evaluating the outputs of the model is repeated for several different cutoff points, and we ended up with some promising results.

Contents

Chapter 1

Dataset Analysis

1.1 Overview of the Data

The dataset contains 116,191 unique songs. It includes music from 32,105 distinct artists. Each song is associated with 17 attributes, with 13 of them being numerical. Table 1.1 provides a detailed description of the numerical features associated with each song. The song popularity score is used as the response (dependent) variable.

1.2 Exploratory Data Analysis

Our analysis begins with an examination of the popularity score distribution in our song dataset. The majority of songs have low popularity scores (< 40), with a mean of 24 and less than 10% exceeding 55.

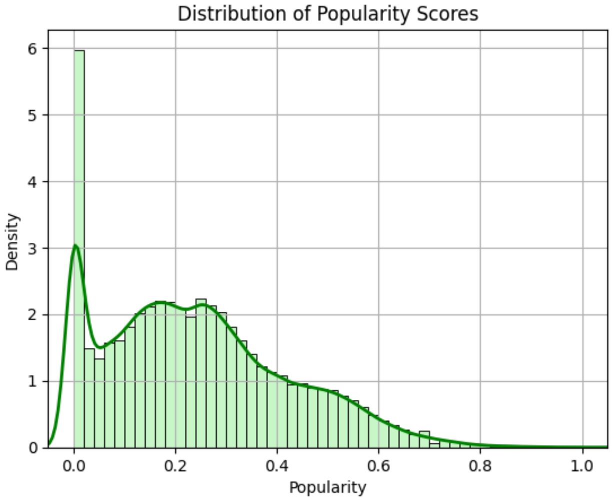


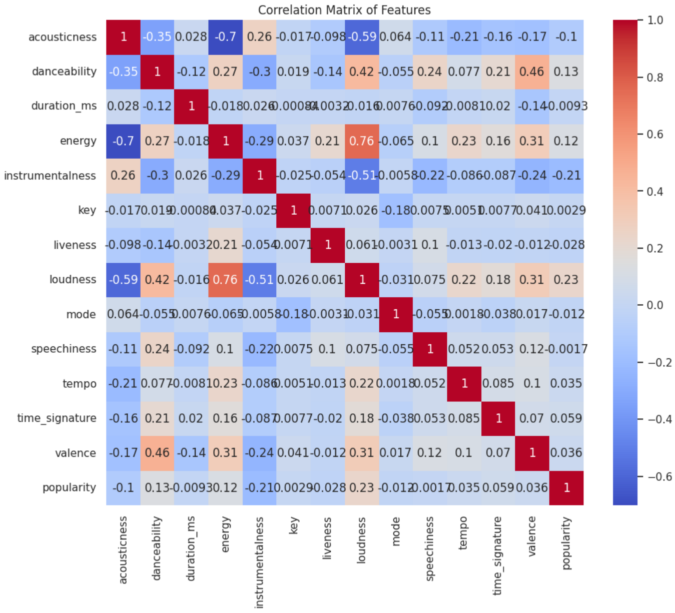
FIGURE 1.1: Plot showing the change in the number of songs with varying popularity

Challenges for Linear Regression The skewed distribution poses challenges for linear regression.

The task at hand presents considerable challenges for linear regression due to the perceived lack of correlation between several features and the target variable. A selection of scatter plots has been provided to elucidate this point, illustrating the relationship between various features and popularity. Notably, data points with popularity scores surpassing 55 have been highlighted for clarity.

Furthermore, when applying a popularity cutoff score of 55, the comparison of means for specific independent variables and their respective distributions accentuates the similarity between 'popular' and 'unpopular' songs, as demonstrated in the violin plots.

Multicollinearity Considerations While a heatmap reveals minimal multicollinearity, independent variables show little correlation with the target, complicating regression modelling.



	acousticness	danceability	duration_ms	energy	instrumentalness	key	liveness	loudness	mode	speechiness	tempo	time_signature	valence	popularity
acousticness	1	-0.35	0.028	-0.7	0.26	-0.017	-0.098	-0.59	0.064	-0.11	-0.21	-0.16	-0.17	-0.1
danceability	-0.35	1	-0.12	0.27	-0.3	0.019	-0.14	0.42	-0.055	0.24	0.077	0.21	0.46	0.13
duration_ms	0.028	-0.12	1	-0.018	0.026	0.00084	-0.0032	-0.016	0.0076	-0.092	-0.0081	0.02	-0.14	-0.0093
energy	-0.7	0.27	-0.018	1	-0.29	0.037	0.21	0.76	-0.065	0.1	0.23	0.16	0.31	0.12
instrumentalness	0.26	-0.3	0.026	-0.29	1	-0.025	-0.054	-0.51	-0.0058	-0.22	-0.086	-0.087	-0.24	-0.21
key	-0.017	0.019	0.00084	0.037	-0.025	1	0.0071	0.026	-0.18	0.0075	0.0051	0.0077	0.041	0.0029
liveness	-0.098	-0.14	-0.0032	0.21	-0.054	0.0071	1	0.061	-0.0031	0.1	-0.013	-0.02	-0.012	-0.028
loudness	-0.59	0.42	-0.016	0.76	-0.51	0.026	0.061	1	-0.031	0.075	0.22	0.18	0.31	0.23
mode	0.064	-0.055	0.0076	-0.065	-0.0058	-0.18	-0.0031	-0.031	1	-0.055	0.0018	-0.038	0.017	-0.012
speechiness	-0.11	0.24	-0.092	0.1	-0.22	0.0075	0.1	0.075	-0.055	1	0.052	0.053	0.12	-0.0017
tempo	-0.21	0.077	-0.0081	0.23	-0.086	0.0051	-0.013	0.22	0.0018	0.052	1	0.085	0.1	0.035
time_signature	-0.16	0.21	0.02	0.16	-0.087	0.0077	-0.02	0.18	-0.038	0.053	0.085	1	0.07	0.059
valence	-0.17	0.46	-0.14	0.31	-0.24	0.041	-0.012	0.31	0.017	0.12	0.1	0.07	1	0.036
popularity	-0.1	0.13	-0.0093	0.12	-0.21	0.0029	-0.028	0.23	-0.012	-0.0017	0.035	0.059	0.036	1

FIGURE 1.2: Correlation matrix corresponding to the features of the data set

Key Insights and Takeaways

1. **Imbalanced Distribution:** Most songs have low popularity, making prediction of highly popular tracks challenging.
2. **Rare Popularity Scores:** Only 0.2% of songs score above 80 in popularity, indicating scarcity of highly popular tracks.
3. **Undersampling Strategy:** Due to popularity score imbalance, undersampling may be necessary for effective model training.

4. **Missing Data Interpretation:** Zeros across features suggest potential missing data, necessitating careful handling during model training.

5. **Popularity Score of Zero:** Approximately one-tenth of songs have a popularity score of zero, which could impact model accuracy.

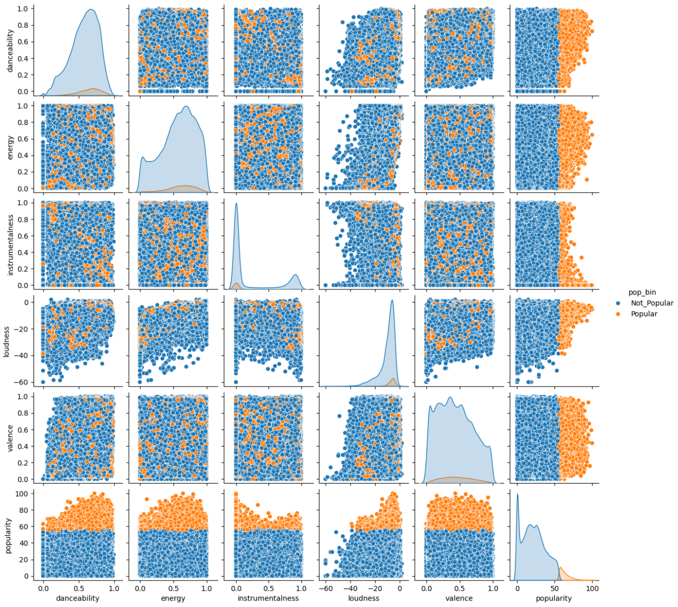


Figure 1.3: Scatter plots

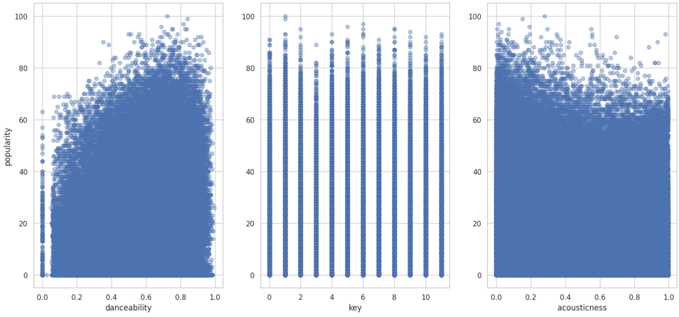


Figure 1.4: Scatter plots with a selected number of features

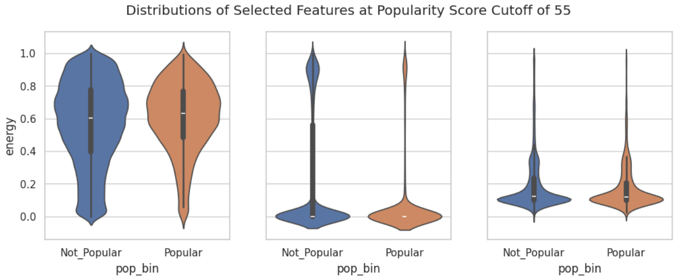


Figure 1.5: Violin Plots

Table 1.1: Descriptions of Unique Numerical Attributes for Each Song

Attribute	**Description**	**Mean**	**Std Dev**
1 - Acousticness (float)	A confidence measure from 0.0 to 1.0 of whether the track is acoustic.	0.34	0.34
2 - Danceability (float)	Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity.	0.58	0.18
3 - duration_ms (int)	Duration of the track in ms	212546	124320
4 - Energy (float)	Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy.	0.57	0.26
5 - Instrumentalness (float)	Predicts whether a track contains no vocals. “Ooh” and “aah” sounds are treated as instrumental in this context.	0.23	0.36
6 - Key (int)	The estimated overall key of the track.	5.24	3.60
7 - Liveness (float)	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live.	0.19	0.17
8 - Loudness (float)	The overall loudness of a track in decibels (dB).	-9.94	6.50
9 - Mode (int)	Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived.	0.61	0.49
10 - Speechiness (float)	Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value.	0.11	0.12
11 - Tempo (int)	The overall estimated tempo of a track in beats per minute (BPM).	119.60	30.15
12 - Time Signature (int)	An estimated overall time signature of a track.	3.88	0.51
13 - Valence (float)	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track.	0.44	0.26

Chapter 2

Methodology

We commence by partitioning the dataset into training and testing sets to facilitate model training and evaluation.

To accomplish this, we use `sklearn.model_selection.train_test_split()`, and divide the data allocating 80% for training and the rest 20% for testing. We shuffle the data before splitting, to randomize the data points in each set.

2.1 Application of Multiple Linear Regression Model

We initiate our analysis by applying multiple linear regression (MLR) utilizing all song features, with popularity scores as the target variable. Though R^2 value does not tell the entire story about a model, it was chosen as an initial metric to determine the accuracy of the model.

Implementation: We train the MLR model on the whole training dataset, and check the significance of coefficients, confidence intervals of the parameters, and finally, the R^2 and adjusted R^2 values for the fitted model.

Results: While the coefficients for *acousticness* and *danceability* were significant, The obtained coefficient of determination (R^2) was 0.0545, which implied that applying a linear regression model on a raw dataset was inefficient. Overall, fitting the MLR model on a raw dataset was a bad approach.

Insight: There was a need to further improve and include relevant parts of the dataset, which would allow the model to take into account the variability of more prominent features.

2.2 Experiments with Dataset

Observing this imbalanced dataset, having particularly a high frequency of songs with a popularity score of 0, we attempted to rectify this imbalance. The initial approach to rectify the dataset was to fully drop the dataset, corresponding to the songs that have a popularity score of 0. After improving the dataset with such methodology, we apply the MLR model in the same way.

Results: Removing the data points that had a popular score of 0 from the initial dataset resulted in the same value of coefficients and parameters and an R^2 value of 0.0532, which was in the same range.

Insight: Since there was no significant improvement in the model's accuracy, and the distribution of the popularity scores reflected upon the fact that the dataset was skewed, this led us to believe that there were either potential multi-collinearity issues or the dataset was heavily imbalanced to directly fit any regression model on.

2.3 Individual Feature Analysis

To check if some irrelevant features / independent variables were being added to the model, and to delve deeper into the relationship between each feature and song popularity, we performed Simple Linear Regression (SLR) on each feature individually. This approach helps identify which specific features contribute most significantly to the popularity of already popular songs.

Implementation: While the scatter plots between each feature and the dependent variable had very high variability, we still decided to train the SLR model on the dataset, which was generated by dropping the popularity score of 0. An analysis was done to check the significance of coefficients, confidence intervals of the parameters, and finally, the R^2 and adjusted R^2 values for the fitted model.

Results: A below par R^2 value (within the range $0.01 - 0.05$ for each feature reflected that no single feature had the upper hand on estimating the popularity score.

Table 2.1: Simple Linear Regression Results Summary

Feature	R^2	**Adjusted** R^2
Acousticness	0.0108	0.0108
Danceability	0.0174	0.0174
Energy	0.0142	0.0142
Instrumentalness	0.0429	0.0429
Liveness	0.0008	0.0007
Loudness	0.0544	0.0544
Tempo	0.0013	0.0013
Time Signature	0.0037	0.0037
Valence	0.0013	0.0013

Insight: The significance of coefficients and the R^2 values reflected upon the fact that this was not the case of multi-collinearity, or an imbalanced attempt at feature selection. Whilst the data had a range of popularity scores, there was more data that had a low popularity score, and hence, there was a need to perform a more relevant data selection process.

2.4 Addressing Multicollinearity

Suspecting multicollinearity issues among features, we conduct correlation matrix analysis and figure out if there are features which are correlated with each other.

Implementation: We construct the correlation matrix between all the features using our data set. We set a threshold of 0.90. We say that if the correlation coefficient of two features is above the decided threshold, then we remove one of the features. To decide, which of the two features is to be removed, we compare their respective Variance Inflation Factor (VIF) values. The feature with the greater value of VIF is removed.

Results: On applying the above method to our data set, we observe that the correlation coefficient of no pair of features exceeds our threshold value. Hence, there is no reduction in number of features.

Insights: From the results obtained, we deduce that multicollinearity is most likely not an issue that we face.

2.5 Applying Principal Component Regression

Even though we end up concluding that we do not face the issue of multicollinearity, we still explore whether the reduction in the dimensions of the input points yields better results. Hence, we apply Principal Component Regression (PCR).

Implementation: We find the eigenvalues and eigenvectors of the input data matrix. We set the threshold as 0.85. The number of dimensions is the minimum number of eigenvalues such that their sum is greater than or equal to the product of the threshold with the sum of all the eigenvalues.

Results: On applying PCR, we observe that the number of dimensions of the input is reduced to 5. There was still no significant improvement in the accuracy score of fitted regression models, and the R^2 value was in the range of $0.02 - 0.024$

Insight: Yet again, the below par value of R^2 reflected upon the fact that the data selection process was of threshold importance for any regression model to perform better on this dataset.

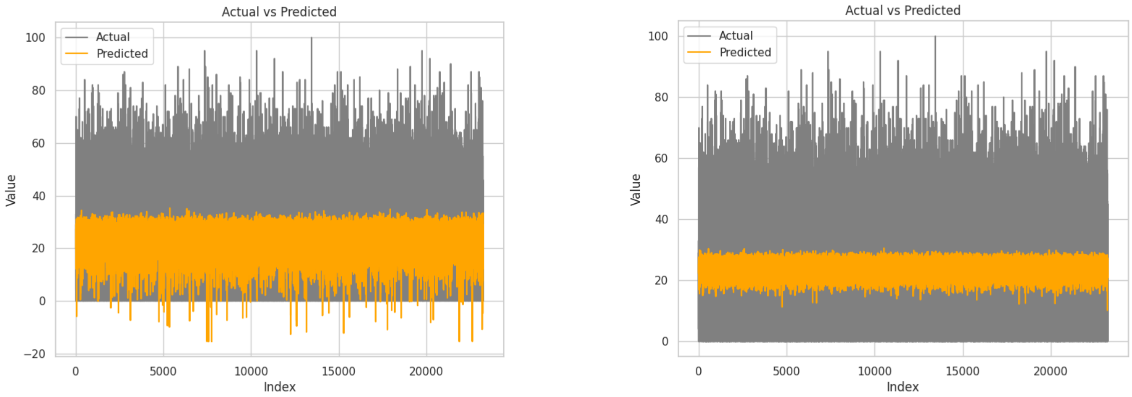


FIGURE 2.1: Actual vs Predicted Plots: MLR (left), PCR (right)

2.6 Extracting relevant samples - Undersampling Procedure

The biggest problem with this initial approach was the fact that with such unbalanced data, predicting the highest and lowest popularity values was extremely difficult. To tackle this main issue, we resorted to a sampling procedure based on the given dataset called **undersampling**.

Methodology of Undersampling: Undersampling basically allows one to balance the ratio of important/unimportant dependent variable values in an attempt to allow the model to see more of the values we care about. This is accomplished by first taking a subset of the data that contains all the important dependent variable values, in this case, all records with a high popularity score. From here on out, this is defined as the **cutoff** point.

Basically, all values with a popularity score $\geq$ the cutoff are included in the model, and then the data with popularity scores below the cutoff point is randomly sampled so that there is a 50/50 split of popular/unpopular songs in the final dataset.

Implementation: By establishing cutoff ranges based on popularity score thresholds, we selectively undersample data points with scores below the threshold. We then repeat the process of applying the MLR model for different ranges of cutoff values.

Results: This results in an increasing trend of improvement of the accuracy of the model, which is also reflected by an R^2 enhancement. The quantitative results for the accuracy and R^2 values for different cutoff values are given in the following table.

Model	R^2	Adjusted R^2
MLR	0.0545	0.0544
MLR - excluding popularity = 0 samples	0.0532	0.0531
PCR	0.0243	-
PCR - without 'mode'	0.0203	-
MLR - undersampling cutoff = 55	0.1259	0.1249
MLR - undersampling cutoff = 65	0.2026	0.1995
MLR - undersampling cutoff = 75	0.2997	0.2881
MLR - undersampling cutoff = 80	0.4183	0.3963
MLR - undersampling cutoff = 85	0.3368	0.2684
MLR - undersampling cutoff = 90	0.5783	0.3674

Table 2.2: Summary of model performances

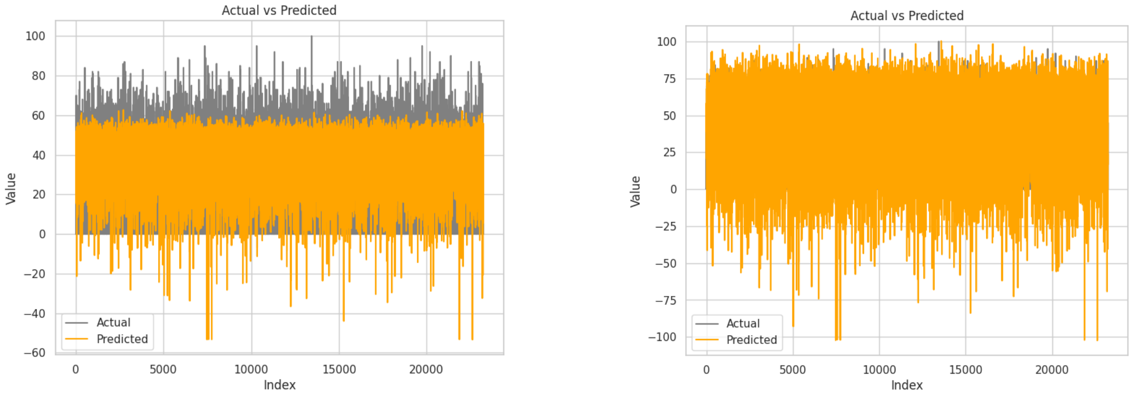


Figure 2.2: Actual vs Predicted Plots: MLR - undersampling cutoff = 55 (left), MLR - undersampling cutoff = 80 (right)

Insights: Undersampling made a big difference in how accurate our MLR model was. But there was a tradeoff: as we raised the cutoff, we ended up with fewer total samples. It makes sense, though, since the dataset does not have a ton of popular songs to work with. As such, the p-values of quite a few of the independent variables started becoming less significant. So, even though R^2 was getting better, the possible ranges for the model's coefficients were getting larger.

2.7 Final Linear Regression Model:

The final linear regression model that we decided to evaluate used a cutoff of 80, as this seemed to have a good balance between a higher R^2 value, but still retained the significance of the most important predictors. Here are the metrics for that model:

Table 2.3: Coefficients and Confidence Intervals with Feature Names

Feature	Coefficient Value	Confidence Interval
Intercept	0.2208	(0.1923, 0.2547)
Acousticness	5.5152	(5.0675, 5.9630)
Danceability	12.1694	(11.4502, 12.8887)
Duration (ms)	1.5568e-06	(6.7759e-07, 2.4360e-06)
Energy	5.1798	(4.4643, 5.8953)
Instrumentalness	-6.6238	(-6.9968, -6.2508)
Key	0.0890	(0.0578, 0.1203)
Liveness	-0.6524	(-1.3600, 0.0552)
Loudness	0.2639	(0.2366, 0.2911)
Mode	0.6151	(0.3842, 0.8459)
Speechiness	-7.5209	(-8.4829, -6.5590)
Tempo	0.0295	(0.0259, 0.0331)
Time Signature	3.8045	(3.6205, 3.9885)
Valence	-5.3295	(-5.8409, -4.8182)

Final Insights: In conclusion, not a great model, and definitely not one to rely on for an accurate popularity score. However, as we saw in the first model, the most important/significant features still appear to be:

- danceability
- energy
- instrumentalness
- loudness
- speechiness
- valence

Overall, songs with a high *danceability* value seem more likely to be popular, while *energy* and *instrumentalness* can lower the score. If a song has too high of a level of *speechiness*, that will bring down the score as well, as will too high of a *valence* score, or if a song is too "happy".

2.8 Error Analysis on the Final Model

Standard statistical error analysis methods, such as calculation of Studentized Residuals and Jackknife to detect outliers, are employed to address the relevance of the dataset for the final model.

Method Formulation:

Studentized Residuals: Studentized residuals are a standardized measure of the deviation of an observation from its predicted value in a regression analysis. It is calculated as:

$$r_i = \frac{e_i}{s\sqrt{1 - h_{ii}}}$$

Where r_i is the studentized residual for observation i, e_i is the residual for observation i, s is the standard deviation of the residuals, h_{ii} is the leverage of observation i.

Jackknife Method:

The jackknife method is a resampling technique used to estimate the bias and variance of a statistic. It involves systematically leaving out one observation at a time from the dataset and recalculating the statistic of interest. The jackknife estimate of the variance of a statistic is given by:

$$\text{Var}(\hat{\theta}_{\text{jack}}) = \frac{n-1}{n} \sum_{i=1}^{n} (\hat{\theta}_{(-i)} - \hat{\theta}_{(\cdot)})^2$$

Where $\hat{\theta}_{\text{jack}}$ is the jackknife estimate of the statistic, n is the number of observations in the dataset, $\hat{\theta}_{(-i)}$ is the statistic calculated with the ith observation omitted, $\hat{\theta}_{(\cdot)}$ is the statistic calculated using the entire dataset.

Plots: The final plots reflected that the outliers detected were negligible compared to the full dataset on which the model was applied. The final plots are shown in the figures below.

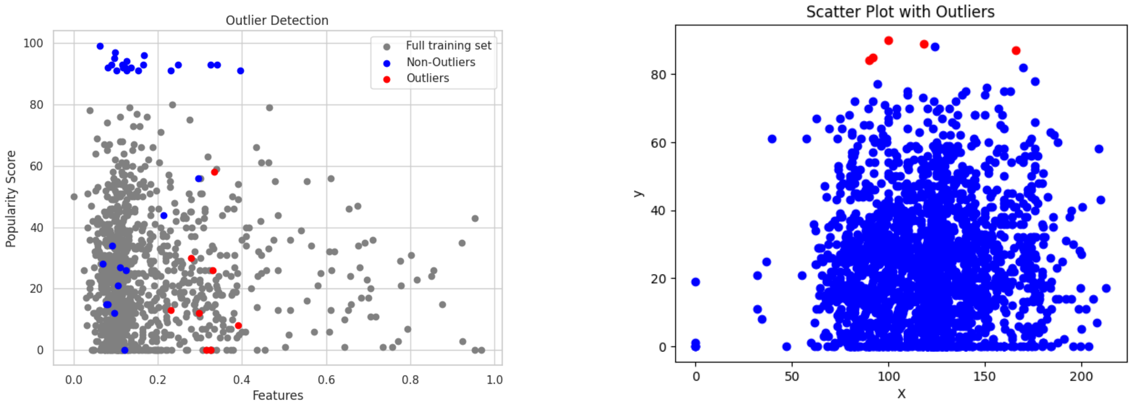


FIGURE 2.3: Outlier Detection: Studentized Residuals (left), Jackknife (right)

2.9 Logistic Regression

Since the accuracies of the linear regression models were not quite as good as what we were hoping that it would turn out to be, with some level of abstraction, we decided if we could convert the original problem of prediction of popularity scores $\in \{0, 1, 2..., 99, 100\}$ to a problem of **classifying** each song in the dataset to 2 classes - **popular** and **not popular**. Essentially, this is a problem that can be solved by **Logistic Regression**.

Although this may seem different from the original problem of regression and prediction, both are similar in the sense that both models belong to the family of Generalised Linear Models (GLMs). It is formulated as:

$$\mathbb{E}[y_i|x_i] = \sigma(\beta^T x_i)$$

where β are the learnable weights and $\beta^T x_i$ is the linear model and

$$\sigma(z) = \frac{1}{1 + exp(-z)},\ z \in \mathbb{R}\text{is the } \textbf{sigmoid function}$$

Undersampling: The distribution of the popularity scores in the dataset is originally left-skewed, indicating that there are more songs that are less popular. This will cause a class disparity when dividing the data into the 2 classes for training based on a **cutoff score**. Hence, comes the requirement for balancing the classes. We decide to take an undersampling approach here. Since the **popular** class will be a minority, we take all the data points from it (say **n**) in our sampled dataset and then from the songs that have a popularity score below the cutoff value, we randomly sample out n songs and add them to our sampled dataset. This new dataset will have an equal representation of the classes.

Standardization of Features: While modelling, we observe that the ranges of feature values are different for different features. This may cause issues during classification as the contributions of features will be skewed, and some good features may lose contribution due to less variability and other problems. Hence, we scale and standardize the features, i.e., we transform their values to have a mean 0 and variance 1 (sampled from Standard Normal Distribution). This will help to apply logistic regression evenly over all the features. The standardization is done as follows:

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

where:
z is the standardized value of the feature.
x is the original value of the feature.
μ is the mean (average) of the feature values in the dataset.
σ is the standard deviation of the feature values in the dataset

This operation is done with the use of `sklearn.preprocessing.standardscaler()` function.

Finally, the popular class is given a label **1**, and the non-popular class is given a label **0** for modelling purposes.

Implementation: We train a Logistic Regression model with **Binary Cross-Entropy Loss** minimisation, given as:

$$min_{\hat{\beta}} : \mathcal{L}(y, \hat{y} = \sigma(\hat{\beta}.\mathbf{X})) = -\frac{1}{N}\sum_{i=1}^{N}[y_i log(\hat{y_i} = \sigma(\hat{\beta}^T x_i)) + (1 - y_i) log(1 - (\hat{y_i} = \sigma(\hat{\beta}^T x_i)))]$$

We use a non-linear **sigmoid** activation function to get the predicted probabilities for the songs. The choice of sigmoid activation is evident since the output from the linear model belongs to $(-\infty, \infty)$, and we need to scale it to the range $(0, 1)$ to get the probabilities of the song belonging to each class. Finally, we apply a threshold (T=0.5) and mark the songs with scores ≥ 0.5 as 1 (popular) and others as 0 (not popular)

Choice of Cutoff Value: The final problem that needs to be addressed is the choice of the value above which we will consider a song as popular. For this, we perform a **grid search** over a discrete set of values of cutoff on the popularity score with: low = 45, high = 90, and we increment the cutoff by 5 units.

From this set of experiments we finally choose the cutoff value which gives the best performance on the metrics mentioned below.

Metrics for Evaluation: We use the following standard metrics and tools for the evaluation of the results on a standard test split:

- Receiver Operating Characteristic (ROC) plots
- Confusion Matrix (we get recall and accuracy values from this)
- Area Under the Curve (AUC) values

All of these metrics are monotone measures and help us decide the best cutoff value as the one for which we have the highest scores.

Results and Insights: As observed in the figures below, we have the ROC curves on both the sampled train and the standard test set here. From figure 2.8, we can observe that all the metrics increase monotonically until the cutoff value of 80, beyond which they decrease drastically. Hence we have a change point at 80 which is the best choice of cutoff for us.

Cutoff	Accuracy	Precision	Recall	AUC	Confusion Matrix
45	0.571	0.228	0.738	0.682	$\begin{pmatrix}2662 & 9006\\ 943 & 10628\end{pmatrix}$
55	0.571	0.113	0.768	0.705	$\begin{pmatrix}1229 & 9602\\ 371 & 12037\end{pmatrix}$
65	0.611	0.041	0.805	0.760	$\begin{pmatrix}384 & 8940\\ 93 & 13822\end{pmatrix}$
75	0.669	0.013	0.853	0.839	$\begin{pmatrix}99 & 7668\\ 17 & 15455\end{pmatrix}$
80	0.676	0.006	0.900	0.848	$\begin{pmatrix}45 & 7518\\ 5 & 15671\end{pmatrix}$
85	0.664	0.003	0.833	0.843	$\begin{pmatrix}25 & 7802\\ 5 & 15407\end{pmatrix}$
90	0.663	0.001	0.700	0.774	$\begin{pmatrix}7 & 7829\\ 3 & 15400\end{pmatrix}$

Table 2.4: Logistic Regression Results at Different Cutoff Values

Another thing to observe is that as the cutoff value increases, the model starts overfitting on the train set, and hence there is a disparity in the scores on the train and the test set.

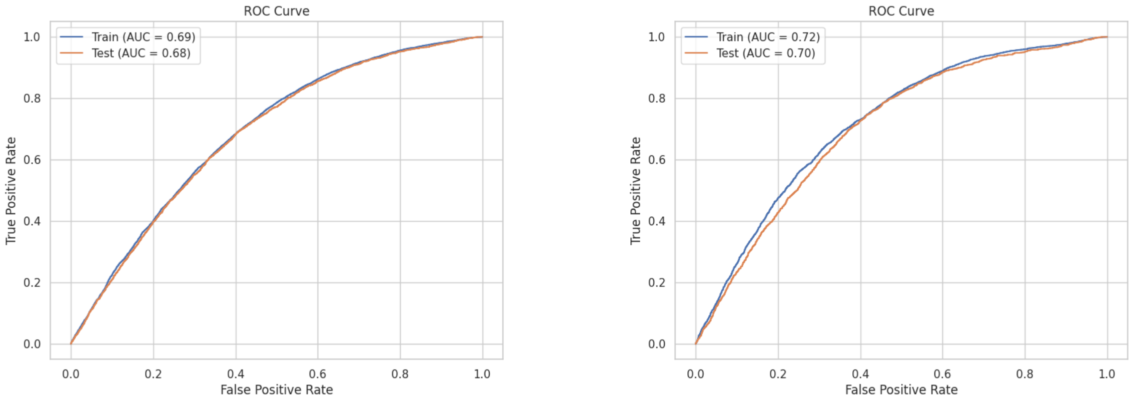


Figure 2.4: ROC plot: Cutoff = 45 (left), 55(right)

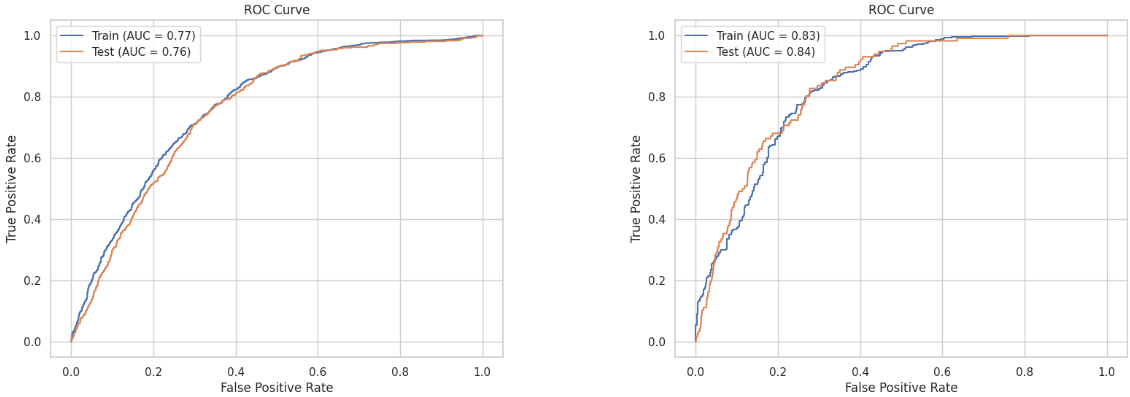


FIGURE 2.5: ROC plot: Cutoff = 65 (left), 75(right)

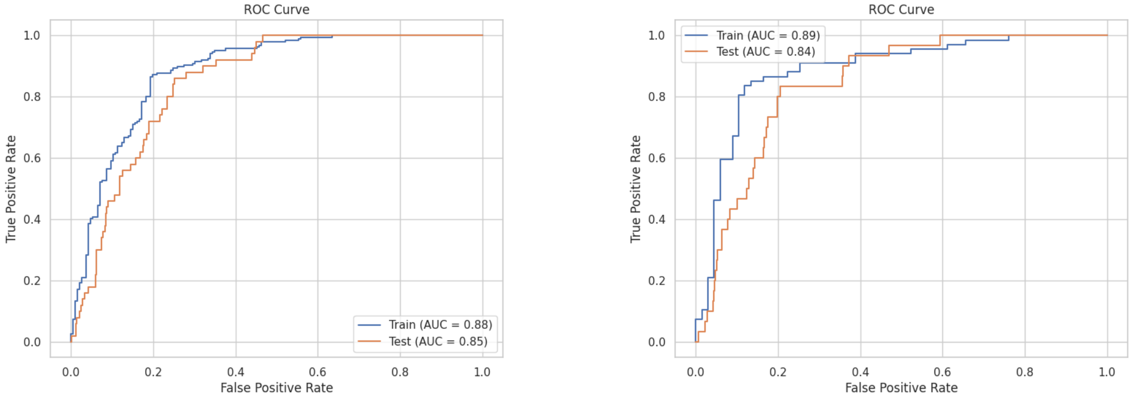


FIGURE 2.6: ROC plot: Cutoff = 80 (left), 85(right)

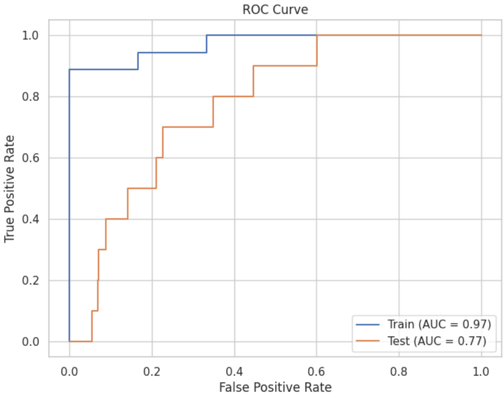


FIGURE 2.7: ROC plot cutoff = 90

2.10 Performance Metrics

We construct Receiver Operating Characteristic (ROC) plots and confusion matrices to visualize and evaluate the logistic regression models' performance. Subsequently, we determine the optimal cutoff point based on accuracy and F1-score metrics.

Receiver Operating Characteristic: The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

Accuracy: Accuracy measures the ratio of correctly predicted observations to the total observations.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Recall Score: Recall, also known as sensitivity or true positive rate (TPR), measures the ratio of correctly predicted positive observations to the total actual positives.

$$\text{Recall} = \frac{TP}{TP + FN}$$

AUC: AUC measures the area under the ROC curve. It quantifies the ability of the model to discriminate between positive and negative classes.

2.11 Final Logistic Regression Model

As observed from the metrics, for our final Logistic Regression model, we choose the popularity cutoff score as **80** and measure the performance of the model on the test set.

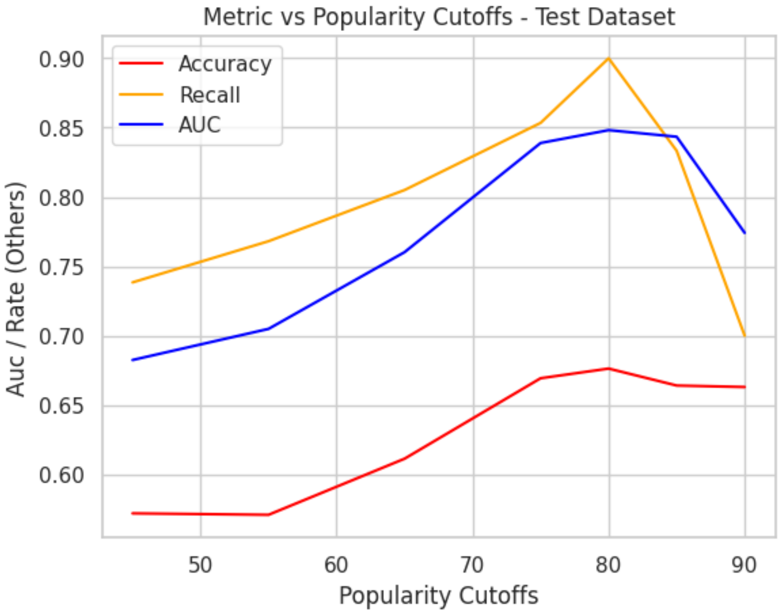


FIGURE 2.8: Metric v/s Popularity Cutoff Scores

Final Insights: In conclusion, we observe that the coefficients of the different features for classification are as follows:

Table 2.5: Coefficients of the Logistic Regression Model

Coefficient	Feature	Value
0	Bias	-2.2935
1	Acousticness	0.0673
2	Danceability	1.1390
3	Duration (ms)	-0.0144
4	Energy	-1.0725
5	Instrumentalness	-1.5294
6	Key	-0.0556
7	Liveness	-0.0491
8	Loudness	3.0367
9	Mode	-0.0147
10	Speechiness	0.2980
11	Tempo	-0.0037
12	Time Signature	-0.4756
13	Valence	-0.4307

The model is able to correctly classify a new song as popular or not with an accuracy of **67.63%**

Chapter 3

Conclusion

3.1 Final Interpretations of the Dataset

- The process of working with this dataset has been both enjoyable and enlightening. The accuracy achieved, particularly with the logistic model, is pleasantly surprising.
- Assessing the potential popularity of a song is inherently complex, and while this dataset provided valuable insights, it also revealed the presence of additional influential factors not captured within its scope.
- Factors such as an artist's current name recognition, past successes, genre, and collaborations with other prominent artists are likely to have a significant impact a song's popularity. Integrating such information into the dataset would undoubtedly enhance the predictive capabilities, enabling more precise estimations of popularity scores.

3.2 Future Work

- Exploring avenues to extend the logistic regression model into a revenue prediction framework presents an intriguing prospect. Such a model could serve as a valuable tool for stakeholders in the music industry, offering insights into a song's potential popularity and allowing for revenue estimation tailored to varying scenarios.
- Moreover, envisioning the integration of this predictive tool within the recording studio environment sparks further curiosity. Providing artists with real-time feedback on the potential popularity of their creations could foster a dynamic feedback loop, enriching the creative process and guiding decision-making during the development of new material.
- In conclusion, while this endeavour has yielded promising results, it also unveils a horizon ripe for further exploration and innovation in leveraging data-driven insights to navigate the dynamic landscape of the music industry.