Designing Models: Prediction of Song Popularity

David Lee, Hyungsuk Lee, Changhee Han

Department of Computer Science, Emory University

CS 334: Machine Learning

Dr. Pouriyeh

May 01, 2023

Abstract

The purpose of this study is to design a model that best predicts popularity of songs using lyrics. As lyrics are one of the most important aspects of music composition, we aimed to predict the popularity of songs using lyrics. Using various models including perceptron, naive bayes, logistic regression, and linear regression, we used different preprocessing techniques to optimize each model and compared the models to find the one that best predicts popularity of songs. Out of all the models, linear regression was our best model with the availability of predicting popularity of songs using different languages of lyrics and the best accuracy. Considering that we did not include other important features that could affect the popularity of songs such as tempo, key, loudness, energy, etc., additional research is needed to improve the models.

Keywords: machine learning, popularity, songs, prediction

Introduction

Music recommendation systems rely on explicit data such as user ratings and play counts to make recommendations, but the lyrics of a song can provide valuable insights into its emotional and thematic content. In this project, we propose to develop a machine learning model that predicts the genre and popularity of a song based on its lyrics. The proposed machine learning model can provide a deeper understanding of the music industry's market preferences, helping to identify the influence of lyrics among the many factors that contribute to a song's success. The insights generated by the model can be used to inform decision-making and create more engaging and successful music.

Lyrics in music are one of the most important features that affect listeners. According to the key findings of Midia, “88% of streaming music subscribers look for lyrics”, “Lyrics are considered central to the song for 61% of streaming lyrics users”, and “The longer consumers have been music subscribers, the more likely they are to use lyrics” (Mulligan, 2017). The data suggests that lyrics play a significant role in the music listening experience, particularly among streaming music subscribers. According to the data, a large majority of streaming music subscribers actively look for lyrics when listening to music. Additionally, a majority of streaming lyrics users consider lyrics to be central to the song, indicating that lyrics are an essential aspect of the music listening experience for many people. Furthermore, the data shows that the longer consumers have been music subscribers, the more likely they are to use lyrics, suggesting that lyrics may become increasingly important to music listeners over time. Overall, the data supports the idea that lyrics are an important component of music popularity, and that they are significant

to music listeners. The data suggests that lyric could be a potential predictor for the popularity of the song itself.

In our project, we experimented with various machine learning models and open-source translating modules to identify the best approach to capture the patterns and themes in song lyrics across different languages. We explored open-source translating modules such as googletrans, boto3, and GoogleTranslator. However, we faced limitations in the number of characters provided for free, and additional fees were required after a certain limit. Therefore, for the perceptron, naive bayes, and logistic regression models, we focused on English songs because they were the most numerous songs in the dataset. We considered using Spanish songs but there were less than 2000 songs available in the dataset so we decided to focus on English songs only for the perceptron, naive bayes, and logistic regression models.

For the linear regression model, we found that LaBSE was the best model for our project. Initially, we used the pre-trained model MiniLM as it was designed for question-answering, semantic search, and text classification. However, we found that this model was not ideal since it was pre-trained only in English. Although LaBSE was originally designed for capturing semantic similarity between sentences in different languages, we found that specific genres of music still exhibit similar patterns of language and themes. For instance, the model created from the dimensions of the lyrics would capture certain words or phrases that are prevalent in a specific type of genre. Even if the word is expressed differently in different languages, the model can still capture the similarity and produce the dimensions in a similar way. The perceptron, naive bayes, and the logistic regression models will predict song popularity based on the presence of the word or the number of the words being present. The linear regression model will predict song popularity based on language and theme patterns.

Background

The dataset available on Kaggle provides a comprehensive collection of audio and textual features for over 25,000 songs available on Spotify. The dataset includes various audio features such as danceability, energy, acoustic, instrumentalness, valence, tempo, loudness, and key, along with lyrics and sentiment analysis of the lyrics. This dataset can be useful for training machine learning models that predict the popularity of a song based on its lyrics. It can also be used to conduct research and analysis on the music industry, including the influence of lyrics on a song's success.

There were several previous research using the same dataset on Kaggle. One research was “Spotify Popularity Prediction - ML Practice” by Pelin Soylu. The author describes a machine learning project that aims to predict the popularity of songs on Spotify based on various features such as acousticness, danceability, and energy. The author provides a detailed analysis of the dataset and applies various data preprocessing techniques such as handling missing values and feature scaling. The author conducts data visualization and feature engineering to derive new features that could improve the accuracy of the machine learning models. The author uses several machine learning models, including logistic regression, random forest, k-nearest neighbors, and support vector machines, and compares their performances. The results indicate that random forest was the best model, achieving an accuracy of 71% in predicting the popularity of songs. On the other hand, our model was based on the conclusion that lyric is a more important feature for listeners as provided by the data from Midia and we focused on selecting

features solely from the lyrics of the songs and proceeded to design the best model to predict the popularity of songs.

Methods

Perceptron

The implementation of the perceptron algorithm was used to apply it to the song popularity classification. The words that appear in more than 30 songs are treated as features of the model. There were mainly 4 cases in this study with the use of binary dataset and count dataset for all the English songs in the total dataset and 6 different genres separated in the edm, latin, pop, rap, r&b, and rock datasets. A binary dataset was created by using the word map that was created for the specific case and determining if the presence of the word resulted in the value of 1.0 or 0.0. A count dataset was created by using the word map that was created for the specific case and determining the number of counts of the words. The popularity column was adjusted to 1 being popular with the value greater than or equal to 75 and 0 being not popular with the value less than 75. We splitted the dataset into training and testing sets in a 70:30 ratio respectively. The models were optimized through tuning the hyperparameters, which were the number of iterations and the learning rate. We thought using perceptron would be useful for predicting song popularity based on lyrics because it can handle high-dimensional feature spaces. Also, for the case of the music industry, the model can update the weights incrementally as new songs become available everyday.

Naive Bayes

We implemented the Naive Bayes algorithm, a popular probabilistic algorithm commonly used for text classification tasks, to predict the popularity of songs based on their lyrics. Naive Bayes assumes that the features (words in this case) are conditionally independent of each other, which makes it computationally efficient and able to handle a large number of features. Our implementation involved preprocessing the lyrics, splitting the data into training and testing sets, and evaluating the performance of the model using the F1 score. We used a Grid Search to find the optimal hyperparameters for a given model. The alpha hyperparameter is used for Laplace smoothing, fit_prior is used to specify whether to learn the class prior probabilities from the training data, and class_prior is used to specify the prior probabilities of the classes. We set the fixed fit_prior and class_prior and only alpha is vary. The alpha hyperparameter is used for Laplace smoothing. In Multinomial Naive Bayes, Laplace smoothing is used to avoid the zero-frequency problem, which occurs when a feature does not appear in the training data for a particular class label. The alpha hyperparameter controls the strength of the smoothing, with smaller values resulting in more smoothing.

Logistic Regression

Logistic regression is a popular classification algorithm used to predict binary outcomes. In our project, we utilized logistic regression to predict the popularity of songs based on their lyrics. We employed a grid search approach to tune the hyperparameters of the model, including the regularization strength parameter C and the type of penalty to apply. 'C' is the regularization parameter that controls the strength of the regularization applied to the model. A small 'C' results in a more constrained model, while a large 'C' results in a less constrained model. 'Penalty' is the

type of regularization to be applied to the model, and it can be 'l1', 'l2', 'elasticnet', or None. 'l1' tends to result in sparse solutions, 'l2' tends to result in smooth solutions, and 'elasticnet' is a combination of both.

Linear Regression

Before implementing the linear regression, we had to perform a PCA on a filtered dataset and plot the results in a 3D scatter plot. The PCA function is used to specify that we want to reduce the dimensionality of the data to three principle components. The method is then used to fit the PCA model to the filtered dataset and transform the data into the new lower-dimensional space. The percentage of total variance explained by the three principle components is calculated and stored in the variable. Finally we had to create a 3D scatter plot of the transformed data, where the three principal components are plotted along the x, y, and z axes. The color of each point is determined by the popularity of the corresponding track in the original dataset, and the title of the plot includes the total explained variance percentage. After performing the PCA, we had to implement the linear regression to predict the popularity of music tracks based on a set of input features. The input dataset is split into two arrays ‘X’, containing the input features and ‘y’, containing the target variable to be predicted, which is the popularity of each track. PCA is then applied to reduce the dimensionality of the input features to three components, and the transformed data is stored. The dataset is split into training and testing sets, and a linear regression model is trained on the training set. The model is then used to predict the popularity of the tracks in the test set, and a scatter plot is created to visualize the relationship between the actual and predicted popularity values.

Results

Preprocessing

Perceptron/Naive Bayes/Logistic Regression

For the perceptron, naive bayes, and logistic regression models, it was necessary to preprocess the lyrics. As the lyrics are not variables that can be used for the models as shown in Table 1a, we decided to build a vocabulary map and create a dataframe that utilizes the presence of such vocabulary as the binary dataset in Table 1b and the number of counts of the vocabulary as the count dataset in Table 1c.

	Unnamed: 0	artist	name	popularity	genre	lyrics
0	0	Steady Rollin	I Feel Alive	0	rock	the trees are singing in the wind the sky blue...
1	1	Bell Biv DeVoe	Poison	0	r&b	na yeah spyderman and freeze in full effect uh...
2	2	CeeLo Green	Baby It's Cold Outside (feat. Christina Aguilera)	0	r&b	i really cant stay baby its cold outside ive g...
3	3	KARD	Dumb Litty	0	pop	get up out of my business you dont keep me fro...
4	4	James TW	Soldier	1	r&b	hold your breath dont look down keep trying da...
...	...	...	...	...	...	...
15400	15400	NAV	Some Way	1	r&b	yeah nah nah nah nah nah nah nah nah nah n...
15401	15401	Qulinez	Rising Like The Sun - Radio Mix	0	edm	caught up in such a head rush wideeyed lately ...
15402	15402	Nicki Minaj	Anaconda	0	pop	my anaconda dont my anaconda dont my anaconda ...
15403	15403	Ponderosa Twins Plus One	Bound	0	r&b	bound bound bound bound bound to fall in love ...
15404	15404	Father MC	I'll Do 4 U (Re-Recorded / Remastered)	0	r&b	would you do for me sweetheart would you do fo...

15405 rows × 6 columns

Table 1a. Total dataset.

This table is the selected columns from the original dataset from Kaggles. There are a total of 15405 songs and we selected the basic song information with the artist and name. The original popularity measurement ranged from 0 to 100 but as our model needs a binary classification, we decided that a popularity greater than 75 is considered to be popular. This splits the data to 0, not popular, with 90 percent of the data and 1, popular, with 10 percent of the data.

	to	a	reality	its	na	you	yeah	men	must	captain	...	homeboys	ended	bridges	jaw	rider	purse	dot	bail	shock	devotion
0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	1.0	1.0	0.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	1.0	1.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	1.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	1.0	1.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
10778	1.0	1.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10779	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10780	1.0	1.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10781	1.0	1.0	0.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10782	1.0	1.0	0.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

10783 rows × 3509 columns

Table 1b. Binary dataset.

This table is the binary dataset that was created from the total dataset. The columns are the words in the vocabulary map that represents the unique vocabulary that appears in more than 30 songs. A binary dataset was created for English songs only. 7 binary datasets were created with the total dataset and the genres edm, latin, pop, rap, r&b, and rock. The value 1.0 describes that the vocabulary in the column name is present in the lyrics and the value 0.0 describes that the vocabulary in the column names is absent.

	to	a	reality	its	na	you	yeah	men	must	captain	...	homeboys	ended	bridges	jaw	rider	purse	dot	bail	shock	devotion
0	17.0	2.0	1.0	2.0	1.0	1.0	3.0	1.0	2.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	13.0	10.0	0.0	0.0	4.0	2.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	5.0	1.0	0.0	0.0	1.0	28.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	7.0	10.0	0.0	1.0	0.0	41.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	12.0	9.0	0.0	3.0	0.0	12.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
10778	16.0	9.0	0.0	0.0	0.0	7.0	3.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10779	7.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10780	2.0	8.0	0.0	8.0	0.0	10.0	2.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10781	18.0	4.0	0.0	2.0	2.0	12.0	4.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10782	3.0	4.0	0.0	1.0	1.0	6.0	6.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

10783 rows × 3509 columns

Table 1c. Count dataset.

This table is the count dataset that was created from the total dataset. The columns are the words in the vocabulary map that represents the unique vocabulary that appears in more than 30 songs. A count dataset was created for English songs only. 7 count datasets were created with the total dataset and the genres edm, latin, pop, rap, r&b, and rock. The value describes the number of counts the vocabulary in the column name is present in the lyrics.

Linear Regression

	track_popularity	lyrics_dim_0	lyrics_dim_1	lyrics_dim_2	lyrics_dim_3	lyrics_dim_4	lyrics_dim_5	lyrics_dim_6	lyrics_dim_7	lyrics_dim_8	...
0	41.0	-0.058687	-0.051955	-0.041783	-0.061065	-0.042640	-0.002660	-0.060148	0.003739	-0.053391	...
1	28.0	-0.065967	-0.027921	-0.034698	-0.068686	0.048735	0.051609	-0.046426	-0.006113	0.004367	...
2	0.0	-0.056247	-0.042539	-0.012083	-0.057370	-0.037971	-0.025456	-0.056762	0.039184	-0.042830	...
3	41.0	-0.049499	-0.042787	0.000948	-0.063282	-0.021570	-0.018737	-0.075551	-0.002493	-0.007746	...
4	65.0	-0.023482	-0.004275	-0.011224	-0.044246	-0.009216	0.022312	-0.085818	0.036650	-0.002408	...
...	...	...	...	...	...	...	...	...	...	...	...
18449	0.0	-0.022080	-0.046012	-0.042908	-0.074712	-0.027986	-0.016963	-0.069969	0.029518	0.017201	...
18450	49.0	-0.043650	-0.032551	0.014667	-0.068469	-0.028351	-0.033002	-0.055935	0.004362	-0.023672	...
18451	40.0	-0.039533	-0.056906	-0.028727	-0.075754	-0.012442	0.029250	-0.025079	0.007055	0.005050	...
18452	36.0	-0.057966	-0.042716	-0.042541	-0.068077	0.020891	-0.050059	-0.046311	-0.004288	-0.058813	...
18453	61.0	-0.022365	-0.035139	-0.053111	-0.066127	-0.030650	-0.023881	-0.063481	-0.002944	0.006779	...

Table 2. 768 dimensions using LaBSE model

Prior to implementing linear regression without PCA reduction the MSE score was 547 with the R^2 score of 0.04 which showed that the data is not fit for a linear regression model. However after conducting the PCA reduction and optimizing the model, the MSE score was 0.00 and R^2 score was 1.0. One possible explanation as to why it performed better was that through PCA reduction, it can help reduce the noise and redundancy in the input features, thus improving the accuracy of the linear regression model. By reducing the dimensionality of the input features, PCA can also help to mitigate the issue of overfitting, which occurs when a model is too complex and performs well on the training data but poorly on the testing data.

Process

Perceptron

Before creating models separately in different genres, we optimized the perceptron model through tuning maximum iterations and learning rates. By selecting the learning rate that converges the quickest and the lowest maximum iterations at the lowest number mistakes, the 7 models for each language were optimized.

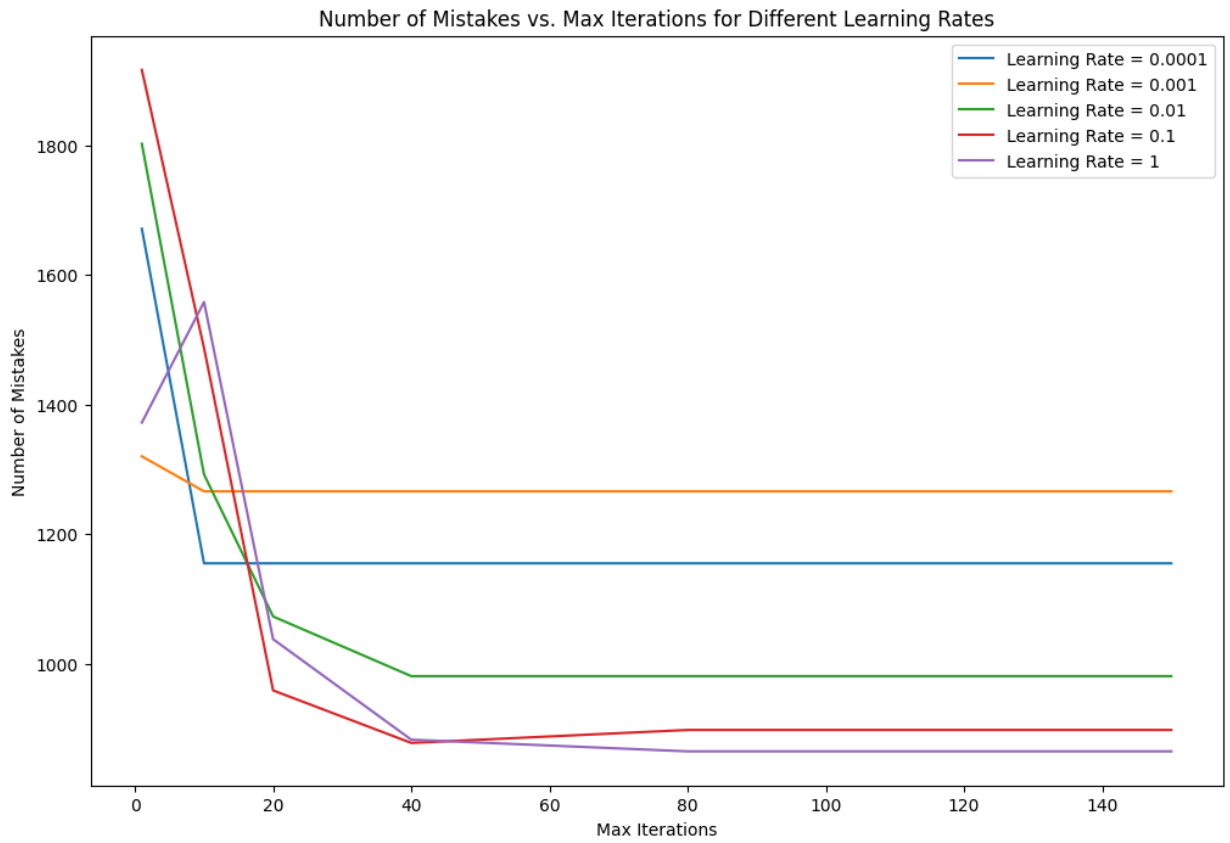

Figure 1. Number of Mistakes over Maximum Iterations for Total Binary Dataset.

This is a graph of the number of mistakes over maximum iterations for the total binary dataset in different learning rates. Different learning rates were trained for different maximum iterations and the number of mistakes in classifying the popularity of the song were measured to choose the optimal hyperparameter for the perceptron model. As shown in the graph, learning rate of 1 and maximum iterations of 80 was shown to be the optimal hyperparameters for the perceptron model for the total binary dataset. Also, for the binary datasets for different genres, learning rate of 1, 1, 1, 1, 0.01, and 1 and maximum iterations of 20, 40, 40, 40, 20, and 40 were chosen for edm, latin, pop, rap, r&b, and rock respectively. Similarly, for the count datasets for different genres, learning rate of 0.01, 1, 0.01, 0.01, 0.01, 0.01, and 1 and maximum iterations of

40, 20, 40, 40, 20, 40, and 20 were chosen for total edm, latin, pop, rap, r&b, and rock respectively.

Naive Bayes/Logistic Regression

Naive Bayes

	Binary Train Accuracy	Binary Test Accuracy	Count Train Accuracy	Count Test Accuracy
total	0.8447	0.8338	0.8189	0.7709
edm	0.9780	0.9678	0.9439	0.9242
latin	0.8764	0.8605	0.9048	0.7713
pop	0.7894	0.7647	0.7677	0.6845
rap	0.8898	0.8802	0.8652	0.7949
r&b	0.8838	0.8967	0.8693	0.8272
rock	0.8870	0.8840	0.8435	0.7729

Table 3. Accuracy Table for Naive Bayes

The evaluation has been done on ten different genres of music. Each variation of the model has been evaluated using binary and count features. The performance of each model has been compared based on train accuracy, train mistakes, test accuracy, and test mistakes. Grid search technique has been used to find the best hyperparameters for each model. The best accuracy through grid search has been obtained using the Multinomial EDM Binary model with an accuracy of 0.978, and the best parameters for the model are {'alpha': 10.0, 'class_prior': None, 'fit_prior': True}. The worst accuracy has been obtained using the Multinomial Pop Count model with an accuracy of 0.684 and the best parameters for the model are {'alpha': 0.01,

'class_prior': None, 'fit_prior': True}. The result shows that the Multinomial EDM Binary model outperforms all other models with a high accuracy of 0.978. The Multinomial Pop Count model performs poorly with an accuracy of 0.684. The train accuracy and test accuracy of all the models are close to each other, which indicates that the models are not overfitting or underfitting.

The result also shows that the train mistakes and test mistakes are high for some models such as the Multinomial Pop Count, Multinomial Pop Binary, Multinomial Latin Count, and Multinomial Rock Count models. This indicates that these models are not able to classify some instances correctly. In conclusion, the Multinomial EDM Binary model is the best model among all other models for the classification of different genres of music.

Logistic Regression

	Binary Train Accuracy	Binary Test Accuracy	Count Train Accuracy	Count Test Accuracy
total	0.8776	0.8771	0.9072	0.8877
edm	0.9780	0.9678	0.9789	0.9678
latin	0.8865	0.8566	0.9349	0.8915
pop	0.7944	0.7834	0.8750	0.8316
rap	0.8841	0.8828	0.9229	0.8948
r&b	0.9078	0.9210	0.9182	0.9146
rock	0.8946	0.8938	0.9059	0.8869

Table 4. Accuracy Table for Logistic Regression

The best accuracy achieved through grid search is around 0.978 for the binary versions of EDM and count versions of Latin. The best parameters for all the models are the same with C=0.001 and penalty=l2. For the binary version of Pop, the model achieved an accuracy of 0.782 with the best parameter C=0.01, while for the count version of Pop, the model achieved an accuracy of 0.817 with the best parameter C=0.001. For the binary version of Rap, the model achieved an accuracy of 0.884 with the best parameter C=0.001, while for the count version of Rap, the model achieved an accuracy of 0.884 with the best parameter C=0.001. For the binary and count versions of RB, the models achieved an accuracy of around 0.908 and 0.907, respectively, with the best parameter C=0.001 for both. For the binary and count versions of Rock, the models achieved an accuracy of around 0.895 and 0.890, respectively, with the best parameter C=0.001 for both. Finally, for the total binary version, the model achieved an accuracy of 0.878 with the best parameter C=0.001.

Linear Regression

Prior to implementing the PCA reduction on the data set which contains 768 dimensions, when the linear regression model was applied, there was no linear relationship shown due to possible noise in the data. But after reducing the dimension using PCA reduction, we were able to see a clear linear relationship. Due to this aspect, we concluded that in this assignment, PCA reduction would be a great option for preprocessing before implementing the linear regression model.

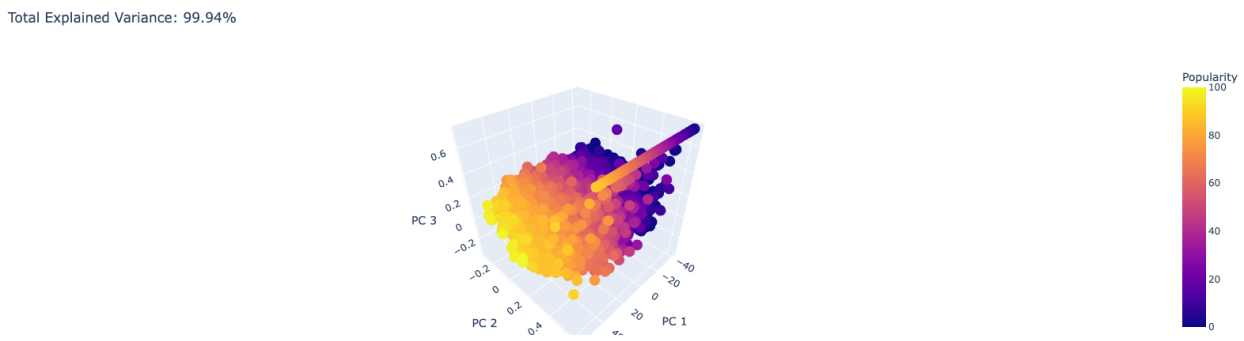

Figure 2. Dimension Reduction Using PCA

Data

Perceptron

	Binary Train Accuracy	Binary Test Accuracy	Count Train Accuracy	Count Test Accuracy
total	0.9198	0.8146	0.8959	0.8503*
edm	0.9943	0.9470	0.9829	0.9564
latin	0.9900	0.7713	0.9733	0.7558
pop	0.8728	0.6720	0.8605	0.7246
rap	0.9886	0.8269	0.9086	0.8682
r&b	0.9485	0.7935	0.9556	0.8504
rock	0.9519	0.7640	0.9068	0.8653
average	0.9577	0.7958	0.9313	0.8368

Table 5. Accuracy Table for Perceptron

As shown in Table 4, the test accuracy for the count dataset without splitting the genres was the best model with an accuracy score of 0.8503. The average row shows the average accuracy of the edm, latin, pop, rap, r&b, and rock perceptron models.

Naive Bayes

	Binary Train Accuracy	Binary Test Accuracy	Count Train Accuracy	Count Test Accuracy
total	0.8447	0.8338	0.8189	0.7709
average	0.8841	0.8757*	0.8657	0.7958

Table 6. Comparison Table for Naive Bayes

As shown in Table 5, the test accuracy for the binary dataset with splitting the genres was the best model with an accuracy score of 0.8757. The average row shows the average accuracy of the 6 different genres: edm, latin, pop, rap, r&b, and rock perceptron models.

Logistic Regression

	Binary Train Accuracy	Binary Test Accuracy	Count Train Accuracy	Count Test Accuracy
total	0.8776	0.8771	0.9072	0.8877
average	0.8909	0.8842	0.9226	0.8979*

Table 7. Comparison Table for Logistic Regression

As shown in Table 6, the test accuracy for the count dataset with splitting the genres was the best model with an accuracy score of 0.8979. The average row shows the average accuracy of the 6 different genres: edm, latin, pop, rap, r&b, and rock perceptron models.

Linear Regression

Analyzing the linear regression model based on the mean squared error which is a commonly used metric to measure the difference between the predicted values and actual values in a regression problem. It is calculated by taking the average of the squared differences between the predicted and actual values. The number R^2 is a statistical measure that represents the proportion of the variance in the dependent variable that is explained by the independent variables in a regression model. An MSE value of 0.00 and an R-squared value of 1.0 in a regression model suggest that the model perfectly fits the data. The MSE value of 0.00 indicates that there is no difference between the predicted values and the actual values, which means that the model has accurately captured the relationship between the independent and dependent variables. Similarly, an R-squared value of 1.0 indicates that all of the variance in the dependent variance is explained by the independent variable in the model, which means that the model is a perfect fit for the data. However, it is important to note that these values can also be a result of overfitting the model to the data. Overfitting occurs when the model is too complex and fits the noise or random fluctuations in the data instead of the underlying pattern. Therefore, it is important to validate the model using an independent test dataset and evaluate other metrics to ensure that the model is accurate and not just fitting the training data too well.

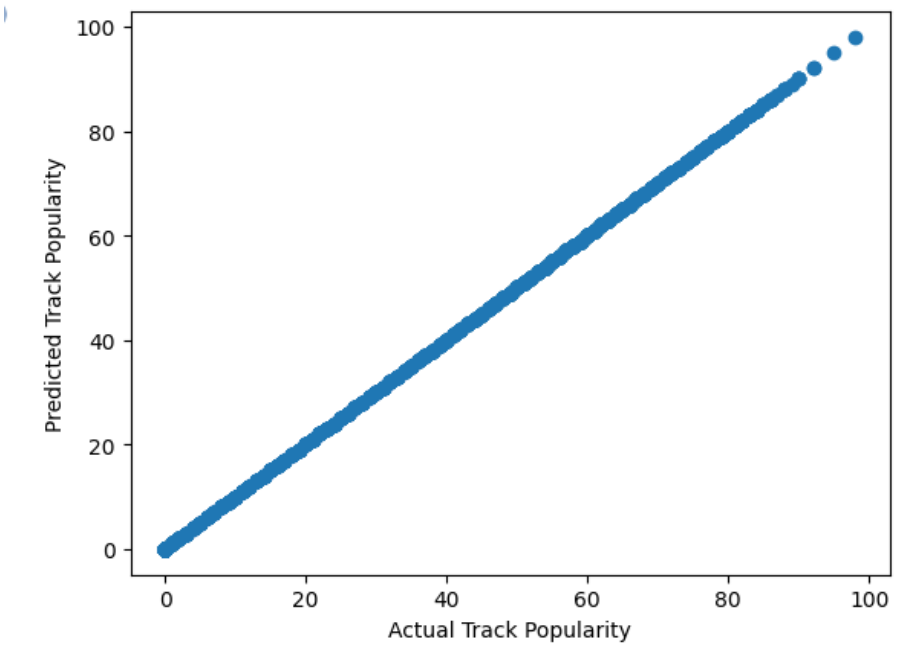

Figure 3. Predicted vs. Actual Track Popularity for Linear Regression

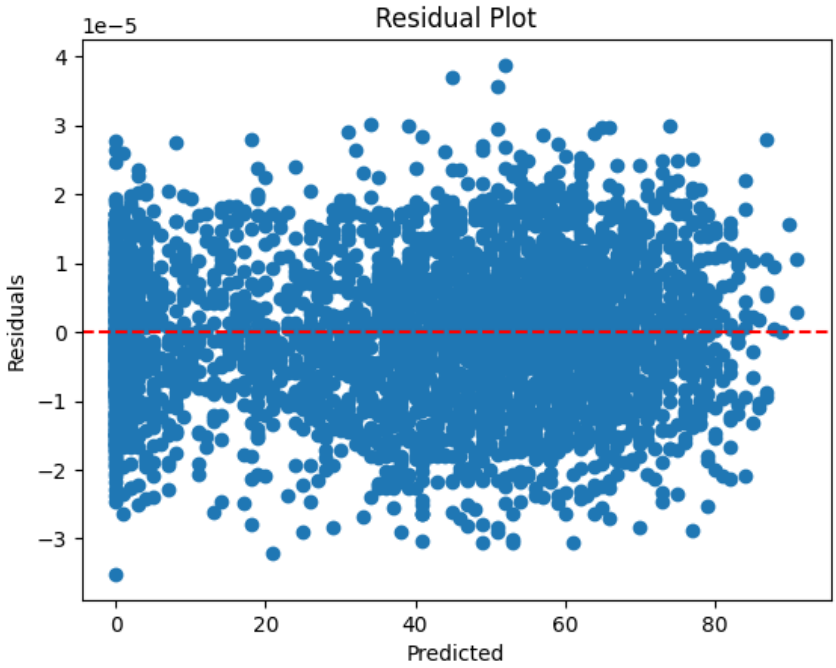

Figure 4. Residual Plot for Linear Regression

This plot shows the relationship between the predicted values(x-axis) and the residuals(y-axis), which are the differences between the actual values and the predicted values. A good model will produce residuals that are randomly scattered around the horizontal line at y=0, without any patterns or trends. If there is a clear pattern or trend in the residuals, it may indicate that the model is not capturing some important aspect of the data.

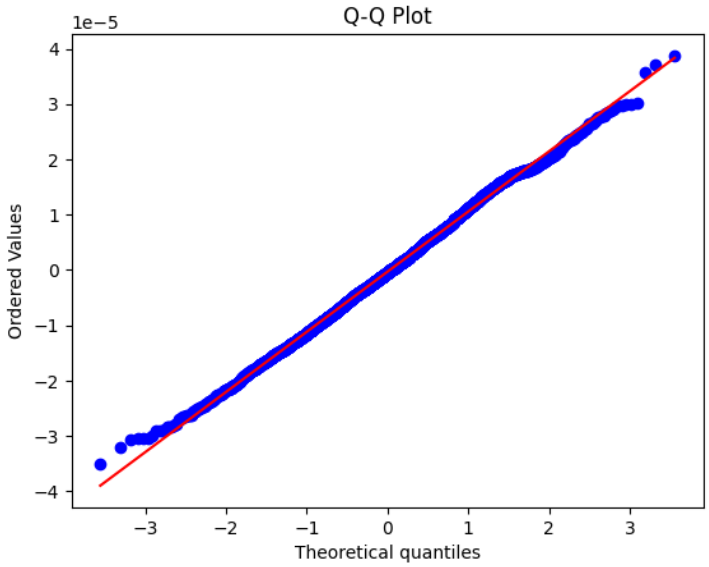

Figure 5. Q-Q Plot for Linear Regression

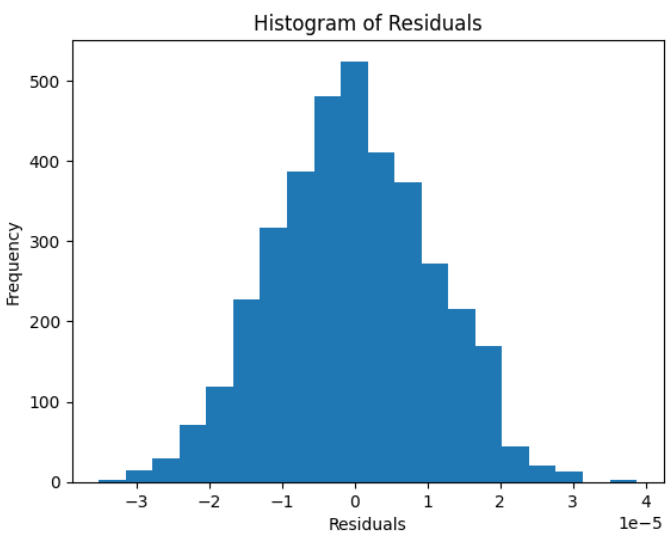

Figure 6. Residual Plot for Linear Regression

This plot shows whether the residuals are normally distributed. If the residuals are normally distributed, the points on the plot will fall along the diagonal line. If the residuals are not normally distributed, the points may deviate from the diagonal line in a systematic way. Another plot shows the distribution of the residuals. A good model will produce residuals that are approximately normally distributed with a mean of 0. If the residuals are not normally distributed, it may indicate that the model is not accurately capturing some aspect of the data.

Discussion

In this study, we evaluated different models to predict song popularity using lyrics. Different preprocessing techniques were used to optimize different models. Also, different optimization techniques, such as tuning hyperparameters, were used to increase the accuracy of the models. Overall, the best model for predicting song popularity using lyrics was the linear regression model.

The linear regression model was considered to be the best not only looking at the overall accuracy but also the ability to predict song popularity in a continuous measurement and the ability to predict song popularity with different languages.

While the MSE score and R-squared value are ideal, it is important to keep in mind the drawbacks of conducting PCA reduction. First potential drawback might be information loss. PCA reduces the number of variables by combining them into a smaller number of principal components. While this can help to simplify the data, it can also result in some loss of information. The degree of information loss depends on the number of components retained and the amount of variance explained by them. Second potential drawback could be interpretability. PCA can make it harder to interpret the underlying meaning of the variables in the dataset. The

principal components are linear combinations of the original variables, and the interpretation of our model is not clear.

The linear regression model had MSE value of 0.0 and R-squared value of 1.0. However, it is noteworthy that only lyrics were used for the features in the model. As shown in the residual plots, the model did not have a complete accuracy in measuring the popularity. Further research is needed in including features such as tempo, key, loudness, etc. that could affect the popularity of the song.

Contributions

- David worked on the preprocessing of the models and worked on the abstract, introduction, background and the sections regarding perceptron in the methods and results.
- Changhee Han worked on implementing linear regression based on different genres. And got the MSE and R^2 value to evaluate the model of the predictions. And I worked on the linear regression part and the discussion section of the report.
- HyungSuk Lee worked on implementing Naive Bayes and Logistic Regression and tested each genre. I also processed and analyzed the results of the two models, and optimized them by the GridSearch method to find the best hyperparameters for achieving optimal outcomes.

Code/Dataset

https://www.dropbox.com/sh/9ymvvwulwvc2gh8/AAAQAjXY8I_FO1xD96R9NFH7a?dl=0

References

Ahmetfurkandemr. “Linear Regression on GPU with Rapids.” *Kaggle*, Kaggle, 6 May 2021, https://www.kaggle.com/code/ahmetfurkandemr/linear-regression-on-gpu-with-rapids

Mulligan, M. (2017). Lyrics Take Centre Stage In Streaming Music A MIDiA Research White Paper Prepared For LyricFind Lyrics Take Centre Stage In Streaming Music. https://musicindustryblog.files.wordpress.com/2018/01/lyrics-take-centre-stage-in-streaming-e28093-lyricfind-report.pdf

Nakhaee, Muhammad. “Audio Features and Lyrics of Spotify Songs.” *Kaggle*, 14 June 2020, https://www.kaggle.com/datasets/imuhammad/audio-features-and-lyrics-of-spotify-songs.

Pelinsoylu. (2020, April 24). *Spotify popularity prediction-ml practice*. Kaggle. Retrieved May 1, 2023, from https://www.kaggle.com/code/pelinsoylu/spotify-popularity-prediction-ml-practice

Ramirez, Victor. “Discovering Descriptive Music Genres Using K-Means Clustering.” *Medium*, LatinXinAI, 14 May 2018, https://medium.com/latinxinai/discovering-descriptive-music-genres-using-k-means-clustering-d19bdea5e443

“Sentence-Transformers/Labse · Hugging Face.” *Sentence-Transformers/LaBSE · Hugging Face*, https://huggingface.co/sentence-transformers/LaBSE

“Sentence-Transformers/MSMARCO-Minilm-L-6-v3 · Hugging Face.” *Sentence-Transformers/Msmarco-MiniLM-L-6-v3 · Hugging Face*, https://huggingface.co/sentence-transformers/msmarco-MiniLM-L-6-v3