Abby Fremaux, Brian Li, Anam Khan

Georgetown University

Project Step II

Spotify genre classification Dataset

Table of Contents

[Project # 1](#_Toc86276921)

[Project Description 1](#_Toc86276922)

[Date Submitted 1](#_Toc86276923)

[Project Priority 1](#_Toc86276924)

[Step 1. Project Deliverables 1](#_Toc86276925)

[Step 2. List of Project Tasks 1](#_Toc86276926)

[Step 3. Out of Scope 1](#_Toc86276927)

[Step 4. Project Assumptions 1](#_Toc86276928)

[Step 5. Project Constraints 2](#_Toc86276929)

[Step 6. Updated Estimates 2](#_Toc86276930)

[Step7. Approvals 2](#_Toc86276931)

[Introduction 3](#_Toc86276932)

[Analysis of the dataset and Trained Model 3](#_Toc86276933)

[Exploratory Analysis and Visualization 3](#_Toc86276934)

[Baseline Model 4](#_Toc86276935)

[Model Selection 5](#_Toc86276936)

|  |  |  |  |
| --- | --- | --- | --- |
| Project # | Project Description | Date Submitted | Project Priority |
| 1 |  |  |  |

## Step 1. Project Deliverables

Please list *all project deliverables* listed in the Project Charter and, if necessary, elaborate on them. *Do not list dates*. Add more rows as necessary.

|  |  |
| --- | --- |
| **Deliverable ID#** | **Description** |
| 1 | Project Charter |
| 2 | Ensemble trained model with preliminary results of the training and testing. |
| 3 | Flask and Heroku application. (customer will choose whether to deploy it locally or on the cloud) |
| 4 | Final report of the project in addition to the cloud (heroku) deployed link. |

## Step 2. List of Project Tasks

Please list ***all project tasks*** to be completed, based on the “Deliverables” specified in the Project Charter. *Do not list dates*. Add more rows as necessary. Optional: you may substitute a work breakdown structure (WBS) or mind-map in lieu of Step 2. Please attach WBS or mind-map to document.

|  |  |  |  |
| --- | --- | --- | --- |
| **Task ID#** | **Task to be completed** | **Delivery Date** | **For Deliverable #** |
| 1 | Submit Project Charter | 09/07/2022 | 1 |
| 2 | Ensemble trained model with preliminary results of the training and testing. | 09/30/2022 | 2 |
| 3 | Flask and Heroku application. (customer will choose whether to deploy it locally or on the cloud) | TBD | 3 |
| 4 | Final report of the project in addition to the cloud (heroku) deployed link. | TBD | 4 |

# Introduction

To make effective recommendations, we must understand what constitutes a specific genre. What is the difference between a hip-hop song and a pop song, and how is this discrepancy represented by data? Is it the tempo? Energy? Danceability? Our project will focus on analyzing how one genre is different from one another, how artists from the same genre could differ based on style, and how to accurately classify songs into the proper genre based on the features from the data. This will allow us to generate playlists and recommendations for users based on their preferences of genres and/or artists.

# Analysis of the dataset and Trained Model

## Exploratory Analysis and Visualization

The following graphs show the distribution of selected variables. The dataset does not have any missing values.

Chart, line chart

Description automatically generated

Figure : Distribution of Danceability

Figure 1 shows the distribution of the variable, Danceability. Price is approximately normally distributed. Figure 2 shows a correlation matrix. Graphical user interface

Description automatically generated

Figure 2: Variation of Genres based on Popularity and Danceability

Chart, treemap chart

Description automatically generated

Figure 3: Correlation of Variables

## Baseline Model

Since this is the classification dataset, we have decided to analyze the model using various classiers. For simplicity, all the model code is provided with an attached jupyter notebook, not in the report. We measured the accuracy of the classifers by precision score. Logistic Regression was the best performing model with a precision score of 0.59. A confusion matrix can be seen below which shows a majority of rock songs were currently identified as rock songs.

Diagram

Description automatically generated

Figure 4: Confusion Matrix of Logistic Regression

# **Model** Selection

Obtaining a baseline provided us necessary insight for model selection. Following the preliminary regression analysis, we have decided to evaluate, LinearRegression, Ridge Regression, Lasso Regression, Decision Tree Regressor, Random Forest Regressor, Bagging Regressor, Gradient Boosting Regressor, and XGB Regressor as candidate models. Figure 4 shows the model results, and we can see that Random forests, Bagging, Gradient boosting, extreme gradient boosting, and light gradient boosting perform the best (have relatively smallest mean RMSE). Because of this, we will select them as the candidate models for level 0 in the stacking regression. We will use Light Gradient Boosting Regression as the level 1 combiner or metamodel to aggregate the results of the level 0 models.

Figure 4 Machine Learning Model Result

In addition to the selected models, we have also opted for cross-validation of the data set. For the current iteration of the model, we will use five-fold validation. Figure 4 shows the performance of the standalone and stacked models. The performance of the stacked model seem to be worse than the candidate model. Stacked models performance can be improved by tuning the hyperparmameters. For now we will focus on the price estimation by using stacked model.

Figure Candidate model and stacked model performance

## Model Performance Evaluation

Below you will see the different precision scores of each classifer. Logistic Regression and Random Forest performed the best at approximately 60%.

# 

Figure 6: Bar Chart of Precision Scores