Predicting Song Popularity, Group 30

Shivaz Sharma
s2444842
s2444842@ed.ac.uk

Sparsh Rawal
s2314252
s2314252@ed.ac.uk

Tom Weatherall
s2436812
s2436812@ed.ac.uk

Abstract

This report focuses on an extension of Hit Song Science (HSS), the study of what characteristics popular songs have. We begin by analyzing and transforming our data to allow for machine learning classification tasks. In this report we attempt to predict if a song will be popular on any given day. We evaluate three classification models; Logistic Regression, Decision Trees and Random Forests. Random Forests return the best classification results with accuracy of 88.22%, we further analyze what features were most important in the Random Forest classification model.

1 Introduction

Music has and always will be a central part of communities and cultures. There is an incredibly wide range of music around the world with there being many different genres and tastes specific to different regions and different peoples. In today's world most people listen to music through streaming platforms with Spotify being one of the biggest [1]. Music is a massive revenue generating industry, 26bn USD in 2021 [6]. Hit Song Science (HSS) is the study of song characteristics to try identify if certain song features are highly related to song success. HSS is critical because if it is accurate, songs could be manufactured to be popular and generate significant income. The rise of streaming platforms and the large datasets of listener habits these platforms generate, has allowed HSS to be studied in some depth [3]. This report will be an extension of HSS as we try to answer the question that "On any given day, can we use artist name and audio features to predict a song's popularity?" There have been similar studies to this in the past [5] [2]. Using our dataset and the associated Spotify audio features of popular songs throughout 2017, we analyze, prepare and classify our data. In this report we use three different classification methods; Logistic Regression, Decision Trees and Random Forests to analyze if we can predict song popularity. In our evaluation of the classification methods we discuss test accuracy, precision and recall. Then we discuss the importance of the various dataset features in the classification of our most successful method. If we can find a classification model that accurately predicts the popularity of a song, this will indicate that HSS is possible and this information could be used to manufacture songs. These songs would be intended to emphasize the "popular" characteristics to give the song the best likelihood of being popular, being streamed by many listeners and generating revenue.

The rest of our paper is organized as follows. In Section 2 we discuss data preparation, Section 3 explores our dataset and identifies any underlying trends in our data, Section 4 discusses the classification methods used, Section 5 discusses the classification results and Section 6 concludes.

2 Data Preparation

2.1 Original Dataset

The original dataset provided for the 'Worldwide trends in Spotify' study contained the daily ranking of the 200 most listened songs from 54 regions around the world from 2017 and 2018 by spotify users. The dataset was arranged in a DataFrame that included daily ranking, date, number of streams that day, spotify URL, song name, artist name and region. There were 371 unique days and 54 regions in the dataset. This meant that there were roughly 3.4 million unique data points in this dataset. A cursory review of the data revealed that, as one would expect, there was a high degree of duplication in the dataset. When filtered on unique songs the dataset shrunk dramatically to 21,328 unique songs. Further analysis of the data revealed that there were a small number of blank songs that we believed have been removed from Spotify. The blank songs were removed from our dataset. This left us with 21319 songs.

2.2 Label Dataset

As our chosen task is to predict daily popularity, a crucial step for this study was to define popularity. This was not as straightforward as we expected. Obviously the two parameters that we have been given that are the best proxies of popularity are number of streams and song position/rank.

Analysis of the "number of streams" parameter raised two issues. First, as shown in Figure 5 in the appendix there is a steep exponential curve in the number of streams. This issue can be easily overcome by transforming the variable by taking the log values. The second issue is a "Timeline Bias" in the data. A song that is released in the beginning or middle of 2017 is going to register more streams in our dataset than a song that is released in December 2017 even if they receive a similar number of streams over their lifetime. While the first issue is easy to resolve, the second issue is insurmountable and so we decided to discard the number of streams as a proxy for popularity. Leaving daily song position ranking as the proxy for song popularity. This again proved to be a complex issue. How could we best capture a song's popularity across all regions by its position ranking? At most a song could have 54 different rankings for a given day i.e. it is in the Top 200 songs for every region.

For each song each day we created a list of the songs' different rankings across all the regions. We ordered these in ascending order and took the median position ranking to be the proxy for song popularity. Given the position ranking is ordinal data the rankings do not suffer from the exponential/outlier issue that number of streams does. Taking the median of the rankings again reduces the outlier effect on this parameter. This measure does not run into the Timeline Bias because a song can have a high popularity on a given day even if its lifetime streams are low. Now we had a popularity measure for each song each day. We then give each song a label. The song is sorted/classified into the Diamond Hit category if its median rank value is 1-40, Golden Hit if its value is 41-80, Silver Hit for 81-120 and Bronze Hit for 121 and above. This gave us a label dataset with 73105 entries.

2.3 Feature Dataset

Now that we had defined our label dataset we needed to determine our feature dataset. After excluding the position feature as it is now the label dataset, the remaining features are daily streams, date, Artist Name, Song Name, Region, Song URL.

Daily streams : Daily streams is a variable that is generated after a song is released. As our task is an attempt to predict ahead of time how popular a song will be, Daily Streams cannot be a variable used to estimate Popularity.

Date : We are encoding the date string into numeric values corresponding to a particular day. For example "1-jan-2017" is encoded as 1, "2-jan-2017" is encoded as 2 and so on. It ranges from 1 to 365. Since our dataset consists of songs which are listened to in 2018 also, we cannot just serially assign a day to the date hence we decided to loop the assignments at the end of the year, So, "1-Jan-2018" is encoded as 1 instead of 366.

Artist Name : As the Artist Name is a string we had to transform this parameter into a numerical variable to allow us to use it as a machine learning feature. To encode the string variable we experimented with One Hot Encoding, Binary Vector Encoding and Frequency Encoding. We decided not to use One Hot or Binary Vector Encoding because these methods would have made our dataset unnecessarily complex. Frequency encoding on the other hand is simple to understand and implement. In line with Occam's Razor we take the simplest approach and transform the Artist Name variable into frequency variables. This is done by calculating the number of songs an artist, say Drake, has in the dataset divided by the total number of songs in the dataset. This frequency variable is calculated for each artist and this is the artist rank variable we use as a feature in the machine learning classification models. An obvious issue with our model is that the Artist Name will not generalize to an unseen

artist in the test data.
Song Name : Song Name is again categorical data. However there was no way to represent this data in a meaningful way that could generalize to new data. Because of this we discarded Song Name as a variable.
Region : The way the popularity of a song is calculated it already encapsulates the region parameter. It does not make sense to include regions in the dataset as a song will have multiple regions each day.
Song URL : We use the Song URL as a unique identifier of the songs and not as a classification feature.
Audio Features : A large part of HSS has been the research of audio features as a predictor of song popularity [1]. We need more feature data in order to build robust classification models. Spotify provides access to the audio features of virtually all songs on Spotify platform through its APIs [7]. We used the Song URL as the unique identifier and then scraped the audio features. The features include: Danceability, Tempo, Valence, Energy, Acousticness, Speechiness, Liveness and Instrumentalness. All the audio features have a value between 0 and 1. A description of the audio features can be found here [1].

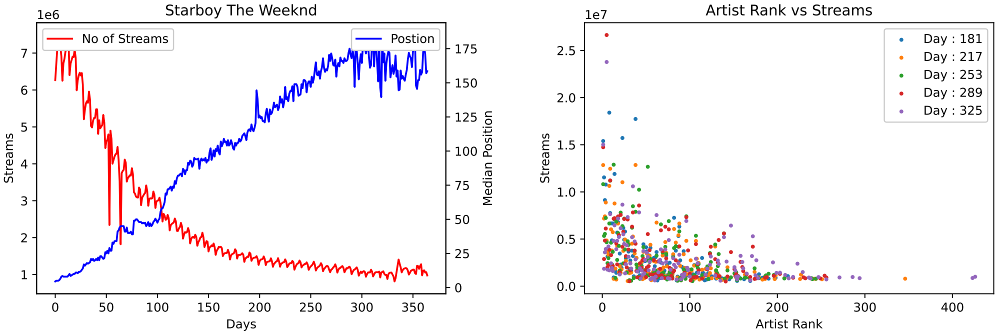
Finalized Feature Dataset
The feature dataset includes Artist Name (frequency encoding), Date and the Spotify Audio Features.

3 Exploratory Data Analysis

After initial preparation of the data as mentioned in Section 2 we are left with a data set consisting of 73,105 rows and 11 columns namely Artist-Rank, Norm-Day, Danceability, Energy, Speechiness, Acousticness, Instrumentalness, Liveness, Valence, Tempo and Class. Class being the dependent variable which our model tries to predict. *We use the Per-Day-Streams for EDA but not for model training.* We begin our EDA by analyzing how the features affect the target variable. First by looking at the trend of a Song's position over the period of time when it is in the Top 200 and then we conduct Principal Component Analysis (PCA) for visualizing the data distribution.

3.1 Song Trend Analysis

Initial analysis of Per Day Streams (PDS) revealed that PDS of a song decreases over a period of time. This is very much in line with our expectations. Because a song starts to lose popularity as it gets older. As the Streams and positions are negatively correlated, we see the position of the song increasing. This trend can be seen in Figure 1(a).



(a) Per Day Streams and Daily Position of the song

(b) Number of Streams and Rank of the Artist

Figure 1: Song Trend Analysis

So as we can see the position of a song is dependent on the PDS and that further depends on the day on which the song is being streamed, we believe that the Day would be a valuable feature towards predicting the popularity class of the song.

Next we analyze how the Artist-Rank plays a role. We assume that a popular artist will have more songs in the data set, so we rank the artists based on the number of their songs in the dataset, for example the most popular artist will have the lowest rank i.e. 1. Based on the above assumption we hypothesize that a more popular artist is bound to have more streams of their song on any given day. Figure 1(b) clearly highlights that the lower ranked (more popular) artists have a higher number of total streams of their song on any given day. Hence any song released by a popular artist is bound to be streamed more and hence it would achieve lower rank overall. This indicates that the popularity of an artist will have an impact on our task.

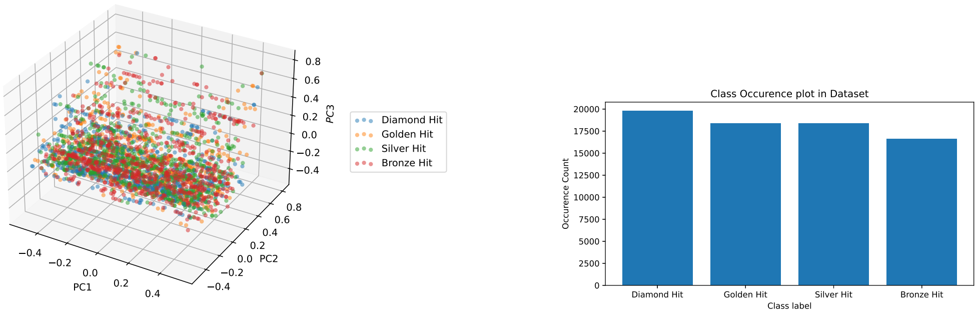
3.2 Principal Component Analysis

Visualization is a critical concept for understanding the dimensionality and distribution of data. It helps in making better decisions while training the model. PCA analysis is a technique to project high dimensional data on a lower dimension. Figure 2(a) clearly depicts there is no linear separability in our target classes. As the data is non-linearly distributed we expect that Logistic Regression will perform badly and that Decision Trees and Random Forests will perform better.

Non-linearly distributed data can be a hurdle for the basic classification algorithms such as Naive Bayes and Logistic regression since they fit a linear hyperplane in order to classify the data whereas more complex models such as Decision Tree model builds the tree based on the individual features and labels. As a result the Decision trees are capable of classifying non linear data. However, it is prone to overfitting the training data and often requires pruning to mitigate overfitting.

3.3 Class Balance

Figure 2(b) shows the counts of each class in the dataset. We observe that the counts are fairly balanced across the four classes.This is important because it means that the classification methods will not become biased towards specific classes and more accurate predictions can be made.



(a) PCA for Data Visualization

(b) Class Counts in Dataset

Figure 2: Exploratory Data Analysis

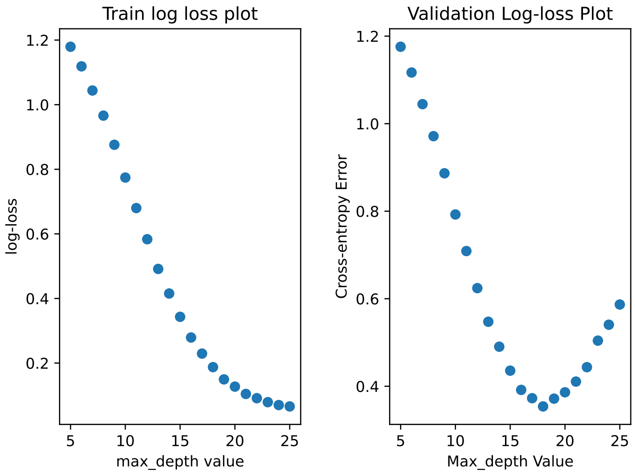
4 Method

4.1 Classification Methods

Logistic Regression : In this report we implement multinomial logistic regression with a cross-entropy loss function to allow classification into multiple classes. The advantage of Logistic Regression over other classification methods is that it is easier to interpret and efficient to train. However, as shown in the EDA, the data is non-linearly distributed. Logistic regression assumes linearity between the dependent and independent variables. Because of this we expect Logistic Regression to perform badly and include it as a sort of control method to compare the following classification methods too.

Decision Trees : We have chosen Decision Trees as our next method because it performs well with non-linearly separable data. Another benefit of Decision Trees is that they are able to generate understandable rules, perform classification with low computation cost and provide a clear indication of which features are most important for classification. As shown in Figure 8(b) in the Appendix, we found that a Decision Tree with a depth of 9 minimized the validation error and set the max depth of the Decision Tree to 9.

Random Forest : Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and the Bootstrap and Aggregation technique. The problem of overfitting is overcome by averaging the results of different decision trees. This classifier has less variance than standard Decision Trees. Random Forest is powerful on classifications of non-linear data, and generally provides higher flexibility and returns higher accuracy than Decision Trees. For hyper-parameter tuning, we have set the number of trees to 100, criterion as entropy and tuned max-depth parameter. As shown in Figure 3(b) we found that a Random Forest with a depth of 18 minimized the validation error and set the max depth of the Random Forest to 18.



(a) Training Loss Plot

(b) Validation Loss Plot

Figure 3: Cross Entropy Loss of Random Forest Classifier

4.2 Implementation

After pre-processing and data preparation of the dataset is completed, the dataset is split into training/validation/testing sets in an 80/10/10 split. We are using the Scikit Learn library [4] for training and evaluating classification models. The Random State parameter was set to 100 to ensure consistent results throughout the experiments. As described for the Decision Trees and Random Forest methods we tune the hyper-parameters (depth) to minimize the validation error. In order to evaluate the classification models we calculate the model accuracy, F1-Macro, Cohen's kappa score and generate confusion matrices for our three classification methods.

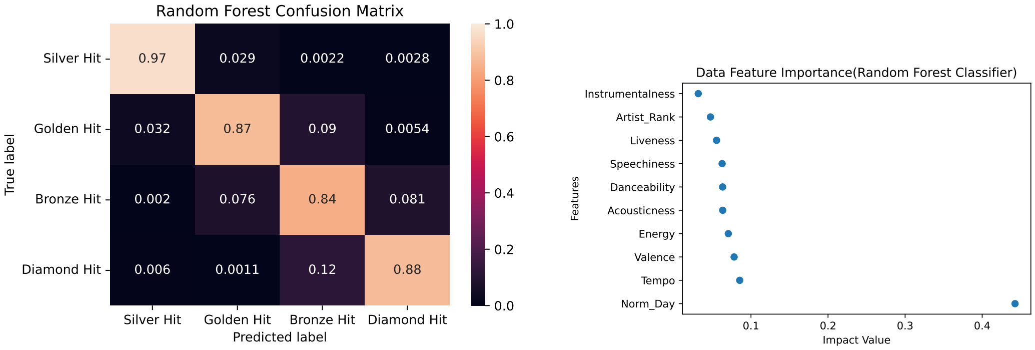
$$\mathcal{L} \;=\; -\sum_{c=1}^{M} y_{o,c} log\left(p_{o,c}\right) \tag{1}$$

Cross Entropy Loss Function For Multi-class Classification

5 Results

As shown in Table 1 in line with our expectations that Logistic Regression is a poor classifier. For all our model metrics Logistic Regression returns a very poor result, significantly below the metrics provided by Decision Trees and Random Forests. Again in line with our expectation Random Forests is the best method returning a test accuracy and F1 score of 0.89 and a Cohen's Kappa of 0.845. Figure 4(a) is the confusion matrix generated by the Random Forest clearly depicting the high accuracy of the model.

While Logistic Regression did return a poor result, the strong results returned by Decision Trees and Random Forests indicate that we can predict the popularity of a song using Artist Name, Date and the



(a) Random Forest Confusion Matrix

(b) Scatter plot highlighting feature importance

Figure 4: Evaluation of Random Forest Model

audio features. Next we turn to analysis of what features explain the most variance in the dataset and so are the most important in classification.

As Random Forest is our best classifier we will analyze the feature importance of our random forest model. As shown in Figure 4(b) the Date is the most important feature. Very interestingly and against expectations the next most important feature is not Artist Rank. Instead all the audio features bar Instrumentalness play a bigger role in classification. With Tempo being the most important audio feature. We hypothesized that Artist Name would be one of the most important features, interestingly it has come in as the second to last most important feature. Running the Random Forest classification experiment without the Artist Name feature hardly impacted the accuracy metrics of the random forest model. Intuitively Artist Name must have an impact on a songs popularity, this indicates that our encoding of the Artist Name feature is not desirable and further study could look into the impact of different handling of the Artist Name feature.

Method	Accuracy	F1-Macro	Cohen's Kappa
Logistic Regression	36.80	0.32	0.148
Decision Tree	58.30	0.57	0.441
Random Forest	88.22	0.88	0.842

Table 1: Evaluation metrics for different classifier models

6 Conclusion

As shown in Section 5 we can predict the popularity of a song on a given day using Decision Trees and Random Forests. Due to the non-linearity of the data Logistic Regression returns poor classification results.Although we stipulated that the Artist would be an essential feature in predicting the songs' position the results show otherwise. An area for further study would be to analyse the effect of different encodings of the Artist Name feature in order to see if the different encodings lead to this feature having more of an impact on the song's popularity. As the Date is the most important feature, further study could try and isolate what types of songs are more popular at different times throughout the year. It stands to reason over periods such as Christmas, Christmas carols would be more popular and these songs would all share very similar audio features. An extension to this study would be to study the effect of regions throughout the year as different regions celebrate different festivals etc. The music associated with these times (such as carols) would be far more popular over these times and the audio features associated with these songs would become far more important during these times.

Citations and References

References

[1] Carlos Vicente Soares Araujo, Marco Cristo, and Rafael Giusti. *A Model for Predicting Music Popularity on Streaming Platforms*, volume 27. 2020.

[2] Faheem Khan, Ilhan Tarimer, Hathal Salamah Alwageed, Buse Cennet Karadağ, Muhammad Fayaz, Akmalbek Bobomirzaevich Abdusalomov, and Young-Im Cho. Effect of feature selection on the accuracy of music popularity classification using machine learning algorithms. *Electronics*, 11(21):3518, 2022.

[3] Brian McFee, Thierry Bertin-Mahieux, Daniel PW Ellis, and Gert RG Lanckriet. The million song dataset challenge. In *Proceedings of the 21st International Conference on World Wide Web*, pages 909–916, 2012.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[5] James Pham, Edric Kyauk, and Edwin Park. Predicting song popularity. *Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, Tech. Rep*, 26, 2016.

[6] Mark Savage. The global music market was worth 26bn in 2021, 2022.

[7] Spotify. Spotify api documentation, 2022.

Appendix

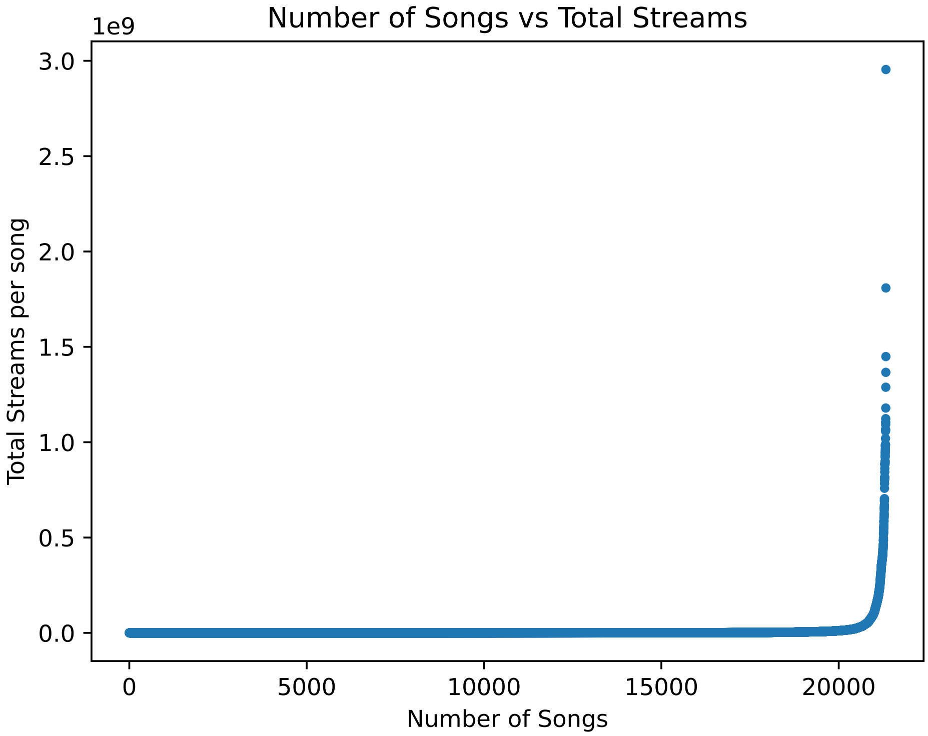


Figure 5: Plot of Total Streams Per Song

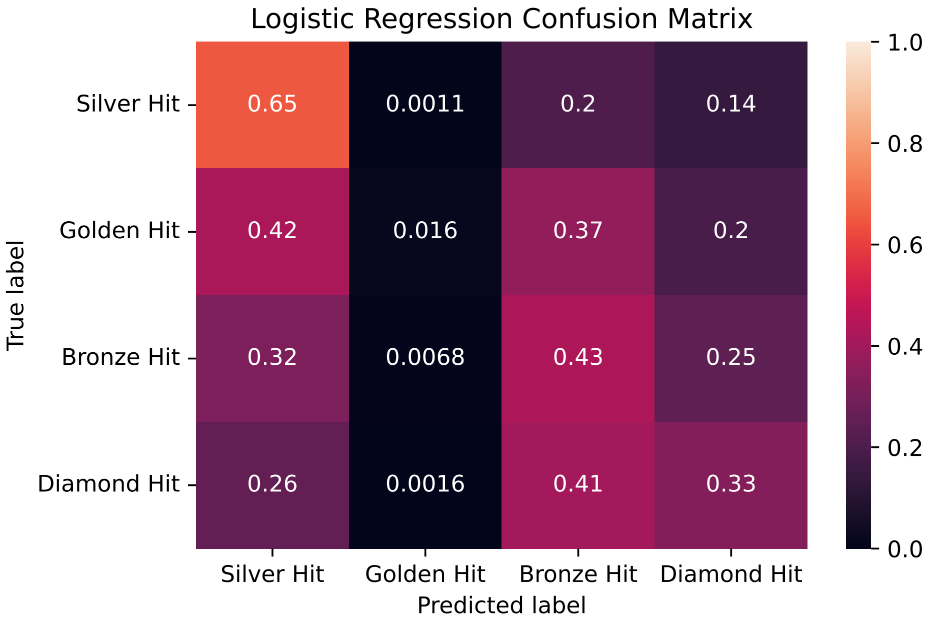


Figure 6: Confusion Matrix of Logistic Regression Model

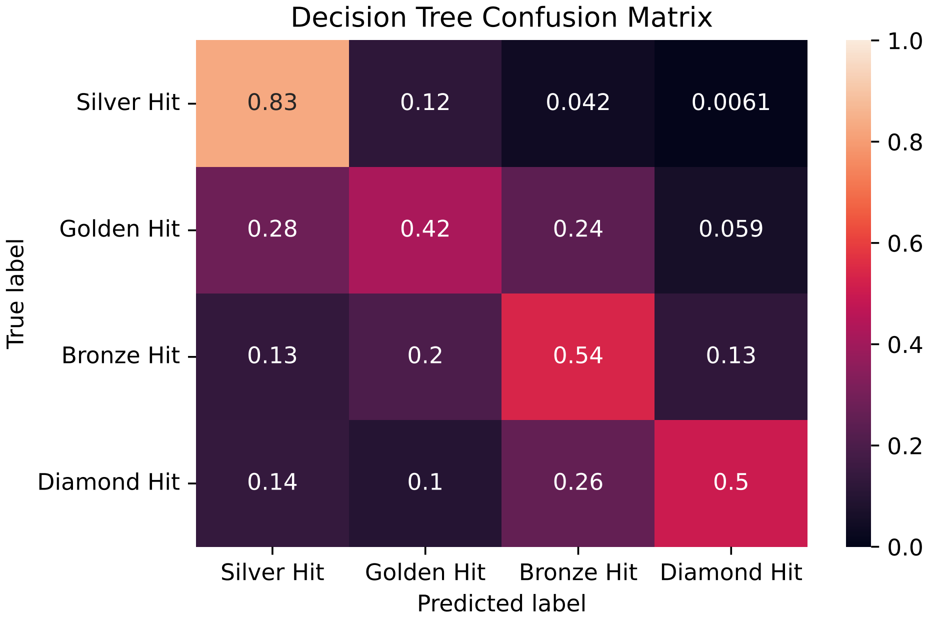
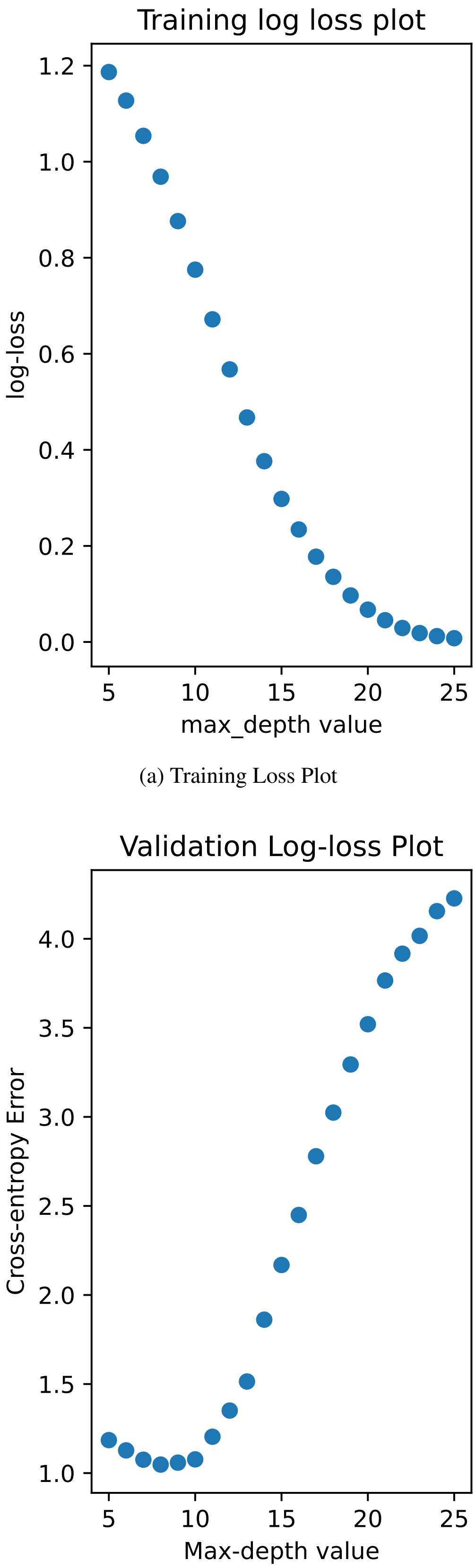


Figure 7: Confusion Matrix of Decision Tree Model



(a) Training Loss Plot

(b) Validation Loss Plot

Figure 8: Cross Entropy Loss of Decision Tree Classifier

Statement of Contribution

Common contribution: We met every week to discuss any issues we had encountered. All decisions were made with common consensus through the meetings or discussion on the Google Doc.

- Shivaz Sharma (s2444842):
  - Worked on writing the script for scraping the audio features with sparsh.
  - Worked on pre-processing the dataset.
  - Worked on the hyperparameter tuning, model training and evaluation metrics. Code clean up and optimization
- Sparsh Rawal (s2314252):
  - Setup Git repository for code maintenance.
  - Worked on pre-processing the dataset.
  - Prepared the dataset by calculating the median position of each song and saving it on the csv.
  - Merged the audio features extracted by shivaz to the main csv
- Tom Weatherall (s2436812):
  - Worked on reporting from scratch and maintained overleaf.
  - Worked on pre-processing the data.
  - Did PCA analysis and visualization.
  - Researched relevant literature.