Predict-the-Hit: Prediction of Hit Songs based on Multimodal Data

Samyak Jain[1*], Parth Chhabra[1*], Sarthak Johari[1*]

{samyak19098, parth19069, sarthak19099}@iiitd.ac.in

[1*] Undergraduate Student, Department of Computer Science, Indraprastha Institute of Information Technology, Delhi

DOI: 10.29322/IJSRP.12.09.2022.p12940
http://dx.doi.org/10.29322/IJSRP.12.09.2022.p12940

Paper Received Date: 15th August 2022
Paper Acceptance Date: 15th September 2022
Paper Publication Date: 26th September 2022

Abstract-

Hit Song Science concerns the possibility of predicting whether a song will be a hit before its distribution using automated means such as machine learning software. This has motivated to dig deeper to unravel how different audio features would help to predict if a song would feature in the Billboard Top 100 Chart and build a two-way usability model - both for the musicians composing the music and the labels broadcasting it. The work in this paper also aligns with our team's vision of exploring real-world applications of machine learning techniques and making them useful in common domains. In this paper, prediction models on data from Million Songs Dataset (MSD), Billboard, and Spotify using machine learning techniques have been explored, and low-level & high-level feature engineering techniques are applied. Finally, a comparison of their performances using various performance metrics has been carried out.

Index Terms- Hit Song, Spotify, Billboard, High-level, Low-level, CNN, MLP, PCA, Logistic Regression, Random forest.

I. INTRODUCTION

With the forever-expanding music industry and the number of people keen on listening to popular music, it becomes essential to come up with a classifier that can predict whether a song is 'hit' or 'non-hit' to help musicians and music labels [6]. Such a-priori prediction would become useful for music streaming industries to identify potentially interesting songs and their writers [9]. In the field of research, scientists are also fancied by the characteristics that make a song popular [10,11]. Therefore, motivated by this idea, a scheme has been proposed that focuses on developing Machine Learning (ML) models to predict whether a song is a "hit" or a "miss". For carrying out this task, data is collected from Billboard, Spotify, and Million Song Dataset, and also considered several features of a song, like audio features and related artist data, and, based on that, applied machine learning based classification algorithms to develop models that could help us achieve the desired classification. Through a series of experiments, it is observed that the proposed model is able to correctly predict what choices of a particular feature make a positive impact so that the musicians and music engineers can plan accordingly to give their songs the best chance of being classified as a 'hit.

We have also included both low-level and high-level analyses. A low-level analysis uses the audio data and raw audio features like spectrograms to train models. The high-level analysis includes using high-level human understandable features like danceability, loudness, and acousticness.

The proposed scheme is depicted in Figure 1.

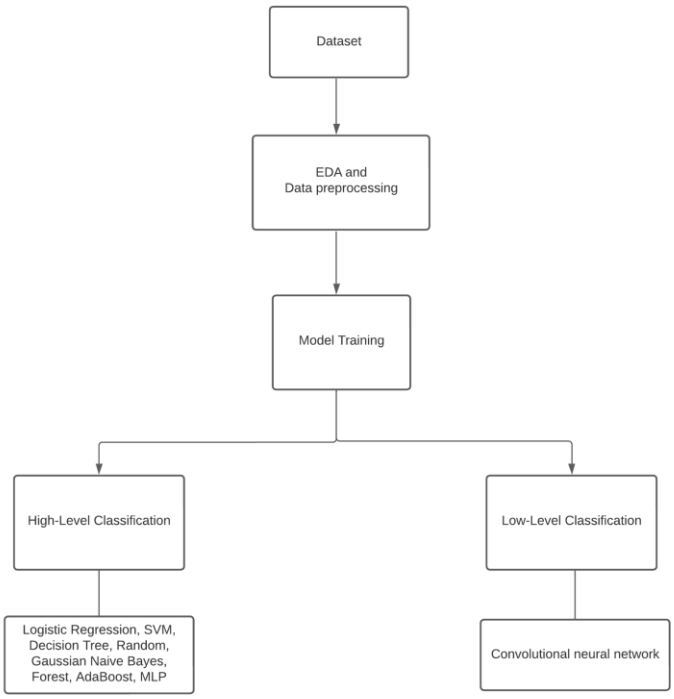


Fig 1. Proposed Scheme

The rest of the paper is organized as follows: Section II includes a comprehensive literature survey. The description of the datasets considered is given in Section III. The detailed methodology is explained in Section IV, with results represented in Section V. Section VI concludes the paper.

II. LITERATURE SURVEY

A significant amount of work has been done in the field of hit song science in terms of hit prediction for songs based on multiple types of features. Pachet and Roy [4] used external features extracted from the musical ecosystem, such as social media presence, and internal features extracted from audio to predict a song's popularity. They used 632 manually labeled features for each song to encapsulate all the internal and external features but could not develop an accurate model and concluded it could not be done by state-of-the-art machine learning techniques.

Ni et al. [3] focused on using low-level internal features to predict a song's popularity. They used a classifier that was a time function along with a shifting perceptron learning agent. It could outperform a random oracle significantly and obtained a 60% accuracy with its predictions but was limited to UK's billboards and could not generalize well.

Singhi and Brown [2] used 31 rhyme, syllable, and meter features like syllables per line, rhymes per line, etc., to develop their Bayesian network model, which gave them a precision of 21.4% and recall of 45.1%. They had an imbalanced dataset with around 7% of total songs as hits and the rest non-hits.

Yang et al. [1] experimented with deep learning models like convolutional Neural Network and JYnet model for supervised pre-training and auto-tagging and also used a combination of both of these. Compared to shallow models, their experiments produced promising results on two different datasets (Mandarin & Western Pop).

III. DATASET

A. Dataset Extraction

We used a subset of the Million Song Dataset (MSD) [7] of 1 million songs, of which we further used a one-tenth subset. We extracted each song's high-level audio features and related artist data using Spotipy [8] to query the Spotify API and further obtained 29,371 data points after narrowing it down to songs released between 2006 and 2020. Using billboard.py to query the Billboard API, we also collected 4,778 songs featured on Billboard Top 100 distributed equally between 2006 to 2020, got the audio feature and artist detail for 4,063 songs, and removed the songs that were not released between 2006 to 2020. Some overlapping songs between billboard and MSD data were removed, and finally, we had data of about 9,758 songs, out of which 3,796 were Billboard Hits and 5,962 Non-Hits. A Billboard Hit is labeled as 1, and a non-hit as 0.

We performed both high-level and low-level classification. High-level classification uses high-level human understandable features like danceability, loudness, and acousticness, whereas low-level classification is done by sampling from the actual audio data and using the audio signal spectrogram. *Note that all models except CNN use high-level features; CNN uses low-level audio data for classification.*

The low-level analysis is helpful as it provides an unbiased classification, i.e., it does not consider high-level factors like artist popularity. This is important because a less-known artist may release a potentially hit song.

B. Data Pre-Processing and Analysis

Out of the extracted 23 total features, we initially reduced the dataset to 16 numeric features that can be used to build classification models. We checked for any missing features (NaN, NULL values, etc.) and found none, as they were already removed during data extraction.

a. Data Standardization

We calculate the skewness of each feature. Since some of the features are skewed from the ideal normal distribution, thus, we have used the Yeo-Johnson power transform to fix the skew and standardize the data. The resultant feature distributions approximate normal distributions with zero mean and unit standard deviation.

b. Feature Selection

To reduce the redundancy and dimensionality of the dataset, we have made a correlation heatmap of the features and have noticed

that - Energy has a high positive correlation with Loudness (0.71) and a high negative correlation with Acousticness (-0.66). Also, Followers and artist popularity have a high positive correlation (0.53), with Artist popularity having a high positive correlation (0.76) with the output label.

Furthermore, we have used pair plots to verify the correlations observed from the heatmap. We observed that energy increases approximately linearly with loudness; the same goes for artist popularity and followers.

Therefore, since the effect of energy and followers on the output label can be modeled just by using loudness and artist popularity, we drop energy and followers. We chose to drop energy because it has a high correlation with two quantities and a low correlation with the output label. Also, loudness approximates a normal distribution better than energy. Artist popularity is preferred as it has a very high correlation with the output label, indicating a high deterministic power.

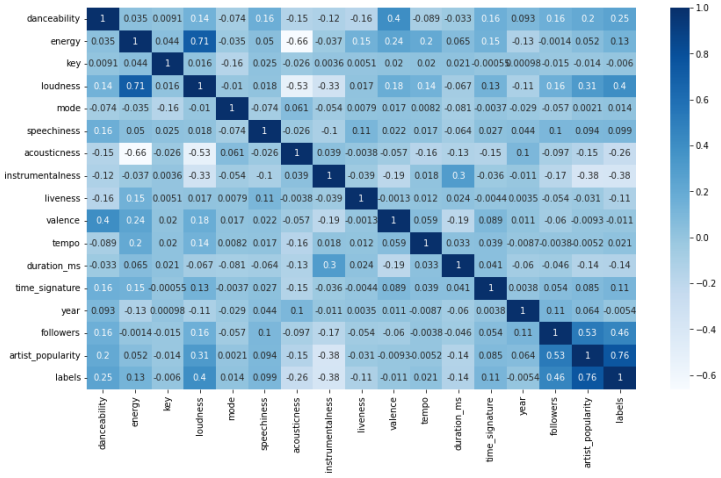


Fig 2. Correlation Heatmap

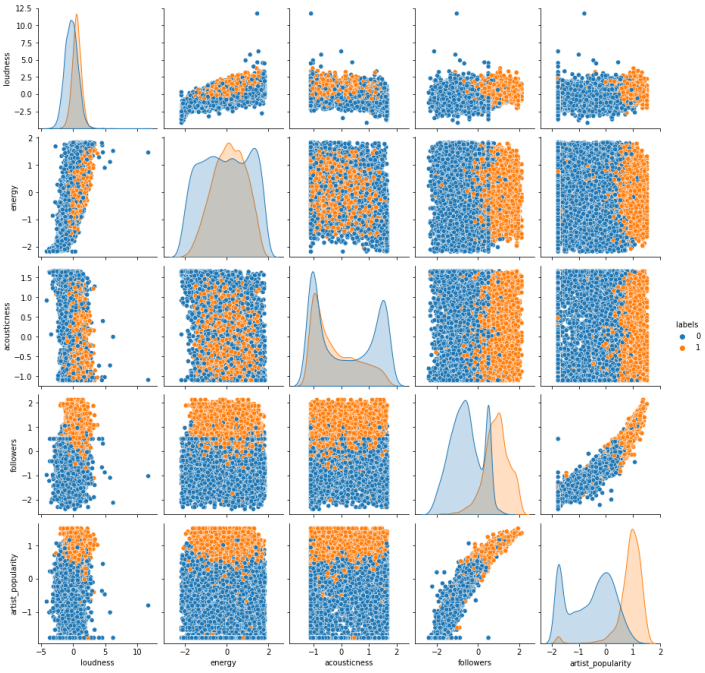


Fig 3. Pair-plots of selected features

c. Handling Outliers

Since the data is standardized for a particular feature; we identify the outliers as data points with the absolute value of a Z-score greater than 2.6. Any data point that lies more than 2.6 standard deviations away from the mean is considered an outlier. Therefore, we have tried the threshold Z-score value for all values in the range [2, 4] with a step size of 0.1. The threshold value of 2.6 gave the best results, and after removing the outliers, we are left with 3,690 hits (output label = 1) and 5,486 non-hits (output label = 0). 40% of the data points are hits, and 60% are non-hits.

Finally, after doing all the above preprocessing steps, the revised dataset has 9,176 data points and 14 features.

d. Dimensionality Reduction

T-Distributed stochastic neighbour embedding (t-SNE)

t-SNE minimizes the divergence between two distributions: a distribution that measures pairwise similarities of the input objects and a distribution that measures pairwise similarities of the corresponding low-dimensional points in the embedding.

We have used t-SNE to reduce the dimensions of the data points to two dimensions. Through this, we were able to visualize higher dimensional data and get a sense of similarity between data points.

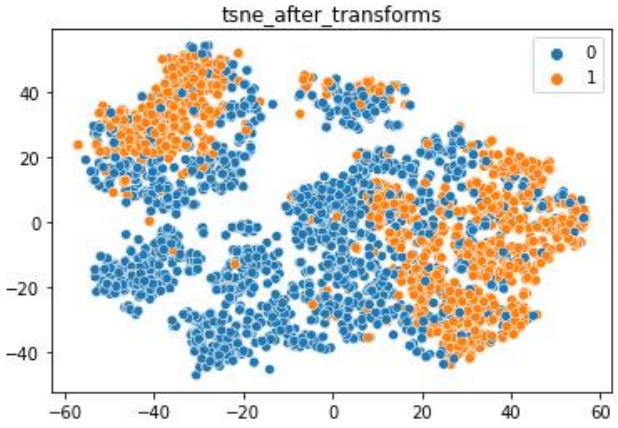
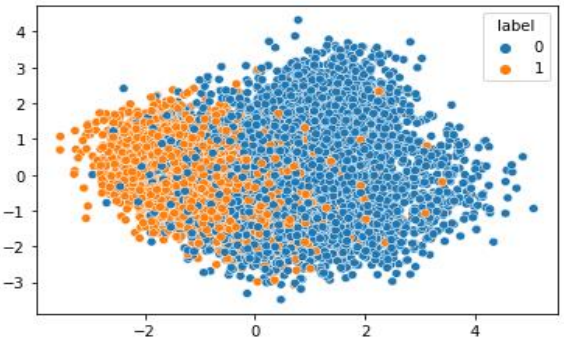


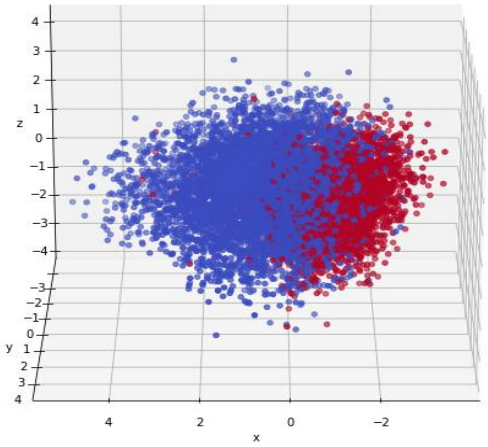
Fig 4. t-SNE plot (3000 random datapoints)

Principal Component Analysis (PCA)

PCA is used to reduce the number of data dimensions while retaining maximum information stored in it. It tries to maximize the variation retained from the original data distribution. We used PCA to reduce the dimensions of the data points to three dimensions and plot the same. The explained variance ratio per component that we got is as follows:- [9.99952230e-01, 4.77703879e-05, 6.88124530e-12]. Furthermore, from the below plots, we could infer that there is a high separability between the classes (hit and not-hit).



(a) 2-dimensional plot



(a) 3-dimensional plot

Fig 5. Dimensionality reduction using PCA

IV. METHODOLOGY

Through the proposed methodology, we aim to make a prediction for a song based on its audio features and corresponding artist-related data (popularity) and perform binary classification by predicting it as a billboard hit (label 1) or non-hit (label 0). To perform this task, the following classification models have been used: logistic regression, Gaussian Naive Bayes, Decision Trees, Random Forest, SVM, AdaBoost (base classifier: decision tree), MLP, and CNN. Hyperparameter tuning is also performed using the grid search technique over selected parameters to arrive at the best results. For evaluating the performance of the different classifiers, we have used the performance metrics, including accuracy, precision, recall, and AUC score for ROC curves.

A. Models and their details

To test our models, we split the dataset into training and testing sets through a 70:30 split, with the training set consisting of 6,268 samples and the testing set having 2,687 samples.

a. Logistic Regression

It is a linear classification supervised model that uses a logistic function for classification. Logistic regression is categorized into two types - binary logistic regression and multi-class logistic regression. But, as our task is predicting whether or not a song is hit or not we have used the binary logistic regression.
To come up with the optimal solution, logistic regression minimizes the log-loss logit function.

b. Decision Tree

A decision tree is a non-parametric supervised model for classification and regression tasks. It has a hierarchical tree structure with nodes and edges. The nodes represent the outcomes, and the edges denote the rules by which the tree is made. To make this tree-like structure, the decision tree has two metrics - Gini Ratio and Entropy. For our model, we have used the Gini ratio as the decision metric.

c. Random Forest & AdaBoost

Random forest is an ensembling technique for classification and regression tasks in which multiple decision trees are used during training. For the classification task, the output of the random forest classifier is governed by the voting process of the decision trees in which the majority class is picked.

AdaBoost, short for Adaptive Boosting, is a boosting technique that is used as an ensembling method. This technique assigns higher weights to incorrectly classified samples, thus helping reduce bias and variance in supervised learning tasks. In simple words, AdaBoost helps in converting weak decision stumps into a strong binary classifier.

d. Gaussian Naive Bayes

Gaussian Naive Bayes is a generative model which assumes that each class follows a gaussian distribution and independence of features meaning that the covariance matrices are diagonal matrices

and also support continuous-valued features. The likelihood of the features is computed using the below-mentioned probability measure.

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

e. Stochastic Gradient Descent (SGD) Classifier

The SGD classifier implements a linear model using stochastic gradient descent (SGD) in which the gradient of the loss function is estimated for each sample at a time, thereby updating the model parameters. We implemented this SGD classifier using the Sckit-Learn library.

f. Support Vector Machine (SVM)

SVM is a supervised learning algorithm for classification and regression problems. It tends to estimate parameters w and b that describe an optimal hyperplane to separate samples of the two classes.

However, soft margin SVM was developed to relax the constraints of hard margin SVM to tackle linearly inseparable problems. The soft margin SVM estimates a hyperplane that optimizes the below function.

$$\omega^*, b^*, \xi^* = \underset{\omega,b,\xi}{\operatorname{argmin}} \frac{1}{2}||\omega||^2 + C\sum_{i=1}\xi_i$$

$$\text{where,} \quad y_i(\omega^T x_i + b) \geq 1 - \xi_i; \quad \forall i = 1,\ldots,m. \qquad (1)$$

$$\xi_i \geq 0; \quad \forall i = 1,\ldots,m.$$

In the above equation, parameter C determines the penalty of misclassification error which is represented by ξ_i. In our work, value of parameter C remains 1 throughout the experiments.

h. Multi-Layer Perceptron (MLP)

Neural networks have become one of the most popular model architectures in today's world after the rise of deep learning. In this paper, we have proposed a 3-layer MLP consisting of an input layer followed by a hidden layer with 100 nodes and a logistic activation function followed by a 2-node output layer with a sigmoid as the activation function. We used SGD as the solver and a learning rate of 0.01.

i. Convolution Neural Network (CNN)

In neural networks, CNN is one of the most important classifiers for doing image classification tasks and extracting spatial information from the data. In our paper, we have used four convolution layers with ReLU as the activation function and applied batch normalization as well. Finally, after flattening the features, a linear classifier was applied at the end with a sigmoid activation function.

B. Performance Metrics

To evaluate our models, we have used the following performance metrics:-

- Accuracy

$$\frac{TP + TN}{TP + TN + FP + FN}$$

- Precision

$$\frac{TP}{TP + FP}$$

- Recall

$$\frac{TP}{TP + FN}$$

- F1-score

$$\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- ROC Curve - Probability curve that shows the trade-off between sensitivity (True Positive Rate) and specificity (1 - False Positive Rate).

AUC Score - Area under a ROC curve which represents the measure of separability and, therefore, how well a model is capable of predicting correctly.

Here, TP - Number of True Positives
TN - Number of True Negatives
FP - Number of False Positives
FN - Number of False Negatives

Generally, high accuracy, precision, and recall are desired. Precision (ratio of true positives to total positive predictions) is particularly important as we want to reduce the number of false positives (non-hits predicted as hits). This is necessary because music labels would not want to invest in songs that are not potential hits. ROC curve is plotted, and Area under the curve (AUC) is observed. Higher AUC is desired and indicates better prediction ability of the model and good performance.

V. RESULTS & ANALYSIS

A.High-Level Classification

We trained our models using 5-fold (best overall 'k' from 2 to 10) cross-validation while performing a grid search on the hyperparameters. Using the best model obtained by grid search (using precision as the scoring metric as we want to minimize the number of false positives), we calculated the accuracy, precision, recall, and F1 score. A summary of the results on the test data in Table 1.

Model	Accuracy	Precision	Recall	F1-score
LR(BGD)	0.927	0.899	0.925	0.912
DT	0.921	0.934	0.871	0.901

RF	0.939	0.943	0.907	0.925
GNB	0.860	0.775	0.931	0.846
LR(SGD)	0.908	0.870	0.913	0.891
ADA	0.914	0.881	0.909	0.895
MLP	**0.972**	**0.954**	**0.976**	**0.965**
SVM-LINEAR	0.915	0.884	0.909	0.896

Table 1. Performance evaluation for classification models

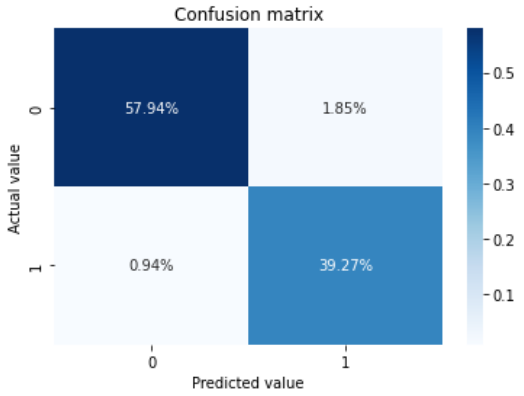


Fig 7. MLP: Confusion Matrix

From the above experiments, we could infer that the MLP classifier was able to perform the best in terms of all the performance metrics. This is not surprising since ANNs are very powerful function approximators.

Another thing to notice was that both the Gaussian Naive Bayes(GNB) and Logistic Regression(LR) classifiers performed poorly on the test set. The reason behind the same could be that the data is not linearly separable, as evident from the t-SNE plots. Also, the features are not completely independent, as is evident from the correlation heatmap in Fig 1, which is a necessary condition for a GNB model to perform well.

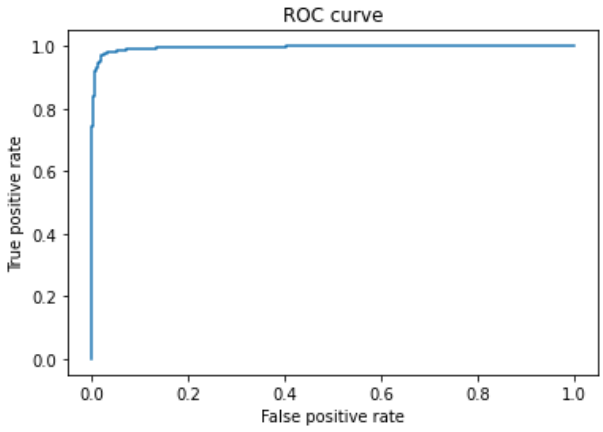


Fig 6 MLP: ROC Curve (AUC = 0.995)

Since the dataset is not linearly separable thus, we expected the Decision Tree and Random Forest classifiers to perform well, which they do. Since random forests are an ensemble model on decision trees, it combines the output of various decision trees and thus performs better than a single decision tree.

Logistic regression (LR using both BGD and SGD) has comparatively lower precision than Decision Trees and Random Forest. This is expected since logistic regression assumes that the data is linearly separable, which is not true in this case.

Furthermore, we tried the SVM classifier with various kernels (RBF, linear and radial), with the linear kernel performing the best. It gave the best performance when compared with LR, AdaBoost & GNB. This is because it tries to find the optimal hyperplane which classifies unseen data better.

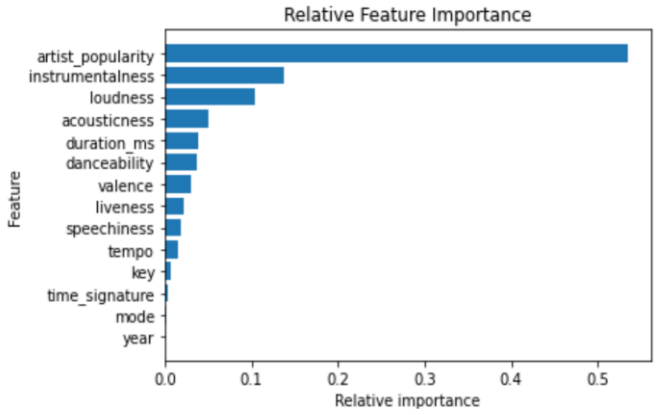


Fig 8. Random Forest: Relative Feature Importance

We also observed relative feature importance in the Random Forest classifier. As expected, artist popularity has the highest relative feature importance and dominates by a huge margin. This is justifiable because we expect artist popularity to be based on the previous performance of the artist on similar metrics (getting featured on Billboard, awards, etc.). The year has low importance as hit songs are equally distributed across years, and there is no linear dependence of year with other features. The trend can also be justified by observing the correlation heatmap of features with

labels.

B. Low-Level Classification

The dataset used for low-level classification contains 7,408 songs (some songs had no audio preview available and so were removed). Due to computational limitations, we sample audio data of only 10 seconds. We do an 80:20 train-test split. The sampling rate of the audio is 44160 (which is the standard sampling rate). This creates data samples of size 331,264.

We used the spectrogram data extracted from the audio files for low-level classification. A CNN with four convolutional layers and one fully-connected layer is used. The model is trained for 30 epochs with a batch size of 128. The initial learning rate is 0.001. Adam optimizer is used for backpropagation.

The results of the model are summarized in Table 2.

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
CNN	0.8751	0.8409	0.6964	0.7619

Table 2. CNN Performance Evaluation

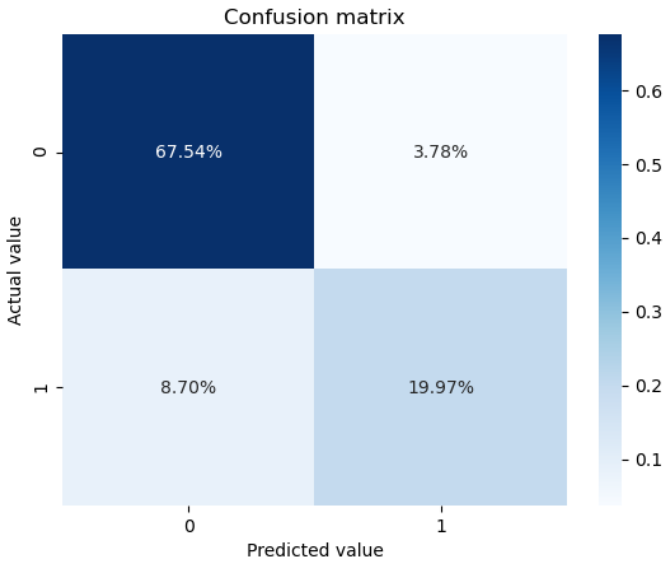


Fig 9. CNN: Confusion Matrix

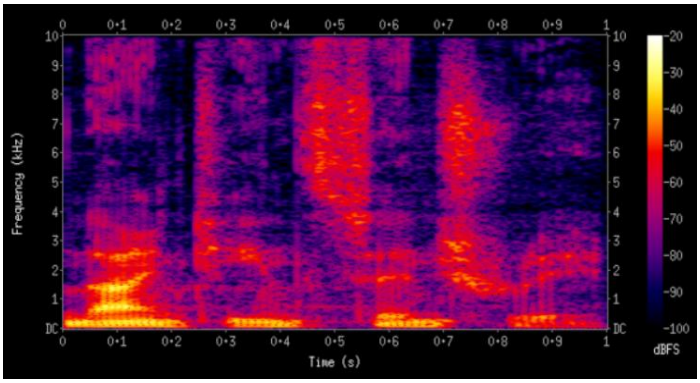


Fig 10. Spectrogram of an audio sample

Even though this model performs only marginally better than Gaussian Naive Bayes, it is essential to note that this uses only the audio data of the song. The high-level features used in the previous model include artist popularity and followers, which are highly correlated with whether an artist will release a hit song. Using only the raw audio data gives an unbiased result, i.e., it does not consider any biases created by the artist. It is also possible that a less-known artist may release a song that becomes a hit.

VI. CONCLUSION & FUTURE WORK

Since the number of people listening to music is increasing daily, it has become essential to provide listeners with the best hit songs possible. Thus, our paper focuses on predicting whether a song will turn out to be a hit or a miss. We followed various data extraction methods using APIs and different data visualization and analysis techniques to do so. Apart from this, we applied data preprocessing and EDA methods to simplify our original dataset and remove any redundancies. For this binary classification task, we tried different classification models like logistic regression, decision trees, random forests, MLP, SVM, CNN, AdaBoost, and Naive Bayes and compared them based on various performance metrics like accuracy, precision, recall, ROC curve, and AUC score. Other than that, we also provided some inferences about the predictions made by these models and explored different kinds of low-level and high-level feature data.

As part of future work, we believe that the low-level analysis can be improved by making the network deeper (which will require more computational resources). Also, combining high-level and low-level features can give even better results. Currently, we only use spectrogram data for low-level analysis. There are other features as well that can be extracted from audio and can be used for classification models. Also, we aim to train and test our models on a larger dataset.

REFERENCES

[1] Yang, L.C., Chou, S.Y., Liu, J.Y., Yang, Y.H. and Chen, Y.A., 2017, March. Revisiting the problem of audio-based hit song

prediction using convolutional neural networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 621-625). IEEE.'

[2] Singhi, A. and Brown, D.G., 2014. Hit song detection using lyric features alone. *Proceedings of International Society for Music Information Retrieval*, *30*.

[3] Ni, Y., Santos-Rodriguez, R., Mcvicar, M. and De Bie, T., 2011, December. Hit song science once again a science. In *4th International Workshop on Machine Learning and Music*.

[4] Pachet, F. and Roy, P., 2008, September. Hit Song Science Is Not Yet a Science. In *ISMIR* (pp. 355-360).

[5] Zangerle, E., Vötter, M., Huber, R. and Yang, Y.H., 2019. Hit Song Prediction: Leveraging Low-and High-Level Audio Features. In *ISMIR* (pp. 319-326).

[6] Liu, J.Y. and Yang, Y.H., 2016, October. Event localization in music auto-tagging. In *Proceedings of the 24th ACM international conference on Multimedia* (pp. 1048-1057).

[7] Google (2022) *Million Song Dataset.* Available at: http://millionsongdataset.com/ (Accessed: 7 May 2022)

[8] Google (2022) *Spotipy Library.* Available at: https://spotipy.readthedocs.io/en/master/# (Accessed: 6 May 2022)

[9] Silk, H., Santos-Rodriguez, R., Mesnage, C., De Bie, T. and McVicar, M., 2016. Data science for the detection of emerging music styles. In *Extended abstracts for the Late-Breaking Demo Session of the 17th International Society for Music Information Retrieval Conference*.

[10] Middleton, R., 1990. *Studying popular music*. McGraw-Hill Education (UK).

[11] Lopes, P.D., 1992. Innovation and diversity in the popular music industry, 1969 to 1990. *American sociological review*, pp.56-71.

AUTHORS

First Authors

Samyak Jain, CSE Undergraduate 2023, Indraprastha Institute of Information Technology, Delhi and email address: samyak19098@iiitd.ac.in

Parth Chhabra, CSE Undergraduate 2023, Indraprastha Institute of Information Technology, Delhi and email address: parth@19069@iiitd.ac.in

Sarthak Johari, CSE Undergraduate 2023, Indraprastha Institute of Information Technology, Delhi and email address: sarthak19099@iiitd.ac.in

Correspondence Author – Samyak Jain, CSE Undergraduate 2023, Indraprastha Institute of Information Technology, Delhi and email address: samyak19098@iiitd.ac.in, +91-9582919388