Prediction of Hit Songs based on Multimodal Data

Samyak Jain
2019098

Parth Chhabra
2019069

Sarthak Johari
2019099

Yash Mathne
2019129

1. Abstract

Hit Song Science concerns with the possibility of predicting whether a song will be a hit, before its distribution using automated means such as machine learning software. This motivated us to dig deeper to unravel how different audio features would help in predicting if a song would feature in the Billboard Top 100 Chart and build a two way usability model - both to the musicians composing the music and the labels broadcasting it.The project also aligns with our team's vision of exploring real world applications of machine learning techniques and making them useful in common domains. We explore prediction models on data from MSD, Billboard and Spotify using ML techniques.

2. Introduction

With the forever-expanding music industry and the number of people keen on listening to popular music, it becomes very important to come up with a classifier that can predict whether a song is 'hit' or 'non-hit' to help the musicians and the music labels. Using the data collected from Billboard, Spotify and Million Song Dataset, our project takes into account several features for a song like audio features and related artist data and based on that uses Machine Learning based classification algorithms to develop models that can help us achieve the desired classification.Our goal is to make accurate as well precise predictions. Our models will indicate 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 'hit'.

We include both low-level and high-level analyses. Low-level analysis is done using the audio data and extracting raw audio features like spectrograms to train models. High-level analysis includes using high-level human understandable features like danceability, loudness, and acousticness.

3. Literature Survey

A significant amount of work has been done in predicting a song's chances of becoming a success.

1. [2] Used external (extracted from the musical ecosystem such as social media presence) and internal features (extracted from the audio) to predict a song's popularity.Used 632 manually labelled features for each song to encapsulate all the internal and external features.Could not develop an accurate model and concluded it could not be done by the state of the art machine learning.

2. [1] Focused on using low-level internal features to predict a song's popularity. Used a classifier which was a function of time along with a shifting perceptron learning agent.Significantly outperformed a random oracle and obtained a 60% accuracy with it's predictions. Limited to UK's billboards and may not generalise well.

3. [3] Focused on using low-level and high-level internal features to predict the commercial success of a song. Added the date as a high-level feature to contextualise the song temporally to improve accuracy. Used wide and deep network regression to obtain a 75% accuracy.

4. Dataset

4.1. Data Extraction

We used a subset of the Million Song Dataset 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 to query Spotify API and obtained 29,371 data points after narrowing down to songs released between 2006 to 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 and 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 labelled as 1 and a non-hit as 0.

We perform both high-level and low-level classification. High-level classification is done using 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.*

Low-level analysis is useful as it provides an unbiased classification i.e. it does not take into account high level factors like artist popularity. This is important because a less known artist may release a potentially hit song.

4.2. Preprocessing 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.

4.2.1 Standardization of Data

We calculate the skewness of each feature. Some features are skewed from the ideal normal distribution and so we use Yeo-Johnson power transform to fix the skew and also standardize the data. The resultant feature distributions approximate normal distributions with zero mean and unit standard deviation.

4.2.2 Feature Selection by observing correlation

We plot the correlation heatmap of the features and make the following observations:

1. Energy has a high positive correlation with loudness (0.71) and high negative correlation with acousticness (-0.66).

2. Followers and artist popularity have a high positive correlation (0.53). Artist popularity also has a high positive correlation (0.76) with output label.

We then use pair-plots to verify the correlations observed from the heatmap. We observe that energy increases approximately linearly with loudness and the same goes for artist popularity and followers. Therefore, since the effect of energy and followers on the output label can be modelled just by using loudness and artist popularity respectively, we drop energy and followers. We chose to drop energy because it has 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 very high correlation with output label indicating a high deterministic power.

4.2.3 Handling Outliers

Since the data is standardized, for a particular feature we identify the outliers as data points with the absolute value of Z-score greater than 2.6. This means that any data point which lies more than 2.6 standard deviations away from the mean is considered an outlier.We tried the threshold Z-score value for all values in the range [2, 4] with step-size of 0.1. Threshold value of 2.6 gave best results. 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.

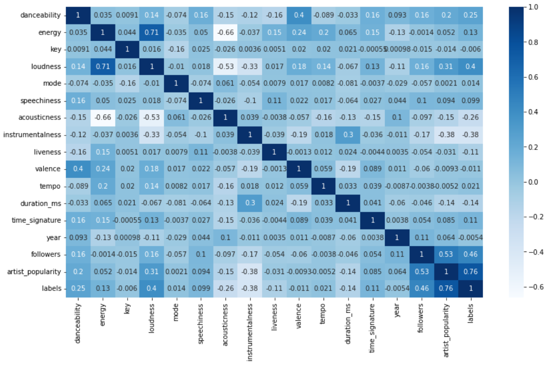


Figure 1. Correlation heatmap

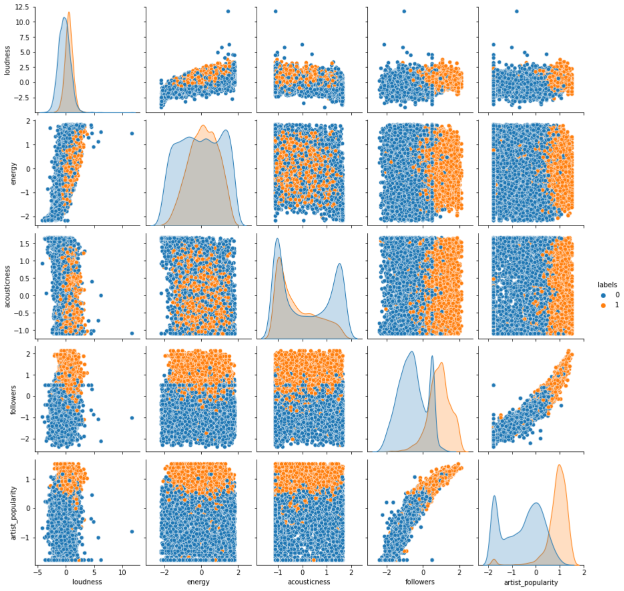


Figure 2. Pair-plots of selected features

After doing all the preprocessing steps, the final dataset has 9,176 data points and 14 features.

4.2.4 Dimensionality reduction

t-SNE:
“t-Distributed stochastic neighbor embedding (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 use t-SNE to reduce the dimensions of the data points to two dimensions. Through this we can visualise higher dimensional data and get a sense of similarity between data points.

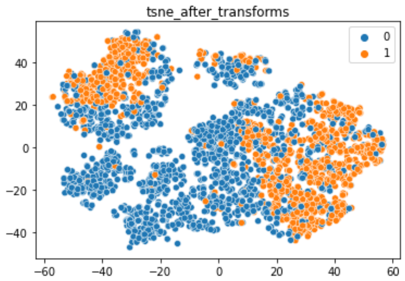
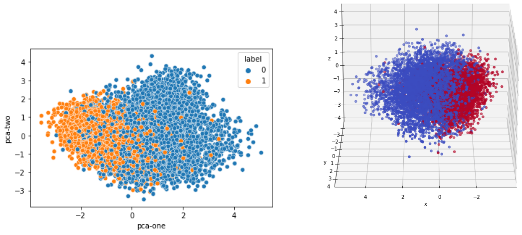


Figure 3. t-SNE plot (3000 random data points)

PCA:
PCA is used to reduce the number of dimensions of data while trying to retain maximum information stored in it. It tries to maximize the amount of variation retained from the original data distribution.We use PCA to reduce the dimensions of the data points to three dimensions and plot the same. The explained variance ratio per component is [9.99952230e-01, 4.77703879e-05, 6.88124530e-12].



(a) (b)

Figure 4. PCA plots: (a) 2-D (b) 3-D

5. 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). For this, we used the following classification models: logistic regression, Gaussian Naive Bayes, Decision Trees, Random Forest, SVM, AdaBoost (base classifier: decision tree), MLP and CNN. We also perform hyperparameter tuning using grid search technique over selected parameters to arrive at the best results. The metrics used overall for evaluation include accuracy, precision, recall and AUC score for ROC curves.

5.1. Clustering

We perform k-means clustering on our data for $k = 3$. The clusters can be observed below: We choose the value of k as 3 because t-SNE plots suggest that the data is separable into 3 groups.

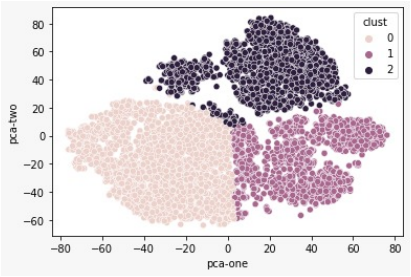


Figure 5. t-SNE plot (3000 random data points)

5.2. Model and details

We split the data into a training set and testing set through a 70:30 split. The training set has 6,268 samples and the testing set has 2,687 samples.

1. **Logistic regression:** It is a linear binary classification model that uses logistic function for classification.We use it with log loss and L2 regularization.

2. **Decision Tree:** It is a classification model that creates decision boundaries to classify data. It creates a tree based model using the features and continuosly splits the data based on the parameters.

3. **Random Forest:** It is an ensemble model over decision trees that combines the output from various decision trees. Since it is an ensemble model, it tends to perform better than regular decision trees.

4. **Gaussian Naive Bayes:** It is a probabilistic model that assumes feature independence to assign probabilities to output classes.

5. **SGD Classifier:** We use SGD classifier to perform logistic regression using stochastic gradient descent and L1 regularization.

6. **SVM:** It is a classification model that generates optimal hyperplanes for separating classes. If the data is not linearly separable, we can use techniques like soft-margin and kernel trick (which increases the dimensions of data). Increasing the dimensions of data usually makes the data linearly separable (Cover's theorem).

7. **AdaBoost:** It is a statistical classification algorithm that boosts the results of weak models.

8. **MLP:** MLP is a class of feedforward artificial neural networks. Strictly speaking, it comprises of multiple layers of perceptrons with threshold activation.

9. **CNN:** CNN is a class of artificial neural networks that usually used for image or image-like data.

5.3. Performance Metric

We use accuracy, precision and recall to evaluate the performance overall. In general, high accuracy, precision and recall is 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 curve (AUC) is observed. Higher AUC is desired and indicates better prediction ability of model and good performance.

6. Result and Analysis

6.1. High-Level Classification

We train our models using 5-fold (best over all ’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 calculate accuracy, precision, recall and F1 score. A summary of the models on the test data is given below:

Model	Accuracy	Precision	Recall	F1
LR(BGD)	0.9270	0.8999	0.9259	0.9127
DT	0.9218	0.9341	0.8717	0.9018
RF	0.9397	0.9436	0.9078	0.9254
GNB	0.8604	0.7751	0.9313	0.8461
LR(SGD)	0.9084	0.8708	0.9132	0.8915
ADA	0.9142	0.8810	0.9096	0.8951
MLP	**0.9720**	**0.9549**	**0.9765**	**0.9656**
SVM(LIN)	0.9157	0.8841	0.9096	0.8967

Table 1. Performance evaluation of classification models

MLP performed the best among all the models chosen for analysis in all the performance metrics. This is not surprising since ANNs are very powerful function approximators.

As is evident from the t-SNE plot, the data is not linearly separable. Decision Trees(DT) work well with such data and thus give high precision and accuracy. Since random forests is an ensemble model on decision trees, it combines the output of various decision trees and thus performs better than a single decision tree.

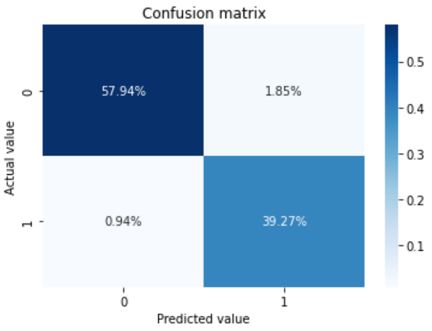


Figure 6. MLP: Confusion Matrix (with percentages)

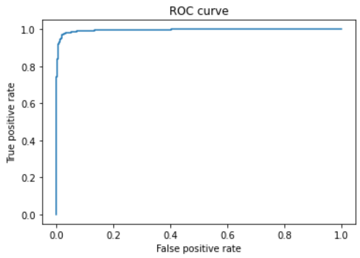


Figure 7. MLP : ROC Curve (AUC = 0.995)

Gaussian Naive Bayes(GNB) gives the worst results. This is expected as Gaussian Naive Bayes assumes the features to be independent which is not true (as is evident from the correlation heatmap).

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

Different SVMs were run. Among them linear SVM gave the best results. It gives better results ADA, LR and GNB. This is because it tries to find the optimal hyperplane which classifies unseen data better.

We also observed relative feature importance in Random Forest classifier(Figure 5). As expected, artist

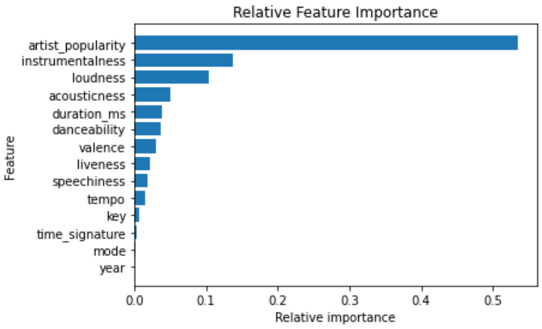


Figure 8. Random Forest: Relative feature importance

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 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 correlation heatmap of features with labels.

6.2. 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 a $80 : 20$ 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 use the spectrogram data extracted from the audio files for low-level classification. A CNN with 4 convolutional layers and 1 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 back propagation.

The results of the model are summarised below:

Model	Accuracy	Precision	Recall	F1
CNN	0.8751	0.8409	0.6964	0.7619

Table 2. CNN performance evaluation

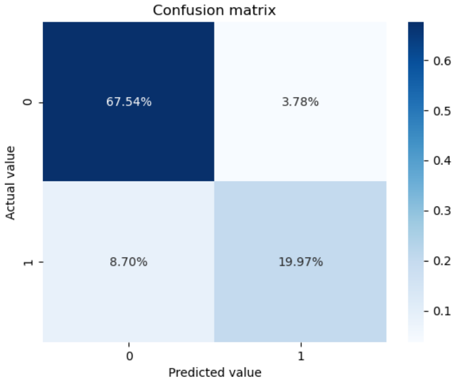


Figure 9. CNN: Confusion matrix (with percentages)

Even though this model performs only marginally better than gaussian naive bayes, it is important to note that this uses only the audio data of the song. The high-level features used in previous model includes artist popularity and followers which has a high correlation with whether an artist will release a hit song. Using only the raw audio data gives an unbiased result i.e. it does not take into account any biases created by the artist. It is possible that a less known artist may release a song that becomes a hit.

7. Conclusion

7.1. Learning

Through the project we have learnt about data extraction methods using APIs. We have learnt about techniques related to data visualization and analysis. We learnt about methods of data preprocessing and EDA techniques. We are able to apply different classification models like logistic regression, decision trees, random forests, MLP, SVM, CNN, AdaBoosting and Naive Bayes for prediction. We also learnt about the importance of data in machine learning projects and explored different kinds of dataset. We also learnt about feature analysis for the same and the importance of exploring different kinds of low level and high level feature data. We got to learn about ways to analyse the performance of our models using different metrics like accuracy, precision, recall and also visualize it through curves like ROC.

7.2. Future Work

The low-level analysis done 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 that can be extracted from audio and can be used for classification models.

7.3. Member Contribution

1. Parth Chhabra: Dataset Extraction (Spotify Data), Low Level Feature Extraction and Analysis, preprocessing, analysis-Standardization, PCA, logistic regression(SGD Classifier), decision tree and analysis, SVM models, CNN, grid search, performance analysis, Report writing.

2. Samyak Jain: Dataset Extraction (Billboard data), Low Level Feature Extraction and Analysis, preprocessing, analysis-Correlation, PCA, logistic regression(BGD), SVM models, CNN, decision tree and analysis, grid search, performance analysis, Report writing.

3. Sarthak Johari: Dataset Extraction (MSD), Low Level Feature Extraction and Analysis, Audio Data Extraction, preprocessing, analysis-outlier handling, t-SNE, Random forest, naive bayes and analysis, Adaboost model, MLP Classifier, grid search, performance analysis, Report writing.

4. Yash Mathne: Data Extraction (MSD Subset) and Analysis-dimensionality, Low Level Feature Extraction and Analysis, Audio Data Extraction, Literature

Review, Random Forest, naive bayes and analysis, Adaboost model, MLP Classifier, grid search, performance analysis, Report writing.

References

[1] Yizhao Ni et al. “Hit Song Science Once Again a Science?” In: 2011. URL: https://www.semanticscholar.org/paper/Hit-Song-Science-Once-Again-a-Science-Ni-Santos-RodrC3ADguez/c645b02ff9053c10151a09baf60be84d3e3ff12f.

[2] Francois Pachet and Pierre Roy. “Hit Song Science Is Not Yet a Science.” In: Jan. 2008, pp. 355–360. URL: https://www.researchgate.net/publication/220723429_Hit_Song_Science_Is_Not_Yet_a_Science.

[3] Eva Zangerle et al. “Hit Song Prediction: Leveraging Low- and High-Level Audio Features”. In: *ISMIR*. 2019. URL: https://archives.ismir.net/ismir2019/paper/000037.pdf.