CSE343 : Machine Learning Project Proposal
OK Computer : Employing Machine Learning to analyse the impact of different parameters in foretelling the next earworm

Ansh Arora 2019022
Jishnu Raj Parashar 2019048
Nandika Jain 2019064
Tushar Mohan 2019393

Abstract

Music, like other forms of art, is something that has historically been subjective to the listener's ears. However, in the age of TikTok and Reels, it is claimed that the songs regularly reaching the top of the charts are getting repetitive, almost following a pattern that guarantees its success. Some artists have even gone on to make rather braggadocious claims that they know well before the release of their song whether it's going to be a hit or not. Are we, as consumers of these tracks, so naive? Is there in fact a way to find out whether a song would be hit or not purely based on its characteristics?

In this project, we wish to answer these burning questions - What is it, after all, that seems to make a song more popular than others, and can we accurately predict the popularity of a track. Modern machine learning techniques might help us identify the features in a song that make it a hit, as well as accurately tell if a song, based solely on the selected characteristics, would be popular or not.
You can find the code for the implementation here - Github Link HitPredictor-ML-Project

1 Introduction

Music being one of the most subjective forms of art is theoretically not supposed to be something that should be predictable to hit levels of popularity. However, models given to us in the Machine Learning paradigm enable us to in fact actually find if this is true or not. While artists claim that they can tell if a song is going to be popular or not, we aim to find out if this is true and if yes, what exactly are the properties of a song that makes it more popular as compared to others. We aim to use techniques such as data visualization, feature extraction and selection, regression, classification and clustering to aid our study. Our main goal is to find which features affect the popularity of the song the most and create a model that is able to as accurately as possible predict how popular a song can get based on these characteristics. Our study can eventually find its uses in helping budding artists to produce music that is more likely to get popular, as well as help people shrug off their mainstream bug.

2 Literature Survey

1. Araujo et al in *Predicting Music Popularity on Streaming Platforms* make use of SVM, Random Forest and Gaussian Naïve-Bayes models to classify whether a song is successful or not based on its appearance on the Spotify Top 50 Global charts [1]. They frame the problem as a classification one, and seem to get mixed results in their study.

2. Ni et al in *Hit Song Science Once Again a Science* make use of features such as duration, loudness and tempo to binarily classify a song as hit or not hit, employing a shifting perceptron [2]. They make use of data from the EchoNest project and present a simplistic view on the topic, also giving information on how the patterns have varied over the different decades.

3. Raza et al in *Predicting a hit song with ML* make use of Logistic Regression, Decision Trees, Naïve Bayes and Random Forest methods to find out if there is a secret formula to predict which song will be on the top of charts on release [3]. This study also made use of lyrics of the songs in its dataset, performing sentiment analysis over them and distinguishing between hit and not-hit based on the data achieved.

3 Dataset

We were after the latest possible data available for our tests. Thus we set out to create our own dataset and made use of Spotify Charts data from 2017-21, which we procured from the Spotify Charts site. We then created a list of songs, each appearing once on it with its peak position on the chart, as well as its Spotify ID and other basic information such as artist name, song length and genre. The Spotify ID was then used with the Spotify Web API to garner the set of acoustic features that it provides. As a result, we were able to generate a dataset of 4247 songs, each with a total of 23 features -
{ID, Rank, Track, Artist, Streams, Week, Album_Name, Explicit, Track_Number_on_Album, Artist_Followers, Artist_Genres, Song_Length, Acousticness, Danceability, Energy, Instrumentalness, Liveness, Loudness, Speechiness, Tempo, Mode, Key, Valence.}

We then extracted the lyrics for these songs from Genius.com, using the lyrics_extractor Python library built on top of Google's Engine API. This returned the lyrics in the JSON format, from which we extracted additional 18 features using LDA for Topic Modelling.

4 Data Visualization, Preprocessing

4.1 Correlation Heatmap

A correlation matrix is a table showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables. Typically the same variables shown in the rows and columns. We use this tool to observe if there exist any obvious patterns in our data that we can observe and utilize. We used Pandas' correlation function and then mapped it to figure using the Seaborn library.

4.2 Uni-variate Graphs

We first create plots to check the count of range of values that are present for each feature in the given dataset to check for the presence of outliers as well as the general trend. We also create plots to check for the correlation between a song's Hit Score and its different macro level features that we consider. They give a basic idea on what features might be driving the Hit level of a song more than others.

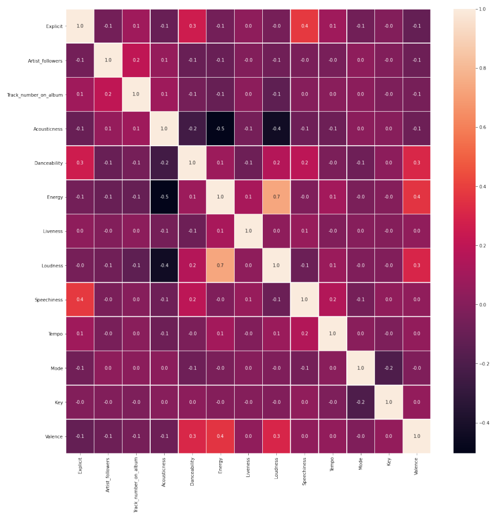


Figure 1. Correlation Heatmap for features

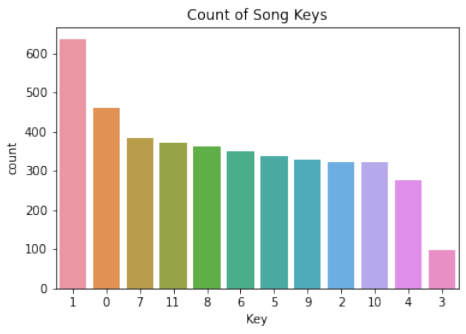
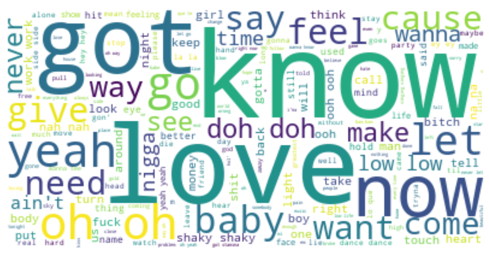


Figure 2. Count of Keys in a song

4.3 Word Cloud



A word cloud is a novelty visual representation of text data, typically used to depict the topmost frequently occuring words in a given dataset. The size of each word in the same decpicts its relative frequency in the given document. The larger the word, higher is its frequency in the dataset.

4.4 Principal Component Analysis (PCA)

PCA is a technique for reducing the number of dimensions in a dataset whilst retaining most information.

PCA involves standardization, computation of covariance matrix and eigenvectors to identify principal components. Given the fact that we have a comparitively high-dimensional dataset, it helps us in visualizing the data in a meaningful manner. We used Scikit-Learn's implementation to reproduce the same.

4.5 Data Standardization

It is a scaling technique used to standardize the data and make it more workable with. The values are centred around the mean over a unit standard deviation. Formula used -

$$z_i = \frac{x_i - x_{mean}}{\sigma}$$

5 Methodologies

To achieve our aim of finding the features that most affect the popularity, we are trying multiple varieties of models. As of now, we have used Classification Algorithms - Logistic Regression and Support Vector Machines and have made use of Regression Algorithms - Logistic Regression, Regularized Regression and Support Vector Regression. We have made use of Holdout as well as K-Fold Cross Validation to get the most accurate scores to the data possible.

5.1 Classification

We created thresholds on what can be identified as a hit song vs a non-hit. We tried multiple threshold values ranging from 10 to 100, and got a true representation around a value of 50 for the models we tried out.

5.1.1 Logistic Regression

Logistic Regression model makes use of sigmoid function to squeeze the calculated values from Linear Regression into a range of 0 to 1. Since we are doing a binary classification hence, we set a threshold of 0.5, which implies that any value above 0.5 is classified as 1 and the rest as 0.

5.1.2 SVC

Support Vector Machine is a statistical algorithm which clusters the data points in order to learn a hyperplane which is efficiently able to divide the data points into 2 halves for binary classification.

5.1.3 Random Forests

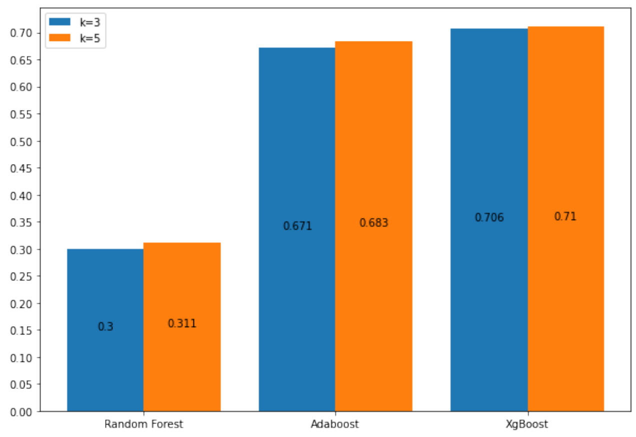


Figure 3. Mean Cross-validation accuracies of classifiers

We applied Random Forest on the extracted features with a tree's max depth set to 10. Via cross-validation with different values of k we were able to reach a maximum mean multi-class accuracy of 31.2% which was indeed quite low. So we tried boosting via 2 different techniques.

5.1.4 Boosting

We utilised Adaboost and Xgboost as the boosting techniques. Both of which performed fairly well on our dataset with Xgboost giving a maximum mean accuracy on cross-validation as 71% while Adaboost gave a max mean accuracy of 68.3%.

5.1.5 ANN Classifiers

We used Keras' dense Sequential Neural Network implementation to construct Classifier ANNs. We tried out multiple hyperparameter values and activation functions including ReLU, Leaky ReLU and tanh. We received the best values of around 73.5% on using ReLUs over a k=3 CV model, which was much better than the baseline performance of Logistic Regression implemented using the same models which gave accuracy values of around 67%, much in line with the predictions of baseline Logistic Regression.

5.2 Regression

To process the Rank labels, we created a Hit Score value measure. This assigns a value of 1 to songs that have hit the peak of the chart and a minimum value of 0.3 to any song that has entered the chart. Songs that have not charted are assigned a value of 0, certifying a non-hit.

5.2.1 Linear Regression

We applied Baseline Linear Regression on 5-fold and 10-fold division of the training set and calculated the average RMSE observed on the predicted Hit Score and calculated Hit Score.

5.2.2 Regularized Regression

We used GridSearchCV to find the best possible lambdas on each of the 10 k-folds in both L1 (Lasso) and L2 (Ridge) regressions, and then calculated the RMSE values of the predicted Hit Score with our test Hit Scores.

5.2.3 SVR

We made use of Support Vector Regression on a 5-fold division, fitting hyperplane decision boundaries through the training data using SKLearn's implementation to get Hit Score predictions, and then calcuated the RMSEs observed.

5.3 Clustering

We implemented K-Means clustering algorithm. It is the most common unsupervised Machine Learning algorithm to identify clusters for a given dataset in its feature space. It uses Euclidean-distance as a metric to assign any datapoint to the nearest cluster. The algorithm identifies 'k' clusters and chooses centroid from the given datapoints itself. Eventually minimising the required number of clusters.

5.4 Natural Language Processing

We were able to extract lyrics for a total of 4247 songs. In order to extract some information from the lyrics, we did sentiment analysis treating the libraries available for the same like a magic box. We used two different libraries to do the same, namely - VADER and TextBlob. The former is a lexicon-level sentiment analysis tool that provided scores for 4 different quantities namely - negative, positive, neutral and complex. The latter uses patterns to calculate sentiment of a given sentence and provides 2 scores namely - polarity and subjectivity.

5.4.1 Latent Dirichilet Allocation (LDA)

We applied LDA to extract topics from the 4247 songs' lyrics we were able to extract. Latent Dirichilet Allocation is a generative statistical model which recursively traverses through a set of documents to classify them into a pre-set number of unobserved groups that explains why a set of data is similar. We used Scikit-Learn's LDA implementation to extract topics from lyrics of 4247 different songs. This gives us a sense of broad topics which can efficiently classify our current dataset.

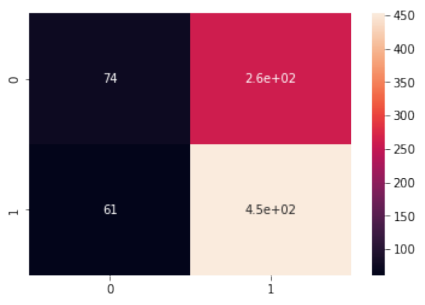


Figure 4. Caption

6 Results and Analysis

From the various models that we applied on our curated dataset, we can say that the features in our dataset have negligible correlation with each other hence our assumption of treating them as independent and identically distributed stands strong.

6.1 Classification

For binary classification, we grouped all the songs into either a hit or not hit by setting a threshold value on the chart rank. SVC performed better in binary classification with an accuracy of 71.85% with a polynomial kernel and 69.29% with a linear kernel.
Logistic regression performed fairly with an average accuracy of 67.644% with the SGD classifier peaking at 71% with precision of 84%.
Bagging techniques like Random Forests performed poorly while Boosting techniques like Xgboost gave a maximum mean accuracy on cross-validation as 71% while Adaboost gave a max mean accuracy of 68.3%.
We finally tried ANNs over multiple epoch values and different activation function, where ReLU performed best for our usecase with 73.5% accuracy and tanh giving 71.8% accuracy.

6.2 Regression

Linear Regression performed surprisingly well with an average RMSE of 0.039 in K-Fold validation. Support Vector Regression didn't perform as well as we hoped

it would with an average RMSE of 0.044 on a 5 fold K-Fold Cross Validation. However, SVR showed a lower error on the training set, which signifies a High Variance model.
On performing the t-test and f-test on Linear Regression model with 20% test set, we get a p-value of 0.66 which is greater than 0.05 hence, we can say that the mean of residuals of the holdout set is statistically similar to 0.

6.3 Clustering

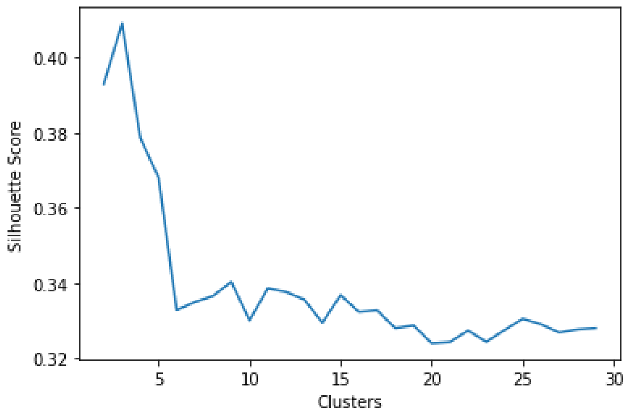


Figure 5. Silhouette Scores for K-Means

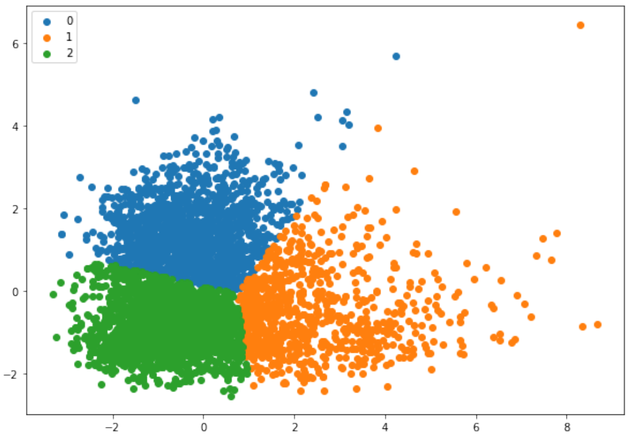


Figure 6. Cluster for k=3 for K-Means

We applied k-means clustering after reducing the dimensions of our dataset via Principal Componenet Analysis (PCA) to only 2 for visulaisation. After analysing the Silhouette score for clusters 2 to 30, we came to the conclusion that the optimal number of clusters for our dataset is 3 which can be seen in the Figure 6.

6.4 Natural Language Processing

We tested LDA for topics 3-100 and for each run we calculated the coherence scores, i.e. c_v scores, which gives us a sense of how well do the topics fit the given set of documents. The highest score subsequently obtained was *0.475*, corresponding to *80* topics.

Next we applied sentiment analysis on our lyrics dataset. From Figure 8, it is clear that whether the song is a hit or not, it is most likely neutral, and not skewed towards either the positive or negative sentiment. This is confirmed by the polarity vs songs histogram. This implies, that if a given song makes it to the top charts, it is most likely to be neutral than to possess a specific sentiment.

7 Conclusion

From the results and analysis, we did find a correlation between the features that we extracted and the songs' corresponding popularity, though with a slightly less accuracy from what we expected. With different models and sets of features, we might be able to come up with the desired results.

7.1 Learnings

Having experimented with multiple forms of data and features, at this early stage we can definitely say that there does seem to exist some correlation between certain features and the song's likeliness to be a Hit or have a higher Hit Score. The metric scores obtained on certain tests, especially the ones on the gradient paradigm seem to show that this in fact might actually be a better measure for popularity as opposed to the classification models created in the works done before.

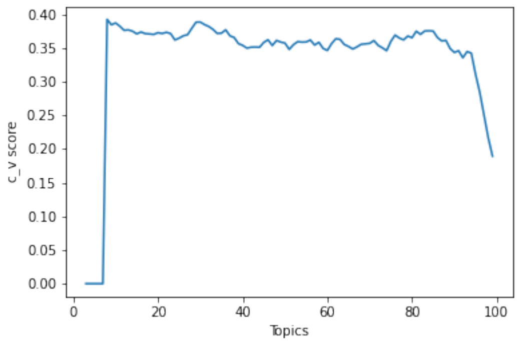


Figure 7. Average Coherence Scores for a window=10

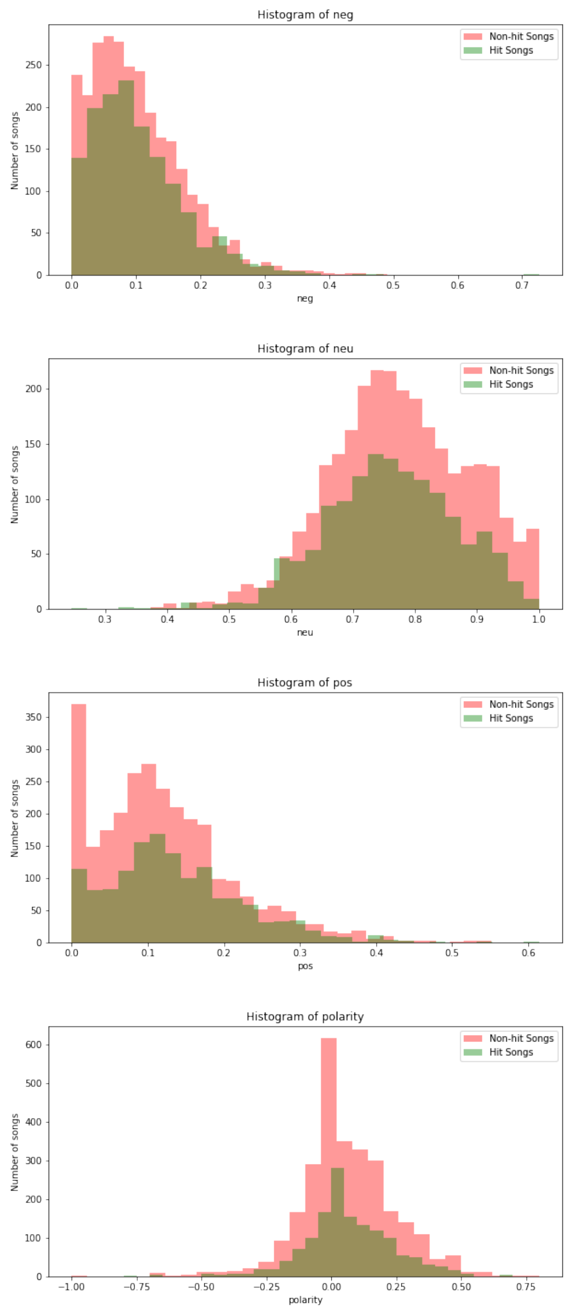


Figure 8. Sentiment distribution across hit and not-hit songs

7.2 Future Work

Following the work already done, the accuracy from basic machine learning techniques is not good enough, so the subsequent work will be focused on improving our multi-class classification accuracy via advanced machine learning and deep learning classifiers. We also wish to inculcate our models into a UI-based environment with human-in-the-loop approach to keep improving the models in the backend. This way we will be able to achieve fairly better accuracy as the dataset will no longer be restricted and the model will keep on learning. The idea of topic modelling can be extended towards a linear combination approach to classify a song as a hit or a non-hit.

7.3 Contributions

We have all aided each other in different parts of the project, and this has been mainly a collective effort.

1. **Ansh** - Lyrics Curation, Exploratory Data Analysis, Support Vector Regression, Regularized Regression, Artificial Neural Network, Clustering, Report + Powerpoint

2. **Jishnu** - Data Curation, Linear/L1/L2 Regression, Logistic Regression, Support Vector Machine, Random Forest, Report + Powerpoint

3. **Nandika** - Data Curation, Linear/L1/L2 Regression, Logistic Regression, Support Vector Machine, Random Forest, Report + Powerpoint

4. **Tushar** - Lyrics Curation, Exploratory Data Analysis, Dimensionality Reduction, Clustering, Natural Language Processing, Artificial Neural Network, Report + Powerpoint

References

[1] Araujo et al *Predicting Music Popularity on Streaming Platforms*.

[2] Ni et al *Hit Song Science Once Again a Science*.

[3] Raza et al *Predicting a hit song with ML.*

[4] *Unlocking the True Power of Support Vector Regression*.

[5] *Canonical correlation analysis*.

[6] M. Blei et al. *Latent Dirichilet Allocation*