Worldwide trends in Spotify

Paulina Weiss S1657612 **Kyriakos Kyriakou S2281922** **David Dagg S2229609**

Abstract

Spotify has become one of the largest music-streaming platforms in the world. With it, the relevance of predicting what makes a song popular on the platform has increased significantly. In this paper, we investigate the problem of predicting the popularity of music. In particular, we consider the words in the track name as a cue that can be used for predicting whether a song will appear in Spotify's Top 100 ranking. We explore two popular feature extraction and dimensionality reduction techniques for text data on different classifiers, including Naive Bayes (NB), Logistic Regressions (LR), Decision Trees (DT), Support Vector Machine (SVM) and Artificial Neural Networks (ANN). When tested on previously held-out data, the ANN classifier performed best obtaining accuracy and AUC above 78%. We explore changes to predictive accuracy when including encoded artist names and conclude by discussing the feasibility of predictions with sparse data.

1 Introduction

People's attitude toward consuming music has drastically changed. As of 2022, music streaming platforms dominate the global music market with a 65% share of the overall revenue, according to the International Federation of the Phonographic Industry (IFPI) annual report[7]. Spotify is by far the biggest streaming platform in the world with a global market share of 31% and reportedly 182 million subscribers in Q1 2022[1]. Consequently, it is critical for artists to perform well on the platform to gain popularity and maximize commercial return.

Numerous papers have used machine learning methods to map auditory features such as danceability, energy, speechiness, acousticness, loudness or, tempo of a song to popularity measures. For example, Gulmatico et al. [6] classify song popularity as measured by the number of Spotify streams finding predictive accuracy of 95%. Moreover, most papers formulate song popularity prediction as a binary classification problem. Araujo [3] collected daily audio features of the Top 50 and Viral 50 public playlists from the Spotify Web API and trained different classifiers on audio features and predicted whether or not a song will appear in Spotify's Top 50 ranking with above 80% accuracy. Similarly, Georgiva et al. [5] predicted the Billboard success of a song with approximately 75% accuracy based on audio features.

Findings of high predictive accuracy of music popularity based on auditory features have led to a controversial claim that there exists a "Hit Song Science", i.e. a recipe of features that would make a song preferred by a majority of people [4]. Strikingly, the most salient feature of a song on Spotify is its title, and may lend itself as a new avenue to drive engagement and streaming. The relevance of headlines of newspaper articles, videos and tweets in attracting and engaging an audience has been extensively studied, especially in online media ([10],[13],[14]). However, song popularity prediction based on text data remains to be investigated. The objective of this report is to explore the feasibility of forecasting a song's popularity solely based on its title.

The rest of this paper is organized as follows. After a preliminary EDA, we introduce two different dimensionality reduction techniques for text data. We then motivate the use of five popular text classification models and apply them to our feature encodings. Lastly, we evaluate their popularity prediction performance and discuss how a model can be extended to include artist names in the feature space.

2 Data Preparation

Our data contains a daily ranking of the 200 most listened songs in 53 countries from 1st January 2017 to 9th January 2018 by Spotify users. The data is freely available on Kaggle, an open source platform for data scientists, and was collected from Spotify's regional chart data [2]. It is set up in a way that each entry represents one song on a given day in a given country. Consequently, there can be multiple entries for the same song if it appears more than once in the ranking. The data was cleaned and entries with empty track names or artists' names were dropped (657 entries, <0.001%).

In this report, we consider the regional rankings of the US only. This subset of the data contains 74,184 entries, which comprises 1,624 unique tracks and 487 artists. The popularity of a track is modeled as a binary variable $Y \in 0, 1$, where entries labeled 1 indicate that a track is in the Top 100 ranking, and 0 otherwise. We will refer to the category membership as *"Popular"* and *"Mildly Popular"* respectively.

It is a common approach to transform text data to a vector representation, where $x^T \in R^d$ represents a collection of T text features [11]. We transform track names into numerical feature vectors where dimension T equals the total number of unique words across track names that can then be fed into classification models. Each word is represented with the Bag of Words (BoW) method that assigns a numeric value to a word corresponding the frequency of appearing in the track name. For this purpose, the track name variable was first transformed into lowercase strings, tokenized and inflected forms of English words were grouped together to a single base form (cf. lemmatization). While it is custom to remove stop words from text data, we have chosen to keep them as they may provide meaningful information for track popularity classification.

The class distribution of the data set is not perfectly balanced, higher popularity rank labels are represented by a higher number of entries. Stratified sampling from the data with an 80/20 ratio yields a training set and hold-out test set respectively with approximately equal class distribution. Both, the BoW scoring as well as the exploratory data analysis, were done solely on the training set.

3 Exploratory data analysis

The track name variable exhibits high cardinality with 1,559 unique values. Cardinality increases with lower popularity rankings from 15 unique track names in the 1st rank to 211 in the 200th rank (cf.Appendix A). Overall, *Popular* track names occur on average 35 times in the Top 100 of the ranking, whereas *Mildly Popular* tracks names 20 times in the Top 100 to 200, suggesting a bias towards certain titles over others. Figure 1 shows the frequency distribution of the Top 3 most frequently occurring track names per class. *Popular* track names are highly frequent in the first 50 rank positions and frequency drops drastically thereafter. Contrastingly, the frequency of the top three songs of *Mildly Popular* track names are distributed much more evenly across rank positions.

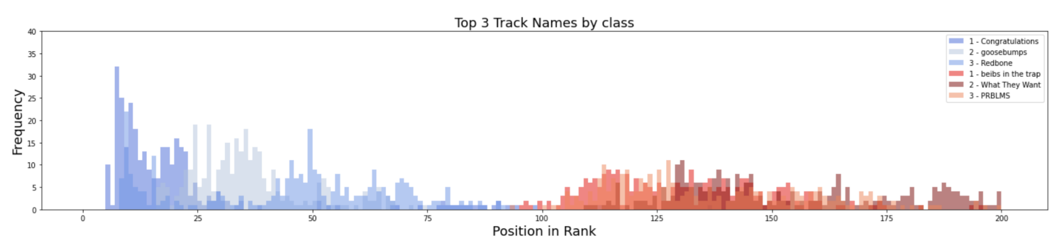
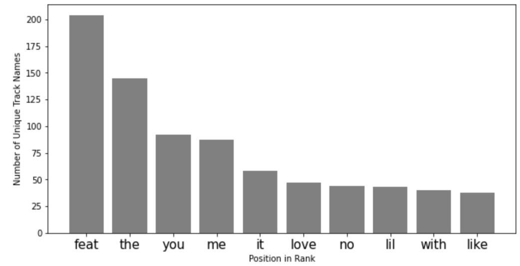


Figure 1: Frequency distribution of occurrences of the top three most frequent *Popular* and *Mildly Popular* track names across popularity rank. Top 3 *Popular* include: "Congratulations" (N = 307), "goosebumps" (306) and "Redbone" (300) (blue). Top 3 *Mildly Popular* include: "beibs in the trap" (285), "What They Want" (279) and "PRBLMS" (273) (red).

The BoW vector representation of track names has high dimensionality, with 1,837 unique words. Figure 2a shows the most frequent words per class and their frequency in the corpus. Interestingly, the same words appear to be most frequent in both classes. Considering Figure 2b, the five most frequent words appear in more than 50 unique track names. However, 66% of the words in the long tail end of the distribution (N=1218) appear only in one unique track name. This suggests that the words themselves exhibit high cardinality and are specific to the track names they appear in.

	Class	
Rank	*Popular*	*Mildly Popular*
1st	**feat** (4785)	**feat** (3600)
2nd	**the** (1720)	**the** (2561)
3rd	**you** (1576)	**me** (1496)
4th	**me** (1283)	**you** (1477)
5th	**in** (1191)	**no** (1129)

(a) Five most common words (count) in the track name per class.



(b) Most frequent words of the corpus mapped to the number of unique track names in which they appear in.

Figure 2

Lastly, cardinality for artist names increases with lower popularity rankings Appendix A. Figure 3 shows that the frequency distribution of the three most frequently occurring artist names is variable. While the most frequently appearing artist "Drake" shows consistently across the rank, the other two artists´distributions seem to be biased to top rank positions. Compared to track names, the distributions are more balanced across classes.

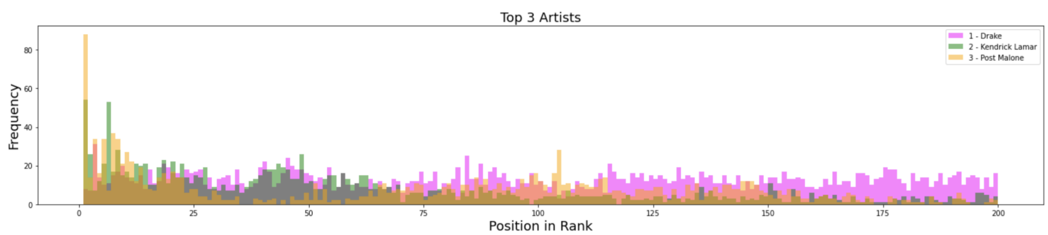


Figure 3: Frequency distribution of the three most frequently occurring artists: Drake (N=2583), Kendrick Lamar (1516) and Post Malone (1483).

4 Dimensionality Reduction

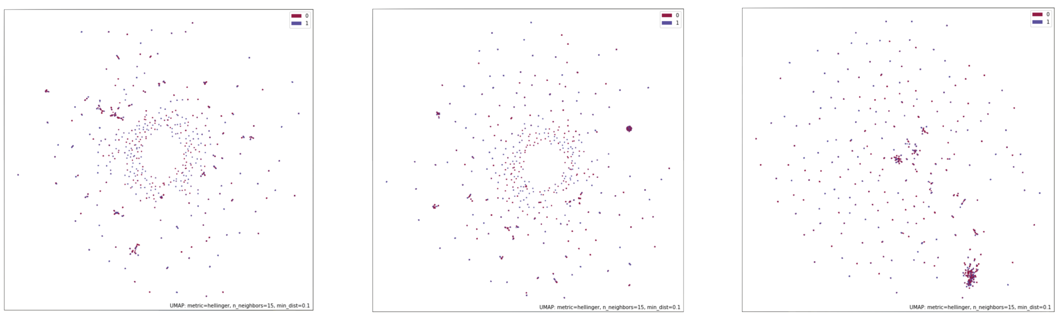
Dimensionality reduction plays a crucial role in text classification and influences model efficiency and effectiveness. It is a technique to reduce the size of a feature vector such that it keeps as much of the variation in the dataset as possible. A challenge of our text categorization tasks is the high cardinality of the feature space (1,837 unique words). Many words are potentially noisy, non-informative or redundant with respect to popularity prediction while increasing the complexity of our models and reducing predictive accuracy due to overfitting on the training data (cf. *curse of dimensionality*).

We choose both a feature selection and feature extraction technique to perform this task. Feature selection is the process of selecting the most meaningful feature subset from the original feature vector. Feature extraction reduces the dimensionality of a feature vector by creating a new smaller set of new features with maximal discriminative capability. Features are encoded as TF-IDF feature vectors - a variant of the Bag of Words (BoW) method - which assigns a value to each word present in a track name corresponding to the number of times a word appears in the track name (*tf*, term frequency) and offset by the number of track names in the data set that contain the word (*idf*, inverse document frequency). Both methods will be compared against a baseline encoding that holds the full BoW feature vector in a sparse feature matrix of size 59347 x 1837.

Filter method with ANOVA, is a feature selection technique that assigns a score to every feature and selects a subset of the highest scoring features. The feature scoring metric is the ANOVA F-value which measures how well a feature discriminates between the two classes. We apply the False Positive Rate test to all scores and select only those features with significantly small total amounts of false detections (p-value <.05, Appendix B). Keeping only the most discriminative features, this dimensionality reduction technique results in a sparse feature matrix of size 59347 x 913. Importantly, filter methods do not remove potential correlation between features (cf. multicollinearity) which may reduce the effectiveness of a classifier.

(Truncated) Singular value decomposition (SVD) is a feature extraction technique that deals with multicollinearity by combining highly correlated variables into a smaller set of uncorrelated variables, thus, reducing the number of features while preserving the relative similarity between track names. Similarly to principal component analysis (PCA), SVD is a linear dimensionality reduction method that attempts to find linear combinations of features in the original high-dimensional feature vector. Using SVD to reduce the dimensionality of TF-IDF feature vectors is also referred to as latent semantic analysis (LSA) in the natural language processing literature. We use truncated SVD - a particular version that can deal with sparse data - to remove all new features below a variance threshold of 90% Appendix B, resulting in a sparse feature matrix of size 59347 x 400.

In Figure 4, we visualise the impact of feature encodings and dimensionality reduction on class clustering in a two-dimensional scatter plot utilising the Uniform Manifold Approximation and Projection (UMAP) technique [9]. Strikingly, the two classes do not clearly cluster in space and no added clustering effect of ANOVA and SVD is visually detectable. Instead, some data points are scattered and others form a herd that includes samples from both classes. From this, we may infer that the ability of text classifiers to achieve a clear separation of the two classes is limited.



(a) BOW (Baseline) (b) TF-IDF x ANOVA (c) TF-IDF x SVD

Figure 4

5 Classifiers and methods

To predict song popularity, we evaluate five different machine-learning classification algorithms: Naive Bayes (NB), Logistic Regression (LR), Decision Trees (DT), Artificial Neural Networks (ANN), and Support Vector Machines (SVM).

Naive Bayes (NB) is a probabilistic classification method based on applying Bayes' theorem widely used in the text classification literature [8]. In the context of text classification, the classifier is based on the (naive) independence assumption that the probability of each word appearing in a document is independent of the occurrence of other words in the same document. Particularly, we apply multinomial NB to classify with our numeric feature vectors of track name. The NB model's biggest strength is that it is performing well despite being simple. Though, model simplicity is often beaten by more complex models.
Decision trees (DT) are a popular classification and regression algorithms that when applied to text classification problems, select informative words according to an information gain criterion. Thanks to their hierarchical structure they can learn complex patterns and generalise well to unseen data. However, DTs tend to be rather unstable classifiers compared to other ML algorithms. A small mistake in a split can lead to a large change in the structure of the DT, leading to poor generalisation.
Logistic regression (LR) is a statistical method capable of carrying out binary classification tasks and can be generalised to multi-class classifications [12]. In our analysis, we are using logistic regression for a binary text classification problem, classifying track names based on their words to a set of two popularity groups. Logistic regression is proven to be a good algorithm for classification, but is not flexible enough to learn multiple and nonlinear decision boundaries.
Artificial Neural Networks (ANN) are powerful non-linear deep-learning algorithms for clas-

sification problems, inspired by the biological neural networks. An ANN consists linear weight vectors with non-linear activation functions grouped in layers. For this task, we use a multi-layer perceptron (MLP) trained on the Adam optimizer that consists of three layers (with 20, 30, 10 neurons respectively) and two output neurons representing the two popularity classes. Neural Networks might perform very well in classification problems but are considered as very expensive machine learning methods (i.e. high memory/time complexity).
Support Vector Machines (SVM) are linear models for classification tasks that separate classes with a hyperplane (based on geometrical properties). The model tends to generalise better with increasing separation margin. A Linear kernel function was used to separate features in the two popularity classes. SVMs are considered very effective classifiers, but they are also very computationally expensive when it comes to large sparse data.

To effectively identify optimal model parameters, all classifiers were tuned on a set of model-specific hyper-parameters using a grid search method and k-fold cross-validation (k=3). In k-fold cross-validation, the training data is randomly split into k subsets of equal size and the classifier is trained k-times on k 1 subsets, holding out a single subset as validation data for testing the classifier instantiation. Grid search allowed us to train classifiers for each combination of hyper-parameters and chose those with the highest accuracy score resulting in fifteen feature vector and classifier pairs Appendix C. In a second iteration of model training and tuning, we extended the feature matrix to include BoW encodings of artist names with a sparse feature matrix of size 59347 x 2311.

6 Results

Each of the fifteen feature vector and classifier pairs (3 x 5) was evaluated using the holdout test dataset (N=14,837) resulting in accuracy scores as shown in Figure 5. In the experiment, the Artificial Neural Network (ANN), Logistic Regression (LR) and Support Vector Machine (SVM) trained on the baseline BoW feature encodings achieved the best overall performance (accuracy of 78%, 77,8% and 77,4% respectively). ANOVA achieved the best results when the NN classifier was used, while SVM was the best for SVD feature encodings. Surprisingly, BoW obtained the highest results across all models with the exception of Decision Trees (DT), closely followed by ANOVA feature encodings which achieved results that differ from BoW only by <1p.p across all models.

Table 1 presents the values achieved across different evaluation metrics for the five models with the highest accuracy scores. The F1 score, AUC ROC, AUC PR, precision, and recall for all model instantiations are shown in Appendix B. Balanced Accuracy was not needed since the dataset is almost perfectly balanced. The best model when comparing accuracy, F1 score, AUC ROC, and AUC PR curve is the NN using BoW encoding.

The above metrics were used to identify the best candidate feature encoding and classifier pair trained on data that also included artist names. The best model obtained from the extended feature space was the LR with BoW encoding achieving a 77.75% accuracy score. Importantly, model performance does not exceed that of the NN trained only on song titles (78.03%) but is very close. Equally, the True Positive and Negate Rate is slightly lower for the extended feature space, as evidenced by the confusion matrices of the two model instantiations Figure 6a. All results are shown in the second table in Appendix D.

ALGORITHM	DIMENSIONALITY REDUCTION	ACCURACY	F1	AUC ROC	AUC PR
Neural Network	BOW	0.7803	0.7877	0.7803	0.869214
Logistic Regression	BOW	0.7775	0.7863	0.7775	0.863598
SVM	BOW	0.7745	0.7823	0.7745	0.862492
Neural Network	ANOVA	0.7681	0.7865	0.7681	0.864319
Logistic Regression	ANOVA	0.7628	0.7591	0.7628	0.859912

Table 1: Results of top 5 models between Naive Bayes, Logistic Regression, Decision Tree, Neural Network, and Support Vector Machine with BoW, ANOVA, and SVD feature vectors.

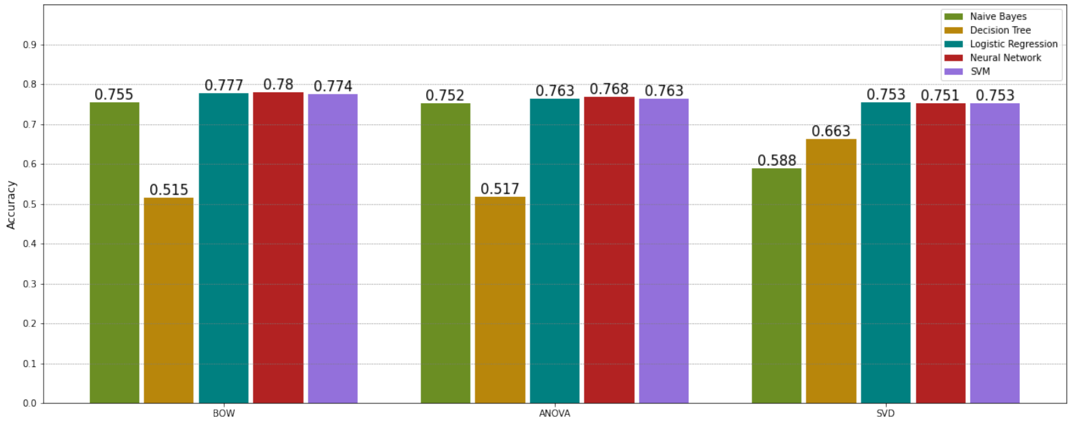
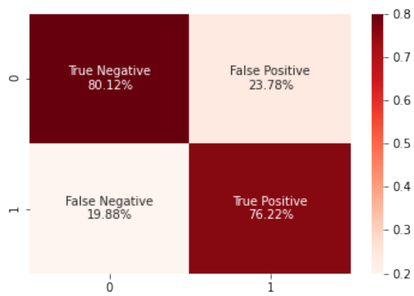
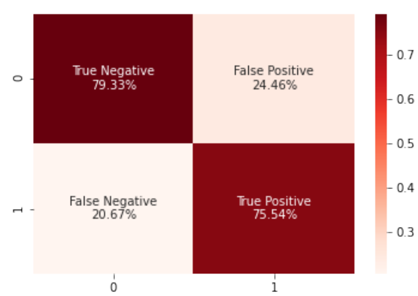


Figure 5: Accuracy for each model for each dimensionality reduction method.



(a) NN x BoW trained on track names.



(b) LR x BoW trained on track names and artist names.

Figure 6: Confusion Matrices for best models with each feature types.

7 Discussion

New avenues for predicting song popularity on music streaming platforms are highly anticipated by the industry and academic community. In this work, a novel approach considering only track and artist names for popularity prediction was tested and did not improve on predictive accuracy of previous work on auditory features.

Most strikingly, TF-IDF feature encoding and ANOVA or SVD dimensionality reduction techniques did not improve predictive accuracy against the baseline BoW encoding. Names tend to be semantically distinct; our EDA has shown that this generalises to track names, their individual words and artist names as they exhibit high cardinality. Thus, TF-IDF transformation does not significantly change the BoW baseline encoding as most words did not occur in multiple different unique track names. Similarly, SVD has limited additive effect as data sparsity reduces its capability to extract meaningful structure from the feature space.

Still, our experiments suggest that predictions based on text data are feasible, and - even if carried out with smaller training data sets - exhibit fairly high levels of accuracy. Further feature engineering of track names such as semantic matching or expanding the feature space to other text data such as lyrics may unveil structural relatedness in the data, mitigate the problem of high cardinality and increase predictive power. Lastly, re-conceptualising popularity classes by extending the data set to tracks outside of the Top 200 popularity ranking may produce a more meaningful decision boundary.

References

[1] *Spotify-Technology-S.A. Announces Financial Results for First Quarter 2022*. https://investors.spotify.com/financials/press-release-details/2022/Spotify-Technology-S.A.-Announces-Financial-Results-for-First-Quarter-2022/default.aspx, 2022.

[2] *Spotify's Worldwide Daily Song Ranking*. https://www.kaggle.com/datasets/edumucelli/spotifys-worldwide-daily-song-ranking, 2022.

[3] C. V. S. Araujo. *Model for Predicting Music Popularity on Spotify*. Recall, 50, 173-90., 2020.

[4] Logan B. Dhanaraj, R. *Automatic Prediction of Hit Songs.* In ISMIR (Vol. 11, No. 15, pp. 488-491)., 2005.

[5] Suta M. Burton N. Georgieva, E. *Hitpredict: Predicting hit songs using spotify data.* 2018.

[6] Susa J. A. B. Malbog M. A. F. Acoba A. Nipas M. D. Mindoro J. N. Gulmatico, J. S. *SpotiPred: A Machine Learning Approach Prediction of Spotify Music Popularity by Audio Features.* In 2022 Second International Conference on Power, Control and Computing Technologies (ICPC2T) (pp. 1-5). IEEE., 2022.

[7] IFPI. *IFPI Global Music Report*$_2022 State of the Industry. https://globalmusicreport.ifpi.org/$, 2022.

[8] Han K. S. Rim H. C. Myaeng S. H. Kim, S. B. *Some effective techniques for naive bayes text classification.* IEEE transactions on knowledge and data engineering, 18(11), 1457-1466., 2006.

[9] Healy J McInnes, L. *Uniform Manifold Approximation and Projection for Dimension Reduction*. ArXiv e-prints 1802.03426, 2018.

[10] Pourmodheji H. An A. Edall G. Omidvar, A. *A novel approach to determining the quality of news headlines.* In Natural Language Processing in Artificial Intelligence—NLPinAI 2020 (pp. 227-245). Springer, Cham., 2021.

[11] Wong A. Yang C. S. Salton, G. *A vector space model for automatic indexing.* Communications of the ACM, 18(11), 613-620., 1975.

[12] Abdulhamit Subasi. *Practical Machine Learning for Data Analysis Using Python.* Academic Press, 978-0-12-821379-7., 2020.

[13] Vishwakarma D. K. Varshney, D. *A unified approach for detection of Clickbait videos on YouTube using cognitive evidences.* Applied Intelligence, 51(7), 4214-4235., 2021.

[14] Y. Zhou. *Clickbait detection in tweets using self-attentive network.* arXiv preprint arXiv:1710.05364., 2017.

APPENDIX

A EDA

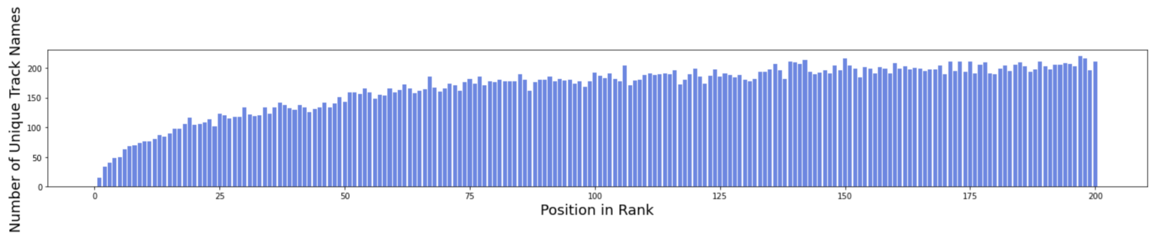


Figure 7: Number of unique track names by position in popularity rank.

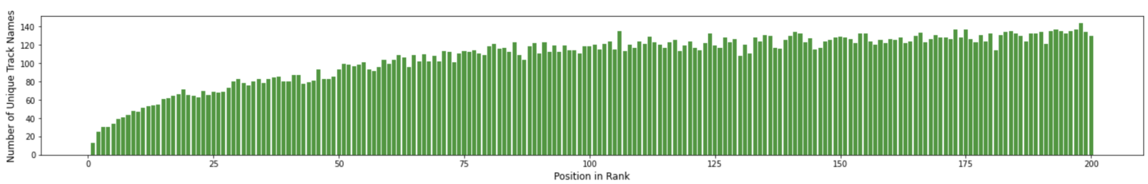


Figure 8: Number of unique artist names with their ranking position.

B Dimensionality Reduction

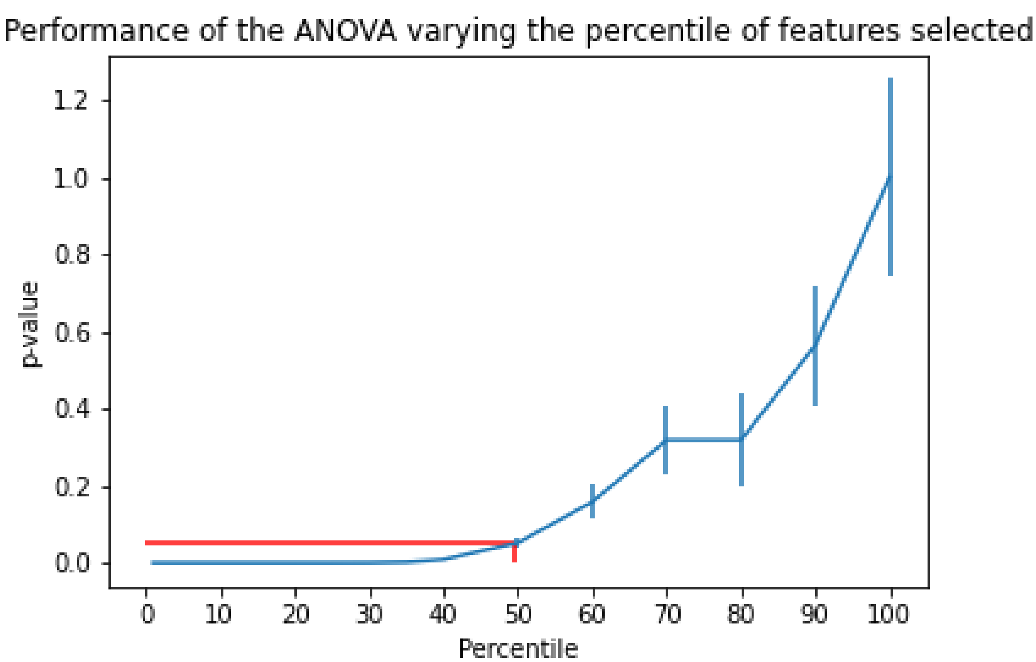


Figure 9: Performance of the ANOVA varying the percentile of features selected.

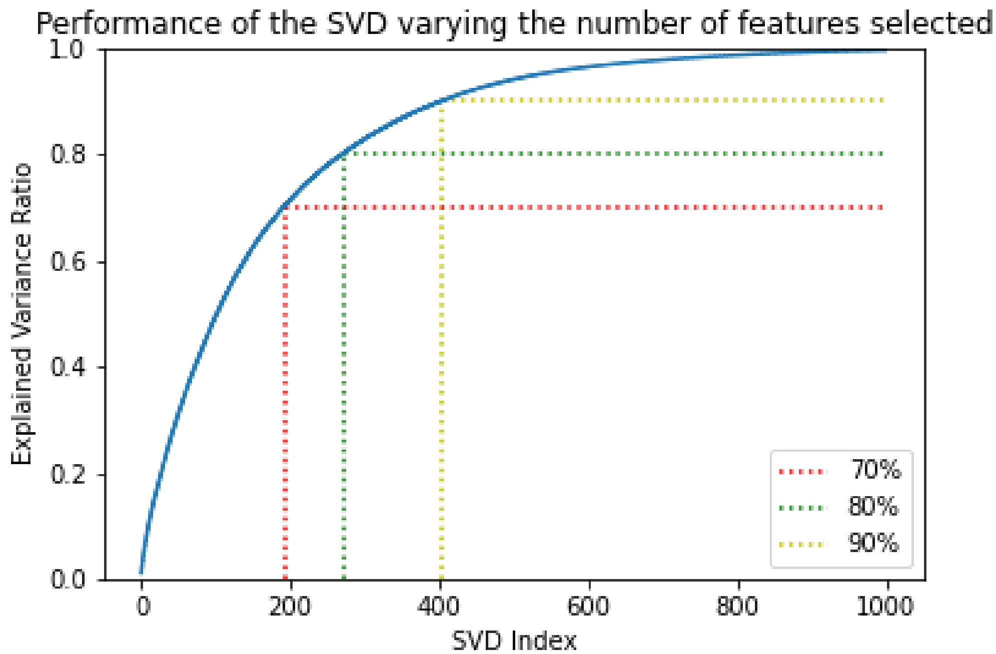


Figure 10: Performance of the SVD varying the number of features selected.

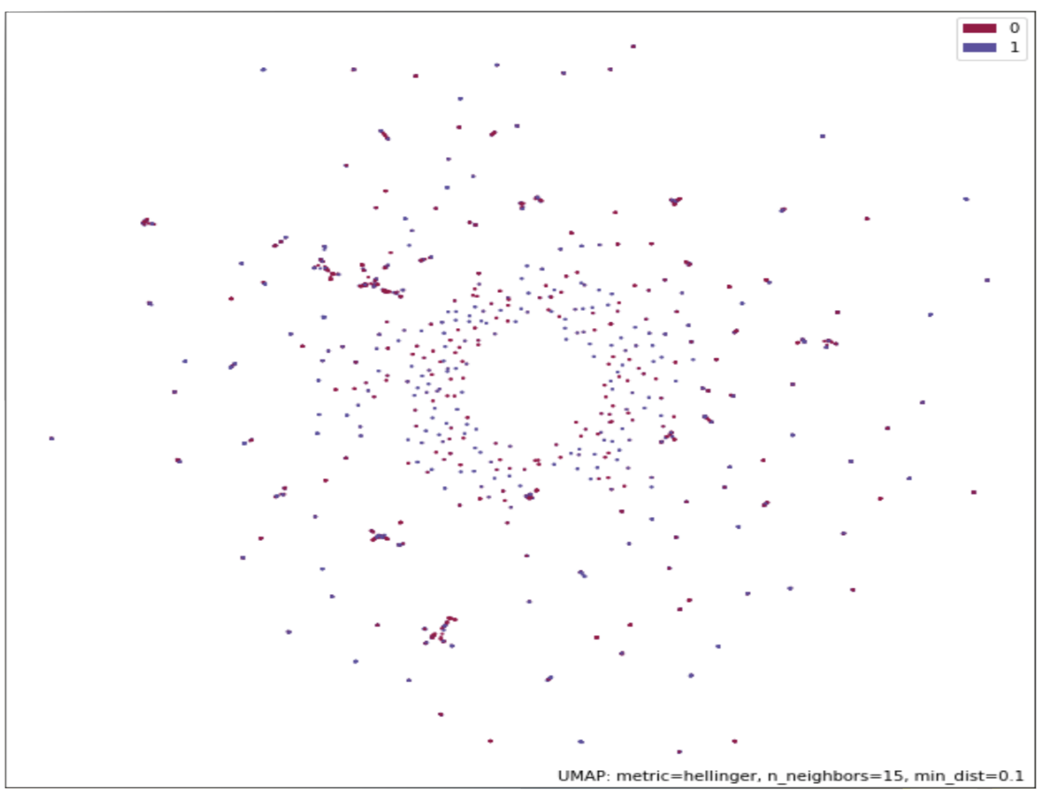


Figure 11: UMAP displaying embedding distribution with BoW and ANOVA.

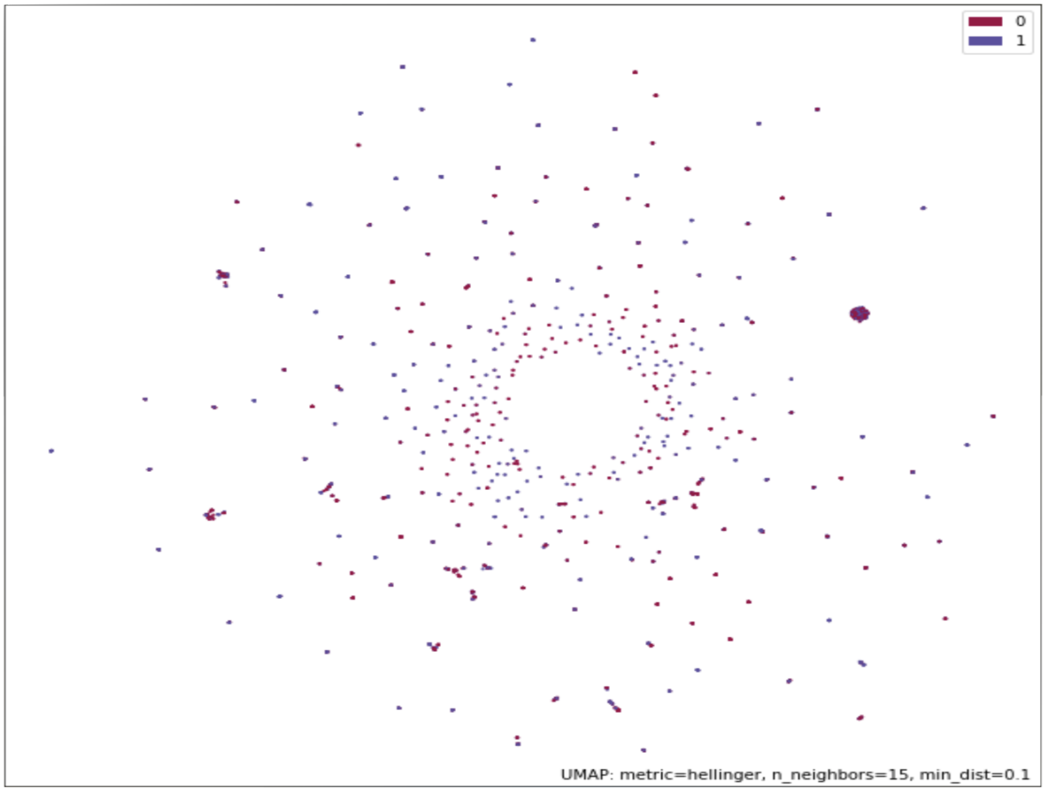


Figure 12: UMAP displaying embedding distribution with TFIDF and ANOVA.

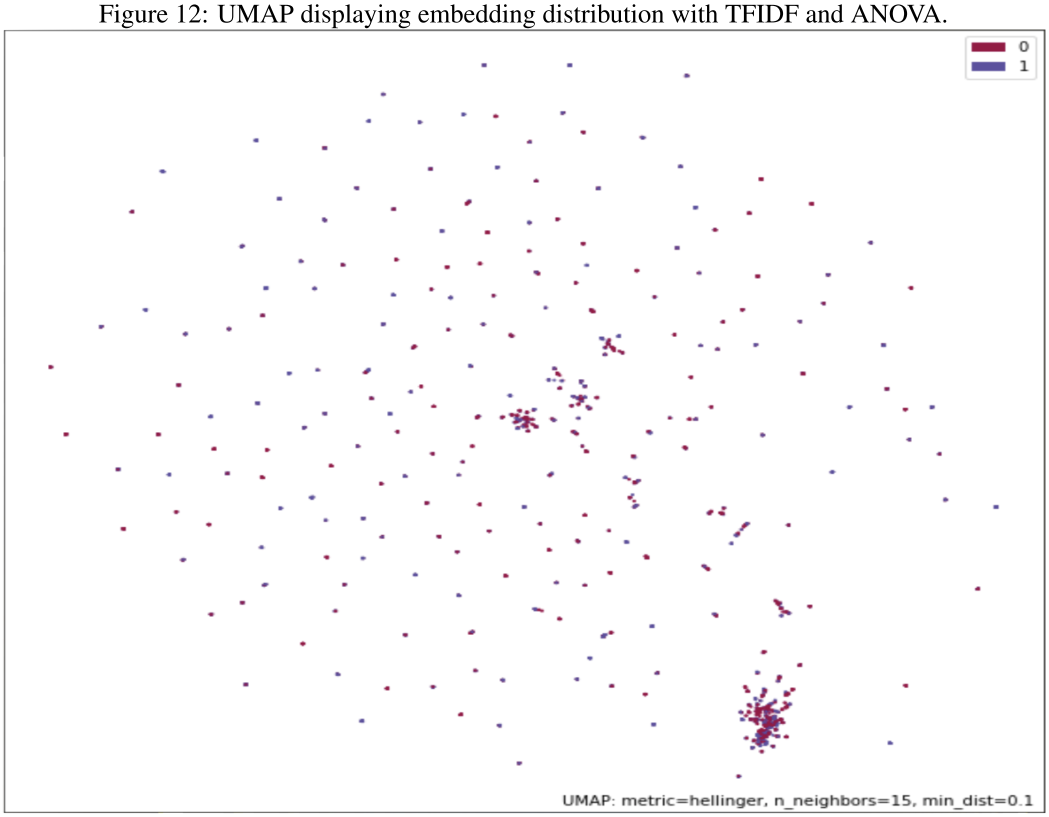


Figure 13: UMAP displaying embedding distribution with TFIDF and SVD.

C Grid Search Parameters

ALGORITHM	DIMENSIONALITY REDUCTION	POTENTIAL HYPERPARAMETERS	BEST HYPERPARAMETERS
Naive Bayes	ANOVA	('alpha': [0.25,0.5,0.75,1,10,20]))	(alpha=0.25)
Naive Bayes	SVD	('alpha': [0.25,0.5,0.75,1,10,20]))	(alpha=0.25)
Decision Tree	ANOVA	('criterion':['entropy'], 'max depth': [5,7], 'max features': ['sqrt','log2'], 'min samples leaf': [15,20]))	(criterion='entropy',max depth=7,max features='sqrt', min samples leaf=15)
Decision Tree	SVD	('criterion':['entropy'], 'max depth': [5,7], 'max features': ['sqrt','log2'], 'min samples leaf': [15,20]))	(criterion='entropy',max depth=7,max features='sqrt', min samples leaf=15)
Logistic Regression	ANOVA	('solver': ['lbfgs'], 'penalty': ['l2'], 'C': [1,10,100,1000], 'max iter':[500])	(C=1000,max iter=500)
Logistic Regression	SVD	('solver': ['lbfgs'], 'penalty': ['l2'], 'C': [1,10,100,1000], 'max iter':[500])	(C=10,max iter=500)
Neural Network	ANOVA	('hidden layer sizes': [(20,30,10)], 'learning rate': ['adaptive'], 'verbose' : [True], 'batch size': [500], 'max iter': [20]\)	(batch size=500,hidden layer sizes=(20,30,10), learning rate='adaptive',max iter=20,verbose=True)
Neural Network	SVD	('hidden layer sizes': [(20,30,10)], 'learning rate': ['adaptive'], 'verbose' : [True], 'batch size': [500], 'max iter': [20]\)	(batch size=500,hidden layer sizes=(20,30,10), learning rate='adaptive',max iter=20,verbose=True)
SVM	ANOVA	(Default)	Default
SVM	SVD	(Default)	Default

Table 2: All algorithms and their hyper-parameters using Grid Search method.

D Full Model Results

ALGORITHM	DIMENSIONALITY REDUCTION	ACCURACY	PRECISION	RECALL	F1	AUC ROC	AUC PR
Naive Bayes	BOW	0.7553	0.7358	0.7966	0.7650	0.7553	0.838025
Naive Bayes	ANOVA	0.7522	0.7193	0.8272	0.7695	0.7522	0.847872
Naive Bayes	SVD	0.5879	0.7692	0.2511	0.3787	0.5879	0.732543
Decision Tree	BOW	0.5150	0.5076	0.9946	0.6722	0.5150	0.700929
Decision Tree	ANOVA	0.5173	0.5088	0.9941	0.6731	0.5173	0.741707
Decision Tree	SVD	0.6628	0.6436	0.7294	0.6839	0.6628	0.697406
Logistic Regression	BOW	0.7775	0.7564	0.8185	0.7863	0.7775	0.863598
Logistic Regression	ANOVA	0.7628	0.7710	0.7475	0.7591	0.7628	0.859912
Logistic Regression	SVD	0.7533	0.7442	0.7716	0.7577	0.7533	0.843775
Neural Network	BOW	0.7803	0.7622	0.8149	0.7877	0.7803	0.869214
Neural Network	ANOVA	0.7681	0.7286	0.8544	0.7865	0.7681	0.864319
Neural Network	SVD	0.7514	0.7246	0.8110	0.7654	0.7514	0.837443
SVM	BOW	0.7745	0.7561	0.8103	0.7823	0.7745	0.862492
SVM	ANOVA	0.7626	0.7721	0.7449	0.7583	0.7626	0.859049
SVM	SVD	0.7526	0.7428	0.7727	0.7574	0.7526	0.842010

Table 3: All the results obtained.

E Full Model Results including Artists

ALGORITHM	DIMENSIONALITY REDUCTION	ACCURACY	PRECISION	RECALL	F1	AUC ROC	AUC PR
Naive Bayes	BOW	0.7468	0.7342	0.7814	0.7571	0.7464	0.833434
Naive Bayes	ANOVA	0.7452	0.7150	0.8238	0.7656	0.7445	0.838678
Naive Bayes	SVD	0.6644	0.6164	0.8879	0.7276	0.6621	0.706037
Decision Tree	BOW	0.5306	0.5184	0.9897	0.6804	0.5260	0.672106
Decision Tree	ANOVA	0.5198	0.5126	0.9973	0.6772	0.5150	0.746012
Decision Tree	SVD	0.6615	0.6830	0.6153	0.6474	0.6620	0.755111
Logistic Regression	BOW	0.7727	0.7554	0.8131	0.7832	0.7723	0.852196
Logistic Regression	ANOVA	0.7560	0.7648	0.7464	0.7555	0.7561	0.854484
Logistic Regression	SVD	0.7579	0.7475	0.7861	0.7663	0.7576	0.844913
Neural Network	BOW	0.7717	0.7416	0.8408	0.7881	0.7710	0.859003
Neural Network	ANOVA	0.7644	0.7698	0.7611	0.7654	0.7645	0.859968
Neural Network	SVD	0.7240	0.6878	0.8305	0.7524	0.7229	0.812163
SVM	BOW	0.7712	0.7616	0.7961	0.7785	0.7709	0.861169
SVM	ANOVA	0.7564	0.7655	0.7461	0.7557	0.7565	0.850588
SVM	SVD	0.7572	0.7489	0.7811	0.7647	0.7570	0.841721

Table 4: All the results obtained with Artists' names.

Statement of contribution

- All authors were considerably involved in the overall experimental design, pre-processing, research process and report writing. Moreover,
- s1657612 mainly contributed to the literature review, exploratory data analysis, BoW and TF-IDF feature encoding, ANOVA dimensionality reduction and discussing report results
- Author s2281922 mainly contributed to building the new classifiers such as DT, SVM and ANN, SVD dimensionality reduction, building intuition on ML methods, Latex set up
- Author s2229609 mainly contributed to building a unified grid search framework, hyper-parameter tuning, LR, DT and NB, performance metrics, results evaluation and cross-validation