PROJECT REPORT:

WHO IS THE NEXT DIVA? BILLBOARD HOT MUSIC 1958-2021

Ang Zhao

Qiaoyi(Joy) Liu

Xiaochen Zhou

IST707 Applied Machine Learning

Professor Joshua Introne

December 16, 2022

INTRODUCTION

Forecasting popularity is a common usage in detecting news features, music billboards, or commercial purchase. This information is extremely time-sensitive; hence it is most desirable to predict the popularity of items/feeds/trends prior to their release, fostering the possibility of appropriate decision-making to modify and adjust (Bandari, 2012). Machine Learning algorithms were broadly used to analyze and predict patterns of user behavior. However, predicting human behavior is always challenging. Take music for example, there are numerous factors that could impact whether a song could reach a certain popularity. One research on the social media impact on music trends found out that traditional media such as radio play has a positive relationship with both song and album sales. On the contrary, the relationship between social media and album sales is largely insignificant, while the relationship between social media and sales at the song level is negative (Dewan, 2014).

In this project, we aim to find patterns for songs that appeared on the Billboard rank and the correlation between music creators, music features, genre, time published, etc. Billboard is an American music and entertainment magazine published weekly by Penske Media Corporation. Its music charts include the Hot 100, the 200, and the Global 200, tracking the most popular albums and songs in different genres of music. Our dataset originated from Billboard and Spotify, one of the largest music streaming service providers. Downloaded from Data World (https://data.world/datasets/open-data), the open access dataset consists of two separate spreadsheets – Hot *100 Audio Features.xlsx* and *Hot Stuff.csv*. The two datasets contain cross-linked information, but both have their own unique variables. The *Hot 100 Audio Features* was scraped through Spotify Web API with music features such as danceability, liveness, tempo, instrumentalness, etc. It also contains a set of music genres for each song labeled by Spotify.

Although predicting hit music is a popular question in machine learning, most models only predict whether a song is ranked in the top 100 (hit) or after 100 (not hit). Reiman and Örnell (2018) used Logistic Regression (accuracy 53%), K-Nearest Neighbors (accuracy 48%), Gaussian Naive Bayes (accuracy 74%), and Support Vector Machine (accuracy 30%) to predict whether a song is a hit or not. Georgieva et al. (2016) also used Billboard hot 100 songs and Spotify music features data to predict whether a song is in the top 100 rank. Their Neural Network model showed the highest accuracy of 76.5%.

These predictions however, focused on whether a song exists in the Billboard 100 rank. There is not a model accurate enough to predict whether a song is more likely to be top 50, 30, or even, 20. We would like to narrow down this range of prediction to see whether music features or genres are correlated with achieving higher popularity.

For record companies, song publishers, singers, and advertising companies whose clients are affiliated with the music industry, the performance of songs on Billboard list is a non-trivial index to make business decisions. For instance, optimization of their marketing costs on songs. Record companies could also determine whether or not to publish a song. Machine learning methods provide the ability to compute high dimensional matrices and deliver prediction results within various aspects of evaluation, so multi-class classification should work well in the music industry.

This paper trained the following classifiers to predict the best positions of songs since 2000: Gradient Boosting, Support Vector Machine, and Decision Tree. The goal is to detect how well classifiers could identify different levels' songs on Billboard list. If the classifiers are reliable, it can provide reasonable advice to the related people/groups.

The results show that with genres, duration and year of that position, prediction is improved in all aspects. When examining songs with high rank/low rank, the gradient boosting classifier reached 70% accuracy. This finding improved the accuracy compared to previous work. When examining songs with top/medium/bottom class, the linear SVM classifier reached 56% accuracy. Additionally, these classifiers show high precision, which enhances the application of this project in business problems addressed above.

GOALS AND OBJECTIVES

Because this project was conducted by three students approaching the problem from various angles, we proposed our objectives individually. This improved our efficiency and methods to solve the problem. The sequence is based on the roles we each took charge in handling the dataset and how one could support the works of another. We collaborated our analyses in the results and conclusions.

In our project, the ultimate goal is to predict the hit of a song using factors that have a strong correlation with the hit of the song. To predict what kind of songs will become hits, it is first necessary to get different characteristics of different factors of songs with different degrees of hits by analysis. Among the factors that we guess will affect the popularity of a song are: song length, song style, song name, etc. Therefore, for our dataset, descriptive statistical analysis was first completed, and then factors with high correlation with song popularity were selected to carry out furthermore detailed analysis. In addition, we would also like to analyze the music style of the players to see if we can summarize it by data analysis.

Research on the psychological correlates of musical preferences indicates that preferences are associated with personality, values, and cognitive abilities (Rentfrow, et al., 2013). There is also evidence that musical preferences are influenced by the social connotations associated with

music (Tarrant, et al., 2002), as people are drawn to musical styles with social characteristics that reflect aspects of their identities (Rentfrow, et al., 2013). Previous research has indicated that particular features in music correlates with the evolution of music trends (Ni, et al., 2015). For instance, slower songs such as ballads were popular from the 1980s to the 1990s, whilst in the new century music listeners prefer faster songs. A trend in hits also indicates popular songs are getting relatively louder.

One of the main objectives in this project is to analyze the time trend of music features and music genre. The longevity of the dataset (from 1958 to 2021) allows us to measure the changes in users' preference and social factors that were represented in the rank i.e. the popularity of the songs. We seek to understand the reasons behind the most popular songs and how it reflects cultural, social problems and circumstances in multidimensional ways.

We would also like to discover whether the genre would be correlated with whether a song is a hit i.e. the top 30, 50 in the Billboard 100 chart. Although the music features of the most popular songs do not indicate a conspicuous trend, we would like to use the genres as predictors to see whether composing a song in trending genres could lead to success.

METHODS

The *Hot 100 Audio Features* and *Hot Stuff* datasets are from Data World. *Hot 100 Audio Features* has 29,503 entries and *Hot Stuff* has 327,895 entries. The *Hot 100 Audio Features* was generated through Spotify API, and the *Hot Stuff* is the dataset with Billboard ranking information.

Before conducting any analyses, we first used SongID from both datasets to merge the data into one CSV file. We kept the raw dataset unchanged to prevent any data loss or faults. We removed any entry with NA values. Because the *Hot Stuff* dataset may contain duplicates of the

same song and performer based on the number of times it appears in the chart. So, we removed duplicates and kept each song singular by the highest-ranking position. All modifications and new datasets created throughout this project were recorded in the Data Dictionary and uploaded to OneDrive shared folder. Based on various tasks, our methods were different according to the needs for analyses.

In producing sample summaries and conducting descriptive statistical analyses, we used measures such as averages, graphs, charts, frequencies, histograms, boxes, and percentages. For example, the different pictures of song duration among songs with different popularity and the trends. For example, the trend of topping time for different songs with different patterns (The results are not shown in this report because a full analysis was not implemented for topping patterns in this project.) For the analysis related to text content, we combined with NLP technology to complete lexical analysis. to get which words or content appear frequently in the song titles of songs with different hit levels, the frequency of different music genres in different musical works with word clouds reflecting different performers, etc.

To generate a pattern for music features, we first reduced the dimension using Principal Component Analysis (PCA). Music features have many attributes such as danceability, liveliness, etc. with numeric or categorical values to define a music piece. One other concern is that each song has multiple music genres named by Spotify. To generate a pattern for that, we used Hierarchical Clustering to classify the songs using music features and defined each cluster with one genre by its maximum count. Based on computational time concerns, we sampled 2000 random songs out of the dataset to process these analyses. To compare the trend from 1958, we abandoned the month and dates and divided the year into decades. We used PCA to reduce the dimension of music features.

Next, we used the Agglomerative Clustering method to cluster the random 2000 pieces of music by their music features. Before that we used the Elbow Method to find out the number of clusters. Similar to the music features, we would also like to see the pattern of genre change in decades, thus it is necessary to deduce the variety in genre types. We used the Elbow Method to define how many clusters and the Agglomerative Clustering method to cluster songs by all of the music features variables. Then, we used NLTK word tokenization and Part-Of-Speech (POS) tagging to count the number of times each genre in each cluster appeared in the cluster. By doing so, we were able to reduce the genre types in each cluster by using only the top genres to replace the genres for songs that belong to this cluster.

After replacing the genres from clustering, we used Naive Bayes, Decision Tree, Random Forest, Gradient Boosting, k-Nearest Neighbor (kNN), Neural Network, Logistic Regression, and Support Vector Machine (SVM) models to predict whether songs are in top 50 or not using the newly created genres and music features - danceability, acousticness, energy, instrumentalness, liveness, tempo, valence, speechiness, key, and loudness (excluding mode). The ratio for training and testing dataset is 8:2. All predictions were conducted in Orange. All categorical variables were discretized by equal-frequency discretization. All models used 3-fold cross validation.

For the purpose of prediction on songs since 2000, split the whole dataset into training and testing data with the ratio 7:3. Fix the random state as 1337 for the below procedure.

In prediction, generate the target variable according to the following tasks separately:

1. Predict high /low rank(peak position 1-50/51-100)
2. identify top/medium/bottom class(peak position 1-30/31-60/61-100)

For feature engineering, generate dummy variables for each genre in the dataset.

For the date, first convert it as three variables: day, month, year. Then create 4 dummy variables of year ranges to enhance the performance of classifiers(2000-2005;2006-2010;2011-2015;2016-2022). Normalized *tempo, Spotify popularity, duration* to reduce the execution time. In the prediction step, only keep music features, duration, year ranges, and genres(applicable).

The prediction procedure compared the results between genres and without genres. This paper chose three classifiers to implement each of the above 2 tasks : Gradient Boosting, Support Vector Machine(SVM), and Decision tree. Gradient Boosting is an ensemble method where new estimators learn from errors created by previous estimators. By increasing the weight of misclassified samples, GBoosting tries to understand the labels of those records correctly.

Support Vector Machine(SVM) is to set boundaries as far as possible from support vectors. It maximizes the “distance” between two separate spaces and finds the most confident prediction. Decision tree is to split data based on attributes leading to prediction results.

When searching the best classifier for each algorithm, the procedure is the following: 1) tune important hyperparameters by using validation curve: learning rate, max depth of each tree, max iteration of learning; 2) with narrowed ranges of those parameters, use Grid Search and Cross validation to find the optimal settings of these classifiers(cv=5). The score is the test accuracy; 3) evaluate selected classifiers by confusion matrix and performance reports(F1-scores, precision, accuracy, etc.); and 4) detect potential overfitting issues with learning curves(cv=5).

Important parameters for each classifier:

1. GBoosting: learning rate, max depth of each tree, max iteration of learning.
2. SVM: kernel, Penalty parameter C and gamma(not applicable in linear). To reduce the execution time, search C and gamma separately for each kernel function(RBF, sigmoid, and linear). Select the one with the highest accuracy.

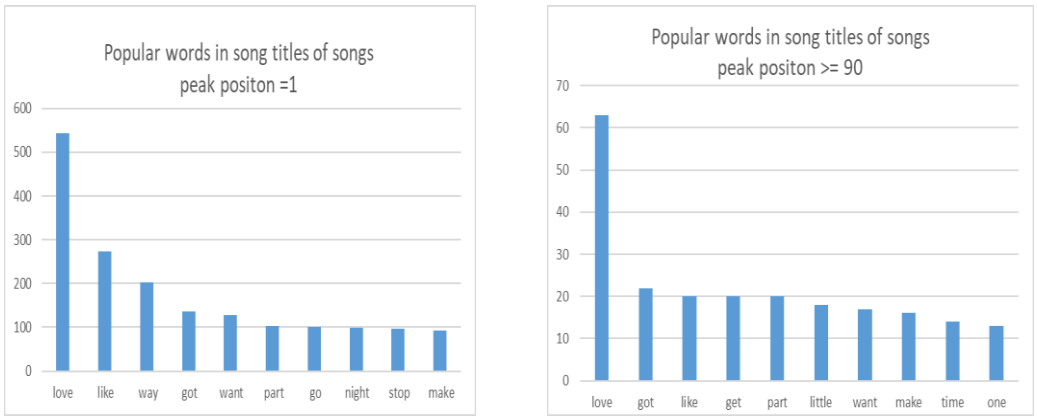
3. Decision tree: criterion, minimum sample to split, maximum depth of the tree, minimum samples on each leaf.

For the evaluation, first consider F1-score/accuracy as they depict the classifier in a general view. Specifically, compare the precision as it reflects how many predictions are correct in fact.

RESULTS

First, we want to know if there is a difference in the content of very popular songs(Peak Position=1)and not so popular songs(Peak Position >=90). Since there is no data on the lyrics, we applied NLP techniques for the song titles and completed Unigram word frequency analysis. From Figure 1, we can see that the song titles of songs with Peak Position=1 and songs with Peak Position >=90 both reflect that most of the songs in these two hit levels are related to love stories.

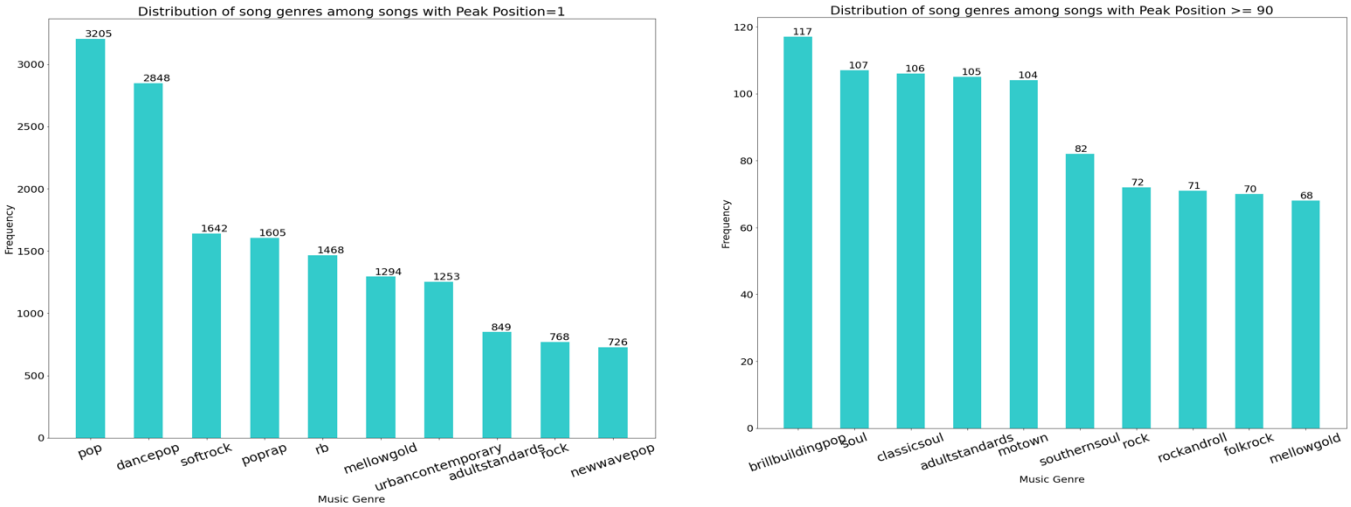
Figure 1. Popular Words in Song Titles in Peak Position = 1 and >= 90



We then explored whether there was a difference in the genres of songs with different hit levels. By splitting the genre set of songs and performing statistical type analysis, we can see that there is a difference in the genres of songs with different popularity levels. Figure 2 shows the distribution of the types of songs with different popularity. For example, the genres Pop, Dance Pop, Soft Rock, Pop Rap, and R&B appear more frequently in songs with the Peak Position = 1. For example, for songs with a Peak Position outside of 90, the genres Brill Building Pop, Soul,

Classic Soul, Adult Standards, and Motown appear more frequently. Therefore, we can use the song genre as an input to help the model better predict the song's popularity.

Figure 2. Distribution of Song Genres among Songs with Peak Position = 1 and >= 90



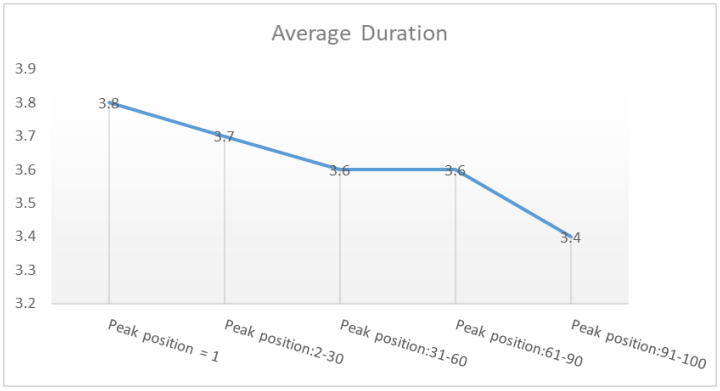
Here, we applied NLP techniques to show the distribution of musical genres of several characteristic players. We can see that the most frequent genre of Madonna's music is Dance pop, the most frequent genre of Eminem's music is Detroit hip hop, and the most frequent genre of Taylor Swift's music is post-teen pop.

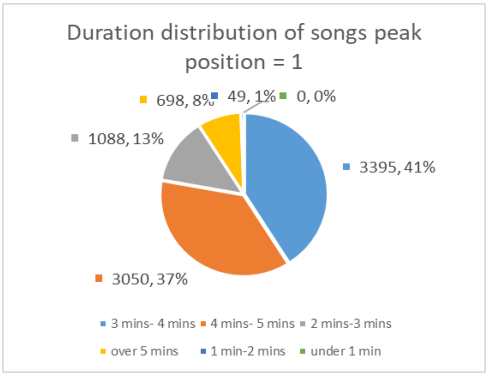
Figure 3. Word Clouds (from left to right: Madonna, Eminem, Taylor Swift)

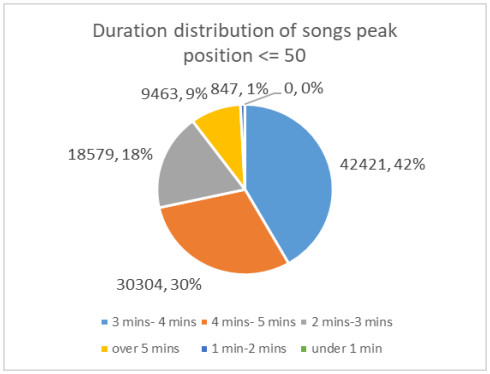
By looking at Figure 4, we can see that there are differences in the duration of songs with different levels of popularity. For songs with a peak position of 50 or less, a 3–5-minute duration

is more. For songs whose peak position is outside 90, a 2–4-minute duration is more. The song duration tends to decrease as the peak position of the song decreases.

Figure 4. Average Song Duration and Distributions







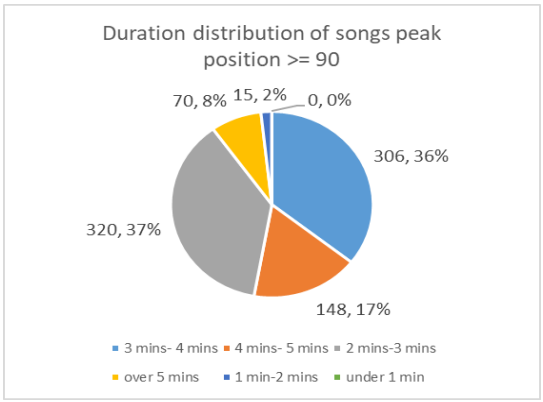
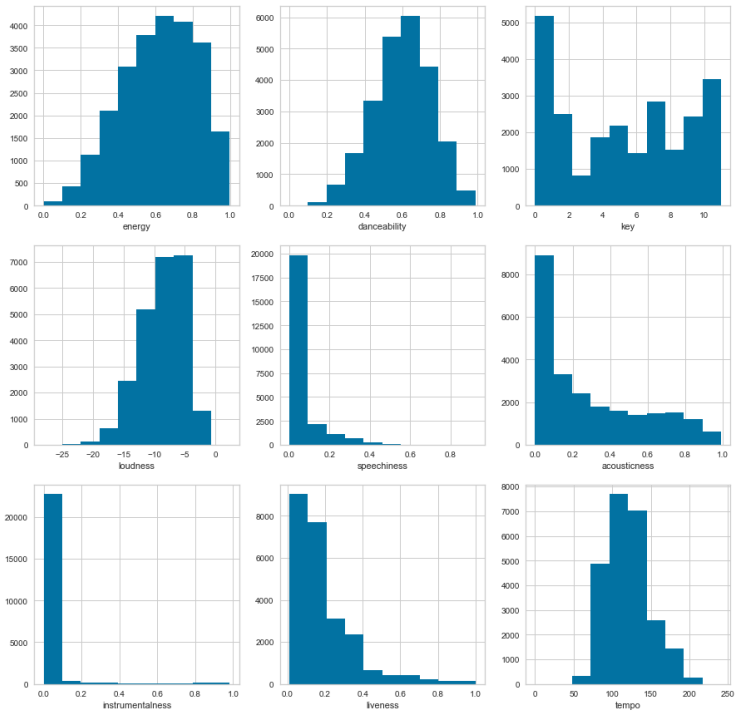


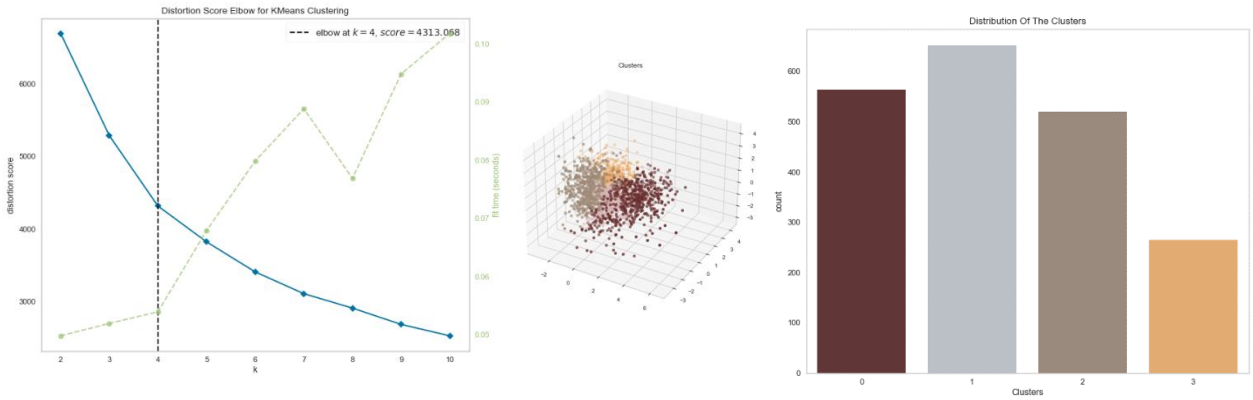
Figure 5. Music Features Histograms



More details and results are presented in supplementary materials (see Appendix B).

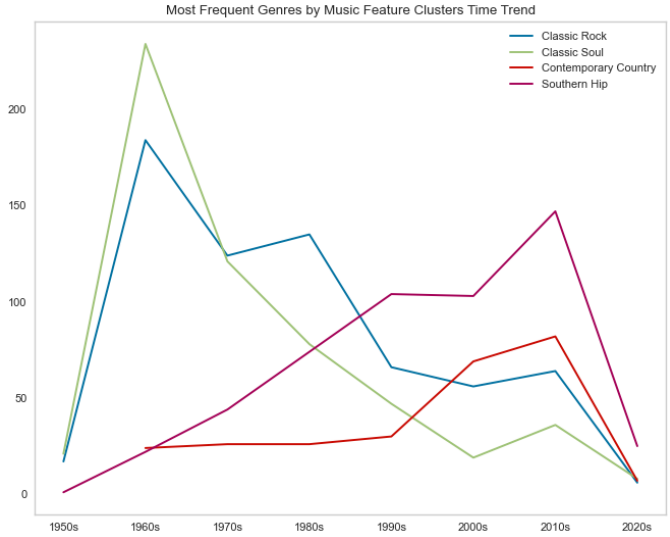
Figure 6. Music Features Preferences via Decades

Figure 7. Elbow Method, Clustering by Music Features and Clusters Distribution



Elbow showed the k means equals 4. After Agglomerative Clustering, the first most frequent genre in the four clusters are: Classic Soul, Classic Rock, Southern Hip, and Contemporary Country. The sum of the genres by decade shows Classic Soul and Classic Rock have a decline and Southern Hip and Contemporary Country have increased in popularity.

Figure 8. Genres Change via Decades



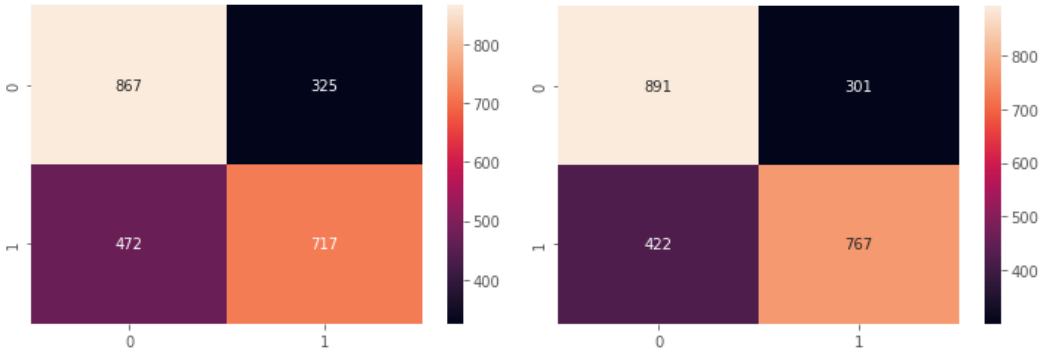
We also used clustered genre, danceability, acousticness, energy, instrumentalness, liveness, tempo, valence, speechiness, key, and loudness to predict songs ranked top 50 or 50 to 100 produced the highest accuracy. The highest F1 was 55.9% using Random Forest. Further details on the results are in Appendix C.

Below are 2 tasks about predicting positions since 2000.

TASK 1: HIGH/LOW RANK

First compare results in Gradient Boosting.

Figure 9. Confusion Matrix(GBT-no genres) (left) and Confusion Matrix (GBT- genres) (right)



From Figure 9, genres did improve the prediction. The on diagonal elements are both enlarged. Table 1 and 2 also illustrated this finding. Each index is improved. Specifically, accuracy reached 70% after including genres. Also, precision of high/low rank improved 3% from Table-B. Parameter settings are slightly different between two classifiers, so genres are the main factor for this high accuracy. Then it can be said that high/low rank can be classified by the Gradient Boosting method.

Table 1. Classification report for Gradient Boosting (no genres)

	precision	recall	f1-score	support
High rank	**0.65**	0.73	0.69	1192
Low rank	**0.69**	0.60	0.64	1189
accuracy	/	/	**0.67**	2381
macro avg	**0.67**	0.67	**0.66**	2381
weighted avg	0.67	0.67	0.66	2381
HistGradientBoostingClassifier: L2=1, learning rate=0.08, max depth=8, max iteration=40, min samples leaf=10, random state=1337				

Table 2. Classification report for Gradient Boosting

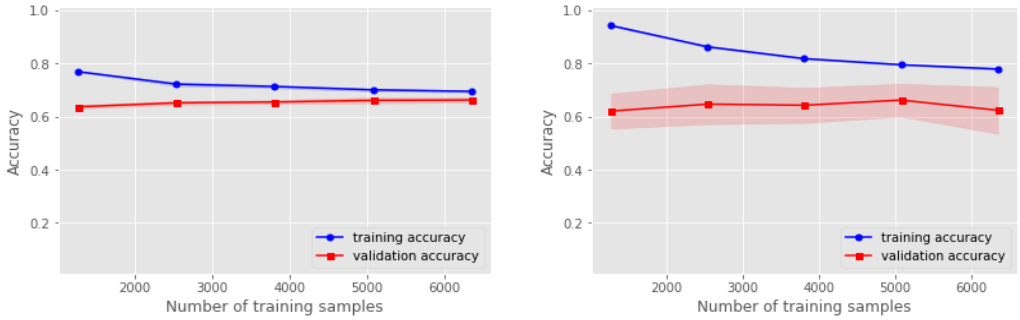
	precision	recall	f1-score	support
High rank	**0.68**	0.75	0.71	1192
Low rank	**0.72**	0.65	0.68	1189
accuracy			**0.70**	2381
macro avg	**0.70**	0.70	**0.70**	2381
weighted avg	0.70	0.70	0.70	2381
HistGradientBoostingClassifier: L2=1, learning rate=0.05,max_depth=8, max iteration =80, min samples leaf =10, random state=1337				

Figure 10 (left) shows there is no overfitting issue in GBoost without genres. Figure 10 (right) shows potential risk on overfitting issues(GBoost including genres), and two accuracy

curves didn't converge as sample size increases. The recommendation is to apply GBoost classifiers without genres.

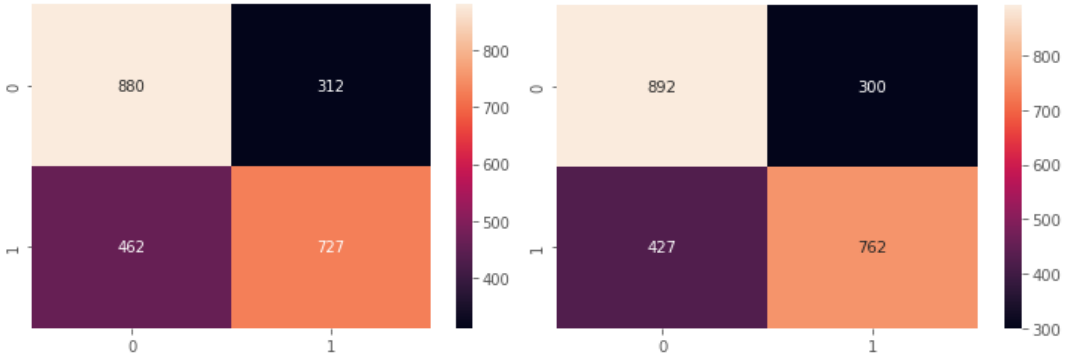
As has been widely known, precision evaluates among all samples predicted as positive(high rank), how many of them are true positive(truly high ranked). With relatively high precision, it helps song publishers to gain confidence about prediction on new songs/unpublished songs. The recommendation is to use Gradient Boosting Classifier with genres for better precision, and without genres for general evaluation.

Figure 10. Learning Curve (GBT-no genres) (left) and Learning Curve (GBT- genres) (right)



Second compare results in Support Vector Machine.

Figure 11. Confusion Matrix (SVM-no genres) (left) and Fig-F Confusion Matrix (SVM-genres) (right)



From Figure 11 (left and right), genres did improve the prediction. The on diagonal elements are both larger. Table 3 and 4 support this finding. Each index is improved. Specifically, accuracy reached 69% after including genres. Also, precision and recall improved for both labels

from Table 4. Parameter settings are different between two classifiers which led to different computational time. Combined with easier understanding and less execution time, the higher accuracy and precision in Table 4 indicates that LinearSVC(included genres) is more suitable for the high/low rank task.

Table 3. Classification report for SVM(no genres)

	precision	recall	f1-score	support
High rank	**0.66**	0.74	0.69	1192
Low rank	**0.70**	0.61	0.65	1189
accuracy			**0.67**	2381
macro avg	**0.68**	0.67	**0.67**	2381
weighted avg	0.68	0.67	0.67	2381
SVC: kernel =RBF, C=1.7, gamma=0.01				

Table 4. Classification report for SVM

	precision	recall	f1-score	support
High rank	**0.68**	0.75	0.71	1192
Low rank	**0.72**	0.64	0.68	1189
accuracy			**0.69**	2381
macro avg	**0.70**	0.70	**0.69**	2381
weighted avg	0.70	0.70	0.69	2381
SVC: kernel =linear, C=0.66				

Figure 12. Learning Curve (SVM-no genres) (left) and Learning Curve (SVM- genres) (right)

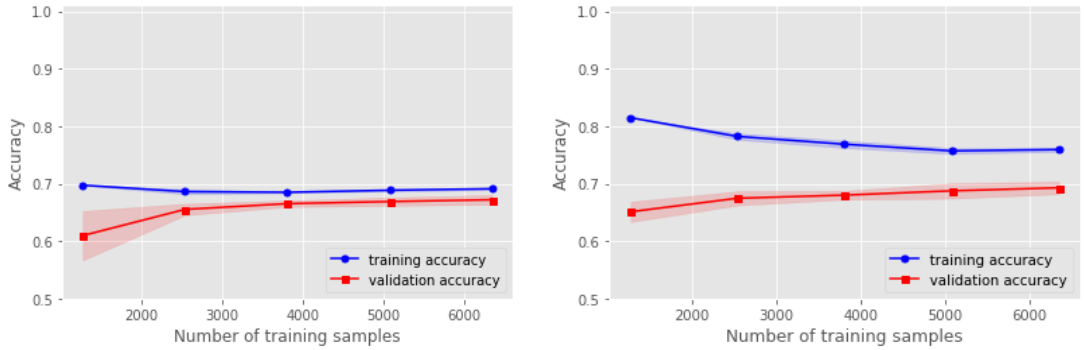
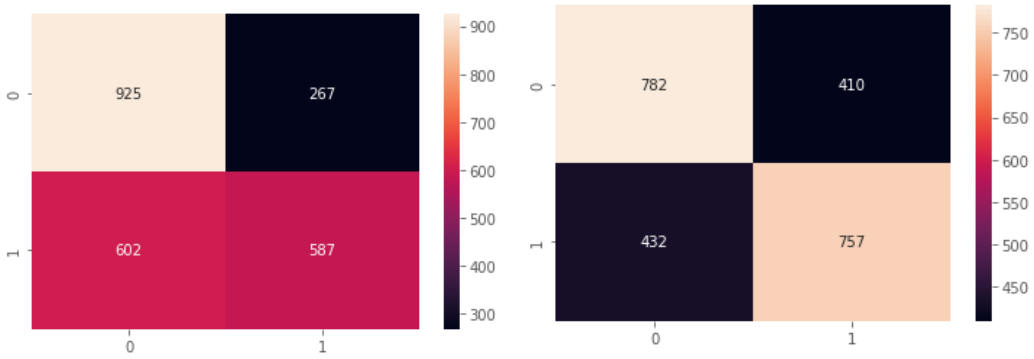


Figure 12 (left and right) shows there is no overfitting issue in SVM regardless of genres, but the converge rates aren't the same. If applying the selected Linear SVC(with genres), one might consider massive dataset.

Last compare results in Decision Tree.

Figure 13. Confusion Matrix(DTC-no genres) (left) and Confusion Matrix (DTC- genres) (right)



From Figure 13 (left and right), genres only made the classifier better when label=low rank. For high ranked songs, it became worse. Table 5 and 6 also found the same pattern on F1-score. Recall went downwards for high rank from Table 6. Only accuracy and F1-score(macro average) improved to 65% after including genres. Considering the business problems this report focused, the Decision tree classifier without genres is better for this task.

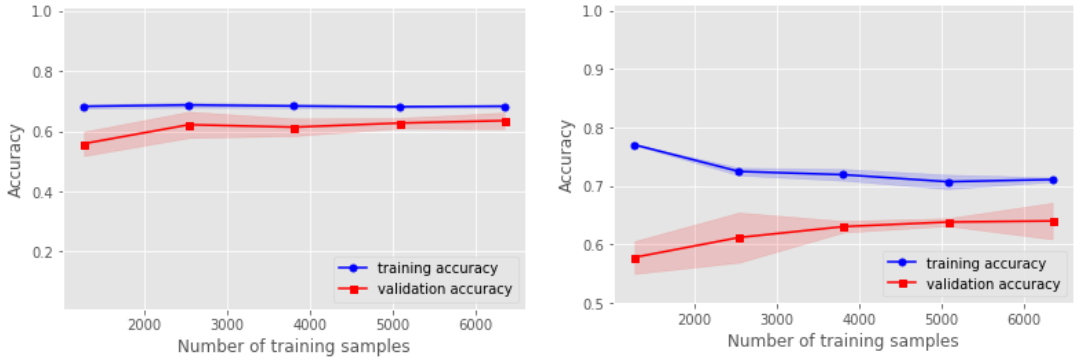
Table 5. Classification report for DTC(no genres)

	precision	recall	f1-score	support
High rank	**0.61**	0.78	0.68	1192
Low rank	**0.69**	0.49	0.57	1189
accuracy			**0.64**	2381
macro avg	**0.65**	0.63	**0.63**	2381
weighted avg	0.65	0.64	0.63	2381
DecisionTreeClassifier: criterion='entropy', max depth=6, min samples leaf=5, random state=1337				

Table 6. Classification report for DTC

	precision	recall	f1-score	support
High rank	**0.64**	0.66	0.65	1192
Low rank	**0.65**	0.64	0.64	1189
accuracy			**0.65**	2381
macro avg	**0.65**	0.65	**0.65**	2381
weighted avg	0.65	0.65	0.65	2381
DecisionTreeClassifier: max depth=7, min samples leaf=4, random state=1337				

Figure 14. Learning Curve (DTC-no genres) (left) and Learning Curve (DTC- genres) (right)

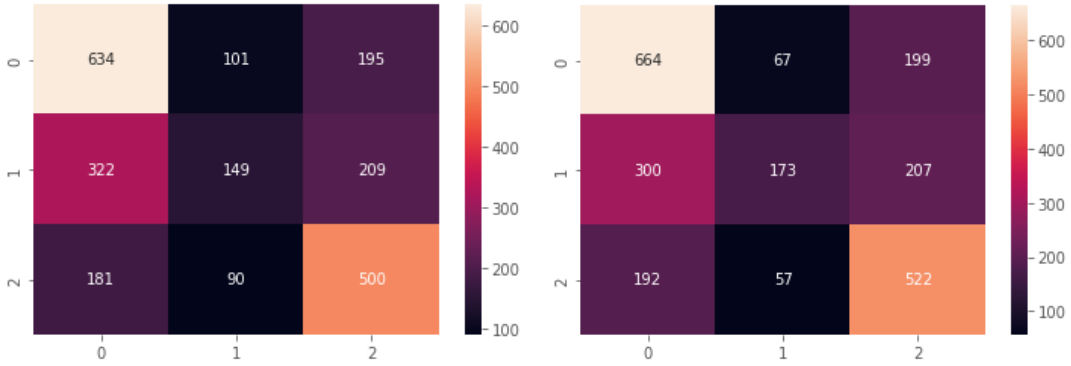


There is no significant overfitting issue in Figure K and L. Combined with most curious questions in industry, this paper recommends Decision tree without genres.

TASK 2: TOP/MEDIUM/BOTTOM CLASS

First compare results in Gradient Boosting.

Figure 14. Confusion Matrix (GBT-no genres) (left) Confusion Matrix(GBT- genres) (right)



From Figure 14 (left and right), genres enhanced the prediction. The on diagonal elements are all enlarged. The medium class is still the one misclassified most. Table 7 and 8 also illustrated this finding. Each index is improved. Specifically, precision of medium class improved from 44% to 58% after including genres. Also, F1-score improved to 54% from Table 8. Since parameter settings are very different between two classifiers, GBoosting learned a lot of information from genres. For GBoost, including genres is more reasonable for higher F1-score and higher precision.

Table 7. Classification report for Gradient Boosting(no genres)

	precision	recall	f1-score	support
Top class	**0.56**	0.68	0.61	930
Medium class	**0.44**	0.22	0.29	680
Bottom class	**0.55**	0.65	0.60	771
accuracy			**0.54**	2381
macro avg	**0.52**	0.52	**0.50**	2381
weighted avg	0.52	0.54	0.52	2381
HistGradientBoostingClassifier: L2=1, learning rate=0.06, max depth=3, min samples leaf=5, random state=1337				

Table 8. Classification report for Gradient Boosting

	precision	recall	f1-score	support
Top class	**0.57**	0.71	0.64	930
Medium class	**0.58**	0.25	0.35	680
Bottom class	**0.56**	0.68	0.61	771
accuracy			**0.57**	2381
macro avg	**0.57**	0.55	**0.54**	2381
weighted avg	0.57	0.57	0.55	2381
HistGradientBoostingClassifier: L2=1, learning rate=0.04,max_depth=5, max iteration =40, min samples leaf =10, random state=1337				

Figure 15. Learning Curve (GBT-no genres) (left) and Learning Curve (GBT- genres) (right)

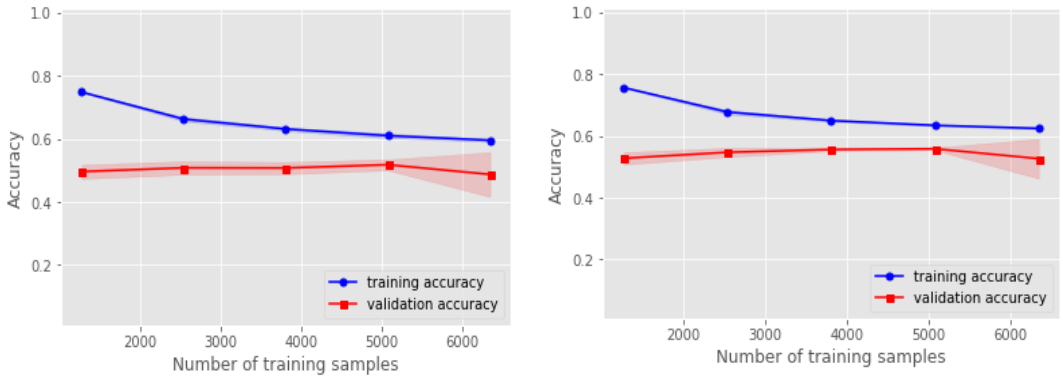
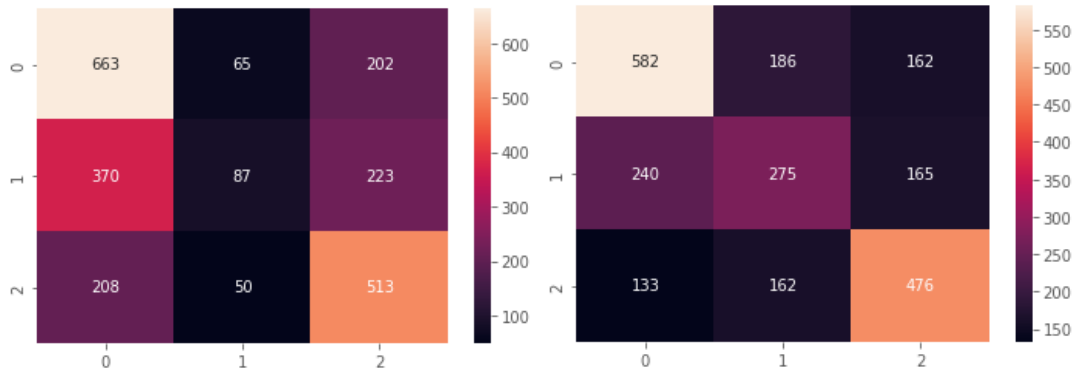


Figure 15 (left and right) didn't find obvious overfitting issues in GBoost, but these two pairs of lines didn't converge when sample size increases. Therefore, one could see overfitting when training samples continue added in. Both are not recommended for general prediction.

Second compare results in Support Vector Machine.

Figure 16. Confusion Matrix (SVM-no genres) (left) and Confusion Matrix (SVM- genres) (right)



From Figure 16 (left and right), it is easy to find the trade-off of adding genres: the medium class are now better to be recognized, but top/bottom class are harder to be identified. Table 9 and 10 indicated this finding in the recall column. Precision of the top class improved from 53% to 61% after including genres. Also, F1-score improved to 55% from Table-10. Within SVM, it's hard to say whether one should include genres.

Table 9. Classification report for SVM(no genres)

	precision	recall	f1-score	support
Top class	**0.53**	0.72	0.61	930
Medium class	**0.43**	0.13	0.20	680
Bottom class	**0.55**	0.67	0.60	771
accuracy	/	/	**0.53**	2381
macro avg	**0.50**	0.50	**0.47**	2381
weighted avg	0.51	0.53	0.49	2381
SVC: C=0.7, gamma=0.01				

Table 10. Classification report for SVM

	precision	recall	f1-score	support
Top class	**0.61**	0.63	0.62	930
Medium class	**0.44**	0.40	0.42	680
Bottom class	**0.59**	0.62	0.60	771
accuracy	/	/	**0.56**	2381
macro avg	**0.55**	0.55	**0.55**	2381
weighted avg	0.56	0.56	0.56	2381
SVC: C=0.67, kernel=linear				

Figure 17. Learning Curve (SVM-no genres) (left) and Learning Curve (SVM- genres) (right)

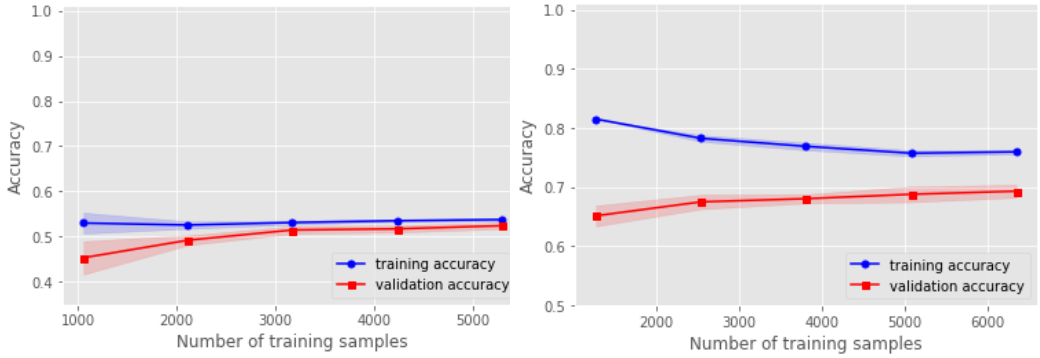
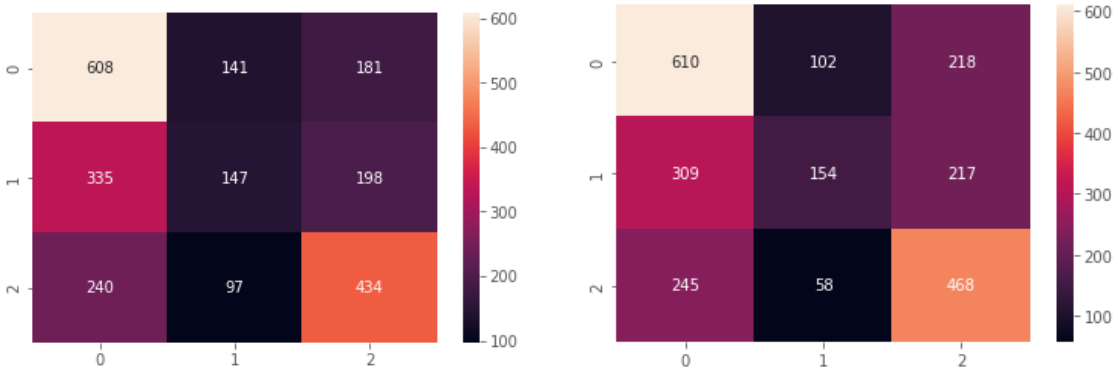


Figure 17 (left and right) didn't show overfitting issues. The larger gap in 17 (right) states that genres weaken the robustness of SVM. Therefore, no strong recommendation is made.

Last compare results in Decision Tree.

Figure 18. Confusion Matrix (DTC-no genres) (left) Confusion Matrix (DTC- genres) (right)



From Figure 18 (left and right), all labels are more easily identified, though the change is not significant. Table 11 and 12 shows that all recalls are slightly improved. Precision of the medium class improved from 38% to 49% after including genres, but for other two classes, changes are not remarkable. Also, F1-score improved 1% from Table 12. Since the priority of prediction is to detect popular songs, the recommendation is to apply Decision tree classifier without genres for the sake of execution time.

Table 11. Classification report for DTC(no genres)

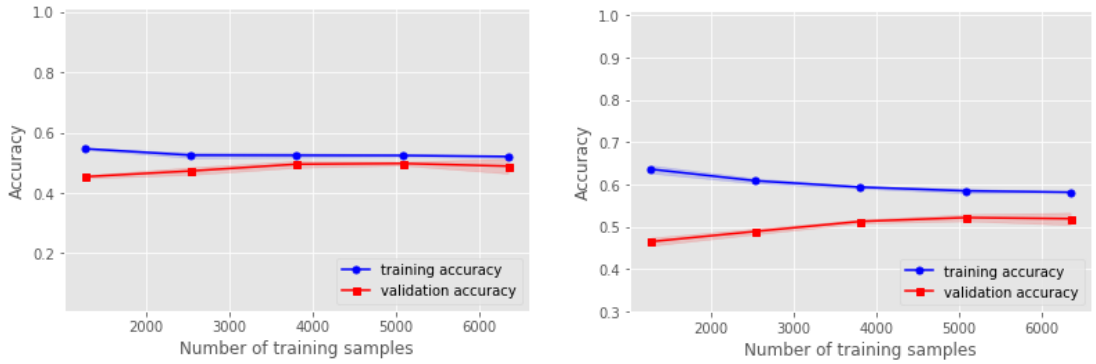
	precision	recall	f1-score	support
Top class	**0.51**	0.65	0.58	930
Medium class	**0.38**	0.22	0.28	680

Bottom class	**0.53**	0.56	0.55	771
accuracy			**0.50**	2381
macro avg	**0.48**	0.48	**0.47**	2381
weighted avg	0.48	0.50	0.48	2381
DecisionTreeClassifier: criterion='entropy', max depth=5, min samples leaf=2,random_state=1337				

Table 12. Classification report for DTC

	precision	recall	f1-score	support
Top class	**0.52**	0.66	0.58	930
Medium class	**0.49**	0.23	0.31	680
Bottom class	**0.52**	0.61	0.56	771
accuracy			**0.52**	2381
macro avg	**0.51**	0.50	**0.48**	2381
weighted avg	0.51	0.52	0.50	2381
DecisionTreeClassifier: criterion='entropy', max depth=7, min samples leaf=3,random_state=1337				

Figure 19. Learning Curve (DTC-no genres) (left) and Learning Curve (DTC- genres) (right)



No overfitting issue has been found from Figure 19 (left and right).

DISCUSSION

From results we can see that since available data in the 1950s, the trending genre in Billboard 100 songs were classic rock and soul. With a peak during the 1960s, the two classics began to decline. Meanwhile, contemporary country and southern hip slowly became the trending genres. We used the four genres combining the music features to predict whether a song is in the top 50 or between 51 to 100. Our best model was the Extreme Gradient Boosting Random Forest from the XGBoost package. It was challenging to predict songs that were already in the top 100 hit list. From data

analyses we were able to conclude that the genres showed a trend in each decade, but there is more to discover of what features contribute to songs reaching the highest popularity.

Overall, predictions of songs since 2000 are improved compared to previous research. From the results, it is straightforward that choice of classifiers depends on the trade-off between different index, different labels. In High/low rank, it is clear to see that with more information from genres, the prediction is better in all aspects. In 3-class task, top/bottom class are easier to be identified. Medium class are often mislabeled without genres. However, genres could cause potential overfitting issues as many learning curves in genre groups don't converge. For the convenience of song publishers and record companies, this paper list the following advice:

1. To detect high/low rank, use the selected GBoost /linear SVM(with genres) for prediction. SVM /GBoost(no genres) for references.
2. To investigate top/bottom class, use the selected linear SVM(with genres) for prediction. For medium class, use GBoost(with genres) for prediction. Use selected SVM/Decision tree (no genres) for references.

CONCLUSION

We were able to practice machine learning theories and algorithms to solve problems regarding the popularity of songs. By tuning parameters, we refined the models to reach a higher accuracy. We used multiple softwares and coding languages such as Orange, Python, Stata, and visualization applications such as Canvas. We used matplotlib, seaborn, etc. to visualize boxplots, violins, histograms, line plots, etc. to showcase our results and present it to the class. Although the model accuracies were not ideal, it showed the significance of conducting more research and analyses on what exactly boosted popular songs to become the hottest on chart. We applied methodologies

from Natural Language Processing to analyze the song titles, song performers, and genres to improve our understanding of the dataset and contribute to the machine learning models.

This research has developed several applications of ML prediction on Billboard position. As findings indicate, more information is needed to raise classifiers' performance. Future work could combine the YouTube MV volume, trends on social media such as Instagram, TikTok, and how the singer(s) market this song. With more time trend data, this project could apply survival analysis to pitch the threshold of being a popular song.

REFERENCE

Bandari, R., Asur, S., & Huberman, B. (2012). The pulse of news in social media: Forecasting popularity. In *Proceedings of the International AAAI Conference on Web and Social Media* (Vol. 6, No. 1, pp. 26-33).

Dewan, S., & Ramaprasad, J. (2014). Social media, traditional media, and music sales. *Mis Quarterly*, *38*(1), 101-122.

Georgieva, E., Suta, M., & Burton, N. (2018). Hitpredict: Predicting hit songs using spotify data.

Reiman, M., & Örnell, P. (2018). Predicting hit songs with machine learning.

Wikipedia. Billboard (magazine). https://en.wikipedia.org/wiki/Billboard_(magazine)

Wikipedia. Spotify. https://en.wikipedia.org/wiki/Spotify

APPENDIX A - TEAM PLAN

Team Member	Tasks
Ang Zhao	• Preprocessing data • Feature engineering(only in prediction): convert year into dummies; scale duration, tempo, popularity. • Predict three tasks by music features, year(dummy) and duration(since 2000): o Binary rank(high rank: 1-50; low rank: 51-100) o 3 ranges(top class: 1-30; medium class: 31-60; bottom class: 61-100) o 4 tiers(tier1: 1-20; tier 2: 21-45; tier 3: 46-75; tier 4: 76-100) o Methodology: o Gradient boosting/SVM/Decision tree: Tune hyperparameters by validation curves, find optimal classifiers with GridSearchCV, evaluate selected classifiers by performance reports. Detect overfitting issues. • Predict three tasks by music features, year(dummy), duration and all genres(dummies) (since 2000): o Binary rank(high rank: 1-50; low rank: 51-100) o 3 rank ranges(top class: 1-30; medium class: 31-60; bottom class: 61-100) o 4 tiers(tier1: 1-20; tier 2: 21-45; tier 3: 46-75; tier 4: 76-100) o Methodology: o Gradient boosting/SVM/Decision tree: Tune hyperparameters by validation curves, find optimal classifiers with GridSearchCV, evaluate selected classifiers by performance reports. Detect overfitting issues. • Construct propensity score o Select first and second terms having significant effects on Pscore. o Calculate estimated propensity score. • Construct blocks based on propensity score(with Imbens and Rubens' procedure) • Testify if blocks are balanced. • Estimate causal effects with subclassification with blocks(-2.2). Due to space limitation, I didn't put it into the report.
Qiaoyi(Joy) Liu	• Selection of dataset for project. • Background research on the developments in song popularity prediction. • Applied clustering skills and models to reduce the dimension of the music features and genres variables. • Visualized results using Python matplotlib and seaborn package. • Described results and progress in Checkpoint1, Checkpoint2, and Final Presentation. • Collaborated with team members on data sharing and analysis improvements. • Developed workflow and project reports outline. • Practiced Orange machine learning models on the clustered data to predict popularity (top 50 or 50-100). • Compared results with other published research and concluded the significance and obstacles of this project. • Evaluated factors that impact the song's popularity from multiple angles.
Xiaochen Zhou	• Data Preprocessing • Descriptive Analysis • Duration characteristics of songs with different hit levels. • The characteristics of song titles of songs with different hit levels. • The distribution of music genres of songs with different hit levels. • The distribution of music styles of different performers. • The trend of top songs of different genres and hits levels. • Visualized results using Python matplotlib and seaborn package. • Collaborated with team members on data sharing and analysis improvements.

APPENDIX B – DATASET VARIABLES AND DEFINITION

Table 1. Hot 100 Audio Features Variables and Description

Column Name	**Description**
SongID	Song title + song performer
Performer	Performer full name (first name + last name)
Song	Song title
spotify_genre	Genres in list
spotify_track_id	Spotify track ID
spotify_track_preview_url	URL
spotify_track_duration_ms	Song duration in milliseconds
spotify_track_explicit	Whether song is explicit
spotify_track_album	Album name
danceability	Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.
energy	Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.
key	The estimated overall key of the track. Integers map to pitches using standard Pitch Class notation . E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on. If no key was detected, the value is -1.
loudness	The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typical range between -60 and 0 db.
mode	Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.
speechiness	Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.
acousticness	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic

instrumentalness	Predicts whether a track contains no vocals. “Ooh” and “aah” sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly “vocal”. The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.
liveness	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.
valence	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).
tempo	The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.
time_signature	An estimated overall time signature of a track. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure).

Table 2. Hot Stuff Variables and Description

Column Name	**Description**
url	URL
WeekID	Week ID
Week Position	Weekly position
Song	Song title
Performer	Performer full name
SongID	Song title + performer full name
Instance	The number of appearances
Previous Week Position	Previous week rank position
Peak Position	The peak rank position
Weeks on Chart	The number of weeks on the Billboard chart

APPENDIX C - SUPPLEMENTARY MATERIALS

Figure 1. Music Features and Pearson Correlation

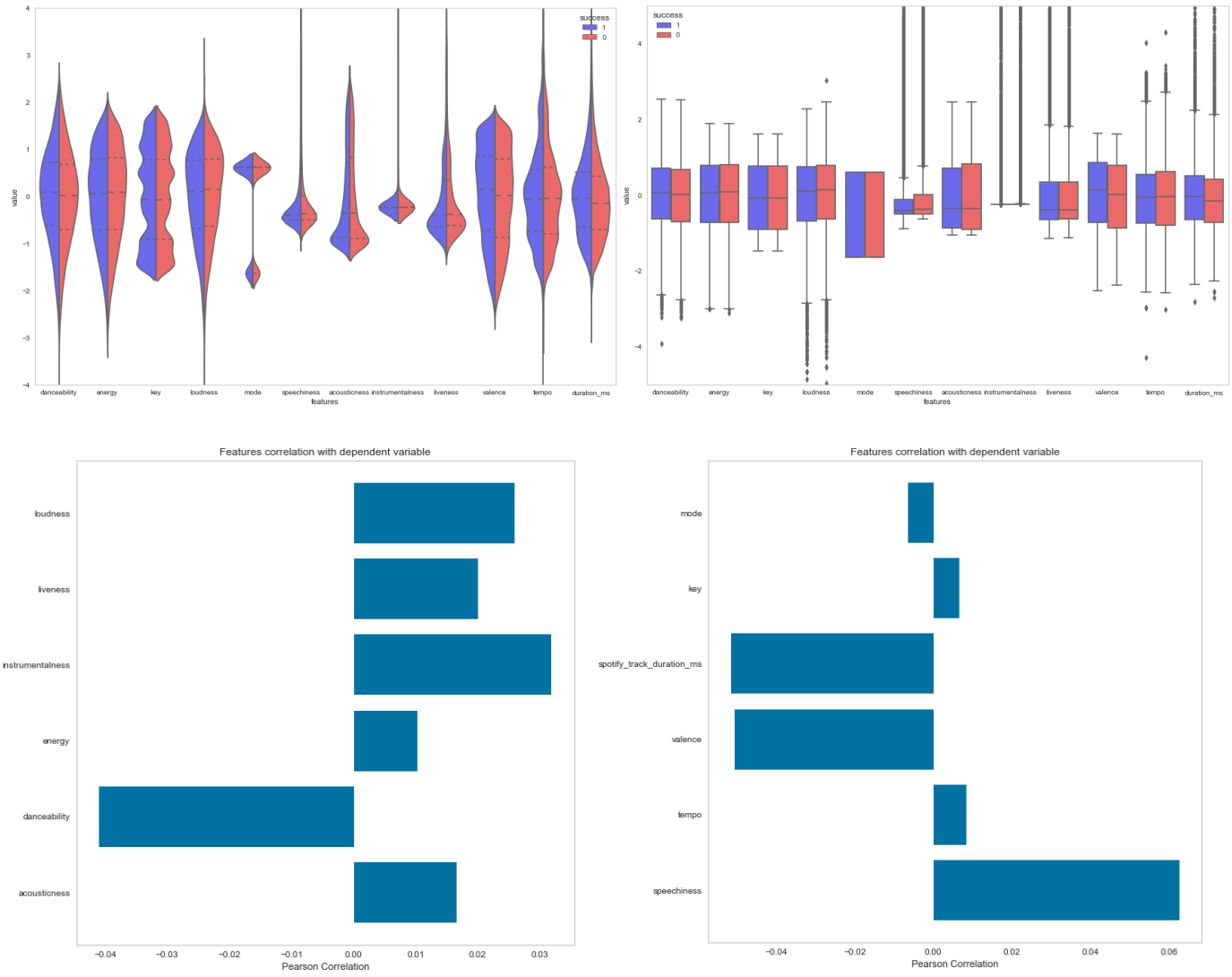
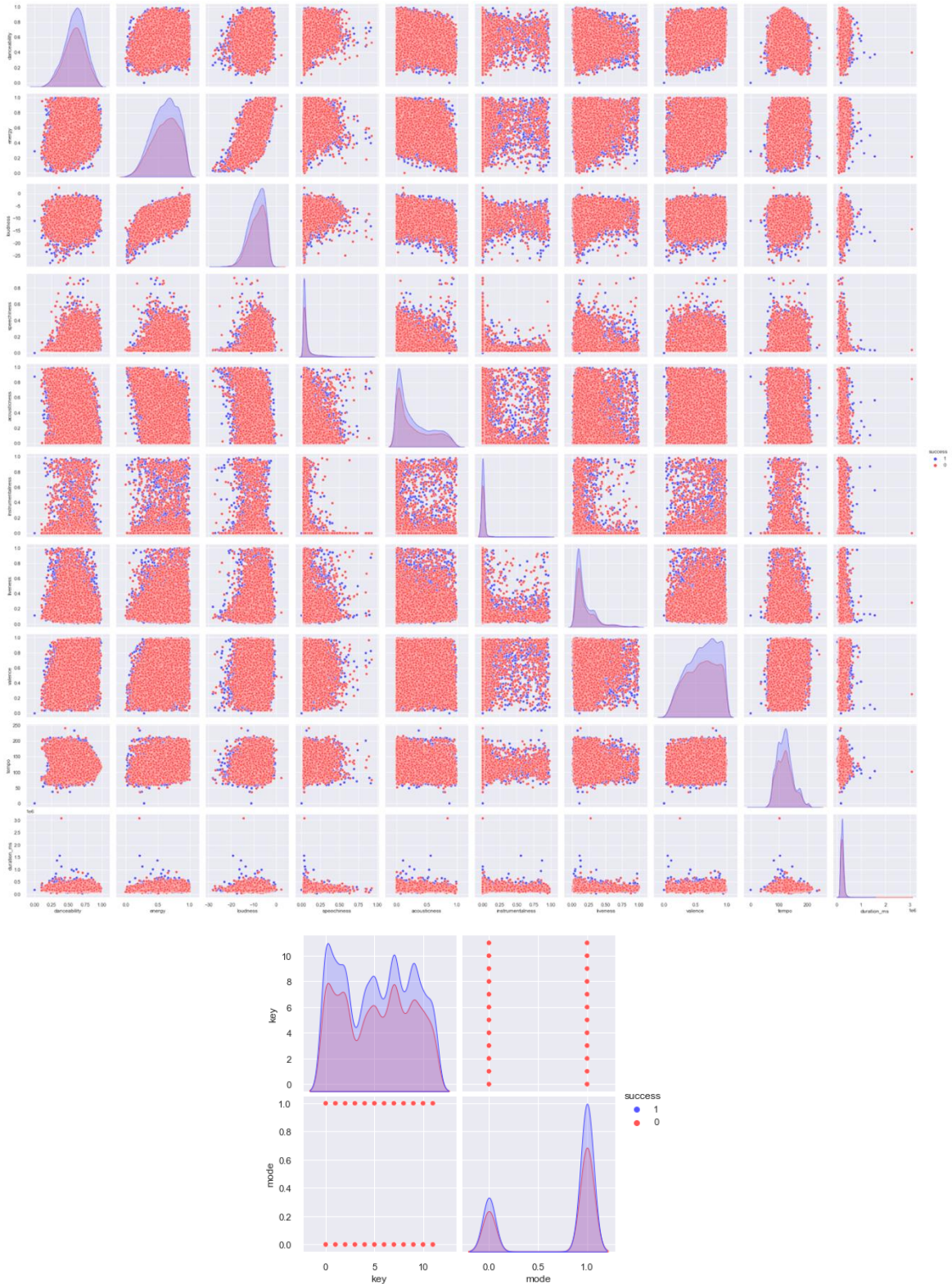


Figure 2. Numeric and Categorical Music Features Correlations



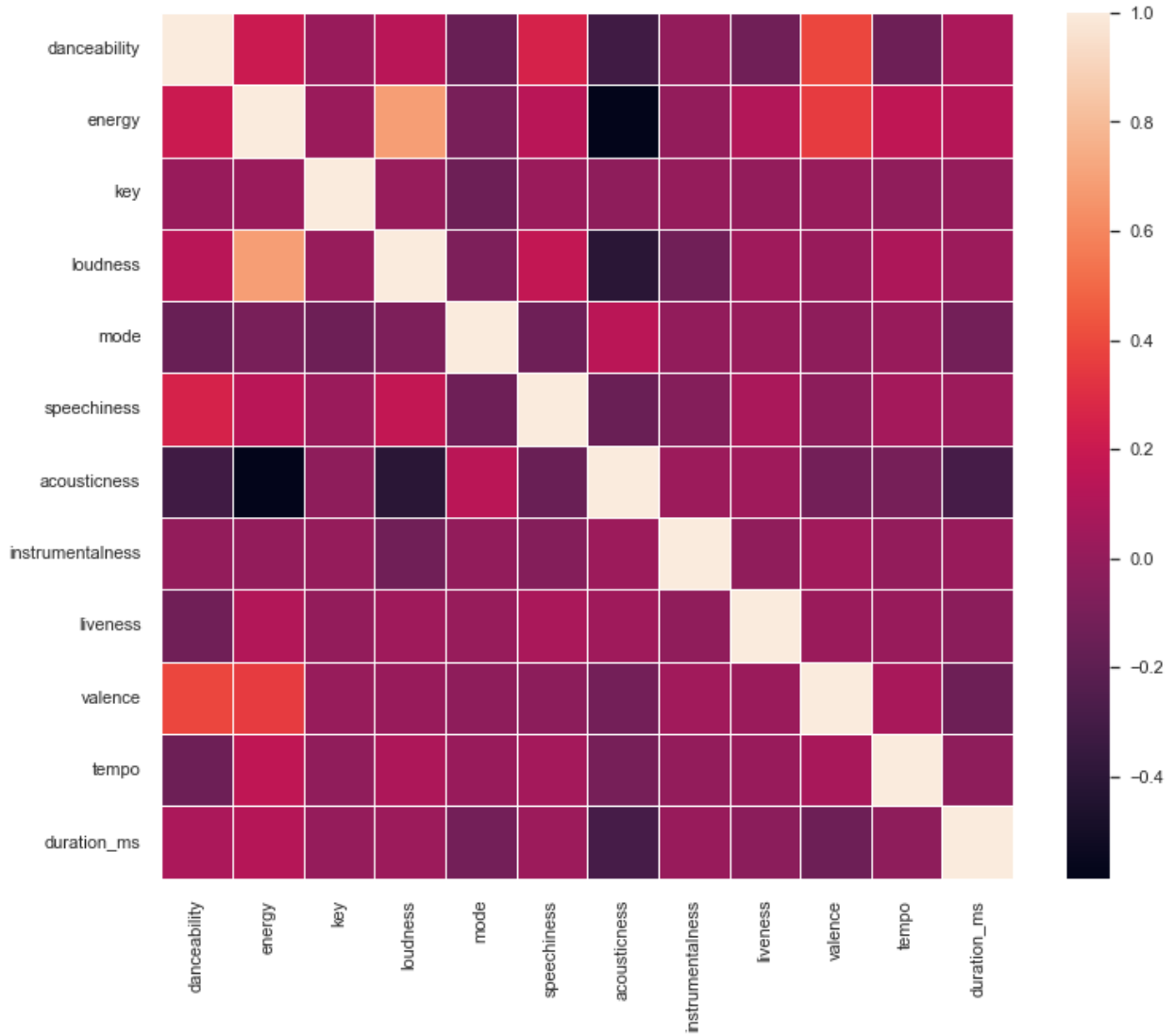


Figure 3. POS Tagging and Frequency Distribution of Genres Clusters

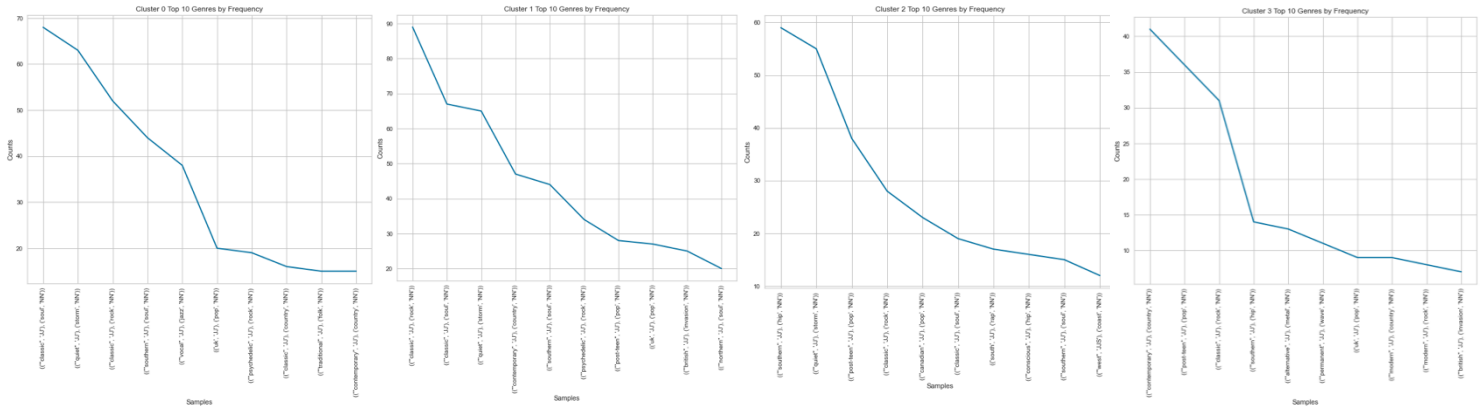
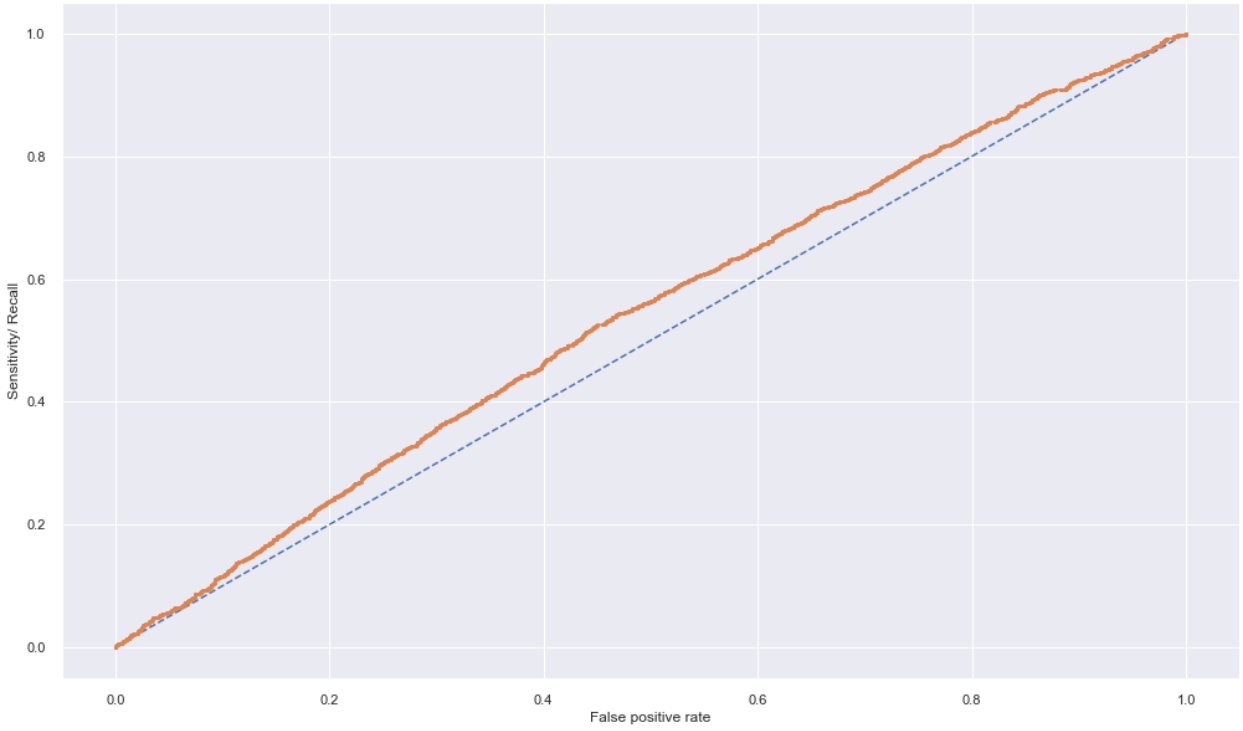


Figure 4. Logistic Regression



AUC - Test Set: 54.04%

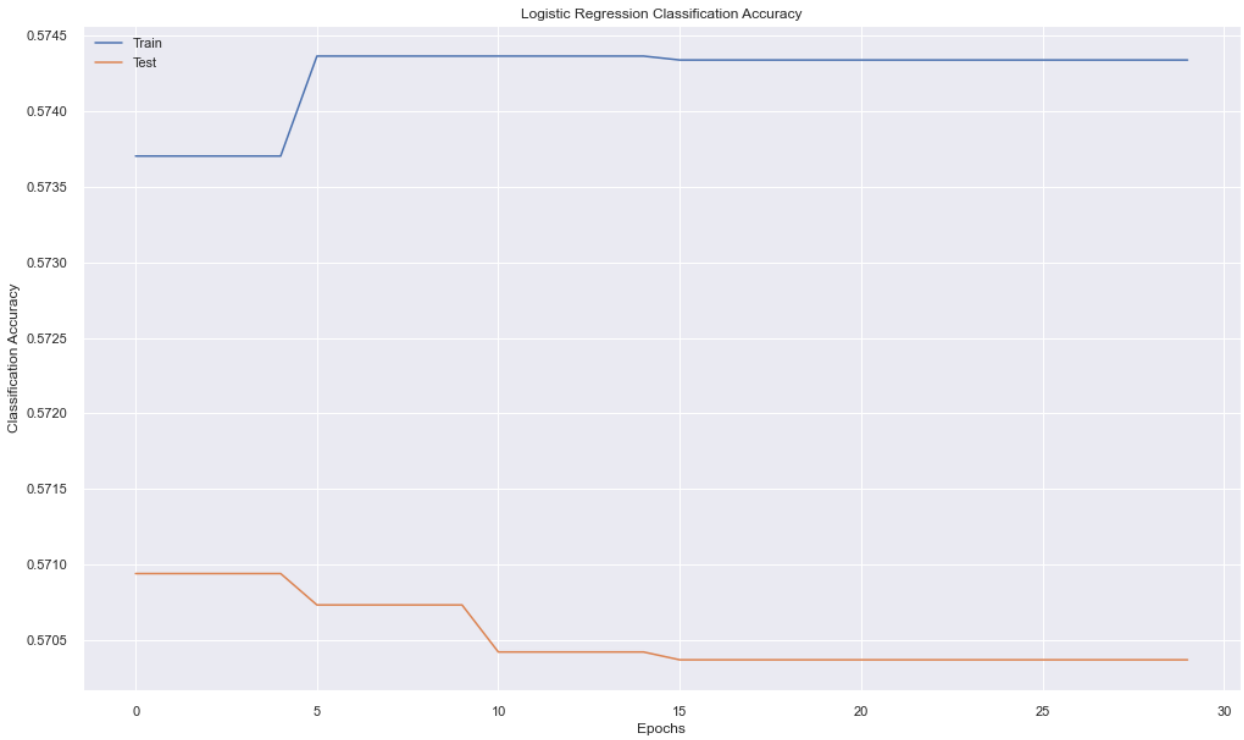
Logloss: 14.79

best params: {'classifier__C': 0.01, 'classifier__max_iter': 100}

best score: 0.571

accuracy score: 0.572

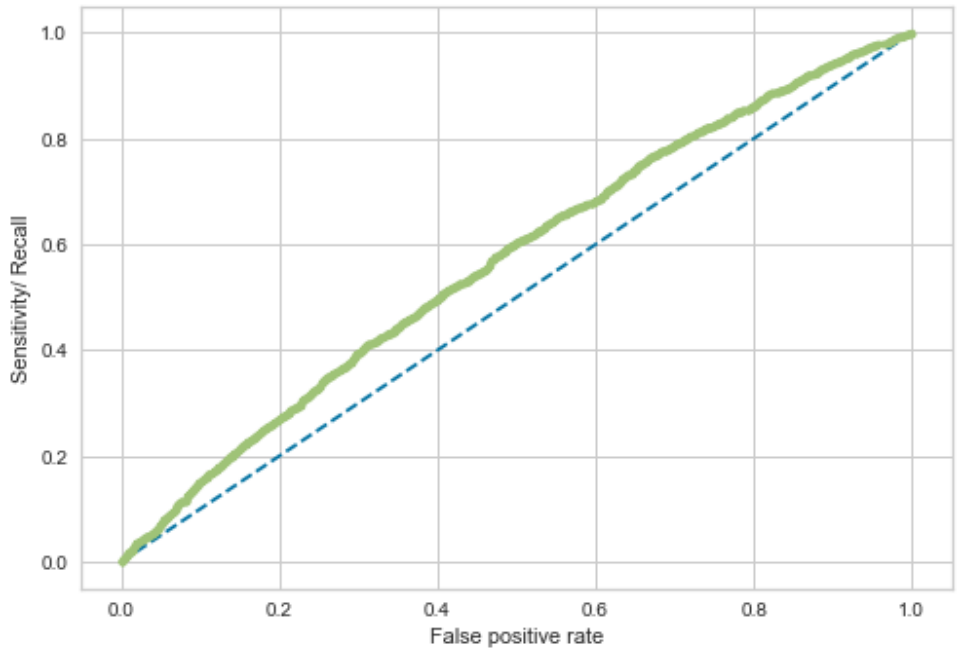
	precision	recall	f1-score	support
0	0.49	0.11	0.18	2053
1	0.58	0.92	0.71	2762
accuracy			0.57	4815
macro avg	0.54	0.51	0.44	4815
weighted avg	0.54	0.57	0.48	4815



Train set mean accuracy 0.574

Test set mean accuracy 0.571

Figure 5. XGBoost



AUC - Test Set: 56.98%

Logloss: 14.41

best params: {'classifier__colsample_bytree': 1, 'classifier__gamma': 1, 'classifier__learning_rate': 0.01, 'classifier__max_depth': 3, 'classifier__n_estimators': 1000, 'classifier__subsample': 1}

best score: 0.573

accuracy score: 0.583

	precision	recall	f1-score	support
0	0.52	0.15	0.23	2038
1	0.59	0.90	0.71	2777
accuracy			0.58	4815
macro avg	0.56	0.53	0.47	4815
weighted avg	0.56	0.58	0.51	4815

USING ORANGE TO PREDICT HOT SONGS

Table 1. Best Performance of Models Using Orange

Model	**AUC**	**CA**	**F1**	**Precision**	**Recall**
kNN	0.520	0.541	0.537	0.534	0.541
Tree	0.529	0.524	0.527	0.536	0.524
SVM	0.467	0.547	0.548	0.549	0.547
Random Forest	0.554	0.568	0.559	0.558	0.568
Naive Bayes	0.543	0.567	0.540	0.546	0.567
Gradient Boosting	0.551	0.563	0.557	0.555	0.563
Neural Network	0.539	0.539	0.536	0.534	0.539
Logistic Regression	0.546	0.565	0.520	0.536	0.565

The best performing model - Random Forest, had an accuracy of 55.9%. kNN was set to 5 nearest neighbors measured by Euclidean distance. The Manhattan distance produced the same F1 score. SVM using Polynomial kernel and SVM type has a lower F1 of 48.1%. Tuning the model to using v-SVM (regression cost = 1.00, complexity bound = 0.50) and Polynomial kernel (g = auto, c = 1.00, d = 3.0) created a 54.1% accuracy. Changing the c in the kernel to 1.10 created the highest accuracy of 54.8%. Random Forest had the best accuracy when setting the number of trees equal to 10. Gradient Boosting from scikit-learn had an accuracy of 54.2%. Using Extreme Gradient Boosting Random Forest (xgboost) has an accuracy of 55.0%. After changing the regularization from lambda = 1.0 to lambda = 0.5, the F1 score increased slightly to 55.7%. The limit depth of individual trees was 6. All subsamplings equal to 1.00. The neural network parameters were set to two hidden layers: 50 and 20 neurons. The activation was ReLu and the Solver was using Adam. Logistic Regression was set to Ridge (L2) and strength was C = 6 that had the highest accuracy.

APPENDIX D – DATASET DESCRIPTION IN TASK 1 AND 2

The size of dataset used in prediction(task 1 and 2) is 7933 rows(since 2000), with the following columns:

Table 2. Information of dataset in tasks

task	target	features				
		Numbers of variables used				
		Music features	Years (dummies)	duration	Spotify Popularity	Genres
Task 1	Ranks(2 labels)	15	4	1	1	882
Task 2	Classes(3 labels)	15	4	1	1	882