BILLBOARD RANKINGS ANALYSIS USING MACHINE LEARNING TECHNIQUES ON SPOTIFY

By

Yijing Tan

A Capstone Project

December 2020

ABSTRACT OF THE DISSERTATION

Billboard Rankings Analysis Using
Machine Learning Techniques On Spotify

by YIJING TAN

In this study, we propose a machine learning approach to explore how various attributes defined by Spotify for each track affect their ranking positions on Billboard year-end charts and to predict the audience's music preference. 5 main machine learning techniques were applied under an efficient process that performs concurrent variables and model selection. The logistic regression achieved the best results among all of the predictive models. Besides, according to this information, it can be good to analyze target marketing by modeling consumer musical taste.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

With the rise of applications like Spotify, Apple Music, Amazon music, etc., users who rely on streaming music services have contributed a great deal to the music industry (Wloemert, 2016). Music producers and labels have noticed the gigantic worth of streaming data on the music industry. While how could they get a clear picture of the audience's taste from a decent amount of data with the rising streaming platforms' help? Artists may want the single they produced is a hit on the market and strategically build their listenership. Especially for nameless artists, the songs precisely designed for music charts earn a chance to appeal interests from investors (Edvardsson, 2019). However, are there any standards or hidden rules that can track the market's preference to give artists useful hints? With close to 40,000 tracks being uploaded to Spotify every day according to data from Spotify founder Daniel Ek on Apr. 4th of 2019, how can artists seek breakthroughs in increasingly fierce competition and enhance their fanbase?

One clustering system on attributes of tracks has already been suggested in Thomes (2011). However, in the current work, we perform a deeper analysis of the predictive capabilities of machine learning models.

Understanding what makes high-ranking tracks popular could hugely impact decision-making for the music business. 'Ranking' is a direct and excellent indicator when making a diagnosis of music consumers' penchant. As such, by getting these interesting song attributes like 'Danceability', 'Energy', 'Tempo' and so on, artists could get the idea that should they focus more on certain attributes than others from statistical analysis of the streaming data.

Accordingly, in this project, we applied predictive algorithms for over 900 tracks which are listed on most recent 10 Billboard year-end charts extracted by

Spotify Web API, in order to analyze the popularity of music based on their Billboard rankings, and produce a predictive model for track's popularity. To be more specific, the aim of this project is to predict the rankings of tracks given a set of features as inputs. These features are 'danceability', 'energy','key',' loudness', 'mode', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo', 'duration_ms', 'time_signature', and 'popularity' taken from Spotify Web API. For the output, the variable is the level of the track's ranking. We have levels being one of these values: [1, 2, 3, 4, 5]. The lower the value the higher the ranking. On top of testing our algorithms for prediction purposes, we would like to analyze the dataset as well aiming to answer questions like which some song attributes have a significant effect on the song rankings than other attributes?

In this paper, we will explain the steps to establish and to compare the models based on the machine learning methods for predicting the rankings of songs. Particularly, we will treat each level of the tracks separately and their aim is to be able and find an appropriate decision boundary that works well for new data.

CHAPTER 2

DATA EXTRACTION AND TRANSFORMATION

2.1 Data Source

Spotify is a streaming music and podcasts service platform, available in iOS, Android, Windows, and Linux system and etc. It has 50 million songs and claims a leading 35% market share on the global music streaming services market of 2019, more than any other streaming service provider like Amazon music and Apple music. Thus, the dataset we captured from Spotify will truly reflect a picture of the current music industry. Furthermore, it also provides friendly access for developers.

'Billboard Hot 100' is a weekly music popularity chart that contains 100 tracks with their rankings computed by a combination of sales, airplay streaming data and etc., produced by Billboard. Billboard will work out a year-end chart based on the 'Billboard Hot 100' of each week at the end of each year. Therefore, it is considered to be the most authoritative music chart that can reflect the US music choice. The selected charts are from 2010 to 2020, a total of 11 in the count to follow late-year music trend.

Here we will retrieve these annual music charts data from Spotify Web API https://www.spotify.com/us/.

2.2 Dataset Description

In our crawled music data, there should be 11years*100songs, 1100 tracks as row data, and 14 features ('danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo', 'duration_ms', 'time_signature', 'popularity') as columns plus 5 columns, including 'id', 'uri', 'artists names', and 'song_name,' describe every track's information. However, our search result is not perfect because, on Spotify, there's still some music lacking

permissions from rights holders. Besides, to avoid over-usage, Spotify API limits the number of service calls that the developer's application to Spotify Platform. Consequently, there are total 1093 rows and 20 features that exclude the class label in this project's dataset.

In more detail,

KEY	VALUE TYPE	VALUE DESCRIPTION
duration_ms	float	The track duration in milliseconds.
key	float	0 stands for C, 1 stands for C♯/D♭, 2 stands for D and so on. Especially if no key is detected, the value will be -1.
mode	int	The modality (major or minor) of a track, the value is only 0 and 1. 0 means low repetition, 1 means high repetition.
time_signature	float	Measure how many beats are in each bar.
acousticness	float	The degree of original sound, the value is between 0 and 1, representing the degree of non-electronic sound contained in the track.
danceability	float	Rhythmicity, the value is between 0 and 1.
energy	float	The value is between 0 and 1, representing the perception of music intensity and activity. The higher value indicates the faster, louder and nosier the music sounds.
instrumentalness	float	The proportion of no vocals part in one track. The value is between 0 and 1, where 1 means absolute music.
liveness	float	The value is between 0 and 1. A value greater than 0.8 indicates the high possibility of the track is live recording.
loudness	float	The loudness of a track in decibels.
speechiness	float	The ratio of spoken words in every track and 0 means no spoken words.
valence	float	Psychological feeling, a measure of sad (0.0) to happy (1.0). The higher the value, the more positive the music feels.
tempo	float	Beats per minute.
id	string	The Spotify ID for each artist.
uri	string	The Spotify URI for each artist.
artists names	string	Singer.
song_name	string	Track title.
popularity	int	The artist's popularity is calculated from the popularity of all the artist's tracks.The value is between 0 and 100, where small number means 'not at all popular' and 100 is 'very hot.

ranking	int	Popularity of the song under Billboard rules.

Table 2.1 Summary of 14 features and response variable

'danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo', 'duration_ms', 'time_signature', 'popularity' are musical properties that depict a track.

'id', 'uri', 'artists names', and 'song_name' describe a track's content information.

'ranking' is a major indicator of song popularity and later used for correlation and data training in this project. It reflects "hotness" by today's music listeners to the track in the charts.

CHAPTER 3

FEATURE ENGINEERING AND DATA ANALYSIS

Feature engineering is thought to be key to success in applied machine learning (Scott Locklin,2014). Therefore, our task is to obtain better train data from raw dataset before feeding the reduced dataset to the modeling algorithms in this section. In other words, we need to pick a set of most statistically significant feature subsets from the original data set.

	id	song_name	artists names	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	duration_ms	time_signature	popularity	ranking
count	888	1093	1093	888.000000	888.000000	888.000000	888.000000	888.000000	888.000000	888.000000	888.000000	888.000000	888.000000	888.000000	888.000000	888.000000	1093.000000	1093.000000
unique	857	990	650	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
top	4pLwZjnHj3SimlyN9SnOz	Dynamite	Drake	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
freq	2	3	16	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
mean	NaN	NaN	NaN	0.666886	0.676983	5.415541	-5.756955	0.623874	0.106002	0.147205	0.004023	0.177014	0.510164	122.737676	219487.802928	3.989865	52.946935	50.193047
std	NaN	NaN	NaN	0.136429	0.157626	3.640816	2.698557	0.484685	0.101044	0.190801	0.035565	0.131787	0.216491	28.355879	36597.586996	0.270514	31.929996	28.714495
min	NaN	NaN	NaN	0.000000	0.000000	0.000000	-60.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	78200.000000	0.000000	0.000000	1.000000
25%	NaN	NaN	NaN	0.586750	0.578000	2.000000	-6.737500	0.000000	0.041475	0.018300	0.000000	0.093975	0.344750	99.966000	198659.750000	4.000000	22.000000	25.000000
50%	NaN	NaN	NaN	0.677000	0.696000	6.000000	-5.413000	1.000000	0.060000	0.067950	0.000000	0.124500	0.505000	123.454500	217753.500000	4.000000	68.000000	50.000000
75%	NaN	NaN	NaN	0.755500	0.797000	8.250000	-4.324500	1.000000	0.126500	0.203000	0.000008	0.212000	0.684000	139.983000	237653.750000	4.000000	77.000000	75.000000
max	NaN	NaN	NaN	0.970000	0.972000	11.000000	-1.190000	1.000000	0.592000	0.978000	0.680000	0.833000	0.966000	205.932000	484147.000000	5.000000	100.000000	100.000000

Table 3.1 Numerical Features

The Table 3.1 above is a descriptive statistics which summarize metrics like the central tendency, dispersion and shape of a dataset's distribution. 'count' indicates the number of observations for each attribute that corresponds to the number of tracks. 'std' denotes the standard deviation of each attribute group, from which we can guess the degree of data dispersion around the average.

It is also understood from the Table 3.1that,

- Drake is the most favor artist of the most 10 years on the Billboards. He appears 16 times in total in our search results.
- 'Dynamite' shows up 3 times as song title.
- The longest track has 484147 milliseconds which is about 8.07minutes.

Not all columns are useful in their raw form. The data showed in the Table 3.1 reveals that some raw data samples contain null values that cannot be utilized to build

predictive modeling. In a simple way, we remove observations that contain NaN (Not-A-Number) values and convert all categorical features into numeric.

All the listed attributes in Table 2.1 mainly characteristics of songs. In fact, some of them may be useless, one example is the ‘id’ attribute, for model building. The preliminary features taken into account are 'danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo', 'duration_ms', 'time_signature', 'popularity', 'ranking' that describe musical properties of tracks. Other features 'id', 'uri', 'artists names', and 'song_name' are string types that describe tracks' contents will be eliminated.

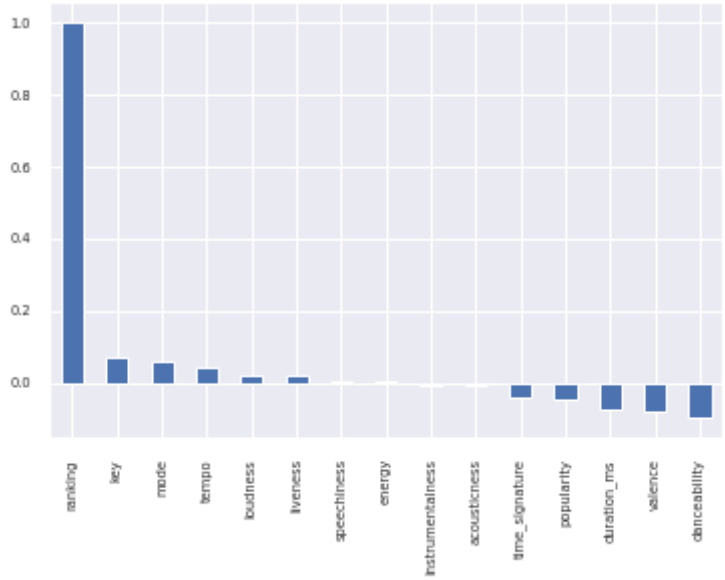


Figure 3.1 Correlation between ‘ranking’ and other features

Based on the above correlation analysis, we will drop features named 'speechiness', 'acousticness', ‘energy’, and ‘instrumentalness’, which has little connection with the target ‘ranking’.

Although we have eliminated 4 attributes, this data set still has many different features, and it is important to understand the relationship between them to analyze the dataset better. For that reason, a correlation map helps to understand these

relations in a single representation thoroughly. A correlation map is made by calculating the covariance of each feature with respect to others.

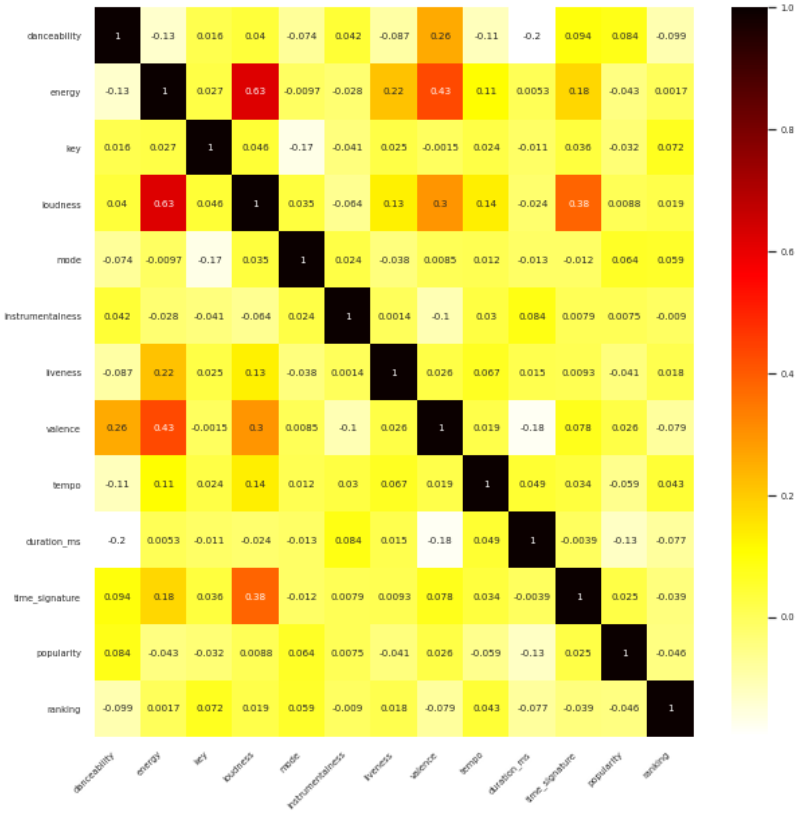


Figure 3.2 Correlation heatmap of attributes

The lower the value, the higher the ranking position. Hence, according to Figure 3.2, ‘key’, ‘tempo’, and ‘mode’ have a more obvious negative correlation with the user's preferences. In contrast, ‘popularity’, ‘danceability’, ‘valence’, ‘duration_ms’ and ‘time_signature’ have a positive correlation with the song ranking.

Meanwhile, because each chart has a maximum range of rankings from 1 to 100, we group the target feature into 5 groups, a rating scale of 1-5, where 1 is 'not at all popular' and 10 is 'very hot. Each group has 20 rankings. For example, from 1 to 20 will be assigned as group 1.

‘danceability’ and ‘valence’ are the top two features related to the target variable ‘ranking’. Let us make a careful study of them.

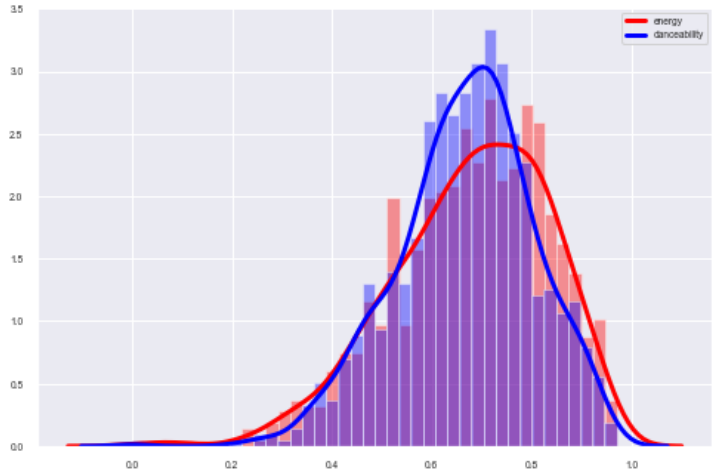


Figure 3.3 Distributions of 'energy' and 'danceability' values

The histogram distributions in Figure 3.3 display a descriptive overview of the attributes from songs that are on the Billboard charts. By visualizing distributions of attributes 'energy' in red line and 'danceability' in blue line, we find out that their distributions are similar.

Level	energy	danceability
1	0.6868172043010752	0.6875698924731185
2	0.6697430167597765	0.6781005586592175
3	0.6690335195530724	0.6578212290502794
4	0.6767410112359554	0.6495168539325843
5	0.6826024096385538	0.6600180722891565
mean	0.6769829954954937	0.6668862612612603

Table 3.2 Means of 'energy' and 'danceability'

It is shown in Table 3.2 that as the ranking decrease, the 'danceability' would diminish except for level 5. Conversely, there seems no regularity in 'energy'.

Here are some other aspects of insights we've acquired based on above attributes analysis:

- The tempo of high-ranked tracks has lower tempo 118 BPM which is closer to Hip Hop (around 80-115 BPM), compared with low-ranked tracks.

- Average high ranked song duration is 219487.8 milliseconds or 3' 66" minutes, somewhat longer than the average radio song, probably due to a few outliers (“Mirror” — 8' 07'', “Get Lucky (feat. Pharrell Williams & Nile Rodgers)”— 6' 16'').

After filtering out less valuable attributes, we get a table for modeling with 888 samples and 10 features that exclude the target variable.

The cleaned data include:

- General numeric features (e.g. popularity)
- Numeric physical properties (e.g. loudness, duration)
- Critical features (Track Ranking)

Principle component analysis (PCA) is a classic method to extract practical features by reducing dimension. It allows us to combine existing attributes into a new data frame consisting of a reduced number of attributes by utilizing the variance in the data. The attributes which "explain" the highest amount of variance in the data form the first few principal components and we can ignore the rest of the attributes if data dimensionality is a problem from a computational standpoint.

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
0	-1.988096	1.404019	0.936115	0.958156	1.500792	0.712481	0.089334	-0.440755	-1.328236
1	-1.177783	1.163384	-1.776084	0.496373	0.061664	-0.871065	0.771382	-0.398346	-0.219924
2	0.636223	0.784102	-0.561871	-0.611344	0.936836	0.903451	-1.112035	-0.086928	-0.592930
3	0.634054	0.896920	-1.570437	-0.550326	-0.040383	-0.204478	1.362640	0.975825	0.853935
4	0.590168	-0.228312	-1.998022	0.081310	1.074263	-1.574916	0.953356	-0.701719	-0.314980
...	...	...	...	...	...	...	...	...	...
883	1.212602	0.341209	-1.376241	-1.061501	0.523308	1.116318	-0.098611	-0.744100	-0.109942
884	-1.027167	0.773348	-1.501222	-0.065239	-0.765968	-0.245130	0.128865	0.362387	-0.849767
885	0.549792	-0.581168	-0.697388	0.315098	1.376433	2.882563	-1.610161	-0.616785	-0.887646
886	-0.656357	-1.101090	-1.163410	1.553472	1.006629	0.477269	-0.138843	-0.077518	-0.151141
887	-0.306254	-1.093817	0.325151	-0.145999	-1.378489	-1.606878	-0.684901	0.193423	1.361942

Table 3.3 PCA with 9 components

Table 3.3 is a data table created by PCA method, it marks born of new 9 features.

CHAPTER 4

PREDICTIVE MODELING

So far, we extract 799 observations, account for 90% of the processed dataset, to create the testing set. The rest 10% will be unseen data to test the performance of our predictive models. To avoid overfitting and acquire representative data in each part, the process of splitting the training and test dataset is random.

The diverse classification algorithms we will use in the project are as follows: Logistic Regression classifier, Random Forest classifier, Decision Tree classifier, K-Nearest Neighbors, Support Vector Machine classifier.

In this part, the accuracy score, a classification metric, will be employed to quantify every model's quality when they do the prediction for the test data. If n represents the number of the predicted samples, y_i means true values of labels, and $\widehat{y_i}$ corresponds to the samples' predicted values. The score is the fraction of correctly classified cases as below:

$$accuracy(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} 1(\widehat{y_i} = y_i)$$

4.1 Logistic Regression

The logistic regression analyzes datasets in which there are one or more independent variables that determine an outcome. The logistic expression is given in:

$$f = \frac{1}{1 + e^{-z}}, z = W^T \psi + \mathrm{b}$$

where ψ is input feature vector (in our case, it is the different statistical features extracted from the clustering process) and W is the weights vector of the linear combination that is estimated on the training set (Cohen, 2017).

- Logistic Regression gave us an accuracy of accuracy of 26.9663%

4.2 Decision Tree

In the decision tree structures, leaves represent classifications (also referred to as labels) where for each leaf a unique class is attached, split nodes like branches represent conjunctions of features that lead to the classifications (Wu, 2008).

First, we will construct a decision tree along with the whole training set. Subsequently, to classify the test set, we evaluate the relative attributes and take the branch corresponding to the test's outcome. Repeat this process until a leaf is encountered, and then, the new object is classified to the class labeling the leaf.

- Accuracy using Decision Tree is 20.2247 %

4.3 Random Forest

A single decision tree has overfitting problems as a tree grows deeper and deeper until the data is separated. This will reduce the training error but potentially results in a larger test error. To address this issue, the random forest constructing multiple decision trees. Each decision tree uses a randomly selected subset of training data and features. The output is calculated by averaging the individual decision tree predictions to get a more accurate and stable prediction.

The random forest algorithm is also easy to measure the relative importance of each feature on the prediction. The feature importances show which features likely to drop because they contribute little to the predicting process. We use tools in the Sklearn package to measures every attribute’s importance. Below is a table visualizing the importance of the 10 features.

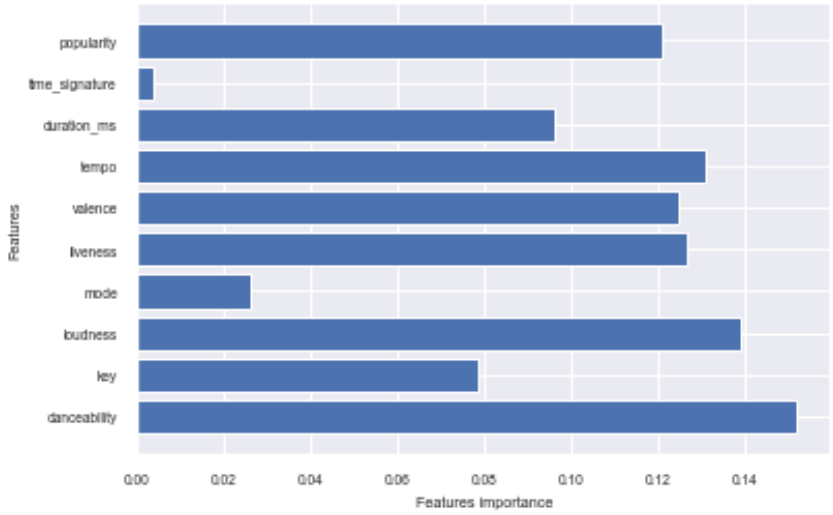


Figure 4.1 Feature importance

Which features are most predictive? The importance-weight bar list of Figure 4.1 gives the answer that the 'popularity' and 'loudness' dominates than all other features, followed by 'tempo', 'liveness', 'valence' and 'popularity'.

- Random Forest comes out to be 24.7191 %

4.4 K-Nearest Neighbors (k-NN)

The K-Nearest Neighbors algorithm (k-NN) is a classifying method proposed by Thomas Cover. The theory in k-NN is that it assigns a predicted value to a new observation based on the plurality or mean (sometimes weighted) of its k "Nearest Neighbors" in the training set (Korn, 2005), like K-Means based on closest training examples in the feature space. In this project, we will apply this instance-based learning with Euclidean distance as a metric to our data.

This approach gave a slightly worse accuracy than the previous classifiers.

- K-Nearest Neighbors gave us an accuracy of 17.9775 %

4.5 Support Vector Machines (SVM)

The support vector machines (SVM) have been used in classification modeling by capturing complex relationships between the data points.

In this case, the objective of implementing the support vector machine algorithm is to find a hyperplane that has the maximum margin in a 10-dimensional space (10 is the number of sample features) that distinctly classifies the data points. Data points falling on either side of this hyperplane can be attributed to different 5 levels.

- Support vector machines (SVM) gave us an accuracy of 19.1011 %

A summary of the above models' performance is presented in Table 4.1. Compared with other machine learning methods, the logistic regression model using is 0.269663, which is better than for all other classification algorithms.

Predictive Model	**Accuracy Score**
Logistic Regression	26.9663%
Decision Tree	20.2247 %
Random Forest	24.7191 %
K-Nearest Neighbors (k-NN)	17.9775 %
Support Vector Machine	19.1011 %
AdaBoostClassifier	18.0 %
GradientBoostingClassifier	19.1 %
XGBoostsClassifier	24.7 %

Table 4.1 Accuracy scores of predictive models

Thus, the logistic regression with simple expressions and interpretable parameters is suitable for the rapid prediction of predicting applications in predicting rankings. In the next subsection 5, we will detailed describe the logistic regression classifier.

CHAPTER 5

VISUALIZATION

Below is the basic logistic function for the 10 features used in the modeling. The denominator looks similar to linear regression with an intercept, coefficient, and error value.

$$p = \frac{1}{1+\exp\left(-(c_0+c_1x_1+c_2x_2+c_3x_3+c_4x_4+c_5x_5+c_6x_6+c_7x_7+c_8x_8+c_9x_9+c_{10}x_{10}+\varepsilon)\right)},$$

p represents target level in our modeling. For the 10 features x_1, x_2, x_3, …, x_{10} of the ith (i=1, 2, …, 888) track, assuming the corresponding Logistic regression value is p_i, then $p_i = p(y \geq i)(i = 1,2,3,4,5)$. Function $logit(p_i)$ will help to calculate the odds of p_i happens. The logistic regression model at the ith level can be expressed as:

$$logit(p_i) = ln\left(\frac{p_i}{1-p_i}\right) = c_{i0} + c_{i1}x_1 + c_{i2}x_2 + \cdots + c_{i10}x_{10} + \varepsilon_i \quad (i = 1,2,3,4,5)$$

Level	1	2	3	4	5
Intercept c_0	-1.3867433	-1.3287916	1.4354508	1.4229128	1.4867170
$x_1\_c_1$	0.1159878	0.1063491	-0.0154887	-0.0931831	-0.0695557
$x_2\_c_2$	0.0981259	-0.005561	0.0614930	-0.0146264	0.0820557
$x_3\_c_3$	0.0147904	-0.0718427	-0.0048691	-0.1001669	0.3328278
$x_4\_c_4$	0.0503505	0.0764487	0.0185116	0.0805406	0.0956974
$x_5\_c_5$	0.0314687	-0.0416121	0.0638624	0.0024102	-0.0126220
$x_6\_c_6$	0.2863994	0.0277767	0.1878567	-0.0695937	-0.0945582
$x_7\_c_7$	0.2083428	-0.0023898	0.1736935	0.1635236	-0.1382114
$x_8\_c_8$	0.2714087	0.0399761	0.0042549	-0.1728009	-0.1845668
$x_9\_c_9$	0.05556407	0.02698651	0.20488122	-0.06517582	-0.1567452

$x_{10\_}c_{10}$	0.1940775	-0.0173921	-0.1246473	-0.0412805	-0.0058248

Table 5.1 Coefficients c_{ik}

Table 5.1 shows the coefficients c_{ik} $(i = 1,2,3,4,5)$ of the 10 features x_k (k=1, 2, 3, …, 10) and intercept in each equation. Then, for each test record, the model will compute its probability $logit(p_i)$ 5 times using different levels to get the highest value as a result.

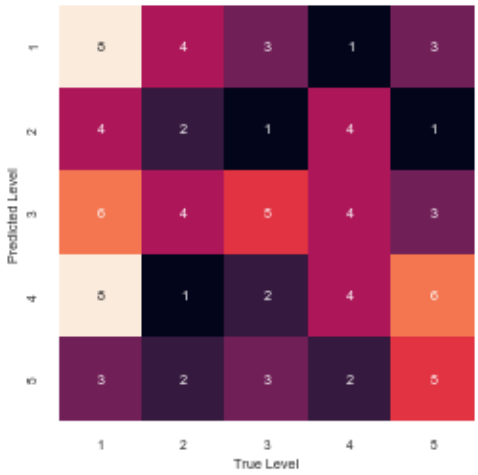


Figure 5.1 Confusion matrix of logistic regression

Figure 5.1 displays logistic regression model's confusion matrix, a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. The X-axis is labeled True label - meaning the original class labels for our test observations. The Y-axis is the predict levels and is labeled Predicted label - meaning our model's predictions given test data. If the confusion matrix above is named C. Then, $C_{i,j}$ will be the value on the ith row and jth column, equal to the number of observations known to be in group i and predicted to be in group j. Take $C_{2,2}$ and $C_{3,2}$which are 2 and 4 separately in the matrix as an example. $C_{2,2}$ indicates that there are 2 values are correctly grouped into right place, $C_{3,2}$ shows there are 4 samples which should be in level 3 but wrongly classified into level 2. The

table contains 4 different combinations of predicted and actual values. Thus, in this 5-class classification problem, the count of true positives is $C_{i,i}$ on diagonal.

CHAPTER 6

DISCUSSION

This study aims to predict the position of the track in the Billboard chart. Using the logistic regression predictive model above, we can estimate a track's popularity approximately. Furthermore,

- Artist's popularity has a significant influence on their high-ranked songs. However, it nearly has no impact on low-ranked positions which means individual artists could also have a great chance of being involved in the famous music charts.
- From the analysis on correlation in the rankings sector and main music attributes, we argue that data portability will affect competition within the EU, other features like 'speechiness', 'acousticness', 'energy', and 'instrumentalness' won't contribute so much. Therefore, we will suggest artists consider focus less on them when composing.

Analysis of music charts tells artists what genre they should consider when starting promoting a new single and that when to release a single that is likely to perform well on the charts. Simulate such a process can help target marketing by modeling consumer musical taste; it can help music producers make smart decisions on maintaining their stay on influential charts. Additionally, after the music player is familiar with listeners' preferences, it will recommend songs based on our patterns, which will give us a better user experience. Hoping to use these data to analyze keys factors that will influence the success of a song and predict how popular the songs will be in the future.

Like Avijit Datta said, while music is an art, it needs science to know just how many people across the world are enjoying the song. Machine learning methods permit us to relish songs in a numeric way.

APPENDIX

THE TABLES

Table 1. Values Of The 13 Attributes Ordered By ‘energy’

	danceability	energy	key	loudness	mode	liveness	valence	tempo	duration_ms	time_signature	popularity	ranking	levels
691	0.585	0.972	9.0	-4.450	0.0	0.0707	0.585	110.006	230253.0	4.0	73	100	5
810	0.508	0.957	11.0	-1.562	1.0	0.5460	0.830	173.555	276920.0	4.0	0	21	2
813	0.855	0.954	0.0	-1.190	0.0	0.2050	0.668	114.635	248133.0	5.0	79	24	2
453	0.624	0.953	4.0	-2.602	0.0	0.6570	0.729	149.992	211573.0	4.0	0	55	3
549	0.494	0.951	9.0	-4.237	1.0	0.3270	0.441	160.025	202496.0	4.0	72	93	5
50	0.572	0.949	4.0	-4.865	1.0	0.1630	0.530	118.974	194867.0	4.0	76	51	3
843	0.529	0.948	0.0	-3.527	1.0	0.2830	0.650	146.024	238000.0	4.0	65	54	3
565	0.457	0.948	10.0	-3.364	1.0	0.0536	0.878	148.000	208960.0	4.0	71	9	1
754	0.644	0.945	7.0	-3.534	1.0	0.1840	0.573	91.017	219560.0	4.0	55	63	4
742	0.586	0.945	7.0	-4.577	0.0	0.3970	0.483	155.953	283733.0	4.0	37	51	3

Table 2. Classification Report From k-NN Method

	precision	recall	f1-score	support
1	0.18	0.37	0.24	19
2	0.20	0.33	0.25	12
3	0.12	0.09	0.10	22
4	0.20	0.10	0.13	21
5	0.33	0.07	0.11	15
accuracy			0.18	89
macro avg	0.21	0.19	0.17	89
weighted avg	0.20	0.18	0.16	89

THE FIGURES

Figure 1. Distribution Of The 13 Attributes

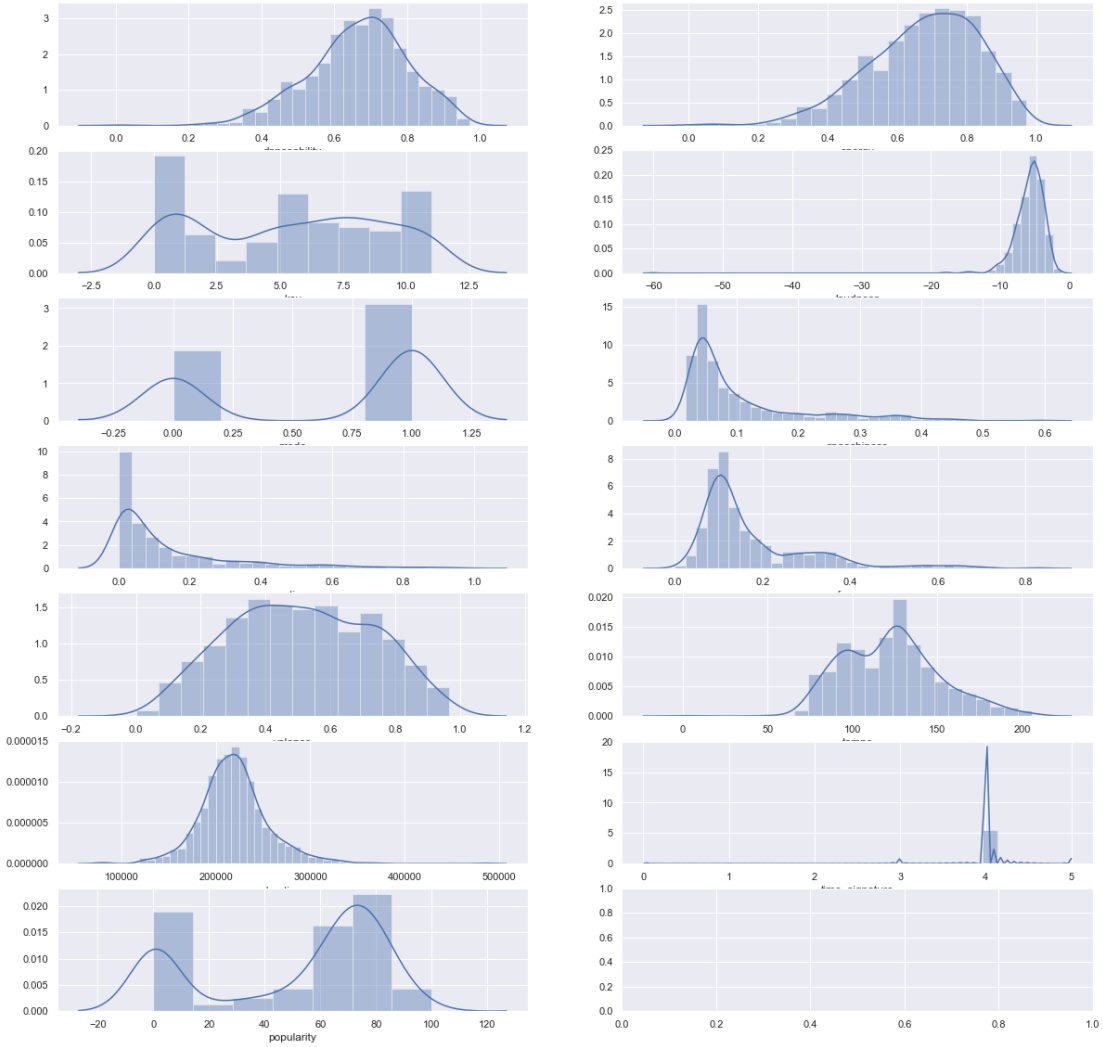


Figure 2. Features Processed After PCA

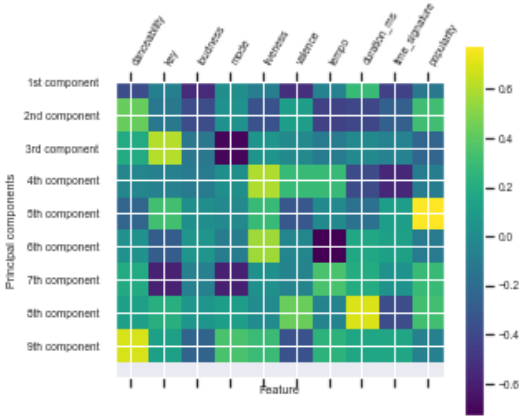


Figure 3. Decision tree Accuracy with different depths

Figure 4. k-NN Accuracy with different neighbors

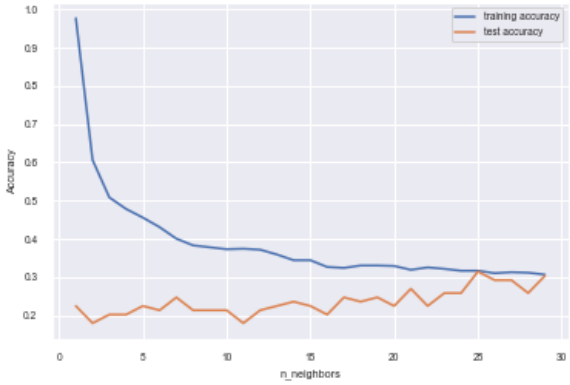
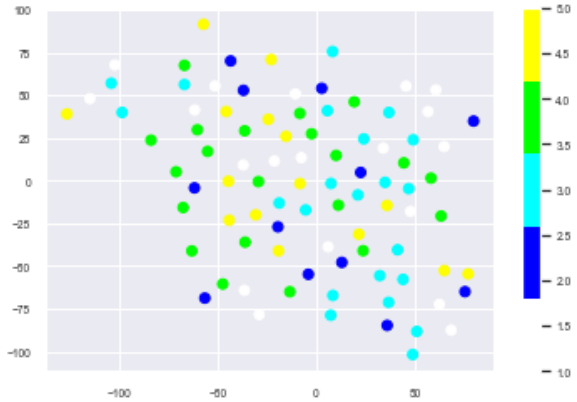


Figure 5. Projection Of The Dataset with 9 Features In Two-Dimensional Space



BIBLIOGRAPHY

Wloemert, N. , & Papies, D. . (2016). *On-demand streaming services and music industry revenues — insights from spotify's market entry. International Journal of Research in Marketing*, 33(2), 314-327.

Edvardsson, B. , & Tronvoll, B. . (2019). *How platforms foster service innovations. Organizational Dynamics.*

Thomes, T. P. . (2011). *An economic analysis of online streaming. how the music industry can generate revenues from cloud computing*. ZEW Discussion Papers.

Scott Locklin. (2014). *Neglected machine learning ideas*. (2014). Retrieved 30 November 2020, from https://scottlocklin.wordpress.com/2014/07/22/neglected-machine-learning-ideas/v

Cohen, Y. , & Lapidot, I. . (2020). *Speaker clustering quality estimation with logistic regression.* Computer Speech & Language, 65, 101139.

Wu, Xindong; Kumar, Vipin; Ross Quinlan, J.; Ghosh, Joydeep; Yang, Qiang; Motoda, Hiroshi; McLachlan, Geoffrey J.; Ng, Angus; Liu, Bing; Yu, Philip S.; Zhou, Zhi-Hua (2008-01-01). *"Top 10 algorithms in data mining". Knowledge and Information Systems*. 14 (1): 1–37.

Korn, J. , & Gould, J. . (2005). *Knnxvalidation documentation*