

DATA MINING ON ORANGE TOOL

Dataset:

Spotify Song Attributes Dataset(Kaggle)

1. Abstract

Today, Online music platform has become very much popular in streaming digital music all over the world. This project is done because it only aims to suggest songs to help people listen to the songs they love. The best thing about music is that it is the best solution to any problem or the best medicine in the world that nothing can relax you more than a pleasing song.

Nowadays, Spotify has become an important and popular online platform for providing digital music all over the world. A key feature of Spotify is that it provides access to any type of music like classical, western, rap of different languages onfrom anywhere in the world. A dataset consists of 2017 rows(songs) along with their audio features from Spotify is taken. Each row represents a song along with its audio features. The dataset contains 16 columns, that are name, artist, and "target" which is the label for the song. It is an attempt to build a model that can predict whether I like a song or not. Each song is labeled either "1" which means I like the song or "0" which means I don't like the song.

Here, we raise the question that is it possible to predict a track that I like or not based on its features(attributes) and also with the help of the major one, 'target'. I checked which type of algorithms could be used for a type of task. Different type of models were made using various algorithms such as Support Vector Machine(SVM), Gaussian Naive Bayes, K-Nearest Neighbour Algorithm(KNN), and Decision tree. Out of this, an accurate model is chosen to predict which all songs I like or not.

2. Introduction

Nowadays almost all the music platforms like amazon, Spotify, and music are using online platform for easy and faster access of songs for all the people around the world which makes these sites more benefit and become popular. They can access any type of songs of any languages and any type, but the issue is that there are a lot of songs present in it. So we need to classify these songs based on different their features like genres, language, etc. and the main aim is to classify these sets of songs as per the taste of the user so that we can attract our customers by providing various valuable services of their interests. So we are using various data mining techniques for this dataset and compared the results or output with one another to find the accurate algorithm that fit best for our model.

Spotify is one of the modern innovations to possess come to audio listening and experience with millions of subscribers. Though the service has become popular faster, it dominates Apple Music and Amazon music within the audio streaming market. From music, they need to extend the audio service to Podcasts, Audiobooks, and so on. Spotify Trends helps any content creator/musician to understand what type of music listeners prefer and how to compete in this immensely growing market.

The Spotify song dataset is taken from the Kaggle website. The data utilized in this was collected from Spotify's Web API. This is a computer algorithm that Spotify has which will estimate various aspects of the audio file. More info on various attributes utilized in the dataset is found on this Spotify Developer page. This study aims to estimate whether I like the song or not. Each song is labeled:

"1" (I prefer it) and "0" (I do not like) according to my taste. I used this data to check if I could make a model that could predict whether I like a song or not.

The data mining software, Orange is used here because it is easy to understand and use within the model and contains many methods. In this context, the dataset aims to develop an effective binary classification model by using it's relevant features and different types of models or algorithms. The results show that the proposed model successfully predicts whether the song we like or not.

3. Materials and Methods

3.1 Dataset

The dataset that is used, is taken from the Kaggle website. The dataset contains 2017 songs along with its audio features and based on these features I can analyze and find whether I like the song or not.

Data fields:

The key attributes present in each event within the data are:

No.	Attribute	Description	Type
1.	Key	Overall key of the music track	Numeric
2.	Mode	Modularity(major or minor) of a track	Numeric(minor-0,major-1)
3.	Accousticness	A confidence whether the track is acoustic	Numeric(measured from 0.0 to1.0)
4.	Danceability	How best is the track for dancing -like tempo, rhythm, beat strength, and overall regularity	Numeric (0.0-denotes least danceable and 1.0-denotes foremost danceable)
5.	Energy	Calculation of intensity and activity(loud & noisy)	Numeric

6.	Instrumentalness	Checks if a track contains neither vocals.	Numeric(closer to 1.0 no vocal content)
7.	Loudness	Loudness of a track	Numeric (ranges b/w -60 and 0dB)
8.	Valence	Musical positivity conveyed by a track	Numeric (measure from 0.0 to 1.0)
9.	Tempo	Estimate tempo of a track	Numeric (in BPM(beats per minute))
10.	Duration_ms	Duration of a track	Numeric (in ms)
11.	Liveness	Presence of an audience within the recording	Numeric (high value means high prob. that was performed live)
12.	Speechiness	Presence of spoken words during a track	Numeric
13.	Time_signature	Overall musical time signature of a track	Numeric
14.	Song_title	Title of the song/song name	Categorical
15.	Artist	Artist of the song	Categorical
16.	Target	Label of the song (like or not)	Categorical(1 or 0)

Table-1 – Short explanation of attributes

1	acousticness	danceability	duration	energy	instrumentalness	key	liveness	loudness	mode	speechiness	tempo	time_signature	valence	target	song_title	artist
2	0.0102	0.833	204600	0.434	0.0219	2	0.165	-8.795	1	0.431	150.06	4	0.286	1	Mask Off	Future
3	0.199	0.743	326933	0.359	0.00611	1	0.137	-10.401	1	0.0794	160.08	4	0.588	1	Redbone	Childish Gambino
4	0.0344	0.838	185707	0.412	0.000234	2	0.159	-7.148	1	0.289	75.044	4	0.173	1	Xanny Family	Future
5	0.604	0.494	199413	0.338	0.51	5	0.0922	-15.236	1	0.0261	86.468	4	0.23	1	Master Of None	Beach House
6	0.18	0.678	392893	0.561	0.512	5	0.439	-11.648	0	0.0694	174	4	0.904	1	Parallel Lines	Junior Boys
7	0.00479	0.804	251333	0.56	0	8	0.164	-6.682	1	0.185	85.023	4	0.264	1	Sneakin'™	Drake
8	0.0145	0.739	241400	0.472	7.27E-06	1	0.207	-11.204	1	0.156	80.03	4	0.308	1	Childs Play	Drake
9	0.0202	0.266	349667	0.348	0.664	10	0.16	-11.609	0	0.0371	144.15	4	0.393	1	GyÅngyhajÅ° lÅiny	Omega
10	0.0481	0.603	202853	0.944	0	11	0.342	-3.626	0	0.347	130.04	4	0.398	1	I've Seen Footage	Death Grips
11	0.00208	0.836	226840	0.603	0	7	0.571	-7.792	1	0.237	99.994	4	0.386	1	Digital Animal	Honey Claws
12	0.0572	0.525	358187	0.855	0.0143	5	0.649	-7.372	0	0.0548	111.95	3	0.524	1	Subways - In Flagranti Extended Edit	The Avalanches
13	0.0915	0.753	324880	0.748	0.00348	10	0.212	-8.62	1	0.0494	104.32	4	0.642	1	Donme Dolap - Baris K Edit	Modern Folk ÅceÅšjÅ°Å¼sÅ¼
14	0.253	0.603	356973	0.434	0.0619	0	0.108	-11.062	1	0.0342	127.68	4	0.381	1	Cemalim	Erkin Koray
15	0.366	0.762	243270	0.476	0	0	0.103	-12.686	1	0.114	130.01	4	0.367	1	One Night	Lil Yachty
16	0.44	0.662	247288	0.603	0	9	0.0972	-8.317	0	0.0793	125.01	4	0.351	1	Oh lala	PNL
17	0.019	0.637	188333	0.832	0.0563	6	0.316	-6.637	1	0.163	99.988	4	0.317	1	Char	Crystal Castles
18	0.0239	0.603	270827	0.955	0.0451	1	0.119	-4.111	1	0.0458	123.92	4	0.773	1	World In Motion	New Order
19	0.233	0.789	447907	0.659	0.00049	4	0.184	-12.654	0	0.0429	122.42	4	0.842	1	One Nation Under a Groove	Funkadelic
20	0.314	0.713	195429	0.611	0	1	0.117	-6.702	0	0.241	140.06	4	0.783	1	Bouncin	Chief Keef
21	0.0242	0.735	214347	0.759	0.185	1	0.0966	-6.914	0	0.0449	109.98	4	0.763	1	C O O L - Radio Edit	Le Youth
22	0.000702	0.854	249253	0.719	0.308	10	0.428	-9.335	0	0.0655	128.05	4	0.471	1	Percolator (Jamie Jones Vault Mix) - mixed	Cajmere
23	0.00024	0.747	307680	0.74	0.369	1	0.0995	-4.134	1	0.0323	130.03	4	0.77	1	House of Jealous Lovers	The Rapture
24	0.118	0.854	287086	0.401	0	9	0.527	-8.553	1	0.395	139.92	4	0.441	1	Imma Ride	Young Thug
25	0.000596	0.224	132760	0.925	1.35E-06	11	0.0663	-1.71	0	0.0834	138.02	4	0.364	1	Girlfriend	Ty Segall
26	0.279	0.512	203400	0.564	1.54E-05	10	0.133	-5.892	1	0.0316	94.498	4	0.401	1	If I Gave You My Love	Myron & E
27	0.00219	0.781	205160	0.795	0.269	7	0.0673	-6.758	1	0.036	109.98	4	0.795	1	This Ready Flesh	TR/ST
28	0.341	0.411	199500	0.684	1.42E-06	11	0.198	-6.889	0	0.383	110.02	4	0.598	1	Love My Mind	A-Trak

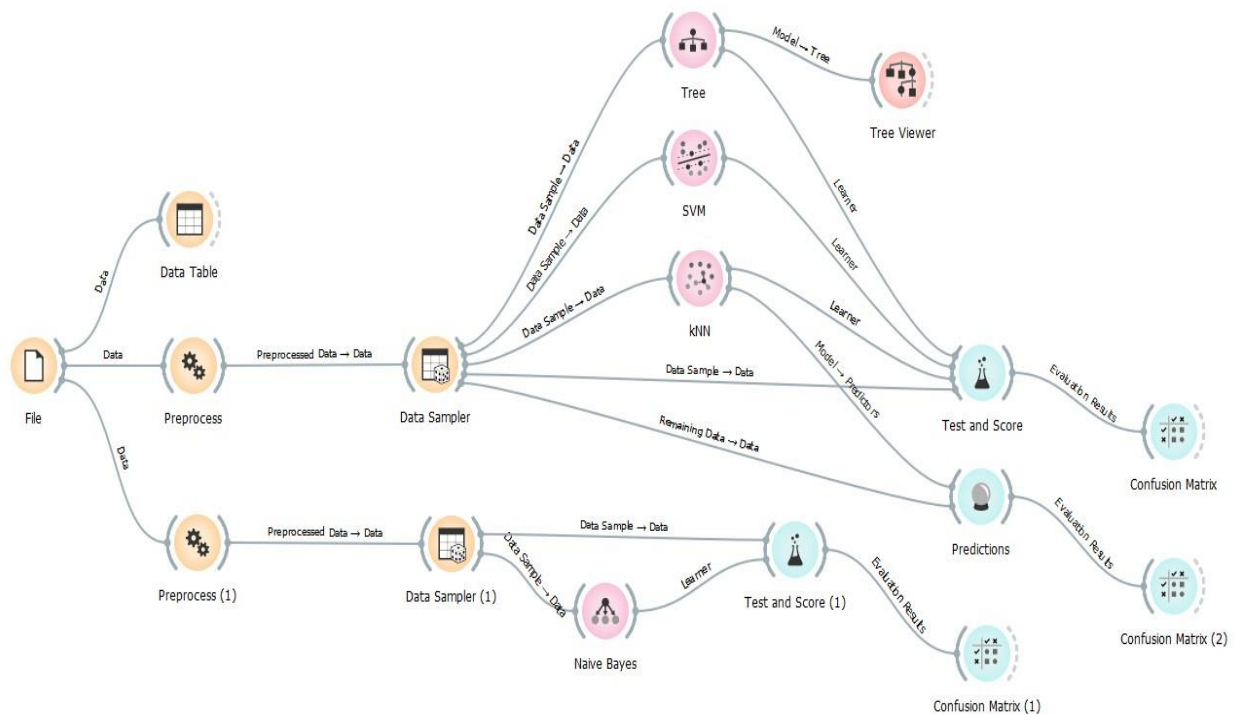
Orange Data Mining

ORANGE, a data mining software were developed by scientists at the University of Ljubljana in 1997 using various programming languages. This software, with the latest version presented with ORANGE 3.8.0 is used here for analysis of data by placing widgets.

The Widget is a user-friendly one that help users to read data, display the data in table and graphical format, feature selection, comparison, and visualization of data items. An important advantage of this program is the comparison of results of different types of algorithms or models.

Orange can be only used with a .tab extension dataset, but it can also use dataset extensions such as txt, basket, CSV, etc. In this context, an analysis of Spotify dataset related to the ORANGE program is done.

Orange Datamining on Spotify Song Attributes



Info

2017 instance(s)
14 feature(s) (no missing values)
Data has no target variable.
2 meta attribute(s)

	Name	Type	Role	Values
1	acousticness	N numeric	feature	
2	danceability	N numeric	feature	
3	duration_ms	N numeric	feature	
4	energy	N numeric	feature	
5	instrumentalness	N numeric	feature	
6	key	N numeric	feature	
7	liveness	N numeric	feature	
8	loudness	N numeric	feature	
9	mode	C categorical	feature	0, 1
10	speechiness	N numeric	feature	
11	tempo	N numeric	feature	
12	time_signature	N numeric	feature	
13	valence	N numeric	feature	
14	target	C categorical	target	0, 1
15	song_title	S text	meta	
16	artist	S text	meta	

3.2 Feature Selection Methods

Feature selection means selecting relevant features or attributes from the dataset that is used as input to the classification method for further analysis. When building a machine learning model in real life, it's almost rare that all the variables or attributes in the dataset are useful to build a model. Adding redundant variables reduces the generalization capacity of the model and may also reduce the overall accuracy of a classifier. Furthermore, adding more and more variables to a model increases the overall complexity of the model. Here, feature selection is one of the important steps while building a machine learning model. Its goal is to find the best possible set of features for building a machine learning model.

We use a technique known as information gain to identify relevant feature. Information gain calculates the reduction in entropy from the transformation of a dataset. It can be used for feature selection by evaluating the Information gain of each variable in the context of the target variable. Information theory measures information in bits. Information gain is the amount of information gained by knowing the value of the attribute.

Rank

The attribute ranking is completed by selecting relevant features based on the rank score of information gain, Gini index, etc. Using the rank score, I can reduce the number of data or features that are essential in further analysis.

Rank

Scoring Methods

- ☒ Information Gain
- ☒ Information Gain Ratio
- ☒ Gini Decrease
- ☐ ANOVA
- ☐ χ^2
- ☐ ReliefF
- ☐ FCBF

	#	Inf...ain	Gai...tio	Gini
<input checked="" type="checkbox"/> key		0.002	0.001	0.001
<input checked="" type="checkbox"/> liveness		0.002	0.001	0.001
<input checked="" type="checkbox"/> time_signature		0.003	0.007	0.002
<input checked="" type="checkbox"/> mode	2	0.004	0.004	0.003
<input checked="" type="checkbox"/> energy		0.004	0.002	0.003
<input checked="" type="checkbox"/> acousticness		0.007	0.004	0.005
<input checked="" type="checkbox"/> valence		0.008	0.004	0.005
<input checked="" type="checkbox"/> tempo		0.009	0.004	0.006
<input checked="" type="checkbox"/> speechiness		0.016	0.008	0.011
<input checked="" type="checkbox"/> duration_ms		0.027	0.013	0.018
<input checked="" type="checkbox"/> danceability		0.033	0.017	0.023
<input checked="" type="checkbox"/> loudness		0.044	0.022	0.030
<input checked="" type="checkbox"/> instru...alness		0.054	0.027	0.037

Here, I am selecting 7 attributes that have good score in information gain, gain ratio and Gini index in ascending order. Those attributes are:

Instrumentalness, Loudness, Danceability, Duration_ms, Speechiness, Tempo and valence.

Preprocessing

Using the above rank, we can select relevant attributes from the data that have high information gain, that is seven attributes: instrumentness, loudness, danceability, duration_ms, speechiness, temp, and valence which are appropriate for building the system and for more accurate analysis. Except for Naive Bayes model, information gain is useful for selecting relevant features because Naive Bayes model consider all attributes independently.

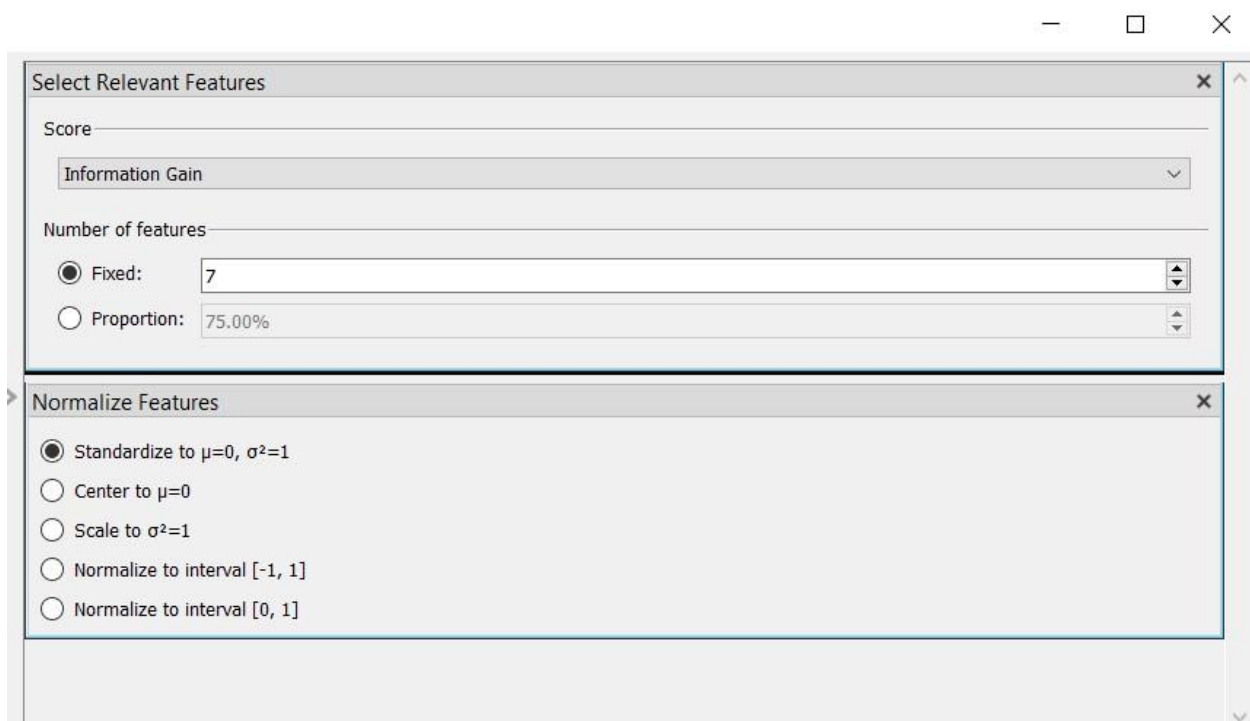
Relevant Feature Selection

Sometimes, on a dataset there can be some attributes that have no value on our analysis so to get accurate value, we can just ignore those attributes by selecting relevant attributes and do further process. So based on rank score, using the value of information gain, we select relevant features among all the attributes of the dataset so the further methods and calculation are much accurate.

Normalization

Normalization is generally required when we are dealing with attributes on a different scale, otherwise, it may lead to a dilution in effectiveness of an important equally important attribute (on lower scale) because of other attribute having values on larger scale. In simple words, when multiple attributes are there but attributes have values on different scales, this may lead to poor data models while performing data mining operations, so they are normalized to bring all the attributes on the same scale.

Distance algorithms like KNN, K-means, and SVM are most affected by the range of features. This is because behind the scenes they are using distances between data points to determine their similarity. If we have features with different scales, there is a chance that higher weightage is given to features with higher magnitude. This will impact the performance of the machine learning algorithm and obviously, we do not want our algorithm to be biased towards one feature. In this project we would be building model using KNN and SVM which uses distance calculation as part of the algorithm. Therefore, it is necessary to normalize data so that higher value attributes do not bias the model. We use z-score normalization here.



The image shows a software interface with two main panels. The top panel, titled "Select Relevant Features", has a "Score" dropdown menu set to "Information Gain". Below this, the "Number of features" section has two radio buttons: "Fixed:" (selected) with a value of 7, and "Proportion:" (unselected) with a value of 75.00%. The bottom panel, titled "Normalize Features", has five radio buttons: "Standardize to $\mu=0, \sigma^2=1$ " (selected), "Center to $\mu=0$ ", "Scale to $\sigma^2=1$ ", "Normalize to interval $[-1, 1]$ ", and "Normalize to interval $[0, 1]$ ".

Select Relevant Features

Score

Information Gain

Number of features

☒ Fixed: 7

☐ Proportion: 75.00%

Normalize Features

☒ Standardize to $\mu=0, \sigma^2=1$

☐ Center to $\mu=0$

☐ Scale to $\sigma^2=1$

☐ Normalize to interval $[-1, 1]$

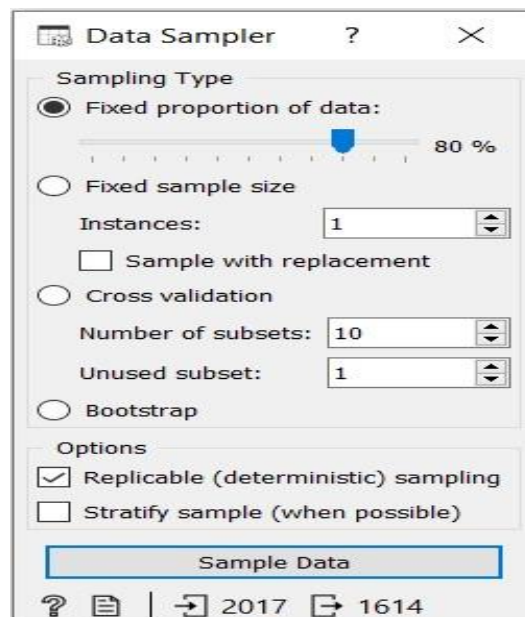
☐ Normalize to interval $[0, 1]$

The attributes of the dataset after feature selection is given below:

target	song_title	artist	instrumentalness	loudness	danceability	duration_ms	speechiness	valence	tempo
0	The Man I Love	Marcus Roberts	0.633	-24.477	0.520	341667	0.0497	0.1340	107.327
1	Money Trees	Kendrick Lamar	0	-7.384	0.739	386907	0.1010	0.3740	143.948
1	Wait & See	Holy Ghost!	0.00626	-5.015	0.646	219754	0.0331	0.9350	119.998
0	Perfect Harmony	Rags Cast	0	-7.356	0.664	145707	0.0322	0.7100	141.916
1	Midnight City	M83	1.31e-06	-13.742	0.517	245013	0.0341	0.3310	104.996
0	Symphony	Clean Bandit	3.29e-05	-4.699	0.718	214867	0.0429	0.4700	122.948
0	Sola (Remix) ...	Anuel Aa	0	-4.024	0.639	307910	0.1470	0.7660	169.801
1	Are you... Can ...	Shabazz Palaces	0.054	-9.971	0.502	287827	0.2710	0.2070	82.738
0	Save My Soul	JoJo	0	-6.085	0.406	224848	0.1300	0.4040	177.916
1	Oh	FIDLAR	0.129	-5.746	0.351	141305	0.0539	0.6040	126.394
0	Sexy Bitch (feat...	David Guetta	0.000371	-5.021	0.809	195853	0.0561	0.7970	130.008
0	Show You Love	Kato	0	-5.121	0.519	182720	0.0901	0.6290	123.659
0	Tearin' up My ...	*NSYNC	7.16e-06	-4.447	0.686	211000	0.0364	0.7840	110.054
0	Raven	John Dahlbäck	0.958	-3.114	0.569	190155	0.0489	0.1130	128.083
0	Sleep Without ...	Brett Young	0	-5.713	0.631	187813	0.0432	0.6370	88.541
1	Taylor Gang - ...	Wiz Khalifa	0	-3.590	0.621	335156	0.1580	0.5240	141.910
0	I'm the One	DJ Khaled	0	-4.267	0.599	288877	0.0367	0.8110	80.984
1	Danger and ...	Brown Bird	0.000632	-9.271	0.462	226333	0.0916	0.7480	209.686
0	Hollaback Girl	Gwen Stefani	6.17e-06	-2.221	0.926	199853	0.0929	0.9030	110.007
1	Say Please	Teams vs. Star ...	0.154	-6.691	0.731	207496	0.0608	0.8610	91.960
0	Honest	Joseph	0.00562	-8.537	0.413	168480	0.0310	0.3360	97.942
0	Fog On the Tyne	Lindisfarne	0	-17.264	0.615	203067	0.1020	0.7070	80.171
1	Fam	Saskilla	0	-4.316	0.709	176589	0.3350	0.4740	139.639
0	I'll Make Love T...	Boyz II Men	0	-7.775	0.563	235853	0.0234	0.2480	142.530
0	Burn It To The ...	Nickelback	0.504	-6.353	0.614	211067	0.0616	0.6150	132.074
0	No Absolution	Thy Art Is Murder	0	-0.994	0.468	201783	0.1500	0.0640	140.003
0	Icy, Creamy Ice ...	Barney	1.52e-06	-11.129	0.693	87705	0.1800	0.8510	78.158
0	Unfinished	Mandisa	0	-4.635	0.621	215050	0.0323	0.2550	92.994
1	Anything New	Bibio	0.000186	-4.276	0.651	247360	0.1670	0.7110	88.237
0	The Folks Who ...	Joshua Redman	0.658	-8.885	0.428	240093	0.0340	0.2210	122.667
1	Angels	The xx	0.114	-18.127	0.342	171613	0.0477	0.3290	91.963
0	Like Nobody's ...	Big Time Rush	0	-4.572	0.586	176360	0.1100	0.3530	94.954
1	Please Stop ...	Father	0.125	-12.766	0.735	186096	0.1020	0.3260	129.004
1	Keep You	Wild Belle	0.00748	-5.780	0.638	210160	0.0504	0.9150	133.948
0	Das Lied Der ...	Vader Abraham	0	-8.759	0.658	243493	0.1240	0.9290	168.883

Splitting Data

After Feature Selection, we divide the data into train and test data in the ratio 80:20 for accurate analysis. Here, out of 2017 songs, I took 80% proportion of data as train data making the remaining data as test data.



From the given dataset, we can see two categorical attributes as meta, i.e., song name and artist as both have no role in analyzing or predicting which song I like or not.

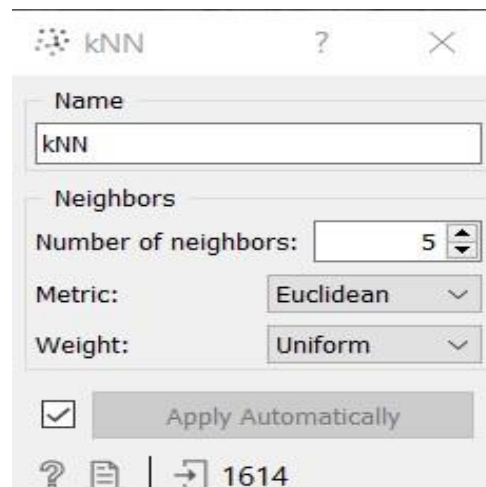
3.3 Data Mining Techniques Used:

In this work, we have used four classifiers namely KNN algorithm, Naïve Bayes Algorithm, SVM, and decision tree which are briefly explained as follows.

K-Nearest Neighbors Algorithm

The k-nearest neighbors (KNN) algorithm is a simple, supervised machine learning algorithm that is easy to understand implement which will solve both classification and regression problems.

The figure shown above is the confusion matrix obtained after implementing the K-nearest neighbor algorithm. The values which are represented in blue shades are correct.



		Predicted		Σ
		0	1	
Actual	0	77.0 %	17.1 %	778
	1	23.0 %	82.9 %	836
	Σ	838	776	1614

KNN models predict correctly that out of 1614 songs, 82.9% of songs I like, and 77% of songs I don't.

NAÏVE BAYES ALGORITHM

Naïve Bayes is a classification technique based on Bayes' theorem with an assumption of independence among predictors. The assumption of this algorithm is that the presence of one attribute in a class is unrelated to the presence of any other attribute.

The Naïve Bayes model is particularly useful for very large data sets. Naive Bayes is understood to perform even on highly advanced classification methods.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

Likelihood
Class Prior Probability

Posterior Probability
Predictor Prior Probability

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$$

Above,

- $P(c|x)$ –the posterior probability of class, given predictor.
- $P(c)$ -the prior probability of class.
- $P(x|c)$ -the probability of the predictor of the given class.
- $P(x)$ -the prior probability of the predictor.

Here, c is the target and x is the attribute.

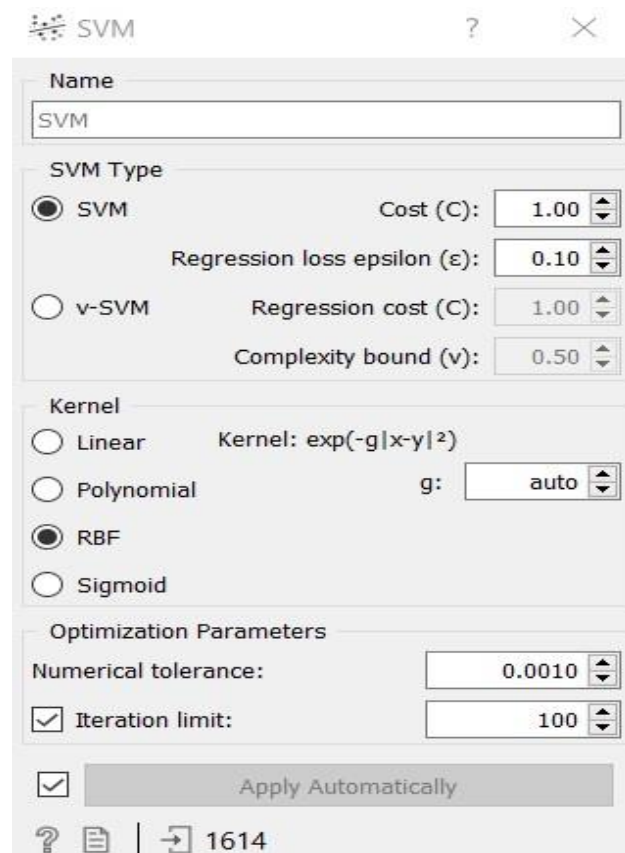
The figure shown below is that the confusion matrix obtained after implementing the Naïve Bayes algorithm. After comparing with both the algorithms we can find that the Naïve Bayes algorithm predicted more likely than the KNN algorithm.

		Predicted		Σ
		0	1	
Actual	0	70.4 %	28.0 %	778
	1	29.6 %	72.0 %	836
Σ		770	844	1614

The Naïve Bayes model predicts correctly that out of 1614 songs, 72% songs I like and 70.4% songs I don't.

SUPPORT VECTOR MACHINES(SVM)

Support Vector Machine is mostly performed as it produces significant accuracy with less computation power. It can be used for both regression and classification tasks. The goal is to create best line or decision boundary(hyperplane) that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence is termed as support vector machine (SVM).



The image shows a configuration window titled "SVM" with a close button (X) and a help button (?). The window contains several sections for configuring the SVM model:

- Name:** A text field containing "SVM".
- SVM Type:** Two radio buttons are present: "SVM" (selected) and "v-SVM".
 - For "SVM":
 - Cost (C): A numeric input field with value "1.00".
 - Regression loss epsilon (ϵ): A numeric input field with value "0.10".
 - For "v-SVM":
 - Regression cost (C): A numeric input field with value "1.00".
 - Complexity bound (v): A numeric input field with value "0.50".
- Kernel:** Four radio buttons are present: "Linear", "Polynomial", "RBF" (selected), and "Sigmoid".
 - For "Linear": The kernel is set to $\exp(-g|x-y|^2)$.
 - For "Polynomial": A parameter "g" is set to "auto".
- Optimization Parameters:**
 - Numerical tolerance: A numeric input field with value "0.0010".
 - Iteration limit: A checked checkbox followed by a numeric input field with value "100".
- Buttons:** A checked checkbox followed by a button labeled "Apply Automatically".
- Footer:** A row of icons (help, save, undo) followed by the number "1614".

The figure shown below is that the confusion matrix obtained after implementing the SVM model.

		Predicted		Σ
		0	1	
Actual	0	54.2 %	38.4 %	778
	1	45.8 %	61.6 %	836
	Σ	1002	612	1614

SVM predicts correctly that out of 1614 songs, 61.6% songs I prefer, and 54.2% songs I do not.

DECISION TREE

The decision tree algorithm, a supervised learning algorithm is typically used for solving regression and classification problems. The aim of a choice Tree is to make a training model which may be used to predict the category or value of the target variable by learning simple decision rules inferred from prior data (training data).

For prediction, we start from the root node and then compare the values of it with the record's attribute. On the basis of comparison, we follow the branch and jump to the next node.

Tree

?

×

Name

Tree

Parameters

☒ Induce binary tree

☒ Min. number of instances in leaves: 2

☒ Do not split subsets smaller than: 5

☒ Limit the maximal tree depth to: 100

Classification

☒ Stop when majority reaches [%]: 66

☒ Apply Automatically

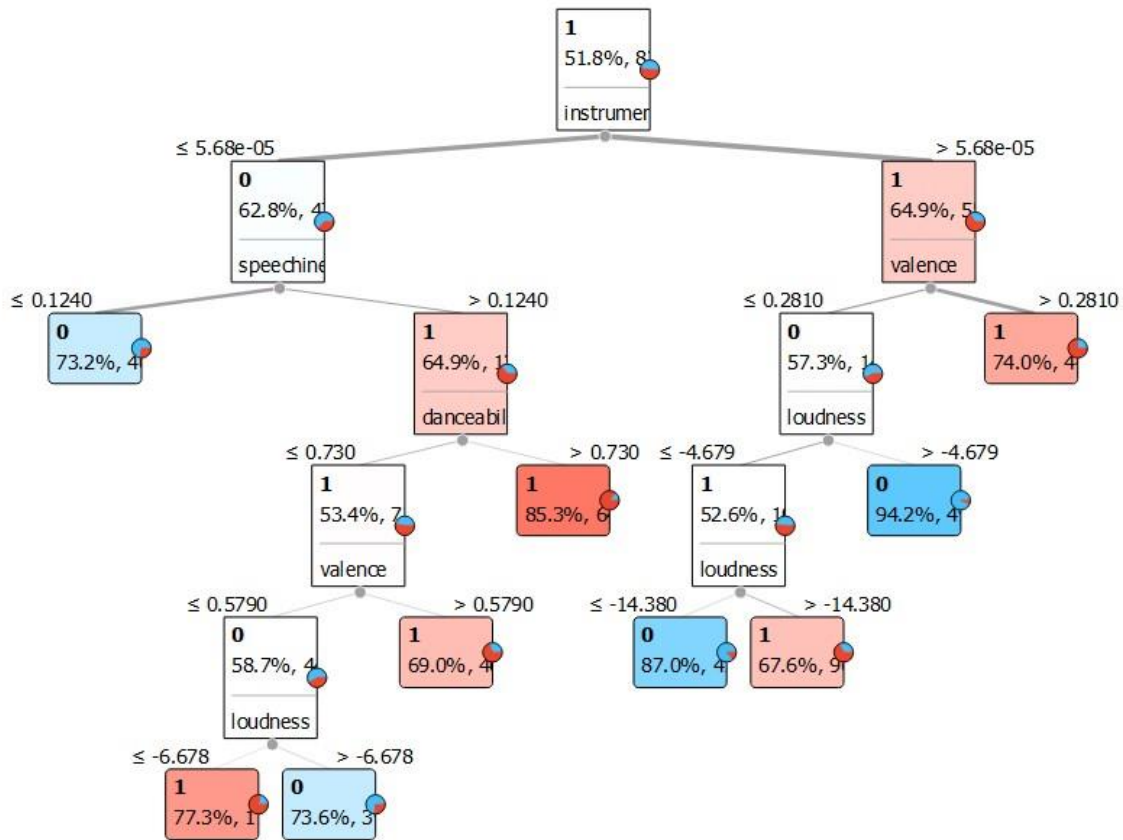
?

1614

The figure shown below is that the confusion matrix obtained after implementing Tree.

		Predicted		
		0	1	Σ
Actual	0	75.8 %	26.3 %	778
	1	24.2 %	73.7 %	836
Σ		714	900	1614

The figure given below is that the tree structure of the Spotify dataset using relevant features.



In this algorithm, the input attributes are taken to construct a tree. A group of rules representing the various classes is then derived from the tree. These rules are used to forecast the class of a new instance with an unknown class.

Using the confusion matrix of the choice tree, we will predict correctly that out of 1614 songs, 73.7% songs I prefer, and 75.8% songs I do not.

3.4 Performance Evaluation Criteria

In this study, three performance evaluation criteria are used during the performance evaluation phase. These are; accuracy, precision, and recall.

Precision finds what percentage of tuples that are labeled as positive is such. This will be seen as a measure of exactness. The Recall finds what percentage of positive tuples are labeled intrinsically, which may be seen as a measure of completeness.

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{FN} + \text{FP} + \text{TN}) * 100\%$$

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) * 100\%$$

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) = \text{TP} / \text{P} * 100\%$$

True Positive (TP) - the number of correct classifications of liked songs supported audio features.

False negative (FN) - the number of incorrect classifications of unliked songs supported audio features.

True Negative (TN) - the right classification number of songs supported features.

False Positive (FP) - the number of incorrect classifications of songs supported features.

4. Experimental Results and Discussions

The dataset of 2017 songs are split in the ratio 80:20 using a data sampler. The data sample with 80% split (1614 songs) were used for the model building purpose and the remaining data was used to check the model performance on new data. Then I identified the important features using information gain, Gini Index and selected all relevant attributes.

A total of five different models were built on our dataset using the classification techniques. To evaluate the prediction made by the models we used confusion matrices.

The test and score calculation is given below of different models is used in prediction and to select the best model by comparing the accuracy values of each model.

Model	AUC	CA	F1	Precision	Recall
kNN	0.885	0.798	0.798	0.800	0.798
Tree	0.765	0.746	0.745	0.747	0.746
SVM	0.621	0.570	0.564	0.580	0.570

Model	AUC	CA	F1	Precision	Recall
Naive Bayes	0.776	0.713	0.712	0.712	0.713

Accuracy of the models:

The accuracy of the KNN model = 79.8%

The accuracy of the Naïve Bayes model = 71.3%

The accuracy of the SVM model = 57%

The accuracy of the tree model = 74.6%

From the above accuracy values of all four models, we can see that the Accuracy, Recall and Precision value of the KNN model is high (79.8%). Hence, I choose Tree model as the best or accurate model for further classification on the Spotify dataset.

The below figure shows the predicted value compared with the actual value of the remaining, test data(20%).

	kNN	target	song_title	artist	instrumentalness	loudness	danceability	duration_ms	speechiness	tempo	valence
1	0.80 : 0.20 → 0	0	Call On Me - ...	Starley	-0.4880565	0.152795	0.1713	-1.0480312	-0.61022	-0.886326	0.37302
2	0.00 : 1.00 → 1	1	Bless My Soul	Nightmares On ...	2.7342718	-1.690453	1.4571	1.3451903	0.31516	-0.810161	0.00884
3	0.20 : 0.80 → 1	1	Lose My Mind	A-Trak	-0.4880513	0.052283	-1.2884	-0.5710755	3.22921	-0.434322	0.40943
4	0.80 : 0.20 → 0	0	Tattooed Heart	Ariana Grande	-0.4880565	0.782456	-1.0213	-0.6295542	-0.71366	-1.848768	-1.07559
5	0.60 : 0.40 → 0	1	Polish Girl	Neon Indian	-0.4879858	0.778468	0.0906	0.2158795	-0.62023	-0.360706	-0.16920
6	0.60 : 0.40 → 0	0	I'd Do Anything ...	Meat Loaf	-0.4880565	0.455128	-1.2139	0.9349123	-0.12751	-0.670313	-0.80448
7	0.40 : 0.60 → 1	0	You Wouldn't ...	Flower Soo	-0.4872766	-0.144752	-0.2759	0.6430069	-0.65916	0.014458	-1.50451
8	0.40 : 0.60 → 1	1	Go	Grimes	0.1966883	0.241075	0.0160	-0.0676683	-0.66694	0.690008	-0.84495
9	0.20 : 0.80 → 1	1	Jumpman	Drake	-0.4880565	-0.119491	1.4571	-0.4932463	1.12709	0.766960	0.72910
10	0.40 : 0.60 → 1	1	Viol - Original ...	Gesaffelstein	2.5328763	0.032340	-0.1331	1.1904591	-0.61133	-0.472667	-1.20507
11	0.60 : 0.40 → 0	0	Trying To Drive	Zac Brown Band	-0.4880565	0.319783	-1.1456	0.2498832	-0.64581	0.909245	-0.07613
12	0.20 : 0.80 → 1	0	물 Water	권나무 Kwon Tree	-0.4880089	-2.275709	1.0782	0.5752922	-0.49010	-1.423527	-0.67500
13	0.00 : 1.00 → 1	1	Every Freakin' - ...	DJ Eleven	-0.4880565	0.458053	0.3266	-0.5557756	2.42840	0.465412	0.75742
14	0.40 : 0.60 → 1	0	Strong Enough	Kina Grannis	-0.4880565	-0.665924	0.1154	-0.6239418	-0.71588	-0.587401	0.27995
15	0.60 : 0.40 → 0	0	Oh Devil	Electric Guest	-0.4342290	-0.053547	1.7739	-0.3503987	-0.36553	-0.922234	0.91119
16	1.00 : 0.00 → 0	1	Northern Lights	Kate Boy	-0.4005410	0.198531	0.7676	-0.3857079	-0.45895	-0.922646	0.76956
17	0.80 : 0.20 → 0	0	When Will I Learn	Kina Grannis	-0.4880226	-1.726085	-0.6548	-0.6391928	-0.70921	0.308510	-0.46863
18	0.20 : 0.80 → 1	1	Slippin'	Quadron	-0.4566022	-0.284086	1.1093	0.0839517	-0.52458	-0.133861	1.86613
19	0.60 : 0.40 → 0	0	Get It On Tonite	Montell Jordan	-0.4879844	-0.441767	1.1838	0.3510649	-0.11750	-0.847194	1.54647
20	0.40 : 0.60 → 1	0	Best I Ever Had	Drake	-0.4880565	0.588080	-1.1890	0.1397462	3.15136	1.505782	0.34064
21	0.40 : 0.60 → 1	0	Whippin' (feat. ...	Kiiara	-0.4750207	-0.109387	1.5627	-1.0457252	-0.42003	-1.464571	0.72505
22	0.60 : 0.40 → 0	0	All Eyes On You ...	Meek Mill	-0.4880565	0.477996	-0.1828	-0.2724840	1.22719	-1.652322	-0.98252
23	0.40 : 0.60 → 1	0	The Sweetest ...	Ellie & The Bunch	2.8844030	-2.737850	-1.2015	-0.8544769	-0.70476	1.758827	-1.09178
24	1.00 : 0.00 → 0	0	Vanilla Twilight	Owl City	-0.4867053	-0.132520	-0.2325	-0.1781226	-0.73590	1.669768	0.21925
25	0.80 : 0.20 → 0	0	Heartatear	To The Grave	-0.0999124	0.814365	-0.8847	0.8369882	0.30404	0.912468	-0.18539
26	1.00 : 0.00 → 0	0	Gave Me ...	Jess Glynne	-0.4880565	0.409127	-0.1828	-0.4778000	0.06379	-0.997199	0.39729
27	0.60 : 0.40 → 1	0	The Boss	James Brown	-0.4578838	-0.672306	0.7925	-0.6780404	-0.54793	-0.887600	0.40134
28	0.60 : 0.40 → 0	0	Viola Sonata in ...	Felix ...	2.7489188	-4.170277	-2.0090	6.5604811	-0.60243	-0.849518	-1.52878
29	0.20 : 0.80 → 1	1	Coming Home	Leon Bridges	-0.4878544	0.130459	-1.1890	-0.4870482	-0.70698	-0.354821	-0.14087
30	0.20 : 0.80 → 1	1	Just A Little Lovin'	Dusty Springfield	-0.4880565	-0.987403	-1.6425	-1.3100445	-0.64692	-0.395303	0.25972
31	0.80 : 0.20 → 0	0	Starships	Nicki Minaj	-0.4880565	1.240609	0.8049	-0.4353166	-0.06189	0.127843	0.74124
32	1.00 : 0.00 → 0	1	Chrome Knight ...	Surkin	-0.4577740	0.571328	-0.2014	-0.3913935	-0.39222	0.391496	0.10595

The Confusion matrix of the 20% dataset is:

		Predicted		Σ
		0	1	
Actual	0	72.6 %	36.1 %	219
	1	27.4 %	63.9 %	184
Σ		201	202	403

The above figure tells that out of 403 rows, I predict correctly that 63.9% of songs I like and 72.6% of songs I don't.

The model gives an accuracy of 68.2% with the new data. Therefore it is observed that the above model works efficiently with a good accuracy on a new dataset.

Model	AUC	CA	F1	Precision	Recall
kNN	0.758	0.682	0.683	0.686	0.682

5.Conclusion

The project goal is to build a model to predict whether I like a song or not based on its audio features. Through several different feature selection algorithms, we found most relevant features from our dataset, namely artist instrumentness, loudness, energy, speechiness, etc.

In our research, we investigated four different machine learning models: KNN, Naïve Bayes, SVM, and decision tree and their abilities to predict like songs on the train data. Out of all the four models, the most accurate model was the KNN model with an accuracy of 79.8%. And the accuracy value of test data was also high (68.2%). Therefore, we choose the tree model as the best model to predict which songs I like and which I don't.

So, from the confusion matrix of the KNN model of the train data, I predict correctly that out of 1614 songs, 82.9% songs I like, and 77% songs I do not and from the confusion matrix of the KNN model of the test data, I predict correctly that out of 403 songs, 63.9% songs I like and 72.6% songs I don't like.

The goals of this project was to find out what type of song I like/dislike. I found from the tree model that I like songs that are higher in valence (less positive songs), loudness, and are more acoustic.

6. References

1. <https://www.kaggle.com/geomack/spotifyclassification>
2. <https://opendatascience.com/a-machine-learning-deep-dive-into-my-spotifydata/>
3. <https://developer.spotify.com/documentation/web-api/reference/#endpointget-audio-features>
4. <https://towardsdatascience.com/a-practical-guide-to-exploratory-dataanalysis-spotify-dataset-d8f703da663e>