Team 34

Content

Overview

DIY music distribution, digital downloads and streaming have changed the relationship between record labels and artists. Record labels are companies that market recorded music and corresponding videos. Today with good music, gigging and social media grit, anyone can launch its own music career. Getting signed usually means we've already done the major legwork. Today, record labels are still leaders of the key elements of the music industry.

To purchase the rights of the music track usually the bidding process came into the picture. In the bidding process mainly the record labels bid on the track according to their popularity. Record labels consider some parameters such as acousticness, apeechiness, liveliness, danceability etc to get some rough idea about which song will become popular among the audience. Songs of higher popularity seek a high bid amount but increase the revenue. Record label does not want to encounter any losses with promotion and distribution of the track.

So here our objective is to develop an efficient prediction model for the record labels so that they can **efficiently bid on the songs** and **maximize their revenue** and also not encounter any huge loss. We have used data visualization to get insights of data and develop intelligent algorithms to make our prediction effective. Here we have identified outliers, identified correlations between various features & added some efficient features. We have developed various statistical and machine learning models and execute them on a training dataset for model selection.

Exploratory Data Analysis

1. Data Description

Summary Statistics

The data provided consists of different audio features and their release dates with **12227 unique id's** .The dataset consists of the popularity of the audio. For summary statistics for the given dataset for all the unique id's refer to Annexure 1.

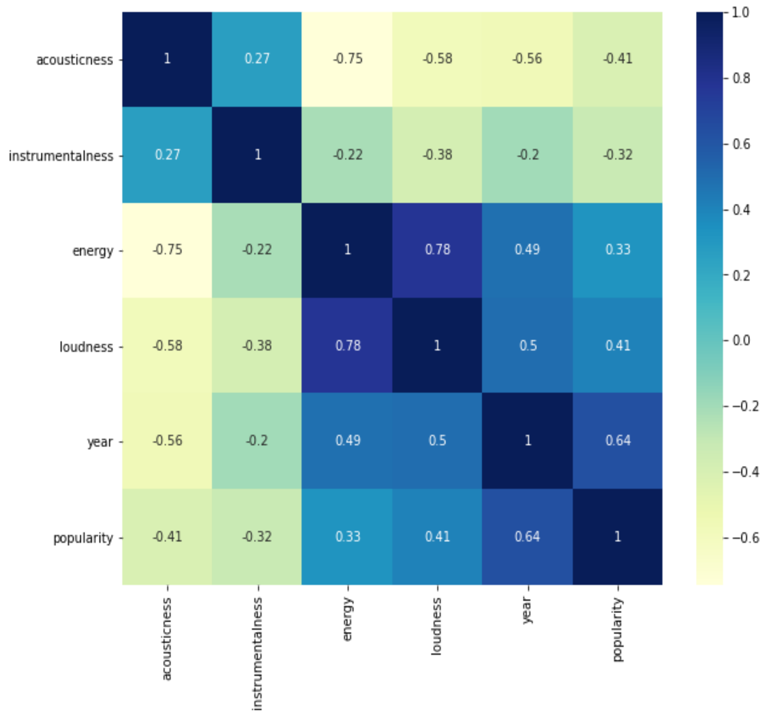
From the correlation matrix we can conclude that popularity of the songs majorly depends on features such as acousticness, release years and date, energy danceability and tempo.

1. **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.
2. **Liveness** score detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live.
3. **Speechiness** detects the presence of spoken words in a track.
4. **Instrumentalness** predicts whether a track contains no vocals like ("Ooh" and "aah").
5. **Loudness** score for each sample which is useful for comparing relative loudness of the tracks.

6. **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.
7. **Key** feature is Integers map to pitches using standard pitch class notation.
8. **Tempo** feature is a measure of BPM(beats per minute) which shows the speed or pace of sample audio.
9. **Acousticness** is a confidence measure from 0.0 to 1.0 of whether the track is acoustic.
10. **Valence** describes 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).
11. **Explicit track** is one that has curse words or language or art that is sexual, violent, or offensive in nature.

Correlation Matrices

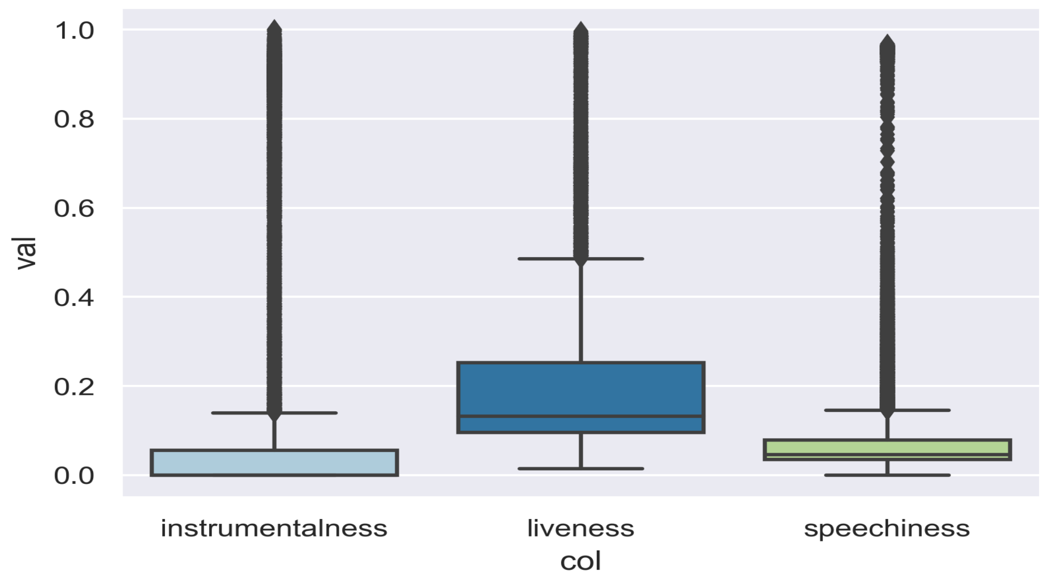
Correlation is a statistical term which in common usage refers to **how close two variables are to having a linear relationship with each other.**
For example, two variables which are linearly dependent (say, x and y which depend on each other as x = 2y) will have a higher correlation than two variables which are non-linearly dependent (say, u and v which depend on each other as u = v^2). Also a positive correlation means that as one feature increases so does the second one and vice versa.
Intuitively we can understand that louder the sound is, the more energy the sound wave contains. The correlation we found out between loudness and energy is pretty high, which validates the above mentioned fact. The popularity of songs increases as we move ahead in time indicating that people prefer new songs to old ones and also there is an overall increase in the number of people listening to music.
The features which are highly correlated with popularity are shown below with the help of heatmap,for boxplot of correlation of all features with popularity refer to Annexure 1.



2. Anomaly Detection

We wanted to check for outliers so we plotted the boxplots of different features. The following are the boxplots of features having outliers more than **6%.** For the features having outliers refer to Annexure 3

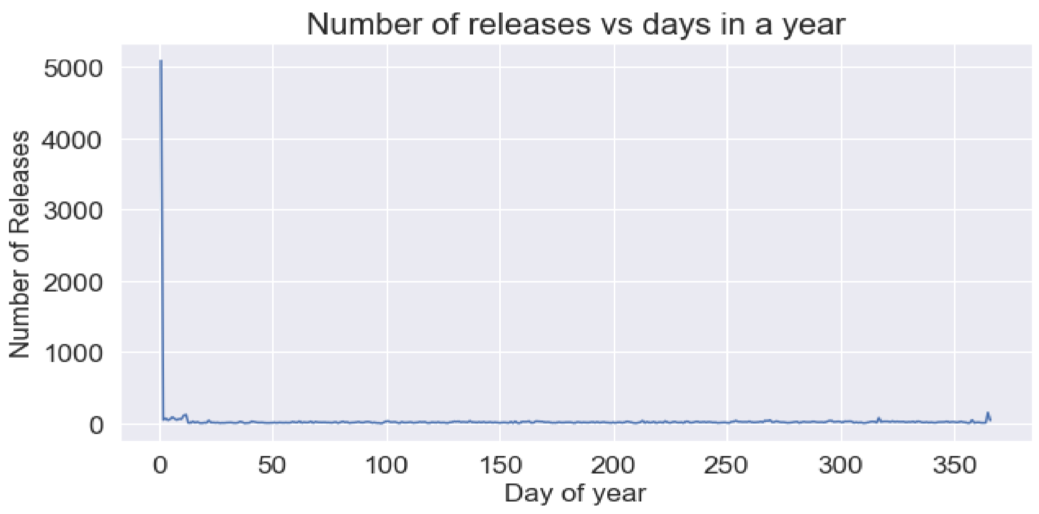
Since we can see from the boxplots that there are a significant number of outliers so we decided to treat them **with the help of 3-sigma rule**.A threshold for upper limit and lower limit were set and the values greater or lower than them were made equal to the threshold respectively.Since the dataset is small so we cannot remove them and hence we used Imputation to handle the outliers.



Feature Engineering

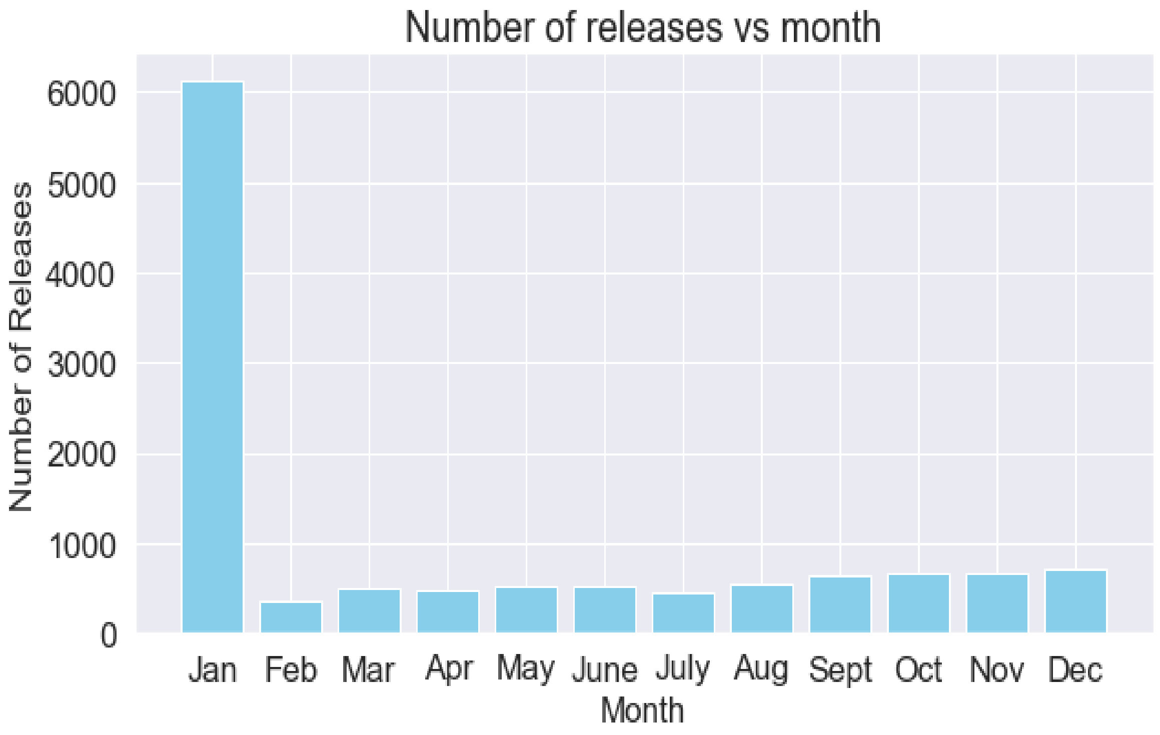
1. Release Day

41.662 percent of the songs were released on 1st January and the Correlation between Day of the year and Popularity is 0.2 (moderate). So, we added a **new feature called Release_day** which contains the day of the year. This feature was obtained from the released_date feature.



2. Release Month

We have identified most of the songs released in January and the correlation between the Month and Popularity is around 0.2. So, we have made **a new feature called Release_month** which contains the month number on which the song was released. So, we dropped the release_date feature from the Dataset.



Feature Importance

CHI-SQUARE TEST

What is a Chi Square Test?

There are two types of chi-square tests. Both use the **chi-square statistic** and distribution for different purposes:

- A chi-square **goodness of fit test** determines if sample data matches a population.
- A chi-square test for **independence** compares two variables in a contingency table to see if they are related. In a more general sense, it tests to see whether distributions of **categorical variables** differ from each other.
    - A very small chi square test statistic means that the observed data fits your expected data extremely well. In other words, there is a relationship.
    - A very large chi square test statistic means that the data does not fit very well. In other words, there isn't a relationship.

What is a Chi-Square Statistic?

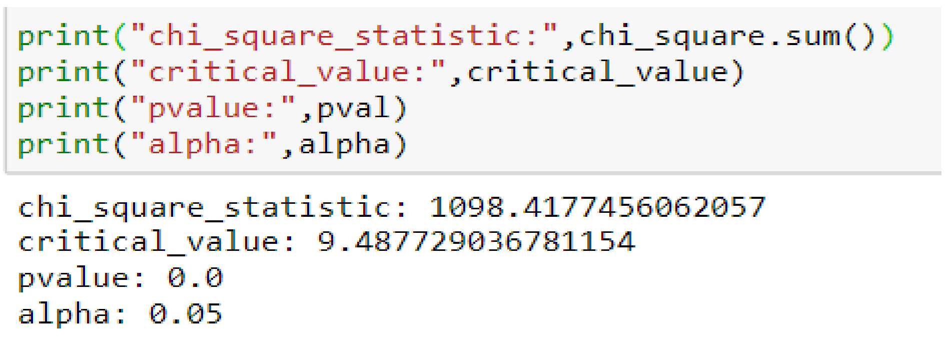
The formula for the chi-square statistic used is:

$$\chi^2_c = \sum \frac{(O_i - E_i)^2}{E_i}$$

Where,"O" is your observed value and E is your expected value and the subscript "c" is the degrees of freedom.

Here we performed chi-square test on Popularity with explicit and mode and found that the chi _square_statistic was greater than the critical_value and the **p value** was **less** than **alpha**(0.05) so the **null hypothesis** was **rejected** and hence there exist a **relationship** between popularity and explicit, popularity and mode.

Popularity vs explicit



```
print("chi_square_statistic:",chi_square.sum())
print("critical_value:",critical_value)
print("pvalue:",pval)
print("alpha:",alpha)
```

```
chi_square_statistic: 1098.4177456062057
critical_value: 9.487729036781154
pvalue: 0.0
alpha: 0.05
```

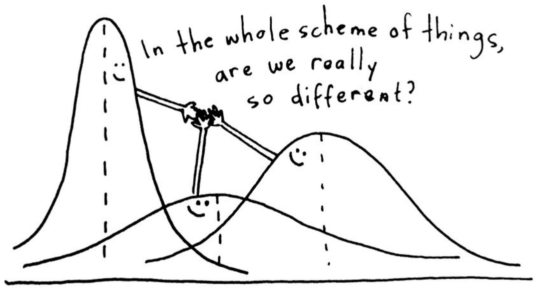
Popularity vs mode

```
print("chi_square_statistic:",chi_square.sum())
print("critical_value:",critical_value)
print("pvalue:",pval)
print("alpha:",alpha)
```

```
chi_square_statistic: 57.60819977292354
critical_value: 9.487729036781154
pvalue: 9.221734487141475e-12
alpha: 0.05
```

ANOVA TEST

Analysis of variance (ANOVA) is a statistical technique that is used to check if the means of two or more groups are significantly different from each other. ANOVA checks the impact of one or more factors by comparing the means of different samples.



Source: https://www.questionpro.com/blog/anova-testing/

There are various types of ANOVA TESTS:

i) ONE WAY ANOVA: The one-way analysis of variance (ANOVA) is used to determine whether there are any statistically significant differences between the means of three or more independent (unrelated) groups.

ii) TWO WAY ANOVA: A two-way ANOVA test is a statistical test used to determine the effect of two nominal predictor variables on a continuous outcome variable.A two-way ANOVA test analyzes the effect of the independent variables on the expected outcome along with their relationship to the outcome itself.

Here Anova test was performed on the features and **top10 features** were taken which **increased** the **accuracy** of our models by about **2-3 %** and **f1_score** by **2%**.

Multi Class Classification Models

Classification is a task that requires the use of machine learning algorithms that learn how to assign a class label to examples from the problem domain.
Majorly used Multi class classification models are :

- Multi-Class Classification
- Imbalanced Classification

1. Multi-class classification refers to those classification tasks that have **more than two class labels**. Unlike binary classification, multi-class classification does not have the notion of normal and abnormal outcomes. Instead, examples are classified as belonging to one among a range of known classes. Popular algorithms that can be used for multi-class classification include:

- k-Nearest Neighbors.
- Decision Trees.
- Random Forest.
- Gradient Boosting.

2. Imbalanced Classification refers to classification tasks where the number of examples in each **class is unequally distributed**. Imbalanced classification tasks are classification tasks where the **majority** of examples in the training dataset belong to the normal class and a **minority** of examples belong to the abnormal class. Specialized techniques may be used to change the composition of samples in the training dataset by under-sampling the majority class or oversampling the minority class. Some examples are

- Random Under-sampling
- SMOTE Oversampling

a) **Random Under-sampling:** Random undersampling involves randomly selecting examples from the majority class to delete from the training dataset**.**This has the effect of reducing the number of examples in the majority class in the transformed version of the training dataset. This process can be repeated until the desired class distribution is achieved, such as an equal number of examples for each class.This approach may be more suitable for those datasets where there is a class imbalance although a sufficient number of examples in the minority class, such a useful model can be fit.

b) **SMOTE(Oversampling):** SMOTE (synthetic minority oversampling technique) is one of the most commonly used oversampling methods to solve the imbalance problem.It aims to balance class distribution by randomly increasing minority class examples by replicating them.SMOTE synthesises new minority instances between existing minority instances. It generates the virtual training records by linear interpolation for the minority class.

Models used for multi-class classification

1. XgBoost

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. It is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.

Benefits of using XgBoost:

1. Execution Speed

2. Model Performance

3. Flexibility in setting on objective function

4. Built-in cross validation and methods of dropout

As we had an imbalanced multi class classification,so to tackle this problem,we changed the weights of the classes with the weights being inversely proportional to the frequency of the class.

Results:

Popularity value=1 Frequency=992 (24.800%)

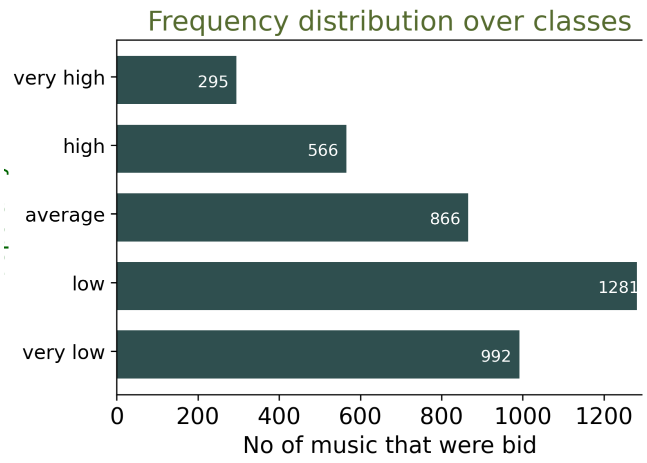
Popularity Value=2, Frequency=1281 (32.025%)

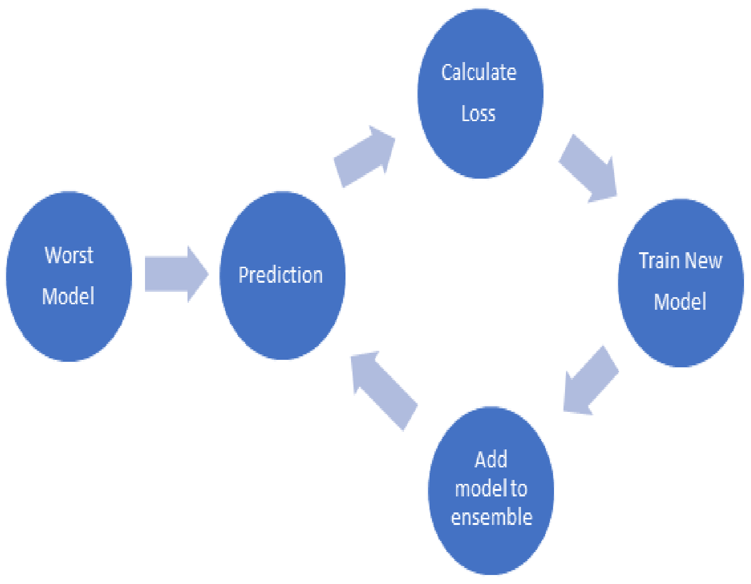
Popularity Value=3, Frequency=866 (21.650%)

Popularity Value=4, Frequency=566 (14.150%)

Popularity Value=5, Frequency=295 (7.375%)

Maximum possible bid : 9891

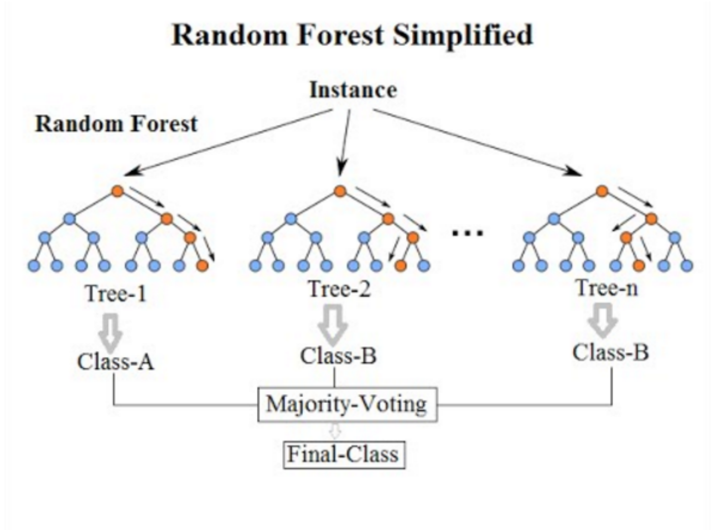




Source: https://medium.com/@siddhraj.maramwar/xgboost-for-fine-modelling-c42685195ae9

2. Random Forest Classifier

Random forest as the name suggests,consists of a large no. of individual trees that operate as an ensemble.Each decision tree is asked for its prediction and the class with the most votes becomes the prediction of our model. Random forest works efficiently as a large number of uncorrelated models(trees) act as a committee which could definitely outperform any of the constituent individual models.



Source:https://williamkoehrsen.medium.com/random-forest-simple-explanation-377895a60d2d

Reasons for using Random Forest Classifier:
Based on observation it was clear that Random Forest Classifier will perform better than other algorithms due the class imbalance(the no of musics with popularity very high is quite less)

1. Random Forest is in general robust to overfitting (which is a major problem when training in large and unbalanced datasets).
2. Parameterization and the use of weights to classify data points is intuitive.
3. Results from the model have shown better performance in the given training dataset.

To handle data imbalance used the following two techniques:

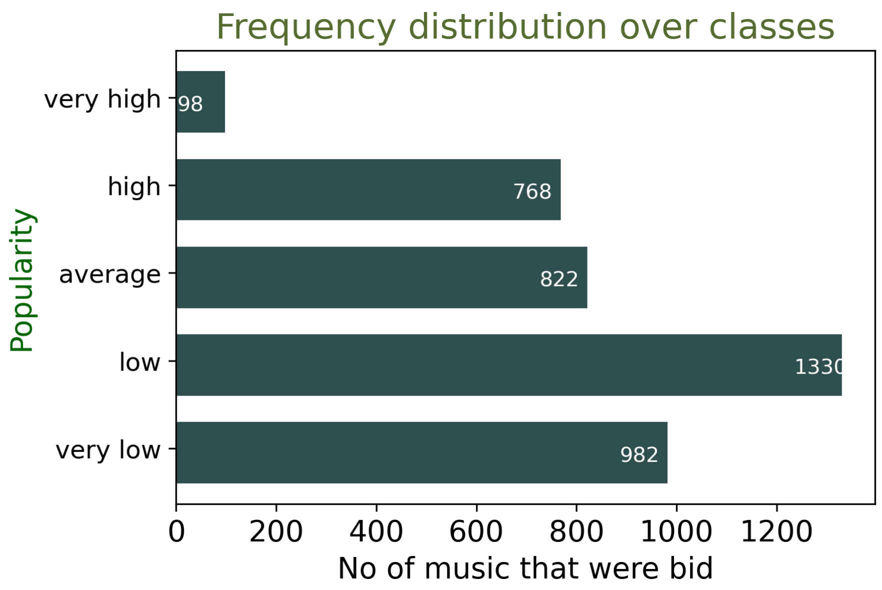
1. **Ensemble Cross-Validation**:
   Use of cross-validation is to justify the robustness of the model predictions.
- The entire dataset was divided into 5 parts.In each CV, 4 out of 5 subsets are used for training, and the remaining set was used to validate the model.
- In each Cross-Validation, the model also predicts the test data. At the end of the prediction we have five prediction probabilities. We take the average of probabilities for all classes.
- The f1-score for the large classes were higher than others. (Find the classification report in annexure)

2. **Set Class Weight**:
   This method is highly recommended for unbalanced datasets.
- The RF tends to be biased on the majority class.
- Therefore, imposing a cost penalty on the minority class misclassification can be useful.
- We assign higher weights to minority classes(thus more penalty for misclassification).
- We get the intuition of weights by the reciprocal of the ratio of class size. We then improved the weights by trial and error basis.

To find the best parameters, We performed a grid search over specified parameter values using *scikit-sklearn* implemented GridSearchCV.

Predictions

While training the model(Random Forest) we were able to achieve a decent accuracy of **84%** in the cross validation set.
With the help of grid search over parameters we were able to increase the cross-validation accuracy from 70% to 84%.
Predictions distribution over 4000 music from the test set.
Using Random Forest:



We see that the ***very high*** class prediction constitutes for 2.4% of the test set which is close to the contribution of 3.0% in the training dataset.
For full classification report refer to Annexure 5

Conclusion

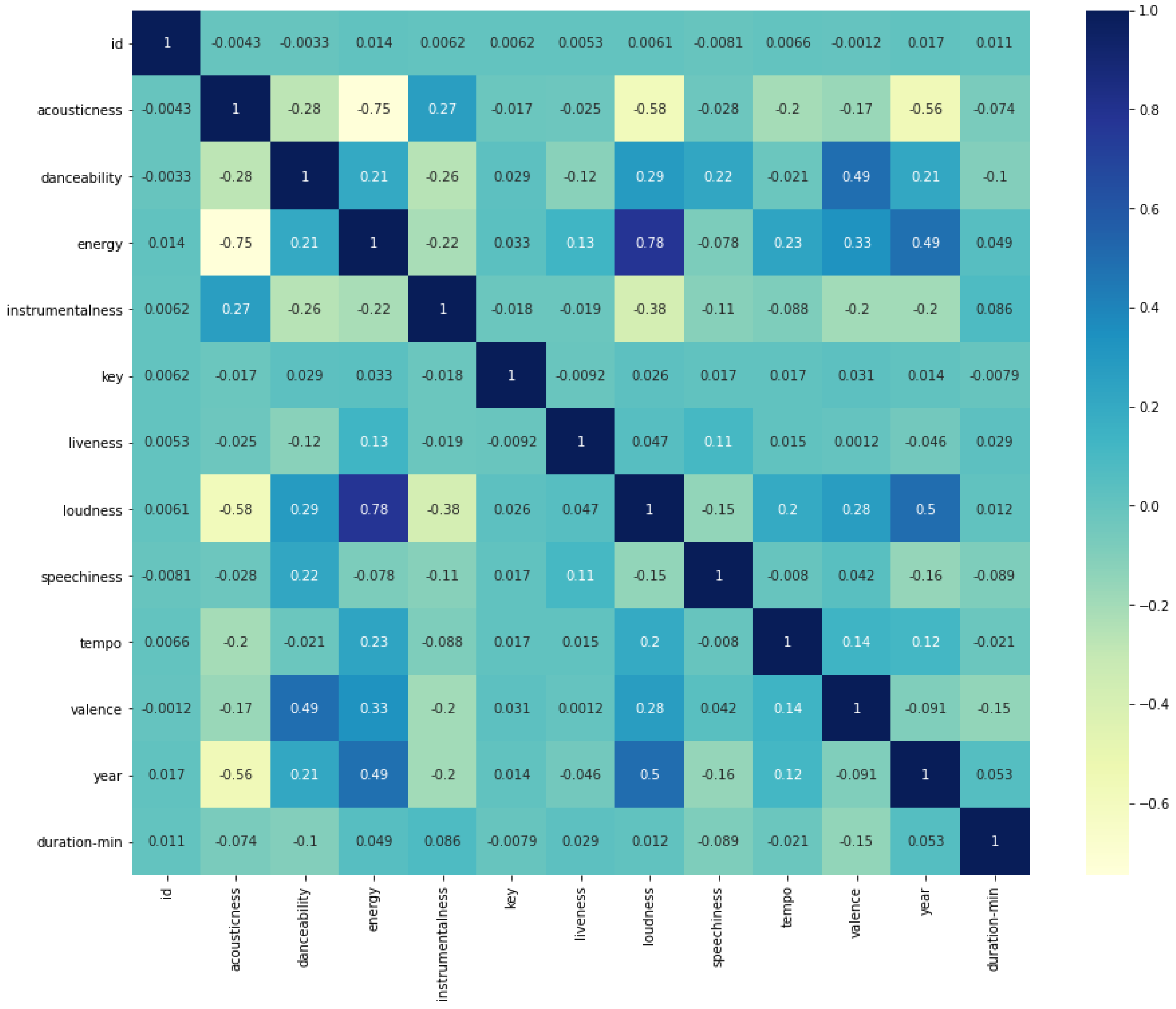
We ran the given dataset on mainly three models: XgBoost, Random Forest. When the predictions are compared with labels, we calculated the revenue on a cross-validation set of size ~ 4000.

Model	**Bidding cost**	**Revenue gained**
Random Forest	9259	17798
XgBoost	8096	14198

Clearly Random Forest is performing better than others in terms of accuracy and revenue gained so we came upon the conclusion that Random Forest outperforms other models when it comes to unbalanced datasets like music popularity.

Annexure(s)

1. Complete Heatmap

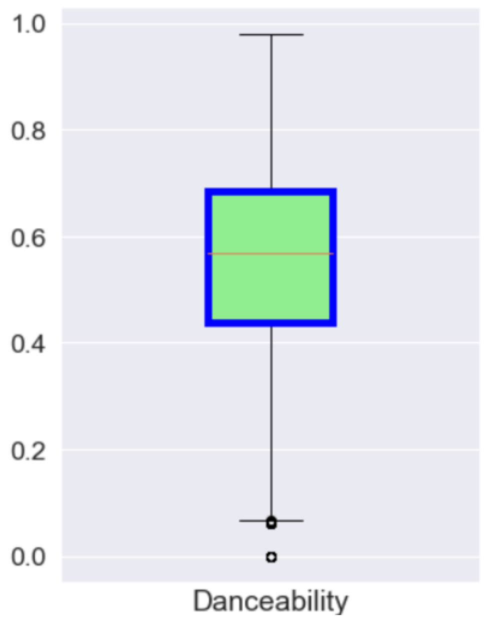


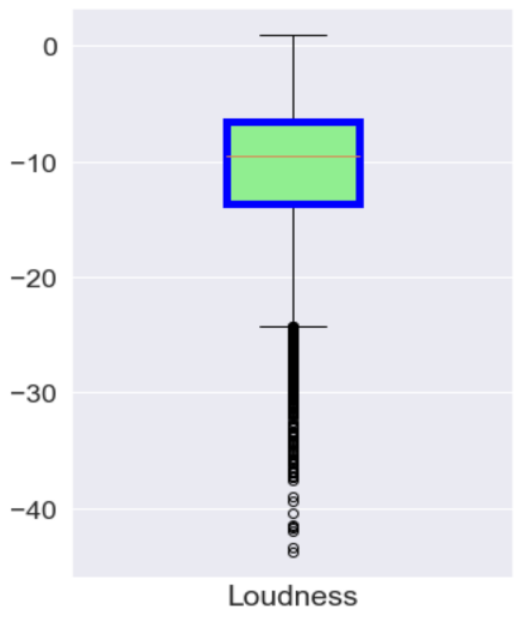
2. Complete Data Description

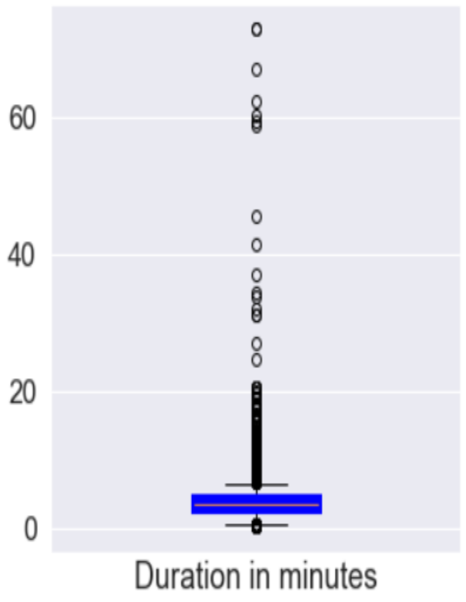
	acousticness	danceability	energy	explicit	instrumentalness	key	liveness	loudness
count	12227	12227	12227	12227	12227	12227	12227	12227
mean	0.430578	0.556353	0.522129	0.10804	0.149321	5.205202	0.201365	-10.6687
std	0.366893	0.175373	0.262482	0.310443	0.297954	3.526954	0.173987	5.506888
min	1.04E-06	0	2.03E-05	0	0	0	0.0147	-43.738
25%	0.05895	0.438	0.303	0	0	2	0.0962	-13.656
50%	0.354	0.569	0.534	0	0.000115	5	0.132	-9.584
75%	0.805	0.685	0.739	0	0.05565	8	0.252	-6.5715
max	0.996	0.98	1	1	1	11	0.997	1.006

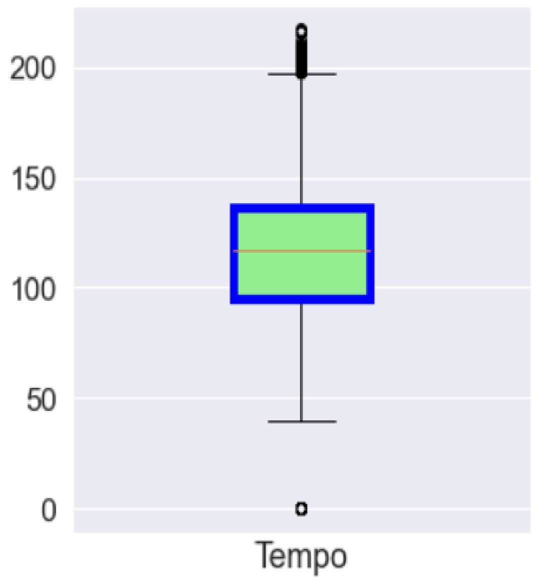
	mode	speechiness	tempo	valence	year	duration-min	popularity	release_month	release_day
count	12227	12227	12227	12227	12227	12227	12227	12227	12227
mean	0.69412	0.09768	118.1675	0.5253	1984.517	3.888133	2.491453	4.265314	108.6336
std	0.460798	0.155895	30.20006	0.258205	25.912	2.383133	1.176612	3.949784	126.411
min	0	0	0	0	1920	0.2	1	1	1
25%	0	0.0347	95.0505	0.321	1966	2.9	1	1	1
50%	1	0.0456	116.915	0.532	1987	3.6	2	1	31
75%	1	0.0789	136.1085	0.737	2008	4.4	3	8	224
max	1	0.968	216.843	1	2021	72.8	5	12	366

3. Boxplots

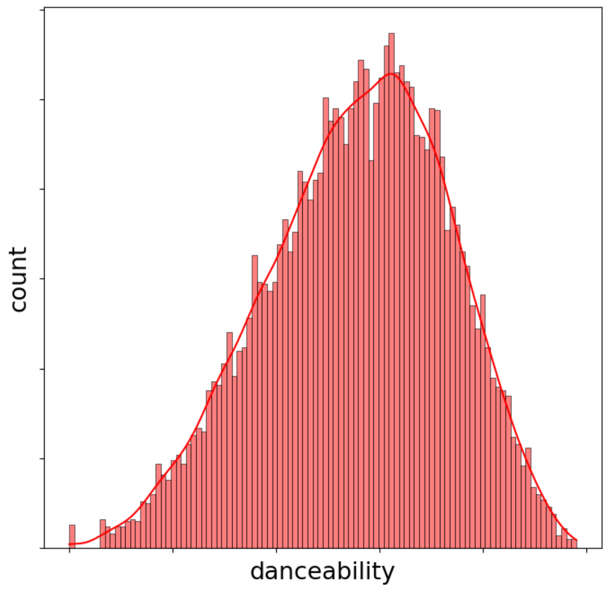

1.0
0.8
0.6
0.4
0.2
0.0
Danceability

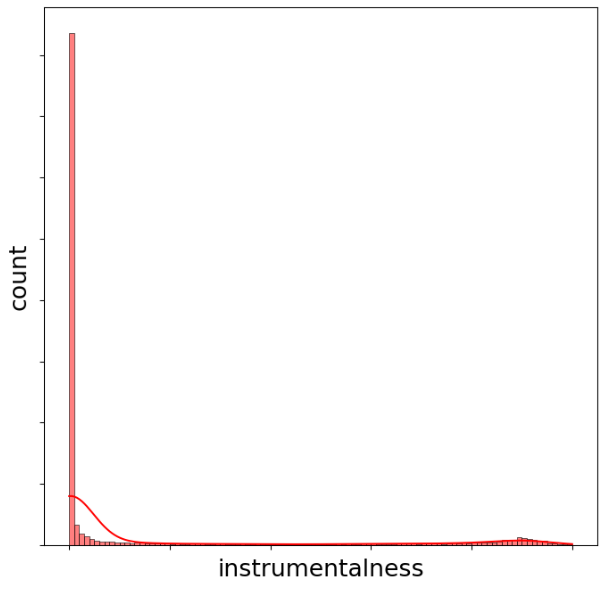

0
−10
−20
−30
−40
Loudness

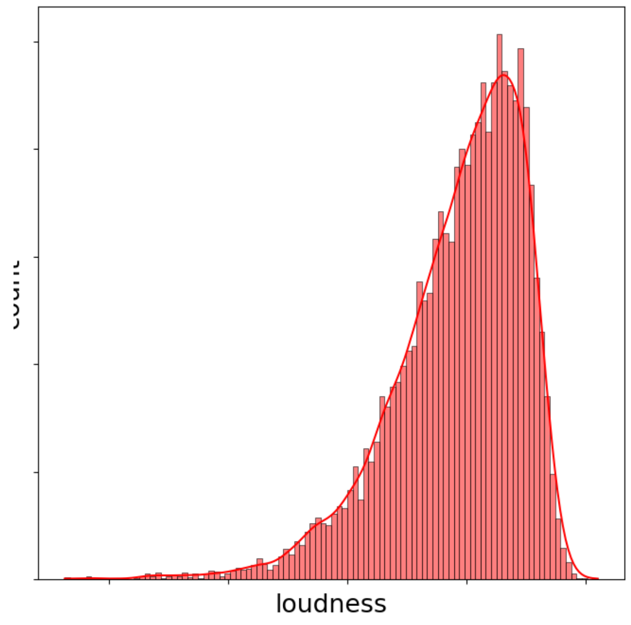

60
40
20
0
Duration in minutes

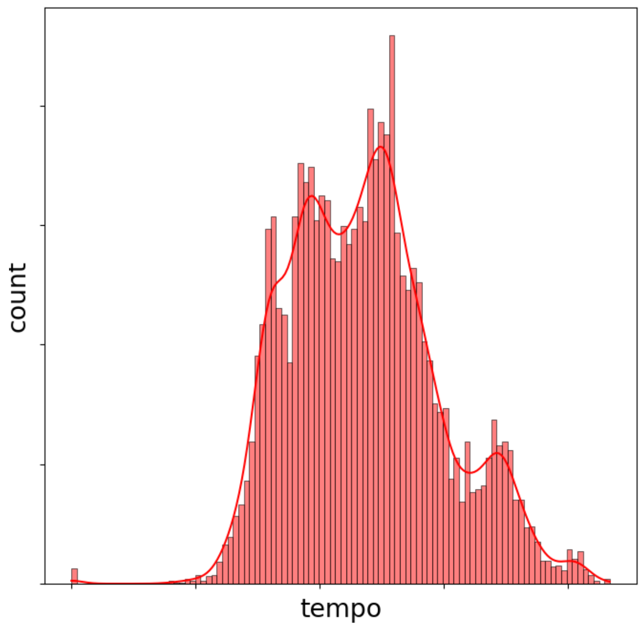

200
150
100
50
0
Tempo

4. Histogram Plots









5. Classification Report for Random Forest

1. The F-1 score of the model 0.57
2. The recall score of the model 0.56
3.

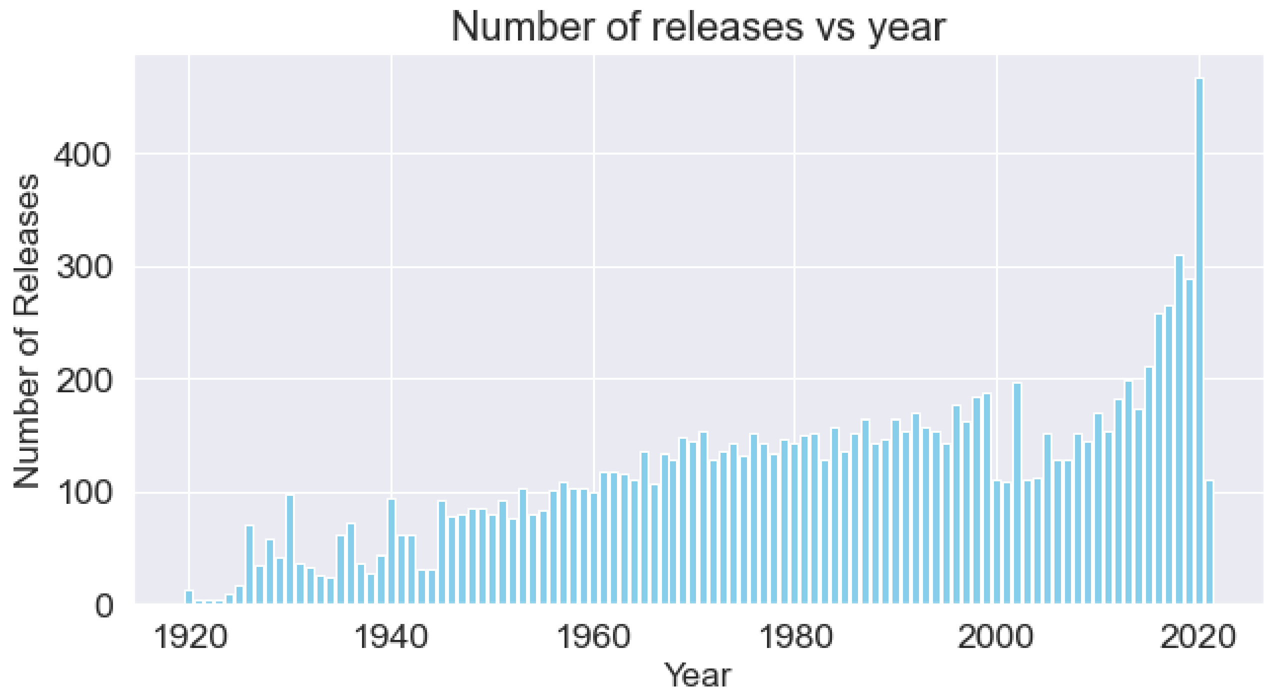
Class	Precision	Recall	f1-score	support
1	0.85	0.78	0.82	3222
2	0.56	0.71	0.63	3118
3	0.51	0.45	0.48	2912
4	0.61	0.58	0.59	2606
5	0.51	0.35	0.42	369

	Precision	Recall	f1-score	Support
macro average	0.61	0.57	0.58	12227
weighted average	0.63	0.63	0.63	12227

4. Confusion matrix

2516	303	97	291	15
310	2215	558	31	4
77	1077	1309	449	0
36	353	609	1501	107
9	11	13	206	109

Number of Releases Vs Year


Number of releases vs year
Number of Releases
Year