PROBLEM STATEMENT

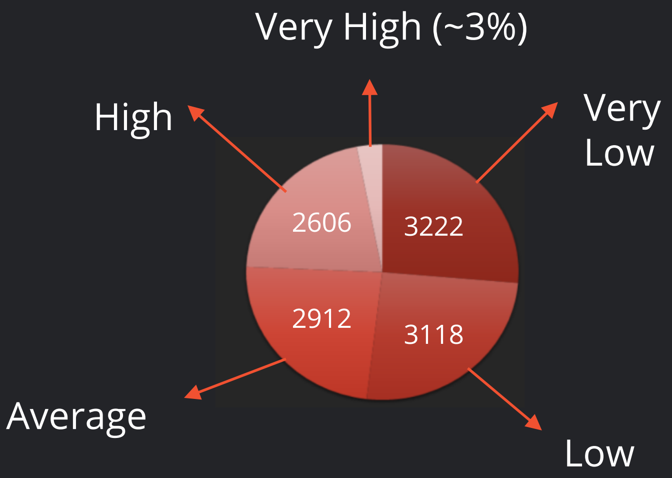
- Reduce losses by bidding correctly
- Predict popularity based on characteristics of music
- Bid valid if popularity predicted is higher or equal to actual

POPULARITY	BID PRICE	EXPECTED REVENUE (in 10k $)
VERY LOW	1	2
LOW	2	4
AVERAGE	3	6
HIGH	4	8
VERY HIGH	5	10

Acounsticness	Mode
Danceability	Release Date
Instrumentalness	Speechiness
Key	Tempo
Energy	Valence
Liveliness	Year
Explicit	Duration min
Loudness	Popularity

DATASET

NUMBER OF TRAINING EXAMPLES	12227
NUMBER OF FEATURES	16
CATEGORICAL FEATURES	2
NUMERIC FEATURES	14
MISSING VALUES	NONE
OUTLIERS	NONE REMOVED
DUPLICATE VALUES	NONE



Oversampling (SMOTE)

Class Imbalance

BASIC WORKFLOW

DATA
Visualization

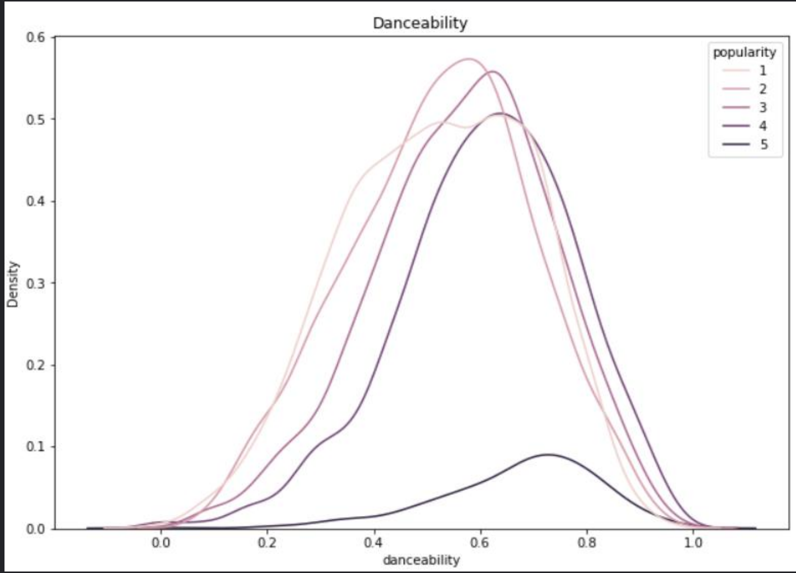
RESEARCH
Past Studies

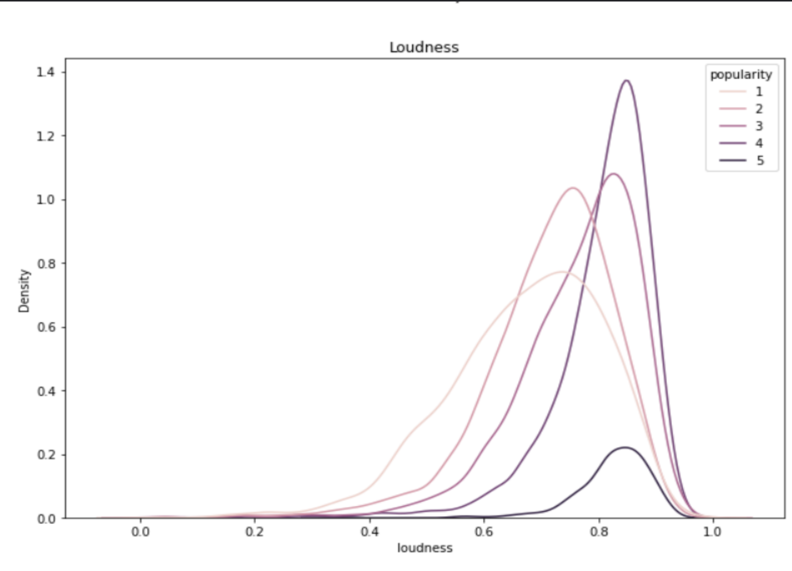
DATA
Feature Engineering
SMOTE

ANALYSIS
Model Selection

RESULT
Future Work

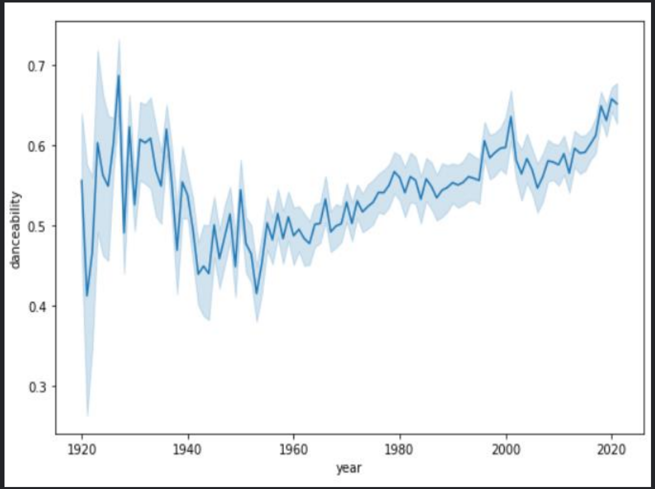
EXPLORATORY DATA ANALYSIS

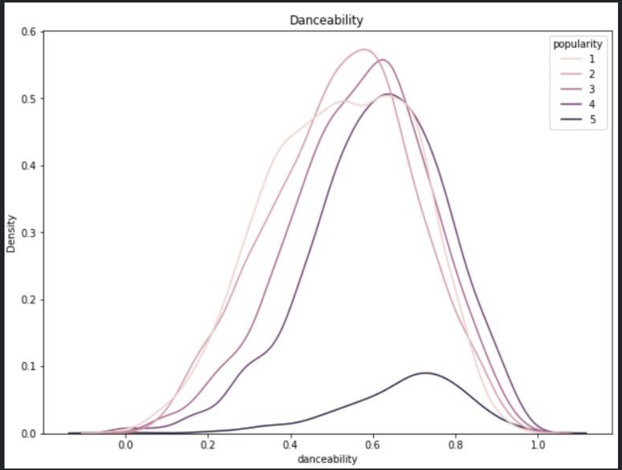


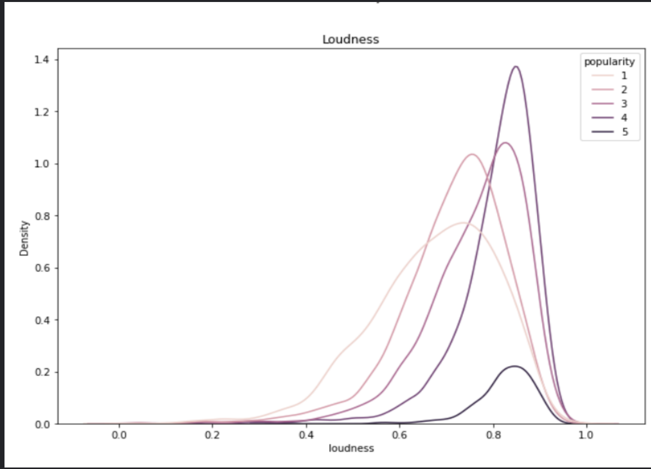


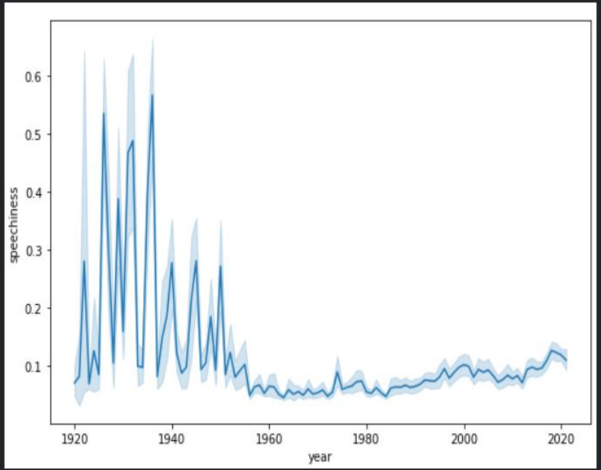
Danceability v/s Popularity

EXPLORATORY DATA ANALYSIS

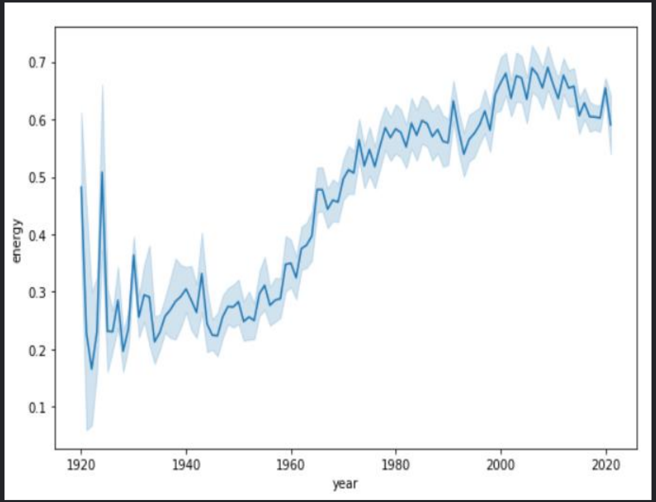


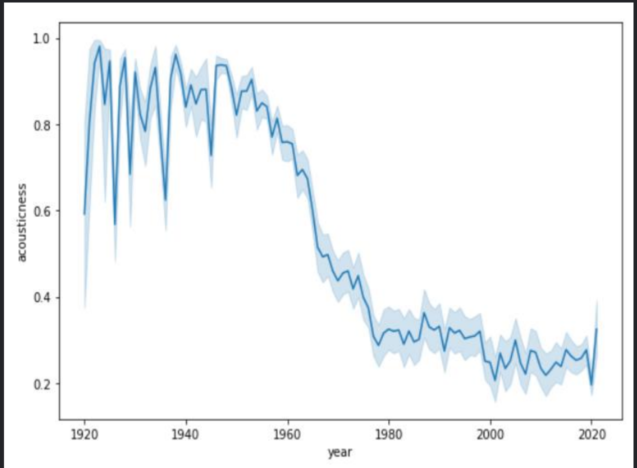


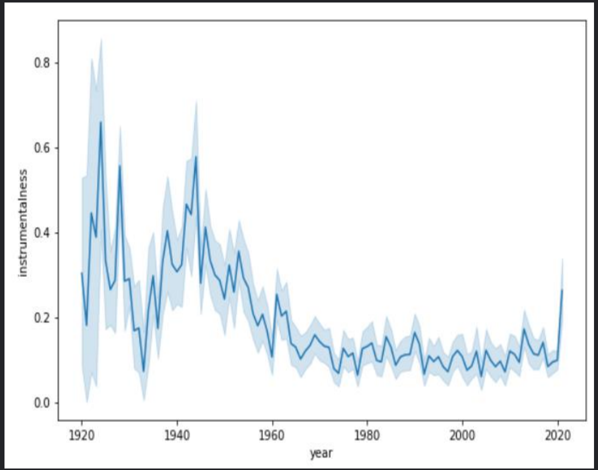


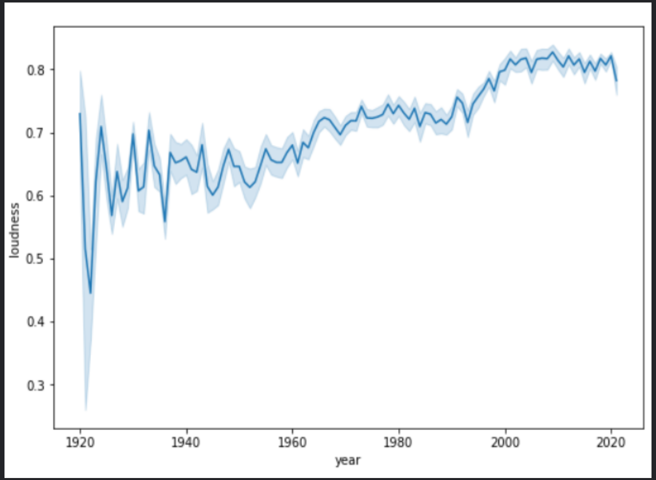


EXPLORATORY DATA ANALYSIS

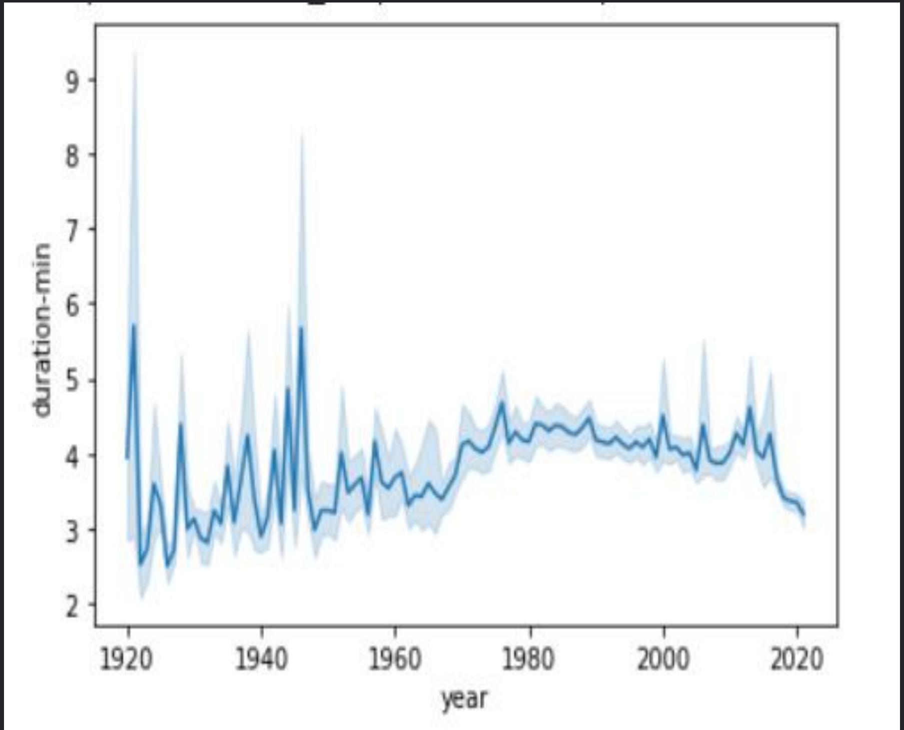


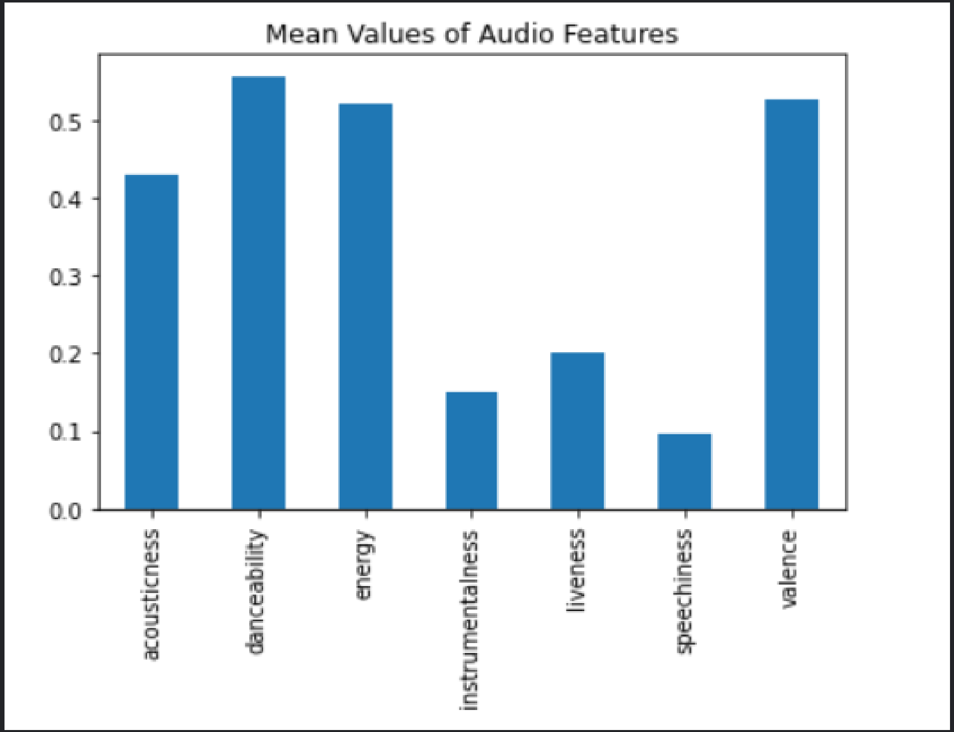




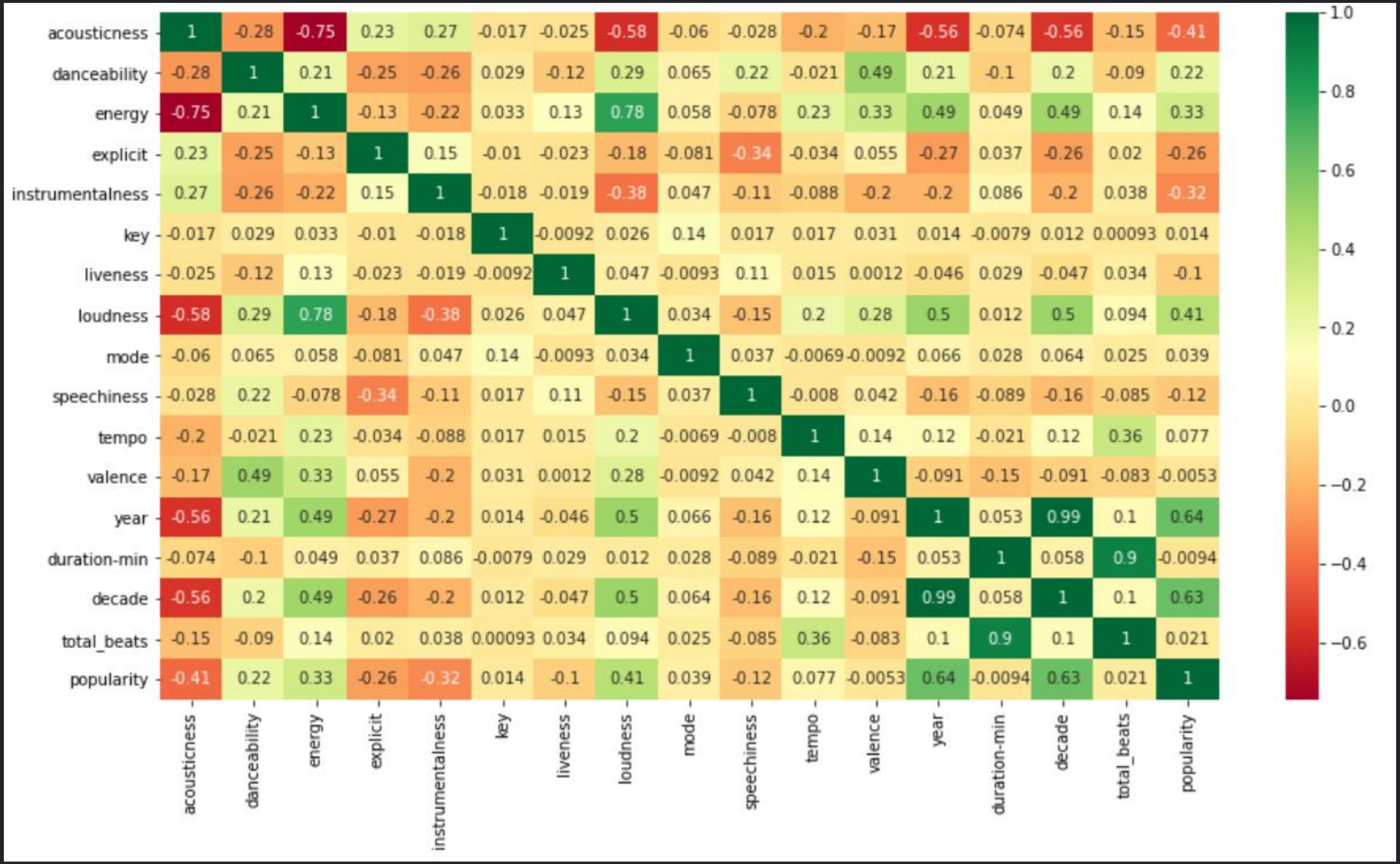


EXPLORATORY DATA ANALYSIS

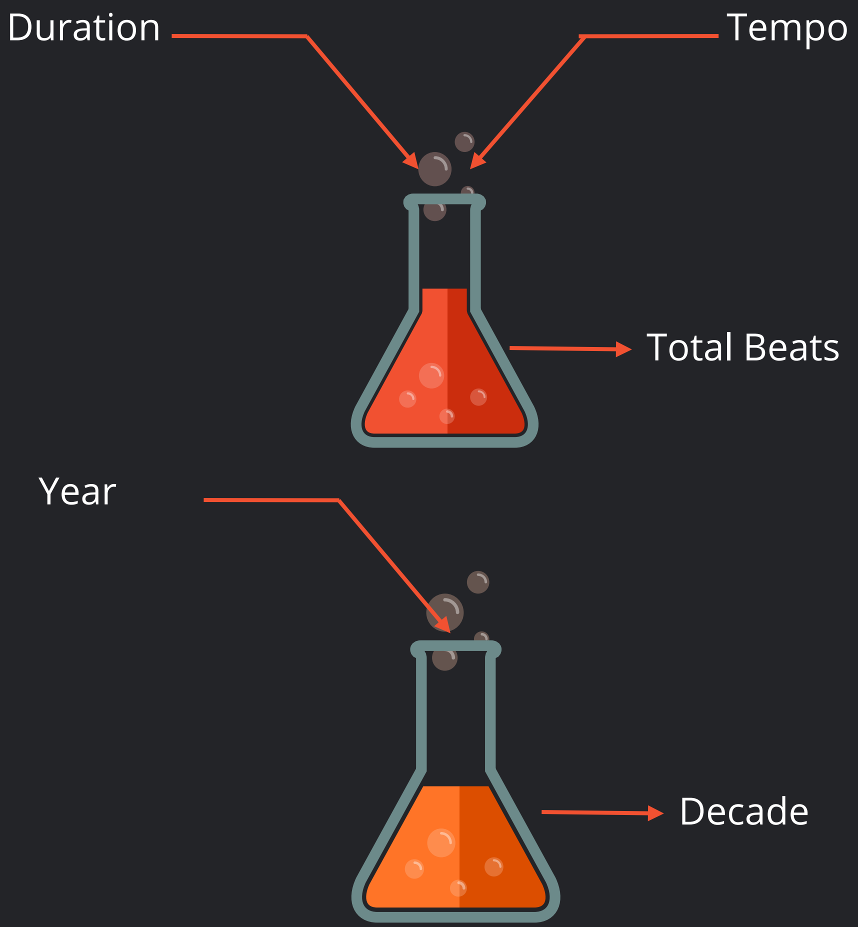




HEATMAP


acousticness 1 -0.28 -0.75 0.23 0.27 -0.017 -0.025 -0.58 -0.06 -0.028 -0.2 -0.17 -0.56 -0.074 -0.56 -0.15 -0.41
danceability -0.28 1 0.21 -0.25 -0.26 0.029 -0.12 0.29 0.065 0.22 -0.021 0.49 0.21 -0.1 0.2 -0.09 0.22
energy -0.75 0.21 1 -0.13 -0.22 0.033 0.13 0.78 0.058 -0.078 0.23 0.33 0.49 0.049 0.49 0.14 0.33
explicit 0.23 -0.25 -0.13 1 0.15 -0.01 -0.023 -0.18 -0.081 -0.34 -0.034 0.055 -0.27 0.037 -0.26 0.02 -0.26
instrumentalness 0.27 -0.26 -0.22 0.15 1 -0.018 -0.019 -0.38 0.047 -0.11 -0.088 -0.2 -0.2 0.086 -0.2 0.038 -0.32
key -0.017 0.029 0.033 -0.01 -0.018 1 -0.0092 0.026 0.14 0.017 0.017 0.031 0.014 -0.0079 0.012 0.00093 0.014
liveness -0.025 -0.12 0.13 -0.023 -0.019 -0.0092 1 0.047 -0.0093 0.11 0.015 0.0012 -0.046 0.029 -0.047 0.034 -0.1
loudness -0.58 0.29 0.78 -0.18 -0.38 0.026 0.047 1 0.034 -0.15 0.2 0.28 0.5 0.012 0.5 0.094 0.41
mode -0.06 0.065 0.058 -0.081 0.047 0.14 -0.0093 0.034 1 0.037 -0.0069 -0.0092 0.066 0.028 0.064 0.025 0.039
speechiness -0.028 0.22 -0.078 -0.34 -0.11 0.017 0.11 -0.15 0.037 1 -0.008 0.042 -0.16 -0.089 -0.16 -0.085 -0.12
tempo -0.2 -0.021 0.23 -0.034 -0.088 0.017 0.015 0.2 -0.0069 -0.008 1 0.14 0.12 -0.021 0.12 0.36 0.077
valence -0.17 0.49 0.33 0.055 -0.2 0.031 0.0012 0.28 -0.0092 0.042 0.14 1 -0.091 -0.15 -0.091 -0.083 -0.0053
year -0.56 0.21 0.49 -0.27 -0.2 0.014 -0.046 0.5 0.066 -0.16 0.12 -0.091 1 0.053 0.99 0.1 0.64
duration-min -0.074 -0.1 0.049 0.037 0.086 -0.0079 0.029 0.012 0.028 -0.089 -0.021 -0.15 0.053 1 0.058 0.9 -0.0094
decade -0.56 0.2 0.49 -0.26 -0.2 0.012 -0.047 0.5 0.064 -0.16 0.12 -0.091 0.99 0.058 1 0.1 0.63
total_beats -0.15 -0.09 0.14 0.02 0.038 0.00093 0.034 0.094 0.025 -0.085 0.36 -0.083 0.1 0.9 0.1 1 0.021
popularity -0.41 0.22 0.33 -0.26 -0.32 0.014 -0.1 0.41 0.039 -0.12 0.077 -0.0053 0.64 -0.0094 0.63 0.021 1
acousticness danceability energy explicit instrumentalness key liveness loudness mode speechiness tempo valence year duration-min decade total_beats popularity
1.0 0.8 0.6 0.4 0.2 0.0 −0.2 −0.4 −0.6

FEATURE ENGINEERING



WORKFLOW OF MODELS

1. Basic Model
2. Handling Class Imbalance
3. Feature Selection
4. Hyper-parameter Tuning
5. Overcoming Overfitting
6. Evaluation of Model

SUPPORT VECTOR CLASSIFICATION

Baseline Accuracy:

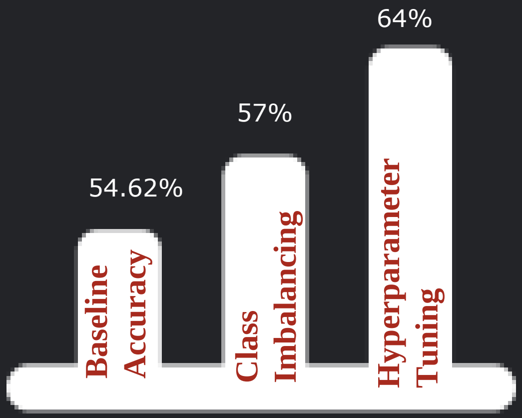
- Removed ID and Release Date
- Trained with default parameter
- Kernel used – Linear

Cross Validation Accuracy : 54.62 %

Class Imbalancing

- Applied SMOTE
- Trained with default parameter.
- Kernel used-Polynomial

Cross Validation Accuracy : 57 %



FEATURE SELECTION

Features Dropped	1.id 2.release_date 3.key 4.model
Features Added	total _beats (tempo*duration)

HYPERPARAMETER TUNING

Kernel	Polynomial
Degree of Polynomial	8
C	1
Cross Val Accuracy	64%

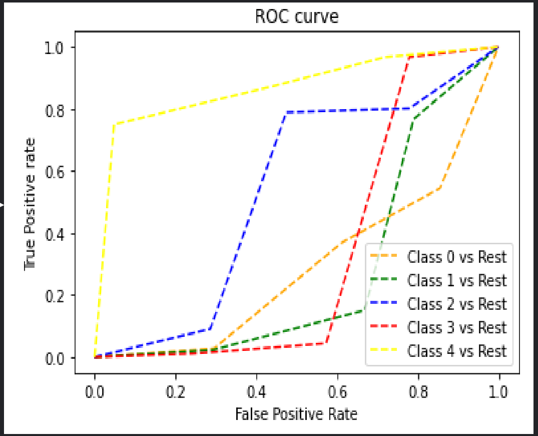
SUPPORT VECTOR CLASSIFICATION

CONFUSION MATRIX

278	192	182	0	39
118	453	70	0	15
173	19	444	0	169
2	85	3	0	1
13	183	41	0	577

	precision	recall	f1-score
0	0.48	0.40	0.44
1	0.49	0.69	0.57
2	0.60	0.55	0.57
3	0.00	0.00	0.00
4	0.72	0.71	0.71
Accuracy			0.57

ROC-AUC CURVE



XGBOOST

Baseline Accuracy:

- Removed ID and Release Date
- No oversampling

Cross Val Accuracy : 62.75 %

Class Imbalancing

- Applied SMOTE
- Removed ID, release date

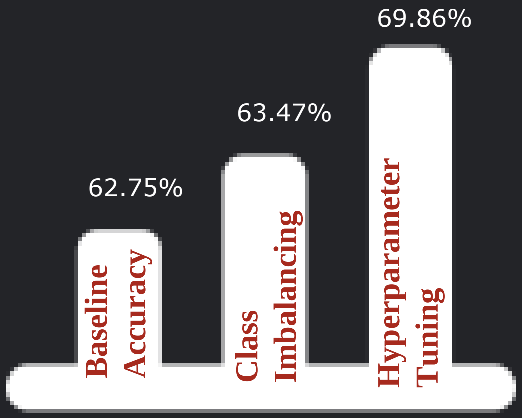
Cross Val Accuracy : 63.47 %

FEATURE SELECTION

Features Dropped	1.id 2.release_date 3.key 4.model
Features Added	-total _beats (tempo*duration) -decade

HYPERPARAMETER TUNING

colsample_bytree	0.7
gamma	0.3
learning_rate	0.15
max_depth	15
n_estimators	1000
min_child_weight	1
Cross Val Accuracy	69.86%



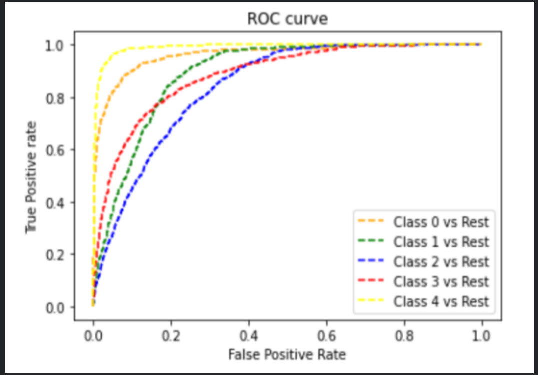
XGBOOST

CONFUSION MATRIX

766	94	39	46	10
103	605	194	17	0
35	309	490	137	7
30	106	203	529	123
4	3	11	45	97

	precision	recall	f1-score
0	0.817	0.802	0.809
1	0.542	0.658	0.594
2	0.523	0.501	0.512
3	0.683	0.534	0.599
4	0.869	0.936	0.901
Accuracy			0.686

ROC-AUC CURVE



LightGBM

Baseline Accuracy:

- Removed ID and Release Date
- boosting type : gdbt

Cross Val Accuracy : 62.00 %

Class Imbalancing

- Applied SMOTE
- Removed ID, Release Date
- boosting type : dart

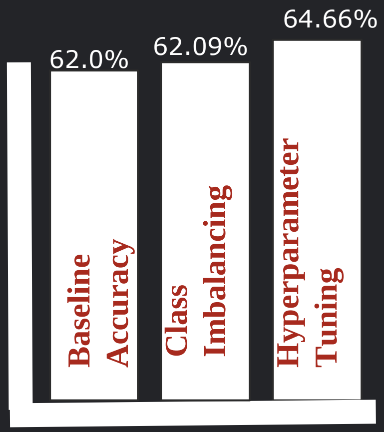
Cross Val Accuracy : 62.09 %

FEATURE SELECTION

Features Dropped	1.id 2.release_date 3.key 4.model 5.explicit

HYPERPARAMETER TUNING

learning_rate	0.01
early_stopping	30
lambda_l1	1.0
lambda_l2	1.0
num_boost_rounds	2000
Cross Val Accuracy=64.66%	



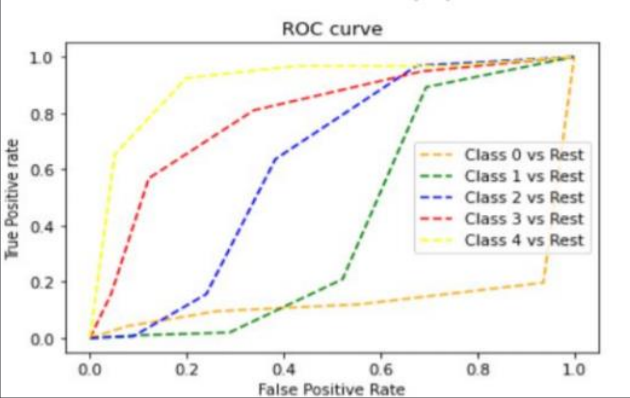
LightGBM

CONFUSION MATRIX

638	61	19	42	34
83	526	146	8	7
23	240	347	109	4
35	94	164	278	107
3	0	4	25	60

Class	Precision	Recall	f1-score
Very Low	0.816	0.800	0.810
Low	0.571	0.683	0.622
Average	0.510	0.480	0.495
High	0.602	0.410	0.488
Very High	0.283	0.652	0.604

ROC-AUC CURVE



ADABOOST

Baseline Accuracy:

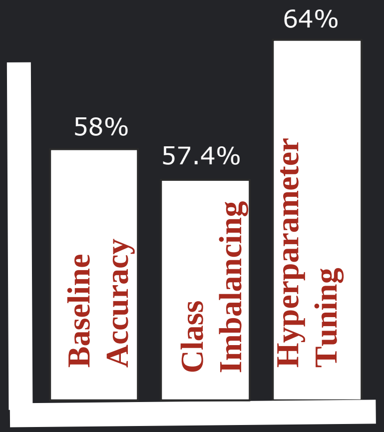
- Removed ID and Release Date
- No oversampling

Cross Val Accuracy : 59 %

Class Imbalancing

- Applied SMOTE
- Removed ID, Release Date

Cross Val Accuracy : 57.4 %



FEATURE SELECTION

Features Dropped	1.id 2.release_date 3.key 4.model 5.explicit
Features Added	total _beats (tempo*duration)

HYPERPARAMETER TUNING

n_estimators	learning _rate	Accuracy
500	0.1	60.88
100	0.2	60.57
100	0.3	61.01
100	0.4	68.60
Cross Val Accuracy=64%		

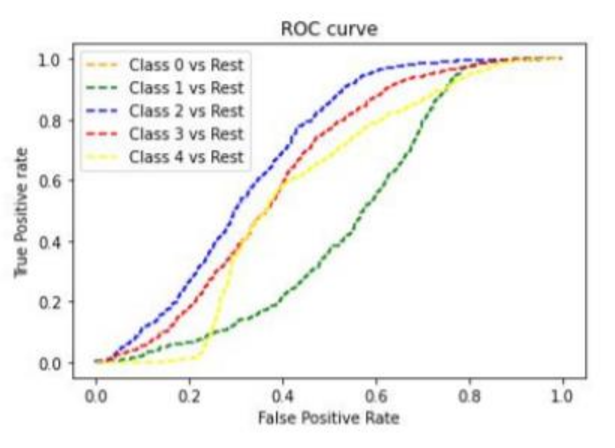
ADBOOST

CONFUSION MATRIX

530	67	22	51	9
68	408	137	12	0
17	190	309	94	0
11	64	117	401	55
1	3	7	24	624

Class	Precision	Recall	f1-score
Very Low	0.845	0.781	0.812
Low	0.557	0.653	0.601
Average	0.522	0.507	0.514
High	0.688	0.619	0.652
Very High	0.907	0.945	0.926

ROC-AUC CURVE



RANDOM FOREST CLASSIFIER

01 Features removed: key, mode, explicit, ID, release_date

02 Oversampling resulted in overfitting

03 Applied repeated stratified k-fold cross validation

04 Hyperparameters tuned

Accuracy of baseline model = ~50%

RANDOM FOREST CLASSIFIER

HYPERPARAMETER TUNING

NO. OF ESTIMATORS	MEAN ACCURACY
100	~68%
500	~70%
500-900	~70%(run time was increasing)

HANDLING CLASS IMBALANCE

TOOLS USED	MEAN ACCURACY
Cost-sensitive learning	~50%
Default SMOTE oversampling	~68%

OVERSAMPLING STRATEGY

NO. OF EXAMPLES IN EACH CLASS	MEAN ACCURACY
3700	~72%
4000	~74%
5000	~77%

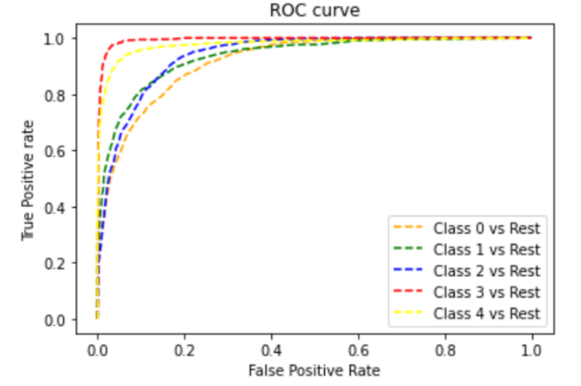
RANDOM FOREST CLASSIFIER

CONFUSION MATRIX

639	103	203	0	23
147	718	85	66	10
151	10	727	0	69
8	32	8	991	0
28	41	68	8	865

	precision	recall	f1-score
0	0.657	0.660	0.658
1	0.794	0.700	0.744
2	0.666	0.760	0.710
3	0.931	0.954	0.942
4	0.895	0.856	0.875

ROC-AUC CURVE



FINAL HYPERPARAMETERS

HYPERPARAMETERS	TUNED VALUE
Estimators	100
Splits	10
Repeats	3
Cross Val Accuracy	~79%

NEURAL NETWORK

MODEL 1

Features removed	Explicit, release_date, tempo, key, year
New Features	Total Beats, Duration-min
Scaling	Standard Scaler
Oversampling	SMOTE

Number of hidden layers	3 (5->32->16->8->5)
Activation in hidden layers	relu
Activation in output layer	softmax
Loss function	Categorical cross entropy
optimizer	Adam
batch size	5
epochs	2000
Cross validation accuracy	53%

NEURAL NETWORK

MODEL 2

Features removed	Explicit, release_date, tempo, key, year
New Features	Total Beats, Duration-min
Scaling	Standard Scaler
Oversampling	SMOTE

Number of hidden layers	5 (5->128->64->32->16->8->5)
Activation in hidden layers	relu
Activation in output layer	softmax
Loss function	Categorical cross entropy
optimizer	Adam
batch size	5
epochs	2000
Cross validation accuracy	65%

COMPARISON

SUPPORT VECTOR MACHINE	XGBOOST	LightGBM	ADABoost	RANDOM FOREST	NEURAL NETWORK
64%	69%	64%	68%	79%	65%

WHY					
- **Kernel Trick** - Don't need to hand-perform complex transformation - Can capture non linear relationships due to this	-**Boosting** - Has been observed to give excellent accuracy - Minimal data preprocessing required - Faster runtime	- Very fast execution compared to XGB - Leaf wise growth, can lead to better accuracy	- **Boosting** has been observed to give superior performance	- The basis is Bagging - Bagging reduces variance and solves overfitting	- Neural Networks are great at capturing patterns and generating decision boundaries - Might eliminate the need to hand engineer features

WHY and WHY NOT?

WHY DON'T THESE WORK WELL?	
SVM	Impacted greatly by correct choice of hyperparameters Hyperparameter tuning is tricky and time consuming Overfitting due to Oversampling Possibly due to presence of outliers
Boosting Algorithms	Boosting algorithms are prone to overfitting Overfitting due to Oversampling
Neural Networks	Overfits due to oversampling and relatively smaller dataset

WHY DOES RANDOM FOREST WORK WELL?

- Random forest - Bagging algorithm
- Biggest problem - overfitting due to oversampling
- Bagging reduces variance - model less prone to overfitting
- Robust to outliers, requires min preprocessing
- Faster runtime allows extensive hyperparameter tuning

FUTURE WORK

1. Alternate methods for dealing with class imbalance
2. Data about song artists and producers
3. Careful removal of univariate and multivariate outliers
4. Meticulous hyperparameter tuning, especially for SVM

THANK YOU!