PREDICTION OF MUSIC GENRES

Pere Arnau Alegre
Andrés Jiménez González
Victor Teixidó López
Max Vives Ribera
You Wu

INDEX

INTRODUCTION

We got our data by searching, in the kaggle website, classification datasets.

Music genres are a common interest among the members of the group.

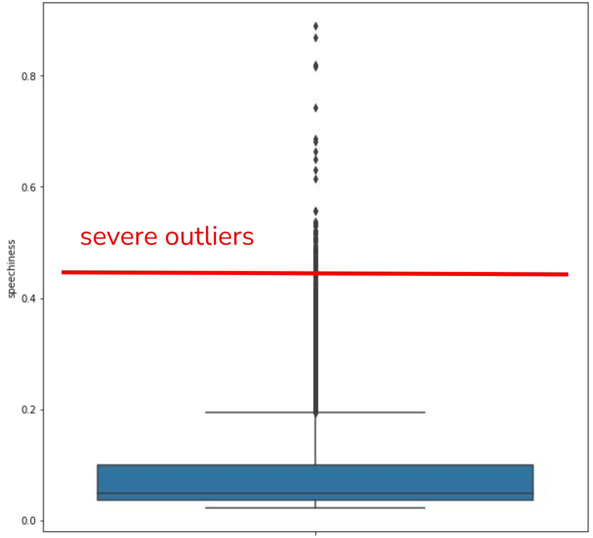
METADATA

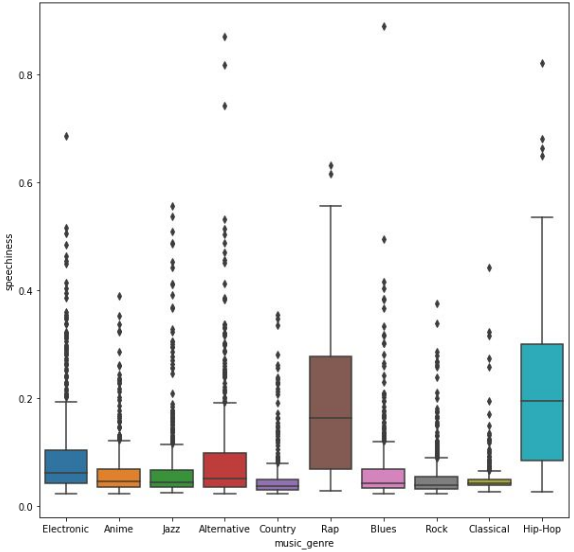
- **Music genre:** ‘Electronic’, ‘Anime’, ‘Jazz’, ‘Alternative’, ‘Country’, ‘Rap’, ‘Blues’, ‘Rock’, ‘Classical’ and ‘Hip-Hop’
- 5000 individuals
- 16 variables: artist_name, track_name, duration_s, key, mode, tempo, popularity, valence, acousticness, danceability, energy, instrumentalness, liveness, loudness, speechiness, music_genre.
- Common used features & Spotify features.

PREPROCESSING

For every variable, we checked for severe outliers, error and missings. We also performed a bit of profiling with our target variable.

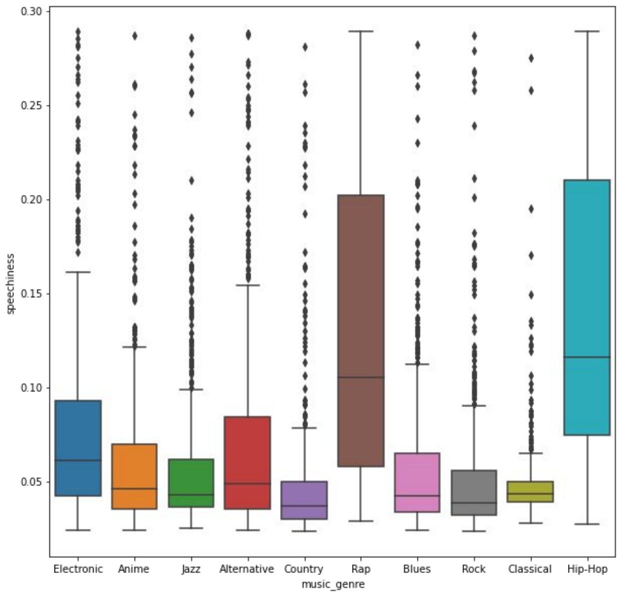
Example:
Variable
speechiness



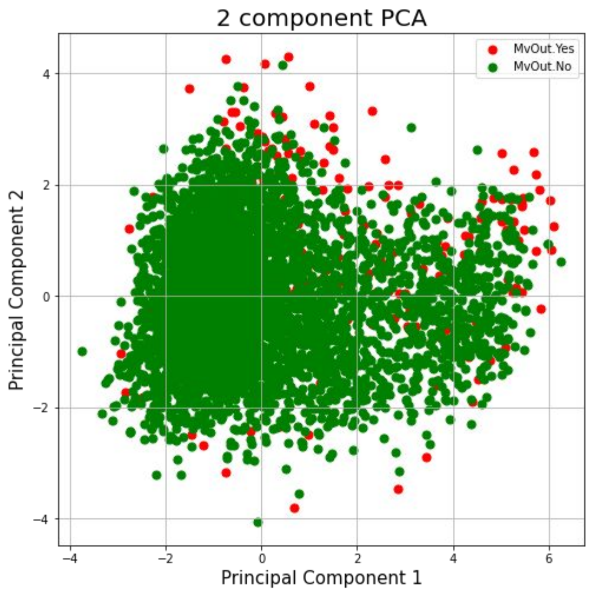


PREPROCESSING

We imputed missing values with an Iterative Imputer and we checked for consistency.



We determined multivariate outliers using Mahalanobis distance and we created a variable to tag them.



MACHINE LEARNING METHODS

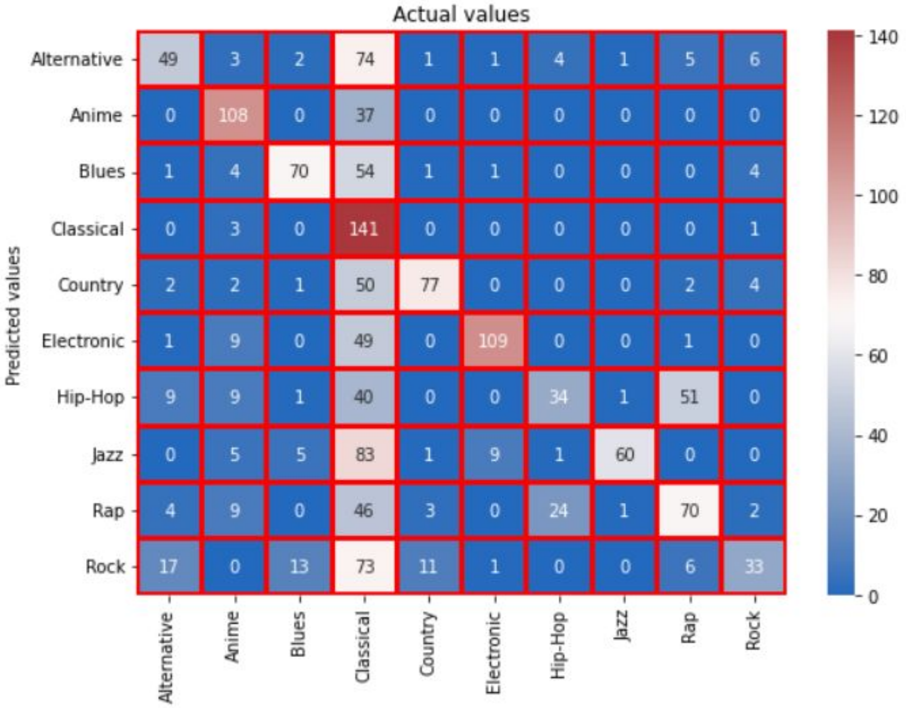
- NAÏVE-BAYES
- K-NN
- DECISION TREES
- META-LEARNING ALGORITHMS
- SUPPORT VECTOR MACHINES

NAÏVE-BAYES

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

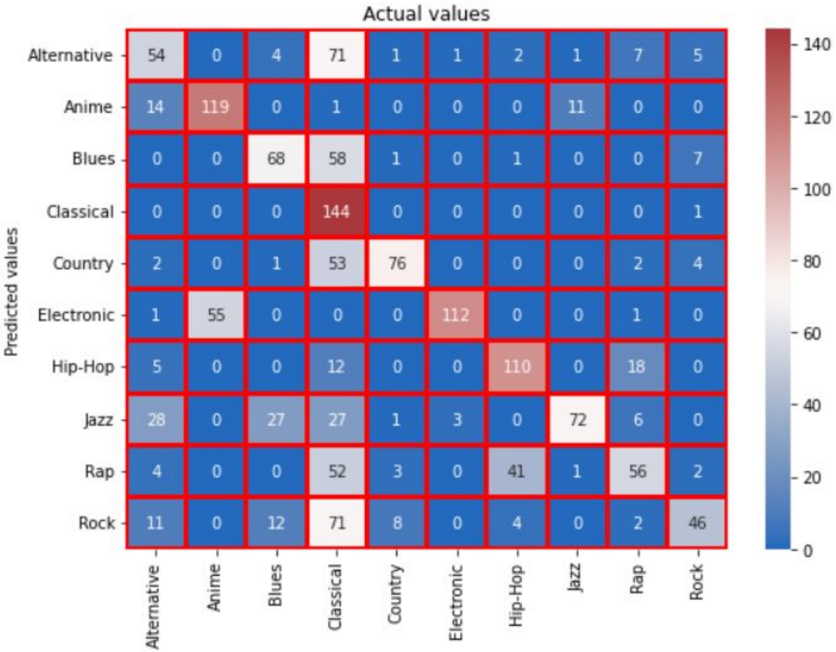
$$P(y|x1, x2, x3..xN) = \frac{P(x1|y).P(x2|y).P(x3|y)\ldots P(xN|y).P(y)}{P(x1).P(x2).P(x3)\ldots P(xN)}$$

NAÏVE-BAYES with multivariate outliers



	precision	recall	f1-score	support
Alternative	0.59	0.34	0.43	146
Anime	0.71	0.74	0.73	145
Blues	0.76	0.52	0.62	135
Classical	0.22	0.97	0.36	145
Country	0.82	0.56	0.66	138
Electronic	0.90	0.64	0.75	169
Hip-Hop	0.54	0.23	0.33	145
Jazz	0.95	0.37	0.53	164
Rap	0.52	0.44	0.48	159
Rock	0.66	0.21	0.32	154
accuracy			0.50	1500
macro avg	0.67	0.50	0.52	1500
weighted avg	0.67	0.50	0.52	1500

NAÏVE-BAYES without multivariate outliers



	precision	recall	f1-score	support
Alternative	0.45	0.37	0.41	146
Anime	0.68	0.82	0.75	145
Blues	0.61	0.50	0.55	135
Classical	0.29	0.99	0.45	145
Country	0.84	0.55	0.67	138
Electronic	0.97	0.66	0.79	169
Hip-Hop	0.70	0.76	0.73	145
Jazz	0.85	0.44	0.58	164
Rap	0.61	0.35	0.45	159
Rock	0.71	0.30	0.42	154
accuracy			0.57	1500
macro avg	0.67	0.57	0.58	1500
weighted avg	0.68	0.57	0.58	1500

NAÏVE-BAYES Conclusions

- Despite of being a very simple model and assuming a false hypothesis gets decent results.
- Is very influenced by outliers.

K-NN

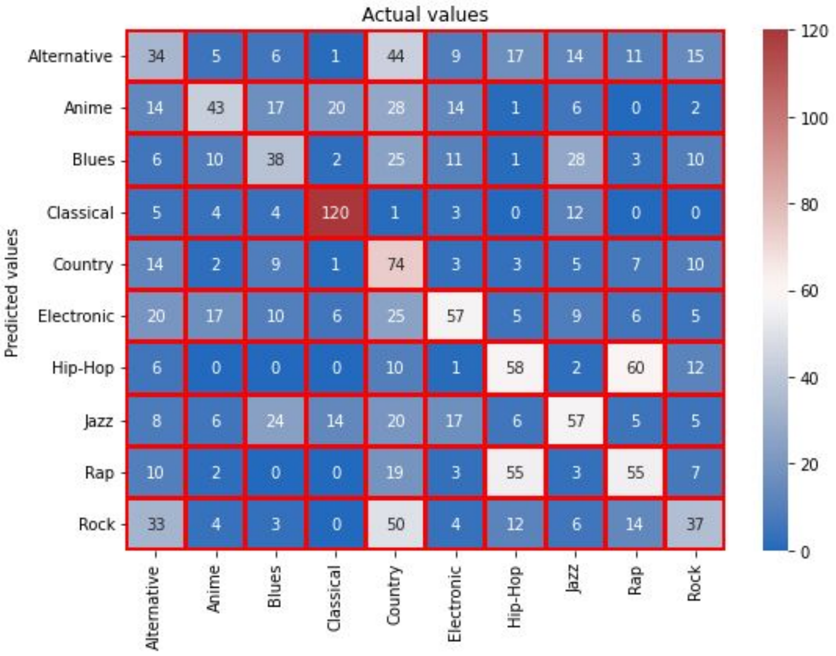
Advantages:

- No training period
- Addition of new data without drawbacks
- Easy implementation

Disadvantages:

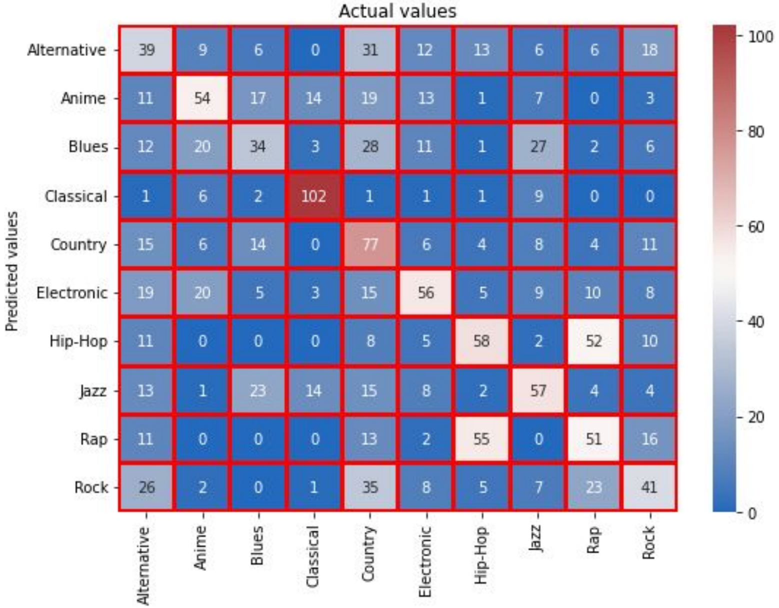
- Does not work well with large datasets
- Does not work well with high dimensions
- Feature scaling
- Sensitive to noisy data, missing values and outliers

K-NN with multivariate outliers




```
              precision    recall  f1-score   support

 Alternative       0.23      0.22      0.22       156
       Anime       0.46      0.30      0.36       145
       Blues       0.34      0.28      0.31       134
   Classical       0.73      0.81      0.77       149
     Country       0.25      0.58      0.35       128
  Electronic       0.47      0.36      0.40       160
     Hip-Hop       0.37      0.39      0.38       149
        Jazz       0.40      0.35      0.38       162
         Rap       0.34      0.36      0.35       154
        Rock       0.36      0.23      0.28       163

    accuracy                           0.38      1500
   macro avg       0.39      0.39      0.38      1500
weighted avg       0.40      0.38      0.38      1500

Interval of confidence: (0.357639282378951, 0.40698172923371395)

Accuracy: 0.382
```

K-NN without outliers




```
              precision    recall  f1-score   support

 Alternative       0.25      0.28      0.26       140
       Anime       0.46      0.39      0.42       139
       Blues       0.34      0.24      0.28       144
   Classical       0.74      0.83      0.78       123
     Country       0.32      0.53      0.40       145
  Electronic       0.46      0.37      0.41       150
     Hip-Hop       0.40      0.40      0.40       146
        Jazz       0.43      0.40      0.42       141
         Rap       0.34      0.34      0.34       148
        Rock       0.35      0.28      0.31       148

    accuracy                           0.40      1424
   macro avg       0.41      0.41      0.40      1424
weighted avg       0.40      0.40      0.40      1424

Interval of confidence: (0.374271686687884, 0.4255455517365586)

Accuracy: 0.39957865168539325
```

Finding best parameters to use

Parameters used in GridSearchCV:

- N_neighbors: list(range(1,30,2))
- Metric values: (‘euclidean’, ‘manhattan’, ‘chebyshev’, ‘minkowski’)
- Weight values: (distance, uniform)

Results:

- Best Params= {'metric': 'manhattan', 'n_neighbors': 7, 'weights': 'distance'}

Conclusions

	precision	recall	f1-score	support
Alternative	0.25	0.28	0.26	140
Anime	0.46	0.39	0.42	139
Blues	0.34	0.24	0.28	144
Classical	0.74	0.83	0.78	123
Country	0.32	0.53	0.40	145
Electronic	0.46	0.37	0.41	150
Hip-Hop	0.40	0.40	0.40	146
Jazz	0.43	0.40	0.42	141
Rap	0.34	0.34	0.34	148
Rock	0.35	0.28	0.31	148
accuracy			0.40	1424
macro avg	0.41	0.41	0.40	1424
weighted avg	0.40	0.40	0.40	1424

Interval of confidence: (0.374271686687884, 0.4255455517365586)

Accuracy: 0.39957865168539325

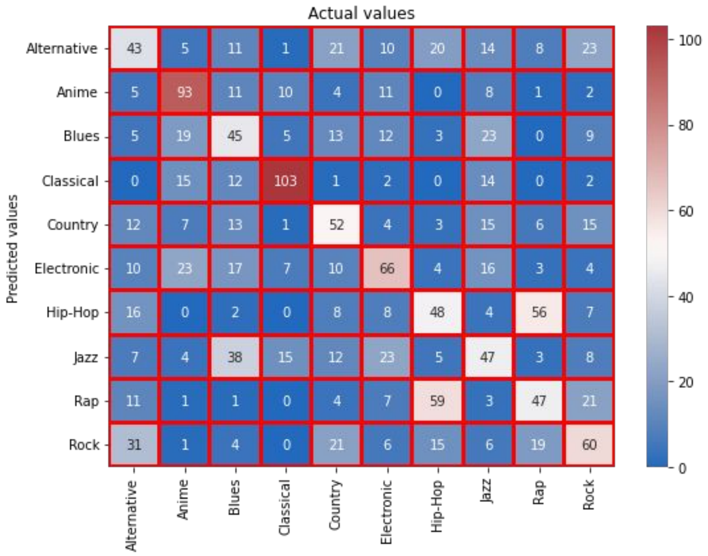
DECISION TREES

Decision Trees are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

- Simple to understand and to interpret.
- The cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the tree.
- Decision-tree learners can create over-complex trees that do not generalize the data well → overfitting.
- Decision trees can be unstable because small variations in the data might result in a completely different tree being generated.

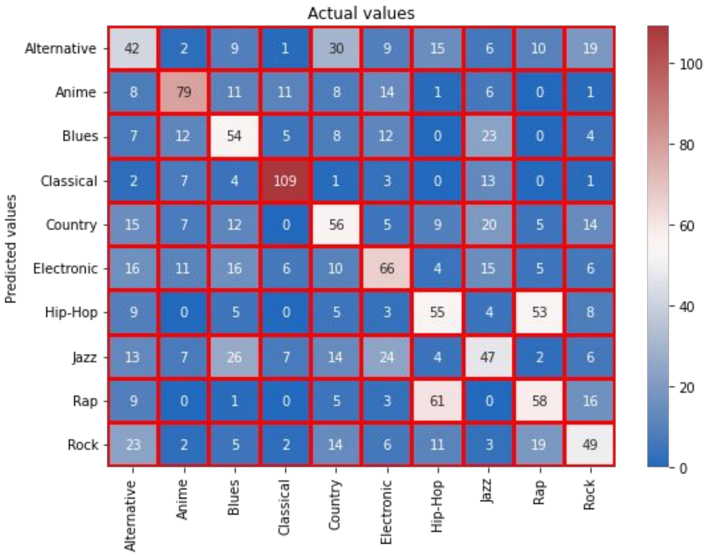
Model with or without outliers

Without outliers



Accuracy = 0.432

With outliers

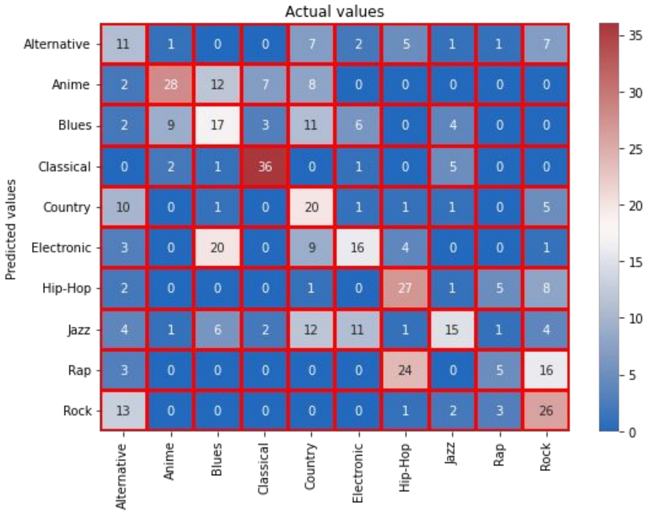


Accuracy = 0.407

Optimizing the decision tree

To reduce overfitting: Reduced the testing sample.

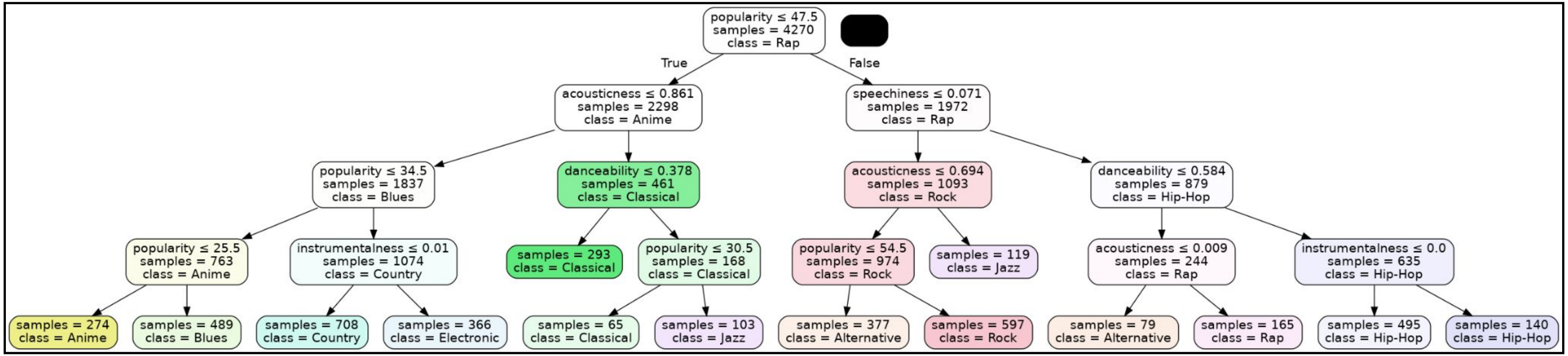
To achieve a readable tree: Added depth and impurity control.



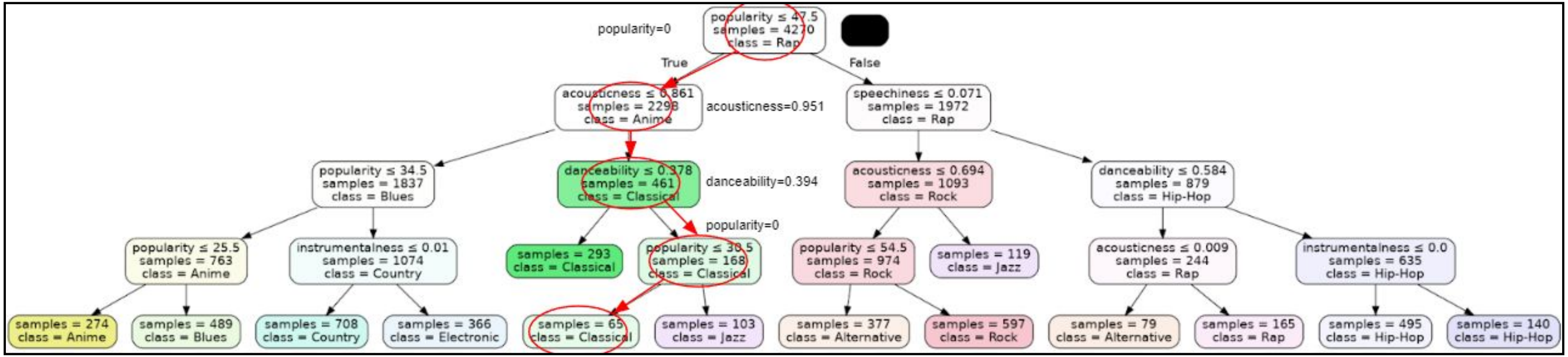

```
Accuracy: 0.4231578947368421

              precision    recall  f1-score   support

 Alternative       0.22      0.31      0.26        35
       Anime       0.68      0.49      0.57        57
       Blues       0.30      0.33      0.31        52
   Classical       0.75      0.80      0.77        45
     Country       0.29      0.51      0.37        39
  Electronic       0.43      0.30      0.36        53
     Hip-Hop       0.43      0.61      0.50        44
        Jazz       0.52      0.26      0.35        57
         Rap       0.33      0.10      0.16        48
        Rock       0.39      0.58      0.46        45

    accuracy                           0.42       475
   macro avg       0.43      0.43      0.41       475
weighted avg       0.45      0.42      0.41       475

Interval of confidence: (0.3788671374867574, 0.4684009521881719)
```

	Feature	Importance
0	popularity	0.618
1	acousticness	0.192
2	speechiness	0.087
3	instrumentalness	0.062
4	danceability	0.041

Resulting decision tree



Example of usage



Conclusions

- Decision trees are a simple and understandable model with very simple rules to predict categories.
- Due to its ease of development, it can produce big problems like overfitting and low accuracy. Those can be fixed by tweaking some parameters like the maximum depth or the minimum impurity decrease.
- The low accuracy of our model was due to the lack of highly correlated variables with the music genre and to the fact that some genres had very similar features.

META-LEARNING ALGORITHMS

- Voting Scheme
- Bagging
- Random Forest
- Boosting

Voting Scheme

- Naïve Bayes, K-nn, Decision Tree
- Weighted Voting accuracy: 0.437
- Majority Voting accuracy: 0.428

Bagging

- Decision Tree with various n_estimators
- *max_features = 1.0 (default)*
    - Best accuracy: 0.537
- *max_features = 0.35*
    - Best accuracy: 0.504

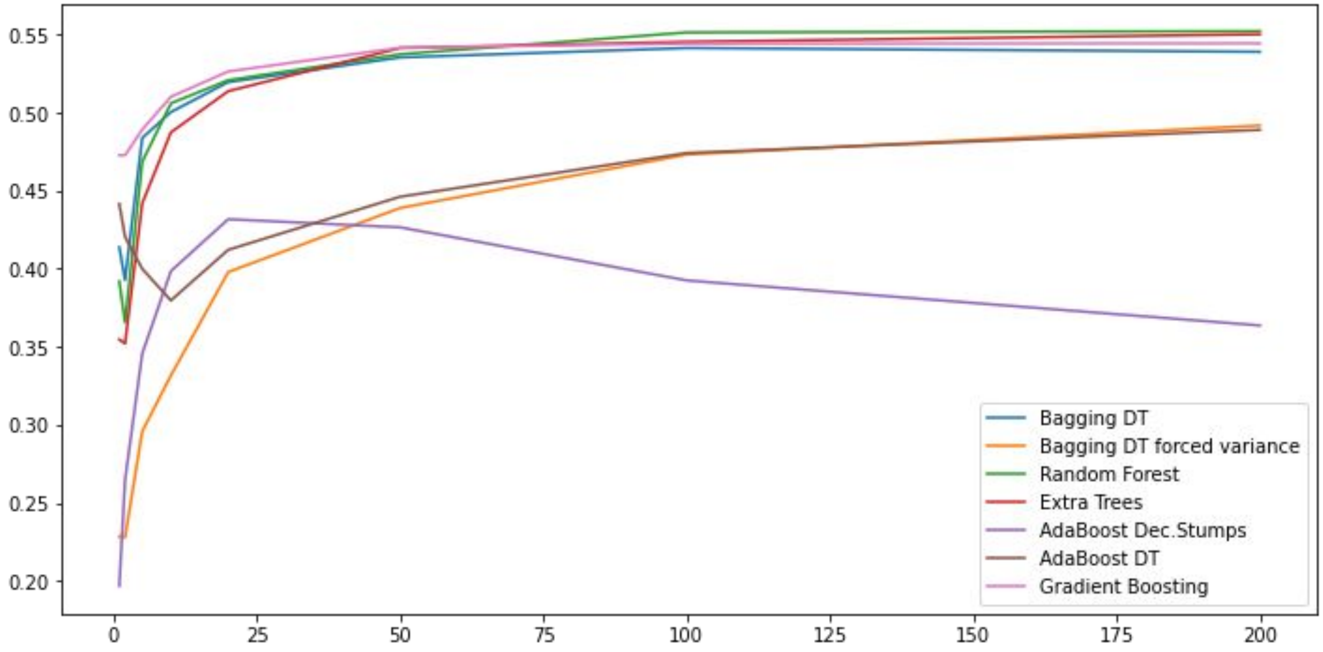
Random Forest

- With different value of n_estimators
    - Best accuracy: 0.552
- Extra Trees methods: accuracy 0.55

Boosting

- Ada Boost
    - *max_depth = None*: best accuracy 0.432
    - *max_depth = 5*: best accuracy 0.489
- Gradient Boosting
    - Best accuracy 0.544

Conclusion of Meta-learning methods

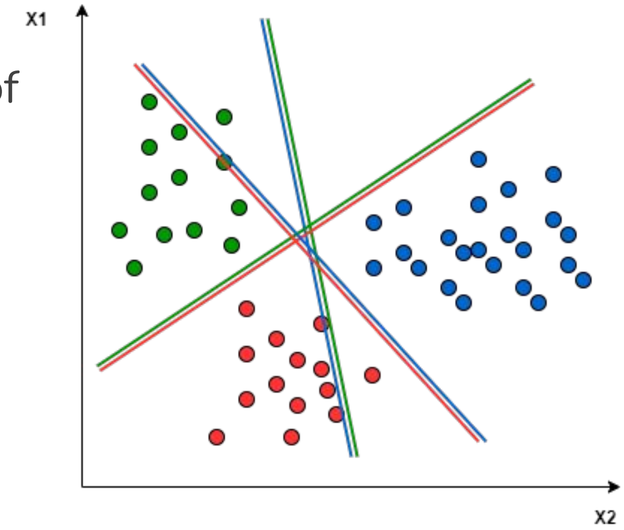


SUPPORT VECTOR MACHINES

Representation of the sample points in the space and separation of every 2 classes at maximum distance possible through a support-vector.

Vector is defined by the 2 closest points of each class.

Apply SVMs without outliers & with outliers.

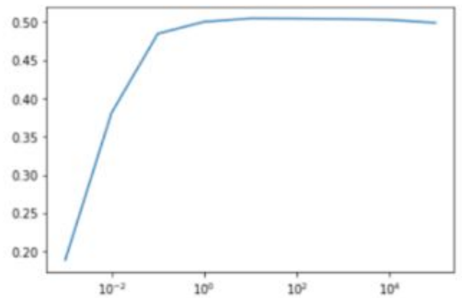


SVMs without outliers

LINEAR KERNEL

Default Linear kernel accuracy: 0.49

Find best C value and use it!



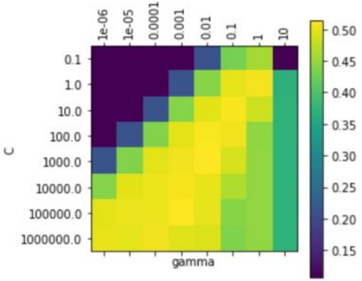

```
Acc. 5-fold cross on train data=  0.504481311
```

```
Confusion matrix on test set:
 [[37  2  1  0 12  9 10  5  5 12]
 [ 2 51  9 15  4  7  0  1  0  3]
 [ 1 11 43  2 13  7  1 11  0  6]
 [ 1  3  3 71  1  2  0  4  0  1]
 [ 7  4 10  0 44  3  4  7  2 17]
 [11 10  6  2  6 50  3  5  0  3]
 [ 8  0  0  0  4  0 35  3 36 10]
 [ 4  4 12  6  9 14  2 40  1  4]
 [ 9  0  0  0  2  0 40  0 33 17]
 [11  1  0  0  6  1  2  2 11 62]]

Accuracy on test set:  0.4910432033719705

Best value of parameter C found:  {'C': 10.0}

Number of supports:  3271 (
Prop. of supports:  0.8616965226554267
```

RBF KERNEL

Default RBF kernel accuracy: 0.508

Find best C and Gamma values and use them!




```
Confusion matrix on test set:
 [[37  1  3  0 14  3 12  5  5 13]
 [ 2 50 11 14  5  6  0  1  0  3]
 [ 1  7 45  4 12  6  0 12  0  8]
 [ 2  3  2 73  1  0  0  5  0  0]
 [ 4  2 11  0 43  4  2  9  5 18]
 [ 8 14  4  1  7 51  1  5  2  3]
 [ 5  0  0  0  3  1 40  1 40  6]
 [ 4  3  9  4 12 18  2 41  1  2]
 [ 6  0  0  0  3  0 36  0 42 14]
 [13  1  0  0  4  1  4  2  8 63]]

Accuracy on test set:  0.5110642781875658

Number of supports:  3179 ( 7892 of them have slacks)
Prop. of supports:  0.8374604847207587
```

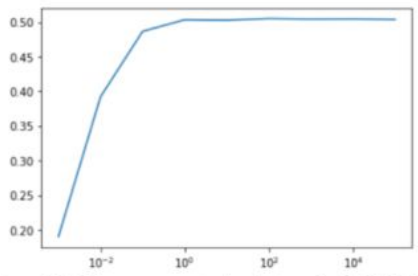
```
Best combination of parameters found:  {'C': 100000.0, 'gamma': 0.001}

Acc. 5-fold cross on train data=  0.5102714763543117
```

SVMs with outliers

LINEAR KERNEL

Default Linear kernel accuracy: 0.51

Find best C value and use it!



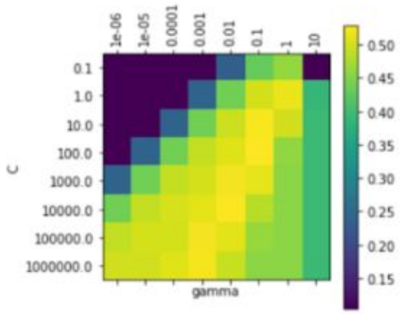

```
Acc. 5-fold cross on train data=  0.50513110137
```

```
Confusion matrix on test set:
 [[31  1  3  0 18  8  7  7  6 14]
 [ 3 60 10  7  6  6  0  3  1  1]
 [ 4 15 38  4 10  8  1 17  0  3]
 [ 0  5  1 84  1  3  0  5  0  0]
 [ 4  2  7  0 45  9  4 12  1 14]
 [ 8  9  3  0  3 63  3 11  4  1]
 [ 6  0  0  0  6  3 40  2 35  7]
 [ 0  4 14  8 12 16  3 44  0  3]
 [ 7  0  0  0  3  2 34  0 44 14]
 [13  0  1  0  6  1  7  5  3 63]]

Accuracy on test set:  0.512

Best value of parameter C found:  {'C': 100.0}

Number of supports:  3407 ( 9737 of them have slacks)
Prop. of supports:  0.8519629907476869
```

RBF KERNEL

Default RBF kernel accuracy: 0.503

Find best C and Gamma values and use them!




```
Confusion matrix on test set:
 [[33  2  3  1 16  3 10  7  5 15]
 [ 4 58 11  9  5  7  0  2  1  0]
 [ 7 12 41  4 12  5  0 17  0  2]
 [ 1  4  0 84  1  1  0  7  0  1]
 [ 9  2  8  0 49  5  5  9  2  9]
 [11  8  3  0  5 59  1  9  7  2]
 [ 5  0  0  0  5  2 41  3 39  4]
 [ 4  3 15  9 12 13  1 44  0  3]
 [ 7  0  0  0  2  1 36  0 48 10]
 [11  0  2  0 11  2  5  5  5 58]]

Accuracy on test set:  0.515

Number of supports:  3271 ( 6537 of them have slacks)
Prop. of supports:  0.8179544886221556
```

```
Best combination of parameters found:  {'C': 100.0, 'gamma': 0.1}

Acc. 5-fold cross on train data=  0.5296357947434294
```

Conclusions of SVMs

Choosing the data set with outliers and RBF kernel is the best option.

Computation time little bit worse than Linear kernel but almost the same.

Accuracy has better results: 0.515.

```
Confusion matrix on test set:
 [[33  2  3  1 16  3 10  7  5 15]
 [ 4 58 11  9  5  7  0  2  1  0]
 [ 7 12 41  4 12  5  0 17  0  2]
 [ 1  4  0 84  1  1  0  7  0  1]
 [ 9  2  8  0 49  5  5  9  2  9]
 [11  8  3  0  5 59  1  9  7  2]
 [ 5  0  0  0  5  2 41  3 39  4]
 [ 4  3 15  9 12 13  1 44  0  3]
 [ 7  0  0  0  2  1 36  0 48 10]
 [11  0  2  0 11  2  5  5  5 58]]

Accuracy on test set:  0.515

Number of supports:  3271 ( 6537 of them have slacks)
Prop. of supports:  0.8179544886221556
```

COMPARISON BETWEEN MODELS

Model	Accuracy	Precision	Recall	F1-Score
Naive Bayes	0.57	0.67	0.58	0.58
KNN	0.399	0.40	0.40	0.40
Decision Trees	0.423	0.43	0.43	0.41
Support Vector Machines	0.515	-	-	-
Voting scheme	0.437	-	-	-
Bagging	0.537	-	-	-
Random Forest	0.552	-	-	-
Boosting	0.544	-	-	-

CONCLUSIONS

- Preprocessing reduce dataset to 5.000 rows
- All methods results between 0.399 and 0.57
- Learn machine learning methods