Realizarea temei

Scopul temei presupune determinarea genului muzical al sample-urilor apartinand datasetului.

Dataset: format din 2400 de exemple, impartit pentru baseline in 2000 de exemple pentru antrenare si 400 pentru testare.

Clasificatori utilizati (fiecare avand o varianta de baza si una imbunatatita):

1. KMeans
2. Random Forests
3. XGBoost
4. SVM
5. Naive Bayes

Functii auxiliare:

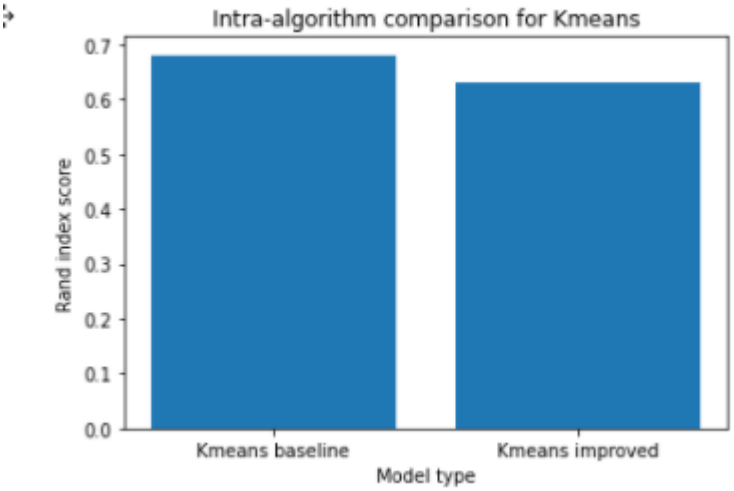
1. get_statistics: intoarce metricile cerute (accuracy, precision, recall, fscore)
2. print_confusion_matrix : afiseaza matricea de confuzie
3. get_cross_validation_metrics: calculeaza si intoarce metricile cerute in cazul cross-validation cu k = 5
4. get_Y: intoarce valoarea prezisa pentru Y, in cazul cross-validation cu k -5

Rezultate obtinute

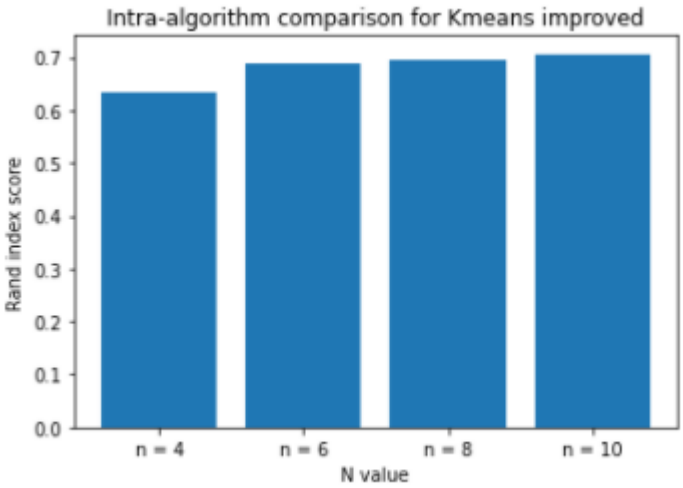
Classificator	Type	Rand_index	Accuracy	Precision	Recall	Fscore
KMeans	baseline	0.6785045852438516	-	-	-	-
KMeans	improved 10	0.6985514797832431	-	-	-	-
KMeans	improved 4	0.6228494511602056	-	-	-	-
KMeans	improved 6	0.6834792274558844	-	-	-	-
KMeans	improved 8	0.7032655967764346	-	-	-	-
Naive Bayes	baseline	-	0.7	0.6978524326481248	0.7	0.6956168769647931
Naive Bayes	improved	-	0.7429166666666667	0.7489827367278078	0.7429166666666667	0.7426818545390586
RandomForest	baseline	-	0.7225	0.7202986492190797	0.7225	0.7196446188898469
RandomForest	improved	-	0.7420833333333334	0.7464005406617531	0.7420833333333334	0.741399347772267
SVM	baseline	-	0.715	0.7148100284517366	0.715	0.712861244578464
SVM	improved	-	0.7454166666666666	0.7515202128045789	0.7454166666666666	0.7448443579756431
XGB	baseline	-	0.73	0.7284619594964422	0.73	0.7267160822449407
XGB	improved	-	0.7470833333333333	0.7516865196354309	0.7470833333333333	0.7462976309821148

KMeans – analiza intra-algoritm

- Variante:
  - Baseline: Am folosit silhouette score pentru a determina numarul ideal de clustere, pe un set de date scalate si am obtinut n = 4, avand un scor rand_index in valoare de 0.6814
  - Improved: Am folosit silhouette score pentru a determina numarul ideal de clustere, pe un set de date scalate, folosind n = 4 , avand un scor rand_index in valoare de 0.6327



- Tehnici de imbunatatire:
  - Am incercat selectarea celor k mai bune features, folosind SelectKBest, cu k =100, dupa ce am introdus in X toate feature-urile existente (237 la numar)
  - Rezultatul nu a fost unul asteptat si, desi am incercat varierea numarului de k features, scorurile au ramas in continuare mai mici pe aceasta varianta decat pe cea baseline. Am incercat rularea algoritmului variind numarul de clustere, manual, obtinand variante mai bune pentru n =6 si n =8, desi numarul de clustere al modelului este de 4. Acest lucru se poate datora numarului mai mare de features si tendintei de overfitting:

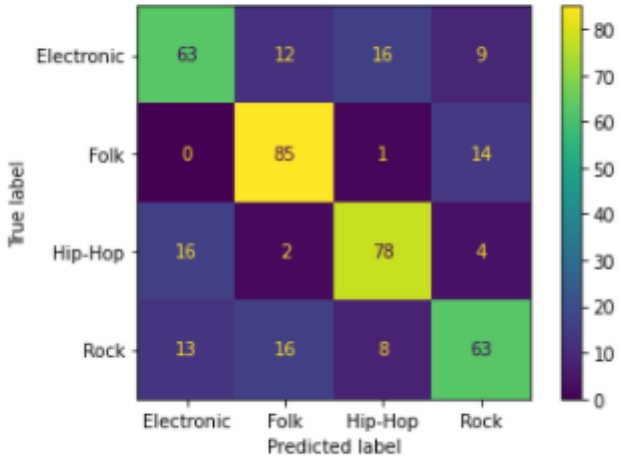


- Concluzii KMeans:
  - Nu au existat imbunatatiri mentinand acelasi numar de clustere intre baseline si improved.
  - S-a constatat o imbunatatire de aprox. 0.07, schimband numarul de clustere.

Random Forests – analiza intra-algoritm

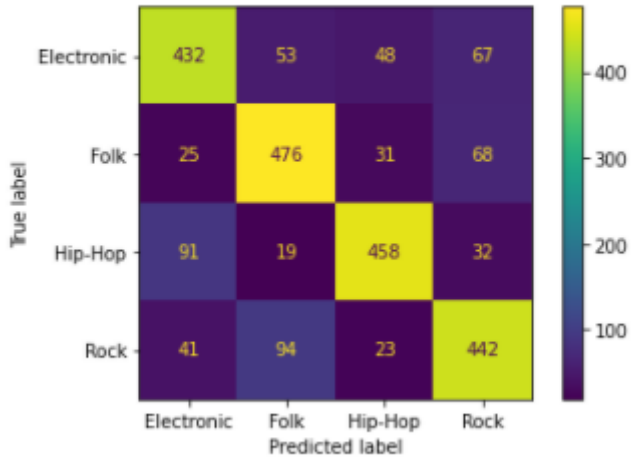
- Variante:
  - Baseline: Am folosit modelul de baza de RandomForestClassifier(), cu toti hyperparametrii setati pe default si setul de date de antrenare nescalat.

```
Accuracy:  0.7225
Recall:  0.7225
Precision:  0.7202986492190797
Fscore:  0.7196446188898469
```

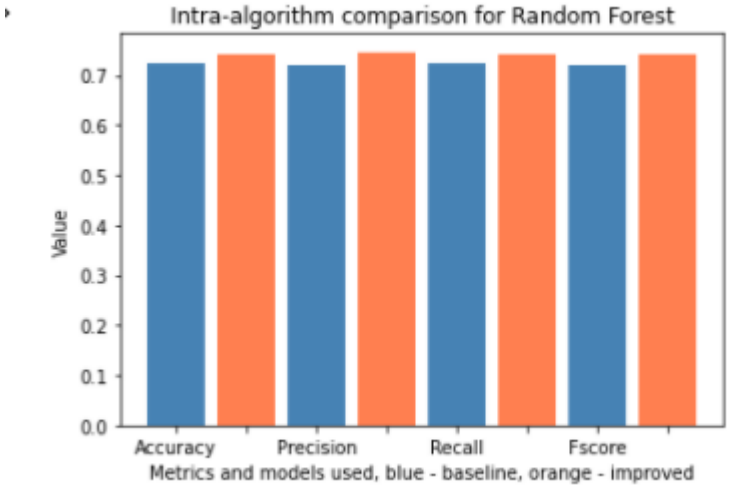


  - Improved: Am folosit modelul de baza de RandomForestClassifier(), facand imbunatatiri atat pe hyperparameters, cat si pe setul de date. In urma cross-validation, am obtinut:

```
Accuracy:  0.7420833333333334
Recall:  0.7420833333333334
Precision:  0.7464005406617531
Fscore:  0.741399347772267
```



- Tehnici de imbunatatire:
  - Am incercat selectarea celor k mai bune features, folosind SelectKBest, cu k =100, dupa ce am introdus in X toate feature-urile existente (237 la numar)
  - Am crescut numarul de n_estimators la 150 de la 100 si max_depth la 50, folosind un set de date de antrenare scalat si cu cele mai bune 100 de features selectate.

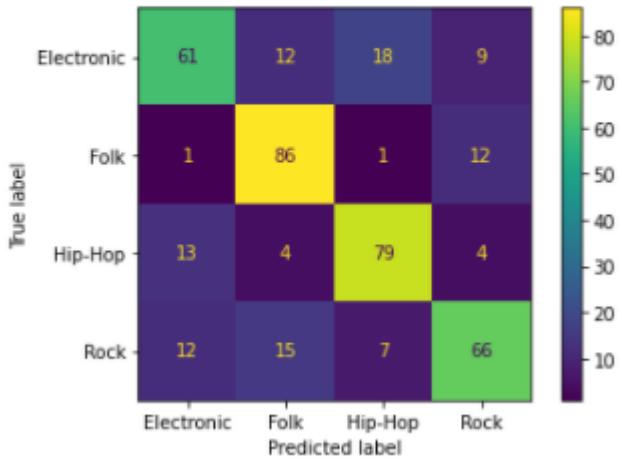


- Concluzii RandomForests:
  - A existat o imbunatatire de 0.02 intre variante (accuracy, recall, precision)
  - Algoritmul clasifica decent toate melodiile in functie de genul acestora, insa se poate observa tendinta de a clasifica unele melodii de Hip-Hop ca fiind Electronic si melodii Rock ca fiind Folk. Acest lucru se poate datora feature-urilor alese pentru antrenare.

XGB – analiza intra-algoritm

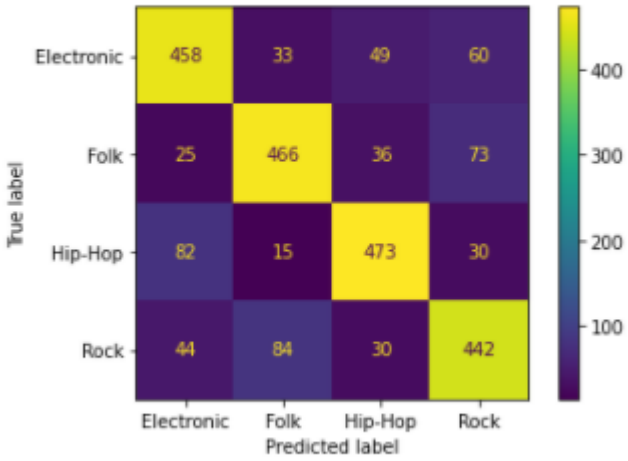
- Variante
  - Baseline: Am folosit modelul de baza de XGBoost(), cu toti hyperparametrii setati pe default si setul de date de antrenare nescalat.

```
Accuracy:  0.73
Recall:  0.73
Precision:  0.7284619594964422
Fscore:  0.7267160822449407
```

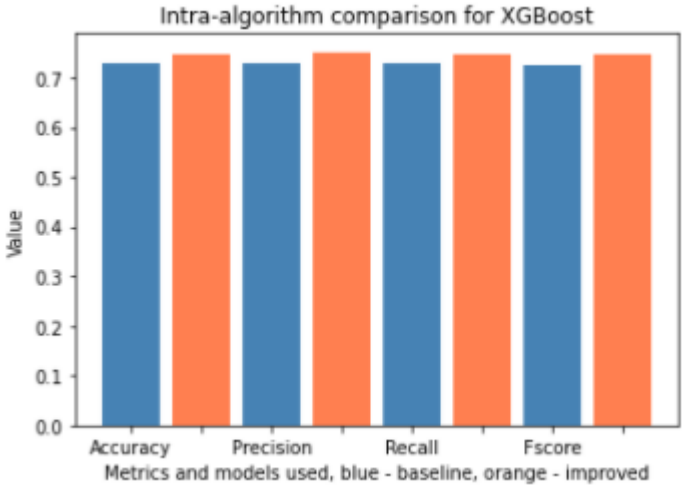


  - Improved: Am folosit modelul de baza de XGBoost(), facand imbunatatiri atat pe hyperparameters, cat si pe setul de date. In urma cross-validation, am obtinut:

```
Accuracy:  0.7470833333333333
Recall:  0.7470833333333333
Precision:  0.7516865196354309
Fscore:  0.7462976309821148
```



- Tehnici de imbunatatire:
  - Am incercat selectarea celor k mai bune features, folosind SelectKBest, cu k =100, dupa ce am introdus in X toate feature-urile existente (237 la numar)
  - Am crescut numarul de n_estimators la 150 de la 100 si max_depth la 10, folosind un set de date de antrenare scalat si cu cele mai bune 100 de features selectate., am am setat objective cu valoarea „multi:softmax” pentru a indica existenta mai multor clase

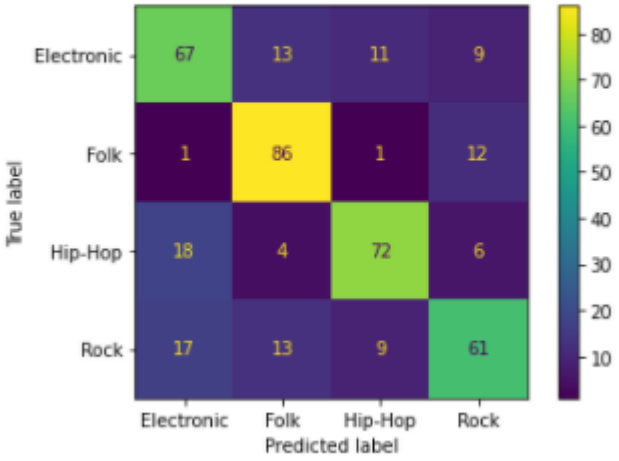


- Concluzii XGB:
  - A existat o imbunatatire de 0.01 intre variante (accuracy, recall, precision)
  - Algoritmul clasifica decent toate melodiile in functie de genul acestora, insa se poate observa tendinta de a clasifica unele melodii de Hip-Hop ca fiind Electronic si melodii Rock ca fiind Folk., dar si cele Folk ca fiind Rock. Acest lucru se poate datora feature-urilor alese pentru antrenare.

SVM – analiza intra-algoritm

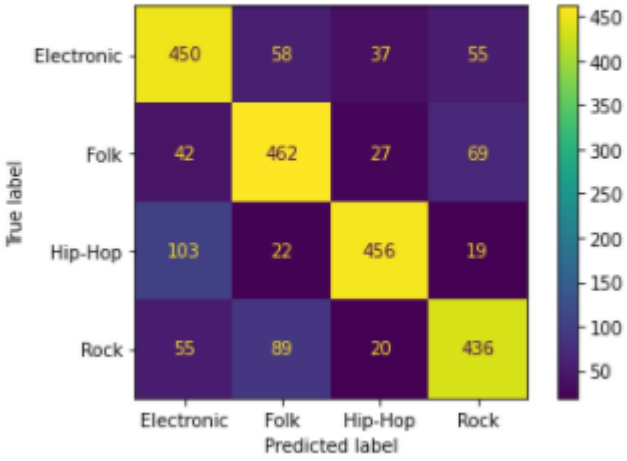
- Variante:
  - Baseline: Am folosit modelul de baza de SVM(), cu toti hyperparametrii setati pe default si setul de date de antrenare nescalat, iar kernelul de tip linear

```
Accuracy:  0.715
Recall:  0.715
Precision:  0.7148100284517366
Fscore:  0.712861244578464
```

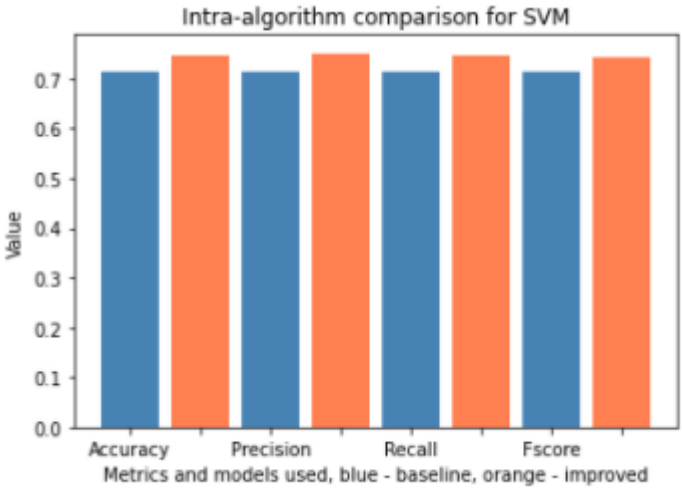


  - Improved: Am folosit modelul de baza de SVM(), facand imbunatatiri atat pe hyperparameters, cat si pe setul de date. In urma cross-validation, am obtinut:

```
Accuracy:  0.7454166666666666
Recall:  0.7454166666666666
Precision:  0.7515202128045789
Fscore:  0.7448443579756431
```



- Tehnici de imbunatatire:
  - Am incercat selectarea celor k mai bune features, folosind SelectKBest, cu k =100, dupa ce am introdus in X toate feature-urile existente (237 la numar)
  - Am folosit modelul de baza de SVM(), cu kernelul de tip ‚rbf' si gamma „scale", in incercarea de a gasi o varianta mai buna a acestui algoritm. Am setat decision_function_shape="ovo", pentru a facilita clasificarea in mai multe clase"

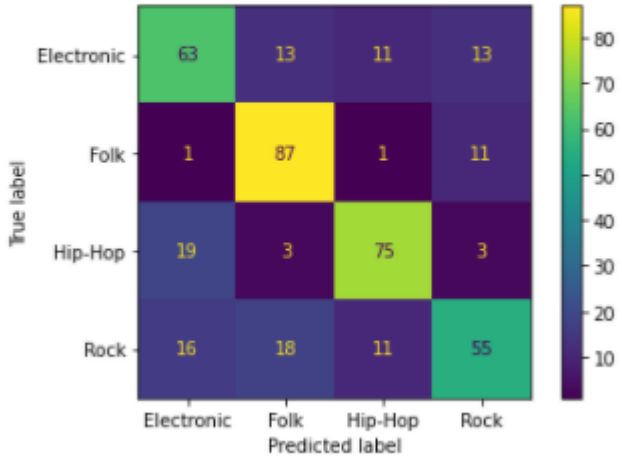


- Concluzii RandomForests:
  - A existat o imbunatatire de 0.03 intre variante (accuracy, recall, precision)
  - Algoritmul clasifica decent toate melodiile in functie de genul acestora, insa se poate observa tendinta de a clasifica unele melodii de Hip-Hop ca fiind Electronic si melodii Rock ca fiind Folk. Acest lucru se poate datora feature-urilor alese pentru antrenare.

Naive Bayes – analiza intra-algoritm

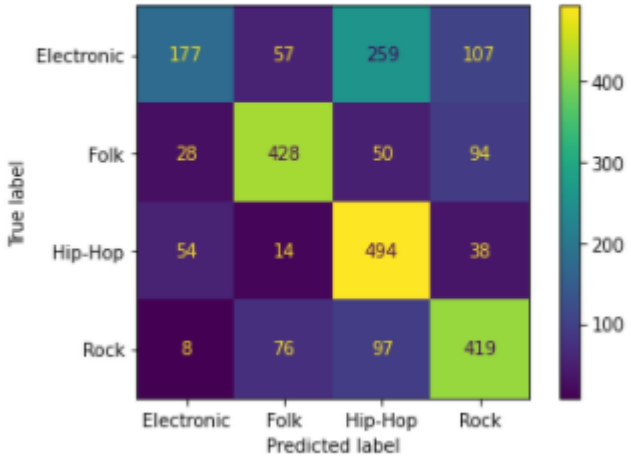
- Variante
  - Baseline: Am folosit modelul de baza de GaussianNB()

```
Accuracy:  0.7
Recall:  0.7
Precision:  0.6978524326481248
Fscore:  0.6956168769647931
```

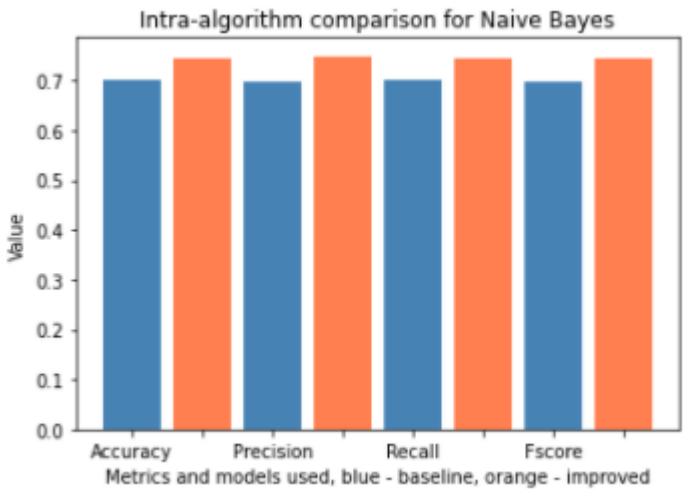


  - Improved: : Am folosit modelul de baza de GaussianNB() facand imbunatatiri pe setul de date. In urma cross-validation, am obtinut:

```
Accuracy:  0.7429166666666667
Recall:  0.7429166666666667
Precision:  0.7489827367278078
Fscore:  0.7426818545390586
```



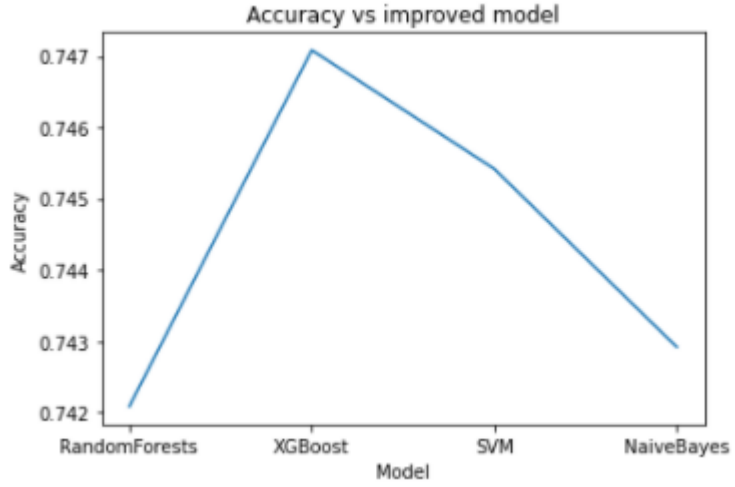
- Tehnici de imbunatatire:
  - Singurele imbunatatiri aplicate au fost cele asupra setului de date: Am incercat selectarea celor k mai bune features, folosind SelectKBest, cu k =100, dupa ce am introdus in X toate feature-urile existente (237 la numar)



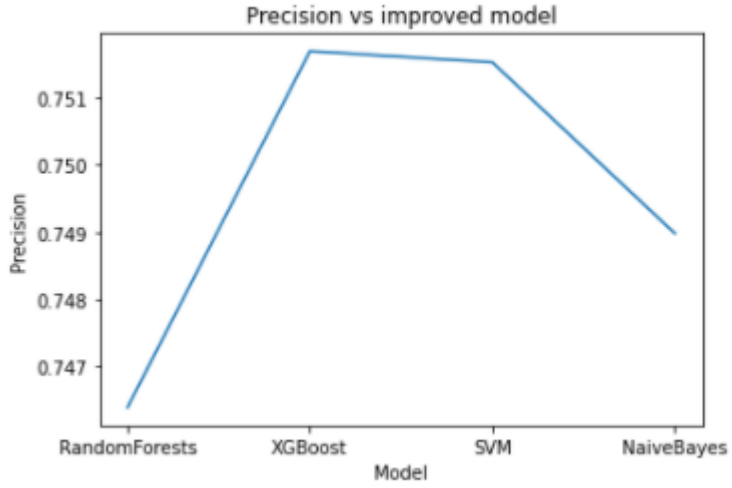
- Concluzii Naive Bayes:
  - A existat o imbunatatire de 0.04 intre variante (accuracy, recall, precision)
  - Algoritmul clasifica decent toate melodiile in functie de genul acestora, insa se poate observa ca in cazul melodiilor de tip Electronic, nu reuseste sa clasifice corect decat un procent foarte mic din acestea, majoritatea fiind clasificate drept Hip-Hop. In schimb, melodiile Hip-Hop sunt corect clasificate drept Hip-Hop intr-un procent foarte ridicat.

Analiza inter-algoritm

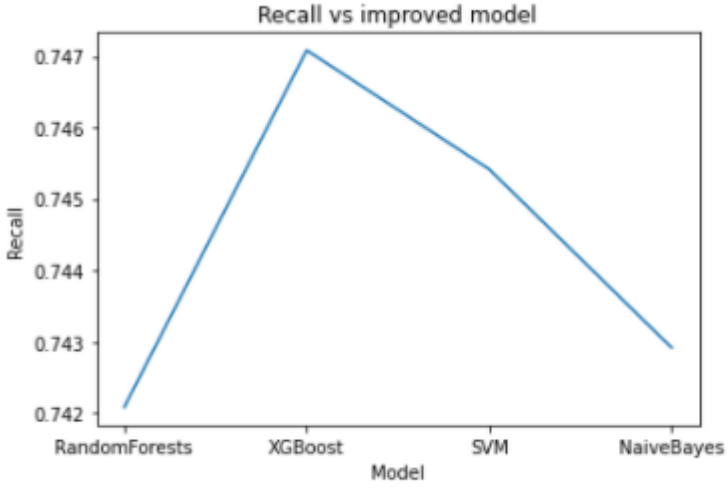
1. Accuracy



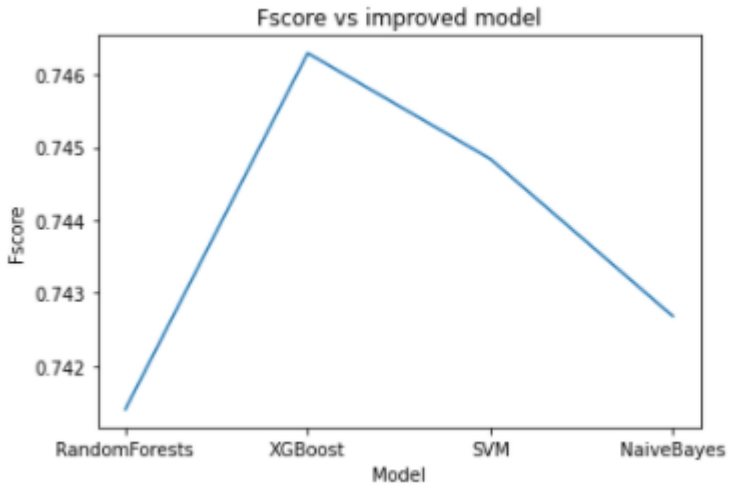
2. Precision



3. Recall



4. Fscore



Concluzii finale

1. Scalarea setului de date de intrare a avut un efect pozitiv asupra performantei algoritmilor, mai putin in cazul KMeans.
2. Extragerea celor mai bune k=100 features a avut un efect pozitiv asupra performantei algoritmilor, mai putin in cazul KMeans. Daca ar fi fost lasate toate features, atunci era posibil ca durata rularii algoritmilor sa fie foarte mare pentru un set de date atat de restrans.
3. Pe acesti parametri particulari, XGBoost a demonstrat cele mai bune performante, insa variantele imbunatatie ale tuturor algoritmilor au avut valori asemanatoare: accuracy de aproximativ 0.74.