

Documentatie Proiect ML

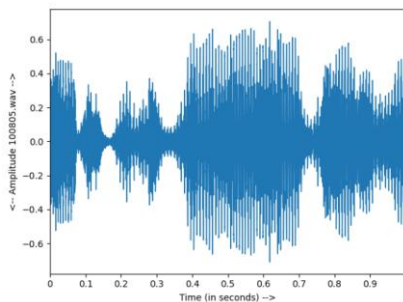
A. Descriere model, parametrii, structurile de date, etc.

Voi descrie procesul de dezvoltare a proiectului in mod cronologic:

1. Primul pas -> extragere de features pentru fisierele audio

Pentru incarcarea fisierelelor am folosit **librosa**, dupa care am plotat 3 fisiere audio din fiecare categorie (train, validation, test).

Exemplu:



← TRAIN

Pentru extragerea de features pentru fiecare fisier audio am folosit functia **librosa.feature.mfcc**, unde am modificat parametrul `n_mfcc` cu 50, care by default este 20. M-am gandit ca daca avem mai multi coeficienti, trainingul va fi mai amanuntit si clasificarea de dupa mai concreta.

Initial am ales sa tin cont de valorile returnate de `zero_crossing_rate`, `mfcc` si `spectral_rollof`, insa ulterior am vazut ca in urma trainingului modelul ajungea sa fie confuz din cauza ca avea atat de multe features de luat in calcul si am hotarat sa folosesc unul singur.

Am ales sa folosesc **.mfcc** pentru ca in urma testelor (am luat individual `mfcc`, `zero_crossing_rate` si `spectral_rollof`) aceste a parut a fi cel mai accurate si citind, am aflat ca este printre cele mai importante metode pentru extragerea feature-ilor si e folosit de majoritatea oamenilor ce lucreaza cu fisiere audio. Acesta returneaza coeficientii realizati in urma scalarii frecventei cu *Mel-scale*. Stiind ca fisierele audio sunt inversate, am transpus matricea primita ca rezultat in urma functiei **librosa.feature.mfcc** dupa care am facut media pe coloane, obtinand astfel un vector, fiecare element corespunzand unui semnal din fisierul audio.

Repet procesul atat pentru datele de validation, cat si pentru cele de test

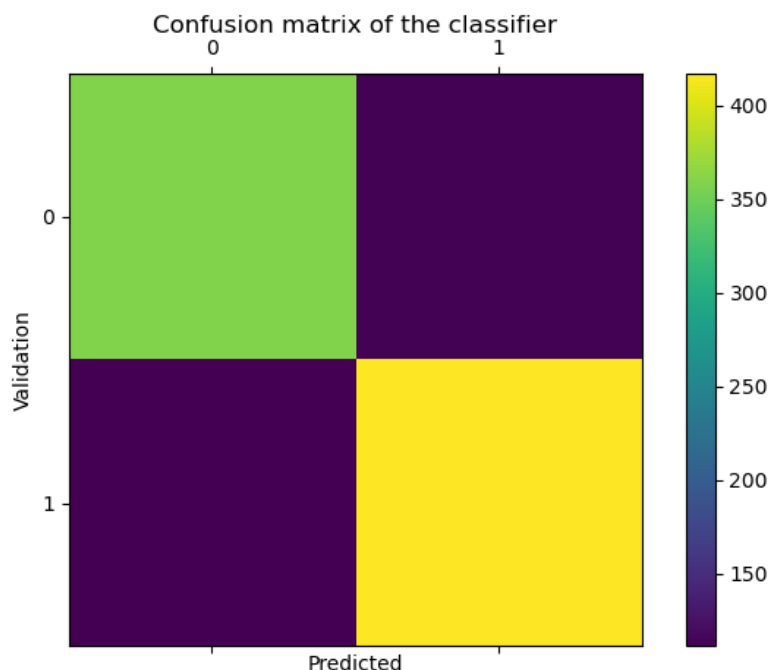
2. Ca model de antrenare si predictie a datelor am folosit masinile cu vector suport (SVM).

Evolutie **scor kaggle** (30% test data) / **acuratete pe validation** in functie de alegerea kernelului si a metricei pentru normalizare si scalare:

- Kernel = linear, metric = L2 \rightarrow 0.58 / 0.58889
- Kernel = linear, metric = standard \rightarrow 0.62 / 0.646
- Kernel = rbf, metrica = standard, C = 10 \rightarrow 0.67 / 0.776

Initial incercasem cu modelul de la laborator (kernel = ,linear', metric = ,l2'), insa dupa mi-am dat seama ca un model cu kernelul ,rbf' ar fi mult mai potrivit tipului de date pentru ca ajuta la evitarea calculului in plus si este foarte flexibil din punct de vedere al stabilirii de granite intre labeluri. Valoarea parametrului C a fost lasata la 10 dupa multe testari, observand ca cu un C > 15 modelul incepea sa faca overfit pe datele pentru predictie.

Confusion_matrix, recall & precision



```
[[359 113]
 [111 417]]
```

CONFUSION MATRIX

← VALUES

Report :				
	precision	recall	f1-score	support
0	0.76	0.76	0.76	472
1	0.79	0.79	0.79	528
accuracy			0.78	1000
macro avg	0.78	0.78	0.78	1000
weighted avg	0.78	0.78	0.78	1000

B. KNN

Ca al doilea model am ales KNN, unde $k = 5$.

In acest caz rezultatul nu a fost unul dorit, acuratetea pe setul de date validation fiind mai mic decat cea obtinuta cu svm.

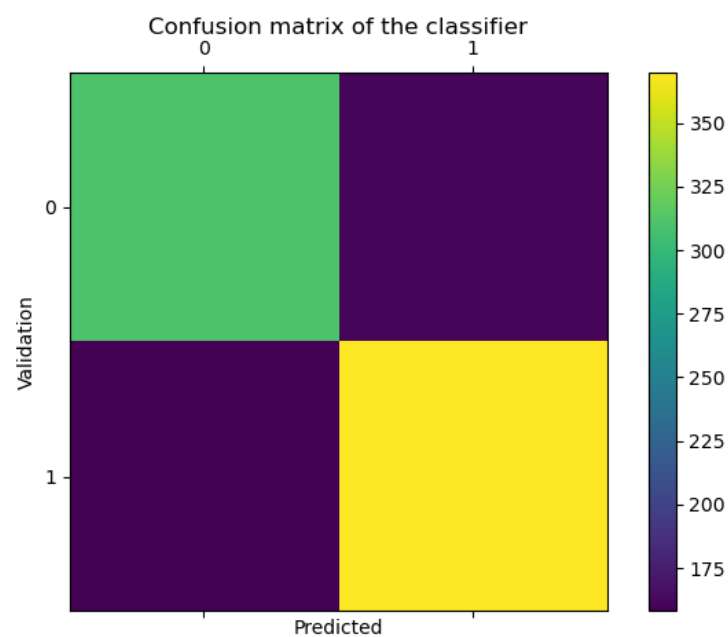
Ca abordare am pastrat una similara cu cea de la svm, extragand features pentru fisierele audio in acelasi fel, doar modul de stabilire a predictiilor fiind diferit.

Pentru fiecare fisier audio in parte, ma folosesc de distanta prestabilita L1, distanta Manhattan, pentru a stabili primii cei 5 mai apropiati vecini si cu ajutorul functiei din numpy ***np.bincount*** stabilim care este cel mai predominant label, pe acesta urmand sa-l atribuim fisierului audio pentru care am apelat functia.

Confusion_matrix, recall & precision

Report :

	precision	recall	f1-score	support
0	0.66	0.66	0.66	472
1	0.70	0.70	0.70	528
accuracy			0.68	1000
macro avg	0.68	0.68	0.68	1000
weighted avg	0.68	0.68	0.68	1000



```
[[311 161]
 [158 370]]
```

CONFUSION MATRIX

← VALUES

Pentru metric = L2, anume distanta euclidiană, acuratetea pe setul de date validation era egală cu 0.63, însă scorul pe kaggle a fost unul mic, aproximativ 0.53.

Am ales să rămân la varianta cu svm pentru că rezultatele au fost mult mai bune, lucru ce se poate vedea și din statisticile din confusion_matrix și report.