Podstawy uczenia maszynowego

PROJEKT 1 – KLASYFIKACJA DANYCH AUDIO

MATEUSZ GAJEWSKI

Baza danych IRMAS

Zawiera zbiór treningowy i 3 zbiory testowe

Zbiór treningowy:

- 6705 plików .wav; 44,1 kHz; 16 bit;
- 3-sekundowe skrawki utworów muzycznych z wskazanym „dominującym” instrumentem;
- 11 instrumentów, w tym struny głosowe;

Zbiory testowe zawierają dane pozyskane w inny sposób

Zbiór treningowy wystarczająco duży na podpodział

Baza danych IRMAS - klasy

Wiolonczela (338)

Klarnet (505)

Flet (451)

Gitara akustyczna (637)

Gitara elektryczna (760)

Organy (682)

Pianino (721)

Saksofon (626)

Trąbka (577)

Skrzypce (580)

Śpiew ludzki (778)

Przygotowanie danych

Brak pustych plików

Wszystkie pliki mają tyle samo próbek

Dwie metody ekstrakcji cech:

1. 30 MFCC + 1. i 2. pochodna
    1. okno 2048 próbek ze skokiem 512 próbek;
    2. spłaszczenie macierzy przez *np.vstack();*
    3. 23312 parametrów na plik;
2. openSMILE eGeMAPSv02
    1. dedykowany do sygnałów mowy;
    2. 88 parametrów związanych z częstotliwością, energią, rozkładem widmowym i zmiennością w czasie;

Przygotowanie danych

Przeznaczenie 20% danych na zbiór testowy

- stratyfikacja klasami;

MFCC ustandaryzowano i zastosowano PCA

- zachowanie ≥ 95% wariancji;
- ostatecznie 1814 parametrów na plik;

openSMILE ustandaryzowano

Algorytmy klasyfikacji

RandomForestClassifier

- szybkie działanie nawet przy dużej liczbie obiektów i cech;
- najważniejsze hiperparametry:
  - *n_estimators*;
  - *max_features*;

SVC

- dobra klasyfikacja danych o skomplikowanym rozkładzie;
- najważniejsze hiperparametry:
  - *C*;
  - *kernel*;

RandomForest na danych MFCC

Hiperparametry:

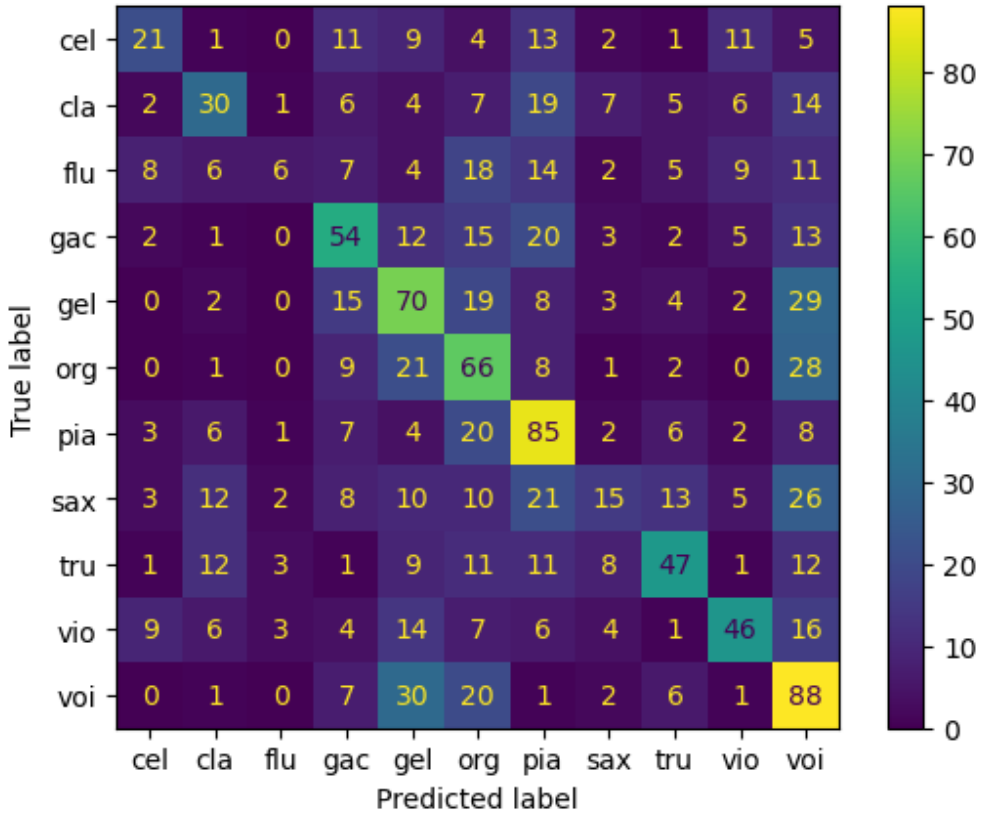
- n_estimators = 100;
- max_features = „sqrt”

Dokładność ok. 40 %

Niektóre instrumenty dobrze rozpoznawane (wiolonczela, organy, pianino)

Flet, saksofon – zgadywanie

Wielu obiektom przypisany głos ludzki



SVC na danych MFCC

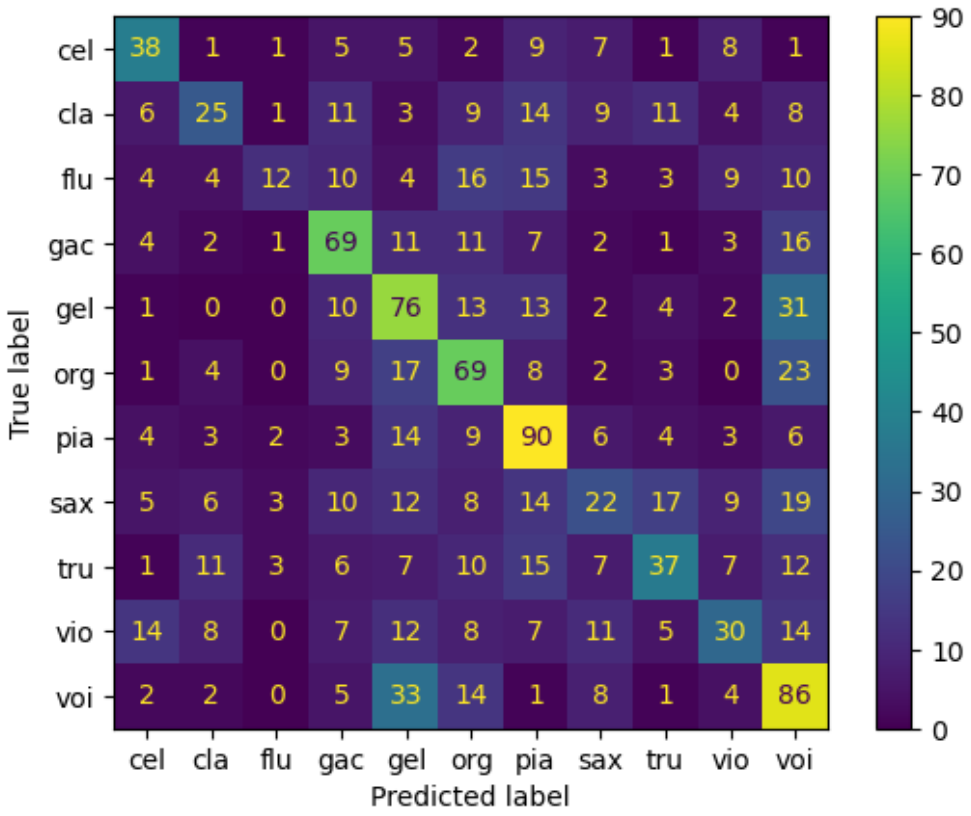
Hiperparametry:

- C = 1.0
- kernel = „rbf”

Dokładność ok. 41%

Problemy z tymi samymi instrumentami

Ten sam problem z głosem ludzkim



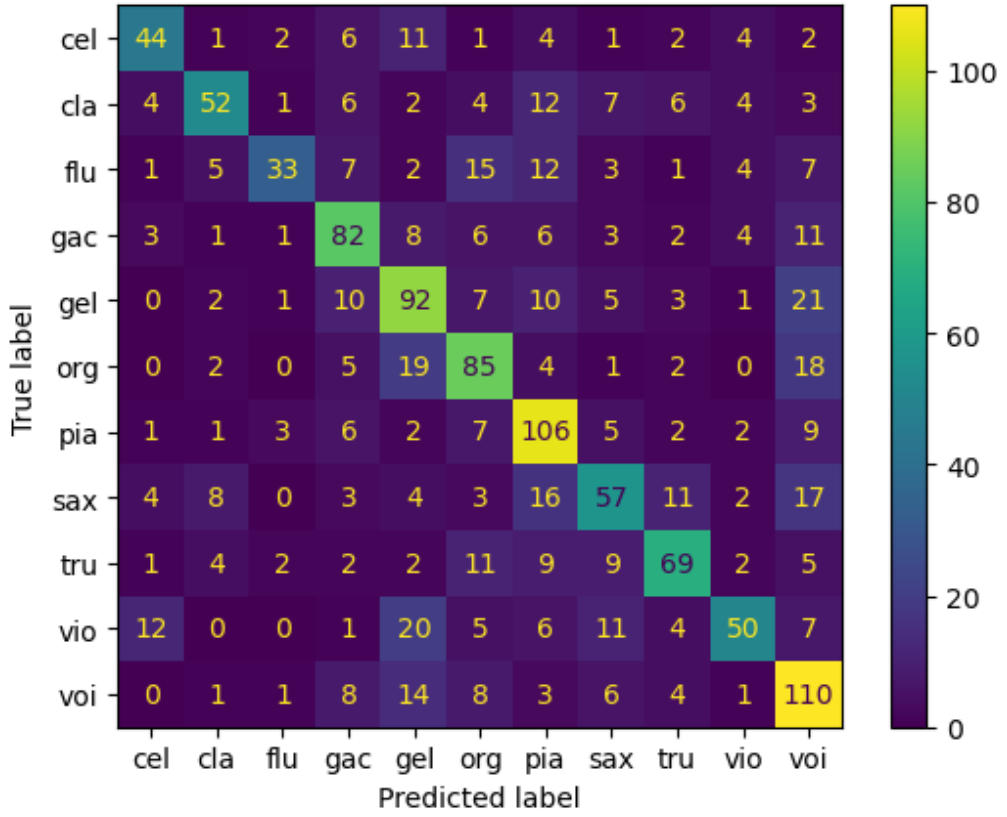
RandomForest na openSMILE

Hiperparametry:

- n_estimators = 100;
- max_features = „sqrt”

Dokładność ok. 58 %

Problematyczne instrumenty lepiej sklasyfikowane względem MFCC



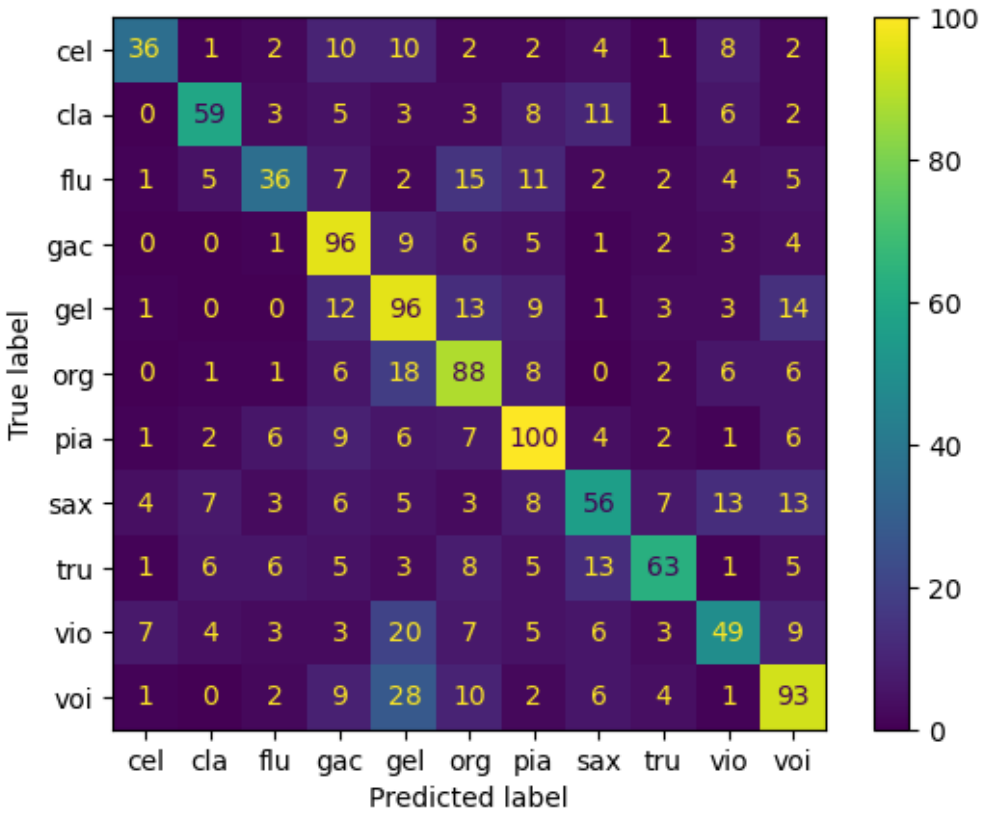
SVC na openSMILE

Hiperparametry:

- C = 1.0
- kernel = „rbf”

Dokładność ok. 58 %

Wyniki praktycznie identyczne jak w RandomForest



Optymalizacja hiperparametrów

Optymalizacja obu modeli ale tylko na danych openSMILE (z uwagi na dużo krótszy czas obliczeń względem MFCC)

Dwie metody optymalizacji: optuna i GridSearchCV

Optymalizacja RandomForest

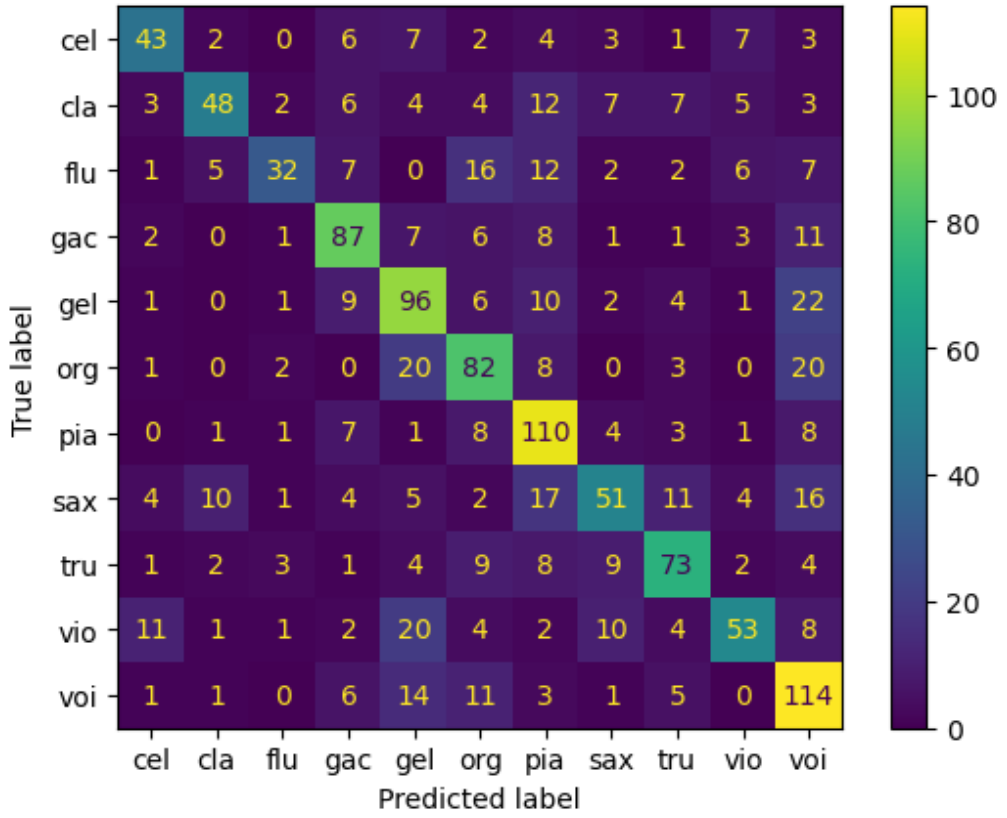
Metoda optymalizacji: optuna

Parametry:

- n_estimators: 50, 300;
- max_features: sqrt, log2, None;
- cel – maksymalizacja dokładności;

Wyniki:

- n_estimators = 220, max_features = log2;
- dokładność ok. 59 % (+1 pkt %);



Optymalizacja SVC

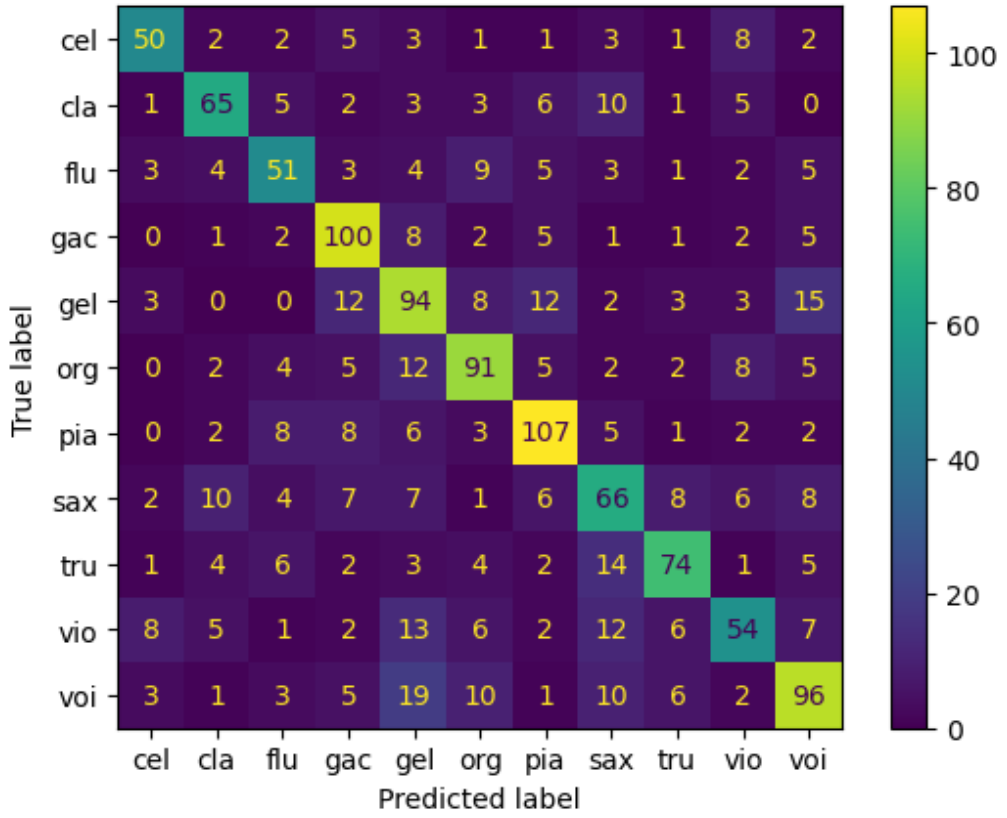
Metoda optymalizacji: GridSearchCV

Parametry:

- 'C': [0.1, 1, 10, 100];
- 'gamma': ['scale', 'auto’];
- 'kernel': ['rbf', 'linear’];
- cel – maksymalizacja dokładności;

Wyniki:

- C = 10.0, gamma = scale,
  kernel = rbf;
- dokładność ok. 63 % (+5 pkt %);
- ryzyko przeuczenia!



Wnioski

openSMILE pozwolił uzyskać lepsze efekty niż MFCC.

Oba algorytmy klasyfikacji dały podobne wyniki, jednak RandomForest wymagał krótszego czasu obliczeń (2-3x).

Zwiększanie wartości parametru C algorytmu SVC prowadzi do lepszych wyników, ale niesie ryzyko przeuczenia modelu.

Ponieważ pliki audio to fragmenty utworów, instrumenty nie są wyizolowane od innych – prawdopodobnie przyczyniło się to do utrudnienia klasyfikacji.

Sprawdzono, jak PCA wpływa na dane openSMILE – redukcja wymiarowości pogarszała wyniki, dlatego dla tych danych jej nie użyto.