Klasyfikacja Gatunków Muzycznych

Karol Denst 305962
Hubert Lawenda 313308
Marcin Latoszek 313307
Dominik Dębiński 304346

March 2023

Spis treści

1 Wstęp

Projekt **MGC** (*Music Genre Clasificator*) posłuży do identyfikacji gatunku muzyki zawartej na podanej ścieżce dźwiękowej. Projekt będzie się składał z dwóch modułów, jeden odpowiedzialny za przetwarzanie danych, drugi za faktyczną klasyfikację.

2 Dane do analizy

Użyjemy następujących danych treningowych dostępnych na platformie kaggle: `https://www.kaggle.com/code/satoru90/music-genre-classification-xgb-deep-learning/notebook`. Dane zawierają 1000 plików w formacie wav poklasyfikowanych na 10 gatunków muzycznych (po 100 plików na każdy gatunek).

3 Wykorzystywane technologie

Projekt zostanie zrealizowany w języku Python z wykorzystaniem bibliotek *pandas*, *numpy*, *tensorflow*, *seaborn* (wizualizacja danych), *librosa* (obróbka audio), za pomocą platformy obliczeniowej JupiterNotebook.

4 Podział zadań

Z powodu braku możliwości zrównoleglenia prac nad oboma modułami (pracę nad klasyfikatorem rozpoczniemy w momencie wytworzenia finalnej wersji modułu przetwarzania danych), zdecydowaliśmy, że każdy członek grupy będzie bezpośrednio zaangażowany w pracę nad aktualnymi zadaniami. Podział obowiązków będzie następował w miarę definiowania kolejnych problemów.

5 Opis rozwiązania

Pierwszym krokiem naszego rozwiązania jest analiza danych wyjściowych. Korzystamy w tym celu z biblioteki `librosa`, za pomocą której wyciągamy szereg parametrów ze ścieżki dźwiękowej.

Stworzyliśmy model klasyfikatora przy użyciu biblioteki XGBoost (model `XGBClassifier`). 80% danych wykorzystaliśmy do trenowania modelu, na pozostałych przeprowadziliśmy klasyfikację wytrenowanym modelem.

5.1 Dane wyciągnięte z plików .wav

```
mfcc = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=20)
```

MFCC to zestaw cech, które reprezentują widmo tonalne sygnału audio. Uchwytują zarówno cechy dźwiękowe, jak i rytmiczne dźwięku. Zwykle oblicza się 20-40 współczynników (parametr n_mfcc, ale najczęściej używane są pierwsze 13.

```
chroma = librosa.feature.chroma_stft(y=y, sr=sr)
```

Cecha chroma reprezentuje 12 różnych klas dźwięków (półtonów) w oktawie muzycznej. Charakteryzuje rozkład dźwięków muzycznych w sygnale audio.

```
spec_contrast = librosa.feature.spectral_contrast(y=y, sr=sr)
```

Ta funkcja oblicza spektralny kontrast, który jest miarą różnic w amplitudzie pomiędzy reprezentacjami częstotliwościowymi dla różnych pasm częstotliwości. Wynikowa cecha składa się z różnicy między maksymalnymi a minimalnymi wartościami amplitud w każdym paśmie częstotliwości. Parametrami wywołania jest sygnał oraz szybkość próbkowania.

```
tonnetz = librosa.feature.tonnetz(y=librosa.effects.harmonic(y))
```

Cecha Tonnetz reprezentuje przestrzenny układ akordów na podstawie informacji o częstotliwościach harmonicznych. Opiera się ona na teorii sieci neuronowych w modelowaniu harmonii muzycznej. Parametrem wywoływanej funkcji jest sygnał audio w postaci tablicy. funkcja `harminics` służy do wyodrębnienia składowej harmonicznej z sygnału audio. Dzięki temu otrzymujemy sygnał zawierający tylko składowe harmoniczne dźwięku.

```
spec_centroid = librosa.feature.spectral_centroid(y=y, sr=sr)
```

Centroid spektralny to ważona średnia częstotliwości (tzw. środek spektralny) obecnych w sygnale audio. Rezultat wskazuje, w jakiej częstotliwości spektrum jest "skoncentrowane". Zapewnia to informacje o "jasności"lub "kolorze"dźwięku.

```
librosa.feature.spectral_bandwidth(y=y, sr=sr)
```

Szerokość pasma spektralnego mierzy zakres częstotliwości wokół centroidu spektralnego. Reprezentuje rozprzestrzenienie lub szerokość zakresu częstotliwości, w którym skoncentrowana jest energia spektralna. Wyższa wartość oznacza szerszy zakres częstotliwości.

```
spec_rolloff = librosa.feature.spectral_rolloff(y=y, sr=sr)
```

Rolloff spektralny to częstotliwość, poniżej której znajduje się określony procent całkowitej energii spektralnej. Reprezentuje punkt odcięcia wysokich częstotliwości. Jest to miara "rozpraszania"energii spektrum wzdłuż osi częstotliwości.

```
zero_crossing_rate = librosa.feature.zero_crossing_rate(y)
```

Funkcja ta oblicza szybkość przekroczenia zera (ZCR - zero crossing rate) dla danego sygnału audio. Zwracana wartość, to tablica zawierająca szybkość przekroczenia zera dla poszczególnych ramek sygnału audio. Szybkość przekroczenia zera jest miarą, mówiącą o tym jak często sygnał zmienia swój znak, czyli

przechodzi przez zero. W kontekście analizy audio, wysoka szybkość przekroczenia zera może sugerować obecność szumów lub głośnej, ostrej perkusji. Niska szybkość przekroczenia zera może sugerować brak dźwięku lub dźwięk o niskiej częstotliwości.

Jest to cecha używana w wielu zastosowaniach przetwarzania dźwięku, w tym w rozpoznawaniu mowy, analizie muzycznej i klasyfikacji gatunków muzycznych.

```
rms = librosa.feature.rms(y=y)
```

Funkcja ta jest używana do obliczenia wartości skutecznej (RMS - Root Mean Square) dla danego sygnału audio. Zwracana wartość, jest tablicą zawierającą wartości RMS dla kolejnych ramek sygnału audio. Wartość RMS dla danego segmentu sygnału dźwiękowego może być interpretowana jako miara jego głośności.

```
harmonic, percussive = librosa.effects.hpss(y)
```

Funkcja ta jest używana w celu rozdzielanie sygnału audio na składniki harmoniczne i perkusyjne. Tak więc harmonic jest to część harmoniczna sygnału audio, która zwykle zawiera melodyjne składniki utworu, a percussive jest to część perkusyjna, która zwykle zawiera rytmiczne składniki utworu.

```
tempo, _ = librosa.beat.beat_track(y=y, sr=sr)
```

Tempo jest to estymowane tempo utworu muzycznego, wyrażone w uderzeniach na minutę. Innymi słowy jest to szybkość rytmu w utworze, co jest podstawą dla taktu i rytmu. Odrzuconym argumentem jest tablica zawierająca indeksy ramek, które odpowiadają wykrytym uderzeniom. Taka konkretna lokalizacja uderzeń nie jest nam potrzebna.

5.2 Opis wykorzystanego algorytmu

W rozwiązaniu korzystaliśmy z pakietu xgboost, a dokładniej z klasy XGBClassifier. Klasa ta implementuje algorytm Extreme Gradient Boosting dla zadania klasyfikacji. Jest to zaawansowany algorytm, który bazuje na koncepcji gradient boosting. Jest to techinka, która buduje model predykcyjny w formie komitetu słabych modeli, zazwyczaj drzew decyzyjnych. Każdy kolejny model jest trenowany w celu minimalizacji błędu pozostałego po poprzednich modelach - stąd nazwa "gradient", ponieważ błąd jest minimalizowany za pomocą metody spadku gradientu.

XGBoost rozwija tę ideę poprzez dodanie regularyzacji. Regularyzacja to technika używana w uczeniu maszynowym, która pomaga w zapobieganiu przeuczeniu modelu (overfitting) poprzez karanie zbyt skomplikowanych modeli. XGBoost również korzysta z bardziej zaawansowanych technik optymalizacji i jest znanym z szybkości działania oraz efektywności.

5.2.1 Opis parametrów wykorzystanego algorytmu

Do dokonania doboru parametrów posłużyła nam funkcja XGBClassifier.GridSearchCV(). Podaje się jej kilka różnych parametrów i ich możliwe wartości. Funkcja ta

sprawdza jakie wyniki dają wszystkie możliwe kombinacje tych parametrów i wypisuje najlepsze. W poniższej tabelce są podane otrzymane wartości parametrów:

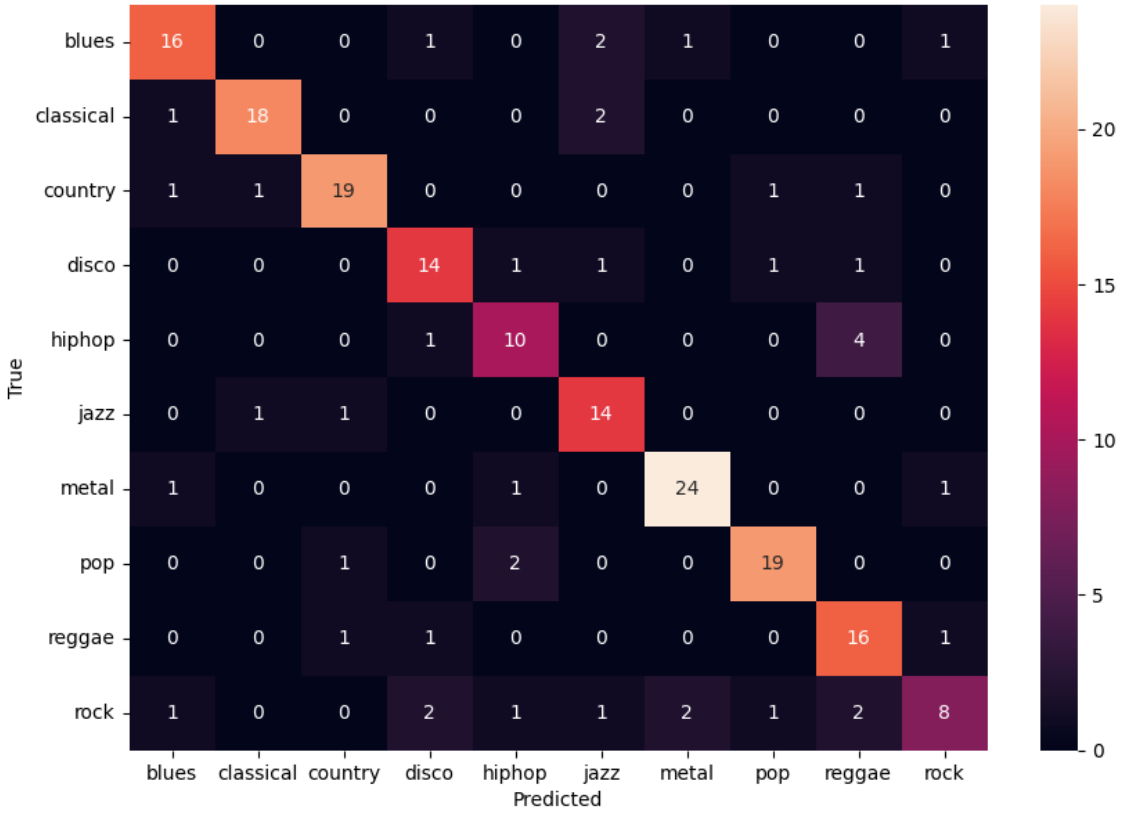
nazwa parametru	wartość
learning_rate	0.3
n_estimators	250
max_depth	4
gamma	0
colsample	1
subsample	1
min_child_weight	3
reg_alpha	0
reg_lambda	1.5
max_delta_step	1

Tabela 1: Parametry

- learning_rate: Jest to szybkość, z jaką model uczy się z błędów z każdej kolejnej iteracji. Wartość 0.3 oznacza, że model "uczy się"z 30 % błędów, które popełnił.
- n_estimators: Liczba drzew w lesie. Czyli ilość drzew decyzyjnych, które zostaną utworzone w ramach algorytmu.
- max_depth: Maksymalna głębokość drzewa. Jest to maksymalna liczba poziomów w drzewie.
- gamma: Minimalna redukcja strat wymagana do dokonania dalszego podziału węzła drzewa. Im większa wartość gammy, tym bardziej konserwatywny jest model.
- colsample_bytree: Proporcja kolumn używanych przy budowaniu każdego drzewa. Wartość 1 oznacza, że używane są wszystkie kolumny.
- subsample: Frakcja próbek treningowych używanych do trenowania każdego drzewa. Wartość 1 oznacza, że używane są wszystkie próbki treningowe.
- min_child_weight: Minimalna suma wag potrzebna do podziału na kolejne poddrzewo. Im większa wartość, tym bardziej konserwatywny model.
- reg_alpha: Współczynnik regularyzacji L1. Im większa wartość, tym bardziej skomplikowany model.
- reg_lambda: Współczynnik regularyzacji L2. Im większa wartość, tym bardziej skomplikowany model.
- max_delta_step: Pozwala na optymalizację kosztu dla nierównomiernych klas. Im większa wartość, tym bardziej konserwatywny model.

6 Wyniki

Na poniższym obrzasku widać jak zgadywał nasz model. Widać że najlepiej poradził sobie ze zgadywaniem muzyki klasycznej co nie dziwi.



Poniższa tabelka zawiera wydajność naszego modelu.Zawiera następujące kolumny: precyzja (precision), pełność (recall), miara F1 (f1-score) i wsparcie (support).

- Precyzja: proporcja prawdziwie pozytywnych prognoz (poprawnych prognoz gatunku) spośród wszystkich pozytywnych prognoz.
- Pełność: proporcja prawdziwie pozytywnych, które zostały poprawnie zidentyfikowane.
- Miara F1: miara dokładności, która uwzględnia zarówno precyzję, jak i pełność, obliczana jako 2 * (precyzja * pełność) / (precyzja + pełność).
- Wsparcie: liczba instancji w zestawie danych dla każdego gatunku.

Dodatkowo na dole tabelki są dodatkowe wiersze: Wiersz "dokładność"pokazuje ogólną dokładność klasyfikatora we wszystkich klasach. "Średnia makro"podaje średnią precyzję, pełność i miarę F1, nie uwzględniając proporcji każdej klasy w zestawie danych. "Średnia ważona"dostarcza to samo, ale uwzględnia proporcję każdej klasy.

Genre	**Precision**	**Recall**	**F1-score**	**Support**
blues	0.80	0.76	0.78	21
classical	0.90	0.86	0.88	21
country	0.86	0.83	0.84	23
disco	0.74	0.78	0.76	18
hiphop	0.67	0.67	0.67	15
jazz	0.70	0.88	0.78	16
metal	0.89	0.89	0.89	27
pop	0.86	0.86	0.86	22
reggae	0.67	0.84	0.74	19
rock	0.73	0.44	0.55	18
accuracy			0.79	200
macro avg	0.78	0.78	0.78	200
weighted avg	0.79	0.79	0.79	200

Tabela 2: Wyniki

7 Wnioski

Podczas realizacji projektu stworzony został model uczenia maszynowego dla klasyfikacji gatunków muzycznych. Metoda Accuracy Cross Gradient Booster okazała się być wysoce efektywna w dokładnym przyporządkowywaniu kategorii muzycznych do zadanych na wejściu utworów. Zrealizowane osiągnięcia w tym projekcie wskazują na duży potencjał technik uczenia maszynowego w automatyzacji klasyfikacji gatunków muzycznych, co może mieć znaczący wpływ w różnorodnych aplikacjach takich jak systemy rekomendacji piosenek oraz muzyczne platformy streamingowe.

8 Wykorzystane Biblioteki

- os
- librosa
- pandas
- numpy
- matplotlib

- seaborn
- sklearn
- xgboost

Literatura

[1] https://xgboost.readthedocs.io/en/stable/

[2] https://www.nvidia.com/en-us/glossary/data-science/xgboost/

[3] https://en.wikipedia.org/wiki/Spectral_centroid

[4] https://en.wikipedia.org/wiki/Zero-crossing_rate