ROZPOZNAWANIE GATUNKÓW MUZYCZNYCH Z WYKORZYSTANIEM UCZENIA MASZYNOWEGO

Bartosz Gracjan Dorobek
Instytut Telekomunikacji, Politechnika Warszawska
bartosz.dorobek.stud@pw.edu.pl

Streszczenie

Praca ma na celu zbadanie trzech modeli nadzorowanego uczenia maszynowego: drzewa decyzyjnego, maszyny wektorów wspierających oraz lasu losowego, w klasyfikacji gatunków muzycznych. Do nauki modeli zostały wykorzystane dane tabelaryczne zawierające informacje o cechach audio tysiąca próbek utworów muzycznych z dziesięciu różnych gatunków muzycznych. Przedstawione zostały wykorzystane metody przetwarzania danych i ewaluacji modeli, które są niezbędne do prawidłowego działania estymatorów oraz pozwalają zwiększyć ich skuteczność. Omówione zostały także podstawowe mechanizmy działania testowanych modeli, a ich rezultaty działania zostały ze sobą porównane. Tematyka pracy wpisuje się w rozwijającą się dziedzinę nauki jaką jest Music Information Retrival (MIR), która znajduje szerokie zastosowanie w serwisach streamingowych.

1. Wstęp

Muzyka towarzyszy nam na co dzień, a rozwój serwisów streamingowych sprawił, że dostęp do niej jest szerszy niż kiedykolwiek wcześniej. Rozwój aplikacji streamingowych nie byłby jednak możliwy bez zastosowania różnych technik analizy danych. Znajdują one zastosowanie w funkcjach takich jak tworzenie spersonalizowanych rekomendacji muzycznych dla użytkownika, czy układanie playlist na podstawie ulubionych gatunków. Analiza utworów muzycznych jedynie na podstawie metadanych wprowadzonych przez wydawcę może być jednak ograniczająca, a same metadane mogą być niekompletne. Stąd potrzeba analizy utworów muzycznych na podstawie cech audio, w której szerokie zastosowanie znajduje sztuczna inteligencja.

W niniejszym artykule opisuję projekt, którego celem było zbadanie wybranych modeli nadzorowanego uczenia maszynowego w klasycznym problemie rozpoznawania gatunków muzycznych.

2. Dane

Dane wykorzystane w projekcie pochodzą z publicznego zbioru GTZAN, który zawiera 30 sekundowe, 16-bitowe, monofoniczne pliki audio w formacie WAV. Zbiór danych obejmuje 10 gatunków muzycznych, każdy reprezentowany przez 100 utworów, co daje sumarycznie 1000 ścieżek audio [1].

Gatunki muzyczne bazy GTZAN: Blues, Classical, Country, Disco, Hiphop, Jazz, Metal, Pop, Reggae, Rock.

Zbiór danych zawiera dane tabelaryczne z wyekstrachowanymi wartościami średniej i wariancji następujących miar cech audio:

- Chroma STFT – reprezentuje intensywność, czyli rozkład energii, na 12 klasach wysokości dźwięku, będących odwzorowaniem skali chromatycznej;
- RMS – jest to średnia kwadratowa sygnału, będąca miarą głośności ścieżki audio;
- Spectral centroid – miara centroidu widmowego wskazuje „środek ciężkości" widma sygnału i odzwierciedla poziom „jasności" brzmienia [2];
- Spectral bandwidth – jest miarą odchylenia widma od centroidu widmowego;
- Rolloff – reprezentuje częstotliwość, poniżej której znajduje się określony procent energii widma, np. 85% [3];
- Zero crossing – jest to miara szybkości przechodzenia sygnału przez zero, czyli zmiany znaku [3];
- Harmony – reprezentuje energię części harmonicznej sygnału uzyskanej przez zastosowanie filtru poziomego;
- Percussive – reprezentuje energię części perkusyjnej sygnału uzyskanej przez zastosowanie filtru pionowego;

- MFCC – jest to zestaw cech (w naszym przypadku 20), które charakteryzują ogólny kształt obwiedni widmowej;
- Tempo – liczba uderzeń na minutę (BPM).

2.1. Przetwarzanie danych

Zanim poddamy nasze dane treningowi kluczowe jest poddanie ich odpowiedniemu przygotowaniu. W tabeli występują dwie kolumny, które nie reprezentują cech utworów - są to “filename” oraz “length”. Nazwa pliku nie jest tytułem utworu, a jedynie nazwą katalogową (np. “blues.0011.wav”), natomiast długości wszystkich plików audio są jednakowe i wynoszą 30 sekund. Z tych względów obie te kolumny usunąłem. Wówczas wszystkie dane w tabeli mają odpowiedni format liczbowy, reprezentują cechy nagranych utworów i nie występują w nich puste pola.

Aby działanie niektórych algorytmów uczenia maszynowego było możliwe (np. SVC, który omawiam w rozdziale 4.) konieczne jest skalowanie danych, którego dokonałem przy użyciu funkcji *StandardScaler*. Skaluje ona oddzielnie każdą próbkę wejściową X, odejmując od niej średnią wartość wszystkich próbek zbioru uczącego U i dzieląc przez ich odchylenie standardowe S. W ten sposób otrzymujemy standaryzowaną wartość Z - zgodnie z poniższym wzorem [4].

$$Z = \frac{X - U}{S}$$

Rezultatem standaryzacji próbek jest przesunięcie ich rozkładu, tak aby średnia była równa 0, a odchylenie standardowe równe 1. W efekcie umożliwiło mi to w dalszej części zastosowanie wszystkich modeli.

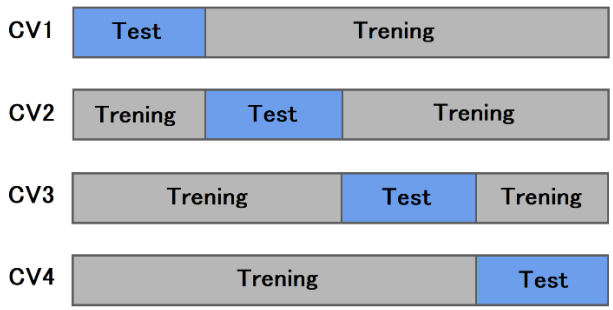
3. Ewaluacja Modeli

3.1 Metryki

Algorytmy uczenia maszynowego posiadają funkcje kosztu, bądź metryki, które należy optymalizować celem rozwiązania problemu. Abyśmy byli w stanie zweryfikować jakość modelu niezależnie od jego rodzaju, musimy zbudować metrykę, która będzie interpretowalna, jak i rozróżniać modele dobre od tych o niskiej jakości. W przypadku klasyfikacji taką metryką może być dokładność (ang. accuracy), która jest proporcją poprawnie zaklasyfikowanych elementów do wszystkich elementów zbioru testowego. Istnieje również wiele innych metryk, które pozwalają oceniać model na przykład pod kątem błędów I i II rodzaju, a każdą z nich możemy stworzyć w oparciu o macierz pomyłek C (ang. confusion matrix). Z definicji C_{ij} jest równa liczbie obserwacji, o których wiadomo, że należą do grupy i, a przewiduje się, że należą do grupy j.

3.2 Data Splitting

Ewaluując jakość modelu nie można używać danych treningowych, ponieważ istnieje ryzyko, że model jest przeuczony i nadmiernie dopasowany (do tego konkretnego podzbioru). Aby tego uniknąć stosuje się data splitting, czyli podział danych na zbiór treningowy i testowy, gdzie testowy przeznaczony jest jedynie do ewaluacji jakości modelu. Podziału zbioru dokonuje się w sposób losowy i również może on mieć wpływ na wyniki, ponieważ w zbiorze testowym może znaleźć się więcej, bądź mniej "łatwych" lub "trudnych" przypadków. Aby zniwelować ten efekt możemy wykorzystać walidację krzyżową (and. cross validation), która uśrednia wyniki z kilku (np. 4) ewaluacji [Rys.1.]. Podczas każdej z ewaluacji krzyżowych dokonywany jest inny podział zbioru na treningowy i testowy, aby zminimalizować błąd wynikający z pojedynczego arbitralnego podziału.



Rys. 1. Schemat walidacji krzyżowej dla czterech ewaluacji

4. Modele

Przy implementacji algorytmów pomocne są open source'owe narzędzia i biblioteki, taka jak Scikit-learn do uczenia maszynowego dla języka programowania Python, której użyłem. Do rozwiązania problemu klasyfikacji wybrałem trzy modele: drzewa decyzyjne, maszynę wektorów wspierających i las losowy (ang. Decision Tree, Support Vector Machine i Random Forest).

4.1. Decision Tree Calassfier

Działanie tego modelu polega na iteracyjnym dzieleniu zbioru danych na coraz mniejsze podzbiory zgodnie z wynikiem instrukcji warunkowej. Dzieje się to w taki sposób, aby w każdym kolejnym węźle próbki były lepiej rozdzielone pod względem 10 klas, które w tym przypadku reprezentują gatunki muzyczne. Liśćmi drzewa decyzyjnego nazywamy ostatnie węzły drzewa, w których oczekujemy jak najlepszego rozdzielenia próbek danych. Testom zostały poddane trzy hiperparametry drzewa decyzyjnego: kryterium podziału, maksymalna głębokość drzewa i minimalna liczba próbek w liściu.

4.2. SVC

Support Vectors Classifier to model, który klasyfikuje dane reprezentowane jako punkty w przestrzeni poprzez rozdzielenie ich na klasy za pomocą hiperpłaszczyzn. Model tworzy hiperpłaszczyzny o maksymalnych marginesach, czyli o największej odległości od najbardziej wysuniętych punktów różnych klas oraz o minimalnym błędzie klasyfikacji.

Często jednak punkty danych w przestrzeni nie są możliwe do odseparowania liniową hiperpłaszczyzną. W tym celu wykorzystywana jest tzw. sztuczka jądro, która mapuje dane wejściowe do przestrzeni o większej liczbie wymiarów, w której jest to już możliwe. Testowane były 4 jądra: liniowe (linear), radialnej funkcji bazowej (RBL), wielomianowe (poly) i funkcji sigmoidalnej (sigmoid). Oprócz tego sprawdzane były dwa hiperparametry: C (definiuje jak bardzo model powinien unikać błędu klasyfikacji kosztem szerokości marginesu) i gamma (definiuje jak bardzo chcemy dopasować nieliniowe hiperpłaszczyzny do punktów danych).

4.3. Random Forest Classfier

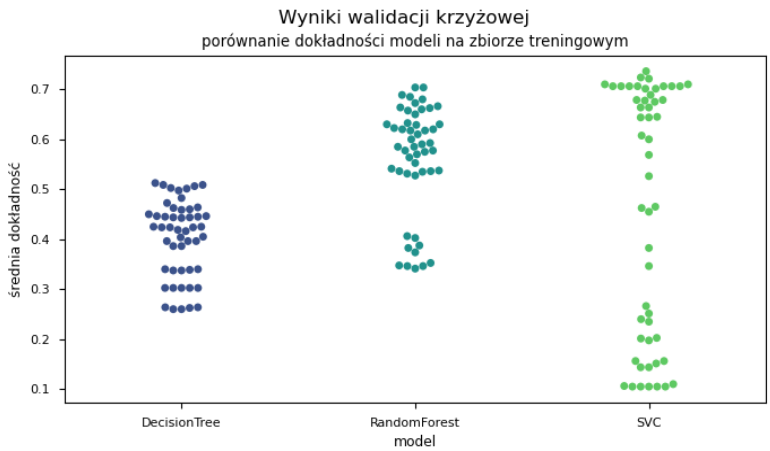
Las losowy, jak sama nazwa wskazuje, jest modelem składającym się z wielu pojedynczych drzew decyzyjnych, które działają zespołowo. Każde pojedyncze drzewo decyzyjne w losowym lesie zwraca predykcję klasy, a klasa z największą liczbą głosów staje się prognozą naszego modelu. Ważne dla skuteczności tego modelu jest, aby drzewa decyzyjne, które wchodzą w jego skład, nie były ze sobą skorelowane [5].

5. Wybów Hiperparametrów

W celu znalezienia zestawu hiperparametrów, które dają najlepszy wynik, wykorzystałem metodę przeszukiwania Grid Search, polegającą na testowaniu wszystkich możliwych kombinacji hiperparametrów.

6. Wyniki

Na poniższym wykresie punktowym [Rys.2.] przedstawione są średnie wyniki walidacji krzyżowej, dla wszystkich badanych modeli o zadanych hiperparametrach, co stanowi graficzne odzwierciedlenie ich skuteczności. Najlepsze rezultaty otrzymał model SVC z maksymalnym dopasowaniem równym 73,625%.



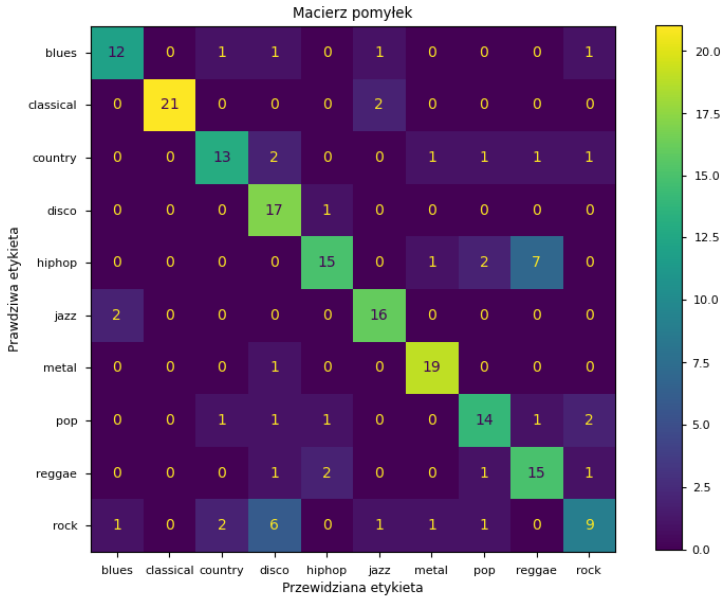
Rys. 2. Wyniki średniej dokładności trenowanych modeli

Sumarycznie zostało przetestowanych 152 modeli z różnym zestawem hiperparametrów. Model SVC uzyskał pierwsze 14 najlepszych wyników w całym teście. Ponadto średni czas treningu dla modeli SVC był bardzo niski. Zarówno pod względem skuteczności i wydajności okazał się być najlepszym model dla problemu klasyfikacji gatunków muzycznych. Najgorsze wyniki uzyskał model drzewa decyzyjnego i bez względu na ustawione hiperparametry nie był on w stanie uzyskać dokładności lepszej niż ok. 51%. Lepiej od niego poradził sobie las losowy, dzięki zastosowaniu mechanizmu głosowania.
Wyniki najlepszych modeli dla trzech badanych klasyfikatorów zostały przedstawione w tabeli [Tab.1].

p.	**classfier**	**mean_test_score**	**mean_fit_time [s]**
1.	SVC	0.73625	0.027
3.	RandomForest	0.69875	0.619
3.	DecisionTree	0.51125	0.017

Tab. 1. Najlepsze wyniki modeli CSV, RandomForest i DecisionTree

Najlepszy wybrany estymator SVC posiadał następujące hiperparametry: jądro RBF, C = 10, gamma = 0.01. Przeprowadziłem z jego wykorzystaniem predykcję gatunków na zbiorze danych testowych, a jej wyniki są zwizualizowane na macierzy pomyłek [Rys. 3.]. Model miał tendencję do fałszywego oceniania utworów hiphopowych jako reggae oraz rockowych jako disco. Poza tym, uznaję wyniki za bardzo satysfakcjonujące.



Rys. 3. Macierz pomyłek dla predykcji najlepszego modelu

7. Podsumowanie

Kluczowe dla rozwiązania problemu było dokładne zbadanie danych i modeli. Wówczas możliwe okazało się znalezienie modeli, których wyniki są zadawalające. Projekt ten pozwolił mi zrozumieć najważniejsze zagadnienia związane z uczeniem maszynowym i potwierdzić jego potencjał w zastosowaniach związanych z przetwarzaniem informacji muzycznych. Dalsze możliwości rozwoju stanowi zastosowanie splotowych sieci neuronowych (CNN) na obrazach spektrogramów, które są popularną metodą w dziedzinie analizy audio.

Literatura

1. Tensor Flow catalog, GTZAN audio dataset, https://www.tensorflow.org/datasets/catalog/gtzan
2. Grey, John M.; Gordon, John W. *"Perceptual effects of spectral modifications on musical timbres"*, The Journal of the Acoustical Society of America, 1995.
3. Joel Jogy, *"How I Understood: What features to consider while training audio files?"*, Towards Data Science, Sep 6, 2019.
4. scikit-learn developers, *Sklearn preprocessing – StandardScaler*, 2007-2022 https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
5. Tony Yiu, *„Understanding Random Forest. How the Algorithm Works and Why it Is So Effective"*, Towards Data Science, Jun 12, 2019

Struktury informacji

1. Poziom całego dokumentu

Tematem artykułu jest zbadanie wybranych modeli uczenia maszynowego w klasyfikacji gatunków muzycznych. Na poziomie całego dokumentu zastosowany jest układ chronologiczny. Kolejne rozdziały odpowiadają kolejnym krokom stosowanym w rozwiązywaniu problemów z wykorzystaniem uczenia maszynowego.

2. Poziom poszczególnych rozdziałów

2.1. Rozdział 1 – Wstęp

Ten rozdział przedstawia motywację dla zrealizowania projektu oraz definiuje jego cel.

2.2. Rozdział 2 – Dane

Ten rozdział omawia dane wykorzystywane podczas treningu oraz testowania modeli.

2.3. Rozdział 3 – Ewaluacja modeli

Ten rozdział przedstawia metody wykorzystane w celu poprawnej oceny skuteczności modeli.

2.4. Rozdział 4 – Modele

Ten rozdział omawia podstawowe zasady działania stosowanych w pracy algorytmów.

2.5. Rozdział 5 – Wybór hiperparametrów

Ten rozdział definiuje sposób wyboru hiperparametrów.

2.6. Rozdział 6 – Wyniki

Ten rozdział przedstawia, analizuje i wizualizuje uzyskane wyniki.

2.7. Rozdział 7 – Podsumowanie

Ten rozdział ma na celu podsumowanie omawianego w artykule badania.

MUSIC GENRE RECOGNITION USING MACHINE LEARNING

Bartosz Gracjan Dorobek
Instytut Telekomunikacji, Politechnika Warszawska
bartosz.dorobek.stud@pw.edu.pl

Abstract

The aim of this thesis is to investigate three models of supervised machine learning in the classification of music genre: decision tree, support vector machines, and random forest. Those models has learned from the input data containing information on the audio characteristics of a thousand samples of musical works from ten different musical genres. There were presented methods of data processing and model evaluation, which are necessary to ensure the proper functioning of the estimators and allow to increase their effectiveness. The results of the conducted experiments are described and discussed. The subject of the article is part of the developing field of science Music Information Retrival (MIR), which is widely used in streaming services.