

Celem tego projektu było wykorzystanie algorytmu K najbliższych sąsiadów (KNN) do określenia gatunku piosenek przy użyciu kompleksowego zbioru danych przechowywanych w tabeli CSV. Zbiór danych zawierał wiele cech dźwiękowych dla każdego wpisu, w tym metadane takie jak nazwa pliku i długość, a także różne cechy statystyczne i spektralne, takie jak chroma, RMS, spectral centroid, spectral bandwidth, roll-off, zero-crossing rate, harmony, perceptual features, tempo, and MFCC coefficients.

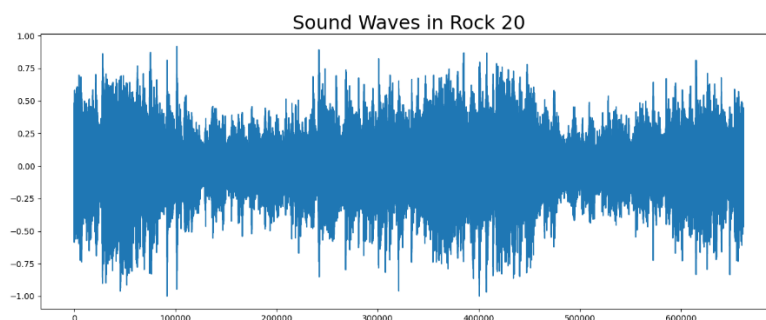
W celu poprawy dokładności i precyzji algorytmu KNN, podczas procesu opracowywania zastosowano kilka technik. Wykorzystano analizę składowych głównych (PCA), aby zmniejszyć wymiarowość przestrzeni cech i zidentyfikować najważniejsze cechy dla klasyfikacji gatunku. Dzięki zastosowaniu PCA algorytm mógł zidentyfikować kombinację cech, które miały największe znaczenie dla zadania identyfikacji gatunku, umożliwiając bardziej precyzyjną i efektywną klasyfikację.

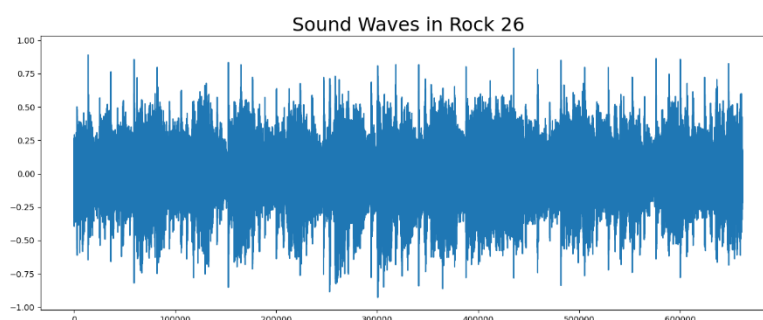
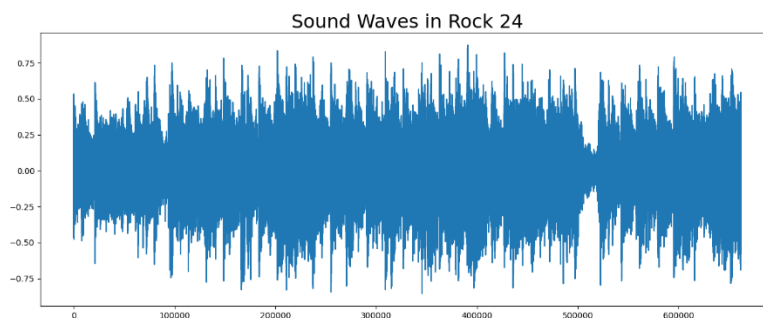
Podczas procesu opracowywania oceniano różne kombinacje cech, aby ocenić ich indywidualne znaczenie w celu dokładnego określenia gatunku. Dzięki systematycznym eksperymentom z różnymi podzbiorami cech zespół projektowy mógł zidentyfikować najbardziej informacyjne cechy i wybrać optymalny zestaw cech w celu uzyskania poprawionej dokładności.

Podjęte iteracyjne podejście podczas opracowywania spowodowało znaczące zwiększenie dokładności algorytmu. Początkowa średnia dokładność wynosząca 60% została znacznie poprawiona, osiągając średnią dokładność na poziomie 70% w modelu końcowym. Poprawa ta może być przypisana starannemu procesowi wyboru cech, w tym wykorzystaniu PCA do identyfikacji wpływowych składowych i eksperymentowaniu z różnymi kombinacjami cech.

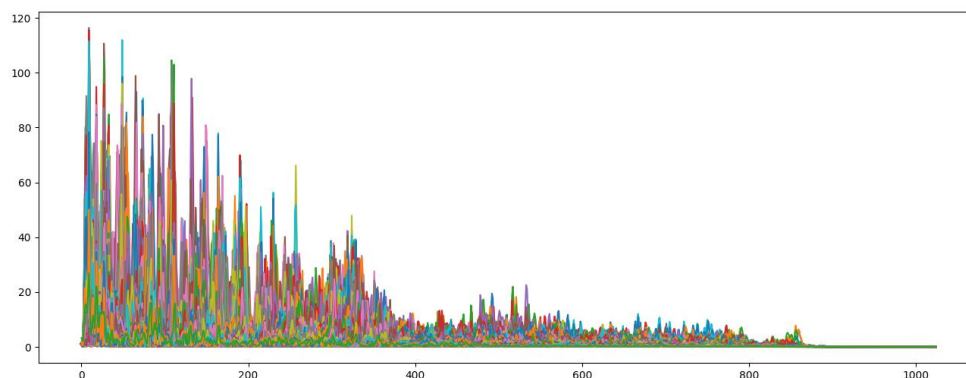
Dzięki wykorzystaniu mocy algorytmu KNN i zastosowaniu technik takich jak PCA i wybór cech, ten projekt skutecznie zwiększył dokładność klasyfikacji gatunku piosenek.

## 2D Representation: Sound Waves

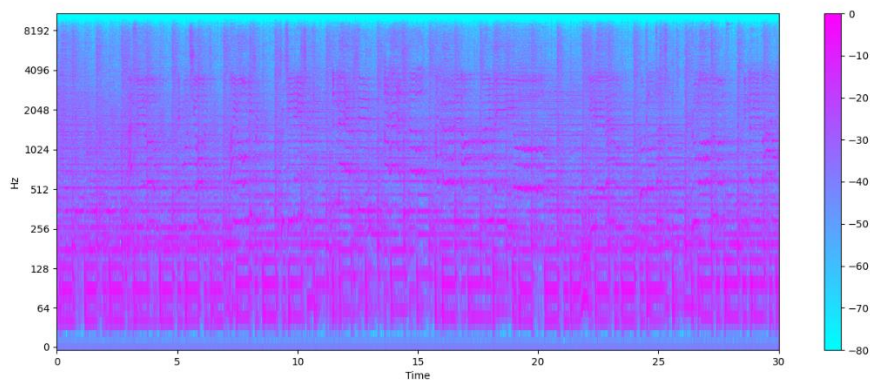




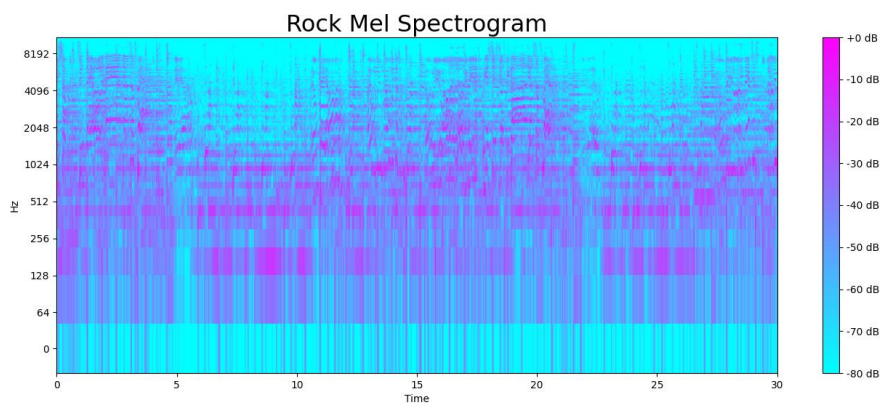
**Krótkoczasowa transformata Fouriera (STFT):** Jest to technika analizy sygnału dźwiękowego, która rozkłada sygnał na jego składowe częstotliwościowe w różnych chwilach czasowych. Dzięki temu można uzyskać informacje o spektralnym charakterze dźwięku w zależności od czasu.



**Spektrogramy:** Spektrogramy są wizualnymi reprezentacjami czasowo-częstotliwościowymi sygnałów dźwiękowych. Wykorzystują one przekształcenie Fouriera, aby przedstawić zmienność energetyczną dźwięku w zależności od czasu i częstotliwości. Spektrogramy składają się z zestawu ram czasowych, a dla każdej ramki oblicza się rozkład widmowy. Wykresy te pokazują intensywność sygnału dźwiękowego w różnych częstotliwościach w zależności od czasu.



**Mel Spektrogramy:** Mel spektrogramy to przekształcenie spektrogramu, które uwzględnia ludzką percepcję dźwięków. Wykorzystują skalę Mel, która jest nieliniową skalą częstotliwości, odzwierciedlającą sposób, w jaki ludzkie ucho odbiera różnice w wysokości tonów. Proces mel-skali obejmuje konwersję częstotliwości w skali Herta na skalę Mel za pomocą specjalnej funkcji transformacji. Mel spektrogramy są szczególnie przydatne w analizie muzycznej i rozpoznawaniu mowy, ponieważ lepiej odzwierciedlają percepcyjne cechy dźwięków.

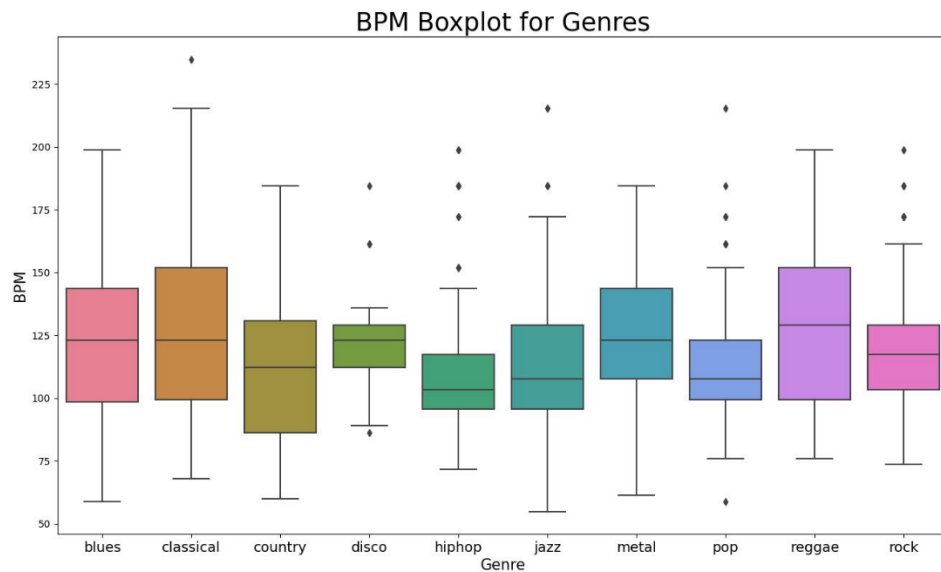


**Zero Crossing Rate (ZCR):** Zero Crossing Rate mierzy częstotliwość, z jaką sygnał zmienia się z wartości dodatnich na wartości ujemne i odwrotnie. Daje informację o szybkości zmiany sygnału lub obecności nagłych przejść w fali dźwiękowej. W kontekście analizy dźwięku pomaga nam zrozumieć, jak szybko fala dźwiękowa oscyluje między wartościami dodatnimi i ujemnymi.

**Harmonia (Harmony):** Harmonia dotyczy stosunku i interakcji różnych dźwięków w kontekście tworzenia akordów i harmonii muzycznej. Odnosi się do tworzenia przyjemnego brzmienia i emocjonalnego oddziaływania muzyki.

**Cechy percepcyjne (Perceptual Features):** Cechy percepcyjne to cechy dźwięku, które mają związek z tym, jak są postrzegane i interpretowane przez nasze zmysły. Obejmują takie elementy jak głośność, balans tonalny i ogólne cechy dźwięku.

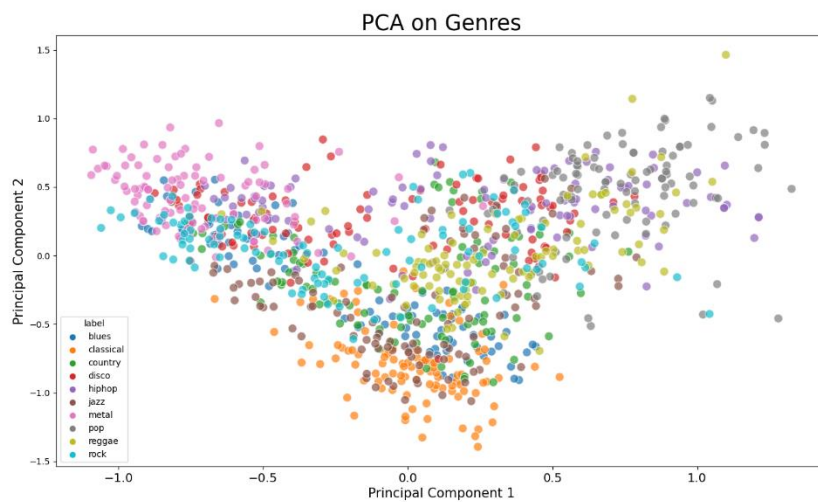
**BPM (Beats Per Minute):** BPM, czyli "Beats Per Minute", odnosi się do liczby uderzeń (beats) występujących w jednej minucie w muzyce. Jest to miara tempo utworu i jest wyrażana w jednostce uderzeń na minutę.

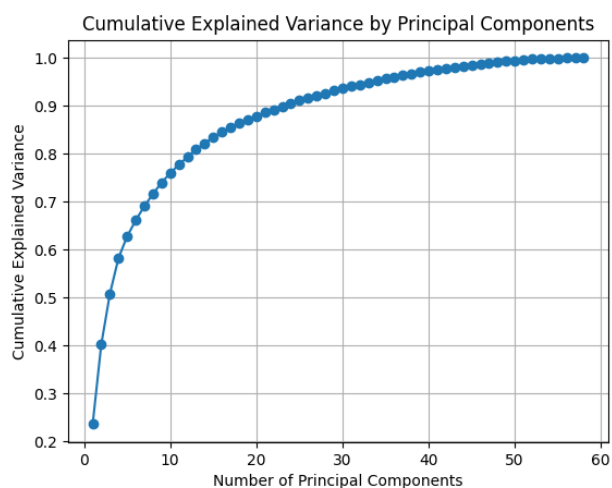


**Środek widmowy (Spectral Centroid):** Środek widmowy jest miarą średniej częstotliwości występującej w spektrum dźwięku. Odnosi się do "barwy" dźwięku, pozwalając określić, czy dźwięk jest wysoki czy niski.

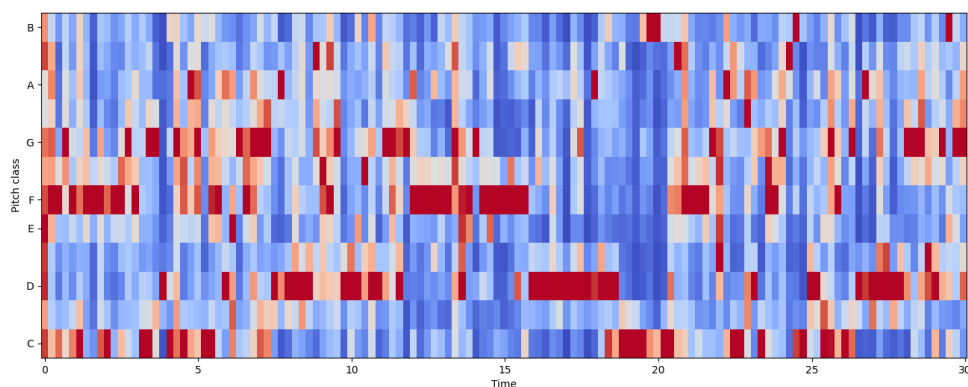
**Punkt przecięcia widma (Spectral Rolloff):** Punkt przecięcia widma to wartość procentowa, która określa, powyżej której częstotliwości znajduje się określony procent całkowitej energii dźwięku. Pomaga zidentyfikować, jak szybko zmienia się dźwięk w spektrum.

## PCA

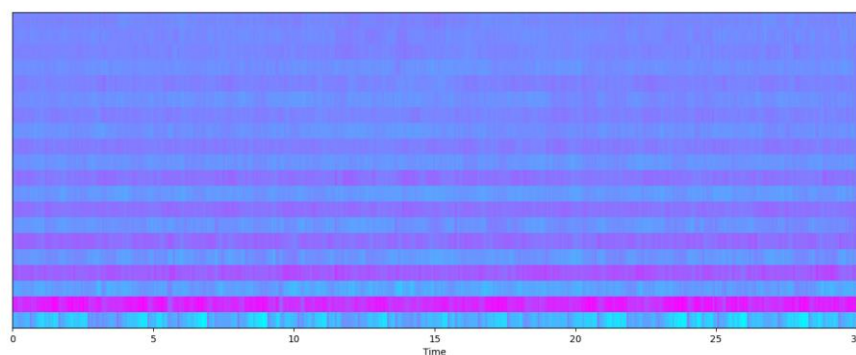


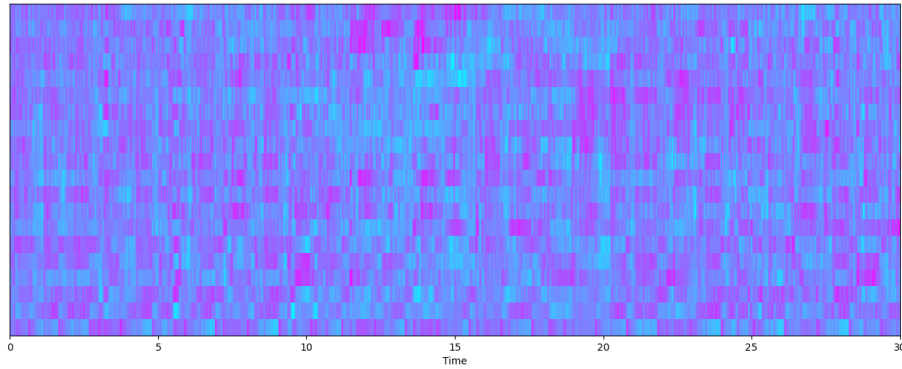


**Chroma:** Chroma odnosi się do wektorowej reprezentacji wysokości dźwięków w utworze muzycznym. Wykorzystuje się go do analizy różnorodności występujących dźwięków i określenia ich tonalności.



**Mel-Frequency Cepstral Coefficients (MFCC):** MFCC to cechy, które reprezentują kształt dźwięku w dziedzinie częstotliwości Mel. Są to współczynniki cepstralne, które przechwytują informacje o tonalności, barwie i cechach dźwięku. Są szczególnie przydatne w rozpoznawaniu mowy i analizie dźwięku.





**Tempo:** Tempo to miara szybkości lub tempa muzyki. Określa, jak szybko lub wolno występują kolejne uderzenia lub zmiany w rytmie utworu.

**Szerokość widmowa (Spectral Bandwidth):** Szerokość widmowa określa rozpiętość częstotliwości obecnych w spektrum dźwięku. Wskazuje na rozproszenie lub koncentrację energii dźwięku w różnych częstotliwościach.

**Root Mean Square (RMS):** RMS to miara średniego poziomu amplitudy sygnału dźwiękowego. Jest obliczana jako pierwiastek kwadratowy z średniej wartości kwadratów próbek sygnału. Informuje nas o ogólnym natężeniu dźwięku.

#### Music-Genres/

- | - data\_processing.py
- | - knn.py
- | - music\_analyzer.py
- | - main.py
- | - dataset

**data\_processing.py:** Ten plik zawiera klasę `DataProcessing` z metodami do mieszania danych, skalowania min-max i podziału danych.

**knn.py:** Ten plik zawiera klasę `KNN` z metodami do wykonywania algorytmu k-najbliższych sąsiadów, grupowania danych i obliczania dokładności.

**music\_analyser.py:** Ten plik zawiera klasę `MusicAnalyser` z metodą do wyszukiwania podobnych piosenek na podstawie podobieństwa kosinusowego.

**main.py:** To główny plik skryptu, w którym rozpoczyna się wykonanie projektu. Importuje on potrzebne klasy z innych plików, przetwarza dane, stosuje algorytm KNN i wykorzystuje klasę `MusicAnalyser` do wyszukiwania podobnych piosenek.