IUM-Fijalkowski-Niewiarowski

Repository for realization of 2024Z IUM project

Cytat tematu:

"Gdybyśmy tylko wiedzieli, kiedy użytkownik będzie chciał przesłuchać bieżący utwór w całości, a kiedy go przewinie – moglibyśmy lepiej zorganizować nasz cache"

Analiza tematu

Definicja problemu biznesowego

Stworzenie modelu klasyfikującego czy użytkownik nie pominie danego utworu. Przyda się to w celu stwierdzenia potrzeby cache-owania konkretnego utworu. Czyli jeśli dla danego utworu dany użytkownik będzie go najprawdopodobniej pomijał, nie ma potrzeby cache-owania go. W innym przypadku w celu usprawnienia działania aplikacji dla użytkownika możemy dany utwór cache-ować.

Zdefiniowanie zadania modelowania

Zadanie

Zajmujemy się klasyfikacją binarną akcji użytkowników. Staramy się określić czy dany użytkownik pominie lub nie pominie dany utwór na podstawie jego preferencji i atrybutów danego utworu. ### Założenia - Użytkownicy mają w miarę stałe preferencje muzyczne (nie zmieniają ich co dwa tygodnie) - Analitycy dostarczają poprawne i pełne dane - Nie rozróżnimy typów pominięcia utworu w zależności od czasu przesłuchanego utworu, utwór pominięty w 1/2 lub w 1/4 jego trwania jest tak samo pominiętym utworem

Model bazowy

Opis

Jako model bazowy stworzyliśmy program który sprawdza czy typ danej piosenki jest wśród lubianych typów muzyki danego użytkownika i definiowaliśmy, że pominie ten utwór jeśli nie jest ### Wyniki Tablica pomyłek dla modelu bazowego:

```
[  255  1386]
[  102 3876]
```

Inne metryki: - Dokładność: 0.29 - Recall: 0.71 - Precyzja: 0.06

Kryteria sukcesu

Biznesowe

Celem jest przyspieszenie działania aplikacji poprzez cache-owanie tylko potrzebnych utworów, więc odpowiednim kryterium sukcesu będzie zbieranie metryk dotyczących na ile dokładne były nasze predykcje w środowisku produkcyjnym. Pozwoli to określić czy nasz model w realny sposób usprawnia działanie aplikacji. Ze względu na specyfikę zadania można zbierać te metryki w czasie rzeczywistym po czym prezentować je w zintegrowanych systemach (użyć w tym celu można np. prometeusza i grafany). ### Analityczna Model bazowy ma bardzo niską dokładność 29%, naszym zadniem będzie utworzenie modelu o większej dokładności

Analiza danych

Klient udostępnił nam dane dotyczące:

- lista dostępnych artystów i utworów muzycznych,
- baza użytkowników,
- historia sesji użytkowników,
- techniczne informacje dot. poziomu cache dla poszczególnych utworów.

Z analizy dostarczonej nowej wersji danych wyciągneliśmy następujące wnioski:

- W drugiej wersji danych, otrzymaliśmy dane 5000 użytkowników, co stanowi duży wzrost względem pierwszej wersji danych kiedy to mieliśmy dane tylko 50 użytkowników. pola opisujące użytkownika są w czytelny sposób opisane. Dodatkowo dane dot. użytkowników są kompletne.
- W przypadku danych o utworach dostępnych w serwsie pozytywka otrzymaliśmy informację o około ~130 000 utworów, dane są w przeważającej części kompletne poza jedną kolumną (mode), w przypadku tej kolumny w 80% (103718) rekordów jej wartość jest pusta.
- Plik trac_storage zawiera informacje o przechowywaniu poszczególnych utworów w cache, informacja o klasie pamięci w jakiej jest przechowywany oraz koszt obsługi każdego utworu. W dostarczonych danych można zauważyć bardzo małą ilość utworów dla których storage_mode jest inna niż 'Slow'. Dla poszczególnych wartości liczby rekordów prezentują się w następujący sposób:

  Liczba wystąpień dla każdej klasy pamięci:

  - Slow: 128182
  - Medium: 1459
  - Fast: 7

  Procentowy rozkład został zaprezentowany na poniższym wykresie:

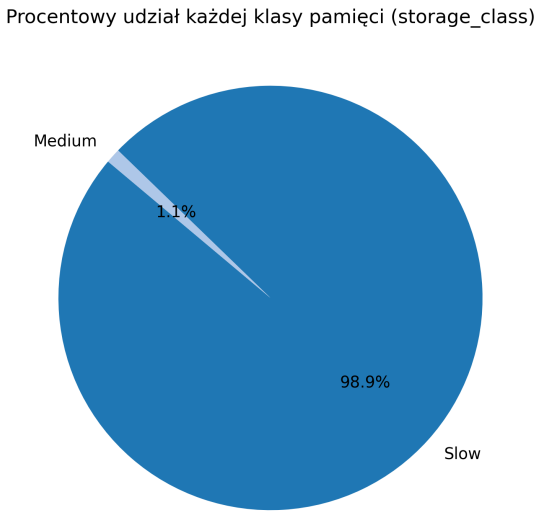


Figure 1: Opis alternatywny

- W pliku sessions otrzymaliśmy informację o około ~4.7*10^6 akcjach podjętch przez użytkowników dla wersji danych v1, wszytskie rekordy są kompletne. Z struktury danych możemy wnioskować że rekordy opisują akcje podejmowane przez użytkownika, użytkownik zazwyczaj w czasie sesji odtwarza kilka utworów, każdy z tych utworów może być odwtworzony w całości, polubiony bądź pominięty na wybranym etapie odtwarzania. Pola opisujące sesje w przeciwieństwie do pierwszej wersji danych zostały czytelnie opisane, co ułatwiło zrozumienie struktury danych opisujących sesje. W ramach dostarczonych danych udostępnione zostały informacje o ~140 000 unikalnych sesjach.

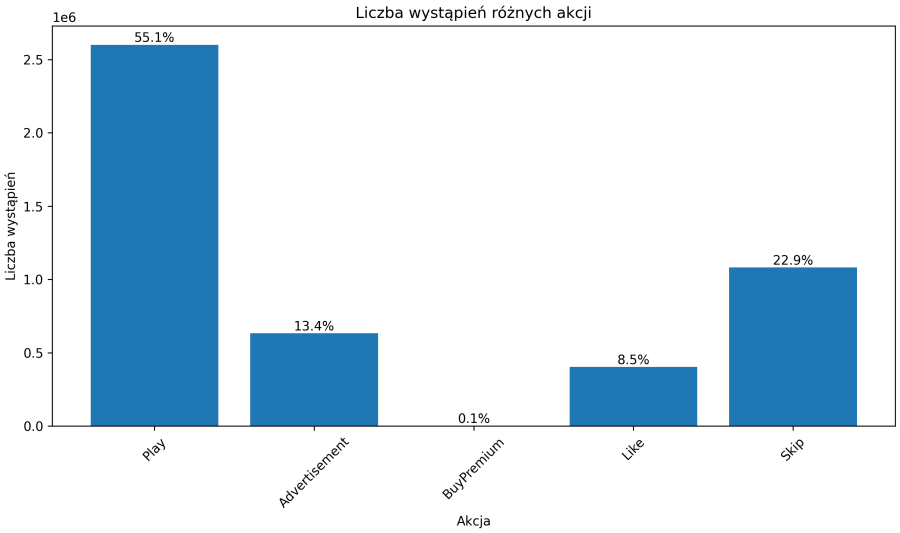


Figure 2: Opis alternatywny

- Dodatkowo wygenerowaliśmy wykresy ukazujące klasę pamięci dla akcji 'Skip' oraz 'Play', jednak nie staraliśmy się wyciągać jakichkolowiek przypuszczeń bądź wniosków ze względu na duży udział klasy 'Slow' w udostępnionych danych.

Liczba dostępnych danych na temat sesji wzrosła w porównianiu do pierwszej wersji danych. Ukazują to dwa sporządzone wykresy, na pierwszym z nich widzimy ilość utworów które wystąpują co najmniej raz w pliku sessions w pierwszej wersji danych, widać że większość utworów, nie było ujętych w zapisach sesji. W później dostarczonych danych, przeważająca część utorów była ujęta w pliku sessions.

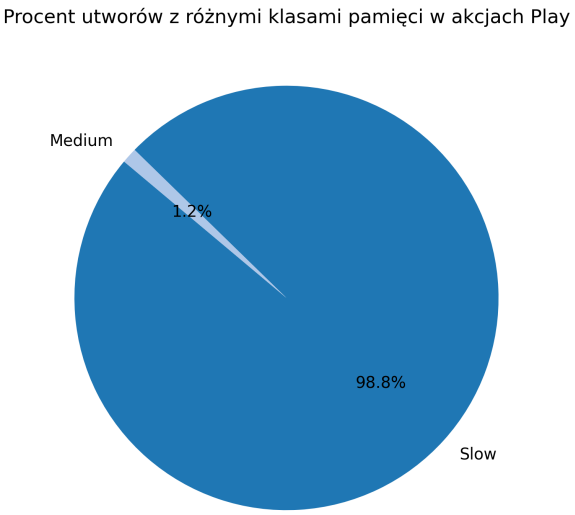


Figure 3: Opis alternatywny

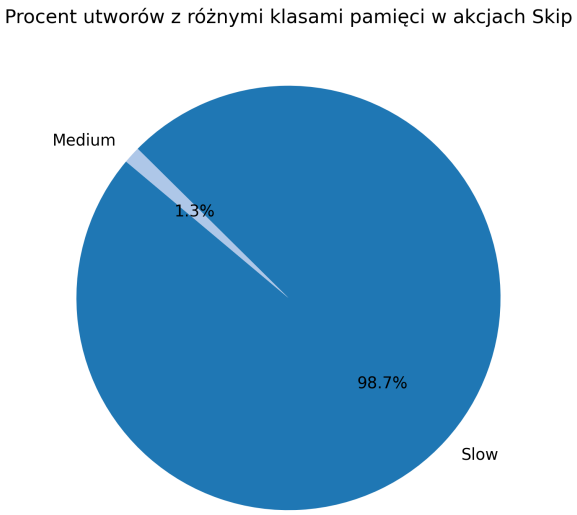
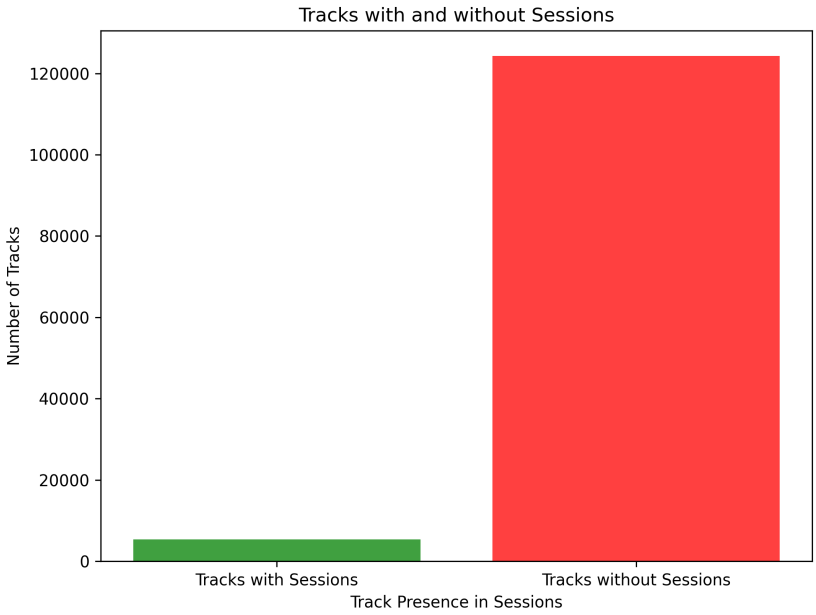
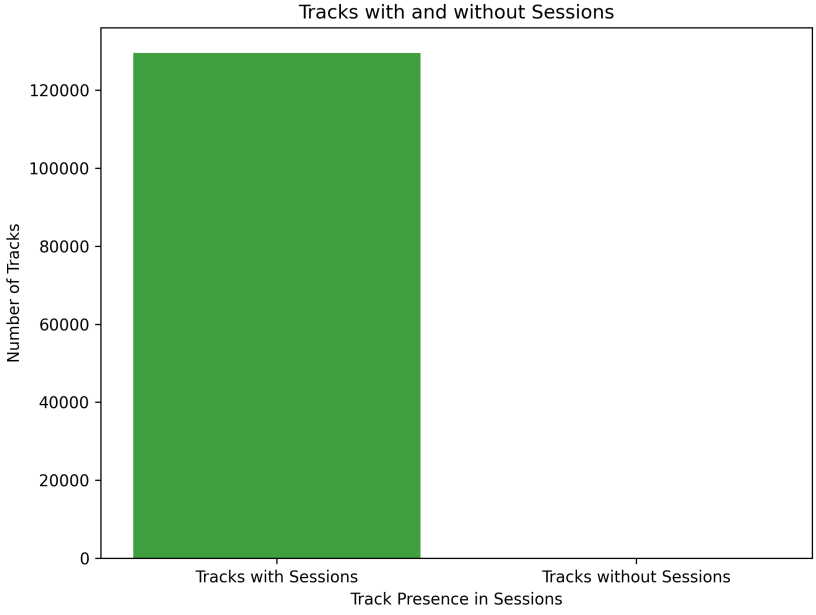
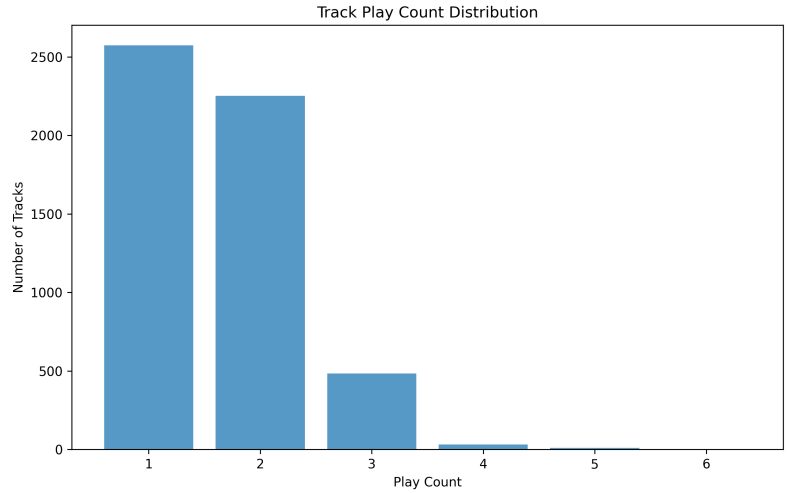


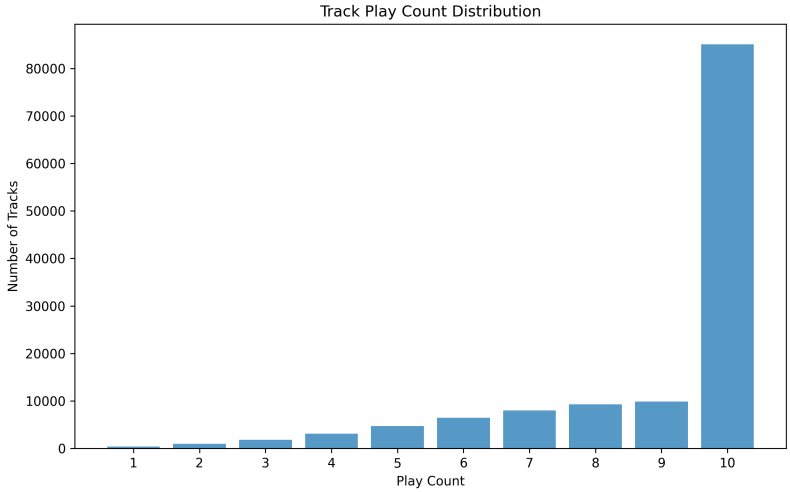
Figure 4: Opis alternatywny





Kolejny wykres ukazuje ilość wystąpień uworów w pliku sessions:





Porównując oba wykresy, również można zauważyć iż druga wersja danych była o wiele bardziej reprezentatywna.

W drugiej wersji danych, unikalne utwory pojawiały się najczęściej po 10 lub więcej razy w pliku sessions, w przeciwieństwie do 1 wersji danych gdzie najczęściej utwory występowały 1 lub 2 krotnie.

- W pliku artists otrzymaliśmy dane na temat 27650 artystów. Dla wszystkich wierszy dane są kompletne, występuje to dla pierwszej oraz ostatniej kolumny.

Dodatkowo sprawdziliśmy jaki rodzaj muzyki występuje w dostarczonych danych

nt. sesji użytkownika, rodzaj muzyki ustalaliśmy na bazie podaych gatunków muzycznych dla artystów przypisanych do konkretnych utworów. Wykres prezentuje się w następujący sposób:

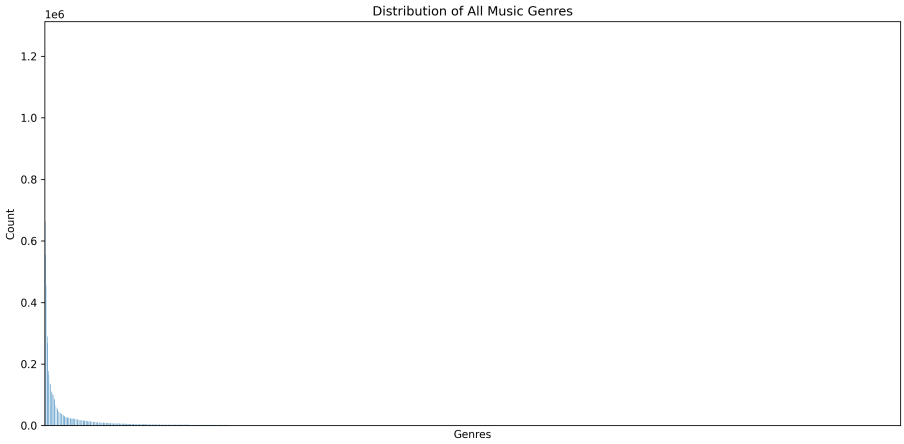


Figure 5: Opis alternatywny

Na wykresie możemy zauważyć istotną różnicę w występowaniu poszczególnych gatunków muzycznych w dostarczonych danych na temat sesji użytkowników, część z gatunków muzycznych jest częściej proponowana przez aplikację.

Wnioski wyciągnięte na podstawie przeprowadzonej analizy

- Nieproporcjonalna liczba sesji użytkownika w stosunku do liczy utoworów:
  - Otrzymano dane dla ~130 000 utworów, oraz tylko ~4.7*10^6 akcji użytkowników. Informacje na temat akcji zawierały zapisy z odtwarzania większości utworów dostępnych w aplikacji.
  - Większość unikalnych utworów w sesjach pojawiła się 10 lub więcej razy. Oznacza to większą niż w 1 wersji danych ilość danych treningowych dla modelu dotyczącego popularności utworów, jednak ilość danych dot. sesji dalej mogłaby być większa, co zapewniłoby większą reprezentatywność.
  - Zwiększona liczba sesji może pomóc w opracowaniu lepszego modelu, oferującego wyższą jakość predykcji.
- Nadreprezentacja klasy pamięci "Slow":
  - W pliku "trac_storage" zdecydowana większość utworów (98.9%) znajduje się w klasie pamięci "Slow". Inne klasy, takie jak "Medium" i "Fast", są marginalnie reprezentowane, co ogranicza różnorodność danych do analizy.
- Większa ilość danych na temat użytkowników:
  - W pliku Users znajdują się informacje na temat tylko 5000 użytkowników, co stanowi o wiele większą grupę reprezentacyjną niż w przy-

padku 1 wersji danych.
- Czytelne opisy dostarczonych danych:
  - W porównaniu do 1 wersji danych, czytelne opisy ułatwiają biznesowe zrozumienie dostarczonych danych.

Model uczelnia maszynowego

Przygotowanie danych

Zostały połączone dane plików: sessions.jsonl, tracks.jsonl, users.jsonl. W celu wytworzenia jednego zbioru danych.

Informacje o sesjach zostały przefiltrowane następująco: - Nie interesują nas wiersze z akcjami innymi niż 'skip' i 'play' - Jeśli dla danej sesji i dla danego utworu została wykonana akcja 'skip' i 'play' oznacza to, że utwór został pominięty, tak więc zostawiamy tylko wiersz z operacją 'skip'

Na podstawie sesji zostały również wyliczone dwa dodatkowe atrybuty: - Średnio ile procent utworów użytkownik pomijał-pokazuje to na tendencję użytkownika do pomijania utworów - Średnio ile procent dany utwór był pomijany-pokazuje to jak często dany utwór jest pomijany

Ulubione gatunki użytkownika zostały czytane z pliku users.jsonl i zmienione do postaci binarnych atrybutów pokazujących czy użytkownik lubi dany gatunek muzyczny.

Podział danych

Dane zostały podzielone na dane uczące, dane walidacyjne oraz dane przeznaczone do przeprowadzenia testu AB, poprzez wybranie po 1000 danych o losowych indeksach i przyniesienie ich z danych uczących ich do danych walidujących oraz danych do testów AB.

Dane zostały następnie zapisane do folderu data/processed w postaci następujących plików csv: - merged_data.csv - dane uczące do modelu - y.csv - etykiety dla danych uczących - validation_data.csv - dane walidujące - validation_classes.csv - etykiety dla danych walidujących - ab_test_data.csv

Macierz korelacji cech

Macierz korelacji opracowana z wykorzytsaniem skryptu `corelation.py`

Model

Został wykorzystany model LiniearSVC z domyślnymi parametrami. Pierwsza wersja modelu bez dodatkowych atrybutów opisanych powyżej osiągnęła następujące wyniki:

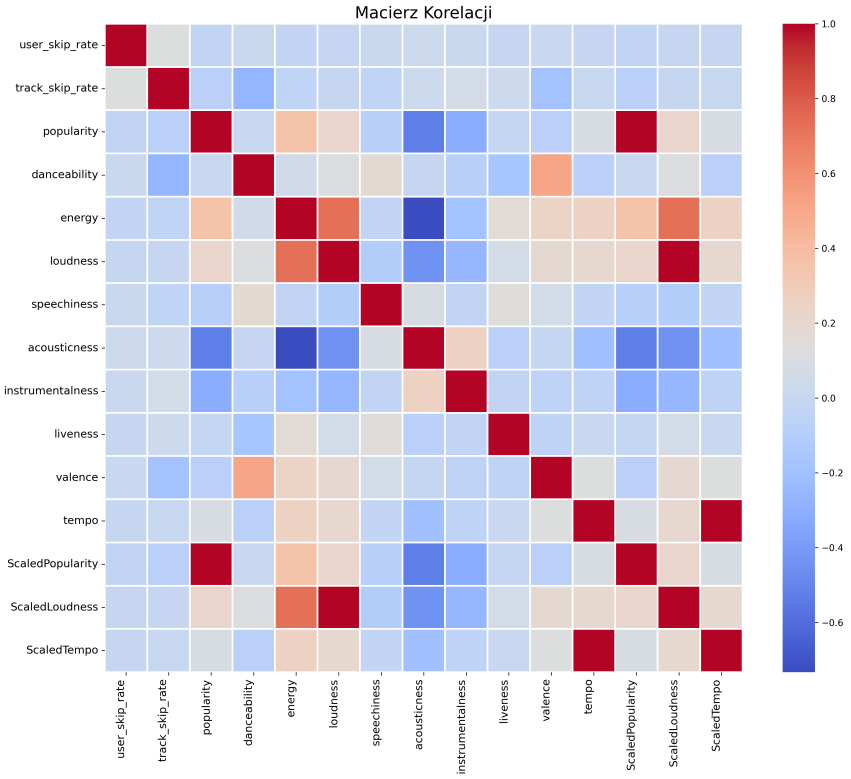


Figure 6: Opis alternatywny

	precision	recall	f1-score	support
0	0.59	0.05	0.10	430
1	0.58	0.97	0.72	570
accuracy			0.58	1000
macro avg	0.58	0.51	0.41	1000
weighted avg	0.58	0.58	0.45	1000

Po zmienie balansu klas w modelu na 'balanced' otrzymaliśmy następujące wyniki:

	precision	recall	f1-score	support
0	0.45	0.51	0.48	430
1	0.59	0.52	0.55	570
accuracy			0.52	1000
macro avg	0.52	0.52	0.51	1000
weighted avg	0.53	0.52	0.52	1000

Po dodaniu dodatkowych atrybutów opisanych powyżej otrzymaliśmy następujące wyniki:

	precision	recall	f1-score	support
0	0.68	0.74	0.71	429
1	0.79	0.73	0.76	571
accuracy			0.74	1000
macro avg	0.73	0.74	0.73	1000
weighted avg	0.74	0.74	0.74	1000

Mikroserwis

Mikroserwis został zaimplementowany przy użyciu frameworka FastAPI, który umożliwia serwowanie predykcji dwóch modeli: naiwnego i docelowego. Serwis zawiera trzy główne endpointy:

1. /naive - obsługuje predykcje wykonywane przez model naiwny (BaseModel).
2. /recommend - zwraca predykcję wygenerowaną przez model docelowy (NormalModel).
3. /abtest - realizuje test A/B, losowo wybierając jeden z dwóch modeli (z równym prawdopodobieństwem) i zwracając jego predykcję. Informacje o wybranym modelu, użytkowniku, utworze i wyniku predykcji są zapisywane w pliku logów (model_usage_log.json). Dane wejściowe są przesyłane jako obiekt JSON, opisany przez model Pydantic (PredictionRequest). Każda predykcja jest przetwarzana przez odpowiedni model, a jej wynik (np.

PLAY lub SKIP) jest zwracany klientowi i zapisywany do logów w celu późniejszej analizy. Logi są inicjalizowane przy pierwszym uruchomieniu aplikacji, jeśli plik logów nie istnieje.

Instrukcja uruchomienia

Zainstaluj wymagane zależności:

```
pip install -r requirements.txt
```

Dodaj aktualą wersję danych w lokalizacji:

```
./data/raw/v2
```

Lista wymaganych plików: - sessions.jsonl - tracks.jsonl - users.jsonl - artists.jsonl

Uruchom Mikroserwis używając uvicorn:

```
uvicorn api.app:app --host 0.0.0.0 --port 8000
```

Przykładowe zapytanie z poziomu PowerShell:

```
Invoke-RestMethod -Uri "http://127.0.0.1:8000/recommend" `
>> -Method POST `
>> -Headers @{ "Content-Type" = "application/json" } `
>> -Body '{"user_id": 101, "track_id": "31PzY79H10HCgJs533Xq6B"}'
```

Testy AB

Testy A/B w mikroserwisie są realizowane za pośrednictwem dedykowanego endpointu /abtest, który wybiera jeden z dwóch modeli (naiwny lub docelowy) z równym prawdopodobieństwem 50/50. Dla każdego żądania serwis:

Losowy wybór modelu: Decyduje, czy predykcja zostanie wygenerowana przez model naiwny (BaseModel), czy docelowy (NormalModel). Wykonanie predykcji: Wybrany model generuje wynik dla danego użytkownika (user_id) i utworu (track_id), wskazując, czy utwór zostanie odtworzony (PLAY) czy pominięty (SKIP). Logowanie wyników: Informacje o wybranym modelu, wejściowych danych użytkownika i wyniku predykcji są zapisywane w pliku logów (model_usage_log.json). Każdy wpis logu zawiera nazwę modelu, identyfikator użytkownika i utworu oraz wynik predykcji.

Aby ułatwić przeprowadzenie testów opracowany został skrypt wczytujący wcześniej przygotowane dane i wysyłający zapytanie do mikroserwisu.

Wyniki przeprowadzonego testu:

Dla obu modeli otrzymaliśmy następującą accuracy:

```
    model  accuracy
0   naive  0.432075
1  target  0.744681
```

W dobry sposób prezentuje to sporządzony wykres:

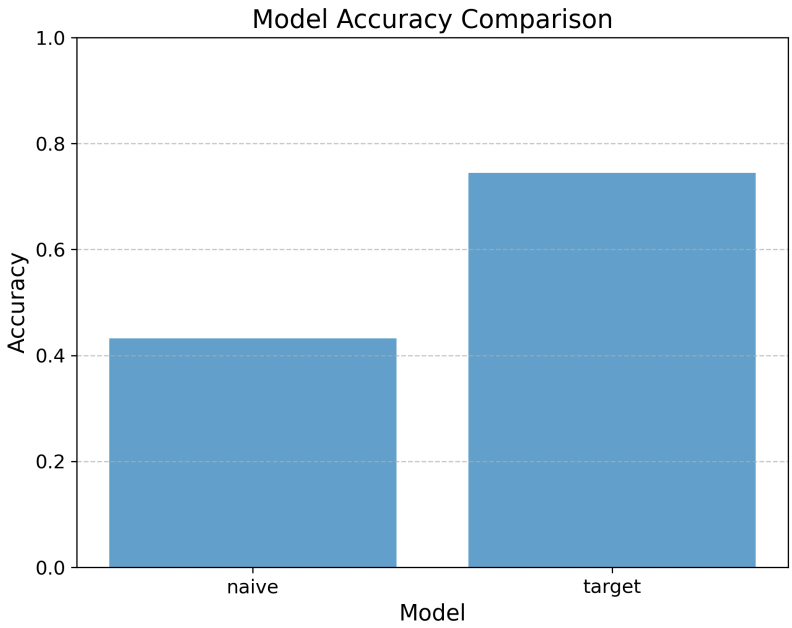


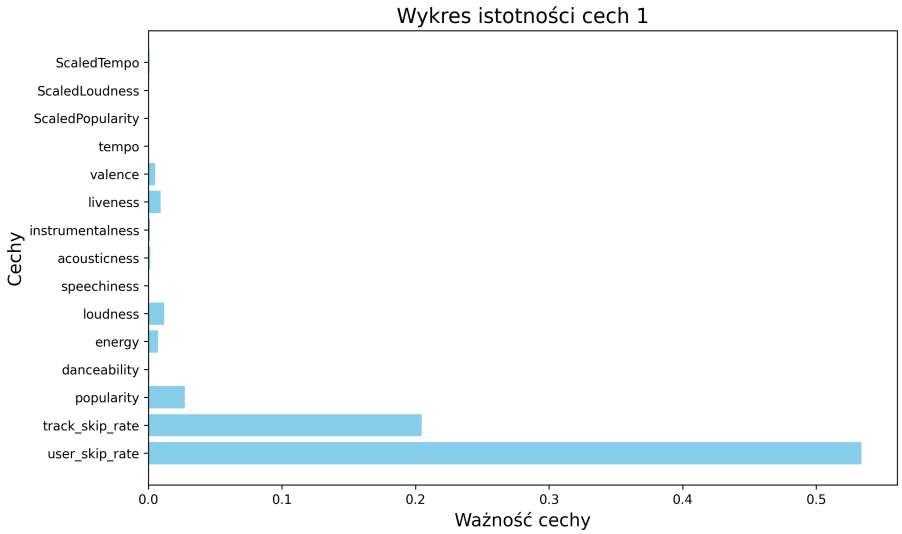
Figure 7: Opis alternatywny

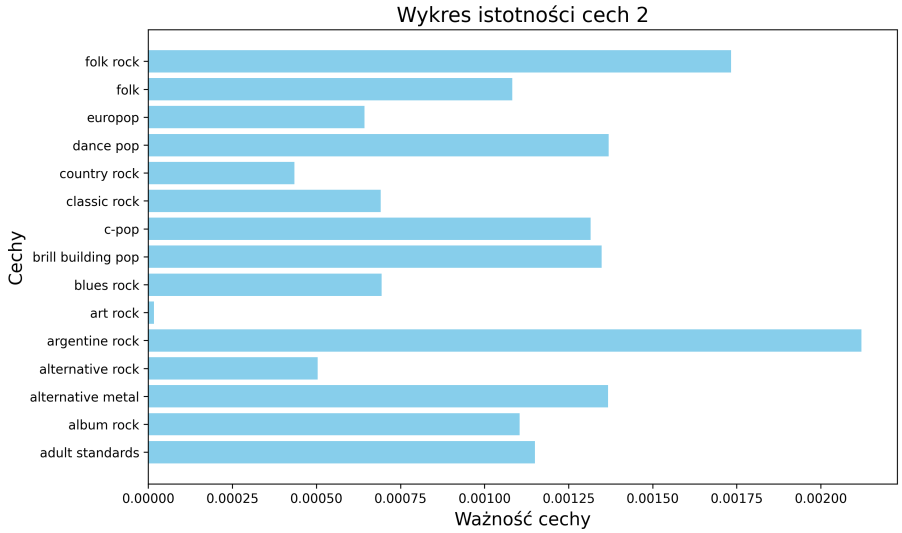
Wnioski na podstawie testów AB

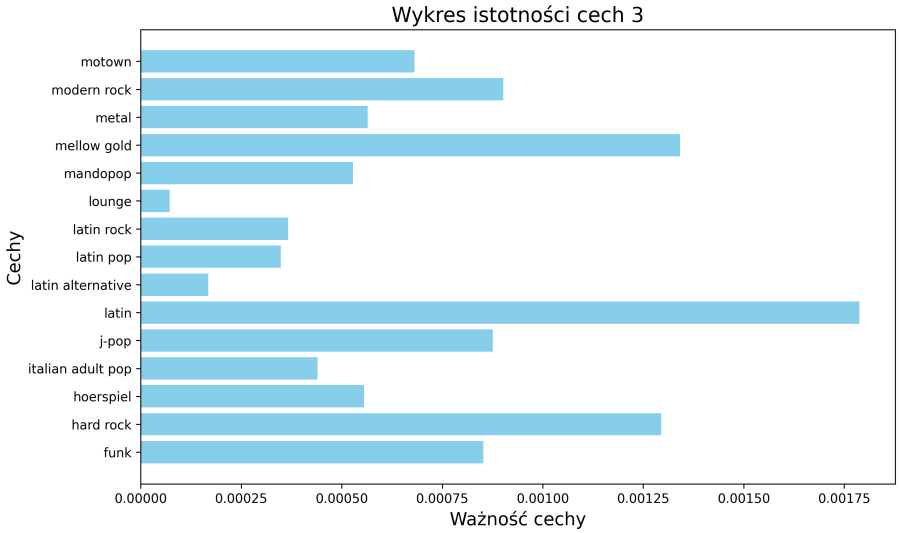
1. Uzyskane wyniki testu A/B wyraźnie wskazują, że model docelowy (target) osiąga znacznie lepszą skuteczność (accuracy: 74,47%) w porównaniu do modelu naiwnego (naive), którego dokładność wynosi jedynie 43,2%. Taka różnica sugeruje, że model docelowy lepiej przewiduje zachowania użytkowników, co czyni go bardziej wartościowym w praktycznym zastosowaniu, np. w systemach rekomendacyjnych.

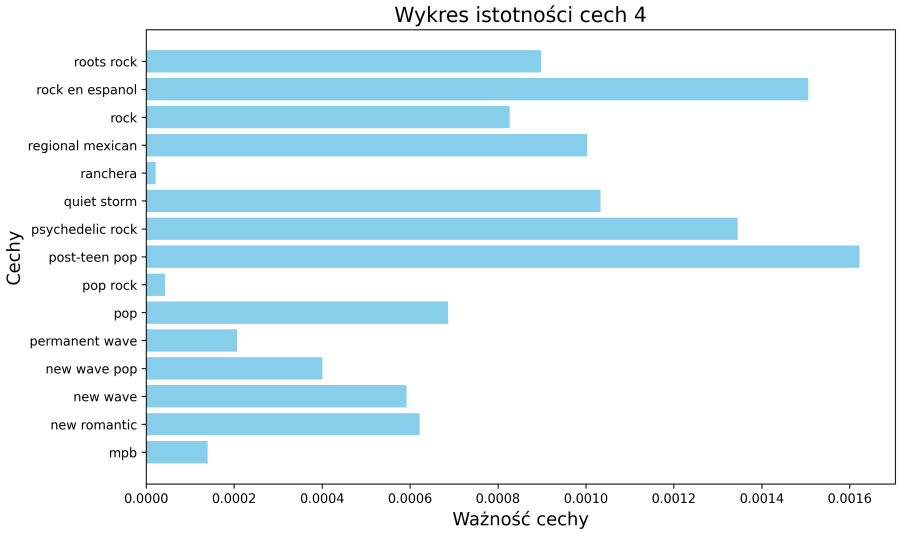
2. Sporządzony wykres nie tylko wizualizuje tę różnicę, ale także podkreśla istotność prowadzenia testów A/B jako narzędzia do podejmowania decyzji opartych na danych. Wyniki wskazują, że wdrożenie modelu docelowego może znacząco poprawić jakość rekomendacji i zadowolenie użytkowników.

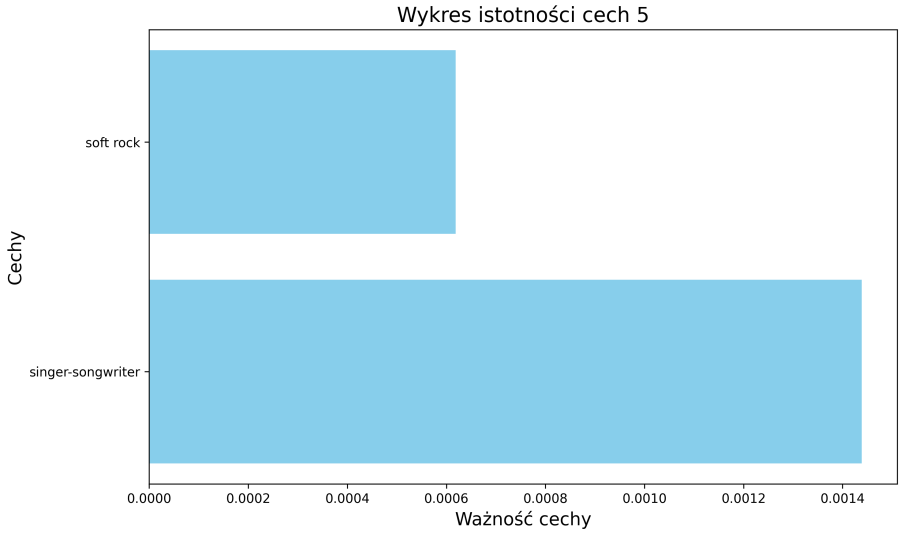
Istotność Cech:






Wykres istotności cech 3
motown
modern rock
metal
mellow gold
mandopop
lounge
latin rock
latin pop
latin alternative
latin
j-pop
italian adult pop
hoerspiel
hard rock
funk
Cechy
0.00000
0.00025
0.00050
0.00075
0.00100
0.00125
0.00150
0.00175
Ważność cechy


Wykres istotności cech 4
roots rock
rock en espanol
rock
regional mexican
ranchera
quiet storm
psychedelic rock
post-teen pop
pop rock
pop
permanent wave
new wave pop
new wave
new romantic
mpb
Cechy
0.0000
0.0002
0.0004
0.0006
0.0008
0.0010
0.0012
0.0014
0.0016
Ważność cechy



Wnioski na podstawie wykresów istotności cech: 1. Analizując przedstawione wykresy można dostrzec że największy wpływ na działanie modelu mają cehcy takie jak: track_skip_rate, user_skip_rate oraz popularność danego utworu.

2. Gatunek muzyczny jaki dany utwór reprezentuje mają nieznaczny wpływ na predykcje modelu, jednak występują między poszczególnymi gatunkami istotne różnice.