



Z przedstawionych danych zdecydowaliśmy się usunąć: tempo, valence, liveness, liveness, speechiness, key i key. Zrobiliśmy to ze względu na dużą korelację z innymi parametrami, bardzo małe wartości MI lub małe różnice pomiędzy wartościami MI dla danych normalnych i zmieszanych.

Pomimo dużej korelacji zdecydowaliśmy się zostawić wszystkie argumenty: month\_0\_play\_count, month\_0\_skip\_count, month\_0\_like\_count ze względu na ich duże wartości MI i lepsze wyniki modeli uwzględniających wszystkie te argumenty. Usunęliśmy natomiast te dane dla poprzednich miesięcy.

Finalne dane za których uczone są modele zawierają więc: popularity, duration\_ms, explicit, id\_artist, release\_date, danceability, loudness, acousticness, genres, month\_1\_skip\_count, month\_1\_play\_count, month\_1\_like\_count. Dla danych genres i id\_artist, które są dyskretne zastosowana została metoda kodowania one hot. Dane ciągłe zostały znormalizowane.

## Zmiany względem poprzednich założeń

Ponieważ większość piosenek nie była nigdy odsłuchana ani polubiona to założone kryterium sukcesu, że utwory wskazane przez model będą bardziej popularne od mediany wszystkich utworów okazało się bardzo proste w realizacji i zostało spełnione przez oba modele. W związku z tym postanowiliśmy dodać nową miarę oceny modelu. Sprawdzamy ile piosenek z playlist top 10 dla każdego gatunku i top 20 dla wszystkich piosenek faktycznie znalazło się w najbardziej popularnych utworach następnego miesiąca.

## Model prosty

Jako model podstawowy użyliśmy model regresji liniowej. Wykorzystaliśmy implementację z biblioteki sklearn. Model przewiduje założona przez nas metrykę popularności utwory w następnym miesiącu. Model ma jeden hiperparametr alfa którego wartość została za pomocą grid\_sercha sprawdzona w zakresie od 0 do 1 i ustawiona na  $\alpha = 0,58$ .

Model osiągnął następujące wyniki na zbiorze testowym:

Mean Absolute Error: średnia - 18,5 ,odchylenie standardowe: - 0,2

Mean Squared Error: średnia - 870,2 ,odchylenie standardowe: - 3,1

R<sup>2</sup> Score: średnia - 0,979 ,odchylenie standardowe: - 0,005

Wskazane piosenki w najbardziej popularnych następnego miesiąca:

Pop: średnia 6,7/10 - ,odchylenie standardowe: - 1,1

Rock: średnia 8,8/10 - ,odchylenie standardowe: - 0,5

Country: średnia 6,3/10 - ,odchylenie standardowe: - 2,8

Rap: średnia 7,4/10- ,odchylenie standardowe: - 1,6

Hip hop: średnia 9,1/10 - ,odchylenie standardowe: - 0,9

All songs: średnia 14,2/20 - ,odchylenie standardowe: - 3,1

## Model zaawansowany

Jako modelu zaawansowanego użyliśmy sieci neuronowej napisanej z wykorzystaniem biblioteki torch. Wielkość sieci i inne hiper parametry zostały ustalone za pomocą ręcznych testów i doświadczenia autorów projektu.

Model osiągnął następujące wyniki na zbiorze testowym:

Mean Absolute Error: średnia - 189 ,odchylenie standardowe: - 4,2

Mean Squared Error: średnia - 938 ,odchylenie standardowe: - 15,6

R<sup>2</sup> Score: średnia - 0,977 ,odchylenie standardowe: - 0,002

Wskazane piosenki w najbardziej popularnych następnego miesiąca:

Pop: średnia - 0,4/10 ,odchylenie standardowe: - 0,5

Rock: średnia - 0,3/10 ,odchylenie standardowe: - 0,4

Country: średnia - 5,3/10 ,odchylenie standardowe: - 1,8

Rap: średnia - 0,4/10 ,odchylenie standardowe: - 0,5

Hip hop: średnia - 0,5/10 ,odchylenie standardowe: - 0,7

All songs: średnia 0,9 - ,odchylenie standardowe: - 1

Niestety modelowi nie udało się przewidzieć większości najbardziej popularnych piosenek. Po sprawdzeniu konkretnych predykcji okazuje się że w większości mieszczą się one w liście stu najpopularniejszych utworów. Wyjątkiem jest tu gatunek Country w którym regularnie udawało się przewidzieć przynajmniej część piosenek. Wynika to prawdopodobnie z tego że gatunek ten nie jest popularny jako całość i często żaden utwór z top 10 country nie łapie się do top 20 wszystkich piosenek.

## Wnioski

Oba modele osiągnęły bardzo podobne wyniki jeśli chodzi o metryki. Model bazowy wyjadę się być nawet nieznacznie lepszy, chociaż różnice są małe. Dużą różnicę widać natomiast w dopasowaniu najbardziej popularnych piosenek. Model bazowy osiągnął tutaj bardzo dobre wyniki skutecznie przewidując większość utworów. Model zaawansowany z kolei nie przewidział poprawnie żadnego utworu w większości gatunków. Może to wynikać z lepszego dopasowania modelu podstawowego do problemu. Odchylenie standardowe na wszystkich metrykach jest stosunkowo małe w obu przypadkach, co świadczy o stabilności wyników. Oba modele spełniają postawione przez nas analityczne kryterium sukcesu. Czyli wskazują utwory bardziej popularne niż mediana wszystkich piosenek. Należy pamiętać, że ponieważ model przyjmuje dane z ostatniego miesiąca należy go trenować regularnie przynajmniej raz w miesiącu lub częściej.

## Mikroserwis

### Opis

Obraz Dockera z serwerem webowym zaimplementowanym z wykorzystaniem Flask. Mikroserwis jest odpowiedzialny za obsługiwanie zapytań GET, dla których zwraca odpowiedniego JSON dla zapytania oraz odpowiedniej grupy, do której został przyporządkowany użytkownik.

Podstawowa ścieżka (127.0.0.1:5001) wypisuje na ekranie do jakiej grupy należy użytkownik a w razie braku przynależności, przypisze losowo użytkownika do jednej z grup (nn lub regression) i poinformuje o tym wypisując komunikat na ekranie.

Mikroserwis obsługuje zapytania GET dla endpointu /<genre>. Początkowo obsługiwane gatunki to: pop, hiphop, country, rap, rock i top20. Mikroserwis po wysłaniu zapytania od użytkownika sprawdza czy użytkownik ma przypisaną grupę i w razie braku przypisania wywołuje funkcję, która losowo przypisze go do jednej z grup (nn lub regression).

Mikroserwis obsługuje zapytania POST dla endpointu /update/<model\_type>/<genre>. Umożliwia to aktualizowanie danych dla dowolnego gatunku, nawet jeżeli wcześniej nie istniał w systemie.

POST http://localhost:5001/update/regression/test?filename=data/regression/top\_10\_rap\_regr...

Params Auth Headers (7) Body Pre-req. Tests Settings Cookies

Query Params

	Key	Value	Bulk Edit
<input checked="" type="checkbox"/>	filename	data/regression/top_10_rap_regression.pkl	

*Przykładowe aktualizowanie danych, które umożliwi wypisanie wyników z pliku dla gatunku test*

Mikroserwis obsługuje zapytanie POST dla endpointu /toggle\_ab\_test. Pozwala to na włączenie i wyłączenie testu a/b. Można w tym także ustawić grupę domyślną która będzie grupą wszystkich użytkowników po wyłączeniu testu a/b.

POST http://localhost:5001/toggle\_ab\_test?Content-Type=application/json

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary JSON

```

1 {
2   "ab_test_enabled": false,
3   "default_group": "regression"
4 }
```

*Wyłączenie testu a/b oraz zmiana grupy domyślnej na regression*

POST http://localhost:5001/toggle\_ab\_test?Content-Type=application/json

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary JSON

```

1 {
2   "ab_test_enabled": true
3 }
```

*Włączenie testu a/b*

## Sposób uruchamiania

Uruchomienie aplikacji docker desktop.

Budowa obrazu dockera:

docker-compose build

Uruchomienie kontenera:

docker-compose up

Po tych komendach serwis będzie dostępny pod adresem <http://127.0.0.1:5001>.

## Endpoints

Serwis przyjmuje wyżej opisane zapytania GET dla których zwraca JSON o strukturze listy słowników:

```
{
  "song_id": string
  "song_name": string
},
```

Dla przykładowego zapytania:

GET

http://localhost:5001/regression/country

Send

Otrzymujemy odpowiedź:

```
1  {
2    {
3      "song_id": "6zFMeegAMYQo0mt8rXtrli",
4      "song_name": "HOLIDAY"
5    },
6    {
7      "song_id": "5tVA6TkbaAH9QMITTQRrNv",
8      "song_name": "Free Fallin'"
9    },
10   {
11     "song_id": "7MRyJPksH3G2cXHN8UKYzP",
12     "song_name": "American Girl"
13   },
14   {
15     "song_id": "5eMZ3EPIMGoWlrrVhxatdv",
16     "song_name": "You Got Lucky"
17   },
18   {
19     "song_id": "5x53pbGk6sbl1BGom19QQ5",
20     "song_name": "Breakdown"
21   },
22   {
23     "song_id": "4tSZr2100TY6upjNYfEYUI",
24     "song_name": "A Face In The Crowd"
25   },
26   {
27     "song_id": "4mcZYzoGwPRDdhWVlygiHf",
28     "song_name": "Refugee"
29   },
30   {
31     "song_id": "7gSQv10HpkIoAdUiRLdmI6",
32     "song_name": "I Won't Back Down"
33   },
34   {
35     "song_id": "3BSoskNONnjn05yR02tYFX",
36     "song_name": "Don't Come Around Here No More"
37   },
38   {
39     "song_id": "685sEpLJe80Zzuxz3siOVT",
40     "song_name": "Here Comes My Girl"
41   }
42 }
```

## Test A/B

Zadanie modelowania zdefiniowano jako wygenerowanie playlisty, z wybranego gatunku, której utwory zdobędą największą popularność w następnym miesiącu po wytrenowaniu modelu. Sprawdzenie czy spełnione zostało biznesowe kryterium sukcesu jest niemożliwe przed faktycznym wdrożeniem modelu. Za to kryterium analityczne zostało spełnione i poprawione względem pierwotnych oczekiwań. Bardzo dobre okazały się szczególnie playlisty wygenerowane przez model regresji. Dlatego sugerujemy dalej korzystać właśnie z tego modelu.