

Projekt IUM – 2020Z – zadanie 2 – etap 2

Adam Steciuk (300263)

Wojciech Moczydłowski (296258)

Styczeń 2021

Część I – Modele:

Osoba odpowiedzialna: Adam Steciuk

Przygotowanie danych:

Przed przystąpieniem do tworzenia projektu przekonwertowaliśmy otrzymane pliki danych z formatu .jsonl na .csv, który to dużo wygodniej można wczytywać przy użyciu pakietu Pandas.

Dalsze przygotowanie danych uczących odbywa się w *data_handler.py*.

1. Dane są wczytywane przy pomocy biblioteki Pandas
2. Odrzucane są kolumny z danymi, które nie będą używane w programie
 - a. W produktach jest to cena i oceny użytkowników
 - b. W sesjach jest to oferowana promocja, id zakupu oraz timestamp sesji
 - c. W użytkownikach jest to imię/nazwisko, miasto i ulica zamieszkania
3. Odrzucane są rekordy sesji z brakującym id produktu
4. Odtwarzane są brakujące id użytkowników w sesjach dla których istnieje sesja o takim samym id sesji i zawierająca id użytkownika
5. Zmieniany jest typ danych w kolumnach zawierających id użytkowników i produktów na int. (początkowo typem jest float, ponieważ wczytywane dane zawierają wartości NaN reprezentowane w bibliotece Pandas jako specjalna wartość float)
6. W tablicy sesji dodawana jest kolumna "event_strength" przypisująca aktywnościom użytkowników zdefiniowane w *config.py* wagi. Domyślnie zakup produktu ma wagę 2.0 a wyświetlenie produktu - wagę 1.0.
7. Z tablicy sesji tworzona jest tablica interakcji użytkowników z produktami poprzez odfiltrowanie i usunięcie ze zbioru danych użytkowników mających interakcje z mniejszą liczbą produktów niż zdefiniowano w *config.py*. Domyślnie 2. Ma to na celu zapobieganie wpływowi na wynik ewaluacji systemu użytkowników, dla których mamy za mało danych, aby zbudować właściwie funkcjonujący profil. Tak uzyskane interakcje dzielone są na zbiór testowy oraz uczący używając funkcji *train_test_split* z pakietu *sklearn* z parametrem *stratify=interactions['user_id']* zapewniającym odpowiedni rozkład interakcji każdego z użytkowników w obydwu zbiorach.
8. Interakcje są grupowane po użytkownikach i produktach a następnie wyznaczana jest suma wag wszystkich interakcji danego użytkownika z danym produktem wygładzona przez funkcję logarytmiczną.
9. Otrzymane dataframe'y są indeksowane po id użytkownika zapewniając szybsze wyszukiwanie.
10. Wyliczane są popularności przedmiotów grupując interakcje po id produktu i sumując wyznaczone wcześniej wagi interakcji.

Tak przygotowane dane są dalej używane do tworzenia modeli.

Model losowy:

Rekomenduje użytkownikom całkowicie losowe przedmioty. Powstał, aby można było porównać z nim 2 pozostałe modele.

Model popularnościowy:

Model jest niespersonalizowany i dla każdego użytkownika zwracana jest tablica posortowanych zgodnie z popularnością przedmiotów.

Model "content_based" (CB):

Model ten opiera się na tworzeniu profili przedmiotów na podstawie ich ścieżki kategorii oraz profili użytkowników na podstawie przedmiotów z jakimi mieli interakcje.

Oparty jest na obiekcie *TfidfVectorizer* z biblioteki *sklearn* i na podobieństwie cosinusowym pomiędzy wektorami reprezentującymi produkty i użytkowników. Matryca TFIDF jako termy przyjmuje kategorie ze ścieżki kategorii rozdzielone średnikami, zatem wektor reprezentujący przedmiot należący do kategorii "Gry i konsole; Gry na konsole; Gry Xbox 360" budowany jest na podstawie termów:

- *Gry i konsole*
- *Gry na konsole*
- *Gry Xbox 360*

Predykcją modelu są przedmioty posortowane zgodnie z malejącą relewancją dla danego użytkownika.

Ewaluator modeli:

Do ewaluacji zdecydowaliśmy się użyć metrykę dokładności Top-N. Algorytm ewaluacyjny:

- Dla każdego użytkownika
 - Dla każdego przedmioty z którym użytkownik miał interakcje w zbiorze testowym
 - Wybierz losowo próbkę w wielkości zdefiniowanej w *config.py* (domyślnie 100) z przedmiotów, z którymi użytkownik nie miał żadnej interakcji. Naiwnie zakładamy, że próbka ta będzie reprezentować przedmioty nieinteresujące dla użytkownika.
 - Poproś model o ocenę relewantności przedmiotów ze zbioru złożonego z tych 100 przedmiotów i tego jednego przedmiotu, z którym użytkownik miał interakcje i posortowanie jej zgodnie z malejącą relewantnością
 - Zlicz ile razy przedmioty z którymi użytkownik miał interakcje znajdują się w Top 5 oraz w Top 10 i podziel przez liczbę interakcji w zbiorze testowym.
- Zagreguj wyniki dla wszystkich użytkowników.

Ocena modeli:

```
Evaluated model: Popularity
994 users processed
{'recall@5': 0.5424758615275924, 'recall@10': 0.716272862862077}

Evaluated model: Content Based
994 users processed
{'recall@5': 0.04943873145458827, 'recall@10': 0.05636235183295392}

Evaluated model: Random
994 users processed
{'recall@5': 0.03654917968443363, 'recall@10': 0.07716461260695502}
```

Zauważyć można, że model popularnościowy ma bardzo wysoką skuteczność. Ponad 70% przedmiotów z którymi użytkownicy mieli styczność w zbiorze testowym pojawiło się wśród 10 najlepszych rekomendacji tego modelu, a ponad 50% z nich wśród 5 najlepszych rekomendacji. Świadczy to o tym, że między użytkownikami jest duża zgodność co do tego jakie przedmioty ich interesują.

Widzimy, że bardziej zaawansowany model bazujący na preferencjach użytkowników ma skuteczność porównywalną z modelem losowym. Długo zastanawialiśmy się dlaczego model CB zdaje się nie działać (choć poprawnie profiluje użytkowników, co pokażemy w dalszej części dokumentacji) i poprzez różnorakie eksperymenty ustaliliśmy, że składa się na to kilka czynników, które razem nie pozwalają na poprawną jego waluację poprzez zaimplementowane przez nas rozwiązania. Przede wszystkim problemem jest niepoprawny dobór implementacji ewaluatora do danych na podstawie których tworzone są profile przedmiotów. Wszystkie przedmioty należące do tej samej kategorii mają według modelu CB taką samą relewantność dla użytkownika, mimo że, te z którymi użytkownik nie miał interakcji traktowane są przez ewaluator jako nierelevantne. Kolejnym powodem jest przeuczenie się modelu na zbyt mało rozbudowanych (jak dla rozwiązania TFIDF) ścieżkach kategorii. Załóżmy sytuację, że użytkownik wyświetlił 4 gry komputerowe i 1 antenę RTV. W rekomendacjach (od najbardziej relewantnych do najmniej) pojawią się najpierw wszystkie gry komputerowe (tu jeszcze żadne nieprawidłowości się nie pojawiają), później wszystkie gry konsolowe, przez zgodność korzenia drzewa kategorii "Gry i konsole" a dopiero później anteny RTV. Gry konsolowe mają według modelu większą relewantność od anten, mimo, że użytkownik nie wyświetlił ani jednej. TFIDF nadawałoby się zatem dla dużo bardziej rozbudowanych drzew kategorii. W rzeczywistości stosowane jest zazwyczaj w systemach rekomendujących artykuły czy wpisy na portalach społecznościowych mogących zainteresować użytkownika.

Aby odtworzyć te wyniki - należy uruchomić skrypt *services/models/evaluate_test.py*.

Dowód działania modeli:

Dodatkowo dowodzić działania modelu popularnościowego nie trzeba. Wystarczą wyjątkowo dobre wyniki testu ewaluacyjnego.

| | product_id | event_strength |
|-----|------------|----------------|
| 0 | 1001 | 3164.767309 |
| 1 | 1318 | 3134.762644 |
| 2 | 1283 | 3123.615450 |
| 3 | 1067 | 2810.579342 |
| 4 | 1276 | 2792.007076 |
| .. | ... | ... |
| 212 | 1145 | 1100.050793 |
| 213 | 1131 | 1099.878617 |
| 214 | 1138 | 1098.859233 |
| 215 | 1059 | 1095.145645 |
| 216 | 1167 | 1091.854725 |

Aby odtworzyć te wyniki - należy uruchomić skrypt `services/models/popularity/popularity_test.py`.

Aby dowiedzieć działanie modelu CB dodałem do danych uczących nowego klienta oraz kilka sesji dla Niego:

| | |
|---|--|
| 1 | user_id,name,city,street |
| 2 | 101,Adam Steciuk,Miasto,Ulica |
| 3 | 102,Aurelia Malon,Police,pl. Brzoskwiniowa 11/53 |
| 4 | 103,Mateusz Kobel,Police,al. Wrocławska 10 |
| 5 | 104,Radosław Ratka,Mielec,pl. Nowa 89/04 |

| session_id | timestamp | user_id | product_id | event_type | offered_discount | purchase_id |
|------------|---------------------|---------|------------|--------------|------------------|-------------|
| 100000 | 2021-01-04T04:50:58 | 101 | 1036 | BUY_PRODUCT | 0 | |
| 100000 | 2021-01-04T04:50:58 | 101 | 1037 | BUY_PRODUCT | 0 | |
| 100000 | 2021-01-04T04:50:58 | 101 | 1038 | VIEW_PRODUCT | 0 | |
| 100000 | 2021-01-04T04:50:58 | 101 | 1039 | VIEW_PRODUCT | 0 | |
| 100000 | 2021-01-04T04:50:58 | 101 | 1315 | VIEW_PRODUCT | 0 | |
| 100000 | 2021-01-04T04:50:58 | 101 | 1316 | VIEW_PRODUCT | 0 | |
| 100000 | 2021-01-04T04:50:58 | 101 | 1317 | VIEW_PRODUCT | 0 | |
| 100000 | 2021-01-04T04:50:58 | 101 | 1318 | VIEW_PRODUCT | 0 | |
| 100001 | 2021-01-04T04:50:58 | 102 | 1276 | VIEW_PRODUCT | 0 | |

Sesje te odpowiadają kupieniu produktów:

| | |
|------|--|
| 1035 | Samsung Galaxy S III GT-i9300,Telefony i akcesoria;Telefony komórkowe,2199.9,1.8 |
| 1036 | LCD Asus VK228H,Komputery;Monitory;Monitor LCD,639,1.5333014113973924 |
| 1037 | LCD Asus VK248H,Komputery;Monitory;Monitor LCD,799,0.598612753711929 |
| 1038 | LCD Asus VK278Q,Komputery;Monitory;Monitor LCD,1117.01,2.5157044769482155 |
| 1039 | LCD Asus VS197D,Komputery;Monitory;Monitor LCD,269,3.1452921577229946 |

Oraz wyświetleniu produktów:

| | |
|------|---|
| 1037 | LCD Asus VK248H,Komputery;Monitory;Monitor LCD,799,0.598612753711929 |
| 1038 | LCD Asus VK278Q,Komputery;Monitory;Monitor LCD,1117.01,2.5157044769482155 |
| 1039 | LCD Asus VS197D,Komputery;Monitory;Monitor LCD,269,3.1452921577229946 |
| 1040 | Crysis 2 (PS3),Gry i konsole;Gry na konsole;Gry PlayStation3,79.9,0.635620429 |
| 1314 | Assassin's Creed (Xbox 360),Gry i konsole;Gry na konsole;Gry Xbox 360,49.99,3.59695214/53123/ |
| 1315 | Jabra Talk,Telefony i akcesoria;Akcesoria telefoniczne;Zestawy słuchawkowe,-54.99,2.336426677699466 |
| 1316 | Plantronics Voyager Legend,Telefony i akcesoria;Akcesoria telefoniczne;Zestawy słuchawkowe,249,4.1363825400661245 |
| 1317 | Plantronics Savi W740,Telefony i akcesoria;Akcesoria telefoniczne;Zestawy słuchawkowe,1303.96,0.5786970368704358 |
| 1318 | Plantronics Savi W710,Sprzęt RTV;Audio;Słuchawki,553,4.834082608435845 |

Wyświetlając profil tego użytkownika możemy zobaczyć, że model rzeczywiście poprawnie profiluje użytkownika:

Users profile:

(1, 28)

| | token | relevance |
|----|-----------------------------------|-----------|
| 0 | monitory | 0.512638 |
| 1 | monitory lcd | 0.512638 |
| 2 | komputery | 0.449592 |
| 3 | zestawy słuchawkowe | 0.318658 |
| 4 | akcesoria telefoniczne | 0.275843 |
| 5 | telefony i akcesoria | 0.259637 |
| 6 | audio | 0.109516 |
| 7 | słuchawki | 0.109516 |
| 8 | sprzęt rtv | 0.056918 |
| 9 | anten rtv | 0.000000 |
| 10 | biurowe urządzenia wielofunkcyjne | 0.000000 |
| 11 | drukarki i skanery | 0.000000 |
| 12 | gry i konsole | 0.000000 |
| 13 | gry komputerowe | 0.000000 |
| 14 | gry na konsole | 0.000000 |
| 15 | gry playstation3 | 0.000000 |
| 16 | gry xbox 360 | 0.000000 |
| 17 | odtwarzacze dvd | 0.000000 |
| 18 | odtwarzacze mp3 i mp4 | 0.000000 |
| 19 | okulary 3d | 0.000000 |
| 20 | przenośne audio i video | 0.000000 |
| 21 | tablety | 0.000000 |
| 22 | tablety i akcesoria | 0.000000 |
| 23 | telefony komórkowe | 0.000000 |
| 24 | telefony stacjonarne | 0.000000 |
| 25 | telewizory i akcesoria | 0.000000 |
| 26 | video | 0.000000 |
| 27 | zestawy głośnomówiące | 0.000000 |

Sprawdzając także rekomendacje modelu (weksportowane do pliku *CB_recs.csv*, żeby móc zobaczyć wszystkie kolumny rekomendacji, nie mieszczące się w oknie konsoli) zauważyć możemy, że model

rzeczywiście poleca produkty z odpowiednich kategorii:

| | A | B | C | D |
|----|------------|---------------------|-----------------------------|---|
| 1 | product_id | rec_strength | product_name | category_path |
| 2 | 1038 | 0.8530703880048081 | LCD Asus VK278Q | Komputery;Monitor;Monitor LCD |
| 3 | 1071 | 0.8530703880048081 | LCD Philips 241B4LPYCB | Komputery;Monitor;Monitor LCD |
| 4 | 1070 | 0.8530703880048081 | LCD Philips 241S4LCB | Komputery;Monitor;Monitor LCD |
| 5 | 1069 | 0.8530703880048081 | LCD NEC EA224WMI | Komputery;Monitor;Monitor LCD |
| 6 | 1068 | 0.8530703880048081 | LCD NEC EA223WMe | Komputery;Monitor;Monitor LCD |
| 7 | 1065 | 0.8530703880048081 | LCD Asus PA248Q | Komputery;Monitor;Monitor LCD |
| 8 | 1017 | 0.8530703880048081 | LCD Dell U2412M | Komputery;Monitor;Monitor LCD |
| 9 | 1039 | 0.8530703880048081 | LCD Asus VS197D | Komputery;Monitor;Monitor LCD |
| 10 | 1025 | 0.8530703880048081 | LCD BenQ GL2250 | Komputery;Monitor;Monitor LCD |
| 11 | 1032 | 0.8530703880048081 | LCD Iiyama E2280WSD | Komputery;Monitor;Monitor LCD |
| 12 | 1033 | 0.8530703880048081 | LCD Iiyama T1932MSC | Komputery;Monitor;Monitor LCD |
| 13 | 1034 | 0.8530703880048081 | LCD NEC EA223WM | Komputery;Monitor;Monitor LCD |
| 14 | 1064 | 0.8530703880048081 | LCD BenQ GL2250HM | Komputery;Monitor;Monitor LCD |
| 15 | 1036 | 0.8530703880048081 | LCD Asus VK228H | Komputery;Monitor;Monitor LCD |
| 16 | 1037 | 0.8530703880048081 | LCD Asus VK248H | Komputery;Monitor;Monitor LCD |
| 17 | 1030 | 0.8530703880048081 | LCD Iiyama B2280WSD | Komputery;Monitor;Monitor LCD |
| 18 | 1066 | 0.8530703880048081 | LCD Iiyama T1931SR | Komputery;Monitor;Monitor LCD |
| 19 | 1317 | 0.49501901155247935 | Plantronics Savi W740 | Telefony i akcesoria;Akcesoria telefoniczne;Zestawy słuchawkowe |
| 20 | 1316 | 0.49501901155247935 | Plantronics Voyager Legend | Telefony i akcesoria;Akcesoria telefoniczne;Zestawy słuchawkowe |
| 21 | 1315 | 0.49501901155247935 | Jabra Talk | Telefony i akcesoria;Akcesoria telefoniczne;Zestawy słuchawkowe |
| 22 | 1319 | 0.49501901155247935 | HTC HS-S200 | Telefony i akcesoria;Akcesoria telefoniczne;Zestawy słuchawkowe |
| 23 | 1074 | 0.2941522980910434 | Gembird BTCC-002 | Telefony i akcesoria;Akcesoria telefoniczne;Zestawy głośnomówiące |
| 24 | 1073 | 0.2941522980910434 | Jabra Drive | Telefony i akcesoria;Akcesoria telefoniczne;Zestawy głośnomówiące |
| 25 | 1201 | 0.2941522980910434 | Jabra Speak 410 | Telefony i akcesoria;Akcesoria telefoniczne;Zestawy głośnomówiące |
| 26 | 1222 | 0.2941522980910434 | Parrot Minikit NEO | Telefony i akcesoria;Akcesoria telefoniczne;Zestawy głośnomówiące |
| 27 | 1072 | 0.2941522980910434 | Jabra Freeway | Telefony i akcesoria;Akcesoria telefoniczne;Zestawy głośnomówiące |
| 28 | 1080 | 0.21316949876739275 | Kyocera FS-3140MFP | Komputery;Drukarki i skanery;Biurowe urządzenia wielofunkcyjne |
| 29 | 1002 | 0.21316949876739275 | Kyocera FS-1135MFP | Komputery;Drukarki i skanery;Biurowe urządzenia wielofunkcyjne |
| 30 | 1081 | 0.21316949876739275 | OKI B840dn | Komputery;Drukarki i skanery;Biurowe urządzenia wielofunkcyjne |
| 31 | 1003 | 0.21316949876739275 | Kyocera FS-3640MFP | Komputery;Drukarki i skanery;Biurowe urządzenia wielofunkcyjne |
| 32 | 1075 | 0.21316949876739275 | Ricoh SG3110DN | Komputery;Drukarki i skanery;Biurowe urządzenia wielofunkcyjne |
| 33 | 1077 | 0.21316949876739275 | Kyocera FS-C2026MFP | Komputery;Drukarki i skanery;Biurowe urządzenia wielofunkcyjne |
| 34 | 1078 | 0.21316949876739275 | Kyocera FS-3540MFP | Komputery;Drukarki i skanery;Biurowe urządzenia wielofunkcyjne |
| 35 | 1079 | 0.21316949876739275 | Kyocera FS-3040MFP | Komputery;Drukarki i skanery;Biurowe urządzenia wielofunkcyjne |
| 36 | 1076 | 0.21316949876739275 | Samsung CLX-6260FR ### Gadł | Komputery;Drukarki i skanery;Biurowe urządzenia wielofunkcyjne |
| 37 | 1277 | 0.1755607455318632 | Apple iPad mini 64GB 4G | Komputery;Tablety i akcesoria;Tablety |
| 38 | 1276 | 0.1755607455318632 | Apple iPad mini 64GB | Komputery;Tablety i akcesoria;Tablety |

Aby odtworzyć te wyniki - należy uruchomić skrypt `services/models/content_based/content_based_test.py`.

Dodatkowe uwagi:

Jeżeli poprosimy dowolny model o rekomendacje nie podając liczby rekomendacji otrzymamy dane diagnostyczne do testów, jeżeli poprosimy o konkretną liczbę rekomendacji (parametr n)- otrzymamy tylko tablice zawierające zadaną liczbę id polecanych produktów.

Budując modele nie korzystamy z niektórych informacji (na przykład z cen produktów). Z tego powodu zdecydowaliśmy nie usuwać ani nie modyfikować rekordów z wadliwymi, lub brakującymi informacjami nie używanymi w systemie.

Dla większej reprezentatywności rzeczywistości zbiór danych mógłby być dzielony na zbiór testowy i uczący na podstawie porządku czasowego występowania interakcji. Dla uproszczenia zrezygnowaliśmy z takiego rozwiązania, choć byliśmy świadomi jego potencjalnej informatywności dla modeli.

Prosząc Pana o udostępnienie wykazów ocen produktów mieliśmy na myśli tablicę podobną do tablicy sesji, gdzie zawarte byłyby oceny wystawione produktom przez każdego z użytkowników, a nie średnia ocen każdego produktu. Dlatego też zrezygnowaliśmy z użycia otrzymanych danych.

Część II – Serwis:

Osoba odpowiedzialna: Wojciech Moczydłowski

Opis serwisu:

W ramach drugiego etapu stworzyliśmy aplikację serwującą spersonalizowane rekomendacje dla użytkowników. Pozwala on na zrealizowanie eksperymentu A/B.

Z racji, że preparacja danych do uczenia modeli, jak i samo uczenie zajmuje dużo czasu, a od mikroserwisu wymaga się natychmiastowych odpowiedzi - postanowiliśmy stworzyć skrypt, generujący batche z 10. rekomendacjami dla każdego użytkownika i każdego modelu i zapisuje je odpowiednio do plików:

```
services/batches/content_base_batch.json
services/batches/popularity_batch.json
```

Wygenerować batche wymagane dla poprawnego działania mikroserwisu należy utworzyć uruchamiając skrypt: `services/batches/generate_batch.py`

Wysyłając zapytanie o rekomendacje do mikroserwisu, odczytywana jest tylko właściwa pozycja z przygotowanych batchy i wysyłana w odpowiedzi do użytkownika. Takie rozwiązanie pozwala na zrealizowanie sprecyzowanego w koncepcji wymagania нефunkcyjnego dotyczącego szybkości odpowiedzi systemu.

Uruchomienie mikroserwisu:

Zainstalowanie wymaganych pakietów:

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

Wygenerowanie wymaganych batchy:

```
python3 ./services/batches/generate_batch.py
```

Uruchomienie mikroserwisu:

```
python3 main.py
```

Użycie

Serwer wystawia jeden endpoint:

`http://localhost/models/user/:user_id`

Który zwraca obiekt:

```
{
  id,
  date,
  user_id,
  model,
  products
},
```

gdzie:

- `id` – id rekomendacji utworzony w celu identyfikacji w późniejszych testach A/B
- `date` – data utworzenia rekomendacji

- `user_id` – id użytkownika, dla którego została utworzona rekomendacja
- `model` – model użyty do stworzenia rekomendacji
- `products` – lista id produktów, które tworzą rekomendację

W ramach każdego zapytania serwer korzysta z jednego z modeli rekomendacyjnych. Decyzja, który model zostanie użyty zostaje podjęta na podstawie id użytkownika. W obecnej chwili, użytkownicy zostali podzieleni na dwie grupy: użytkownik z id większym od 550 otrzyma rekomendację modelu: "CONTENT_BASE", natomiast w przeciwny wypadku rekomendację modelu: "POPULARITY"

Logi

W ramach każdego zapytania zapisywane są logi w pliku `logs/recomendations.json`. Umożliwi to późniejsze przeprowadzenie eksperymentu A/B poprzez identyfikowanie rekomendacji.

Przykłady

Dla zapytania pod adres: **`http://localhost:5000/models/user/555`** otrzymamy:

```
{
  "id": "b52f3dcc-beda-4ee4-a546-ff6c1dab6b12",
  "date": "2021-01-15 14:48:09",
  "user_id": 555,
  "model": "CONTENT_BASE",
  "products": [
    1221,
    1197,
    1223,
    1224,
    1226,
    1220,
    1227,
    1228,
    1229,
    1230
  ]
}
```

Dla zapytania pod adres: **`http://localhost:5000/models/user/41`** otrzymamy:

```
{
  "id": "fb2ecddb-a5c3-4f32-8f2e-83c171686eaa",
  "date": "2021-01-15 00:32:19",
  "user_id": 41,
  "model": "POPULARITY",
  "products": [
    1001,
    1318,
    1283,
    1067,
    1276,
    1278,
    1234,
    1316,
    1073,
    1043
  ]
}
```


Dla zapytania pod adres: **http://localhost:5000/models/user/701** otrzymamy:

```
{
  "id": "943147b7-dcf5-421f-98e0-595fc78bf1e9",
  "date": "2021-01-15 14:48:47",
  "user_id": 701,
  "model": "CONTENT_BASE",
  "products": [
    1206,
    1200,
    1223,
    1224,
    1225,
    1226,
    1221,
    1227,
    1228,
    1198
  ]
}
```

Czego się nauczyliśmy:

- Implementacji systemów rekomendacyjnych
- Pracy ze zbiorami danych przy użyciu pakietu Pandas
- Przygotowywania danych uczących dla tworzonych modeli
- Praktycznego podejścia do porównywania jakości modeli (i “niestety” dociekania dlaczego implementowany sposób ich porównywania nie daje przewidywanych rezultatów)
- Dobierania modelu do posiadanych danych
- Dobrych praktyk tworzenia oprogramowania w języku Python
- Obsługi ramownika Flask umożliwiającego wystawienie mikro serwisu