

IUM - Opis implementacji

Sebastian Pietras, Maciej Kapuściński

1 Wstęp

W ramach projektu został zaimplementowany serwer webowy, obsługujący ruch sklepu i generujący odpowiednie wartości używając wyuczonych modeli oraz graficzny klient, który komunikuje się z serwerem.

Szczegółowe informacje dotyczące obsługi tych aplikacji znajdują się w odpowiednich plikach `README`.

2 Serwer

Wymagania:

- Python
- Flask
- Pandas
- Numpy

Serwer korzysta z dostarczonej bazy danych w postaci plików `.jsonl`. Dla ułatwienia do obsługi danych został użyty Pandas (w realnej aplikacji oczywiście lepiej użyć prawdziwej bazy danych).

Serwer używa wcześniej wyuczonych modeli oraz narzędzi używanych w trakcie tego procesu (uczenie zostało przeprowadzone w module `learning`, w szczególności w notatniku `learning/Models.ipynb`).

Komunikacja z serwerem odbywa się za pomocą żądań HTTP (metodą `GET`) do odpowiednich punktów końcowych:

- `users`

Parametry: brak

Pobiera listę użytkowników

- categories

Parametry: brak

Pobiera nowe id sesji oraz listę kategorii produktów

- products

Parametry: category

Pobiera listę produktów danej kategorii

- predict

Parametry: user, product, session

Pobiera przewidzianą zniżkę, która powinna zostać zaoferowana użytkownikowi

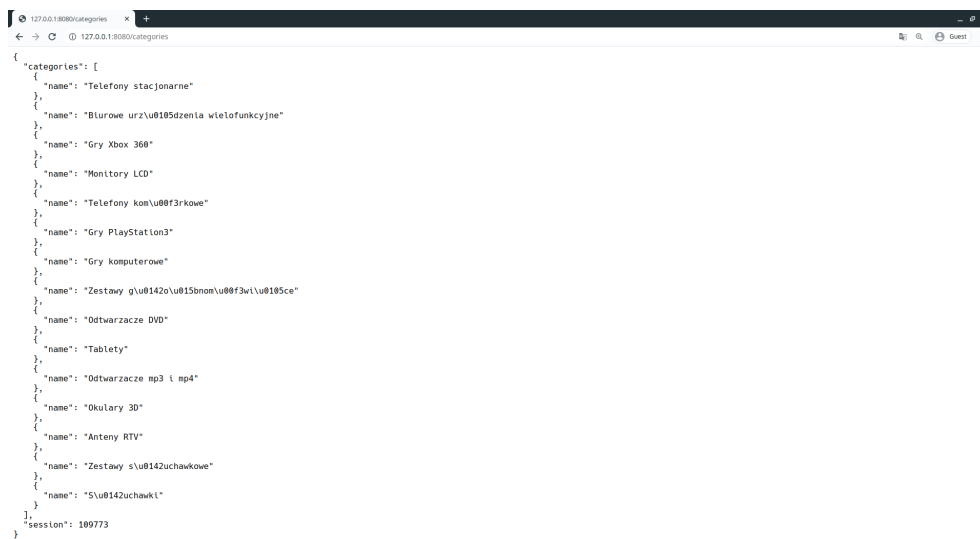
- buy

Parametry: user, product, session, discount

Rejestruje zdarzenie kupienia produktu

Serwer odpowiada danymi w formacie **json**.

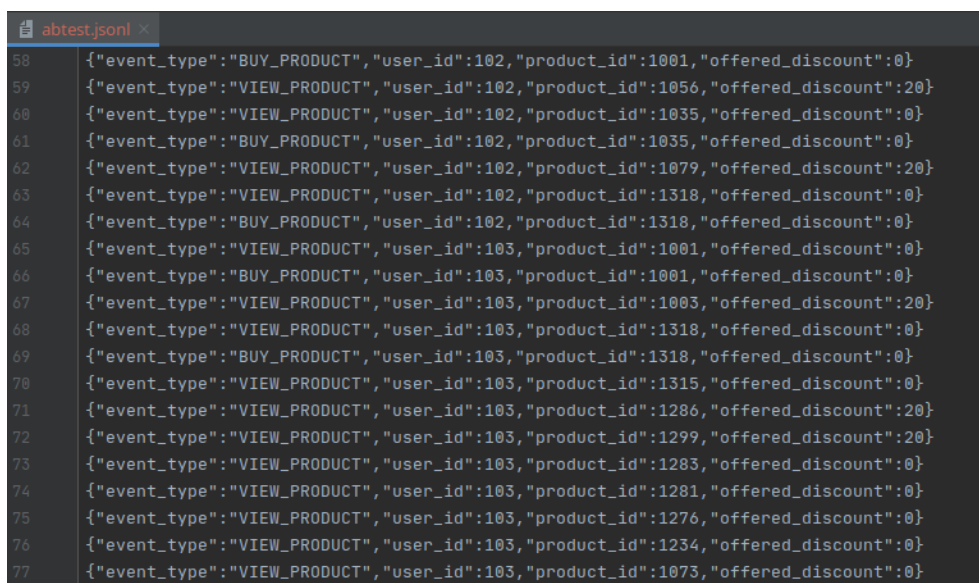
Rysunek 1: Przykładowa odpowiedź. Serwer dostępny z poziomu przeglądarki.



```
{
  "categories": [
    {
      "name": "Telefony stacjonarne"
    },
    {
      "name": "Blurowe urz\u0105dzenia wielofunkcyjne"
    },
    {
      "name": "Gry Xbox 360"
    },
    {
      "name": "Monitory LCD"
    },
    {
      "name": "Telefony kom\u00f3rkowe"
    },
    {
      "name": "Gry PlayStation3"
    },
    {
      "name": "Gry komputerowe"
    },
    {
      "name": "Zestawy g\u0142o\u015bnom\u00f3w\u0142\u0105ce"
    },
    {
      "name": "Odtwarzacze DVD"
    },
    {
      "name": "Tablety"
    },
    {
      "name": "Odtwarzacze mp3 i mp4"
    },
    {
      "name": "Okulary 3D"
    },
    {
      "name": "Anteny RTV"
    },
    {
      "name": "Zestawy s\u0142uchawkowe"
    },
    {
      "name": "S\u0142uchawki"
    }
  ],
  "session": 109773
}
```

Serwer może być uruchomiony w trybie przeprowadzania eksperymentu A/B. Wtedy ruch zostanie podzielony i różni użytkownicy będą korzystali z dwóch różnych modeli. Za przydział użytkownika do modelu odpowiedzialna będzie funkcja mieszająca. Serwer będzie zbierał i zapisywał dane dotyczące ruchu, które później posłużą do oceny modeli.

Rysunek 2: Przykładowe zebrane dane



```
58 {"event_type": "BUY_PRODUCT", "user_id": 102, "product_id": 1001, "offered_discount": 0}
59 {"event_type": "VIEW_PRODUCT", "user_id": 102, "product_id": 1056, "offered_discount": 20}
60 {"event_type": "VIEW_PRODUCT", "user_id": 102, "product_id": 1035, "offered_discount": 0}
61 {"event_type": "BUY_PRODUCT", "user_id": 102, "product_id": 1035, "offered_discount": 0}
62 {"event_type": "VIEW_PRODUCT", "user_id": 102, "product_id": 1079, "offered_discount": 20}
63 {"event_type": "VIEW_PRODUCT", "user_id": 102, "product_id": 1318, "offered_discount": 0}
64 {"event_type": "BUY_PRODUCT", "user_id": 102, "product_id": 1318, "offered_discount": 0}
65 {"event_type": "VIEW_PRODUCT", "user_id": 103, "product_id": 1001, "offered_discount": 0}
66 {"event_type": "BUY_PRODUCT", "user_id": 103, "product_id": 1001, "offered_discount": 0}
67 {"event_type": "VIEW_PRODUCT", "user_id": 103, "product_id": 1003, "offered_discount": 20}
68 {"event_type": "VIEW_PRODUCT", "user_id": 103, "product_id": 1318, "offered_discount": 0}
69 {"event_type": "BUY_PRODUCT", "user_id": 103, "product_id": 1318, "offered_discount": 0}
70 {"event_type": "VIEW_PRODUCT", "user_id": 103, "product_id": 1315, "offered_discount": 0}
71 {"event_type": "VIEW_PRODUCT", "user_id": 103, "product_id": 1286, "offered_discount": 20}
72 {"event_type": "VIEW_PRODUCT", "user_id": 103, "product_id": 1299, "offered_discount": 20}
73 {"event_type": "VIEW_PRODUCT", "user_id": 103, "product_id": 1283, "offered_discount": 0}
74 {"event_type": "VIEW_PRODUCT", "user_id": 103, "product_id": 1281, "offered_discount": 0}
75 {"event_type": "VIEW_PRODUCT", "user_id": 103, "product_id": 1276, "offered_discount": 0}
76 {"event_type": "VIEW_PRODUCT", "user_id": 103, "product_id": 1234, "offered_discount": 0}
77 {"event_type": "VIEW_PRODUCT", "user_id": 103, "product_id": 1073, "offered_discount": 0}
```

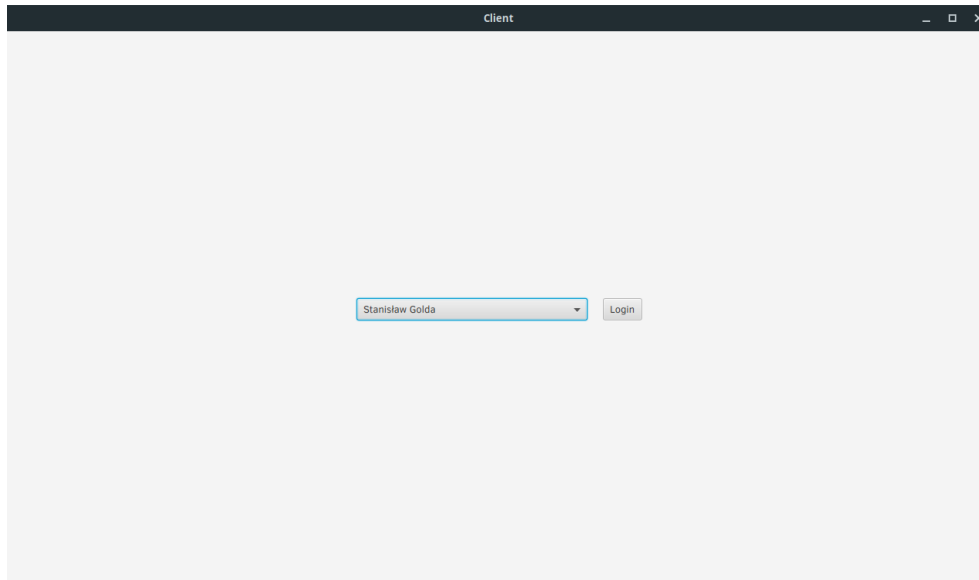
3 Klient

Wymagania:

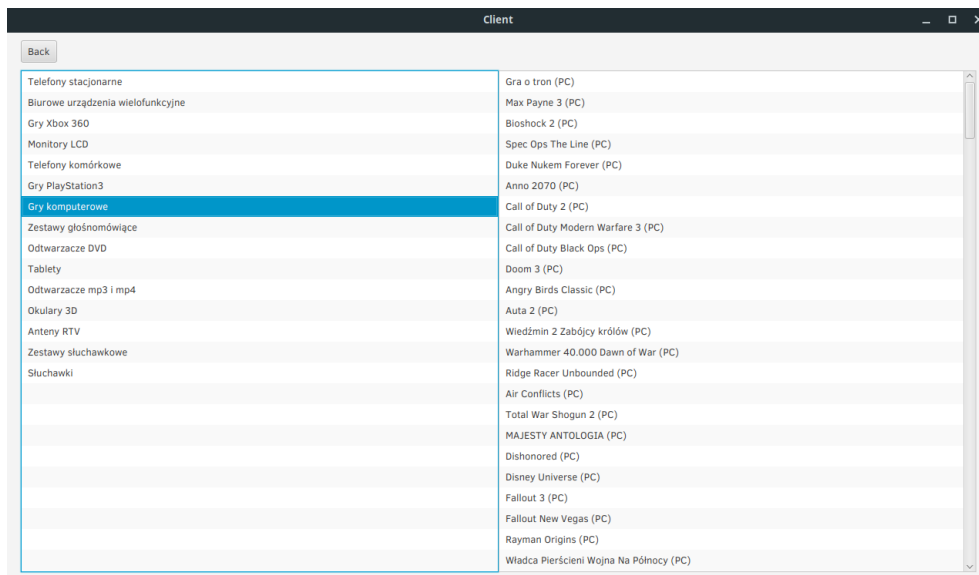
- JDK 11
- Gradle 4.10.3 (bez instalacji może zostać użyty Wrapper)

Klient pobiera listę użytkowników z serwera. Można wybrać użytkownika z listy i zalogować się, pobierając listę kategorii. Można rozwinąć każdą kategorię, pobierając listę produktów. Po kliknięciu w produkt zostaje pobrana i wyświetlona zaoferowana cena. Użytkownik może kupić produkt lub wrócić do poprzedniego widoku.

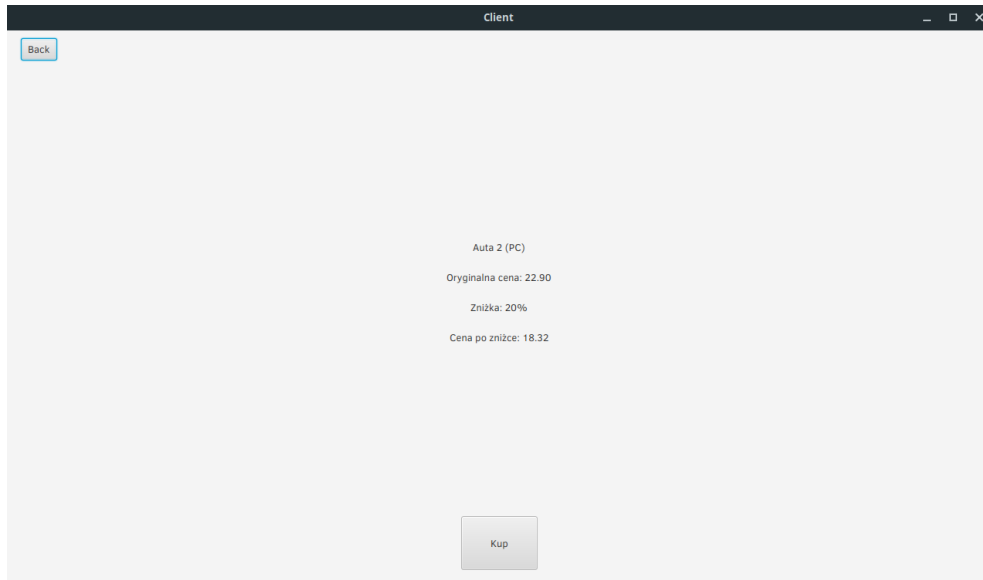
Rysunek 3: Ekran logowania



Rysunek 4: Lista produktów



Rysunek 5: Szczegóły produktu (możliwość kupna)



4 Eksperyment A/B

Do przeprowadzenia eksperymentu A/B potrzebny jest ruch prawdziwych użytkowników. Tylko oni są w stanie wygenerować zdarzenia zgodne z ich zachowaniem.

Po zakończonym eksperymentcie A/B możliwa jest analiza wyników z poziomu notatnika Jupyter, który można znaleźć w `server/ABTest.ipynb`.