

Budowanie systemu rekomendacji (cz. 2)



Profilowanie klienta na podstawie logów własnego sklepu internetowego

Aby zbudować system rekomendacji musimy mieć bazę danych klientów. Każdy klient powinien być opisany w sposób automatyczny na podstawie logów aplikacyjnych. Niezbędne jest więc uzyskanie do nich dostępu.



W logach aplikacyjnych oraz innych logach można znaleźć wszelkie hasła, które były wpisywane przez klienta do wyszukiwarki sklepu, można znaleźć informacje co kupił, jak długo zajęło mu podejmowanie decyzji, jak reagował na banery reklamowe oraz informacje, czy zrezygnował ze swoich zakupów na pewnym etapie procesu. Logi aplikacyjne to proste informacje tekstowe, które bardzo łatwo odczytać bez konieczności użycia narzędzi. Niestety logów jest za dużo, żeby czytać i wyciągać wnioski. W sklepie może jednocześnie być nawet 50 osób dokonujących różnych transakcji i czynności. Aby skutecznie wyciągać informacji z logów aplikacyjnych trzeba te informacje obrabiać za pomocą prostych narzędzi języka Python.

Skąd wziąć logi?

Większość stron internetowych zbudowanych jest na bazie języka HTML na szablonach WordPress. Budowanie takiej strony jest dziecinnie łatwe. Po zbudowaniu strony można zainstalować wtyczkę sklepu internetowego. Wtyczki na stronach WordPress działają podobnie jak wtyczki instalowane na telefonach komórkowych. To mniej więcej podobna koncepcja. Budowanie na bazie takiej wtyczki sklepu internetowego z np. tortami to już bardziej złożone zajęcie. Zarówno strona internetowa, jak i sklep muszą być gdzieś opublikowane w internecie. Należy więc wykupić miejsce w chmurze. Nazywa się to hostingiem. Kiedy już mamy hostowaną stronę internetową, na której prowadzimy sklep internetowy z wyrobami cukierniczymi możemy zacząć szukać logów.

Jako właściciel sklepu opartego na WordPressie masz dostęp do wielu różnych logów systemowych, aplikacyjnych i serwerowych, ale gdzie i jak je zdobyć zależy od

kilku rzeczy: hostingu, konfiguracji WordPressa oraz dodatkowych wtyczek.

Logi serwera – logi systemowe (np. logowania, błędy, ruch) znajdują się w panelu hostingowym (np. cPanel, DirectAdmin, Plesk, itp. zależy od dostawcy hostingu). Możemy tam napotkać następujące pliki:

- access.log – lista wszystkich zapytań HTTP (kto wchodził, z jakiego IP, kiedy, jaki URL),
- error.log – błędy serwera (np. błędy PHP, brakujące pliki, błędy 404),
- auth.log (czasami) – logi prób logowania do FTP, SSH, itp.

Aby je znaleźć należy zalogować się do panelu hostingowego i szukać zakładki: „Logi”, „Statystyki”, „File Manager” > /logs/. Ewentualnie można pobrać te listy logów przez FTP (np. katalog logs/, logs/error_log, itp.). Zwykle pliki logów systemowych mają zakończenie .log lub .txt, czasem w formacie Apache lub Nginx log format (czytelny tekst). Logi dają się odczytać w zwykłym edytorze txt.

Naszym celem jest zbadanie jak zachowuje się klient w sklepie. Najważniejsze do tego są logi aplikacyjne, a nie systemowe, o których wspomniałem powyżej. WordPress domyślnie nie prowadzi logów zdarzeń użytkownika (np. kto się logował, kto co przeglądał), ale możesz je włączyć lub dodać wtyczki. Należy zainstalować jedną z podanych:

- WP Activity Log – pełne logi co robią użytkownicy,
- Simple History – pokazuje logowania, edycje, aktywność adminów,
- User Activity Log – lista kto się logował i co robił.

Logi są wtedy zapisywane w panelu WordPressa, czasem także jako pliki JSON lub bazy danych.

Jeżeli ktoś prowadzi sklep internetowy na stronie typu WordPress prawdopodobnie korzysta z WooCommerce, czyli modułu sklepu WordPress

Logi aplikacyjne można wtedy znaleźć:

WordPress Admin > WooCommerce > Status > Zakładka Logi

Żeby obserwować klientów, zbierać o nich informacje, tworzyć bazę danych, trzeba po prostu dostać się do logów. Sklepy można prowadzić na gotowych modułach



sklepów internetowych, są też wielkie witryny internetowe typu Allegro, gdzie można wykupić sobie miejsce. Nie potrafię podać wszystkich możliwości w jaki sposób uzyskać logi aplikacyjne. Mogę tylko powiedzieć, że jest to niezbędne do stworzenia systemu rekomendacji, który będzie stymulował naszą sprzedaż. Jeżeli dostawca usługi sklepu internetowego nie zapewnia dostępu do logów aplikacyjnych można przejść do innego sklepu. Przecież w grę wchodzi rozwój, a nie lojalność w stosunku do dostawcy usług. Serwis techniczny hostingu albo sklepu internetowego powinien pomóc w uzyskaniu dostępu do logów aplikacyjnych. Jest to bardzo ważne. To lekcja, którą trzeba odrobić aby pójść dalej.

Rodzaje logów i dostępne informacje

Typowa witryna e-commerce generuje różne logi serwerowe, które możemy wykorzystać do analizy zachowań klientów. Podstawowym źródłem są logi dostępu WWW (np. Apache/Nginx), rejestrujące każde żądanie HTTP. Każdy wpis logu zawiera m.in. adres IP odwiedzającego, znacznik czasu, metodę i adres żądania (URL wraz z ewentualnymi parametrami zapytania), kod odpowiedzi oraz identyfikator klienta przeglądarki (User-Agent). Przykładowy wpis loga dostępu może wyglądać następująco:

plaintext

CopyEdit

```
81.174.152.222 - - [30/Jun/2020:23:38:03 +0000] "GET /search?query=ciasto+bez+glutenu HTTP/1.1" 200 6763 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64) Firefox/77.0"
```

Powyższy wpis (tu zaadaptowany do zapytania o ciasto bez glutenu – zaznaczone na żółto) pokazuje kluczowe elementy: adres IP, datę/godzinę, metodę GET i ścieżkę żądania (w tym przypadku endpoint/search z parametrem query=ciasto+bez+glutenu), kod odpowiedzi 200 oraz ciąg user-agent przeglądarki.

Oprócz logów serwera WWW, są wspomniane wyżej logi aplikacyjne opisujące akcje użytkownika w sklepie. Jeśli aplikacja webowa loguje zdarzenia (np. w osobnym pliku lub bazie), możemy uzyskać bardziej szczegółowe wpisy akcji użytkowników, np. wyszukiwanie produktu, wyświetlenie konkretnej kategorii, dodanie produktu do koszyka, porzucenie koszyka czy finalizację zakupu. Tego typu logi akcji użytkownika często zawierają identyfikator użytkownika (ponieważ klient musi się zalogować), znacznik czasu oraz nazwę i parametry akcji. Przykładowo, wpisy w logu akcji mogą wyglądać tak:

plaintext

CopyEdit

```
2025-07-26 10:15:32 INFO user_id=123 ACTION=SEARCH query="ciasto bez glutenu"
2025-07-26 10:16:05 INFO user_id=123 ACTION=VIEW CATEGORY name="Torty bezglutenowe"
```

```
2025-07-26 10:17:10 INFO user_id=123 ACTION=ADD_TO_CART product_id=456 name="Tort bezglutenowy czekoladowy"
```

```
2025-07-26 10:18:45 INFO user_id=123 ACTION=PURCHASE order_id=789 value=120.00 PLN
```

Taki log aplikacyjny pokazuje ścieżkę zachowań konkretnego klienta user_id=123: wyszukał frazę *ciasto bez glutenu*, obejrzał kategorię *Torty bezglutenowe*, dodał do koszyka produkt o ID 456 i ostatecznie dokonał zakupu. W literaturze takie dane nazywa się często danymi click-stream lub logami śledzenia użytkownika, gdzie każdy wpis reprezentuje interakcję użytkownika z elementami sklepu (kliknięcie, dodanie do koszyka, polubienie produktu, zakup itp.) wraz z identyfikatorem użytkownika, przedmiotem akcji (np. ID produktu) i czasem. Dzięki temu, że klienci nie są anonimowi (logują się w sklepie jako użytkownicy), możemy powiązać wszystkie te zdarzenia z konkretną osobą i budować jej profil behawioralny na podstawie historii działań. Pamiętajmy, że naszym celem jest zbudowanie bazy danych o każdym kliencie sklepu.

Wydobywanie danych z logów przy użyciu Pythona

Mając dostęp do logów, możemy za pomocą Pythona zautomatyzować ekstrakcję potrzebnych informacji. Logi to pliki tekstowe, które należy przekształcić w uporządkowaną strukturę danych (tabelę DataFrame). Istnieją biblioteki ułatwiające to zadanie – przykładowo lars (Library for Apache/Nginx log processing) pozwala łatwo parsować logi WWW i wczytać je do Pythona. Podobnie biblioteka apache-log-parser. Najlepiej i najprościej użyć łatwych i skutecznych funkcji Pandas. Python umożliwia wczytanie logu linia po linii, rozbicie każdej linii na pola (np. metodą split lub re.findall według formatu logu) i złożenie listy słowników reprezentujących wpisy. Następnie łatwo konwertujemy to na DataFrame Pandas.

Tak utworzona tabela DataFrame będzie miała kolumny odpowiadające polom logu (zapytanie, czas, cena itp.) i każdy wiersz odpowiada jednemu klientowi. Dzięki temu możemy filtrować i agregować dane. Nie jestem w stanie opisać krok po kroku jak to się robi. Każdy kto zna Pandas i umie agregować dane i tworzyć ekstrakcję z długich ciągów znaków poradzi sobie z tym z łatwością.

W naszym kontekście najważniejsze jest, by po wczytaniu logów powiązać zdarzenia z konkretnymi klientami. Identyfikujemy klienta po ID, każdy klient się loguje do sklepu, dobrym rozwiązaniem jest logować akcje użytkownika według wzmianki w logach user_id. Dzięki temu w DataFrame możemy mieć kolumnę user_id i łatwo grupować dane po użytkowniku.

Kluczowe metadane, które warto wydobyć z logów:

- Frazy wyszukiwania – na podstawie parametrów zapytań (np. query= w URL) z logów HTTP lub pola query w logach akcji.

- Identyfikatory i nazwy przeglądanych produktów/kategorii – z adresów URL odwiedzanych stron (np. /category/dla-dzieci/ w ścieżce URL) lub z logów akcji (VIEW_CATEGORY, VIEW_PRODUCT).

- Akcje koszyka i zakupów – wpisy dodania do koszyka (ADD_TO_CART), usunięcia z koszyka, przejścia do kasy oraz finalizacji transakcji (PURCHASE), wraz z znacznikami czasu.

- Interakcje z promocjami – np. kliknięcia w banery promocyjne (jeśli logowane), użycie kodów rabatowych (może pojawić się w logu zakupu), wejścia na stronę „promocje”.

- Czasowe aspekty wizyt – daty i godziny akcji pozwalające wyznaczyć, ile czasu klient spędza na stronie, o jakich porach jest aktywny, jak długo trwa od pierwszego wejścia do zakupu itp.

Analiza zachowań i mapowanie cech klienta z logów



Gdy dane z logów są już zebrane w tabelarycznej formie, kolejnym krokiem jest wydobywanie przydatnych informacji biznesowych, czyli zrozumienie, co mówią nam te dane o preferencjach i profilu klienta. Należy zdefiniować pewne cechy (atrybuty) klientów, które chcemy rozpoznać.

Na podstawie zachowań widocznych w logach możemy automatycznie profilować klientów według takich cech. Oto przykładowe cechy i sposoby ich wyznaczania.

- **Preferencje dietetyczne** (np. dieta bezglutenowa, wegańska) – jeśli w logach wyszukiwania lub przeglądania widać częste zapytania w stylu „bez glutenu”, „wege”, „keto” itp., można odnotować, że klient ma szczególne wymagania dietetyczne. Przykładowo klient, który wielokrotnie szuka tortów niskoglutenowych lub bezcukrowych, otrzyma flagę Dietetyczne=TAK w swoim profilu (kolumna w DataFrame). Z czasem możemy nawet rozróżniać typ diety (np. bezglutenowiec, weganin itp.), jeśli takie kategorie pojawiają się w danych.

- **Posiadanie małych dzieci** – tę cechę wydedukujemy, analizując produkty i kategorie oglądane lub kupowane przez klienta. Jeśli logi pokazują, że użytkownik często odwiedza sekcję dla dzieci (np. torty z postaciami z bajek, świeczki w kształcie cyfr, produkty dla niemowląt) lub wyszukuje frazy typu „dla dziecka”, to prawdopodobnie jest rodzicem małych dzieci. Zakupy typu pieluchy, zabawki, torty na 1. urodziny itp. również silnie wskazują na tę grupę. Wówczas w profilu klienta zaznaczamy np. Małe_dzieci=TAK. Im więcej sygnałów (np. regularne kupowanie tortów z motywem dziecięcym), tym pewniejsza kategoryzacja.

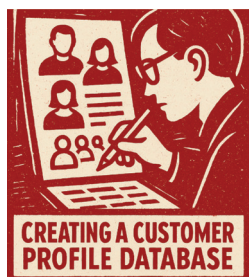
- **Szacowany wiek klienta** – bez bezpośrednich danych demograficznych musimy szacować wiek

pośrednio. Logi sprzedaży mogą tu pomóc: rodzaj kupowanych produktów i okazji może sugerować przedział wiekowy. Np. klient zamawiający torty na 18-te urodziny (napis „18 lat”), na rocznicę 50-lecia lub produkty retro vs. bardzo nowoczesne smaki – takie informacje pozwalają zgadywać. Jeśli klient kupuje głównie artykuły dziecięce, można założyć, że jest w wieku np. 25-40 (typowo rodzice małych dzieci). Z kolei częste zakupy produktów prozdrowotnych, ortopedycznych mogą sugerować wyższy przedział wiekowy. Choć to mniej precyzyjne, wzorce zakupowe i okoliczności (np. kupowanie tortów na „30 urodziny”, „50 urodziny”) pozwalają w przybliżeniu oszacować wiek lub etap życia. Tę informację można zapisać np. jako Wiek_szacowany=30-40. (Należy pamiętać, że to zgadywanie na podstawie danych – przydatne w marketingowej segmentacji, choć nie 100% dokładne.)

- **Wrażliwość na promocje** – logi mogą ujawnić, jak bardzo dany klient reaguje na promocje i okazje. Możemy na przykład śledzić: czy użytkownik klika w linki promocyjne z newslettera (pojawi się odpowiedni referrer lub unikalny URL kampanii w logu), czy często zagląda na stronę „Promocje/Wyprzedaże”, czy podczas zakupów używa kodów rabatowych. Jeśli logi akcji mają informację o zastosowanym kuponie rabatowym przy zamówieniu, to wyraźny sygnał, że klient poluje na zniżki. Można także obserwować częstotliwość zakupów podczas sezonowych wyprzedaży (np. klient kupuje tylko w czasie Black Friday albo gdy coś jest przecenione). Gdy stwierdzimy takie zachowania, profil klienta wzbogacamy o cechę typu Promocjolubny=TAK (czyli skłonny do korzystania z promocji).

- **Szybkość podejmowania decyzji/długość procesu zakupowego** – z danych czasowych w logach da się wyczytać czy klient kupuje impulsywnie, czy długo się zastanawia. Można zmierzyć np. średni czas od pierwszego kliknięcia w produkt do jego zakupu albo liczbę wizyt zanim sfinalizuje transakcję. Jeśli z logów wynika, że użytkownik często dodaje produkt do koszyka, po czym dopiero po kilku dniach finalizuje zamówienie (albo porzuca koszyk), oznacza to dłuższy proces decyzyjny. W DataFrame można mieć kolumnę Decision_Time, gdzie zapiszemy np. „krótki” lub „długi” (na podstawie progu, np. >24h uznajemy za długie zastanawianie się). Klient, który wielokrotnie przegląda ten sam produkt przed zakupem lub długo trzyma produkty w koszyku, ewidentnie potrzebuje więcej czasu – co może być cenną informacją dla działu marketingu (np. wysyłanie przypomnień o porzuconym koszyku).

- **Okazje zakupowe (np. urodziny, święta)** – analizując treści wpisywane przez klienta (np. dedykacje na torcie) lub momenty zakupów, możemy odkryć wzorce okazjonalne. Wspomniany przykład z napisem „100 lat” sugeruje tort urodzinowy – jeśli zauważamy, że dany klient regularnie zamawia torty z typowo urodzinowymi napisami („Sto lat”, imię solenizanta itp.), to można założyć, że kupuje głównie na urodziny bliskich. Taki wniosek



zapiszemy np. jako cechę `Oka-
zja_zakupu=urodziny`. Inny klient
może zawsze kupować duże ilo-
ści czekoladek w połowie lutego
– zapewne na Walentynki – co
zanotujemy jako inny profil za-
chowania. Logi akcji mogą do-
starczyć takich detali, jeśli np.
w formularzu zamówienia klienci

wpisują życzenia (i te dane są logowane lub dostępne w danych transakcyjnych).

W praktyce, tworząc profil klienta, zliczamy lub flagujemy takie zdarzenia w kolumnach. Używając Pythona i Pandas, możemy grupować w DataFrame wpisy po `user_id` i agregować informacje, np.:

- ile razy użytkownik szukał fraz z określonej kategorii (dietetyczne, dla dzieci itp.),
- jakie kategorie produktów najczęściej ogląda (np. zbudować ranking kategorii dla użytkownika),
- średni czas od pierwszego wejścia do zakupu,
- odsetek transakcji z użyciem kodu rabatowego,
- maksymalna wartość koszyka, częstotliwość zakupów (miesięcznie, rocznie),
- dni/godziny największej aktywności itp.

Te wszystkie dane doklejamy do tabeli profili klientów. W efekcie otrzymujemy np. DataFrame, gdzie każdy wiersz to jeden klient, a kolumny to cechy/profil:

UserID	Dietetyk*	Małe dzieci*	Szac. wiek†	Promocjolutubny‡	Decyzje§	Okazje zakupu¶	Najcz. kategoria§§
101	1	1	3	2	2	1	4
102	0	0	2	0	0	3	1
103	1	0	4	1	1	2	3
104	0	1	3	2	2	1	2
105	1	1	1	0	0	5	4
106	0	0	5	1	1	4	2
107	0	1	2	0	0	3	1

Legenda do tabeli:
Dietetyk, Małe dzieci: 1 = TAK 0 = NIE
Szac. wiek (grupy): 1 = < 18 lat | 2 = 18-25 | 3 = 26-35 | 4 = 36-50 | 5 = > 50
Promocjolutubny: 0 = nie używa rabatów | 1 = umiarkowanie | 2 = często
Decyzje (czas namysłu): 0 = krótkie | 1 = średnie | 2 = długie
Okazje zakupu: 1 = urodziny | 2 = święta | 3 = walentynki | 4 = rocznice | 5 = inne
Najcz. kategoria (kod przykładowy): 1 = torty czekoladowe | 2 = torty owocowe | 3 = ciasta sernik | 4 = wypieki bezglutenowe

Taki profil pozwala nam segmentować klientów, można dokonać segmentacji grupując klientów o podobnych profilach w klastry (np. „młodzi łowcy okazji”, „rodzice na diecie”, „lojalni tradycjonałści”, itp.). W literaturze podkreśla się, że odpowiednio przeanalizowane dane z logów serwera mogą posłużyć do budowy profilu użytkownika i personalizacji oferty. Firmy wykorzystują taką segmentację, by lepiej targetować marketing – np. kierować promocje rodzinne do tych, którzy mają dzieci, albo reklamować produkty bezglutenowe klientom z flagą dietetyczną. To właśnie istota segmentacji

klientów: identyfikacja różnych grup i dostosowanie do nich działań marketingowych. Trzeba jednak dodać, że segmentacja nie jest naszym celem. Celem jest utworzenie bazy danych klientów na podstawie logów. Poza tym segmentacja klientów jest nam niepotrzebna o czym przekonamy się w kolejnym artykule.

Podsumowanie

Aby skutecznie profilować klientów bez gotowych narzędzi analitycznych (jak Google Analytics), należy skupić się na maksymalnym wykorzystaniu logów z własnego sklepu. Należy wykorzystać skrypty Pythonowe do automatycznego przetwarzania logów, najlepiej zrobić to w bibliotece Pandas. Następnie należy zdefiniować zestaw cech klientów istotnych z punktu widzenia biznesu. Wyniki należy zapisać w bazie danych klientów najlepiej w formacie DataFrame tak, aby każdy kolejny log nowej aktywności mógł zaktualizować profil (proces ten można zautomatyzować).

Dane z logów z czasem pozwolą personalizować ofertę i komunikację tak samo skutecznie, jak gotowe platformy analityczne. Tabela analityczna cech klientów sklepu będzie podstawą do zastosowania potężnego narzędzia rekomendacyjnego jakim jest Factorization Machines. Jak użyć tego narzędzia dowiemy się w następnym artykule.

Wojciech Moszczyński

Wojciech Moszczyński – absolwent Katedry Ekonometrii i Statystyki Uniwersytetu Mikołaja Kopernika w Toruniu, specjalista z zakresu ekonometrii, finansów, *data science* oraz rachunkowości zarządczej. Specjalizuje się w optymalizacji procesów produkcyjnych i logistycznych. Prowadzi badania w obszarze rozwoju i zastosowania sztucznej inteligencji. Od lat zaangażowany w popularyzację *machine learning* oraz *data science* w środowiskach biznesowych.