Projekt systemu rekomendacji produktów w firmie E-Beauty

Consult It 2023

Hubert Perliński
Michał Krzyżański

Abstrakt **2**
Wstęp do projektu **3**
Źródła danych **4**
Hurtownia danych 4
Konstrukcja hurtowni 5
Proponowany podział danych 6
Implementacja systemu danych 10
Hurtownia lokalna vs w chmurze 10
Rozwiązanie komercyjne i koszt 11
Logistyka wdrożenia 11
Modele preferencji **12**
Algorytm Apriori 12
Collaborative filtering 14
Segmentacja klientów K-means 14
Porównywanie klientów K-Nearest Neighbours 15
Content based filtering używając Random Forest 16
Model rekomendacji **17**
Przykłady zastosowań algorytmów 18
Rekomendacje na stronie głównej 18
Rekomendacje na stronie produktu 19
Rekomendacje przy zakupie 20
Mailing 21
Rozwiązania problemu “cold start” **21**
Nowy klient 21
Nowy produkt 22
Implementacja **22**
Przyszłe rekomendacje **23**
Usprawnienie modelu mailowego 23
Rekomendacje aktualizowane w czasie rzeczywistym 23
Przejście z chmury na on-premise 23
Podsumowanie **23**
Słownik **24**
Źródła **25**
Dodatki **26**
Zmienne w tablicach danych 26

Abstrakt

Raport zawiera propozycję procesu implementacji inteligentnego systemu rekomendacji produktów dla firmy E-Beauty.

Pierwszym wskazanym krokiem jest konstrukcja infrastruktury danych typu data warehouse. Rozwiązanie to posłuży jako podstawa dla systemu rekomendacji, jak i przyszłych projektów analitycznych.

Kolejnym jest tworzenie i implementacja modeli przewidywania preferencji klientów typów Apriori, Collaborative filtering i Content based filtering. Wyniki tych modeli są używane jako podstawa do hybrydowego silnika rekomendacji, który generalizuje priorytetyzacja produktów, osiągając lepsze wyniki niż pojedynczy model.

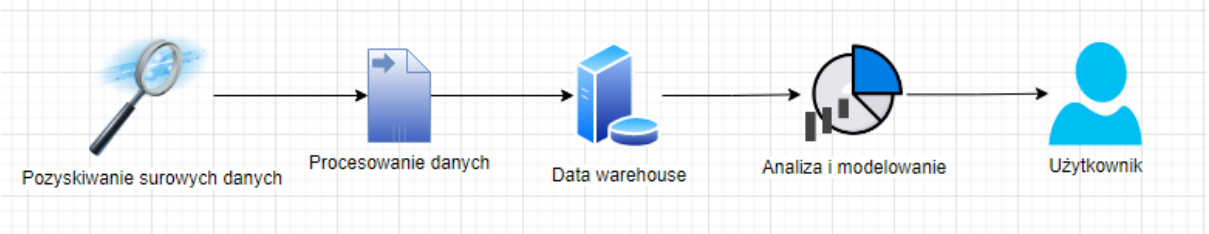
Dokument zawiera również ocenę kosztów finansowych i czasowych wprowadzenia projektu.

Wstęp do projektu

System rekomendacji w sklepie internetowym umożliwia sprzedawcom zwiększenie sprzedaży, cross-selling, podnosi zaangażowanie konsumenta oraz średni czas pobytu na stronie. Jest to jedno z najbardziej skutecznych i rozpowszechnionych zastosowań uczenia maszynowego w e-commerce. Prawidłowo skonfigurowane i zaimplementowane umożliwia wzrost sprzedaży, zaangażowania i zaufania klientów. Według McKinsey aż 35% sprzedaży na serwisie Amazon jest bezpośrednio wynikiem systemu rekomendacji[1].

System rekomendacji działa w pełni automatycznie i umożliwia precyzyjne śledzenie sentymentu klientów i zadowolenia z produktów. Pełen wgląd w statystyki i efektywność systemu rekomendacji zapewnia przejrzysty dashboard.

Implementacja takiego systemu wiąże się jednak z kosztami. Potrzebne jest stworzenie data warehouse: systemu zbierania, przechowywania i łączenia danych, które potem można użyć do stworzenia i konfiguracji systemu automatycznej rekomendacji.



Do wytrenowania modelu potrzebne są dane historyczne, dlatego stworzenie tej infrastruktury powinno być priorytetem. Następnie można przejść do modelowania preferencji klientów. Cały ten proces może zająć nawet 5 miesięcy zanim system zacznie być w pełni operacyjny.

Wraz ze wzrostem branży e-commerce coraz ważniejsze staje się przewidzenie zachowań i preferencji klienta w coraz bardziej chaotycznym środowisku informacyjnym. W związku z tym pojedynczy model rekomendacji nie jest w stanie efektywnie sprostać ewaluacji potrzeb. Potrzebne jest wzięcie pod uwagę różnych metod i podejść do rozwiązania problemu. Nasza propozycja opiera się na innowacyjnym schemacie modelu hybrydowego, który posiada stosunkowo łagodny cold start, jest efektywny w początkowej fazie zastosowań jak i posiada duże możliwości skalowania. Składałby się on z kilku modeli preferencji, których wpływ na finalną decyzję byłby dynamicznie determinowany na podstawie decyzji konsumentów.

[1] (*How retailers can keep up with consumers* | *McKinsey*, 01 October 2013)

Źródła danych

Ta część raportu opisuje proces pozyskiwania i przetwarzania danych dla systemu rekomendacji produktów i również innych zastosowań analitycznych. Nasz zespół zaleca wdrożenie standardowej Centralnej Hurtowni Danych oraz utworzenie nowego etatu zajmującego się jej utrzymaniem i rozwojem. Miesięczne koszty utrzymania infrastruktury tego rozwiązania są elastyczne oraz na start wyceniane na 1604€ / miesiąc. Jego wdrożenie zajmie około 5 tygodni przy 4-etatowym nakładzie pracy (2 konsultantów i 2 pracowników E-Beauty). Planowany system danych nie tylko umożliwia zaawansowaną rekomendacje produktów, ale będzie również niezbędną podstawą dla przyszłych projektów analitycznych.

Hurtownia danych

Podstawą każdego algorytmu będącego częścią składową systemu rekomendacji są dane, zatem ich gromadzenie i przetwarzanie należą do zadań priorytetowych. Dane są często przechowywane w systemach obsługujących różne elementy sklepu, takich jak oprogramowanie magazynowe, marketingowe, podsumowania analityczne, jak i platformy marketingowe. W celu skutecznego wykorzystania danych w projektach analitycznych, takich jak system rekomendacji, ważne jest stworzenie jednolitej i przejrzystej hurtowni danych, która gromadzi informacje z różnych obszarów i stanowi centrum wszelkich operacji.

Zastosowanie takiego rozwiązania dostarcza znaczne korzyści dla całej organizacji. Poprzez konsolidację danych z różnych źródeł, takich jak dokonywane transakcje, zachowanie podczas przeglądania produktów i atrybuty produktów, hurtownia oferuje kompletny i dokładny obraz preferencji i zachowań klientów. Ułatwia to proces poszukiwania zmiennych do modeli oraz pozwala na łączenie i agregację informacji z wielu systemów na raz. Dzięki temu procesy tworzenia analiz i algorytmów nauczania maszynowego są znacznie przyśpieszone. Liczba wzajemnych zależności systemów jest zmniejszona, co pozwala unikać nieprzewidzianych

konsekwencji będących rezultatem prac nad strukturą danych lub pojedynczymi systemami.

Konstrukcja hurtowni

Centralna hurtownia danych składa się z tablic, które charakteryzują się różnym poziomem ustrukturyzowania informacji i częstotliwością odświeżania. Systemy kierujące sklepem dostarczają nieustrukturyzowane dane, które są przetwarzane w celu otrzymania tablic, które mogą być wykorzystane w algorytmach. Tablice mogą być publikowane w *batchach* (wypuszczane partiami) , albo *streamach* (aktualizowane na żywo). W proponowanym rozwiązaniu preferujemy jednak korzystanie z batchowych tabeli, ponieważ pozwalają one na prostszą i szybszą implementacje oraz mają niższe koszty utrzymania. Rozwiązania korzystające ze stream-owych tabeli są bardziej użyteczne przy większej skali operacji i mogą być rozważone w przyszłości.

W przygotowanym rozwiązaniu możemy wyróżnić cztery typy tablic:

1) **Tablice z definicjami (dict)** - są one krótsze i pełnią funkcje słowników pozwalających na identyfikację i opis kampanii marketingowych, produktów w sklepie i podobnych wartości, które nie są generowane przez aktywność klientów, lecz są ustalane przez właścicieli biznesowych. Pozwalają między innymi na na sprawdzenie kontekstu wydarzenia, któremu zazwyczaj jest przypisany jedynie numer identyfikacyjny (na przykład na sprawdzenie tagów zakupionego produktu)

2) **Tablice z surowymi danymi (raw)** - są to dłuższe tablice, charakteryzujące się mniejszą strukturyzacją danych. Zawierają one nie zagregowane informacje o aktywnościach klientów, takich jak dokonane transakcje albo interakcję ze stroną sklepu. Dane te są mniej przydatne w modelowaniu preferencji klienta, mogą być użyteczne w zastosowaniach wymagających spersonalizowanych wyników w czasie rzeczywistym.

3) **Tablice agregujące (table)** - są to długie tablice o wysokiej strukturyzacji, które podsumowują aktywność w danym okresie. Są one publikowane w regularnych interwałach i stanowią podstawę do większości stosowanych modeli i analiz.

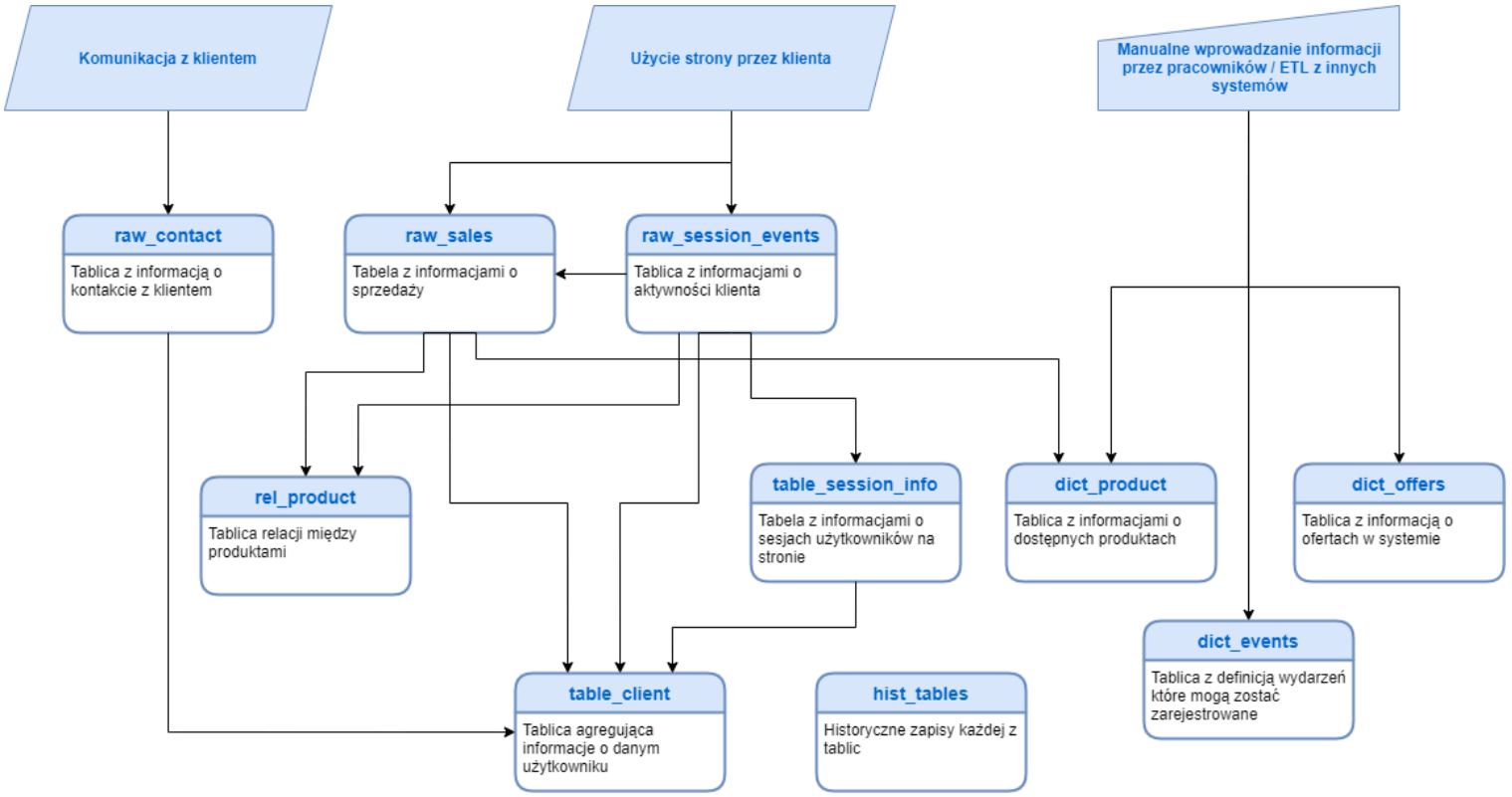
4) **Tablice relacyjne (rel)** - krótsze tablice zawierające informacje o relacji par produktów, na przykład częstotliwość ich wspólnego występowania.

Taki podział typów przechowywanych danych gwarantuje przejrzyste systematyczne podejście, jednocześnie pozostawiając przestrzeń na rozwój w wypadku

rozszerzenia typów informacji. Nowe zmienne mogą być elastycznie dodawane i przeliczane dla przeszłych okresów.

Proponowany podział danych

Wyżej opisany schemat konstrukcji centralnej hurtowni pozwala na stworzenie ekosystemu strumieni danych, dopasowanego do potrzeb firmy E-Beauty. W skupieniu na potrzeby systemu rekomendacji proponujemy następujący początkowy układ zależności tablic:



Układ ten pozwala na bezproblemowe zasilanie projektowanego systemu oraz znacząco ułatwi tworzenie zautomatyzowanych, lub unikalnych raportów. Każda z zaprojektowanych tabel ma swój cel oraz zastosowania:

Dict_events

Tablica-słownik; zawiera informacje o wszystkich rejestrowanych wydarzeniach na stronie. Kod strony internetowej, który rejestruje wydarzenia, używa sprecyzowanych w tej tabeli identyfikatorów. Znaczenie i powiązania danego id mogą zostać w razie potrzeby sprawdzone w tej tabeli.

Przykładowe zmienne:

- Event_id - kod identyfikacyjny wydarzenia
- Descr - opis wydarzenia
- Type - typ wydarzenia (związane z wyszukiwaniem, związane z wybraniem oferty, związane z płatnością, etc.)
- **Pełne listy przewidzianych zmiennych dla każdej tabeli są załączone w dodatkach do raportu.**

Dict_offers

Tablica-słownik; zawiera informacje o wszystkich akcjach marketingowych i promocyjnych. Każda wprowadzona oferta (na przykład kampania mailingowa, kod promocyjny, przypomnienie SMS) ma sprecyzowany identyfikator, zawarty w tej tabeli.

Przykładowe zmienne:

- Offer_ID - numer identyfikacyjny akcji promcyjnej
- Type - typ kampanii, np. Mailing, SMS.
- product_IDS - lista produktów, których dotyczy promoocja

Dict_product

Tablica-słownik; zawiera informacje o wszystkich oferowanych w sklepie produktach.

Przykładowe zmienne:

- Product_ID - numer identyfikacyjny produktu
- Brand_name - nazwa marki produktu
- Premium_flag - flaga wskazująca produkt segmentu premium
- Categories_list - lista zwierająca kategorie i tagi produktu

Raw_session_events

Tablica z surowymi danymi; zawiera informacje o wszelkich aktywnościach użytkownika podczas jego wizyty w sklepie internetowym. Każde wydarzenie uznane za wartościowe do rejestracji i określone w Dict_events tworzy rząd w tej tabeli. W przyszłości może być zmodyfikowana na tabelę typu stream w celu dostarczania personalnych sugestii na podstawie krótkoterminowej aktywności użytkownika.

Przykładowe zmienne:

- Session_ID - numer identyfikujący wizytę klienta w sklepie
- Client_ID - numer identyfikujący klienta
- Event_ID - identyfikator typu wydarzenia
- Datetime - dokładny moment wydarzenia

Raw_sales

Tablica z surowymi danymi; zawiera informacje o złożonych zamówieniach. Każde zrealizowane zamówienie tworzy rząd w tej tabeli. Pozwala na tworzenie zaawansowanych statystyk dla wskazanego okresu.

Przykładowe zmienne:

- Order_ID - numer identyfikacyjny zamówienia
- Client_ID - numer identyfikacyjny klienta
- Datetime - data i godzina zakupu
- Cart_size_value - Łączna wartość koszyka
- Cart_time - czas otwarcia koszyka przed złożeniem zamówienia

Raw_contact

Tablica z surowymi danymi; zawiera informacje o kontakcie marketingowym sklepu z klientem. Pozwala na analizę skuteczności marketingu.

Przykładowe zmienne:

- Client_ID - numer identyfikacyjny klienta
- Offer_ID - numer identyfikacyjny akcji promocyjnej
- Datetime - Data i godzina kontaktu
- Success_cat - zmienna kategoryczna określająca skuteczność kontaktu, (0 - kontakt zignorowany/stan nieznany, 1 - sklep odwiedzony, 2 - dokonany zakup)

Rel_product

Tablica relacyjna zawierająca powiązania pomiędzy parami produktów. Jest ona zasilana przez przetworzone dane z tablic raw_session_events i raw_sales. Powiązania między produktami są szczególnie ważne w modelu rekomendacji Apriori i zrozumieniu sieci powiązanych produktów.

Przykładowe zmienne:

- Product_1_ID - numer identyfikacyjny pierwszego produktu
- Product_2_ID - numer identyfikacyjny drugiego produktu
- Kupowane_Razem_Szansa - wartość określająca jak często produkt 1 jest kupowany razem z produktem 2

- Przejscie_szansa - Wartość wskazująca jak często klient oglądający produkt 1 obejrzał również produkt 2 podczas tej samej sesji

Table_session_info

Tablica agregująca; zasilana przez tabelę raw_session_event. Gromadzi ona wszystkie zarejestrowane wydarzenia i przetwarza na użyteczne zmienne:

Przykładowe zmienne:
- Session_ID - numer identyfikacyjny sesji
- Visit_source - źródło wizyty na stronie
- Visit_duration - czas spędzony przez klienta w sklepie
- Sale_flag - flaga wskazująca czy wizyta doprowadziła do sprzedaży
- Products_viewed_cnt - liczba produktów obejrzanych podczas wizyty

Table_client_info

Tablica agregująca; stanowi główne źródło informacji o klientach i ich aktywnościach. Jest podstawą do trenowania modeli i wyliczania scoringów. Zawarte w niej zmienne są wyliczane na podstawie informacji ze wszystkich tablic z surowymi danymi oraz table_session info. Stosowne zmienne występują w formie trendów zliczeniowych (średnia/całkowita wartość z tygodnia/miesiąca/3 miesięcy itp.)

Przykładowe zmienne
- Client_ID - numer identyfikacyjny klienta
- Client_status - zmienna kategoryczna wskazująca czy klient jest rozpoznany np. po adresie dostawy, czy posiada konto w sklepie.
- Sex_cat - informacja o płci, pozyskana przez wybór Pan/Pani/Inne
- Last_visit_days - liczba dni od ostatniej wizyty na stronie
- Last_sale_days - liczba dni od ostatniego zakupu
- Days_between_sales_avg - średnia liczba dni pomiędzy zakupami
- Cart_value_avg - średnia wartość koszyka
- visits_sum_3M - łączna liczba wizyt przez ostatnie 3 miesiące
- skincare_order_amt_6M - Łączna liczba zamówionych produktów kategorii skincare przez ostatnie 6 miesięcy
- Client_cluster - informacja o przypisanym do klienta segmencie przez model

Hist_tables_{Date}_{Name}

Tablice zawierające dane historyczne z końca każdego miesiąca. Na przykład Hist_table_012022_Raw_sales jest zapisem tabeli Raw_sales z czasu jej publikacji w styczniu 2022 roku. Dane historyczne są niezbędne w długoterminowych analizach i trenowaniu modeli nauczania maszynowego. Długość przechowywania danych zależy od potrzeb zastosowania. Dane rzadko używane mogą zostać

przeniesione do tańszej alternatywnej przestrzeni dyskowej, przeznaczonej do archiwizacji.

Implementacja systemu danych

Hurtownie danych są aktualnie wiodącym systemem przechowywania informacji na dużą skalę, dzięki czemu jest dostępnych wiele gotowych rozwiązań ułatwiających ich implementacje. Wybór pomiędzy nimi jest skomplikowany, dlatego proponujemy w raporcie najbardziej odpowiednie według naszej analizy dla E-Beauty, razem z uzasadnieniem jego wyboru.

Hurtownia lokalna vs w chmurze

Wybór pomiędzy rozwiązaniem lokalnym, a opcją wykorzystania chmury jest zależny od różnych czynników związanych z zastosowaniem technologii. Oba typy infrastruktury mają swoje plusy i minusy, które trzeba ocenić w kontekście projektu.

+ Niższe koszty stałe
+ Pełna kontrola nad systemem
- Wysokie koszty początkowe
- Niższa skalowalność
- Konieczność zatrudnienia specjalistów

+ Wysoka skalowalność
+ Niskie koszty początkowe
+ Zaawansowane systemy bezpieczeństwa
+ Mniejsze zapotrzebowanie na specjalistów
- Trudniejsze w integracji
- Mniejsza personalizacja
- Długoterminowo droższe

Biorąc pod uwagę wyżej wymienione czynniki, **zalecamy zastosowanie chmury** do stworzenia Centralnej Hurtowni Danych. E-Beauty jest młodą, dynamicznie rozwijającą się firmą. Stworzenie lokalnej hurtowni danych wiązałoby znaczące środki finansowe i pochłonęło czas pracowników. Niosłoby to również za sobą znaczące ryzyko związane z płynnością firmy w wypadku potrzeby zmniejszenia rozmiaru operacji. Aktualnie nie posiadamy również pełnych danych dotyczących zapotrzebowania organizacji na moc obliczeniową i przestrzeń dyskową, więc zakup wymaganych urządzeń musiałby się opierać na wyliczeniach obarczonych błędem.
Przy wykorzystaniu rozwiązania opartego na chmurze projekt może się sam finansować i rosnąć w nieograniczonym tempie. Ryzyka związane z utratą danych,

bądź ich wyciekiem są zminimalizowane. W przyszłości, kiedy zapotrzebowanie firmy będzie dobrze udokumentowane i zrozumiałe możliwa będzie migracja na rozwiązanie lokalne.

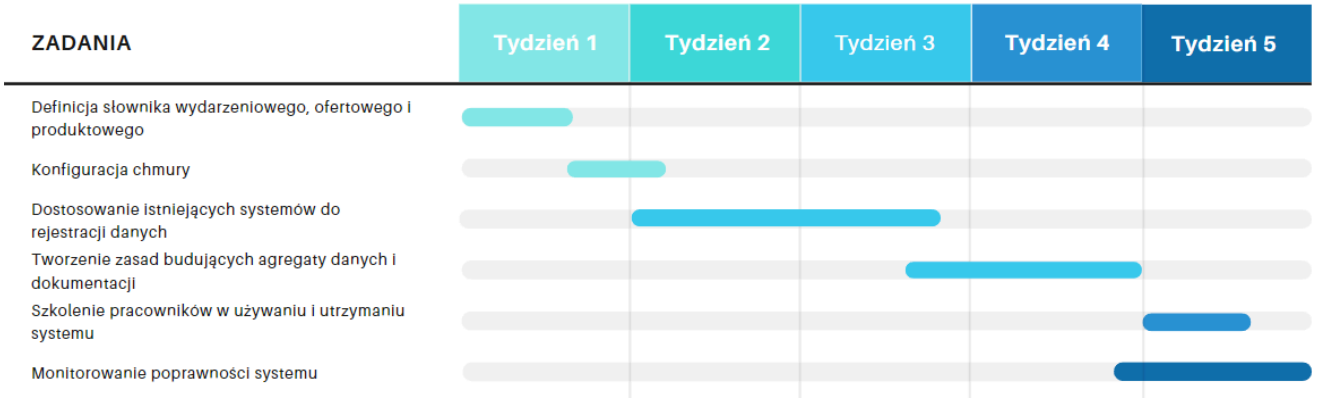
Rozwiązanie komercyjne i koszt

Rekomendowanym przez nas dostawcą chmury jest **Azure Synapse.** Usługa ta charakteryzuje się wysoką skalowalnością oraz efektywnością w operacjach na dużych zbiorach danych, osiąganą dzięki technologii rozproszonych obliczeń. Nasz zespół posiada doświadczenie we wprowadzaniu tej technologii u klientów, co zapewnia bezproblemową implementacje i możliwość szybkiego przeprowadzenia dedykowanych szkoleń z obsługi systemu. Jej minusem zwiększona, lecz nie nadto znacząco, trudność w integracji z systemami nie stworzonymi przez Microsoft.

Zakładając początkowe zapotrzebowanie na 25 TB miejsca na dysku, oraz rezerwacje urządzenia klasy Premium V2 (4 rdzenie, 14 GB RAM, 250GB storage) w celu przeliczania modeli, miesięczne koszty rozwiązania będą wynosiły 1064€ za przechowywanie danych oraz 540€ za dostęp do mocy obliczeniowej. Łączny koszt stosowania rozwiązania to **1604€ / miesiąc[2].** Specyfikacja usługi może w każdym miesiącu być dostosowywana do potrzeb, aby zapewnić optymalne użycie środków.

Logistyka wdrożenia

System danych jest podstawą do budowy systemu rekomendacji, dlatego powinien zostać wprowadzony w pierwszej kolejności. Zakładając alokowanie 4 etatów do projektu (2 konsultantów i 2 pracowników E-Beauty), przewidywana chronologia projektu wygląda następująco:



Do utrzymania i rozwoju hurtowni danych **zalecamy stworzenie jednego etatu**. Wraz z rozwojem zaawansowania systemu może zostać stworzony zespół

[2]Azure calculator

specjalistów od Inżynierii Danych i Data Science, do którego odpowiedzialności będzie należało utrzymanie i rozwój środowiska oraz przygotowywanie automatycznych i unikalnych analiz.

Modele preferencji

Posiadając jednolity system informacji, można rozpocząć konstrukcję architektury silnika poleceń. W celu osiągnięcia skutecznej rekomendacji produktu możemy użyć licznych modeli preferencyjnych. Ich generalnym założeniem jest użycie posiadanych przez nas danych do stworzenia hierarchii produktów, według której będziemy priorytetować co zaprezentować klientowi.

Poszczególne modele preferencji sprawdzają się najlepiej w określonych warunkach. Mają swoje unikatowe źródła danych i założenia. Pojedynczo nie są w stanie stworzyć wszechstronnej metody rekomendacji produktów. W tym celu zastosujemy schemat hybrydowego modelu rekomendacyjnego, opisany w literaturze (Y. Gao et. al., 2017), który połączy wyniki algorytmów preferencyjnych reprezentujących zupełnie inne podejścia.

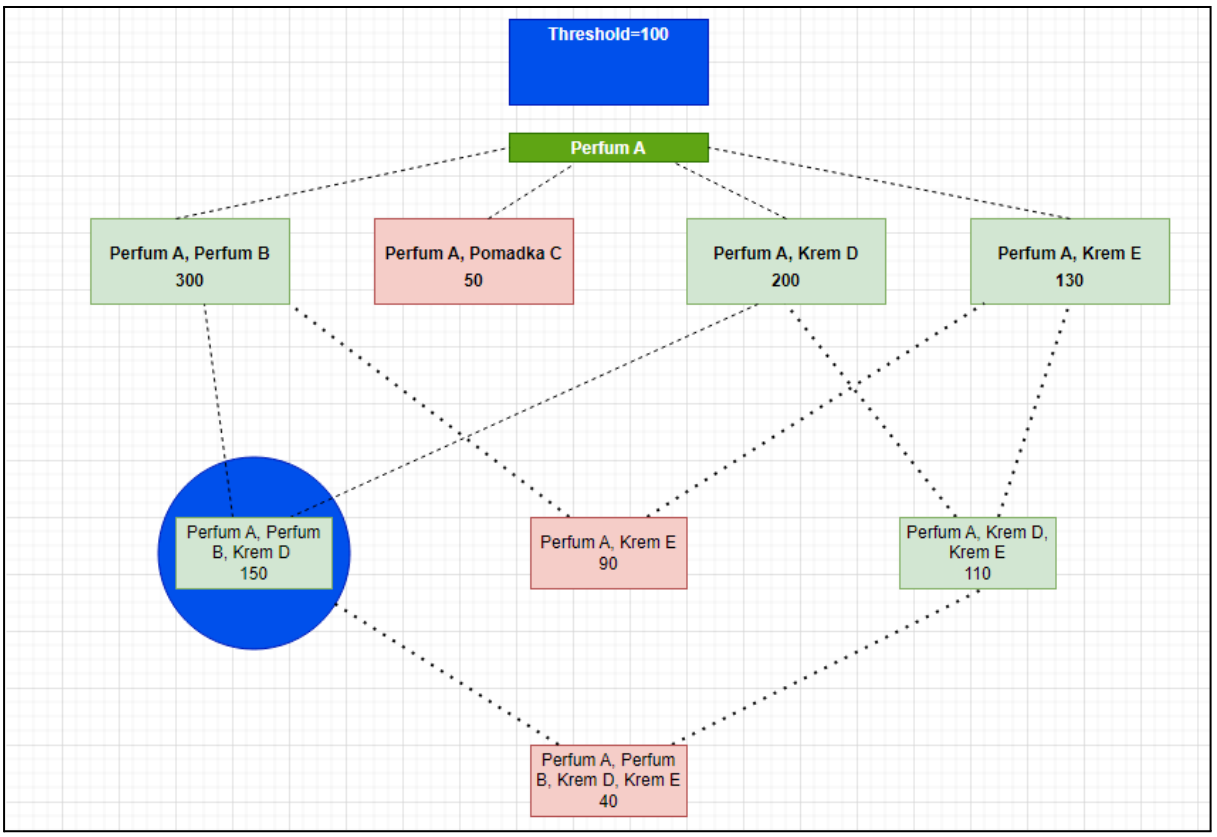
W tym dziale zgłębiamy poszczególne proponowane modele preferencyjne, ich silne i słabsze strony oraz implementacje w architekturze IT firmy. W przyszłości jest możliwy dalszy rozwój i tworzenie nowych, ambitnych modeli, które są podsumowująco opisane w dziale “Przyszłe usprawnienia”. Na tym etapie rozwoju planujemy zbudowanie **czterech modeli preferencyjnych**, które mogą być **implementowane w operacji sklepu niezależnie** od wzajemnego stanu rozwoju.

Algorytm Apriori

Pierwszym proponowanym modelem jest algorytm *Apriori*, którego przewidywania operają się na wcześniej dokonanych transakcjach. Zakłada on, że pewne produkty naturalnie występują razem. Porównuje obecny koszyk użytkownika z innymi koszykami które zostały zakupione w przeszłości. W wypadku gdy koszyk jest pusty, algorytm może również użyć oglądanych przez użytkownika produktów, co pozwala na użycie go zarówno podczas procesu zakupów i procesu check-outu.

Rozwiązania używające tego algorytmu charakteryzują się wysoką dokładnością oraz skalowalnością. Są w stanie działać nawet przy bardzo małej ilości danych, ponieważ są zależne od danych całego sklepu, a nie konkretnego użytkownika. Ich wadą jest natomiast brak personalizacji wyjątkowej dla każdego klienta.

Aby wytłumaczyć działanie algorytmu przeanalizujmy przykład. W ofercie mamy 5 produktów (A-E) i klient wsadza do koszyka produkt A. Algorytm żeby ocenić najlepsze preferencje będzie rekursywnie symulował dodawanie po jednym kolejnych produktów do koszyka. Każda kombinacja poniżej wartości krytycznej, (którą jest ilość zakupionych koszyków o danej specyfikacji),będzie odrzucona (zaznaczone na czerwono na schemacie poniżej), a do każdej pozycji powyżej tej wartości dodawany nowy produkt (zaznaczone na zielono). Algorytm kończy swoją pracę gdy ostatnia warstwa drzewka będzie cała czerwona (wszystkie kolejne propozycje odrzucone z powodu zbyt małej ilości koszyków), lub zostanie tylko jedna zielona propozycja w ostatniej warstwie. W pierwszym przypadku algorytm wraca poziom wyżej i wybiera najczęściej występujący zielony koszyk, w drugim przypadku jest to zielony koszyk w ostatniej warstwie. W poniższym przykładzie (scenariusz 1) finalną opcję zaznaczono niebieskim kółkiem. Algorytm w pierwszej kolejności poleci perfum B i krem D. Na mniej ważnej pozycji znajdzie się krem E.



W momencie gdy mamy dużo produktów a chcemy polecić tylko ich ograniczoną ilość, możemy dodać wartość “items” która limituje wartość drzewka do określonej ilości warstw. Jeżeli chcemy polecić tylko 2 przedmioty drzewko zatrzymuje się na drugiej iteracji tworzenia nowych koszyków i wybiera najczęstszy z nich, oszczędzając dalsze obliczenia.

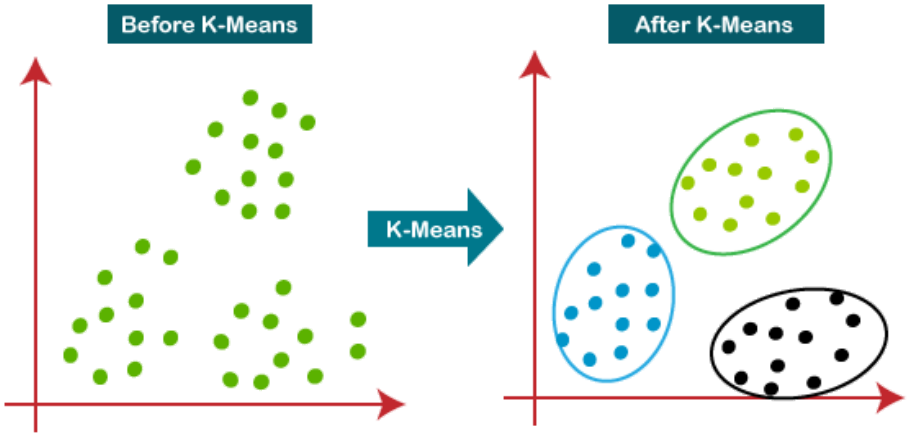
W architekturze IT firmy E-Beauty algorytm Apriori może zostać zastosowany na dwa sposoby.
Podstawowe zależności między produktami mogą być okresowo wyliczane z danych sprzedażowych zawartych w tablicy raw_sales_info i rejestrowane w tablicy relacyjnej rel_products. Bardziej skomplikowane relacje mogą być przechowywane w podobnej tablicy w zależności od dostępności miejsca dyskowego. W takim wypadku, kiedy klient dodaje produkt do koszyka, lub obserwuje dany produkt, algorytm zczytuje wcześniej wyliczoną hierarchię rekomendacji.
Drugim możliwym rozwiązaniem jest przygotowywanie rekomendacji dopiero w momencie kiedy klient dodaje produkt do koszyka. W tym wypadku dane sprzedażowe są trzymane lokalnie na serwerze i bardziej skomplikowane relacje, bazujące na 4 i więcej produktach nie muszą być przechowywane. Dojście rekomendacji do skutku może zależeć od aktualnej dostępności mocy obliczeniowej serwera strony.

Collaborative filtering

Rozwiązania typu collaborative filtering zakładają ekstrapolowanie na podstawie porównań z podobnymi przypadkami obserwacji. W celu prowadzenia bardziej spersonalizowanego marketingu oraz zasilenia modeli o dodatkową zmienną proponujemy najpierw użycie nienadzorowanego nauczania maszynowego. Dzięki niemu zrozumiemy grupy klientów sklepu. Dla predykcji preferencji użyjemy również metody KNN.

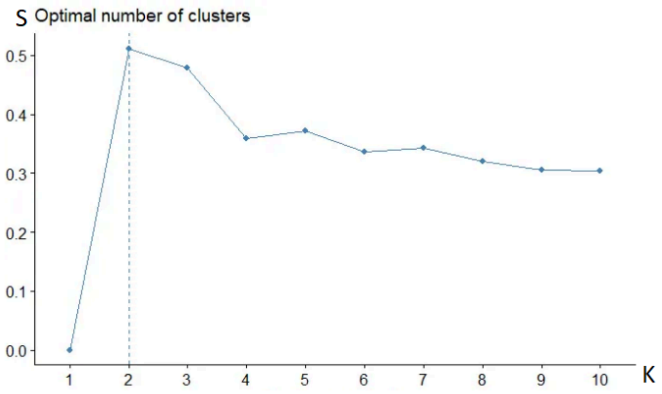
Segmentacja klientów K-means

Jedną z metod pozwalającą na szukanie nieoczywistych podobieństw między grupami klientów jest K-means clustering. Rozwiązanie to dzieli klientów na grupy, które są do siebie algebraicznie zbliżone. Analiza danych każdej grupy często pozwala na intuicyjne zrozumienie w jaki sposób dani klienci są do siebie podobni. Każdy wyznaczony segment można nazwać i opisać, co jest użyteczne na potrzeby skutecznego marketingu i może posłużyć jako pochodna zmienna do innych modeli.



Źródło: https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning

Model pozwala na przydzielanie segmentu również nowym użytkownikom po zebraniu wystarczającej ilości danych. Trudnością w jego zastosowaniu jest konieczność sprecyzowania liczby zidentyfikowanych segmentów. W wyborze może pomóc między innymi statystyka Silhouette, która mierzy jak bardzo obserwacje są podobne do swojej grupy w porównaniu do innych. Liczba segmentów jest zazwyczaj optymalna dla najwyższej kumulatywnej statystyki Silhouette.



Wynik modelu tworzy nową zmienną tablicy table_client_info - segment do którego klient należy. Dzięki temu możmy kierować bardziej spersonalizowane oferty do danego segmentu. Segmentacje również są dobrym źródłem dla innych modeli. Model musi być okresowo (np. Co 6 miesięcy) odświeżany przez osobę posiadającą umiejętności z zakresu Data Science.

Porównywanie klientów K-Nearest Neighbours

Algorytm K-Nearest neighbours zakłada, że podobni ludzie będą dalej podejmować podobne decyzje konsumenckie. Bierze on pod uwagę wielowymiarowe zagregowane dane o kliencie i porównuje z innymi klientami, żeby znaleźć najbardziej podobnych. Następnie poleca on najczęściej występujące produkty w koszykach zidentyfikowanych osób, których analizowany klient jeszcze nie kupił. Hiperparemetrem algorytmu jest liczba podobnych klientów (k), ustalana porównując wyniki różnych wersji modelu.

W odpowiednich okolicznościach model może być nadzwyczaj skuteczny i precyzyjnie identyfikować nieoczywiste zależności. Potrzebuje jednak do tego dużo danych o użytkowniku i podobnych klientach. Używa też on wiele zasobów obliczeniowych, sprawiając, że proces przewidywania musi się odbywać jeszcze przed ponownym odwiedzeniem strony przez klienta.

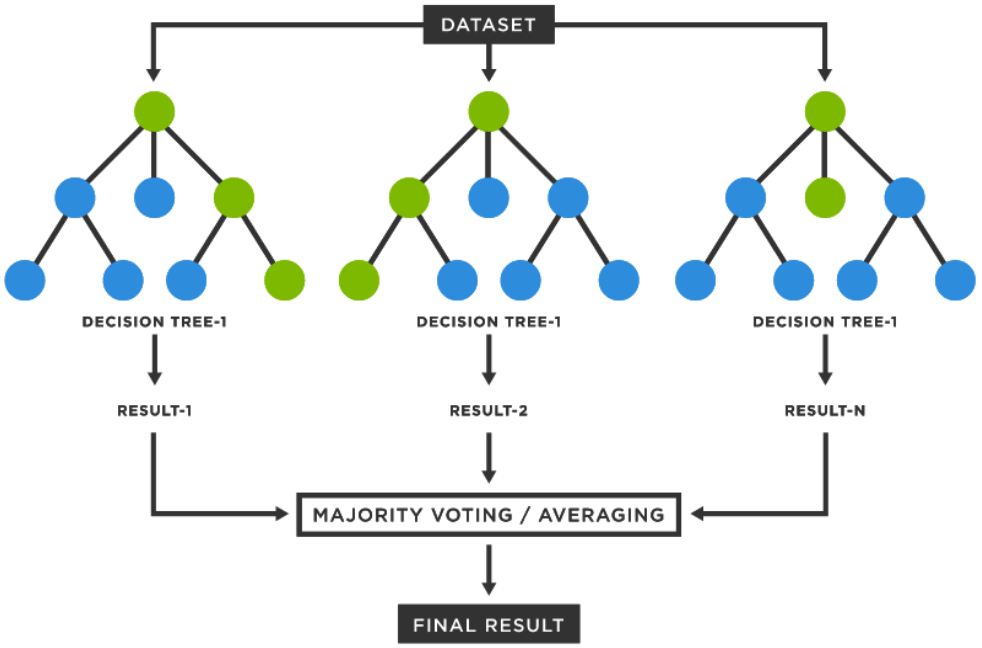
Specyfika biznesu firmy E-Beauty bardzo dobrze pasuje do tego typu modelu. Hierarchia rekomendacji dla stałych klientów może być ciągle odświeżana w tle

działalności strony. Kiedy klient odwiedzi sklep będą na niego czekały gotowe rekomendacje.

Content based filtering używając Random Forest

Ten typ przewidywania preferencji klienta zakłada, że na podstawie wcześniejszej aktywności klienta w sklepie i jego pozostałych cech charakterystycznych (wiek, płeć etc.) jesteśmy w stanie przewidzieć jego przyszłe preferencje zakupowe. W tym celu może zostać wykorzystanych wiele modeli (na przykład regresja logistyczna, ekstremalny boosting gradientowy, drzewko decyzyjne). Na podstawie naszego wcześniejszego doświadczenia i przykładów z literatury proponujemy i opisujemy użycie modelu typu Random Forest o specyfikacji One vs Rest, który jest często używany w serwisach e-commerce. Jeżeli prace nad danymi wskazałyby lepszą efektywność innego modelu tej kategorii, może on być w łatwy sposób zmieniony.

Ideą modelu jest generacja wielu małych losowych drzew decyzyjnych, które grupują wszystkie obserwacje danych i informują o statystycznym prawdopodobieństwie przynależności obserwacji do danej grupy. W naszym przykładzie trenowany jest osobny las dla każdego produktu, który decyduje o prawdopodobieństwie, że klient kupi dany produkt w stosunku do grupy wszystkich pozostałych produktów. Każdy produkt otrzymuje ocenę prawdopodobieństwa, co tworzy hierarchię sugestii. Model działania algorytmu jest zilustrowany poniżej.

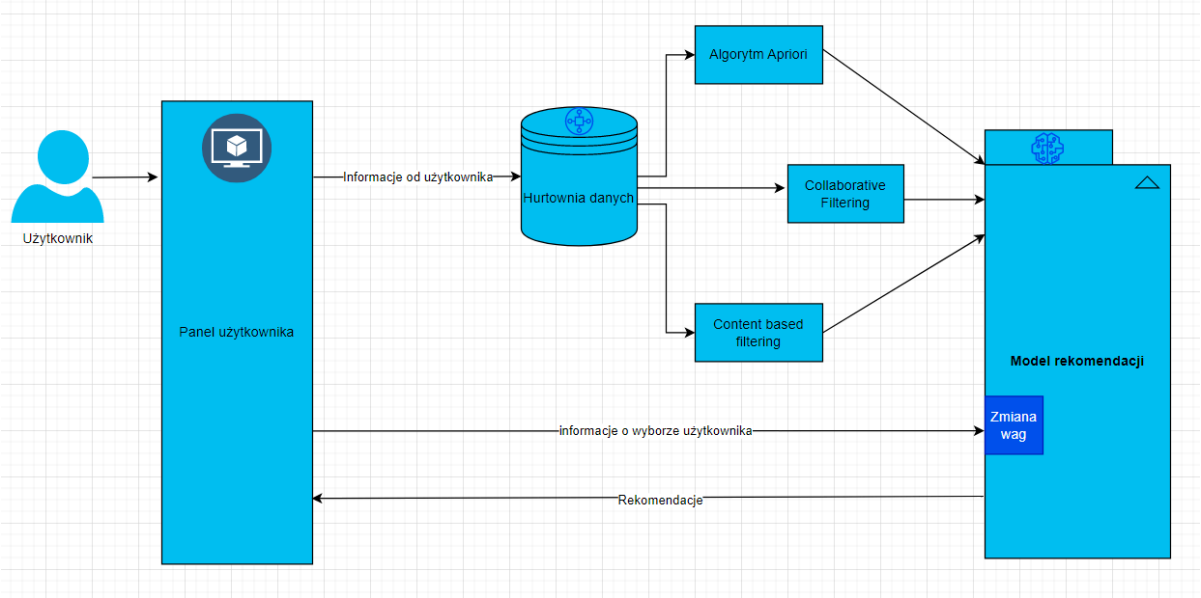


Źródło: https://www.tibco.com/reference-center/what-is-a-random-forest

Przy odpowiedniej ilości danych rozwiązanie to prowadzi do bardzo dobrych wyników. Jego problemem jest długi proces trenowania modelu i konieczność częstego odświeżania na potrzeby wprowadzania nowych produktów. Po wytrenowaniu modelu przewidywanie nowych obserwacji jest już szybszym procesem. Implementacja tego rozwiązania w systemach E-Beauty jest analogiczna do modelu K-Nearest Neighbours z różnicą konieczności częstszego odświeżania modelu.

Model rekomendacji

Aby uzyskać jak największą precyzję i poprawność generalnej rekomendacji proponujemy model będący hybrydą powyżej opisanych metod. Model będzie automatycznie cyklicznie aktualizowany. Jego pierwszym etapem jest zbieranie danych i przygotowanie do analizy danych od konsumentów. Następnie za pomocą tych informacji wyliczone zostaną scoringi poszczególnych modeli preferencji. Informacje o ich estymowanej relatywnej poprawności, precyzji, i aplikowalności w kontekście obserwacji w połączeniu z danymi o poprzednich udanych rekomendacjach, posłużą do determinacji wag które zostaną dane poszczególnym wynikom. Co ważne, wagi są spersonalizowane, co pozwala na dokładne predykcje zróżnicowanych sposobów działania konsumenta. Następnie średnia ważona wyników decyduje o finalnej hierarchi rekomendacji, która określa wyświetlane produkty. Po zakupie, lub sprawdzeniu produktu dane w tabelach w hurtowni danych zostaną zaktualizowane a wraz z nimi model, który ponownie zacznie swój cykl.



Taka struktura systemu jest elastyczna i pozwala na efektywne rekomendacje w każdym momencie rozwoju biznesu. W momencie posiadania małej ilości danych algorytm Apriori jest w stanie stworzyć rozsądne predykcje, a wraz ze wzrostem liczby danych content based i collaborative filtering stają się coraz bardziej dokładne

i celne. Wcześniejsze zastosowania wskazują, że algorytm w pełni swoich możliwości jest w stanie operować z 67% dokładnością, podczas gdy jego poszczególne elementy osobno są w stanie operować w granicach tylko 40-50% dokładności[3].

W przyszłości, w przypadku chęci rozwoju systemu można dodać lub zastąpić algorytmy preferencji bez potrzeby zmiany innych części modelu rekomendacji. Architektura pozwala również na stosunkowo łatwe przejście na updateowanie modeli w czasie rzeczywistym jeżeli w przyszłości będzie to opłacalne.

Przykłady zastosowań algorytmów

Zastosowane rozwiązanie hybrydowe pozwala na etapową implementację wielu różnych typów rekomendacji w zależności od lokalizacji na stronie, lub kanału informacji. Każdy typ może być wspierany osobnym algorytmem oraz może pojawić się w produkcyjnej wersji strony niezależnie od pozostałych. Proponowane przez nas implementacje są zaprezentowane poniżej.

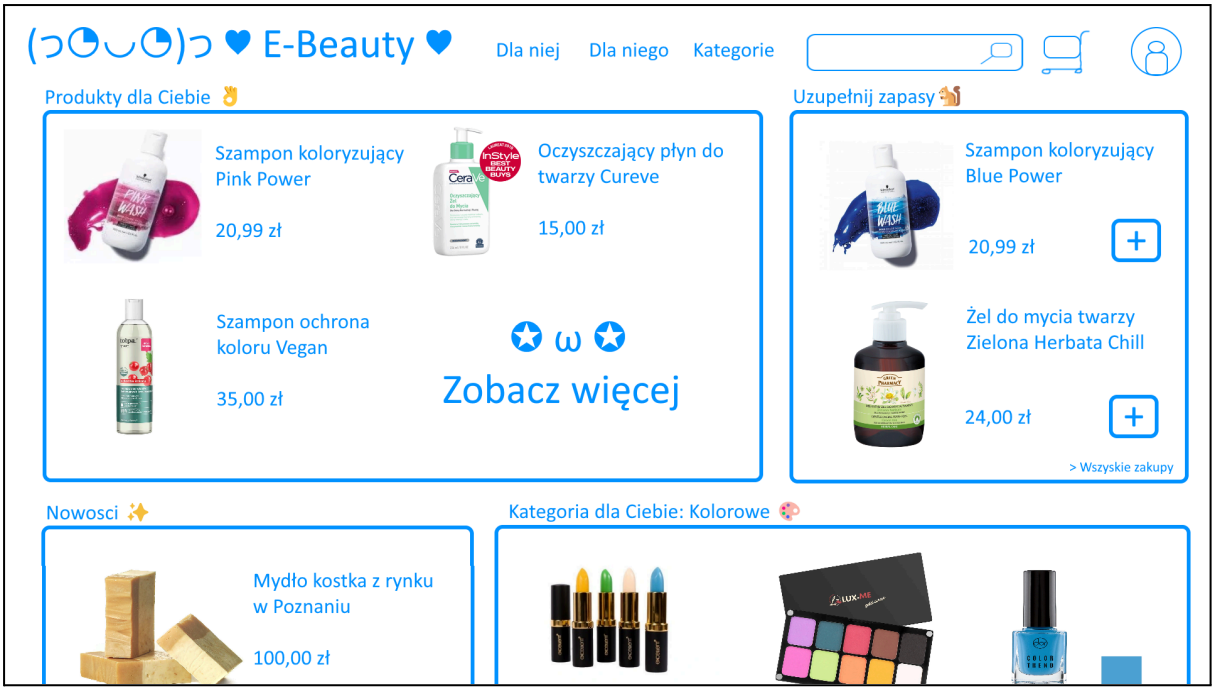
Rekomendacje na stronie głównej

W momencie w którym użytkownik wejdzie na stronę główną, pojawią się 4 zakładki:

- Produkty dla Ciebie: rekomendacje stworzone przez silnik
- Uzupełnij zapasy: kupione jakiś czas wcześniej produkty, które mogły się skończyć na podstawie zbieranych danych o ponownym zakupie
- Nowości: Najnowsza oferta sklepu, nie ma nic wspólnego z preferencjami, ale pomaga gromadzić dane o nowych produktach do użytku w modelach
- Kategoria dla Ciebie: Produkty z kategorii która pojawia się najczęściej w szczycie hierarchii preferencji produktów

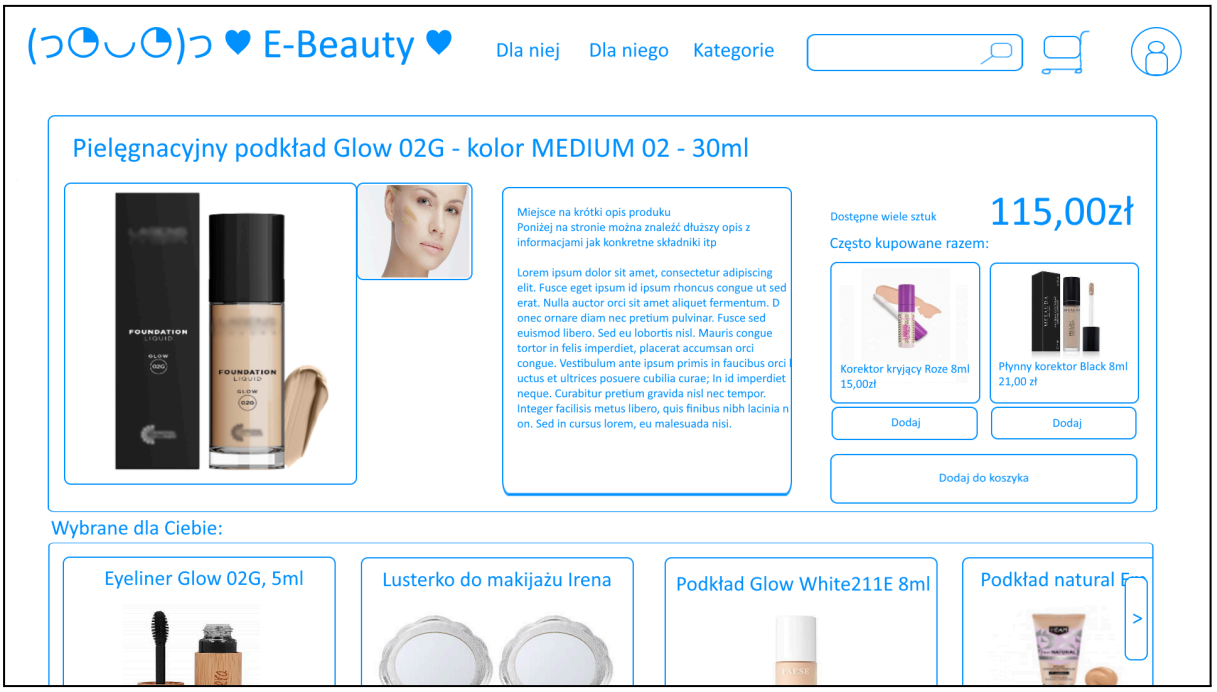
Użytkowniczka ma 20 lat, zakupiła miesiąc temu żel do mycia twarzy, niebieski szampon koloryzujący, fioletową szminkę i czerwony lakier do paznokci. W zakładce “Uzupełnij zapasy 🐿️” znajduje się oferta ponownego zakupu szamponu i żelu. W zakładce “Produkty dla ciebie👌” znajdują się oferty polecane przez model rekomendacji: szampon do ochrony włosów, szampon koloryzujący (tym razem innego koloru) oraz oczyszczający płyn do twarzy. W zakładce Kategoria dla Ciebie znajdują się produkty z najbardziej popularnym wśród wyników preferencji użytkowniczki tagiem: kolorowe. Są to szminki i lakiery do paznokci. Tak zakładka ma za zadanie poszerzyć zainteresowania użytkownika produktami, dla największej efektywności biorąc pod uwagę jego preferencje. Kolejność i zawartość zakładek jest elastyczna i może zostać ustalona z zespołem biznesowym.

[3] (Guo, Wang and Li, 2017)



Rekomendacje na stronie produktu

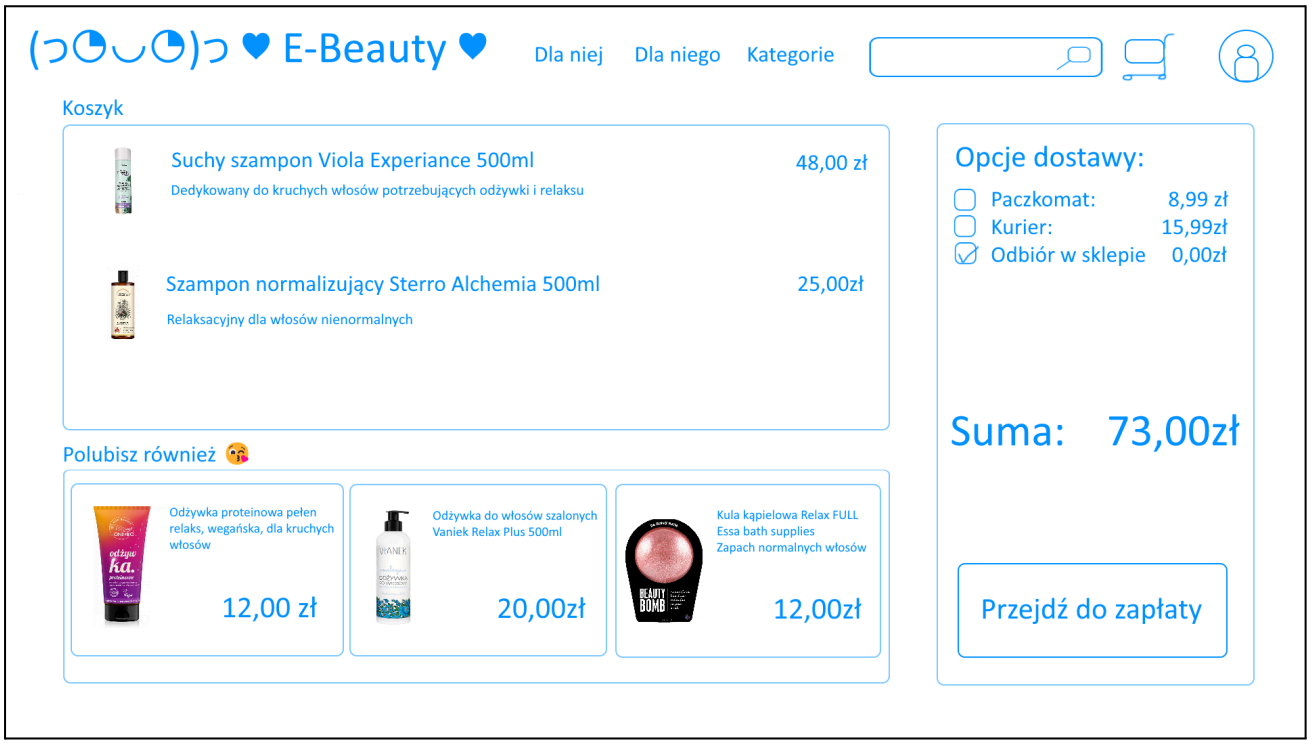
Na stronie produktu pojawią się dwie lokalizacje z rekomendacjami. Pasek “wybrane dla ciebie” zawiera predykcję silnika rekomendacji, podczas gdy element “często kupowane razem” zawiera produkty najczęściej kupowane razem z oglądanym produktem według tablicy rel_produkty. Można je łatwo i przyjemnie dodać do koszyka.



Rekomendacje przy zakupie

W momencie w którym klient przejdzie do koszyka pod jego zawartością wyświetli się pasek z rekomendacjami algorytmu zatytułowany “Polubisz również 😘”.

Użytkownik x, o którym nie mamy żadnych informacji, pierwszy raz odwiedził stronę, nie jest zarejestrowany i dodał do koszyka dwa szampony pielęgnujące. W tym momencie algorytm Apriori po sprawdzeniu najczęstszych koszyków z dodanymi produktami i algorytm content based filtering ustaliły 3 najbardziej prawdopodobne produkty jakim użytkownik byłby zainteresowany. Są to 2 odżywki do włosów i jedna kula do kąpieli. Algorytm collaborative filtering otrzymał w tym przypadku wagę 0 z powodu braku informacji na temat klienta. Gdy klient kupi produkty, dostarczy nam dane na temat, płci, wieku, adresu ip, itp, co sprawi, że collaborative zacznie działać i z kolejnymi wizytami jego wpływ na rekomendacje zwiększy się.



Mailing

W ramach budowania relacji z klientem warto regularnie wysyłać, przypomnienia, oferty czy zapytania. Taką formę komunikacji można również efektywnie zintegrować z silnikiem rekomendacji. Poniżej zamieściliśmy nasze propozycje na kategorie mailów:

-Mail ze zniżką na rekomendowane przez silnik produkty przed świętami komercyjnymi

-Mail ze zniżką na urodziny na rekomendowane produkty

-Mail z ofertą “Czy chcesz kupić ponownie” jeżeli średni czas od kolejnego zakupu użytkowników miną. W mailu znajdowałyby się również rekomendacje “Polubisz również”. Mail taki były wysłany nie częściej niż 2 tygodnie do użytkowników którzy kupili co najmniej 2 razy ten sam produkt. Celem jest utrzymanie kontaktu z klientem.

-Mail z prośbą o ocenę produktu. W mailu znajdowałyby się również rekomendacje “Podobał ci się produkt? Polubisz również:”. W ten sposób zdobywamy cenne dane o produktach jednocześnie utrzymując kontakt z klientem.

-Maile z ofertami promocyjnymi wygenerowanymi dla specyficznego klastra konsumentów z modelu collaborative filtering. Częstotliwość w zależności od preferencji sklepu.

Rozwiązania problemu “cold start”

Problemem proponowanego systemu rekomendacji może być uwzględnienie w nim pojawienia się nowego produktu lub nowych użytkowników. Model może w przypadku użytkownika z niewystarczającą ilością danych nie móc ustalić dobrych rekomendacji a nowy produkt może być omijany. W obu przypadkach proponujemy bardzo proste i skuteczne rozwiązania.

Nowy klient

W momencie gdy nowy klient zakłada konto, wyświetla mu się “personalny asystent”, który pyta go czym byłby zainteresowany. W odpowiedzi użytkownik klika tagi, które filtrują sklep i potem są używane do wstępnego trenowania modelu. Ta forma “interakcji z asystentem” postrzegana bardziej jako interakcja z człowiekiem dodatkowo zwiększa zaufanie klienta.[4]

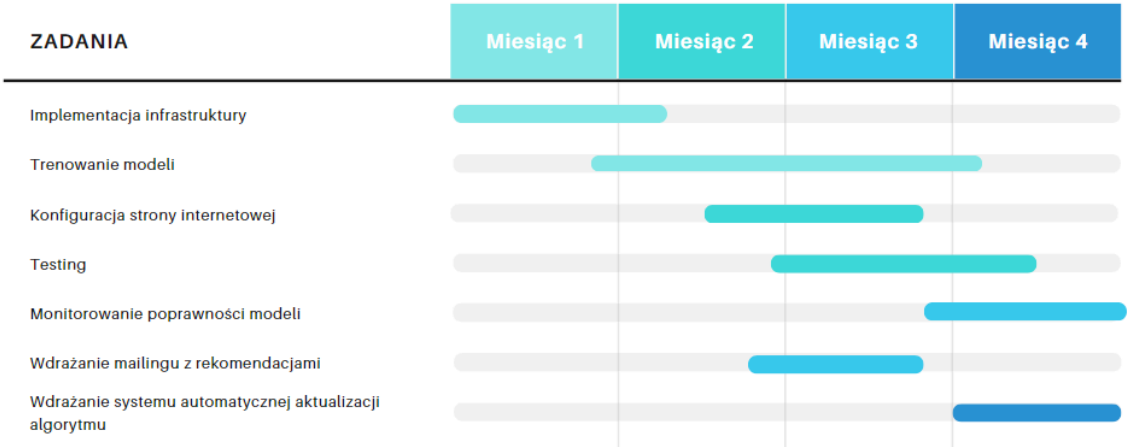
[4] (Swearing, K.; Sinha, R. Beyond algorithms, 2011)

Nowy produkt

W celu pozyskania danych o klientach zainteresowanych produktem, jest on wyróżniany w zakładce “Nowość!” na stronie głównej. Może on być również wypromowany do wybranego segmentu klientów poprzez mailing (jeśli wprowadzamy na przykład nowy perfum o wysokiej cenie, może on być wysłany do klastra charakteryzującego się preferencją produktów premium).

Implementacja

Szacujemy, że implementacja przygotowanego rozwiązania zajmie 4 miesiące, co przedstawiono na diagramie poniżej. Wdrażanie systemu można rozpocząć dopiero 2-3 miesiące od wdrożania hurtowni danych, ponieważ potrzeby jest czas na zebranie i przeprocesowanie danych od konsumentów. Pierwszy miesiąc zajęłoby tworzenie infrastruktury, następnie można zacząć budować modele i od połowy drugiego miesiąca zacząć konfigurować stronę internetową poprzez integrowanie modeli w kolejności ich gotowości. Następnie w 3 miesiącu ważnym elementem będzie intensywne testowanie, żeby wyłapać wszystkie błędy i nieścisłości w początkowej fazie działania. Działalność modelu powinna być jakiś czas po tym monitorowana. Mając wytrenowany pierwszy model na koniec 2 miesiąca zaczynamy pracę nad systemem mailingu. W 4 miesiącu do działającego i zbudowanego modelu stworzymy system automatycznej aktualizacji algorytmu.



Do stworzenia systemu potrzeba będzie co najmniej 2 konsultantów, pracownika zajmującego się data warehouse i jednego nowo zatrudnionego inżyniera machine learning. Pomimo, że proces zajmie 4 miesiące, pojedyncze modele można deployować na stronie już w połowie drugiego miesiąca. Pozwoli to jednocześnie zacząć testing i jak najszybciej czerpać korzyści z inwestycji. Na stałe do opieki nad systemem powinien zostać przydzielony jeden pracownik.

Przyszłe rekomendacje

Usprawnienie modelu mailowego

Żeby zbudować lepszą relację z klientem należy dążyć do stworzenia jak największego średniego impaktu per mail. W związku z tym należy przeanalizować reakcję użytkowników na wysyłane im wiadomości: pozbyć się tych które mają najmniejszy odzew a wysyłać więcej tych, z którymi jest najwięcej pozytywnych interakcji.

Rekomendacje aktualizowane w czasie rzeczywistym

Obecnie system rekomendacji jest aktualizowany w danym interwale czasowym. Wraz ze wzrostem sklepu stanie się opłacalne stworzenie infrastruktury która pozwoli na pobranie od klienta danych w momencie ich wygenerowania i natychmiastowej aktualizacji systemu. Pozwoli to na jeszcze większą dokładność i dynamikę predykcji.

Przejście z chmury na on-premise

Rozwiązanie w chmurze jest efektywnym rozwiązaniem dla małych i średnich biznesów. Jednak gdy przedsiębiorstwo staje się większe, bardziej opłacalnym staje się stworzenie własnej infrastruktury i wyszkolenie własnego personelu. Migracja danych do takiego systemu jest bardzo czasochłonnym procesem a sama decyzja wiąże za sobą duże obciążenie finansowe w krótkim terminie. W długim terminie pozwala zaoszczędzić dużo środków finansowych i polepszyć cyberbezpieczeństwo.

Podsumowanie

Powyższy raport zgłębia kompleksową implementację systemu rekomendacji. Podstawą tego systemu jest data warehouse: infrastruktura gromadząca, przechowująca i agregująca dane do analizy i trenowana algorytmów. Oparty na 3 wytrenowanych modelach preferencji model rekomendacji pozwala na dokładne i celne przewidzenie potrzeb klientów. Algorytm ten można zastosować zarówno bezpośrednio na stronie internetowej jak i w systemie mailowym, który oprócz rekomendacji pozwala budować relacje z klientem. Logistycznie całość systemu jest do wprowadzenia w około 7 miesięcy (1 miesiąc wprowadzanie data warehouse, 2 miesiące zbierania danych i 4 miesiące tworzenia modeli). Do utrzymania i rozbudowy systemu zalecamy zatrudnienie dwóch nowych wyspecjalizowanych pracowników.

Słownik

Centralna hurtownia danych	System przechowujący wszystkie obecne i historyczne dane biznesowe w jednym miejscu
Batch tables	Tablice aktualizowane partiami
Stream tables	Tablice aktualizowane w czasie rzeczywistym
Surowe dane	Nieprzetworzone dane w formie w jakiej zostały sczytane
Apriori	“Uprzedzając fakty” Również algorytm którego celem jest stworzyć asocjacje między danymi
Collaborative filtering	Metoda polegająca na filtrowaniu produktów na podstawie tego czy podobnym użytkownikom podoba się dany produkt
Content based filtering	Metoda polegająca na filtrowaniu produktów na podstawie poprzednich zakupów użytkownika i/lub feedbacku
Silnik rekomendacji	Całość systemu podejmującego decyzje w sprawie przewidywanych preferencji klienta
Data warehouse	Hurtownia danych
Cold start	Problem z brakiem danych podczas wprowadzania nowych produktów i użytkowników
Coco Jumbo Heurystyka International	Dedykacja dla tego zespołu z zeszłego roku 💙

Źródła

1. McKinsey. (2013, October 1). How retailers can keep up with consumers. Retrieved April 2, 2023, from https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers

2. 2. Guo, Y., Wang, M. and Li, X. (2017) ‘An Interactive Personalized Recommendation System Using the Hybrid Algorithm Model’, *Symmetry*, 9(10), p. 216. Available at: https://doi.org/10.3390/sym9100216.

3. Azure cost calculator, available at: https://azure.microsoft.com/en-us/pricing/calculator/?service=synapse-analytics

4. Swearing, K.; Sinha, R. Beyond algorithms: An HCI perspective on recommender systems. In Proceedings of the ACM SIGIR Workshop on Recommender Systems, New York, NY, USA, September 2011; pp. 1–11.

5. https://www.be-terna.com/insights/recommendation-systems-in-e-commerce-whats-the-thing-youve-never-known-but-always-wanted-to

Dodatki

Zmienne w tablicach danych

Poniżej zamieszczone są informacje ilustracyjne dotyczące możliwych zmiennych w tablicah danych. Kompletne listy powstaną po dogłębnym zrozumieniu możliwości strony.

raw_sales_info

client_ID	Po koncie klienta, albo adresie email do potwierdznia, karcie etc.
Rozmiar koszyka - suma sprzedaży	
Rozmiar koszyka - ilość produktów	
Data zakupu	
Produkty z przeceny?	
ID_promocji_przeceny	lista kodów użytych
Array product_ID	
Prezent?	
Czas w koszyku	Ile czasu produkty były w koszyku - czas od otwarcia koszyka do zakupu
Metoda płatności	
Funding?	Raty, buy now pay later
Użyty kod promocyjny	
Lokalizacja dostawy	Miasto
metoda_dostawy	kurier, poczta polska etc.
numer_zamówienia	

table_visit_info

client_ID	
session_id	ID wizyty klienta na stronie
link	Źródło wizyty
czas_na_stronie	

Raw_session_events

event_id	
session_id	
client_id	
datetime	

Table_client_info

client_ID	
płeć_C	Klient przy dostawie wybiera Pan/Pani/Inne
wiek_N	
data_urodzenia	
Miasto	
wydatki_suma_M	Suma płatności w miesiącu
wydatki_suma_3M	Suma płatności z 3 miesięcy
wydatki_suma_6M	Suma płatności z 6 miesiący
wydatki_suma_12M	Suma płatności z roku
wydatki_suma_Total	Całkowita suma płatności
wydatki_srednia_M, 3M, 6M, 12M, Total	Średnia wartość zamówienia w danym miesiącu
zamówienia_suma_M, 3M, 6M, 12M, Total	Liczba wykonanych zamówień
zamowienia_srednia_M, 3M, 6M, 12M	Średnia liczba wykonanych zamówień w miesiącu
zwroty_suma_M, 3M, 6M, 12M, Total	Liczba zwróconych produktów w danym okresie
zwroty_srednia_M, 3M, 6M, 12M	Średnia liczba zwróconych produktów w danym okresie
aktywny_flaga	Flaga czy klient jest aktywny (wykonał zamówienie przez ostatni rok)
wizyty_na_stronie_suma_M, 3M, 6M, 12M, Total	Suma wizyt na stronie w danym okresie

wizyty_na_stronie_srednia_M, 3M, 6M, 12M	Średnia liczba wyzyt na stronie w danym okresie
dni_od_wizyty_n	Liczba dni od ostatniej wizyty na stronie
skincare_wydatki_suma_M, 3M, 6M, 12M, T	Suma wydatków na produkty z kategorii Skincare w danym okresie
skincare_liczba_suma_M, 3M, 6M, 12M, T	Suma zamówionych produktów z kategorii Skincare w danym okresie
kategoria_liczba/wydatki_suma/średnia_M, 3M, 6M, 12M	Analogiczne zmienne dla innych kategorii

Dict_product

product_ID	ID produktu w bazie danych
opis	Opis produktu
kategorie_list	Lista kategorii w których produkt się znajduje
nazwa_marki	Nazwa marki produktu
nazwa_serii	Nazwa serii produktu
cena_n	Aktualna cena produktu
średnia_ocena	Średnia ocena klientów
popularność	średnia miesięczna liczba zamówień
procentowa_ilość_zwrotów_M,3M	
liczba_sprzedanych_M,3M	
liczba_zwróconych_M,3M	
premium_flag	Czy jest to produkt premium?
waga_n	Waga produktu
czas_użycia_srednia	Średni czas na jaki produkt starczy klientowi
kolor	Kolor produktu
dostępnosc_cnt	Liczba dostępnych sztuk produktu

Dict_offers

promocja_ID	
product_ID	
typ_promocji	
ilość_użytych_promocji	
product_IDS	

Rel_product

produkt_1_ID	ID pierwszego produktu
produkt_2_ID	ID drugiego produktu
kupowane_razem_szansa	Wartość wskazująca częstotliwość kupowania wskazanych produktów razem
przejscie_szansa	Wartość wskazująca czy klient który oglądał produkt 1 kupił produkt 2

Raw_contact

client_ID	
promocja_ID	
data	data, godzina, etc
sukces_cat	0 - mail zignorowany 1 - mail otwarty 2 - link otworzony 3 - dokonany zakup
kanal_cat	Kanał informacji: email, SMS, telefon

Hist_tables_{Date}_{Name}

Wszystkie informacje w rzędzie, jak w aktualnej tablicy	
data_informacji	dzień z którego rząd pochodzi