Mikołaj Kita

1. Pomysł na rozwiązanie

Moim początkowym pomysłem na rozwiązanie miało być podejście łączące Content Filtering i 'odwrócony' Colaborative Filtering. Odwrócony Collaborative Filtering miał polegać na tym, że skoro z danych wejściowych nie możemy w jednoznaczny sposób wyznaczyć ID użytkownika, co za tym idzie poznać jego dotychczasowej historii zakupów i historii ocen, które wystawił przedmiotom i zastosować Colaborative Filtering, to należy odwrócić nasze podejście: wyznaczyć trzy ulubione kategorie dla wszystkich użytkowników, a następnie przefiltrować je dodatkowo o lokalizację. Wtedy otrzymamy zbiór użytkowników podobnych do naszego użytkownika wejściowego i będziemy mogli zobaczyć, które przedmioty cieszyły się największa popularnością oraz zostały najwyżej ocenione, dzięki czemu będziemy je mogli zaproponować użytkownikowi wejściowemu. Dla użytkownika niezalogowanego postanowiłem wyznaczyć zbiór przedmiotów, które przynoszą największe przychody z całego zbioru, mnożąc liczbę zamówień razy średnią cenę i sortując od największych wartości do najniższych. Moim uzasadnieniem takiego podejścia będzie logika biznesowa – Data Science ma pomóc przedsiębiorstwu maksymalizować zysk, więc wystawienie przedmiotów przynoszących największe przychody może pomóc zrealizować ten cel. Rozwiązanie przedstawione poniżej zostało lekko zmienione ze względu na braki w danych.

2. Przegląd danych

Na początku postanowiłem przejrzeć wszystkie dostępne dane w poszukiwaniu przydatnych rzeczy i ewentualnych problemów. Podczas przeglądu zostały podjęte następujące decyzje:

a. Brak wykorzystania zbioru 'olist_geolocation_dataset' oraz zbioru 'olist_sellers_dataset', ponieważ potrzebne dane o lokalizacji użytkownika są również w zbiorze 'olist_customer_dataset', a dane o sprzedawcy nie będą użyteczne w moim rozwiązaniu.
b. Ograniczenie danych z 'olist_orders_dataset' tylko do tych, które zostały dostarczone użytkownikowi, ponieważ uważam, że użytkownicy mogą obiektywnie ocenić tylko dostarczony produkt.
c. Część produktów nie ma przypisanej kategorii – usunąłem te obserwacje ze zbioru, ponieważ brakujące wartości przeszkodziłyby później w rozwiązaniu.
d. Zbiór danych zostanie ograniczony do wartości z ostatnich 13 miesięcy – wydaje mi się, że w ciągu dłuższego okresu może nastąpić zmiana w potrzebach konsumentów. Co prawda ten zbiór danych nie jest jakoś bardzo rozpięty w czasie, ale gdyby był, to myślę, że warto to zaznaczyć. Jeżeli chodzi o to, dlaczego wybrałem akurat 13 miesięcy, to moje rozumowanie jest następujące: jeżeli 27.08 jest w Brazylii świętem/specjalnym dniem (co próbowałem sprawdzić, ale nic nie znalazłem), to wydaje mi się, że dobrym pomysłem byłoby podwójnie reprezentować dane z tego okresu. Hipotetyczny przykład obrazujący moje rozumowanie: jeżeli w Brazylii zaczynałby się wtedy np. rok szkolny, to chcielibyśmy zaproponować użytkownikom podręczniki dla ich dzieci (które w hipotetycznym przykładzie świetnie sprzedają się właśnie przed rozpoczęciem roku szkolnego, ale w ciągu roku sprzedają się raczej słabo), a niekoniecznie np. świąteczne bombki, które są ogólnym bestsellerem, ale słabo sprzedają się w sierpniu.

3. Opis rozwiązania

Pierwszym krokiem w rozwiązaniu zadania jest zmergowanie potrzebnych danych w jeden DataFrame. Podczas złączania danych ograniczyłem niepotrzebne kolumny do minimum, starając się wykorzystywać tylko dane potrzebne do wykonania złączenia/potrzebne w późniejszych etapach.

Pierwszy problem pojawił się na etapie grupowania użytkowników: okazało się, że w zbiorze danych tylko 45 użytkowników kupiło coś z trzech kategorii lub więcej, natomiast 1340 z dwóch kategorii lub więcej. Przy łącznej liczbie kategorii produktów wynoszącej ponad 700 powoduje to brak pokrycia wszystkich kategorii – wręcz żadne kategorie nie są pokryte. Dlatego nie próbowałem dodatkowo filtrować dodatkowo ze względu na lokalizację – liczba użytkowników zmniejszyłaby się jeszcze bardziej do bardzo małych wartości. Postanowiłem rozszerzyć zbiór użytkowników, który uznaje za podobny do użytkownika wejściowego - wystarczy zrobić zakupy w dwóch z trzech kategorii podanych jako ulubione kategorie użytkownika wejściowego. Następnie wyciągnąłem średnią z recenzji dla przedmiotów zakupionych przez użytkowników w grupach podobnych do naszego użytkownika wejściowego. Niestety, zbiór użytkowników podobnych dalej był mniejszy od 10 dla użytkownika 1 i 2, dlatego postanowiłem go uzupełnić w następujący sposób: przefiltrować zbiór danych w poszukiwaniu użytkowników, którzy zrobili zakupy w jednej z trzech kategorii podanych przez użytkownika wejściowego oraz są z tego samego miasta i tego samego województwa („state”). Następnie zmniejszyłem zbiór danych poprzez usunięcie produktów, które zostały zamówione mniej razy niż wynosi 3-krotność średniej ilości zamówień dla wszystkich produktów. Zdecydowałem się na wartość 3-krotności średniej, ponieważ reguluje się ona sama w przypadku zastosowania tej samej funkcji dla zbioru danych, gdzie ilość zamówień liczona jest np. w tysiącach. Wtedy hard-coding wartości (np. jako 100 zamówień) mógłby przynieść skutki odwrotne do zamierzonych i nie spełniać swojej funkcji jako punkt odcięcia. Następnie postanowiłem wyznaczyć średnią ocen dla wszystkich produktów oraz posortować je malejąco. Produkty o najwyższej średniej ocenie z tej grupy zostają zmergowane z produktami ocenionymi przez użytkownikach z podobnymi dwoma kategoriami na trzy. Użytkownikowi wejściowemu rekomenduje tylko produkty o średniej ocenie wyższej od 4. Produkty od użytkowników, którzy mają podobne dwie na trzy kategorie mają zawsze priorytet nad produktami od użytkowników, którzy mają podobną jedną na trzy kategorie i mają takie same położenie.

Funkcje korzystają ze zmiennych zdefiniowanych i spreparowanych wcześniej, żeby za każdym razem nie powtarzać niepotrzebnie operacji, które mogą zostać wykonane tylko raz.

4. Wyniki:

Użytkownik 1: (cama_mesa_banho, papelaria, fashion_calcados), (sao paulo, SP):

0 013ee64977aaa6b2b25475095162e0e9
1 28a652ff04e43c1bc57937a9f8770f9b
2 736f1b87428f9cfe5f5184c4ac0fbe05
3 fcaab5d7f656094e49fbe4ee3a506658
4 f1c7f353075ce59d8a6f3cf58f419c9c
5 2a2d22ae30e026f1893083c8405ca522
6 8a443635fdf9759915c9be5be2e3b862
7 47cd48073d67f91f09cb5ef9496c920b

8 e03102efbc2229024c89be731f0aedcb
9 bb42f37fc3d9130e4a4339d24a47dd7c

Użytkownik 2: (esporte_lazer, moveis_decoracao, telefonia), (rio de janeiro, RJ):

0 4231002e80d2a25aed31d65b4b91f479
1 72de34cc9f1e580f4c11d830be654271
2 a54f350cdb1f303fe39221171d003852
3 f48eb5c2fde13ca63664f0bb05f55346
4 fa9cf28beaafe4f4bec59550f6c76481
5 a237de12bdf0bfe4fe220bae65a89731
6 44a34214a57dc373dcd80f54c919d006
7 764292b2b0f73f77a0272be03fdd45f3
8 e44f675b60b3a3a2453ec36421e06f0f
9 4a300735bc293723103db0d0c1bc1585

Użytkownik niezalogowany: (), ():

0 bb50f2e236e5eea0100680137654686c
1 6cdd53843498f92890544667809f1595
2 d6160fb7873f184099d9bc95e30376af
3 d1c427060a0f73f6b889a5c7c61f2ac4
4 99a4788cb24856965c36a24e339b6058
5 3dd2a17168ec895c781a9191c1e95ad7
6 25c38557cf793876c5abdd5931f922db
7 5f504b3a1c75b73d6151be81eb05bdc9
8 53b36df67ebb7c41585e8d54d6772e08
9 aca2eb7d00ea1a7b8ebd4e68314663af