Zd.4

# Dokonaj analizy koszykowej wyznaczając: a. Zbiory częste (min\_sup = częstotliwość 20% najczęstszych artykułów w zbiorze wszystkich transakcji) b. Reguły asocjacyjne (min\_sup = jak wyżej, min\_conf >= 0.7) Jeśli trzeba, dokonaj transformacji danych do wymaganego formatu.

Podobnie jak w poprzednich zadaniach wykorzystam Pythona oraz pandas, jednak w tym wypadku użyję również biblioteki mlxtend.

Na początku łącze oba arkusze w jeden dataframe oraz oczyszczam je ze zwrotów i transakcji niezrealizowanych.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Teraz podjąłem próbę przekształcenia danych w tablicę transakcji z wszystkimi produktami i pojawiły się pierwsze problemy.

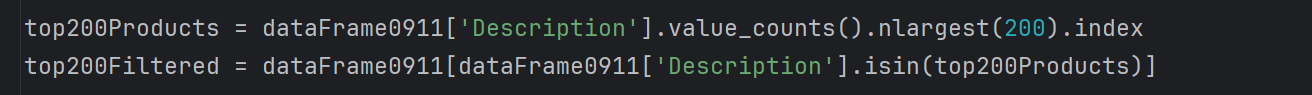
Obraz zawierający tekst, zrzut ekranu

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Po uruchomieniu zakomentowanej części kodu program wyświetlał błąd z informacją o niewystaczającej ilości pamięci. Miało to miejsce ponieważ było zbyt wiele unikalnych przedmiotów.

*numpy.\_core.\_exceptions.\_ArrayMemoryError: Unable to allocate 307. GiB for an array with shape (558096, 2, 36975) and data type int64*

Postanowiłem ograniczyć produkty do 200 najlepszych. Tak też zrobiłem, a następnie owe produkty poddałem filtracji.



Ponowiłem próbę utworzenia macierzy transakcji wraz z obliczeniem jak często dany produkt pojawia się w transakcjach oraz minimalny próg wsparcia:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Teraz wyszukuje częste zbiory produktów korzystając z algorytmu apriori.



Korzystam również z reguł asocjacyjnych. Działa ona w taki sposób że jeśli dany klient kupił produkt X, to z pewnym prawdopodobieństwem kupi również produkt Y. Dodatkowo ustawiam również próg, co oznacza że brane pod uwagę są wyłącznie reguły o co najmniej 70% pewności.

