Inteligentne

systemy decyzyjne

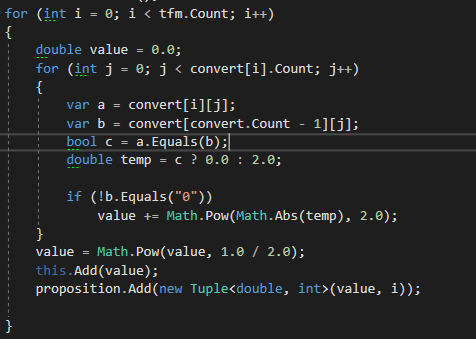
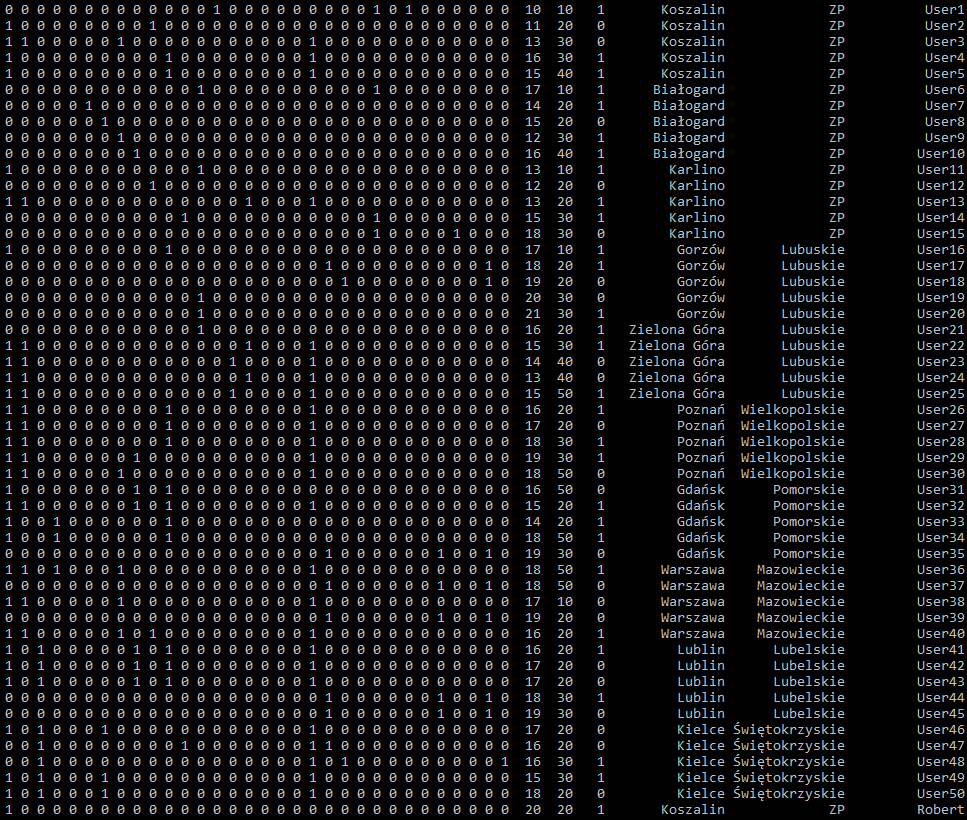
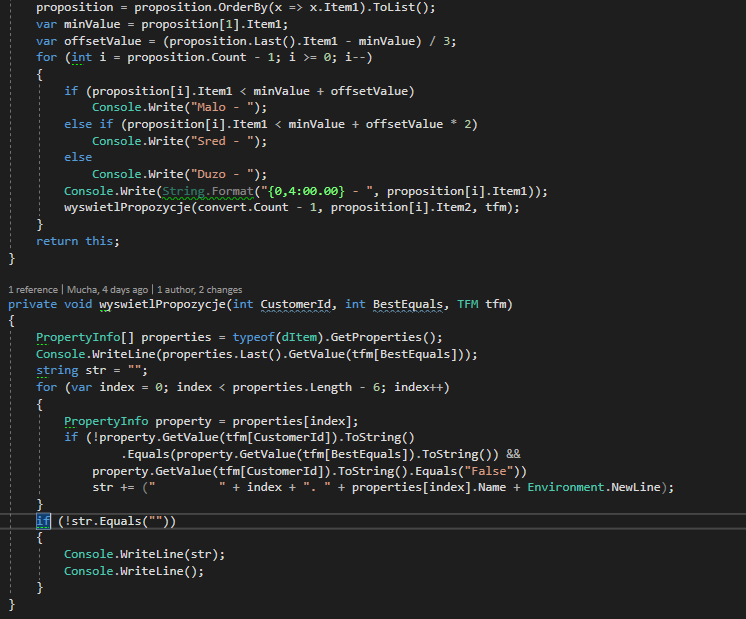
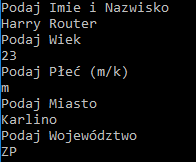
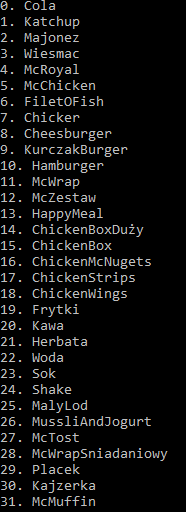
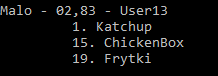
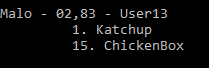
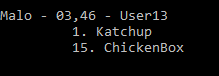
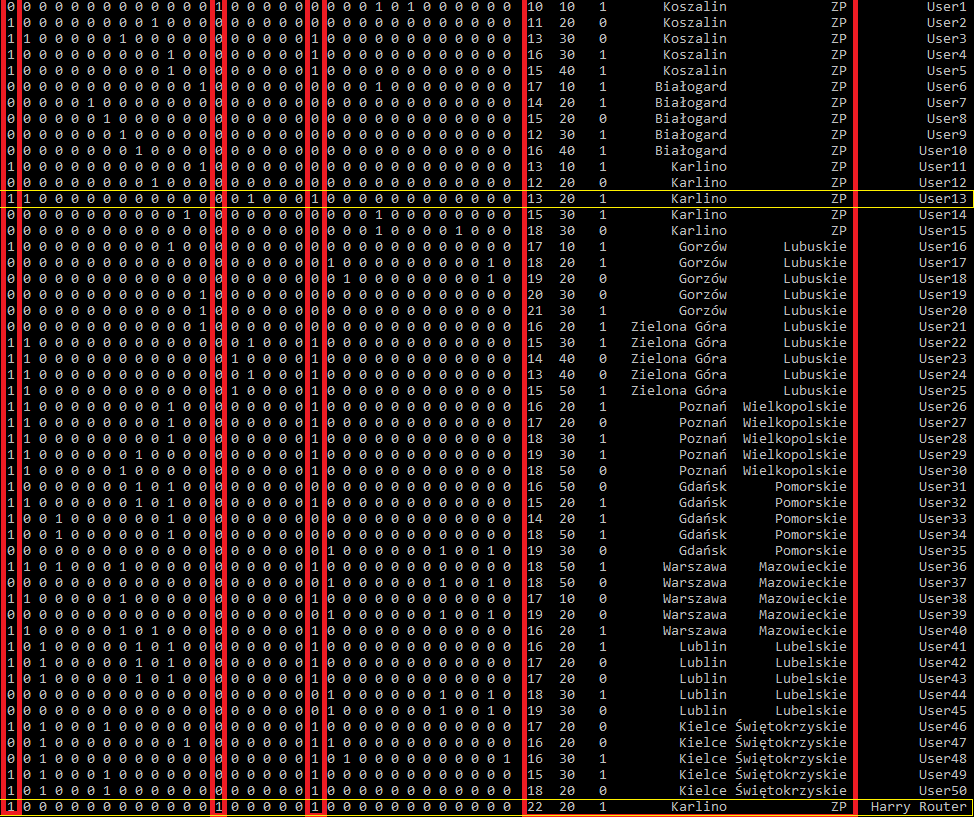
Temat: *System rekomendacyjny produktów typu FastFood*

Autor: Mucha Robert U-10459

1. **Istota problemu**  
   Powodem dla którego coraz częściej porusza się tematykę “Systemów rekomendacyjnych” jest fakt iż dobrze napisany system jest w stanie przynieść znaczne “dodatkowe” dochody dla firmy. Głównym założeniem jest znalezienie produktu podobnego lub powiązanego z już wybranym przez użytkownika. Przykładem może być kupno śruby gdzie prawidłowo działający system powinien zaproponować podkładki i nakrętki.   
   Jednakże zgodnie z tematem projektu użytkownik powinien dostawać kolejne propozycje produktów z oferty sklepu na podstawie zamówień składanych przez innych klientów.Oczywiście rekomendacje są uzależnione od danych na których pracują i w jaki sposób je przetwarzają. W tym wypadku system przechowuje dane o 50 klientach (nazywane w owych systemach termami)
2. **Budowa termu opisującego klienta**.  
   [nazwa - nr kolumny w tablicy TFM znajdującej sie ponizej]
   1. Godzina zakupu - 33
   2. Wiek klienta - 34
   3. Płeć klienta - 35
   4. Miasto - 36
   5. Województwo - 37
   6. Lista produktów zamówionych przez klienta - 32 produktów lecz nie ma przeciwwskazań by modyfikować ilość produktów ponieważ aplikacja bez większych problemów jest w stanie się przeskalować.  
      * cola - 1
      * katchup - 2
      * majonez - 3
      * wiesmac - 4
      * mcRoyal - 5
      * mcChicken - 6
      * filetOFish - 7
      * chicker - 8
      * cheesburger - 9
      * kurczakBurger - 10
      * hamburger - 11
      * mcWrap - 12
      * mcZestaw - 13
      * happyMeal - 14
      * chickenBoxDuży - 15
      * chickenBox - 16
      * chickenMcNugets - 17
      * chickenStrips - 18
      * chickenWings - 19
      * frytki - 20
      * kawa - 21
      * herbata - 22
      * woda - 23
      * sok - 24
      * shake - 25
      * malyLod - 26
      * mussliAndJogurt - 27
      * mcTost - 28
      * mcWrapSniadaniowy - 29
      * placek - 30
      * kajzerka - 31
      * mcMuffin - 32

Każdy nowy klient jest porównywany do owych 50 klientów.

Sposób zapisu informacji jest realizowany binarnie (0,1) ponieważ interesują nas produkty które wchodziły w skład zamówienia a nie ich ilość. W chwili gdybyśmy przechowywali dane w sposób ciągły czyli ilość produktów przy każdym zamówieniu na nasze rekomendacje miałyby wpływ sprzedane sztuki a nie ilość różnych zamówień.

1. **Realizacja projektu oraz metodyka obliczania**   
   Mając termy oraz dane na których będziemy mogli pracować, możemy przystąpić do stworzenia macierzy TFM która zawiera informacje o wszystkich użytkownikach.  
     
   Przykładowe wypełnianie macierzy TFM z poziomu kodu aplikacji.  
     
     
     
     
     
     
     
     
     
     
     
     
     
     
   A tak prezentuje się nasza macierz TFM.  
   Następnym krokiem jest wyznaczenie odległości pomiędzy naszym klientem a klientami których posiadamy już w bazie.  
   gdzie:  
   x to nasz klient   
   y to klient już istniejący w bazie.   
   Dodatkowo algorytm został ograniczony do liczenia kolumn które zostały wypełnione przez klienta w postaci odpowiedzi na pytania oraz składania zamowienia na kolejne produkty. (To co nie znajduje się w zamówieniu użytkownika nie jest brane pod uwagę przy analizie podobieństwa)  
     
   Nastepnie szukamy takie d(x,y) które jest najmniejsze ale rózne od 0 ponieważ d=0 oznacza dokladnie takiego samego użytkownika wiec nic on nie zasugeruje. Po odnalezieniu takiej osoby proponujemy naszemu klientowi to co zamówiła “najbliższa mu” osoba, a czego jeszcze klient nie zamówił.  
     
   Implementacja wzoru w kodzie:  
     
   oraz wyszukanie najbliższego klienta:  
     
   gdzie osoba pierwsza na liście jest najbliżej.  
     
   Pozostaje nam tylko już wyświetlić rekomendowane produkty. Podczas wyświetlania znajdują się określenia Mało/Średnio/Dużo i są wyznaczane na podstawie **offset**=(d(x,y).Max-d(x,y).Min)/3 w wyniku czego powstają grupy   
   Mało dla najbliższych klientów < d(x,y).Min() + offset  
   Średnio dla klientów < d(x,y).Min + offset \* 2  
   Dużo dla klientów < d(x,y).Min + offset \* 3  
     
   
2. **Działanie aplikacji - Studium Przypadków**  
     
   Jest to pierwsza część aplikacji gdzie podajemy swoje dane   
   a nastepnie wyświetlana jest lista produktów które można zamówić  
   i wyświetlana jest az do wciśniecia klawisza “esc”.  
   Po wyborze produktu (w tym wypadku 0.Cola) otrzymujemy informacje na temat najbliższego nam zamówienia skłądającego się z “słownego określenia odleglosci” - “wartości liczbowej odleglosci” - “nazwy użytkownika” -”oraz wypunktowanej listy produktów wraz z numerem produktu w menu”  
     
   na podstawie tego zamowienia zostały wybrane 19.frytki w wyniku czego najlepsza rekomendacja wygląda następująco.  
     
   Teraz żeby sprawdzić działanie zostanie wybrana opcja(13.HappyMeal) która nie jest sugerowana, a oto wynik:  
     
   Jak widać dalej najbliższy użytkownik sie nie zmienił ale za to odleglosc z 2.83 zmienił się na 3.46. A tak prezentuje się w tej chwili term naszego klienta oraz najbliższego uzytkownika - user13.  
     
     
   
3. **Wnioski**  
   Aplikacja została wykonana bez wiekszych przeszkód, a wszystkie zamierzone funkcjonalności zostały zaimplementowane. Rekomendacje działają poprawnie i sensownie w stosunku do posiadanej bazy wiedzy na której oparte są rekomendacje. Ważnym elementem jest fakt iż aplikacja wraz z rozszerzającą się bazą wiedzy będzie w stanie coraz lepiej rekomendować zamówienia klientom.  
   Realizacja owego projektu była bardzo ciekawym doświadczeniem ponieważ miałem okazję poznać coś szeroko rozpowszechnionego oraz wciąż popularnego na rynku. Jak się okazało nie jest to aż takie trudne, choć bez zdobytej wiedzy na zajęciach zabrałbym się pewnie od najtrudniejszej strony do rozwiązania owego problemu.