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]
   a. Godzina zakupu - 33
   b. Wiek klienta - 34
   c. Płeć klienta - 35
   d. Miasto - 36
   e. Województwo - 37
   f. 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.

3. **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.

```
temp.Add(new dItem(10, 10, true, "Koszalin", "ZP", "User" + (temp.Count  +1))
{ HappyMeal=true,Sok= true, MalyLod= true });
```

A tak prezentuje się nasza macierz TFM.

0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	10	10	1	Koszalin	ZP	User1
1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	20	0	Koszalin	ZP	User2
1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	13	30	0	Koszalin	ZP	User3
1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	16	30	1	Koszalin	ZP	User4
1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	15	40	1	Koszalin	ZP	User5
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	17	10	1	Białogard	ZP	User6
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	20	1	Białogard	ZP	User7
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	20	0	Białogard	ZP	User8
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	30	1	Białogard	ZP	User9
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	40	1	Białogard	ZP	User10
1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	10	1	Karlino	ZP	User11
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	20	0	Karlino	ZP	User12
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	13	20	1	Karlino	ZP	User13
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	15	30	1	Karlino	ZP	User14
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	18	30	0	Karlino	ZP	User15
1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	10	1	Gorzów	Lubuskie	User16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	18	20	1	Gorzów	Lubuskie	User17
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	19	20	0	Gorzów	Lubuskie	User18
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	30	0	Gorzów	Lubuskie	User19
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21	30	1	Gorzów	Lubuskie	User20
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	20	1	Zielona Góra	Lubuskie	User21
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	15	30	1	Zielona Góra	Lubuskie	User22
1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	14	40	0	Zielona Góra	Lubuskie	User23
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	13	40	0	Zielona Góra	Lubuskie	User24
1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	15	50	1	Zielona Góra	Lubuskie	User25
1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	16	20	1	Poznań	Wielkopolskie	User26
1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	17	20	0	Poznań	Wielkopolskie	User27
1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	18	30	1	Poznań	Wielkopolskie	User28
1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	19	30	1	Poznań	Wielkopolskie	User29
1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	18	50	0	Poznań	Wielkopolskie	User30
1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	50	0	Gdańsk	Pomorskie	User31
1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	15	20	1	Gdańsk	Pomorskie	User32
1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	20	1	Gdańsk	Pomorskie	User33
1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	50	1	Gdańsk	Pomorskie	User34
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	19	30	0	Gdańsk	Pomorskie	User35
1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	18	50	1	Warszawa	Mazowieckie	User36
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	18	50	0	Warszawa	Mazowieckie	User37
1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	17	10	0	Warszawa	Mazowieckie	User38
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	19	20	0	Warszawa	Mazowieckie	User39
1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	16	20	1	Warszawa	Mazowieckie	User40
1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	16	20	1	Lublin	Lubelskie	User41
1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	17	20	0	Lublin	Lubelskie	User42
1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	17	20	0	Lublin	Lubelskie	User43
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	18	30	1	Lublin	Lubelskie	User44
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	19	30	0	Lublin	Lubelskie	User45
1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	17	20	0	Kielce	Świętokrzyskie	User46
0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	16	20	0	Kielce	Świętokrzyskie	User47
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	16	30	1	Kielce	Świętokrzyskie	User48
1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	15	30	1	Kielce	Świętokrzyskie	User49
1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	18	20	0	Kielce	Świętokrzyskie	User50
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	20	1	Koszalin	ZP	Robert

Następnym krokiem jest wyznaczenie odległości pomiędzy naszym klientem a klientami których posiadamy już w bazie.

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2}$$

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:

```
for (int i = 0; i < tfm.Count; i++)
{
    double value = 0.0;
    for (int j = 0; j < convert[i].Count; j++)
    {
        var a = convert[i][j];
        var b = convert[convert.Count - 1][j];
        bool c = a.Equals(b);
        double temp = c ? 0.0 : 2.0;

        if (!b.Equals("0"))
            value += Math.Pow(Math.Abs(temp), 2.0);
    }
    value = Math.Pow(value, 1.0 / 2.0);
    this.Add(value);
    proposition.Add(new Tuple<double, int>(value, i));

}
```

oraz wyszukanie najbliższego klienta:

```
proposition = proposition.OrderBy(x => x.Item1).ToList();
```

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

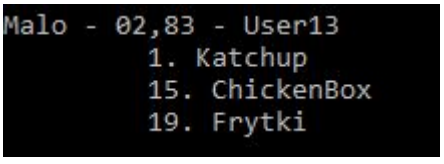
```
    proposition = proposition.OrderBy(x => x.Item1).ToList();
    var minValue = proposition[1].Item1;
    var offsetValue = (proposition.Last().Item1 - minValue) / 3;
    for (int i = proposition.Count - 1; i >= 0; i--)
    {
        if (proposition[i].Item1 < minValue + offsetValue)
            Console.Write("Malo - ");
        else if (proposition[i].Item1 < minValue + offsetValue * 2)
            Console.Write("Sred - ");
        else
            Console.Write("Duzo - ");
        Console.Write(String.Format("{0,4:00.00} - ", proposition[i].Item1));
        wyswietlPropozycje(convert.Count - 1, proposition[i].Item2, tfm);
    }
    return this;
}

1 reference | Mucha, 4 days ago | 1 author, 2 changes
private void wyswietlPropozycje(int CustomerId, int BestEquals, TFM tfm)
{
    PropertyInfo[] properties = typeof(dItem).GetProperties();
    Console.WriteLine(properties.Last().GetValue(tfm[BestEquals]));
    string str = "";
    for (var index = 0; index < properties.Length - 6; index++)
    {
        PropertyInfo property = properties[index];
        if (!property.GetValue(tfm[CustomerId]).ToString()
                .Equals(property.GetValue(tfm[BestEquals]).ToString()) &&
            property.GetValue(tfm[CustomerId]).ToString().Equals("False"))
            str += ("         " + index + ". " + properties[index].Name + Environment.NewLine);
    }
    if (!str.Equals(""))
    {
        Console.WriteLine(str);
        Console.WriteLine();
    }
}
```

4. **Działanie aplikacji - Studium Przypadków**

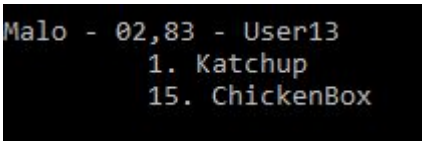
```
Podaj Imie i Nazwisko
Harry Router
Podaj Wiek
23
Podaj Płeć (m/k)
m
Podaj Miasto
Karlino
Podaj Województwo
ZP
```

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 składającego się z “słownego określenia odleglosci” - “wartości liczbowej odleglosci” - “nazwy użytkownika” -”oraz wypunktowanej listy produktów wraz z numerem produktu w menu”



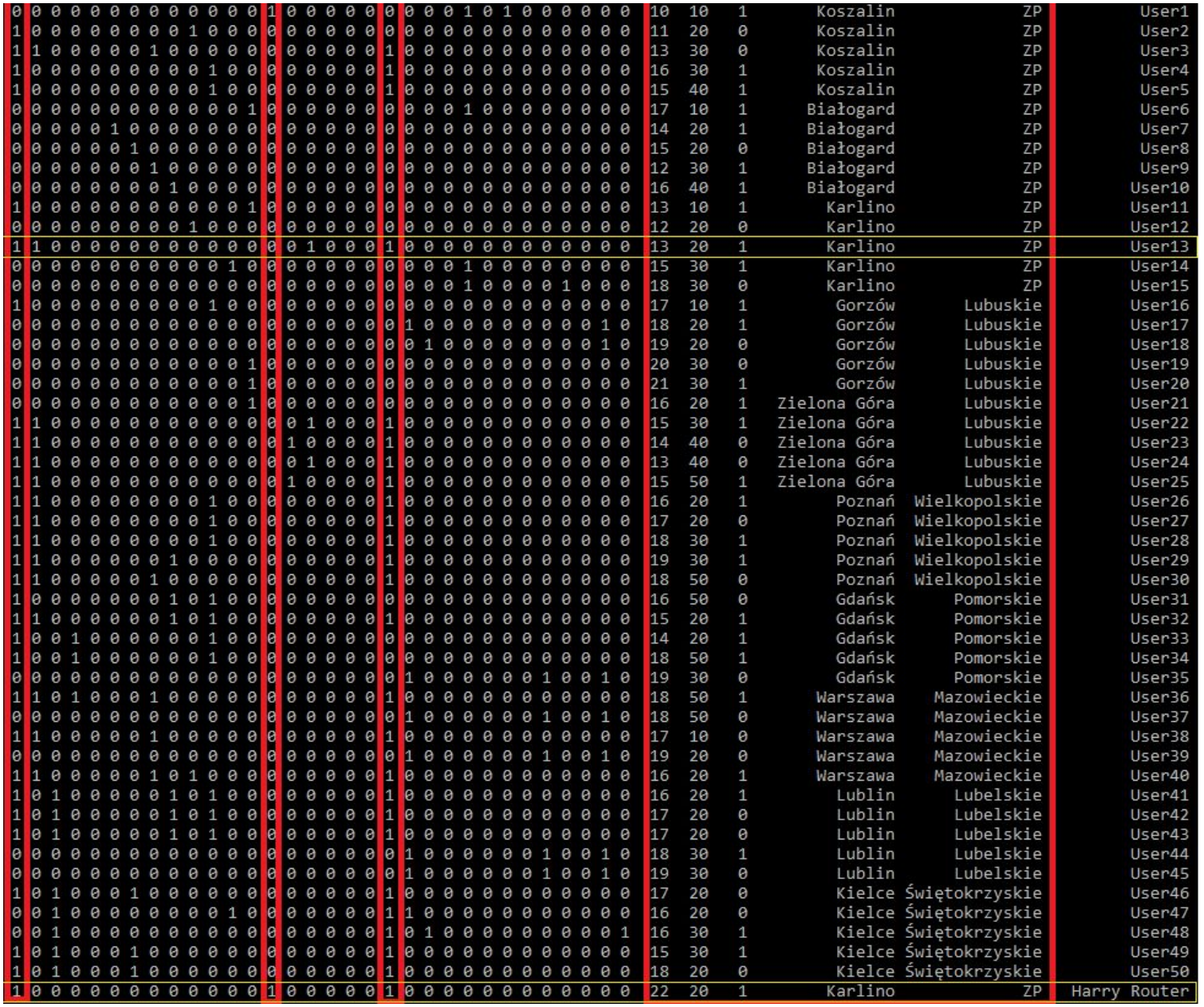
na podstawie tego zamowienia zostały wybrane 19.frytki w wyniku czego najlepsza rekomendacja wygląda następująco.



Teraz żeby sprawdzić działanie zostanie wybrana opcja(13.HappyMeal) która nie jest sugerowana, a oto wynik:

```
Malo - 03,46 - User13
        1. Katchup
        15. ChickenBox
```

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.



10	10	1	Koszalin	ZP	User1
11	20	0	Koszalin	ZP	User2
13	30	0	Koszalin	ZP	User3
16	30	1	Koszalin	ZP	User4
15	40	1	Koszalin	ZP	User5
17	10	1	Białogard	ZP	User6
14	20	1	Białogard	ZP	User7
15	20	0	Białogard	ZP	User8
12	30	1	Białogard	ZP	User9
16	40	1	Białogard	ZP	User10
13	10	1	Karlino	ZP	User11
12	20	0	Karlino	ZP	User12
13	20	1	Karlino	ZP	User13
15	30	1	Karlino	ZP	User14
18	30	0	Karlino	ZP	User15
17	10	1	Gorzów	Lubuskie	User16
18	20	1	Gorzów	Lubuskie	User17
19	20	0	Gorzów	Lubuskie	User18
20	30	0	Gorzów	Lubuskie	User19
21	30	1	Gorzów	Lubuskie	User20
16	20	1	Zielona Góra	Lubuskie	User21
15	30	1	Zielona Góra	Lubuskie	User22
14	40	0	Zielona Góra	Lubuskie	User23
13	40	0	Zielona Góra	Lubuskie	User24
15	50	1	Zielona Góra	Lubuskie	User25
16	20	1	Poznań	Wielkopolskie	User26
17	20	0	Poznań	Wielkopolskie	User27
18	30	1	Poznań	Wielkopolskie	User28
19	30	1	Poznań	Wielkopolskie	User29
18	50	0	Poznań	Wielkopolskie	User30
16	50	0	Gdańsk	Pomorskie	User31
15	20	1	Gdańsk	Pomorskie	User32
14	20	1	Gdańsk	Pomorskie	User33
18	50	1	Gdańsk	Pomorskie	User34
19	30	0	Gdańsk	Pomorskie	User35
18	50	1	Warszawa	Mazowieckie	User36
18	50	0	Warszawa	Mazowieckie	User37
17	10	0	Warszawa	Mazowieckie	User38
19	20	0	Warszawa	Mazowieckie	User39
16	20	1	Warszawa	Mazowieckie	User40
16	20	1	Lublin	Lubelskie	User41
17	20	0	Lublin	Lubelskie	User42
17	20	0	Lublin	Lubelskie	User43
18	30	1	Lublin	Lubelskie	User44
19	30	0	Lublin	Lubelskie	User45
17	20	0	Kielce	Świętokrzyskie	User46
16	20	0	Kielce	Świętokrzyskie	User47
16	30	1	Kielce	Świętokrzyskie	User48
15	30	1	Kielce	Świętokrzyskie	User49
18	20	0	Kielce	Świętokrzyskie	User50
22	20	1	Karlino	ZP	Harry Router

5. **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.