

# Inteligentne systemy decyzyjne

Temat: *System rekomendacyjny  
produktów pszczelarskich*

## 1. Istota problemu

Obecnie coraz częściej wykorzystuje się systemy rekomendacyjne. Powodem takiego stanu rzeczy jest ich niezwykła skuteczność której efektem jest na przykład zwiększenie przychodów sklepu poprzez trafne propozycje produktów. Głównym założeniem systemu rekomendacyjnego stworzenie potrzeby na produkt w chwili jego wyświetlenia poprzez np „uzupełnienie zamówienia”. Najlepszym przykładem może być kupno gwoździ. Użytkownik taki może potrzebować młotka do ich wbicia czy desek które będzie przybijał. Jeżeli system to przewidzi można go uznać za skuteczny.

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 system pracuje 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 się poniżej]

- a. Godzina zakupu - 21
- b. Wiek klienta - 22
- c. Płeć klienta - 23
- d. Miasto - 24
- e. Województwo - 25
- f. Lista produktów zamówionych przez klienta - 20 produktów lecz nie ma przeciwwskazań by modyfikować ilość produktów ponieważ aplikacja bez większych problemów jest w stanie się przeskalować.

- o PierścieńDoPowałki - 1
- o SitkoPająkDoPowałki - 2
- o Przegonka8-wylotowa - 3
- o EtykietyNaMiódGryczanyE217-100Szt - 4
- o PodkarmiaczkaPowałkowa4l - 5
- o PokarmCiastoFondant15kgDiamant - 5
- o PodkurzaczTypuDadantNierdzewnyDuży - 6
- o Książka„ŁatwePszczelarstwo”(drGerhardLiebig) - 7
- o RękawicePszczelarskieZOwczejSkóryDługieXL - 8
- o MiotełkaPasiecznaDwurzędowaZKońskiegoWłosa - 9
- o Fajkadołapaniamatekpszczelich - 10
- o BluzaPszczelarskaZKapeluszemRozpinanaXL - 11
- o KapeluszPszczelarskiBIAŁY - 12
- o WiaderkoOdstojnikNaMiódZZaworem33lBiałe - 13
- o DrutDoRamek-StalNierdzewna-025Kg - 14
- o UchwytDoRamekNierdzewny - 15
- o DwufunkcyjnyNóżPasiecznyZDłutem - 16
- o RamkaWielkopolskaHoffmanowska-10Szt - 17
- o UniwersalnySkrobakPasieczny - 18
- o NakrętkaNaSłoikZMiodemND47-10Szt - 19
- o UlikOdkładowyWielkopolski - 20



#### 4. Wyznaczanie odległości euklidesowej

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

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

gdzie:

x to nasz klient

y to klient już istniejący w bazie.

Dodatkowo algorytm został ograniczony do liczenia kolumn które zostały wypełnione przez klienta. To oznacza że w procesie liczenia biorą udział tylko kolumny w których występują wartości '1' w wierszu klienta.

Następnie szukamy takie d(x,y) które jest najmniejsze ale różne od 0 ponieważ d=0 oznacza dokładnie takiego samego użytkownika więc 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 najbliższej.

Pozostaje nam tylko już wyświetlić rekomendowane produkty. Podczas wyświetlania znajdują się określenia Mało/Średnio/Dużo i są wyznaczone na podstawie  $\text{offset} = (d(x,y) \cdot \text{Max} - d(x,y) \cdot \text{Min}) / 3$  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  $< d(x,y).Min + offset * 3$

## 5. Działanie aplikacji - Studium Przypadków

Jest to pierwsza część aplikacji gdzie podajemy swoje dane a następnie wyświetlana jest lista produktów które można zamówić i wyświetlana jest aż do wciśnięcia klawisza "esc".

Po wyborze produktu (w tym wypadku 0.PierscienDoPowalki) 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”

```
Malo - 03,46 - User1
    1. SitkoPajakDoPowalki
    4. PodkarmiaczkaPowalkowa
```

Na podstawie tego zamówienia zostało wybrane 1.Sitko Pająk Do Powalki w wyniku czego najlepsza rekomendacja wygląda następująco.

Malo - 03,46 - User1  
4. PodkarmiaczkaPowalkowa

Teraz żeby sprawdzić działanie zostanie wybrana opcja(6.EtykietaNaMiódGryczany) która nie jest sugerowana, a oto wynik:

Malo - 04,00 - User1  
4. PodkarmiaczkaPowalkowa

Jak widać dalej najbliższy użytkownik się nie zmienił ale za to odległość z 3,46 zmieniła się na 4,00. A tak prezentuje się w tej chwili term naszego klienta oraz najbliższego użytkownika - user1.

[illegible]

## 6. Wnioski

Aplikacja została wykonana bez większych przeszkód, a wszystkie zamierzone funkcjonalności zostały zaimplementowane. Rekomendacje działają poprawnie w zgodzie z jej bazą wiedzy. Należy pamiętać iż system taki wraz z rosnącą ilością zamówień może coraz lepiej rekomendować produkty.

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.