



Tema 4 Multimi Rough

Sisteme Semantice

Draghici Andreea-Maria
Inginerie Software
IS 2.1

Tema 4 Multimi Rough

Contents

1. Tabel de informații	2
1.1 Proiectarea unui Dialog cu Utilizatorul pentru a Selecta Entitățile din Tabel Conform Cerințelor Utilizatorului (Mulțimea X)	2
1.2 Proiectarea Aproximărilor Mulțimii X Folosind Atributele din Tabel	3
2.1 Demo aplicativ	4
Dialogul cu utilizatorul (se citește de la tastatură):	5
Rezultatele filtrării	5
Aplicarea Teoriei Mulțimilor Rough	6
2. Concluzii finale	6
3. Cod sursă	7

1. Tabel de informații

ID	Produs	Categorie	Pret (\$)	Stoc	Vandut online	Rating mediu	An lansare
1	Telefon mobil Samsung	Smartphone	300	DA	DA	4.7	2021
2	Tabletă Samsung	Tabletă	450	NU	DA	4.0	2020
3	Laptop Asus	Laptop	800	DA	NU	4.8	2022
4	Smartwatch	Ceas	200	DA	DA	4.0	2023
5	Consolă	Gaming	350	NU	NU	4.5	2023
6	Laptop Dell	Laptop	750	DA	DA	4.3	2022
7	Telefon iPhone	Smartphone	350	DA	DA	5.0	2024
9	Camera Canon	Aparat Compact	400	DA	DA	4.3	2021

Detalii ale Setului de Date:

1.1 Proiectarea unui Dialog cu Utilizatorul pentru a Selecta Entitățile din Tabel Conform Cerințelor Utilizatorului (Mulțimea X)

Codul inițiază un dialog cu utilizatorul prin solicitarea inputurilor pentru diferite criterii:

- Categorie (ex. Laptop, Smartphone)
- Prețul maxim dorit
- Necesitatea ca produsul să fie în stoc
- Ratingul minim dorit
- Dacă produsul trebuie să fie vândut online
- Anul minim de lansare

```
criterii = {  
    'categorie' : input("Introduceti categoria produsului (ex. Laptop, Smartphone): "),  
    'pret_max' : int(input("Introduceti pretul maxim dorit: ")),  
    'stoc' : input("Produsul trebuie să fie în stoc? (DA/NU): "),  
    'rating_min' : float(input("Introduceti ratingul minim dorit: ")),  
    'vandut_online' : input("Produsul trebuie să fie vândut online? (DA/NU): "),  
    'an_min' : int(input("Introduceti anul minim de lansare: "))  
}
```

Tema 4 Multimi Rough

Aceste inputuri sunt folosite pentru a stabili setul de criterii care definește mulțimea X. Astfel, mulțimea X este definită ca fiind ansamblul tuturor produselor din tabel care îndeplinesc criteriile specificate de utilizator.

1.2 Proiectarea Aproximărilor Mulțimii X Folosind Atributele din Tabel

Voi defini mulțimea X ca fiind setul de produse care îndeplinesc anumite criterii specificate de utilizator (e.g., categoria, preț maxim, disponibilitate în stoc, rating minim).

Atributul U reprezintă universul de produse, iar A setul de atribute pentru fiecare produs, cum ar fi preț, stoc, rating, etc.

Aproximarea Inferioară $\text{aprox_inferioara}(X)$

Aproximarea inferioară a mulțimii X este definită ca mulțimea tuturor produselor care îndeplinesc **toate** criteriile specificate:

$$\text{aprox_inferioara}(X) = \{ x \in U \mid \forall a \in A, \text{ condițiile pe } a \text{ sunt îndeplinite pentru } a \text{ fi în } x \}$$

Aceasta include produsele pentru care avem certitudinea că respectă toate condițiile.

Aproximarea Superioară $\text{aprox_superioara}(X)$

Aproximarea superioară a mulțimii X include produsele care îndeplinesc **cel puțin una** dintre condițiile specificate:

$$\text{aprox_superioara}(X) = \{ x \in U \mid \exists a \in A, \text{ cel puțin un criteriu pe } a \text{ este îndeplinit pentru } a \text{ fi în } x \}$$

Aceasta reprezintă posibilitatea ca produsele să fie acceptabile conform criteriilor, dar cu o certitudine mai mică.

Regiunea de Frontieră BND (X)

Regiunea de frontieră include produsele pentru care există incertitudine între a fi sau nu în X:

$$\text{BND}(X) = \text{aprox_superioara}(X) - \text{aprox_inferioara}(X)$$

Aceste produse nu pot fi clasificate cu certitudine folosind doar informațiile disponibile.

Exemplu de Aplicare

Dacă utilizatorul caută un "Laptop" cu prețul maxim de 800\$, care este în stoc și are un rating de minim 4.5, setările pentru X ar putea fi:

Tema 4 Multimi Rough

- Categorie = Laptop
- Preț (\$) ≤ 800
- Stoc = DA
- Rating mediu ≥ 4.5

În acest caz, utilizând setul de date:

- aprox_inferioara (X) va conține "Laptop Asus" și "Laptop Dell" dacă ambele îndeplinesc toate condițiile.
- aprox_superioara (X) va include orice laptop cu oricare dintre aceste condiții îndeplinite parțial sau complet.
- BND (X) va cuprinde laptopurile pentru care informațiile nu sunt suficiente pentru a determina dacă îndeplinesc absolut toate criteriile.

2.1 Demo aplicativ

Codul inițializează un DataFrame pandas cu un set de date și folosește criterii specificate de utilizator pentru a calcula trei tipuri de rezultate:

1. **Aproximarea inferioară** - care reprezintă produsele ce îndeplinesc toate criteriile specificate. Aceasta este calculată folosind un operator logic "și" între toate condițiile (folosind operatorul & în Python).
2. **Aproximarea superioară** - care include produse ce îndeplinesc oricare dintre condițiile specificate. Acesta este calculat folosind un operator logic "sau" între condiții (folosind operatorul | în Python).
3. **Regiunea de frontieră** - care reprezintă diferența dintre aproximarea superioară și cea inferioară, indicând elementele pentru care există incertitudine în clasificare. Aceasta este calculată folosind diferența între indicii produselor din cele două mulțimi.

```
def filtreaza_produce(df, criterii):  
    """Filtreaza produsele folosind criteriile specificate."""  
    conditii = {  
        'Categorie': df['Categorie'] == criterii['categorie'],  
        'Preț ($)': df['Preț ($)'] <= criterii['pret_max'],  
        'Stoc': df['Stoc'] == criterii['stoc'],  
        'Rating mediu': df['Rating mediu'] >= criterii['rating_min'],  
        'Vândut online': df['Vândut online'] == criterii['vandut_online'],  
        'An lansare': df['An lansare'] >= criterii['an_min']  
    }  
  
    # C filtru pentru aproximarea inferioară și superioară  
    filtru_inferior = pd.Series([True] * len(df))  
    filtru_superior = pd.Series([False] * len(df))  
  
    for key, value in conditii.items():  
        filtru_inferior &= value  
        filtru_superior |= value  
  
    aproximare_inferioara = df[filtru_inferior]  
    aproximare_superioara = df[filtru_superior]  
    regiune_frontiera = aproximare_superioara[~aproximare_superioara.index.isin(aproximare_inferioara.index)]  
  
    return aproximare_inferioara, aproximare_superioara, regiune_frontiera
```

Tema 4 Multimi Rough

Dialogul cu utilizatorul (se citește de la tastatură):

1. **Categorie:** Laptop - Utilizatorul este interesat de produse din categoria "Laptop".
2. **Preț maxim dorit:** 800 - Produsele trebuie să aibă un preț de până la 800 de dolari.
3. **Stoc:** DA - Produsele trebuie să fie în stoc.
4. **Rating minim dorit:** 4.3 - Ratingul mediu al produselor trebuie să fie cel puțin 4.3.
5. **Vândut online:** DA - Produsele trebuie să fie disponibile pentru vânzare online.
6. **An minim de lansare:** 2020 - Produsele trebuie să fie lansate începând cu anul 2020

Rezultatele filtrării

- **Aproximare inferioară:** Contine produsele care îndeplinesc toate criteriile specificate, fără ambiguitate. În cazul meu, doar "Laptop Dell" se potrivește perfect tuturor criteriilor.
- **Aproximare superioară:** Include produsele care îndeplinesc cel puțin unul dintre criteriile specificate, indicând o incertitudine mai mare. Practic, fiecare produs din lista inițială apare aici, deoarece toate îndeplinesc cel puțin un criteriu (de exemplu, fiecare are un rating peste 4.0, sunt vândute online, etc.).
- **Regiune de frontieră:** Conține produsele pentru care există incertitudine între a fi sau nu complet adecvate conform tuturor criteriilor. În acest caz, regiunea de frontieră include toate produsele care apar în aproximarea superioară, dar nu sunt prezente în cea inferioară. Practic, orice produs care nu îndeplinește toate criteriile, dar îndeplinește unele, se află în această categorie.

```
D:\Data\_DevTools\Conda\envs\Tema4\python.exe D:\Data\Work\Tema4\src\rezolvare_execitiu1.py
Introduceți categoria produsului (ex. Laptop, Smartphone): Laptop
Introduceți prețul maxim dorit: 800
Produsul trebuie să fie în stoc? (DA/NU): DA
Introduceți ratingul minim dorit: 4.3
Produsul trebuie să fie vândut online? (DA/NU): DA
Introduceți anul minim de lansare: 2020

Aproximare inferioară:
  ID      Produs  Categorie  ...  Vândut online  Rating mediu  An lansare
5   6   Laptop Dell   Laptop  ...           DA           4.3       2022

[1 rows x 8 columns]

Aproximare superioară:
  ID      Produs  Categorie  ...  Rating mediu  An lansare
0   1   Telefon mobil Samsung  ...           4.7       2021
1   2   Tabletă Samsung  ...           4.0       2020
2   3   Laptop Asus  ...           4.8       2022
3   4   Smartwatch  ...           4.0       2023
4   5   Consolă Gaming  ...           4.5       2023
5   6   Laptop Dell  ...           4.3       2022
6   7   Telefon iPhone  ...           5.0       2024
7   9   Camera Canon  ...           4.3       2021

[8 rows x 8 columns]
```

Tema 4 Multimi Rough

```
Regiune de frontieră:
  ID      Produs  ... Rating mediu  An lansare
0  1  Telefon mobil Samsung  ...      4.7      2021
1  2      Tabletă Samsung  ...      4.0      2020
2  3      Laptop Asus    ...      4.8      2022
3  4      Smartwatch    ...      4.0      2023
4  5      Consolă Gaming  ...      4.5      2023
6  7      Telefon iPhone  ...      5.0      2024
7  9      Camera Canon   ...      4.3      2021

[7 rows x 8 columns]

Process finished with exit code 0
```

Aplicarea Teoriei Multimilor Rough

Teoria multimilor Rough în acest context ajută la gestionarea incertitudinilor în setul de date:

- **Aproximarea inferioară** ne dă certitudinea că produsele listate aici îndeplinesc toate condițiile impuse.
- **Aproximarea superioară** ne arată posibilitatea ca produsele să fie suficient de bune, chiar dacă nu îndeplinesc fiecare condiție complet.
- **Regiunea de frontieră** este utilă pentru a identifica produsele despre care nu avem suficiente informații pentru a decide dacă îndeplinesc complet toate cerințele.

2. Concluzii finale

- Dialogul construit în script permite utilizatorilor să specifice criterii detaliate, care sunt folosite direct pentru a defini mulțimea X de produse de interes. Această abordare îmbunătățește experiența utilizatorului și oferă rezultate personalizate.
- Folosirea Aproximărilor Rough:
 - **Aproximarea Inferioară:** Oferă o listă de produse care îndeplinesc toate criteriile, garantând satisfacția cerințelor utilizatorului fără ambiguitate.
 - **Aproximarea Superioară:** Extinde opțiunile utilizatorului la produse care pot îndeplini cel puțin una dintre condiții, mărinnd astfel diversitatea opțiunilor disponibile.
 - **Regiunea de Frontieră:** Identifică produsele pentru care există incertitudini legate de îndeplinirea criteriilor, informând utilizatorul despre posibilele riscuri sau neclarități.

3. Cod sursă

```
import pandas as pd
```

```
def filtreaza_produce(df, criterii):
```

```
    """Filtreaza produsele folosind criteriile specificate."""
```

```
    conditii = {
```

```
        'Categorie': df['Categorie'] == criterii['categorie'],
```

```
        'Preț ($)': df['Preț ($)'] <= criterii['pret_max'],
```

```
        'Stoc': df['Stoc'] == criterii['stoc'],
```

```
        'Rating mediu': df['Rating mediu'] >= criterii['rating_min'],
```

```
        'Vândut online': df['Vândut online'] == criterii['vandut_online'],
```

```
        'An lansare': df['An lansare'] >= criterii['an_min']
```

```
    }
```

```
    # C filtru pentru aproximarea inferioară și superioară
```

```
    filtru_inferior = pd.Series([True] * len(df))
```

```
    filtru_superior = pd.Series([False] * len(df))
```

```
    for key, value in conditii.items():
```

```
        filtru_inferior &= value
```

```
        filtru_superior |= value
```

```
    aproximare_inferioara = df[filtru_inferior]
```

```
    aproximare_superioara = df[filtru_superior]
```

```
    regiune_frontiera
```

```
=
```

```
    aproximare_superioara[~aproximare_superioara.index.isin(aproximare_inferioara.index)]
```

```
    return aproximare_inferioara, aproximare_superioara, regiune_frontiera
```

```
def afiseaza_rezultate(aprox_inferioara, aprox_superioara, reg_frontiera):
```

```
    """Afișează rezultatele filtrării."""
```

```
    print("Aproximare inferioară:")
```

```
    print(aprox_inferioara)
```

```
    print("\nAproximare superioară:")
```

```
    print(aprox_superioara)
```

```
    print("\nRegiune de frontieră:")
```

```
    print(reg_frontiera)
```

```
def main():
```

```
    data = {
```

```
        'ID': [1, 2, 3, 4, 5, 6, 7, 9],
```

```
        'Produs': ['Telefon mobil Samsung', 'Tabletă Samsung', 'Laptop Asus', 'Smartwatch', 'Consolă Gaming',
```

```
                  'Laptop Dell', 'Telefon iPhone', 'Camera Canon'],
```

```
        'Categorie': ['Smartphone', 'Tabletă', 'Laptop', 'Ceas', 'Gaming', 'Laptop', 'Smartphone', 'Aparat Compact'],
```

```
        'Preț ($)': [300, 450, 800, 200, 350, 750, 350, 400],
```


Tema 4 Multimi Rough

```
'Stoc' : ['DA', 'NU', 'DA', 'NU', 'NU', 'DA', 'DA', 'DA'],
'Vândut online' : ['DA', 'DA', 'NU', 'DA', 'NU', 'DA', 'DA', 'DA'],
'Rating mediu' : [4.7, 4.0, 4.8, 4.0, 4.5, 4.3, 5.0, 4.3],
'An lansare' : [2021, 2020, 2022, 2023, 2023, 2022, 2024, 2021]
}

df = pd.DataFrame(data)

criterii = {
    'categorie' : input("Introduceți categoria produsului (ex. Laptop, Smartphone): "),
    'pret_max' : int(input("Introduceți prețul maxim dorit: ")),
    'stoc' : input("Produsul trebuie să fie în stoc? (DA/NU): "),
    'rating_min' : float(input("Introduceți ratingul minim dorit: ")),
    'vandut_online' : input("Produsul trebuie să fie vândut online? (DA/NU): "),
    'an_min' : int(input("Introduceți anul minim de lansare: "))
}

# Aplică filtrarea
aprox_inferioara, aprox_superioara, reg_frontiera = filtreaza_produce(df, criterii)

# Afișează rezultatele
afiseaza_rezultate(aprox_inferioara, aprox_superioara, reg_frontiera)

if __name__ == "__main__":
    main()
```