

Recommender – Dokumentacija

(Content-based recommender za MenuItem u RestoraNow)

Datum: 08.09.2025.

1) Sažetak (Šta radi sistem)

Sistem preporuke je **content-based**: za svaku stavku menija (MenuItem) gradi se **vektor značajki** iz tekstualnih polja (*Name*, *CategoryName*, *Description*) pomoću **ML.NET** featurizacije teksta.

Za korisnika se pravi **profil** na osnovu njegovih **recenzija** i **narudžbi**, a zatim se dostupne stavke rangiraju po **kosinusnoj sličnosti** sa tim profilom.

Ako nema historije ili model nije izgrađen, radi **cold-start fallback**:
specijali dana → **najbolje ocijenjene** → **ostale dostupne**.

2) Arhitektura (komponente i lifecycle)

- **Klasa:** MenuRecommendationService : IMenuRecommendationService
 - Zavisnost: ApplicationDbContext (EF Core).
 - Statički cache:
 - MLContext _ml
 - Dictionary<int, float[]> _itemVectors (mapiranje MenuItemId → vektor).
 - **Thread-safe** inicijalizacija modela uz lock (_sync).
 - **Lijena inicijalizacija**: model i vektori se grade **prvi put** kad se servis kreira/pozove.

Napomena o DI životnom ciklusu: zbog EF Core DbContext (obično **Scoped**), servis registrajte kao **Scoped** (ne kao Singleton). Statički cache i dalje dijeli vektore među instancama.

3) Kako radi (korak-po-korak)

1. **Izgradnja vektora za sve stavke**
 - Učita se lista MenuItem (+ Category), za svaku se spoje tekstovi: Name, CategoryName, Description.
 - ML.NET pipeline (vidi §4) izgradi **Features** i normalizira ih.
 - Rezultat se spremi u _itemVectors[MenuItemId] = float[].
2. **Korisničke interakcije** → **težine**

- **Recenzije (MenuItemReview)**: za svaki ocijenjeni item
Težina = $2 \times (\text{Rating} / 5) \rightarrow$ raspon **0.2–1.0 × 2**.
 - **Narudžbe (OrderItem)**: težina = **Quantity** (min 1).
 - Ako nema interakcija \rightarrow **cold-start**.
3. **Profil korisnika**
- Izračuna se **ponderisani prosjek** vektora stavki sa dobijenim težinama.
4. **Kandidati i rangiranje**
- Kandidati = sve **dostupne** stavke (`IsAvailable == true`) **koje korisnik nije već ocjenjivao/naručivao** (isključujemo viđene).
 - Za svakog kandidata računa se **kosinusna sličnost** sa profilom.
 - Vraća se **top N** (parametar take, podrazumijevano 10).
5. **Cold-start fallback**
- Ako nema profila ili nema kandidata:
 - a) `IsSpecialOfTheDay == true` (ograniči na take)
 - b) Top-ocijenjene po prosjeku i broju recenzija (do popune)
 - c) Ostale dostupne (do popune).

4) ML.NET pipeline (vektORIZACIJA)

- `FeaturizeText("NameFeats", Name)`
- `FeaturizeText("CatFeats", Category)`
- `FeaturizeText("DescFeats", Description)`
- `Concatenate("Features", "NameFeats", "CatFeats", "DescFeats")`
- `NormalizeLpNorm("Features")`

Pipeline se **fit-uje jednom**, transformira sve stavke i rezultat drži u memoriji (`_itemVectors`).

5) Pretpostavke o podacima (EF/DB)

- **MenuItem**: Id, Name, Description, IsAvailable, IsSpecialOfTheDay, CategoryId, Category.Name
- **MenuItemReview**: MenuItemId, UserId, Rating
- **Order**: Id, UserId
- **OrderItem**: OrderId, MenuItemId, Quantity, Order.UserId

Ako je šema različita, prilagoditi LINQ upite u sekcijama za recenzije i narudžbe.

6) Funkcije i formule (suština)

- **Ponderisani prosjek (profil):**
 - Za svaku stavku i sa težinom w_i i vektorom v_i :

$$\text{profil} = (\sum w_i * v_i) / (\sum w_i)$$
- **Kosinusna sličnost (rangiranje):**
 - $\cos(a, b) = (a \cdot b) / (\|a\| \times \|b\| + \epsilon)$ (ϵ radi numeričke stabilnosti)

7) Performanse i kompleksnost

- **Izgradnja modela:** $O(N)$ po broju stavki (radi se jednom po procesu).
- **Skoriranje:** $O(M \times D)$
 M = broj kandidata, D = dimenzija vektora.
- **Memorija:** linearna u broju stavki (pohrana `float[]` vektora).

8) Edge slučajevi i fallback

- Prazna baza / nema vektora → **cold-start**.
- Korisnik bez historije → **cold-start**.
- Nema kandidata nakon filtriranja → **cold-start**.

9) Integracija (primjer korištenja)

Registracija servisa (Program.cs):

```
builder.Services.AddScoped<IMenuRecommendationService, MenuRecommendationService>();
```

Primjer endpointa (User-scoped preporuke):

```

[ApiController]
[Route("api/MenuItem")]
[Authorize]
0 references
public class MenuItemRecommendationsController : ControllerBase
{
    // GET: /api/MenuItem/recommendations?take=10
    [HttpGet("recommendations")]
    0 references
    public async Task<ActionResult<IEnumerable<MenuItemResponse>>> GetRecommendations(
        [FromServices] IMenuRecommendationService recommender,
        [FromQuery] int take = 10,
        CancellationToken ct = default)
    {
        var claimVal = User?.FindFirst(ClaimTypes.NameIdentifier)?.Value;
        if (!int.TryParse(claimVal, out var userId))
            return Unauthorized();

        var recItems = await recommender.RecommendAsync(userId, take, ct);
        return Ok(recItems);
    }
}

```

Poziv sa klijenta: GET /api/menu/recommendations?take=10 (JWT obavezan).

10) Testiranje i validacija

- **Jedinice:**
 - Cosine(a,b) – provjeriti rubne slučajeve (prazni vektori, različite dužine).
 - WeightedAverage(weights) – normalizacija i ponašanje za wsum=0.
- **Integracija (mali uзорak):**
 - 5–10 stavki sa poznatim opisima, par recenzija i narudžbi → provjeriti da preferirane stavke zaista izlaze u top-N.
- **Load test:**
 - Mjerenje vremena za $M \approx 1000$ kandidata (prosječna dimenzija od FeaturizeText), cilj < 50–100 ms na serveru prosječne snage.

11) Ograničenja i prijedlozi za unapređenje

1. **Invalidacija modela/cachea**
 - Trenutno se gradi jednom. Dodati osvježavanje:
 - Na CRUD promjene MenuItem/Category,
 - Periodično (npr. timer), ili
 - “Cache stamp” + signal (npr. via event ili Redis pub/sub).
2. **Vremenski raspad (time-decay)**

- Novije recenzije/narudžbe neka vrijede više (npr. $w = \text{baseWeight} \times \exp(-\lambda \times \text{ageDays})$).
 - 3. **Diverzitet rezultata**
 - Penalizirati previše slične stavke (MMR – *Maximal Marginal Relevance*).
 - 4. **Hybrid pristup**
 - Kombinovati sa kolaborativnim filtriranjem kada skup korisnika/ocjena naraste.
 - 5. **Jezička podešavanja**
 - Prilagoditi tokenizaciju/stop-riječi lokalnom jeziku ili preći na naprednije embedinge (npr. sentence transformers) ako zatreba.
 - 6. **Operativno**
 - Logirati rebuild modela, vrijeme skoriranja i veličinu kandidata.
 - Postaviti metrike (npr. prosječan kosinus top-N, CTR).
-

12) Kratak pseudokod (za orijentir)

```
// ----- model build -----
1 reference
private void BuildModelIfNeeded()
{
    if (_itemVectors != null) return;

    lock (_sync)
    {
        if (_itemVectors != null) return;

        _ml = new MLContext(seed: 1);

        var rows = LoadItemRows();
        if (rows.Count == 0)
        {
            _itemVectors = new Dictionary<int, float[]>();
            return;
        }

        var data = _ml.Data.LoadFromEnumerable(rows);

        var pipeline =
            _ml.Transforms.Text.FeaturizeText("NameFeats", nameof(ItemRow.Name))
                .Append(_ml.Transforms.Text.FeaturizeText("CatFeats", nameof(ItemRow.Category)))
                .Append(_ml.Transforms.Text.FeaturizeText("DescFeats", nameof(ItemRow.Description)))
                .Append(_ml.Transforms.Concatenate("Features", "NameFeats", "CatFeats", "DescFeats"))
                .Append(_ml.Transforms.NormalizeLpNorm("Features"));

        var model = pipeline.Fit(data);
        var transformed = model.Transform(data);

        var vectors = _ml.Data.CreateEnumerable<VectorRow>(transformed, reuseRowObject: false);
        _itemVectors = vectors.ToDictionary(v => v.MenuItemId, v => v.Features);
    }
}
```

13) Cold-start detalji

1. IsSpecialOfTheDay == true (do take).
2. Preostalo popuniti **top-ocijenjenim** po prosjeku Rating i zatim po Count recenzija.
3. Ako još fali, uzeti **bilo koje dostupne** (IsAvailable == true) po alfabetu ili po nekom poslovnom kriteriju.