Sistem preporuka u aplikaciji Leami

Leami koristi content-based collaborative filtering pristup zasnovan na ML.NET Matrix Factorization algoritmu kako bi korisnicima predložio najrelevantnije artikle na osnovu njihove i tuđe historije narudžbi. Preporuke se generišu u dva koraka:

1. Mapiranje ko-kupovina

- Iz baze (Orders + OrderItems) se grupišu sve narudžbe i za svaku narudžbu prikupljaju listu artikala.

- Za svaki par različitih artikala (A, B) koji su kupljeni u istoj narudžbi, kreira se zapis CoPurchaseInput { ProductId = A, CoProductId = B, Label = 1f }.

- Ovaj skup parova se koristi za treniranje MF modela koji uči koliko su artikli međusobno “slični” na osnovu historije zajedničkih kupovina.

2. Predviđanje i rangiranje

- Kada korisnik gleda članak X, izdvajaju se svi ostali članci kao kandidati.

- Za svaki kandidat Y formira se ulazni vektor CoPurchaseInput { ProductId = X, CoProductId = Y, Label = 0f } (label nije bitan za predikciju).

- Transformiranim modelom (ITransformer) se izračuna score za svaki kandidat, koji označava vjerojatnost da bi korisnik koji je kupio X kupio i Y.

- Kandidati se sortiraju po opadajućem score-u i vraća se Top N artikala.

Ako ne postoji dovoljno podataka (npr. nema ni jedne prethodne narudžbe), metoda RecommendAsync vraća “featured” artikle .

Putanja i glavna logika

- Service:

Leami.Services\Services\ArticleService.cs

- Metoda RecommendAsync(int articleId, int take = 3)

- Pomoćna metoda EnsureModel(List<CoPurchaseInput> pairs) za građenje ili osvježavanje MF modela svakih 30 minuta.

- Interni DTO-i:

```
private sealed class CoPurchaseInput
{
  public uint ProductId { get; set; }
  public uint CoProductId { get; set; }
  public float Label { get; set; }
}
private sealed class CoPurchaseScore
{
  public float Score { get; set; }
}
```

- Kontroler:

LeamiAPI\Controllers\ArticlesController.cs

```
[HttpGet("{id}/recommend")]
public async Task<ActionResult<List<ArticleResponse>>> GetRecommendations(int id,
[FromQuery]int take = 3)
{
    var list = await _articleService.RecommendAsync(id, take);
    return Ok(list);
}
```