

## Recommender sistem – CafeEase

Recommender sistem u aplikaciji CafeEase implementiran je kao jednostavniji sistem preporuke baziran na historiji narudžbi korisnika. Sistem analizira podatke o narudžbama(Orders) i stavkama narudžbi(OrderItems) kako bi identifikovao proizvode koji se često kupuju zajedno, te na osnovu toga generiše preporuke.

Ako se određeni proizvodi često pojavljuju u istoj narudžbi, sistem će ih međusobno povezati i preporučiti jedan proizvod prilikom pregleda drugog proizvoda.

Ovaj recommender sistem koristi item-based collaborative filtering, zasnovan na analizi historije narudžbi.

### Korišteni podaci i struktura sistema

Recommender sistem koristi sljedeće podatke iz baze:

- Orders – narudžbe korisnika
- OrderItems – proizvodi unutar pojedinačne narudžbe
- Recommendations – tabela u kojoj se čuvaju generisane preporuke

Svaka narudžba može sadržavati više proizvoda, a upravo se ta informacija koristi za analizu proizvoda koji se često pojavljuju zajedno.

### Algoritam preporuke i proces treniranja

Recommender sistem koristi collaborative filtering baziran na zajedničkoj kupovini proizvoda (co-purchase analysis).

Proces treniranja funkcioniše na sljedeći način:

1. Sistem prolazi kroz sve postojeće narudžbe
2. Za svaku narudžbu se izdvajaju različiti proizvodi  
(U okviru jedne narudžbe svaki proizvod se uzima samo jednom (distinct). Ako je isti proizvod dodan više puta u istoj narudžbi, to ne povećava score za parove unutar te narudžbe).
3. Generišu se parovi proizvoda koji su naručeni zajedno  
(Parovi se generišu kao usmjereni odnosi: za proizvode A i B iz iste narudžbe bilježe se i  $(A \rightarrow B)$  i  $(B \rightarrow A)$ , dok se par  $(A \rightarrow A)$  preskače. Na taj način preporuke se čuvaju posebno za svaki “osnovni” proizvod).
4. Broji se koliko puta se isti par proizvoda pojavljuje
5. Za svaki proizvod se biraju tri najčešće povezane preporuke
6. Rezultati se spremaju u tabelu Recommendations

Vrijednost Score predstavlja učestalost zajedničke kupovine i koristi se za rangiranje preporuka.

```

public async Task TrainModel()
{
    _context.Recommendations.RemoveRange(_context.Recommendations);
    await _context.SaveChangesAsync();

    var orders = _context.Orders.Include(o => o.OrderItems).ToList();
    var pairs = new Dictionary<int, int>, int>();

    foreach (var order in orders)
    {
        var items = order.OrderItems.Select(x => x.ProductId).Distinct().ToList();

        for (int i = 0; i < items.Count; i++)
        {
            for (int j = 0; j < items.Count; j++)
            {
                if (i == j) continue;

                var key = (items[i], items[j]);

                if (!pairs.ContainsKey(key))
                    pairs[key] = 0;

                pairs[key]++;
            }
        }
    }

    var grouped = pairs.GroupBy(x => x.Key.Item1).ToDictionary( g => g.Key, g => g.OrderByDescending(p => p.Value).Take(3).ToList());

    foreach (var kvp in grouped)
    {
        var productId = kvp.Key;

        foreach (var rec in kvp.Value)
        {
            _context.Recommendations.Add(new Database.Recommendation
            {
                ProductId = productId,
                RecommendedProductId = rec.Key.Item2,
                Score = rec.Value
            });
        }
    }

    await _context.SaveChangesAsync();
}

```

## Generisanje i dohvat preporuka

Nakon što je model treniran, preporuke se ne računaju svaki put iznova, već se čitaju direktno iz baze. Za odabrani proizvod sistem vraća listu proizvoda sa najvećim Score vrijednostima.

Ako za dati proizvod ne postoje prethodno generisane preporuke (nema zapisa u tabeli Recommendations), sistem koristi fallback mehanizam i vraća tri najpopularnija proizvoda u sistemu. Popularnost se određuje prema broju pojavljivanja proizvoda u tabeli OrderItems (ukupan broj naručivanja), pri čemu se trenutni proizvod izuzima iz rezultata.

Ovaj proces omogućava brzo izvršavanje i efikasno korištenje sistema čak i sa većim brojem korisnika i narudžbi.

```

public async Task<List<Model.Product>> GetRecommendedProducts(int productId)
{
    var recs = await _context.Recommendations.Where(r => r.ProductId == productId)
        .OrderByDescending(r => r.Score).Take(3).ToListAsync();

    if (recs.Any())
    {
        var productIds = recs.Select(r => r.RecommendedProductId).Where(id => id != productId).Distinct().ToList();

        var dbProducts = await _context.Products.Where(p => productIds.Contains(p.Id)).ToListAsync();

        return _mapper.Map<List<Model.Product>>(dbProducts);
    }

    var popularIds = await _context.OrderItems.GroupBy(oi => oi.ProductId)
        .OrderByDescending(g => g.Count()).Select(g => g.Key)
        .Where(id => id != productId).Take(3)
        .ToListAsync();

    var fallbackProducts = await _context.Products.Where(p => popularIds.Contains(p.Id)).ToListAsync();

    return _mapper.Map<List<Model.Product>>(fallbackProducts);
}

```

## Brisanje preporuka

Omogućeno je potpuno brisanje svih preporuka iz baze.

```

public async Task DeleteAll()
{
    _context.Recommendations.RemoveRange(_context.Recommendations);
    await _context.SaveChangesAsync();
}

```

## API sloj recommender sistema

Recommender sistem je izložen putem REST API-ja i sadrži sljedeće funkcionalnosti:

- Treniranje recommender sistema
- Dohvat preporučenih proizvoda za određeni proizvod
- Brisanje svih preporuka

API je implementiran kroz poseban controller, dok se sva poslovna logika nalazi u servisnom sloju.

```

[ApiController]
[Route("[controller]")]
1 reference
public class RecommendationsController : ControllerBase
{
    private readonly IRecommendationService _service;

    0 references
    public RecommendationsController(
        IRecommendationService service)
    {
        _service = service;
    }

    [HttpPost("train")]
    0 references
    public async Task<IActionResult> Train()
    {
        await _service.TrainModel();
        return Ok("Recommendation model trained.");
    }

    [HttpGet("{productId}/recommended")]
    0 references
    public async Task<IActionResult> GetRecommended(int productId)
    {
        var result = await _service.GetRecommendedProducts(productId);
        return Ok(result);
    }

    [HttpDelete("clear")]
    0 references
    public async Task<IActionResult> Clear()
    {
        await _service.DeleteAll();
        return Ok("All recommendations deleted.");
    }
}

```

## Prikaz preporuka u Flutter aplikaciji

Implementirani recommender sistem omogućava generisanje preporuka proizvoda na osnovu prethodnih narudžbi korisnika. Sistem koristi collaborative filtering baziran na zajedničkoj kupovini proizvoda, čime se postiže jednostavno, ali efikasno rješenje. Preporuke se treniraju offline, spremaju u bazu podataka i brzo dohvaćaju putem API-ja, a krajnjim korisnicima se prikazuju u mobilnoj Flutter aplikaciji.

