

Univerzitet "Džemal Bijedić"  
Fakultet Informacijskih Tehnologija



**Recommender sistem dokumentacija**  
RAZVOJ SOFTVERA II

**Predmetni profesor:**

Prof. dr. sc. Elmir Babović

**Student:**

Mustafa Isić, IB190021

## Sadržaj

Sistem preporuke.....	3
Putanja i printscreen source code-a glavne logike recommender sistema .....	4
Putanja i printscreen iz pokrenute aplikacije gdje se prikazuju preporuke .....	5

## Sistem preporuke

U sklopu ovog rada implementiran je sistem za preporuku postova, čiji je cilj da svakom korisniku omogući personalizirane prijedloge sadržaja na osnovu njegovih prethodnih aktivnosti. Sistem koristi tehniku kolaborativnog filtriranja, koja predviđa interesovanja korisnika upoređujući ih sa ponašanjem drugih korisnika koji imaju slične obrasce interakcije.

Za potrebe obuke preporučivačkog modela koristi se kombinacija dviju vrsta interakcija: lajkova i dodavanja postova u omiljene. Svaka od ovih interakcija dobija odgovarajuću ocjenu: lajk se bilježi kao ocjena 1, dok se omiljeni post vrednuje kao 3. Na osnovu ovih ocjena, sistem gradi matricu korisnika i postova, koja se zatim koristi za predikciju budućih preferencija.

Model je izrađen uz pomoć biblioteke ML.NET i temelji se na algoritmu za faktORIZACIJU matrice (matrix factorization), koji omogućava da se iz postojećih podataka izdvoje latentne karakteristike korisnika i postova. Ove karakteristike se zatim koriste kako bi se procijenila vjerovatnoća da će se određeni post svidjeti korisniku. Za potrebe treniranja modela, koristi se `SquareLossOneClass` kao funkcija gubitka, uz dodatne parametre kao što su `Alpha`, `Lambda`, `NumberOfIterations` i `C`, koji podešavaju način na koji model uči iz podataka.

Kao ulazni podaci u model koristi se skup instanci koji sadrži:

- identifikator korisnika (`UserId`),
- identifikator posta (`ItemId`),
- ocjenu (`Rating`) koja zavisi od vrste interakcije.

Nakon što se model obuči, pristupa se generisanju preporuka za konkretnog korisnika. U ovom koraku, sistem prvo filtrira sve postove koje korisnik već poznaje, odnosno:

- postove koje je već lajkovao,
- postove koje je označio kao omiljene,
- postove koje je sam kreirao,
- te postove koji su deaktivirani (čiji status nije aktivan).

Na ovaj način se osigurava da korisniku ne budu predloženi sadržaji s kojima je već imao kontakt, kao ni njegovi vlastiti postovi, koji nisu korisni u kontekstu preporuke.

Za svaki preostali post koji zadovoljava ove uslove koristi se prethodno istrenirani model kako bi se predvidjela ocjena. Predikcija se vrši pozivom `CreatePredictionEngine`, kojem se proslijeđuju identifikatori korisnika i posta. Rezultat predikcije je numerički "score" koji odražava vjerovatnoću da će se dati post svidjeti tom korisniku.

Nakon predikcije, rezultati se sortiraju opadajuće prema dobijenom score-u, a korisniku se

prikazuje top 4 posta sa najvećom predikcijom relevantnosti.

## Putanja i printscreen source code-a glavne logike recommender sistema

Putanja u source code:

PlantCare/PlantCare/PlantCare.Services/PostRecommender.cs

Putanja na Github repozitoriju:

<https://github.com/mustafaisic000/PlantCare/blob/main/PlantCare/PlantCare.Services/PostRecommender.cs>

```
1  using Microsoft.EntityFrameworkCore;
2  using Microsoft.ML;
3  using Microsoft.ML.Data;
4  using Microsoft.ML.Trainers;
5  using PlantCare.Services.Database;
6  using System;
7  using System.Collections.Generic;
8  using System.Linq;
9
10 public class PostRecommender
11 {
12     private static readonly MLContext mlContext = new MLContext();
13     private static readonly object _lock = new object();
14     private static ITransformer model;
15     private readonly PlantCareContext _context;
16
17     public PostRecommender(PlantCareContext context)
18     {
19         _context = context;
20     }
21
22     public List<Post> Recommend(int korisnikId)
23     {
24         lock (_lock)
25         {
26             var interactions = _context.Lajkovi
27                 .Select(x => new UserItemEntry
28                 {
29                     UserId = (uint)x.KorisnikId,
30                     ItemId = (uint)x.PostId,
31                     Rating = 1
32                 })
33                 .Union(_context.OmiljeniPostovi
34                     .Select(x => new UserItemEntry
35                     {
36                         UserId = (uint)x.KorisnikId,
37                         ItemId = (uint)x.PostId,
38                         Rating = 3
39                     }))
40                 .ToList();
41
42             if (interactions.Count == 0)
43                 return new List<Post>();
44
45             var dataView = mlContext.Data.LoadFromEnumerable(interactions);
46
47             var options = new MatrixFactorizationTrainer.Options
48             {
49                 MatrixColumnIndexColumnName = nameof(UserItemEntry.UserId),
50                 MatrixRowIndexColumnName = nameof(UserItemEntry.ItemId),
51                 LabelColumnName = nameof(UserItemEntry.Rating),
52                 LossFunction = MatrixFactorizationTrainer.LossFunctionType.SquareLossOneClass,
53                 Alpha = 0.01,
54                 Lambda = 0.025,
55                 NumberOfIterations = 100,
56                 C = 0.00001
57             };
58         }
59     }
60 }
```

Slika 1: Printscreen source code-a glavne logike recommender sistema – prvi dio

```

53         alpha = 0.01,
54         Lambda = 0.025,
55         NumberOfIterations = 100,
56         C = 0.00001
57     };
58
59     var trainer = mlContext.Recommendation().Trainers.MatrixFactorization(options);
60     model = trainer.Fit(dataView);
61 }
62
63 var unseenPosts = _context.Postovi
64     .Include(p => p.Korisnik)
65     .Include(p => p.Subkategorija)
66     .Where(p =>
67         !_context.Lajkovi.Any(l => l.KorisnikId == korisnikId && l.PostId == p.PostId) &&
68         !_context.OmiljeniPostovi.Any(o => o.KorisnikId == korisnikId && o.PostId == p.PostId) &&
69         p.KorisnikId != korisnikId &&
70         p.Status == true)
71     .ToList();
72
73
74 var results = new List<Tuple<Post, float>>();
75
76 foreach (var post in unseenPosts)
77 {
78     var predictionEngine = mlContext.Model.CreatePredictionEngine<UserItemEntry, PostPrediction>(model);
79     var prediction = predictionEngine.Predict(new UserItemEntry
80     {
81         UserId = (uint)korisnikId,
82         ItemId = (uint)post.PostId
83     });
84     results.Add(new Tuple<Post, float>(post, prediction.Score));
85 }
86
87 return results.OrderByDescending(x => x.Item2).Select(x => x.Item1).Take(4).ToList();
88 }
89
90
91 7 references | mustafaisic000, 4 days ago | 1 author, 1 change
92 public class UserItemEntry
93 {
94     [KeyType(count: 1000)]
95     4 references | mustafaisic000, 4 days ago | 1 author, 1 change
96     public uint UserId { get; set; }
97
98     [KeyType(count: 1000)]
99     4 references | mustafaisic000, 4 days ago | 1 author, 1 change
100     public uint ItemId { get; set; }
101
102     3 references | mustafaisic000, 4 days ago | 1 author, 1 change
103     public float Rating { get; set; }
104 }
105
106 1 reference | mustafaisic000, 4 days ago | 1 author, 1 change
107 public class PostPrediction
108 {
109     1 reference | mustafaisic000, 4 days ago | 1 author, 1 change
110     public float Score { get; set; }
111 }
112
113 }

```

Slika 2: Printscreen source code-a glavne logike recommender sistema – drugi dio

Kreiranje mobilne aplikacije koja će korisnicima omogućiti lak pristup informacijama o biljkama, pretraživanje i filtriranje po kategorijama, kao i dodavanje vlastitih savjeta i iskustava.

## Putanja i printscreen iz pokrenute aplikacije gdje se prikazuju preporuke

Putanja komponente u source code:

PlantCare/PlantCare/UI/plantcare\_mobile/lib/common/widgets/stripe\_payment\_widget.dart

Putanja do home screen gdje se prikazuje komponenta na GitHub-u:

PlantCare/PlantCare/UI/plantcare\_mobile/lib/screen/home/home\_screen.dart

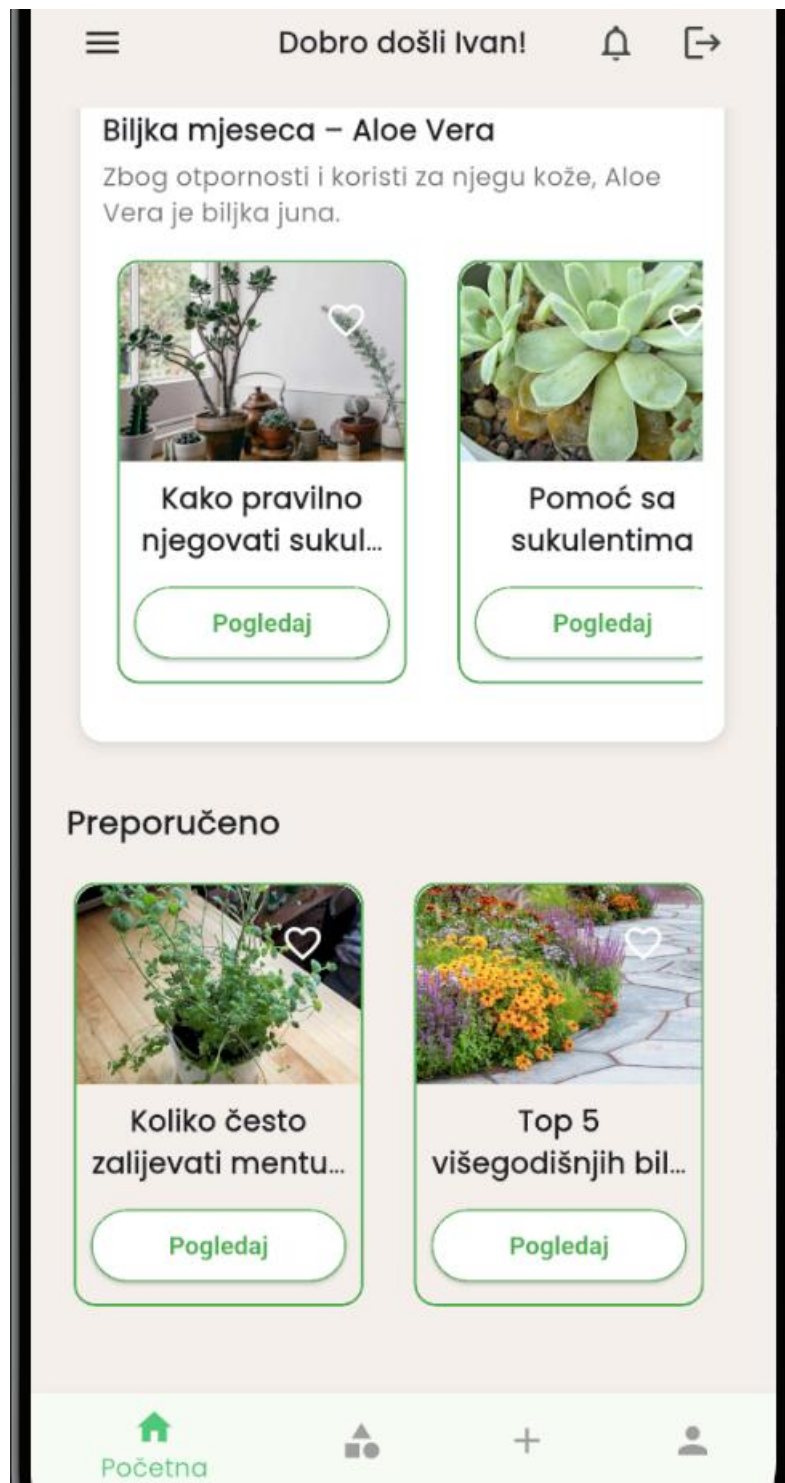
Putanja komponente na GitHub-u:

[https://github.com/mustafaisic000/PlantCare/blob/main/PlantCare/UI/plantcare\\_mobile/lib/com](https://github.com/mustafaisic000/PlantCare/blob/main/PlantCare/UI/plantcare_mobile/lib/com)

[mon/widgets/stripe\\_payment\\_widget.dart](mon/widgets/stripe_payment_widget.dart)

Putanja do home screen gdje se prikazuje komponenta na GitHub-u:

[https://github.com/mustafaisic000/PlantCare/blob/main/PlantCare/UI/plantcare\\_mobile/lib/screens/home/home\\_screen.dart](https://github.com/mustafaisic000/PlantCare/blob/main/PlantCare/UI/plantcare_mobile/lib/screens/home/home_screen.dart)



Slika 3: Printscreen iz pokrenute aplikacije gdje se prikazuju preporuke