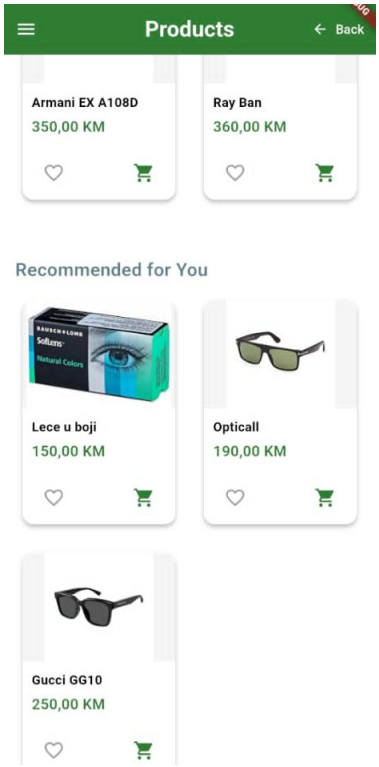
Aplikacija sadrži rekomender sistem koji koristi mašinsko učenje kako bi preporučio slične proizvode na osnovu prethodnih narudžbi korisnika. Model koristi algoritam Matrix Factorization, koji je popularna metoda za tzv. „collaborative filtering“. Analiziraju se prethodne narudžbe kako bi se utvrdilo koji su proizvodi često naručivani zajedno. Na primjer, ako je korisnik naručio Proizvod A i Proizvod B u istoj narudžbi, sistem bilježi par (A, B). Korištenjem MatrixFactorizationTrainer iz ML.NET biblioteke, model se trenira nad skupom tih parova. Model uči obrasce koji proizvodi imaju tendenciju da se pojavljuju zajedno u narudžbama. Preporuke za sve proizvode se spremaju u bazu. Ako već postoje rezultati, vrši se ažuriranje; višak se briše ili dodaju novi zapisi po potrebi. Potrebno je imati najmanje 5 proizvoda i barem 3 narudžbe koje sadrže više od jednog proizvoda kako bi se omogućilo kreiranje kvalitetnih preporuka.

OftamoloskaOrdinacija/OframoloskaOrdinacija.Services/RecommendResultService.cs

```
public List<Model.Proizvodi> Recommend(int? id)
{
    lock (isLocked)// we lock it until it is finished
    {
        if (mlContext == null)
        {
            mlContext = new MLContext();

            var tmpData = _context.Narudzbas.Include("StavkaNarudzbes").ToList();

            var data = new List<RatingEntry>();

            foreach (var x in tmpData)
            {
                if (x.StavkaNarudzbes.Count > 1)
                {
                    var distinctItemId = x.StavkaNarudzbes.Select(y => y.ProizvodId).ToList();

                    distinctItemId.ForEach(y =>
                    {
                        var relatedItems = x.StavkaNarudzbes.Where(z => z.ProizvodId != y).ToList();

                        foreach (var z in relatedItems)
                        {
                            data.Add(new RatingEntry()
                            {
                                RatingId = (uint)y,
                                CoRatingId = (uint)z.ProizvodId,
                            });
                        }
                    });
                }
            }
            var traindata = mlContext.Data.LoadFromEnumerable(data);
            MatrixFactorizationTrainer.Options options = new MatrixFactorizationTrainer.Options();
            options.MatrixColumnIndexColumnName = nameof(RatingEntry.RatingId);
            options.MatrixRowIndexColumnName = nameof(RatingEntry.CoRatingId);
            options.LabelColumnName = "Label";
            options.LossFunction = MatrixFactorizationTrainer.LossFunctionType.SquareLossOneClass;
            options.Alpha = 0.01;
            options.Lambda = 0.025;
            options.NumberOfIterations = 100;
            options.C = 0.00001;
```

```
            options.C = 0.00001;

            var est = mlContext.Recommendation().Trainers.MatrixFactorization(options);

            modeltr = est.Fit(traindata);
        }
    }

    var allItems = _context.Proizvods.Where(x => x.ProizvodId != id);
    var predictionResult = new List<Tuple<Database.Proizvod, float>>();

    foreach (var item in allItems)
    {
        var predictionEngine = mlContext.Model.CreatePredictionEngine<RatingEntry, Copurchase_prediction>(modeltr);
        var prediction = predictionEngine.Predict(new RatingEntry()
        {
            RatingId = (uint)id,
            CoRatingId = (uint)item.ProizvodId
        });

        predictionResult.Add(new Tuple<Database.Proizvod, float>(item, prediction.Score));
    }
    var finalResult = predictionResult.OrderByDescending(x => x.Item2).Select(x => x.Item1).Take(3).ToList();

    if (finalResult != null)
        return _mapper.Map<List<Model.Proizvodi>>(finalResult);
    return null;
}
```


Products
Back
Armani EX A108D
350,00 KM
Ray Ban
360,00 KM
Recommended for You
Lece u boji
150,00 KM
Opticall
190,00 KM
Gucci GG10
250,00 KM