Zahvaljujem doc. dr. sc. Mariju Brčiću i mag. ing. Kristijanu Poje na pristupačnosti, vremenu i pruženoj pomoći pri pisanju ovog rada.

SADRŽAJ

1. Uvod

Sustavi za preporučivanje često su korišteni alati koji mogu uvelike poboljšati korisničko iskustvo pri korištenju digitalnih platformi. Koriste algoritme strojnog učenja i prikupljene podatke kako bi na temelju raznih faktora korisniku preporučili proizvode od interesa [2]. Tako mogu smanjiti prostor pretraživanja s velike količine informacija na platformama samo na one koje su korisniku relevantne. Velika im je prednost i to što korisniku mogu preporučiti proizvode koje se oni sami ne bi sjetili pretražiti. S obzirom na faktore prema kojima se preporučuju proizvodi razlikuju se dvije glavne vrste sustava za preporuku: sustavi temeljeni na kolaborativnom filtriranju i sustavi temeljeni na sadržaju [9]. Sustav za preporučivanje temeljen na sadržaju koristi značajke proizvoda kako bi pojedinom korisniku preporučio proizvode slične njegovim preferencama. Pritom ne uzima u obzir preference drugih korisnika niti sličnosti među njima kao što to čini sustav temeljen na kolaborativnom filtriranju.

U ovom radu implementiran je sustav za preporuku kozmetičkih proizvoda temeljen na sadržaju. U drugom poglavlju predstavljen je skup podataka i transformacije koje su nad njim provedene kako bi se podaci pripremili za sustav za preporuku. Nadalje, obrađene su tri metode redukcije dimenzionalnosti i uspoređeni rezultati dani svakom metodom. U istom poglavlju dan je i pregled algoritama koji su korišteni za ostvarenje sustava. U trećem su poglavlju analizirani i uspoređeni rezultati ostvarenih sustava za preporuku. U četvrtom poglavlju dan je zaključak te su navedena moguća poboljšanja sustava.

2. Problematika preporuke proizvoda za njegu kože

2.1. Sustav za preporučivanje temeljen na sadržaju

Sustav za preporučivanje temeljen na sadržaju koristi značajke proizvoda kako bi korisniku preporučio druge slične proizvode na temelju njegovih prijašnjih kupnji, interakcija ili povratnih informacija [3]. Proces preporuke temelji se na sličnosti odnosno bliskosti proizvoda koja se mjeri kao sličnost sadržaja tih proizvoda, na primjer pripadnost istoj kategoriji ili žanru. Koristi se u mnoštvu raznih područja kao što su web trgovine i portali vijesti, a u ovom je radu korišten za preporuku proizvoda za njegu kože. Sustav od korisnika prima jedan ili više proizvoda na temelju čijih karakteristika, kao što su sastav, cijena, prosječna ocjena, kategorija i tip kože, preporuča slične proizvode. Ostvaren je kako bi se korisniku suzio prostor pretraživanja među mnoštvom proizvoda i poboljšalo iskustvo kupovine.

2.2. Skup podataka o proizvodima za njegu kože

Skup podataka korišten u ovom radu preuzet je s internetske stranice Kaggle koja čini online zajednicu praktičara u području strojnog učenja i znanosti o podacima [11]. Skup sadrži informacije o 1472 proizvoda koje su prikupljene s web-stranice Sephore. Sephora je francuski multinacionalni maloprodajni lanac kozmetičkih proizvoda. Skup podataka sastoji se od 11 stupaca koji označavaju karakteristike proizvoda. Stupac *'Label'* predstavlja kategoriju kojoj proizvod pripada, a poprima jednu od mogućih vrijednosti: *'Cleanser'* (čistač), *'Moisturizer'* (krema za lice), *'Eye cream'* (okoloočna krema), *'Face mask'* (maska za lice), *'Treatment'* (serum) i *'Sun protect'* (krema za sunčanje). Stupac *'brand'* označava proizvođača proizvoda od kojih je 116 različitih. Slijede stupci *'name'* i *'price'* u kojima su dani ime i cijena proizvoda u američkim

dolarima. Stupac *'rank'* označava prosječnu ocjenu svih korisnika koji su ocijenili taj proizvod na ljestvici od 0 do 5. Pet one-hot enkodiranih stupaca naziva *'Combination'* (mješovita koža), *'Oily'* (masna koža), *'Dry'* (suha koža), *'Normal'* (normalna koža) i *'Sensitive'* (osjetljiva koža) daju informaciju o tome kojem je tipu kože proizvod namijenjen. Stupac *'ingredients'* sadrži listu sastojaka proizvoda. Zapis sastojaka nije standardiziran te su isti sastojci zapisani na različite načine ovisno o proizvođaču što dovodi do brojke od 6670 različitih sastojaka i problema prevelike dimenzionalnosti.

2.3. Standardizacija sastojaka

Kako bi razlika u zapisima sastojaka bila što manja iz njih su prvo izbačeni svi posebni znakovi, nakon čega je korištena pythonova biblioteka *FuzzyWuzzy* koja na temelju Levenshteinove udaljenosti računa razlike među sekvencama. Funkcija [4] kao parametre prima skup podataka (df), stupac skupa koji želimo uspoređivati ($column$) s određenom sekvencom ($string\_to\_match$) i minimalni postotak sličnosti (min_ratio). Funkcija je pozvana za svaki sastojak u skupu podataka i tako je mijenjala zapise, koji su bili 85% slični, jednim standardiziranim zapisom tog sastojka. Time je broj različitih sastojaka smanjen na 3559. Zatim su izbačeni svi sastojci koji se pojavljuju u manje od 10 proizvoda jer ne bi pridonosili preporuci čime je broj ukupnih sastojaka od početnih 6670 smanjen na svega 736 različitih sastojaka na kojima je primjenjena redukcija dimenzionalnosti.

2.4. Redukcija dimenzionalnosti

Redukcija dimenzionalnosti je metoda reduciranja varijabli skupa podataka koji se koristi za izgradnju modela strojnog učenja. U procesu redukcije podaci visoke dimenzionalnosti, u ovom slučaju sastojci proizvoda, projiciraju se na prostor niže dimenzionalnosti u kojem je sačuvana srž tih podataka [5]. Postoji više metoda za redukciju dimenzionalnosti od kojih su odabrane tri za primjenu na skup sastojaka: analiza glavnih komponenti (PCA), t-SNE i UMAP. Na 736 sastojaka primjenjeno je one-hot enkodiranje. One-hot enkodiranje je tehnika mapiranja kategoričkih podataka na nule i jedinice na način da se u stupcu koji odgovara određenoj kategoričkoj varijabli nalazi jedinica ako joj proizvod pripada (sadrži neki sastojak u ovom slučaju) ili nula ako joj ne pripada. Time su dobiveni vektori dimenzije 736 za svaki od 1426 proizvoda u skupu podataka. Iz vektora sastojaka izbačen je stupac koji označava vodu i stupac

koji označava da nema podataka o sastojcima. Voda je izbačena zato što se nalazi u gotovo svakom proizvodu čime ne pridonosi različitosti među proizvodima, a stupac '*no info*' (nema podataka) izbačen je kako se sličnost proizvoda nebi mjerila po manjku podataka. Konačna matrica koja je predana metodama za redukciju dimenzionalnosti sastoji se od 1462 (retci) proizvoda i 734 sastojaka (stupci).

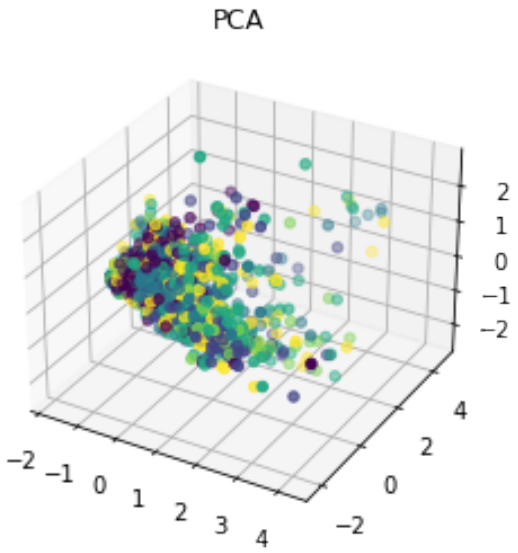
2.4.1. Analiza glavnih komponenti

Analiza glavnih komponenti (engl. *Principle Component Analysis*, PCA) je statistička metoda koja sažima informacije sadržane u većem skupu podataka na set glavnih komponenti kako bi se te informacije lakše vizualizirale i analizirale. Cilj metode je minimizirati pogrešku između originalnog uzorka i projekcije uzorka u sažeti potprostor. Korištena je implementacija metode PCA iz pythonove biblioteke za strojno učenje scikit-learn [8]. Podaci su reducirani na tri komponente. Zatim je pozvana metoda $explained\_variance\_ratio\_$ koja vraća postotak zadržane informacije za svaku od komponenti nakon primjene redukcije dimenzionalnosti.

```
pca3 = PCA(n_components=3, random_state=2023)
pca_data = pca3.fit_transform(df_data)
print(np.sum(pca3.explained_variance_ratio_))
```

Ispis 2.1: PCA

U ispisu 2.1 dan je isječak koda koji prikazuje poziv metode PCA s tri komponente i nasumično odabranim stanjem. U varijabli df_data nalazi se matrica koja ima 1426 redaka (proizvodi) i 734 stupca (sastojci). Funkcija $print$ ispisuje sumu rezultata poziva metode $explained\_variance\_ratio$ po komponentama iz kojeg je vidljivo da je ukupno sačuvano tek 11.98% informacije. Vizualizacija podataka smanjene dimenzionalnosti dana je na slici 2.1. Prikazane točke predstavljaju pojedini proizvod, a obojane su s obzriom na kategoriju kojoj pripadaju (čistač, krema za lice, serum, krema za zaštitu od sunca, maska za lice, okoloočna krema).



Slika 2.1: Podaci nakon redukcije metodom PCA

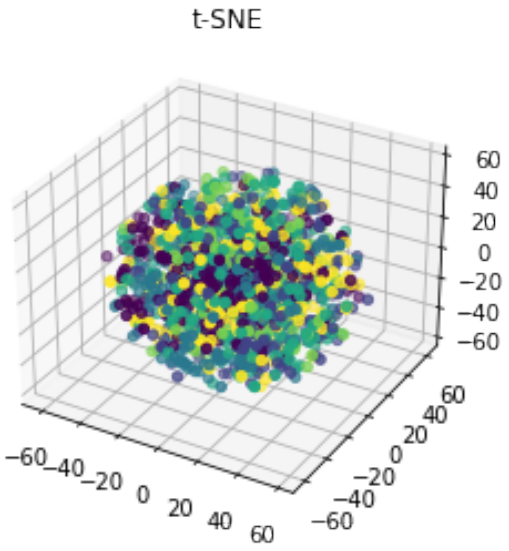
2.4.2. t-SNE

t-distribuirano ugniježđivanje stohastičkih susjeda (engl. *t-distributed Stochastic Neighbor Embedding*, t-SNE) je nenadzirana, nelinearna tehnika za vizualizaciju visokodimenzionalnih podataka [7]. Metoda stvara distribuciju vjerojatnosti u originalnom, visokodimenzionalnom prostoru nakon čega kreira prostor smanjenih dimenzija sa što sličnijom distribucijom. t-SNE minimizira udaljenosti između sličnih točaka čime su očuvane lokalne strukture. Korištena je implementacija metode TSNE, također iz biblioteke scikit-learn, a prostor je reduciran na tri komponente. t-SNE, za razliku od PCA, nema ugrađenu metodu koja izračunava postotak sačuvane informacije, stoga nije moguće direktno usporediti te dvije metode.

```
tsne = TSNE(n_components=3, random_state=2023)
tsne_data = tsne.fit_transform(df_data)
```

Ispis 2.2: t-SNE

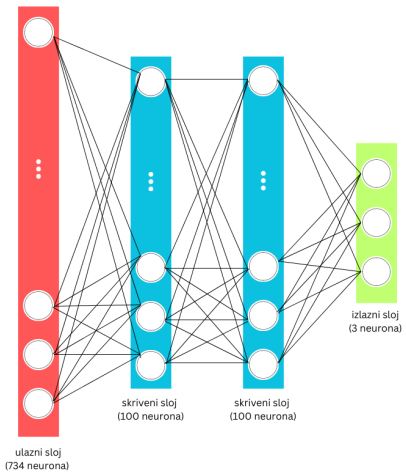
U isječku koda prikazanom u ispisu 2.2 instancira se t-SNE objekt putem metode TSNE s istim parametrima kao i PCA. Nakon čega se matrici proizvoda i sastojaka reducira dimenzionalnost na tri komponente. Vizualizacija prostora smanjene dimenzionalnosti, dobivenog metodom t-SNE dana je na slici 2.2. t-SNE za ovaj skup podataka nije pronašao skrivene uzorke u podacima



Slika 2.2: Podaci nakon redukcije metodom t-SNE

2.4.3. UMAP

Jednoliko aproksimiranje i projekcija višediomenzionalnih skupova za smanjenje dimenzija (engl. *Uniform Manifold Approximation and Projection for Dimension Reduction*, UMAP) ima vrlo sličan način rada kao t-SNE, ali čuva globalne strukture jednako kao i lokalne. Korišten je parametarski UMAP, koji za razliku od klasičnog UMAP-a, odnose među podacima uči koristeći neuronsku mrežu. U implementaciji neuronske mreže korišteni su Keras i Tensorflow, a sastoji se od 3 sloja po 100 potpuno povezanih neurona [10].



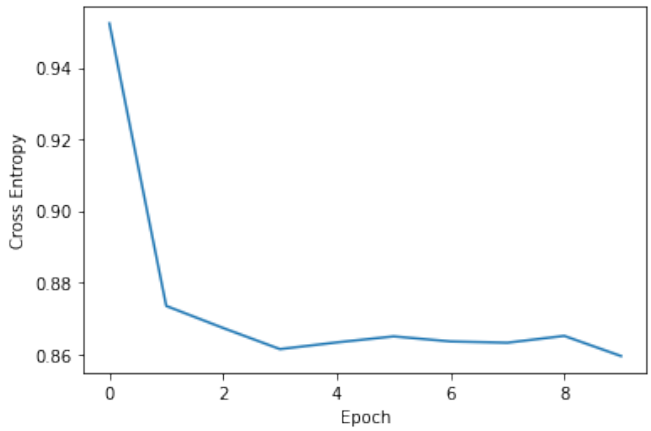
Slika 2.3: Pojednostavljeni prikaz arhitekture neuronske mreže

Zbog velikog broja neurona u stvarnoj mreži prikaz arhitekture je pojednostavljen i prikazan je na slici 2.3. Prvi sloj je ulazni sloj koji je dimenzije ulaznih podataka, u ovom slučaju 734 sastojka, zatim slijede dva sloja od 100 potpuno povezanih neurona i izlazni sloj. Sljedeći isječak koda prikazuje poziv metode kojim se instancira objekt $ParametricUMAP$ nakon čega se poziva njegova funkcija fit koja pokreće treniranje neuronske mreže nad predanim podacima.

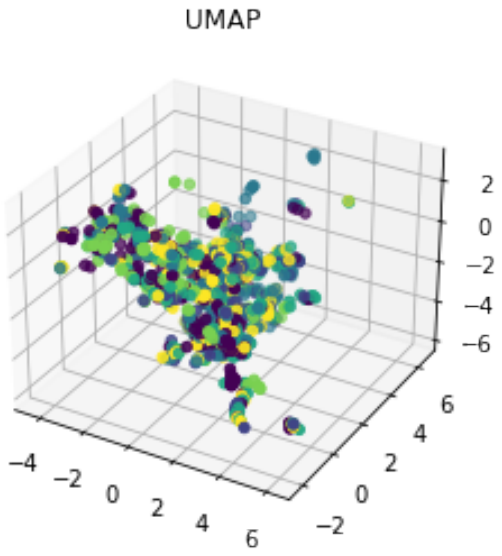
```
embedder = ParametricUMAP(n_neighbors=6, n_components
    =3, min_dist=0.001)
umap_all = embedder.fit(df_data)
```

Ispis 2.3: UMAP

Metoda fit prikazana u ispisu 2.3 trenira neuronsku mrežu u 10 epoha. Nakon svake epohe računa funkciju gubitka koju nastoji minimizirati. Na slici 2.4 prikazan je rezultat funkcije gubitka nakon svake epohe. Nakon posljednje epohe funkcija gubitka iznosi 0.8595. Vizualizacijom prostora smanjene dimenzionalnosti (slika 2.5) vidljivo je da niti ova metoda nije pronašla dobre, jasno odvojene skrivene uzorke među sastojcima. U usporedbi s dvije prethodno obrađene metode, UMAP daje najbolje rezultate stoga su tako enkodirani podaci korišteni u vektoru značajki proizvoda.



Slika 2.4: Iznos funkcije gubitka nakon pojedinih epoha



Slika 2.5: Podaci nakon redukcije metodom UMAP

2.5. Algoritmi za preporuku temeljeni na sadržaju

Kako bi sustav korisniku preporučio relevantne proizvode, potrebno je odrediti koji su proizvodi međusobno slični na temelju njihovih značajki, odnosno sadržaja. U ovom radu implementirana su tri metode pronalaženja sličnih proizvoda: kosinusna sličnost, model Gaussovih mješavina i metoda k-srednjih vrijednosti. Posljednje dvije metode rade po principu klasteringa. Klastering pripada metodama nenadziranog učenja kojemu je zadatak pronaći skrivene uzorke među ulaznim podacima bez poznavanja izlaza. To je tehnika strojnog učenja koja grupira točke podataka u različite klastere koji se sastoje od sličnih točaka. Zatim se, kako bi dobili preporuku, računaju udaljenosti točaka u istom klasteru [1]. Preporučeni su proizvodi najbliži odabranim točkama koje predstavljaju vektor proizvoda na temelju kojeg je tražena preporuka.

2.5.1. Vektor značajki proizvoda

Vektor značajki je n-dimenzionalni vektor numeričkih vrijednosti koji opisuje neki objekt. U korištenom skupu podataka dvije značajke nisu numeričke: kategorija i proizvođač. Ti su stupci enkodirani pomoću klase *LabelEncoder* iz biblioteke sklearn.preprocessing [8]. Nadalje, značajke *'rank'* i *'price'* skalirane su *MinMaxScaler*-om, također iz biblioteke sklearn.preprocessing. Varijable se skaliraju kako bi se uravnotežio njihov utjecaj na računanje udaljenosti. Dobiveni vektor značajki X ima 12 dimenzija i samo numeričke vrijednosti te je korišten kao ulaz u algoritmima za

preporuku.

	Label	brand	price	rank	Combination	Dry	Normal	Oily	Sensitive	umap_comp_1	umap_comp_2	umap_comp_3
0	Cleanser	ALGENIST	38	4.1	0	0	0	0	0	-3.462126	-1.281736	-0.404654
1	Cleanser	ALGENIST	25	4.4	0	0	0	0	0	-1.403818	-1.107007	2.699228
2	Cleanser	ALGENIST	38	4.6	0	0	0	0	0	-2.836174	-1.236796	0.511404
3	Cleanser	AMOREPACIFIC	50	4.5	1	0	1	1	0	-4.772723	-1.295502	-0.937622
4	Cleanser	AMOREPACIFIC	60	4.7	1	1	1	1	1	-3.255179	-1.451754	-0.393702

Slika 2.6: Vektor značajki prvih pet proizvoda prije enkodiranja i skaliranja

	Label	brand	price	rank	Combination	Dry	Normal	Oily	Sensitive	umap_comp_1	umap_comp_2	umap_comp_3
0	0	0	0.095368	0.82	0	0	0	0	0	-3.462126	-1.281736	-0.404654
1	0	0	0.059946	0.88	0	0	0	0	0	-1.403818	-1.107007	2.699228
2	0	0	0.095368	0.92	0	0	0	0	0	-2.836174	-1.236796	0.511404
3	0	1	0.128065	0.90	1	0	1	1	0	-4.772723	-1.295502	-0.937622
4	0	1	0.155313	0.94	1	1	1	1	1	-3.255179	-1.451754	-0.393702

Slika 2.7: Konačni vektor značajki prvih pet proizvoda

2.5.2. Kosinusna sličnost

Kosinusna sličnost je metrika koja se koristi za mjerenje sličnosti dvaju vektora. U ovom radu korištena je za mjerenje sličnosti vektora značajki proizvoda. Vektori moraju biti dio istog unitarnog prostora, a sličnost među njima mjeri se kosinusom kuta koji međusobno zatvaraju, otkud i dolazi naziv metode. Za računanje kosinusne sličnosti korištena je metoda $cosine\_similarity(X, Y = None, dense\_output = True)$. Metoda računa kosinusnu sličnost kao normalizirani skalarni umnožak *X* i *Y* [8].

Ispis 2.4: Stvaranje matrice kosinusnih sličnosti

```
cosine_sim = cosine_similarity(X, X)
```

U ispisu 2.4 prikazan je poziv metode kojoj je kao argument predan vektor značajki opisan u prethodnom potpoglavlju. Metoda vraća kvadratnu matricu dimenzije 1426 koja sadrži kosinusnu sličnost svakog proizvoda sa svakim drugim iz skupa podataka. Sustav za preporuku ostvaren je na način da se funkciji $recommendations\_cossim$ kao argument preda ime proizvoda kojem želimo naći najsličnije proizvode. Funkcija zatim sortira kosinusne sličnosti tog proizvoda sa svim ostalima i vraća pet proizvoda s najvećom kosinusnom sličnosti kao preporuku. Isječak navedene funkcije dan je u ispisu 2.5.

```
def recommendations_cossim(name,n=5,cosine_sim =
    cosine_sim):
```

```
    #get index of the product that matches the name
    idx = indices[indices == name].index[0]

    #find highest cosine_sim this title shares with other
        titles extracted earlier and save it in a Series
    score_series = pd.Series(cosine_sim[idx]).sort_values
       (ascending = False)

    #get indexes of the 'n' most similar products
    top_n_indexes = list(score_series.iloc[1:n+1].index)
```

Ispis 2.5: Funkcija za preporuku proizvoda na temelju kosinusne sličnosti

2.5.3. K-srednjih vrijednosti

K-srednjih vrijednosti je algoritam nenadziranog učenja koji za cilj ima grupirati slične točke podataka i pronaći skrivene uzorke. Kako bi to ostvario algoritam traži unaprijed određeni broj (k) klastera u danom skupu podataka [6]. Korištena je implementacija algoritma *KMeans* iz biblioteke scikit-learn [8]. Metoda *KMeans* grupira podatke u k klastera i svakoj točki pridodaje oznaku klastera kojem pripada. Ostvarene su dvije inačice sustava za preporuku ovom metodom, a razlikuju se u ulaznim podacima. Prva inačica prima ime jednog proizvoda na temelju kojeg se želi dobiti preporuka, dok druga inačica prima vektor proizvoda. Druga inačica pruža realniji model sustava zato što se kao ulazni vektor mogu predati stvarni podaci o proizvodima koje je korisnik kupio. Inačice su ostvarene jednakom logikom uz manje preinake s obzirom na ulaz. Funkcija $recommendations\_kmeans$ na temelju imena proizvoda dohvaća oznaku klastera kojem pripada. Zatim dohvaća sve točke iz tog klastera i poziva funkciju $nearest$ koja računa euklidsku udaljenost svake točke od zadanog proizvoda i vraća prvih 5 proizvoda s najamnjom udaljenosti.

```
    def recommendations_kmeans(product_name):
    product_index = df[df['name'] == product_name].index
       [0]
    cluster_label = df2.loc[product_index, 'cluster_label
       ']
    cluster_points = df2[df2['cluster_label'] ==
```

```
        cluster_label].index
    nearest_points = nearest(product_index,
        cluster_points, cluster_label)
```

Ispis 2.6: Funkcija za preporuku proizvoda temeljena na algoritmu *k-means*

Isječak metode $recommendations\_kmeans$ dan je u ispisu 2.6. U varijabli $nearest\_points$ spremljeni su parovi indeksa proizvoda i njegove udaljenosti od zadanog proizvoda. Inačica fukcije koja prima vektor proizvoda poziva funkciju $nearest$ za svaku od proizvoda, zatim sortira dobivene rezultate po udaljenosti i preporuča prvih 10 proizvoda s najmanjom udaljenosti. Što su udaljenosti proizvoda manje to su proizvodi sličniji i time je preporuka bolja.

3. Analiza rezultata

U iduća tri potpoglavlja dan je pregled rezultata sustava za preporuku za svaku korištenu metodu i njihova usporedba.

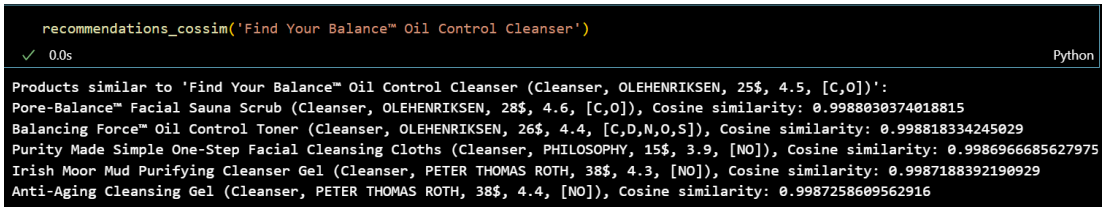
3.1. Kosinusna sličnost

Nasumično su izabrani proizvodi iz različitih kategorija kako bi se provela analiza rezultata. Korišten skup podataka nema informacije o stvarnim kupnjama korisnika stoga se analiza odnosno validacija provodi na temelju stvarne sličnosti proizvoda koji su preporučeni.

```
recommendations_cossim('GENIUS Liquid Collagen')
✓ 0.0s                                                                    Python
Products similar to 'GENIUS Liquid Collagen (Treatment, ALGENIST, 115$, 4.0, [C,D,N,O,S])':
GENIUS Ultimate Anti-Aging Vitamin C+ Serum (Treatment, ALGENIST, 118$, 3.9, [C,D,N,O,S]), Cosine similarity: 0.022903356323467017
FUTURE RESPONSE Age Defense Serum (Treatment, AMOREPACIFIC, 160$, 4.2, [C,D,N,O,S]), Cosine similarity: 0.022419222471582553
Pore Corrector Anti-Aging Primer (Treatment, ALGENIST, 42$, 4.4, [C,D,N,O,S]), Cosine similarity: 0.02268234147923616
ELEVATE Advanced Lift Contouring Cream (Treatment, ALGENIST, 96$, 4.3, [C,D,N,O,S]), Cosine similarity: 0.023605417031051935
MOISTURE BOUND Rejuvenating Serum (Treatment, AMOREPACIFIC, 100$, 4.4, [C,D,N,S]), Cosine similarity: 0.022271403662499543
```

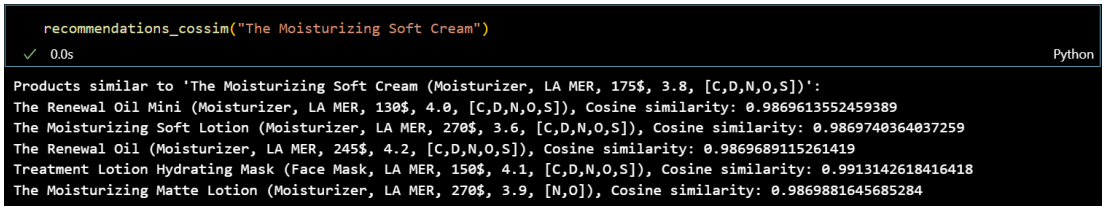
Slika 3.1: Preporuka na temelju proizvoda *'GENIUS Liquid Collagen'*

Na slici 3.1 prikazan je ispis funkcije za preporuku na temelju proizvoda *'GENIUS Liuquid Collagen'*. Ispis proizvoda je u obliku: *ime_proizvoda* (*kategorija*, *proizvođač*, *cijena*, *prosječna_ocjena*, [*tip_kože*]). Proizvod je namjenjen svim tipovima kože i spada u kategoriju seruma. Preporučeni proizvodi također spadaju u kategoriju seruma i imaju slične prosječne ocjene te su svi osim zadnjeg preporučenog također namijenjeni svim tipovima kože. Preporučeni su proizvodi različitih proizvođača, a sve ih povezuje karakteristika da imaju *anti-age* učinak kao i zadani proizvod, što je vidljivo iz njihovog imena. Dobar pokazatelj kvalitete preporuke je što je sustav prepoznao sličnost proizvoda iz iste linije što je vidljivo jer je prvi preporučeni proizvod upravo iz *'GENIUS'* linije.



Slika 3.2: Preporuka na temelju proizvoda *'Find Your Balance™ Oil Control Cleanser'*

Slika 3.2 prikazuje preporuke na temelju proizvoda iz kategorije čistača za kombiniranu i masnu kožu. Svi prepourčeni proizvodi također su iz kategorije čistača, slične prosječne ocjene, u istom cjenovnom rangu. Prva dva preporučena proizvoda odgovaraju istom tipu kože kao zadani, dok druga tri nemaju podataka o tome za koji su tip kože namijenjeni. Svi preporučeni proizvodi imaju kosinusnu sličnost od 0.99 što znači da su izrazito slični zadanom proizvodu i da je preporuka kvalitetna. Posljednji proizvod nad kojim je provedena analiza rezultata je iz kategorije krema za lice i odgovara svim tipovima kože. Rezultati sustava za preporuku dani su na slici 3.3.



Slika 3.3: Preporuka na temelju proizvoda *'The Moisturizing Soft Cream'*

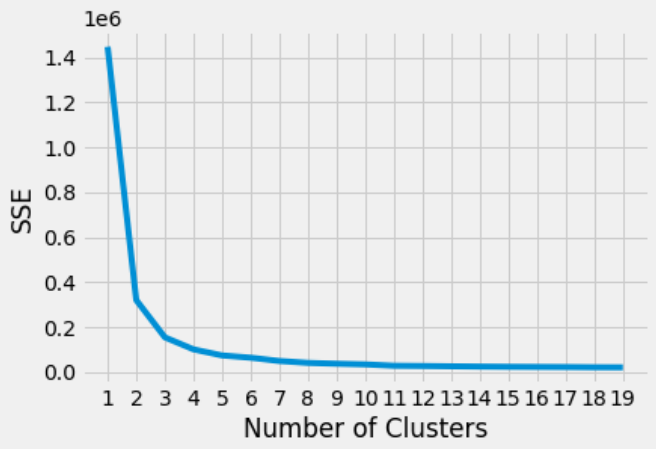
Ovaj proizvod je primjer za kojeg sustav nije preporučio isključivo proizvode iz iste kategorije. Svi preporučeni proizvodi osim posljednjeg, za kojeg nema podataka, odgovaraju svim tipovima kože. Svi preporučeni proizvodi od istog su proizvođača, i imaju kosinusnu sličnost od 0.98 i više što je izrazito dobar rezultat. Zanimljivo je primijetiti da su dvije preporuke zapravo isti proizvod, samo je jedan u putnoj verziji (*'The Renewal Oil Mini'* i *'The Renewal Oil'*) što je također dobar pokazatelj da sustav daje smislene preporuke i zaista prepoznaje sličnost među proizvodima.

3.2. Algoritam k-srednjih vrijednosti

3.2.1. Definiranje parametara

Definicija optimalnih parametara značajno utječe na performanse algoritma nad određenim skupom podataka. Iz tog razloga provodi se podešavanje hiperparametara u

kojem se isprobavaju zadane kombinacije parametara i traži ona kombinacija koja daje najmanju grešku ili najbolju evaluaciju modela. Za odabir broja klastera korištena je metoda lakta. Metoda lakta je vizualna metoda za određivanje najboljeg broja klastera koja uspoređuje razlike sume kvadrata pogreške (engl. *sum of square error*, SSE) svakog klastera. Najveća razlika tvori "lakat" koji označava najbolji broj klastera k. Na slici 3.4 vidljivo je da je lakat nastao na vrijednostima od dva i tri klastera.



Slika 3.4: Metoda lakta

Za podešavanje parametara iskorišten je algoritam *grid search*. *Grid search* provjerava sve kombinacije parametara iz definiranog rječnika i za njih računa koeficijent siluete. Koeficijent siluete je metrika za evaluaciju klastering algoritama, može poprimiti vrijednosti od -1 do 1, gdje 1 označava da su klasteri kompaktni i jasno odvojeni jedni od drugih. U ispisu 3.1 prikazana je inicijalizacija prostora pretraživanja parametara i provođenje *grid search* algoritma pomoću funkcije *GridSearchCV* iz scikit-learn biblioteke [8].

```
param_grid = {
    'n_clusters': range(2, 20),
    'init': ['k-means++', 'random'],
    'max_iter': [100, 300, 500]
}
grid_search = GridSearchCV(kmeans, param_grid, cv=5)
grid_search.fit(X)
best_estimator = grid_search.best_estimator_
best_estimator.fit(X)
silhouette_avg = silhouette_score(X, best_estimator.
    labels_)
```

```
best_params = grid_search.best_params_
```

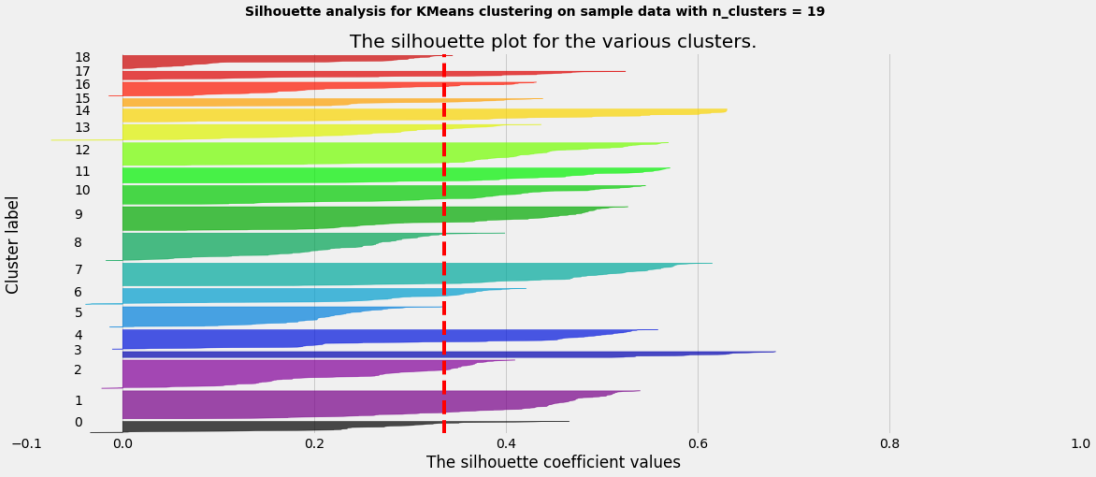
Ispis 3.1: Podešavanje hiperparametara

Na slici 3.5 prikazan je ispis algoritma *grid search* koji je kao optimalne parametre procijenio: broj klastera = 19, maksimalni broj iteracija = 300 i inicijalizacijska metoda = k-means++.

```
Silhouette Score: 0.33841040011961177
Best Parameters:  {'init': 'k-means++', 'max_iter': 300, 'n_clusters': 19}
```

Slika 3.5: Koeficijent siluete i optimalni parametri dani grid search algoritmom

Na slici 3.6 prikazan je koeficijent siluete svakog proizvoda u svakom od 19 klastera, a crvenom iscrtkanom crtom prikazan je prosječni koeficijent siluete.



Slika 3.6: Koeficijent siluete za 19 klastera

3.2.2. Rezultati

Analiza rezultata sustava za preporuku algoritmom k-srednjih vrijednosti provedena je nad nasumično odabranim proizvodima iz različitih kategorija.

```
recommendations('Bio-Performance LiftDynamic Serum')
✓ 0.1s                                                    Python
Products similar to 'Bio-Performance LiftDynamic Serum (Treatment, SHISEIDO, 98$, 3.9, [NO])':
--Bio-Performance Glow Revival Serum (Treatment, SHISEIDO, 92$ , 4.3, [NO]), Distance: 0.16
--Ultimate Sun Protection Cream Broad Spectrum SPF 50+ Wetforce For Face (Sun protect, SHISEIDO, 36$ , 4.1, [NO]), Distance: 1.36
--Bio-Performance Advanced Super Restoring Cream (Moisturizer, SHISEIDO, 127$ , 4.3, [NO]), Distance: 2.0
--Bio-Performance Glow Revival Cream (Moisturizer, SHISEIDO, 105$ , 4.8, [NO]), Distance: 2.01
--Benefiance WrinkleResist24 Night Emulsion (Moisturizer, SHISEIDO, 63$ , 4.3, [NO]), Distance: 2.02
```

Slika 3.7: Preporuka na temelju proizvoda *'Bio-Performance LiftDynamic Serum'*

Na slici 3.7 prikazane su preporuke na temelju proizvoda iz kategorije seruma koji nema podatke o tipu kože. Svi su preporučeni proizvodi od istog proizvođača, ali nisu iz iste kategorije kao zadani proizvod. Drugi preporučeni proizvod odstupa od prosjeka cijene ostalih proizvoda, a svi su preporučeni proizvode otprilike jednako rangirani. Prvi preporučeni proizvod ima najmanju udaljenost od zadanog te je time ta preporuka najsigurnija i najbolja, što se vidi i iz velike sličnosti karakteristika proizvoda. Ove preporuke dobar su primjer kako sustav prepoznaje proizvode iz iste linije kao slične jer je preporučio čak tri proizvoda iz linije *'Bio-Performance'*. Proizvodi iz iste linije obično su namijenjeni određenom stanju kože, stoga je velika vjerojatnost da ako korisniku odgovara proizvod iz neke linije, da će mu odgovarati i ostali zbog čega je ih je korisno preporučivati.

```
recommendations('Find Your Balance™ Oil Control Cleanser')
✓ 0.0s                                                                    Python

Products similar to 'Find Your Balance™ Oil Control Cleanser (Cleanser, OLEHENRIKSEN, 25$, 4.5, [C,O])':
--Pore-Balance™ Facial Sauna Scrub (Cleanser, OLEHENRIKSEN, 28$ , 4.6, [C,O]), Distance: 0.32
--Balancing Force™ Oil Control Toner (Cleanser, OLEHENRIKSEN, 26$ , 4.4, [C,D,N,O,S]), Distance: 1.74
--Reviving Eye Cream (Eye cream, OMOROVICZA, 145$ , 3.8, [NO]), Distance: 2.29
--Stay Balanced™ Oil Control Cleansing Cloths (Cleanser, OLEHENRIKSEN, 8$ , 4.5, [C,O]), Distance: 2.58
--Face the Truth™ Gel Cleanser (Cleanser, OLEHENRIKSEN, 24$ , 4.5, [C,D,N,O,S]), Distance: 2.97
```

Slika 3.8: Preporuka na temelju proizvoda *'Find Your Balance™ Oil Control Cleanser'*

Slika 3.8 prikazuje preporuke na temelju čistača za kombiniranu i masnu kožu. U ovom primjeru prisutno je i odstupanje u kvaliteti preporuke. Treći preporučeni proizvod nije iz iste kategorije, niti je od istog proizvođača, a odstupa i u cijeni i ocjeni zbog čega to nije dobra preporuka. Ostali preporučeni proizvodi su od istog proizvođača i iz iste kategorije, a kvalitetu preporuke pokazuje i to što dva preporučena proizvoda imaju iste ključne riječi *'oil control'* u imenu kao i zadani proizvod.

```
recommendations_multiple(["Treatment Cleansing Foam", "Rejuvenating Serum", "The Moisturizing Soft Cream"])
✓ 0.2s                                                                    Python

Products similar to:
Treatment Cleansing Foam (Cleanser, AMOREPACIFIC, 50£, 4.5, [C,N,O])
Rejuvenating Serum (Treatment, TATA HARPER, 110£, 3.7, [C,D,N,O,S])
The Moisturizing Soft Cream (Moisturizer, LA MER, 175£, 3.8, [C,D,N,O,S])
--The Renewal Oil Mini (Moisturizer, LA MER, 130$, 4.0, [C,D,N,O,S]), Distance: 0.25
--The Moisturizing Soft Lotion (Moisturizer, LA MER, 270$, 3.6, [C,D,N,O,S]), Distance: 0.26
--The Renewal Oil (Moisturizer, LA MER, 245$, 4.2, [C,D,N,O,S]), Distance: 0.3
--Resurfacing Serum (Treatment, TATA HARPER, 88$, 4.2, [C,D,N,O,S]), Distance: 0.33
--Treatment Lotion Hydrating Mask (Face Mask, LA MER, 150$, 4.1, [C,D,N,O,S]), Distance: 1.05
--The Moisturizing Matte Lotion (Moisturizer, LA MER, 270$, 3.9, [N,O]), Distance: 1.76
--Treatment Enzyme Peel (Cleanser, AMOREPACIFIC, 60$, 4.7, [C,D,N,O,S]), Distance: 2.15
--GENIUS Ultimate Anti-Aging Melting Cleanser (Cleanser, ALGENIST, 38$, 4.1, [NO]), Distance: 2.45
--OIL OBSESSED™ Total Cleansing Oil (Cleanser, BAREMINERALS, 30$, 4.6, [C,D,N,O]), Distance: 2.85
--Resurfacing Mask (Face Mask, TATA HARPER, 62$, 4.3, [C,D,N,O,S]), Distance: 3.02
```

Slika 3.9: Preporuka na temelju vektora proizvoda

Na slici 3.9 prikazan je ispis preporuka na temelju vektora od tri proizvoda. Udaljenost prva četiri proizvoda od proizvoda iz ulaznog vektora manja je od 1 zbog čega

su to najsigurnije i najbolje preporuke. Korist ovakvog sustava s vektorom ulaznih podataka jest što korisnik može upisati sve proizvode iz svoje rutine koji mu odgovaraju i na temelju njih dobiti preporuke.

3.3. Usporedba algoritama za preporuku

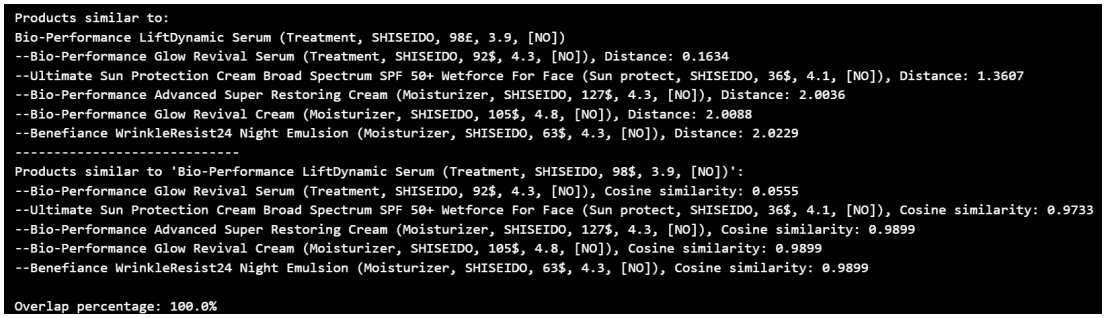
Zbog nedostatka podataka o stvarnim kupnjama korisnika i nemogućnosti validacije oba algoritma istom metrikom, algoritmi su uspoređeni na temelju postotka istih preporuka za pojedini proizvod. U nastavku su prikazani postotci preklapanja preporuka za proizvode analizirane u potpoglavljima 3.1 i 3.2.2. Slike prikazuju ispis nakon poziva metode $compare\_recommendations$ koja poziva metode za preporuke i računa postotak preporučenih proizvoda koji se preklapaju. Prvo su ispisane preporuke sustava ostvarenog algoritmom k-srednjih vrijednosti, zatim preporuke sustava ostvarenog kosinusnom sličnosti te naposljetku postotak preklapanja tih preporuka.

```
Products similar to:
GENIUS Liquid Collagen (Treatment, ALGENIST, 115£, 4.0, [C,D,N,O,S])
--GENIUS Ultimate Anti-Aging Vitamin C+ Serum (Treatment, ALGENIST, 118$, 3.9, [C,D,N,O,S]), Distance: 1.1465
--Pore Corrector Anti-Aging Primer (Treatment, ALGENIST, 42$, 4.4, [C,D,N,O,S]), Distance: 1.4874
--FUTURE RESPONSE Age Defense Serum (Treatment, AMOREPACIFIC, 160$, 4.2, [C,D,N,O,S]), Distance: 1.5252
--ELEVATE Advanced Lift Contouring Cream (Treatment, ALGENIST, 96$, 4.3, [C,D,N,O,S]), Distance: 2.0668
--MOISTURE BOUND Rejuvenating Serum (Treatment, AMOREPACIFIC, 100$, 4.4, [C,D,N,S]), Distance: 2.1305
----------------------------
Products similar to 'GENIUS Liquid Collagen (Treatment, ALGENIST, 115$, 4.0, [C,D,N,O,S])':
--GENIUS Ultimate Anti-Aging Vitamin C+ Serum (Treatment, ALGENIST, 118$, 3.9, [C,D,N,O,S]), Cosine similarity: 0.0229
--FUTURE RESPONSE Age Defense Serum (Treatment, AMOREPACIFIC, 160$, 4.2, [C,D,N,O,S]), Cosine similarity: 0.0224
--Pore Corrector Anti-Aging Primer (Treatment, ALGENIST, 42$, 4.4, [C,D,N,O,S]), Cosine similarity: 0.0227
--ELEVATE Advanced Lift Contouring Cream (Treatment, ALGENIST, 96$, 4.3, [C,D,N,O,S]), Cosine similarity: 0.0236
--MOISTURE BOUND Rejuvenating Serum (Treatment, AMOREPACIFIC, 100$, 4.4, [C,D,N,S]), Cosine similarity: 0.0223

Overlap percentage: 100.0%
```

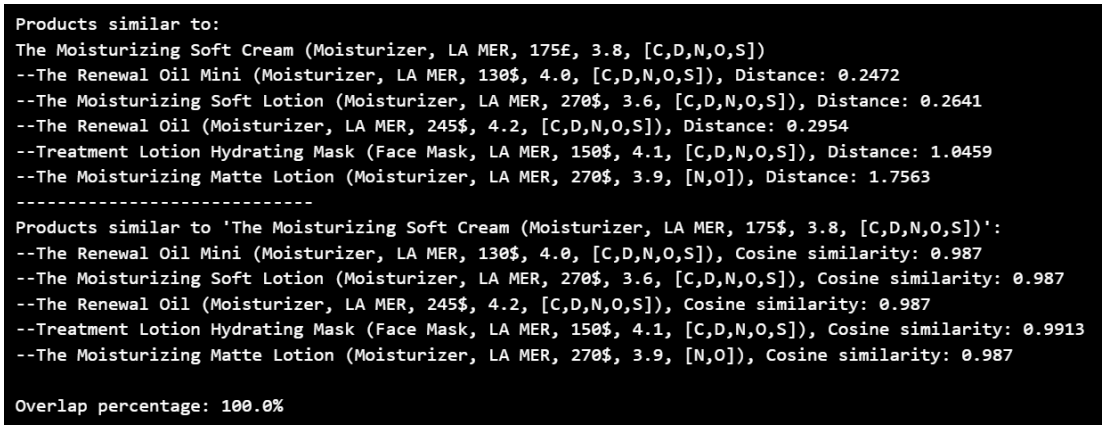
Slika 3.10: Usporedba preporuka za proizvod *'GENIUS Liquid Collagen'*

```
Products similar to:
Find Your Balance™ Oil Control Cleanser (Cleanser, OLEHENRIKSEN, 25£, 4.5, [C,O])
--Pore-Balance™ Facial Sauna Scrub (Cleanser, OLEHENRIKSEN, 28$, 4.6, [C,O]), Distance: 0.3165
--Balancing Force™ Oil Control Toner (Cleanser, OLEHENRIKSEN, 26$, 4.4, [C,D,N,O,S]), Distance: 1.7403
--Reviving Eye Cream (Eye cream, OMOROVICZA, 145$, 3.8, [NO]), Distance: 2.29
--Stay Balanced™ Oil Control Cleansing Cloths (Cleanser, OLEHENRIKSEN, 8$, 4.5, [C,O]), Distance: 2.5843
--Face the Truth™ Gel Cleanser (Cleanser, OLEHENRIKSEN, 24$, 4.5, [C,D,N,O,S]), Distance: 2.9693
----------------------------
Products similar to 'Find Your Balance™ Oil Control Cleanser (Cleanser, OLEHENRIKSEN, 25$, 4.5, [C,O])':
--Pore-Balance™ Facial Sauna Scrub (Cleanser, OLEHENRIKSEN, 28$, 4.6, [C,O]), Cosine similarity: 0.9988
--Balancing Force™ Oil Control Toner (Cleanser, OLEHENRIKSEN, 26$, 4.4, [C,D,N,O,S]), Cosine similarity: 0.9988
--Purity Made Simple One-Step Facial Cleansing Cloths (Cleanser, PHILOSOPHY, 15$, 3.9, [NO]), Cosine similarity: 0.9987
--Irish Moor Mud Purifying Cleanser Gel (Cleanser, PETER THOMAS ROTH, 38$, 4.3, [NO]), Cosine similarity: 0.9987
--Anti-Aging Cleansing Gel (Cleanser, PETER THOMAS ROTH, 38$, 4.4, [NO]), Cosine similarity: 0.9987

Overlap percentage: 40.0%
```

Slika 3.11: Usporedba preporuka za proizvod *'Find Your Balance™ Oil Control Cleanser'*



Slika 3.12: Usporedba preporuka za proizvod *'Bio-Performance LiftDynamic Serum'*



Slika 3.13: Usporedba preporuka za proizvod *'The Moisturizing Soft Cream'*

Sustavi daju identične preporuke za sve proizvode osim *'Find Your Balance™ Oil Control Cleanser'*. Za taj proizvod daju samo dvije jednake preporuke i to su upravo one s najmanjom udaljenosti od zadanog proizvoda u sustavu ostvarenog metodom k-srednjih vrijednosti.

4. Zaključak

Cilj ovog rada bio je implementirati sustav za preporuku proizvoda za njegu kože temeljen na sadržaju. Kako bi se podaci o proizvodima mogli koristiti u sustavu bilo je potrebno analizirati skup podataka i transformirati ga. Nad podacima o sastojcima koji čine proizvode provedena je redukcija dimenzionalnosti pomoću tri različite metode: PCA, t-SNE i UMAP. Navedene metode redukcije nisu dale dobre rezultate za korišteni skup podataka. Rezultati bi se mogli poboljšati korištenjem enkodera koji grupira sastojke na temelju njihovog kemijskog sastava i molekulskog oblika, no to je izlazilo iz opsega ovog rada te se pitanje enkodiranja sastojaka ostavlja daljnjim istraživanjima. Nakon pripreme podataka ostvaren je sustav za preporuku metodom kosinusne sličnosti i metodom k-srednjih vrijednosti. Naposljetku su analizirani rezultati i dana je usporedba ta dva sustava. Oba sustava dala su dobre i sigurne preporuke s manjim odstupanjima. Validacija i evaluacija sustava može biti poboljšana generiranjem sintetičkih podataka o kupnjama korisnika, nakon čega bi se mogao izračunati postotak stvarno kupljenih proizvoda od onih preporučenih.

LITERATURA

[1] Clustering in machine learning. URL https://www.javatpoint.com/clustering-in-machine-learning.

[2] Recommendation system. URL https://www.nvidia.com/en-us/glossary/data-science/recommendation-system/.

[3] Content-based filtering, 2022. URL https://developers.google.com/machine-learning/recommendation/content-based/basics.

[4] Thanh Huynh. Fuzzywuzzy: Find similar strings within one column in python, 2020. URL https://towardsdatascience.com/fuzzywuzzy-find-similar-strings-within-one-column-in-a-pandas-data-frame-99f6c2a0c212.

[5] Vijay Kanade. What is dimensionality reduction? meaning, techniques, and examples, 2022. URL https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-dimensionality-reduction/.

[6] Education Ecosystem (LEDU). Understanding k-means clustering in machine learning, 2018. URL https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1.

[7] Matej Marić. Clustering in machine learning, 2022. URL https://www.megatrend.com/vizualizacija-visokodimenzionalnih-podataka/.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, i E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[9] Baptiste Rocca. Introduction to recommender systems, 2019. URL https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada.

[10] Tim Sainburg, Leland McInnes, i Timothy Q Gentner. Parametric umap embeddings for representation and semisupervised learning. *Neural Computation*, 33 (11):2881–2907, 2021.

[11] Domino Weir. Sephora skincare ingredients, 2022. URL https://www.kaggle.com/datasets/dominoweir/skincare-product-ingredients.

Sažetak

Sustavi za preporučivanje popularni su alati koji znatno poboljšavaju iskustvo korisnika tijekom korištenja digitalnih platformi. U ovom radu implementiran je sustav za preporuku proizvoda za njegu kože temeljen na sadržaju. Sustavi za preporuku temeljeni na sadržaju preporučaju proizvode na temelju njihovih značajki i preferenca određenog korisnika. Nad podacima o sastojcima koji čine proizvode provedena je redukcija dimenzionalnosti pomoću tri različite metode: PCA, t-SNE i UMAP. Za ostvarenje sustava za preporuku korištena su dva algoritma: kosinusna sličnost i k-srednjih vrijednosti. Rezultati preporuka validirani su na temelju stvarne sličnosti zadanog i preporučenih proizvoda. Usporedbom rezultata dvaju implementiranih sustava utvrđeno je da daju iste preporuke za 4/5 zadanih proizvoda.

Ključne riječi: sličnost, kosinusna sličnost, k-srednjih vrijednosti, redukcija dimenzionalnosti, sustav za preporuku

Content-based recommender system for skincare products

Abstract

Recommendation systems are popular tools that significantly enhance the user experience when using digital platforms. A content-based recommendation system for skincare products has been implemented in this paper. Content-based recommendation systems recommend products based on their features and the preferences of a specific user. Dimensionality reduction was performed on ingredient data using three different methods: PCA, t-SNE and UMAP. Two algorithms were utilized to achieve the recommendation system: cosine similarity and k-means clustering. The recommendation results were validated based on the actual similarity between the given and recommended products. By comparing the results of the two implemented systems, it was determined that they provide the same recommendations for 4 out of 5 given products.

Keywords: similarity, cosine similarity, k-means, dimensionality reduction, recommender system