**Universitatea Tehnică ”Gheorghe Asachi” din Iași**
**Facultatea de Automatică și Calculatoare**
**Domeniul Calculatoare și Tehnologia Informației**
**Specializarea Tehnologia Informației**

# Aplicație bazată pe microservicii pentru regăsirea în scopuri educaționale a unor persoane cu interese comune
# Referat 2

**Student:**
**Cornea Radu-Valentin, 1411A**

**Coordonator științific:**
**S.l. dr. inf. Dumitriu Tiberius**

**2022-2023**

# Introducere

Proiectul constă într-o aplicație pentru interconectarea oamenilor, în funcție de preferințele lor, utilizând diverși algoritmi. Unii dintre algoritmii utilizați sunt mai simpli, iar alții folosesc inteligență artificială. Se dorește și recomandarea utilizatorilor în funcție de alte filtre precum distanța geografică, dar momentan recomandările se fac doar în funcție de preferințe. Scopul proiectului este de a uni oamenii mai ușor în scop educațional în funcție de preferințele legate de tehnologii sau concepte teoretice, însă proiectul ar putea fi folosit și în alte arii decât cele educaționale.Totuși, există o mulțime de studii pe tema recomandărilor, unii dintre oameni gândindu-se deja la sisteme de recomandări bazate pe inteligență artificială și pe psihologia educației, tocmai pentru a recomanda studenților resurse de studiu, în funcție de cum interacționeaza cu imaginile de pe site-ul respectivei instituții și cu elementele video din acestea. Studenții au fost clasați în: studenți activi, studenți cu potențial și studenți inactivi. Toate aceste trei grupuri au fost împărțite pe baza modului în care interacționau cu paginile respective, urmând ca fiecare dintre ei să primească resurse educative potrivite [1]. Există foarte multe domenii de aplicare pentru algoritmii de recomandare, și pot fi folosiți nu doar în scopuri educative, ci și divertisment, precum filmele [2].

În urma unui studiu de piață, s-a constatat că, în ciuda faptului că există aplicații care încearcă să recomande oameni după anumite criterii, acestea nu realizează în totalitate obiectivul dorit. Multe dintre aplicațiile găsite pe Play Store includ opțiunea de alegere de preferințe (de exemplu, mâncare, hobby-uri, muzică), însă filtrele de utilizatori sunt inexistente, neputând primi utilizatori similari unei ținte, cel puțîn din punctul de vedere al preferințelor. Singura aplicație care s-a constatat că ar face o parte dintre aceste funcționalități de recomandare este Meetup, dar acolo au loc recomandări de evenimente, nu de persoane. Panion ar fi fost un exemplu bun, dar în prezent nu mai funcționează publicului larg.

Pe baza acestui studiu de piață, s-a ajuns la concluzia că o astfel de aplicație pentru recomandarea persoanelor în funcție de preferințele lor ar fi necesară pieței. Obiectivul principal este acela de a găsi și de a filtra cât mai mulți utilizatori potriviți cu ținta în cauză, dar și de a avea un produs funcțional, sigur și securizat, care să fie ușor de folosit, plăcut și de înțeles de oricine.

O posibilitate de a rezolva această problemă este de a folosi microserviciile și inteligența artificială. Arhitecturile bazate pe microservicii oferă scalabilitate, rulare independentă, performanțe adiționale, avantajul de a fi ușor de menținut, costuri reduse și multe altele [3]. În timp ce inteligența artificială oferă posibilități multiple de modelare și antrenare a datelor după anumite seturi de date. Câțiva dintre acești algoritmi ce pot fi folosiți în acest sens sunt: KNN, SVM, SVD, Random Forest, filtrul colaborativ, filtrul bazat pe conținut, abordări hibride [4, 5].

Tocmai de aceea, în încercarea de a rezolva problema propusă, tehnologiile și conceptele teoretice folosite în cadrul aplicației pot fi clasificate astfel:

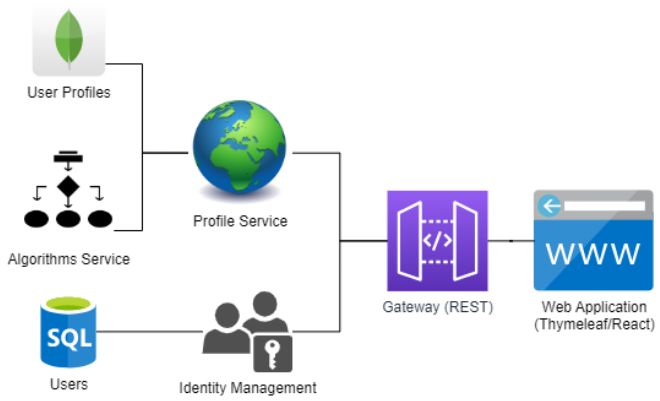
- Backend: Spring, Spring Security, JWTs, REST, Kotlin, MariaDB, MongoDB, criptare, decriptare;
- Frontend: Thymeleaf, JavaScript, CSS, HTML;
- AI: KNN, Python, sklearn, pandas;
- Aplicații software: IntelliJ IDEA, PyCharm, Visual Studio Code, Postman, MongoDB, Compass, DBeaver.

# 1. Proiectarea software a aplicației

O posibilitate de a rezolva soluția propusă, așa cum s-a discutat în capitolul anterior, este de a folosi microserviciile împreună cu inteligența artificală. Arhitectura întregii aplicației ar putea fi împărțită astfel: pentru servicii, se pune accentul pe cum comunică între ele și cum sunt împărțite, încercând să se respecte cât mai bine principiile SOLID, atât pentru clase, cât și microservicii, și cea dea doua arhitectură este pentru inteligență artificială, în care importante sunt datele care intră, datele care se așteaptă drept ieșiri, dar și modul în care au fost prelucrate acestea, putând asemana arhitectura aceasta cu white box.

Alte părți de cod au o granularitate fină, având propriul proiect și server, urmărindu-se în acest fel și decuplarea aplicației dezvoltate. Aplicația se folosește de modelul client-server, fiind bazată pe servicii/microservicii. Câteva dintre serviciile implementate până acum cuprind: IDM (Identity Management), Profile, Algorithms și Gateway-ul ce leagă serviciile. Alte exemple de servicii utile aplicației, dar neimplementate încă, sunt cele pentru identificarea locației, comunicarea între utilizatori, și pentru suport-ul utilizatorului (în cazul în care acesta are nevoie de ajutor, să poată depune cereri).

Pe lângă cele două arhitecturi menționate, au fost necesare și diagrame de servicii, diagrame UML și ER. Pentru o bună funcționalitate, dar și dezvoltare mai facilă a aplicației, părțile de cod au fost despărțite pe module și clase, în unele situații fiind necesare șabloanele de proiectare. Unul dintre acestea este chiar șablonul de proiectare strategie. Acesta a fost folosit pentru a decide modul în care utilizatorii vor fi recomandați unei ținte. În prezent, sunt patru strategii de a decide acest aspect: căutari directe pe utilizatori utilizând paradigma funcțională, iar celelalte strategii bazându-se pe algoritmul KNN și câteva metrici ale acestuia.




# 2. Rezultate intermediare obținute

Până în momentul de față, s-a realizat o comparare între câteva dintre metricile algoritmului KNN, acestea fiind Jaccard, euclidiană și cosine. Dar pe lângă aceste comparări, s-au întocmit și niște rapoarte care ar sugera diverse situații de utilizare mai potrivite acelor metrici.

Prin aceste experimente se urmăresc: compararea metricilor cosine, euclidiană și Jaccard; compararea k-urilor pentru KNN (3, 5, 7, 11, lungimea datelor de antrenare); dar și urmărirea potrivirii utilizatorilor în funcție de o țintă cu preferințe oarecare, încât modelul să fie unul nu doar funcțional, ci și corect.

În principiu, modelul KNN are urmatorul algoritm:
knn = KNNCosine(training_preferences)
knn.train(k)
indices = knn.fit_indices(fitting_preferences)

În urma rulării acestei secvențe de cod se vor obține vecinii cei mai apropiați utilizatorului cu preferințele alese. Intrând mai adânc în detalii, la nivelul constructorului se binarizează preferințele și se ține cont de aceste detalii, astfel încât să poată fi comparate corect folosind mai multe metrici. Atunci când se apelează funcția train(k), se antrenează modelul folosind metrica aleasă și preferințele binarizate. Indicii utilizatorilor cu preferințele apropiate se obțin prin binarizarea vectorului de preferințe nou inclus (kneighbors([target_user_binary])). Similar se pot obține și distanțele utilizatorilor. Dar ce este de interes în final este vectorul de preferințe similare țintei, care se poate obține prelucrând aceste date menționate.

C. Cosine
cos_sim = np.dot(u, v) / (np.linalg.norm(u) * np.linalg.norm(v))

D. Euclidiană

dist = np.linalg.norm(u - v)

E. Jaccard
intersection = len(u.intersection(v))
union = len(u.union(v))
jaccard_sim = intersection / union

| Preferințele țintei | Metricile evaluate | | |
|---|---|---|---|
| | Cosine | Euclidiană | Jaccard |
| [5, 19, 32] | [[19, 99, 81, 4, 5], [58, 19, 4, 5, 6, 14], [58, 19, 4, 5, 6, 66], [142, 19, 4, 5, 6, 14], [58, 19, 4, 5, 6, 15]] | [[18], [19, 145, 146], [8], [19, 29, 212], [19, 29, 212]] | [[19, 99, 81, 4, 5], [142, 19, 4, 5, 6, 15], [58, 19, 4, 5, 6, 15], [142, 19, 4, 5, 6, 14], [58, 19, 4, 5, 6, 14]] |
| [16] | [[16], [13, 5, 14, 15, 6, 4, 16, 17], [18, 19, 6, 4, 5, 14, 20, 21, 17, 16], [116, 117, 80, 118, 119], [159, 160]] | [[16], [113], [18], [21], [161, 162]] | [[16], [13, 5, 14, 15, 6, 4, 16, 17], [18, 19, 6, 4, 5, 14, 20, 21, 17, 16], [58, 19, 8, 31, 29, 50, 28], [205, 206, 207, 208]] |
| [141, 5, 19, 55, 13] | [[142, 19, 4, 5, 6, 15], [142, 19, 4, 5, 6, 14], [142, 19, 4, 5, 6, 66, 67], [13, 5, 14, 15, 6, 4, 16, 17], [19, 8]] | [[19, 8], [8], [215, 19, 209], [19, 29, 212], [19, 29, 212]] | [[142, 19, 4, 5, 6, 15], [142, 19, 4, 5, 6, 14], [142, 19, 4, 5, 6, 66, 67], [13, 5, 14, 15, 6, 4, 16, 17], [19, 8]] |

# Concluzie

Există numeroase metode pentru a rezolva problemele din viața de zi cu zi, iar abordările ce utilizeaza inteligența artificală nu sunt o excepție. În cadrul proiectului, s-au realizat cu succes recomandările de utilizatori utilizând metricile Jaccard, cosine, și chiar și euclidiană. Există mai multe modaliăți de a recomanda diverși utilizatori în funcție de preferințele lor, dar în momentul de față, abordarea KNN reușește să rezolve problema aceasta corect și cu succes.

Sunt numeroase studii legate de recomandări și inteligență artificială, dar în ciuda acestui fapt, aplicații precum cea realizată în contextul proiectului nu au fost găsite cu succes, în sensul că multe dintre acestea fie realizează recomandările la nivel minimalist, fie au doar o funcționalitate precară, încât utilizatorii recomandați să fie aleși, de fapt, stohastic.

În felul acesta, s-a atins obiectivul dorit, acela fiind de a uni oameni cu preferințe similare în scop educațional, dar nu limitat numai la acest domeniu, putând fi extins ușor și către altele.

# Bibliografie

[1] https://www.frontiersin.org/articles/10.3389/fpsyg.2021.767837/full

[2] https://www.sciencedirect.com/science/article/pii/S0950584921000793?via%3Dihub

[3] https://www.mdpi.com/2227-7390/10/7/1192

[4] https://link.springer.com/article/10.1007/s40747-020-00212-w

[5] https://docs.spring.io/spring-security/reference/servlet/authorization/expression-based.html, ultima accesare: aprilie 2023