Prezentarea are un tipic – vindem marfa xd

10-15 minute, MAXIM – 12 minute GOOD TO FOLLOW!!!

10-15 minute speech ambele situatii – de invatat un speech de vreo 12 minute, estimativ

Unele aplicatii – nu toate (slide 3)

Foarte pe scurt de ce am folosit alea – stocare date, securizare, algoritmi

Baze de date, doar spun ca exista pentru conturi/profiluri, prin sql/nosql

FARA PE PARTEA. Se poate remarca un exemplu de pagina pentru inregistrare

Graficul din figura pune in evidenta...

Slide 1:

Buna ziua. Eu sunt Cornea Radu-Valentin. Astazi va voi prezenta proiectul meu, o aplicație bazată pe microservicii pentru identificarea persoanelor cu interese similare.

Slide 2 (Sumar):

In prezentarea aceasta, dupa introducere, voi trece in revista tehnologiile utilizate, dupa care arhitectura, precum si modulele aplicatiei. La final vor fi remarcate cateva rezultate si concluzii pentru experimentele realizate.

Slide 3 (Introducere. Obiective):

(Trebuie introdus in contextul actual tema mea, dar la obiect).

In principiu, aplicatia are ca drept scop recomandarea de personae. Unei tinte i se recomanda diversi utilizatori, folosind diverse strategii.

(Obiective – un picut mai multe detalii)

In urma unui studiu de piata in care s-au testat mai multe aplicatii de tip recomandari de persoane, s-a constat ca multe dintre acestea nu realizeaza recomandarile bazate pe preferinte asa cum s-ar astepta. De aceea, acest proiect incearca sa acopere golurile celorlalte aplicatii.

Pentru realizarea obiectivelor propuse, s-au folosit microservicii si inteligenta artificiala.

Slide 4 (Tehnologii si concepte teoretice):

Tehnologii – am folosit limbajul pentru ca si pentru ca. Nu intru in detalii. De ce am ales algoritmul, NU algoritmii. Nu insist, mai mult doar exemplific. Nu insist pe teoria pura

Pentru Backend am ales sa folosesc ecosistemul Spring Boot, impreuna cu Kotlin, cateva mecanisme de persistare a datelor, precum si de securizare a acestora.

Pentru Frontend am folosit Thymeleaf, deoarece merge mana in mana cu Spring Boot.

Pentru Inteligenta artificiala, am folosit KNN in scopul gruparii persoanelor in functie de cat de apropiate sunt acestea fata de o tinta, din punct de vedere al preferintelor. Ca limbaj de programare am ales Python cu scikit learn si pandas, incat sa fie mai usor de urmarit experimentele si rezultatele.

Slide 5 (Arhitectura aplicatiei):

Microserviciile au ajutat foarte mult la modularizarea aplicatiei, putand astfel esua independent. Gateway-ul este liantul dintre F si B. El are rolul de a ruta cererile utilizatorilor, precum si de a securiza acele rute in functie de tokenii clientilor, introducand astfel un strat de securizare al aplicatiei.

Slide 6 (Aplicatia Web):

Aplicatia Web a fost realizata cu Thymeleaf, un exemplu de pagina de inregistrare fiind vizibil pe slide. M-am asigurat ca utilizatorul sa poata primi mesaje corespunzatoare atunci cand interactioneaza cu paginile, atat in ceea ce priveste textul, cat si modul in care raspund paginile la actiunile clientului. De exemplu, in form-ul acesta se realizeaza request-uri odata la 0.25 de secunde pe masura ce completeaza field-urile username si email, incat sa nu fie deja luate.

Utilizatorul primeste mesaje corespunzatoare actiunilor sale, de exemplu, daca formularul e completat cu succes, va primi un mesaj pentru asta cand apasa pe butonul de Register, asta daca la nivelul Backend-ului nu survin erori, caz in care va primi un mesaj de eroare si va relua procesul de inregistrare.

Slide 7 (Gestionarea identitatilor)

Odata ce utilizatorul se autentifica cu succes, el va primi un token, bazat pe autoritatile pe care le detine. Tokenul este tinut la nivelul clientului, dar este criptat, incat sa nu poata schimba din autoritatile detinute.

Cat timp token-ul lui este valid, acesta poate accesa rutele din aplicatie, dar doar atat timp cat are autoritatile necesare. La fiecare request trebuie verificata semnatura token-ului, precum si validitatea token-ului si a autoritatilor utilizatorului.

Tokenii acestia criptati la nivelul utilizatorului vor fi decriptati la server, de unde se vor extrage cele necesare. Tot la nivelul server-ului sunt persistati tokenii, tocmai pentru acele situatii in care ar pica server-ul, ei continuand sa existe.

Slide 8 (Profilurile utilizatorilor)

Odata ce un utilizator se inregistreaza, contul lui cu care se autentifica este persistat la nivelul serviciului IDM, dar detaliile acestuia sunt tinute in serviciul Profile, prin MongoDB. Structura profilurilor utilizatorilor poate varia, unii avand mai multe preferinte decat altii. De aceea, s-a ales aceasta abordare.

Slide 9 (Algorithms Service)

Serviciul acesta se ocupa de strategiile de recomandari ale utilizatorilor. In cazul algoritmulului KNN si a metricilor acestora utilizate, se poate stabili strategia la nivelul rutei, prin <strategy>. Apoi, la nivelul serviciului acesta, are loc antrenarea utilizatorilor si potrivirea tintei cu modelul astfel realizat. La final se obtin utilizatorii cu preferintele lor cele mai potrivite cu tinta.

Slide 10 (Experimente)

S-a urmarit testarea metricilor si a valorilor lui k. Alt test efectuat este cel al verificarii preferintelor obtinute si compararea lor cu realitatea. Acest test este important, deoarece se doreste si corectitudinea algoritmilor. S-au realizat mai multe tabele pe baza acestora. In dreapta, de exemplu, este un tabel ce contine valorile metricilor agregate, pentru fiecare k. Mai multe detalii cu privire la experimentele realizate se pot gasi in lucrare.

Slide 11 (Rezultate)

In graficul acesta, au fost rezumate toate experimentele realizate. Se compara valorile lui k si metricile utilizate, in urma a 100 de teste. Pe baza a 159 de utilizatori, au fost obtinute 5 categorii in care se pot incadra diversi utilizatori, de la Very High (foarte similar), la Very Low (foarte putin similar).

Slide 12 (Concluzii)

(Nu conteaza doar ce am spus, ci si cum am spus lucrurile. 2-3 cuvinte rol app in context actual, obiective, daca consider ca le-am atins (< 1 minut) + ce e bun si valoros in aplicatia mea (damn) + o propozitie despre ceea ce pot adauga/facut (super sec de facut – adaug ce si de ce)).

In contextul real, aplicatia reuseste sa recomande unei tinte noi utilizatori, astfel incat acestia sa aiba preferintele cat mai similare. Toate acestea sunt realizate cu succes, multumita microserviciilor si inteligentei artificiale.

Important in aplicatie este obtinerea gasirea mai multi utilizatori similari preferential, dar si de a pune in evidenta diversi algoritmi de inteligenta artificiala in ceea ce priveste sistemele de tip recomandari.

Pe viitor: improve UX cu aplicatia.