Pronalaženje skrivenog znanja - Projektni zadatak za junsko-julski rok 2022. godinu

Projektni zadatak se sastoji iz pet celina na kojima se može ostvariti ukupno 60 poena. Zadaci se odnose na prikupljanje podataka, njihovu analizu, vizuelizaciju i implementaciju algoritama mašinskog učenja.

Zadatak 1: Prikupljanje podataka

Realizovati veb indekser (eng. *web crawler*) sa veb parserom (eng. *web scraper*), koji prikuplja podatke o novim i polovnim automobilima sa jednog ili više od sledećih sajtova:

- https://www.polovniautomobili.com/
- https://www.mojauto.rs/
- https://www.mojtrg.rs/
- neki sajt za ponudu automobila u Srbiji koji nije u ovoj listi, a ima dovoljan broj zapisa.

Formirati sopstvenu relacionu bazu podataka sa svim relevantnim informacijama o automobilima koji se prodaju u Srbiji. Bazu realizovati u tehnologiji *MySQL* ili *PostgreSQL*. Baza treba da ima najmanje 20 hiljada zapisa o automobilima.

Šta je veb indekser?

Cilj veb indeksera je da se poveže na određenu veb stranu i da preuzme njen sadržaj. Parsiranjem date strane možemo naći linkove, koji vode na neke druge strane, na koje veb-indekser ponovo može da uđe i da ponovi celu proceduru. Pored otkrivanja linkova, parser može da prepozna i druge sadržaje koje veb strana ima. U vašu bazu treba da prikupite informacije o svim automobilima – stanje (novo/polovno), marka, model automobila, godište, kilometraža, tip karoserije (npr. hečbek, limuzina, džip/SUV i slično), vrsta goriva (benzin, dizel, itd.), kubikaža, snaga motora, menjač (manuelni ili automatski), broj vrata, boja i lokacija prodavca (mesto). Takođe, uključiti i dodatne informacije koje su dostupne unutar oglasa.

Implementaciju veb-indeksera možete raditi u programskim jezicima: C, C++, C#, Java, Python, NodeJS ili PHP. Dozvoljeno je i korišćenje i prilagođavanje neke od postojećih implementacija otvorenog koda: crawler4j, Heritrix, Nutch, Scrapy, PHP-Crawler, itd.

Zbog ograničenog broja zahteva na serverima sa iste IP adrese, koristiti rotirajuće proxy-je ili neku drugu tehniku za sigurno dohvatanje podataka.

Šta je veb parser?

Uloga veb parsera je da otkrije potreban sadržaj sa primljenih veb strana. Pri tome potrebno je odrediti značenje sadržaja kako bi se baza podataka popunjavala tačnim podacima. Najčešće tehnike koje se koriste pri implementaciji veb parsera su: HTML parser, DOM parser, tehnika regularnih izraza koji izdvajaju potreban sadržaj i tehnika prepoznavanja semantičkih anotacija. Za potrebe veb parsiranja takođe možete koristiti neku od postojećih implementacija (npr. biblioteka jsoup – parsira veb stranu kao stablo elemenata).

Kao rezultat zadatka 1 treba da prikažete realizovanu relacionu bazu podataka popunjenu traženim podacima o automobilima i da priložite implementacije koje su korišćene za dohvatanje podataka. Podaci treba da budu preuzeti u konačnom vremenskom intervalu (ne dužem od 6 sati).

Zadatak 2: Analiza podataka

Iz navedenih zapisa ubačenih u bazu podataka (iz zadatka 1), potrebno je uraditi sledeće:

a) izlistati koliki je broj automobila za svaku od dostupnih marki;
b) izlistati koliko automobila se prodaje u svakom od gradova/lokacija (izlistati sve gradove i pored svakog napisati broj automobila u ponudi);
c) izlistati po bojama (karoserije) koliko je kojih automobila;
d) prikazati rang listu prvih 30 najskupljih automobila koji se prodaju, i 30 najskupljih iz potkategorije džipovi/SUV (ako ovog podatka nema u oglasu, labelirati one modele automobila koji pripadaju ovoj kategoriji);
e) prikazati rang listu svih automobila proizvedenih 2021. i 2022. godine, i izlistati ih opadajuće prema ceni po kojoj se prodaju;
f) prikazati automobil koji imaju:
    - najveću kubikažu,
    - najveću snagu motora,
    - najveću kilometražu (u oglasi gde nema tačne, dati najveći opseg).

Kao rezultat zadatka 2 treba priložiti bazu podataka (zadatak 1), realizovane upite i generisane rezultate (izvoz rezultata uraditi u Word fajl).

Zadatak 3: Vizuelizacija podataka

Iz navedenih zapisa ubačenih u bazu podataka (iz zadatka 1), potrebno je vizuelizovati sledeće podatke:

a) 10 najzastupljenijih lokacija koje imaju najveći broj automobila u ponudi.
b) Broj automobila prema kilometraži (po sledećim klasama: ispod 50 000, 50 000 do 99 999, 100 000 do 149 999, 150 000 do 199 999, 200 000 do 249 999, 250 000 do 299 999, preko 300 000).
c) Broj automobila po godini proizvodnje (starije od 1960, 1961-1970, 1971-1980, 1981-1990, 1991-2000, 2001-2005, 2006-2010, 2011-2015, 2016-2020, 2021-2022)[1].
d) Broj (i procentualni odnos) automobila sa manuelnim ili automatskim menjačem.
e) Broj (i procentualni odnos) svih automobila za prodaju, koje po ceni pripadaju jednom od sledećih opsega:
    - manje od 2000 €,
    - između 2 000 i 4 999 €,
    - između 5 000 i 9 999 €,
    - između 10 000 € i 14 999 €,
    - između 15 000 € i 19 999 €,
    - između 20 000 € i 24 999 €,
    - između 25 000 € i 29 999 €,
    - 30 000 € ili više.

[1] Za sajtove koje nemaju godinu proizvodnje automobila, umesto ovoga izlistati koliko pripada nekoj klasi (nova vozila, polovna vozila).

Kao rezultat zadatka 3 treba priložiti bazu podataka (zadatak 1), realizovane upite i generisane rezultate u vidu grafikona (*charts*). Za grafikone možete koristiti bilo koji alat / implementaciju.

Zadatak 4: Implementacija regresije

Realizovati malu aplikaciju koja na osnovu zapisa iz Vaše baze podataka primenjuje višestruku linearnu regresiju na nekoliko nezavisnih ulaznih promenljivih (npr. koristeći sledeće karakteristike: marka automobila, kubikaža, kilometraža, godina proizvodnje, i dr.) i pravi što bolji model zavisnosti između prediktora i ciljne (izlazne) promenljive. Isprobati različite ulazne promenljive i njihov uticaj na izlaz. Podatke podeliti na skup za treniranje i skup za testiranje, a obučavanje realizovati korišćenjem gradijentnog spusta.

Ciljna promenljiva treba da bude cena automobila za prodaju. Aplikacija treba da na osnovu ulaznih promeljivih koje korisnik (prodavac automobila) treba da unese preko forme i realizovanog modela, prikaže prediktivnu vrednost automobila za prodaju.

U ovom zadatku nije dozvoljeno korišćenje gotovih funkcija iz neke biblioteke programskog jezika, osim u cilju provere ispravnosti sopstvenih rezultata. Sve funkcije treba da budu samostalno napisane.

Zadatak 5: Implementacija klasifikacije

U okviru iste aplikacije, primeniti još i algoritam K-najbližih suseda (kNN) na osnovu istih ulaznih promenljivih (karakteristika automobila), kao u zadatku 4. Opseg izlazne vrednosti (cene automobila) podeliti na nekoliko klasa, kao što je navedeno u zadatku 3.e). Faktor K odrediti automatski (kao najoptimalnije na osnovu broja automobila u skupu koji se razmatra), ali dozvoliti i manuelnu promenu tog faktora, na ulazu, pre pokretanja samog algoritma. Realizovati i bar 2 različite funkcije rastojanja suseda.

Kao rezultat zadataka 4 i 5 treba priložiti programski kod realizovane aplikacije (implementacije realizovanih konačnih modela, procedure za obučavanje, sve realizovane, i eventualno pomoćne funkcije i klase, koje su korišćene). Takođe, priložiti izveštaje sa kratkim komentarom o realizovanim implementacijama, šta ste sve probali da biste došli do finalne implementacije. U zadacima 4 i 5 takođe je poželjno koristiti tehnike vizuelizacije podataka koje smo radili tokom semestra.