

Pronalaženje skrivenog znanja – Projektni zadatak za 2020. 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 indeks (eng. *web crawler*) sa veb parserom (eng. *web scraper*), koji prikuplja podatke o nekretninama sa jednog ili više od sledećih sajtova:

- <https://www.nekretnine.rs/>
- <https://www.halooglasi.com/nekretnine>
- <https://www.4zida.rs/>
- neki sajt za ponudu nekretnina u Srbiji koji nije u ovoj listi, a ima dovoljan broj zapisa.

Formirati sopstvenu relaciju bazu podataka sa svim relevantnim informacijama o nekretninama koje se izdaju i nekretninama koje se prodaju u Srbiji. Bazu realizovati u tehnologiji *MySQL/MariaDB* ili *PostgreSQL*. Baza treba da ima najmanje 20 hiljada zapisa o nekretninama.

Šta je veb indeks?

Cilj veb indeksa 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-indeks 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 nekretninama – tip nekretnine (stan ili kuća), tip ponude (prodaja ili iznajmljivanje), lokacija - grad i deo grada gde se lokacija nalazi, kvadratura nekretnine, godina izgradnje, površina zemljišta (samo za kuće), spratnost (ukupna i sprat na kojoj se nalazi, samo za stanove), uknjiženost (da/ne), tip grejanja, ukupan broj soba, ukupan broj kupatila. Podaci koji nisu dostupni u oglasu, u bazi treba da ostanu praznog polja.

Implementaciju veb-indeksa 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.

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 prikazete realizovanu relaciju bazu podataka popunjenu traženim podacima o nekretninama i da priložite implementacije koje su korišćene za dohvaćanje podataka. Podaci treba da budu preuzeti u konačnom vremenskom intervalu.

Zadatak 2: Analiza podataka

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

- a) izlistati koliko je nekretnina za prodaju, a koliko za iznajmljivanje
- b) izlistati koliko nekretnina se prodaje u svakom od gradova (izlistati sve gradove, obuhvatiti i kuće i stanove)
- c) izlistati koliko je uknjiženih, a koliko neuknjiženih kuća, a koliko stanova
- d) prikazati rang listu prvih 20 najskupljih kuća koje se prodaju, i 20 najskupljih stanova koji se prodaju u Srbiji
- e) prikazati rang listu prvih 100 najskupljih kuća i 100 najskupljih stanova za iznajmljivanje (rentiranje)
- f) prikazati rang listu svih nekretnina izgrađenih u 2019. godini, i izlistati ih opadajuće prema ceni prodaje, odnosno ceni iznajmljivanja
- g) prikazati nekretnine koje imaju:
 - najveći broj soba unutar nekretnine,
 - najveći broj kupatila unutar nekretnine,
 - najveću površinu zemljišta (samo za kuće).

Kao rezultat zadatka 2 treba priložiti bazu podataka (zadatak 1), realizovane upite i generisane rezultate.

Zadatak 3: Vizuelizacija podataka

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

- a) 8 najzastupljenijih delova Beograda koji imaju najveći broj nekretnina u ponudi (i u sekciji za prodaju, i u sekciji za iznajmljivanje, zbirno).
- b) Broj stanova za prodaju prema kvadraturi, u celoj Srbiji (do 35 kvadrata, 36-50, 51-65, 66-80, 81-95, 96-110, 111 kvadrata i više).
- c) Broj izgrađenih nekretnina po dekadama (1950-1959, 1960-1969, 1970-1979, 1980-1989, 1990-1999, 2000-2009, 2010-2019), a obuhvatiti i sekcije za prodaju i za iznajmljivanje.
- d) Broj (i procentualni odnos) nekretnina koje se prodaju i nekretnina koje se iznajmljuju, za prvih 5 gradova sa najvećim brojem nekretnina (za svaki grad posebno prikazati grafikon `BROJ_ZA_PRODAJU : BROJ_ZA_IZNAJMLJIVANJE`).
- e) Broj (i procentualni odnos) svih nekretnina za prodaju, koje po ceni pripadaju jednom od sledećih opsega:
 - manje od 49 999 €,
 - između 50 000 i 99 999 €,
 - između 100 000 i 149 999 €,
 - između 150 000 € i 199 999 €,
 - 200 000 € ili više.

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

Iz dobijenih rezultata izdvojiti samo stanove koji se prodaju u Beogradu. Realizovati malu aplikaciju koja na osnovu zapisa iz Vaše baze podataka primenjuje višestruku linearnu regresiju na nekoliko nezavisnih ulaznih promenljivih (npr. koristeći sledeće attribute: lokaciju - numerički pokazatelj udaljenosti kvarta od centra grada, kvadraturu nekretnine, godinu izgradnje, broj soba, spratnost) i pravi što bolji model zavisnosti između prediktora i ciljne (izlazne) promenljive. Podatke podeliti na skup za treniranje i skup za testiranje, a obučavanje realizovati korišćenjem gradijentnog spusta.

Ciljna promenljiva treba da bude cena nekretnine (stana) za prodaju. Aplikacija treba da na osnovu ulaznih promenljivih koje korisnik (prodavac stana) treba da unese preko forme i realizovanog modela, prikaže prediktivnu vrednost nekretnine 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 (atributa nekretnine), kao u zadatku 4. Opseg izlazne vrednosti (cene nekretnine) podeliti na nekoliko klasa, kao što je navedeno u zadatku 3.e). Faktor K odrediti automatski (kao najoptimalnije na osnovu broja nekretnina za prodaju u skupu Vaših podataka), 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 finalnih 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 moguće koristiti vizuelizaciju podataka.