# Pronalaženje skrivenog znanja
# Projektni zadatak za junsko-julski rok 2024. godine

Projektni zadatak se sastoji iz šest celina na kojima se može ostvariti ukupno 60 poena. Zadaci se odnose na prikupljanje podataka, njihovu analizu, vizuelizaciju i implementaciju algoritama mašinskog učenja. Obavezno je uraditi bar jedan zadatak iz skupa {4, 5, i 6} da biste ostvarili prolazan broj poena.

Projektni zadatak se radi samostalno. Predati projekti se upoređuju međusobno. Studenti kod kojih se utvrdi sličnost dobijaju 0 poena na projektu u ovoj školskoj godini i u ukupnom broju poena (bez obzira da li su izlazili na pismeni ispit) i nemaju prava da izlaze više na odbrane projekata u ovoj školskoj godini. Stoga molimo studente da samostalno rade svoje projekte, kako ne bismo bili prinuđeni da prijave šaljemo Disciplinskoj komisiji.

## Zadatak 1: Prikupljanje podataka

Realizovati veb indekser (eng. *web crawler/web spider*) sa veb parserom (eng. *web scraper*), koji prikuplja podatke o knjigama sa jednog ili više od sledećih sajtova:

- https://laguna.rs/
- https://www.knjizare-vulkan.rs/
- https://bigzknjizara.rs
- neki sajt za ponudu knjiga u Srbiji koji nije u ovoj listi, a ima dovoljan broj zapisa.

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

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 knjigama – naziv knjige, autor(i), žanr/kategorija, izdavač, godina izdanja, broj strana, tip poveza, format, opis i cena. Podaci koji nisu dostupni u opisu, u bazi treba da ostanu praznog polja.

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* za Python, *PHP-Crawler* za PHP, itd.

Zbog ograničenog broja zahteva na serverima sa iste IP adrese, koristiti rotirajuće proxy-je ili neku drugu tehniku, kako ne biste kršili uslove korišćenja usluga izvornih veb sajtova. Prikupljanje ovih podataka koji nisu orijentisani ka ličnim podacima i koji jesu javno dostupni je dozvoljeno, ali uz molbu da broj zahteva ka serveru prilagodite, i da između zahteva bude vremenske razlike, kako ne biste domaćinu sajta izvršili *Denial-of-service attack (DoS)*.

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, *BeautifulSoap*, *Scrapy*, ili *PySpyder*, za Python, itd.).

Kao rezultat zadatka 1 treba da prikažete realizovanu relacionu bazu podataka popunjenu traženim podacima o knjigama i da priložite implementacije koje su korišćene za dohvatanje 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 preprocesirati podatke (odbaciti one koji nemaju veći broj potrebnih vrednosti polja/dopuniti neka polja), zabeležiti to u novoj bazi (sa brojem zapisa koji je preostao, ne manji od 15 hiljada), i uraditi sledeće:

a) izlistati koliki je broj knjiga za prodaju po kategorijama knjige (žanrovima);
b) izlistati koliko knjiga se prodaje od strane svakog izdavača;
c) izlistati koliko knjiga u opisu sadrži reč „ljubav“;
d) izlistati koliko knjiga je izdato po godinama, u poslednjih 7 godina;
e) prikazati rang listu prvih 30 najskupljih knjiga koje se prodaju;
f) prikazati rang listu svih knjiga izdatih u 2023. ili 2024. godini, i izlistati ih rastuće prema ceni prodaje;
g) prikazati knjige (Top30) koje imaju:
    - najveći broj strana,
    - najveću cenu,
    - najveći format.

Kao rezultat zadatka 2 treba priložiti bazu podataka (revidiranu i prečišćenu, iz zadatka 2), realizovane upite i generisane rezultate (npr. izvesti u *Excel* ili *Word* fajl).

## Zadatak 3: Vizuelizacija podataka

Iz navedenih zapisa ubačenih u bazu podataka (iz zadatka 2), potrebno je vizuelizovati sledeće podatke:

a) 10 najzastupljenijih izdavača koji imaju najveći broj knjiga u ponudi.
b) Broj knjiga po kategorijama (žanrovima).
c) Broj izdatih knjiga po dekadama (1961-1970, 1971-1980, 1981-1990, 1991-2000, 2001-2010, 2011-2020, 2020-danas).
d) Broj (i procentualni odnos) knjiga koje se prodaju, za prvih 5 izdavačkih kuća sa najvećim brojem knjiga.
e) Broj (i procentualni odnos) svih knjiga za prodaju, koje po ceni pripadaju jednom od sledećih opsega:
    - manje ili jednako od 500 dinara,
    - između 501 i 1500 dinara,
    - između 1501 i 3000 dinara,
    - između 3001 i 5000 dinara,
    - između 5001 i 10000 dinara,
    - između 10001 i 15000 dinara,
    - 15001 ili više.
f) Broj (i procentualni odnos) knjiga za prodaju koje imaju tvrd povez, u odnosu na ukupan broj knjiga za prodaju (izdatih u poslednje 3 godine).

Kao rezultat zadatka 3 treba priložiti bazu podataka (iz zadatka 2), realizovane upite i generisane rezultate u vidu grafikona (*charts*). Za grafikone možete koristiti bilo koji alat / biblioteku, a izvoz uraditi kao slike.

## Zadatak 4: Implementacija regresije

Realizovati malu aplikaciju koja na osnovu zapisa iz Vaše filtrirane baze podataka primenjuje višestruku linearnu regresiju na nekoliko nezavisnih ulaznih promenljivih 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. Ulazni atributi (*features*) koje možete analizirati mogu biti: autor, žanr/kategorija, izdavač, godina izdanja, broj strana, informaciju o povezu i formatu.

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

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 logističke regresije na osnovu istih ili sličnih ulaznih promenljivih (atributa knjige) i na osnovu potpuno iste (filtrirane) baze podataka, kao u zadatku 4. Opseg izlazne vrednosti (cene knjige) podeliti na nekoliko klasa, kao što je na primer navedeno u zadatku 3.e). Za rešavanje multiklasne klasifikacije, implementirati i jednu od opcija kombinovanja binarnih klasifikatora („jedan nasuprot svima“ ili „jedan nasuprot jednog“), kao i multinomijalnu logističku regresiju. Korisnik može da izabere pristup prilikom unosa atributa knjige.

## Zadatak 6: Implementacija klasterizacije

U neophodnom broju iteracija, primeniti metod K-srednjih vrednosti (*k-Means*), na najmanje 3 ulazna atributa koje ćete posmatrati iz filtrirane baze podataka sa knjigama (iz Zadatka 4).

Kao rezultat zadataka 4, 5 i 6 treba priložiti programski kod realizovane aplikacije ili aplikacija (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 i koji su sve dobijeni rezultati. U ova tri zadatka takođe je poželjno koristiti vizuelizaciju podataka i grafikone (samo potpuno urađeni zadaci donose najveći broj poena!).