

**Napomena:**

Na Desktop-u se nalazi direktorijum `vi.sept1`. Preimenujte ga u oblik `VI_Ime_Prezime_BrojIndeksa_GodinaUpisa`. Na primer, student Nikola Stojanović sa brojem indeksa 283/2015 treba direktorijum `vi.sept1` da preimenuje u `VI_Nikola_Stojanovic_283_2015`.

U ovom direktorijumu ostavite Vaše rešenje.

Na Desktop-u možete pronaći dokumentaciju za potrebne Python biblioteke.

Vreme za rad: 2 sata

1. Palačinke različitih veličina su naslagane na tanjir. Mali Perica želi da sortira palačinke po veličini od najmanje do najveće. Perica na raspolaganju ima samo spatulu kojom može okrenuti sve palačinke od neke pozicije naslaganih palačinki do palačinke na vrhu.

Koristeći BFS algoritam pretrage pronaći najkraći put od početnog redosleda palačinki do završnog, sortiranog, redosleda palačinki na tanjiru. Redosled palačinki u tanjiru predstavlja jedno stanje i reprezentovano je nizom čiji elementi redom odgovaraju veličinama palačinki na tanjiru.

Primer:

Pocetno stanje:	Završno stanje:	Najkaraci put:
[1, 2, 5, 3, 4]	[1, 2, 3, 4, 5]	[[1, 2, 5, 3, 4], [1, 2, 5, 4, 3], [1, 2, 3, 4, 5]]

2. Napisati prolog predikat `prijatelji` koji rešava sledeću zagonetku:

- Četiri prijatelja se bave razlicitim poslovima i imaju različite hobije.
- Rajko je stolar.
- Molerov hobi je slikanje.
- Ni vodoinstalater, ni Danko se ne bave planinarenjem.
- Janko, koji nije trgovac, za hobi ima pecanje.
- Markov hobi je skijanje.
- Ko se bavi kojim poslom i kojim hobiem?

3. Data je signatura  $\mathcal{L}(\Pi, \Sigma, ar)$ , pri čemu je:

- $\Pi = \{b, r, s\}$
- $\Sigma = \{\}$
- $ar(b) = 2, ar(r) = 2, ar(s) = 2$

Metodom rezolucije pokazati da je sledeći skup formula nezadovoljiv:

- $(\forall x)(\forall y)(r(x, y) \Rightarrow s(x, y))$
- $(\forall x)(\forall y)\neg s(x, y)$
- $(\forall x)(\forall y)(\exists z)(b(x, y) \Rightarrow (r(z, x) \wedge r(z, y)))$
- $(\exists x)(\exists y)b(x, y)$

U datoteci `sablon/3_logika/resenje.txt` ostaviti korake algoritma.

Za simbole u tekstualnoj datoteci koristiti:

- **A** - univerzalni kvantifikator
- **E** - egzistencijalni kvantifikator
- **!** - negacija
- **&** - konjunkcija
- **|** - disjunkcija
- **=>** - implikacija
- **<=>** - ekvivalencija

4. Koristeći linearnu regresiju iz biblioteke `scikit-learn` potrebno je napraviti klasifikator koji pogađa težinu u zavisnosti od visine čoveka. U datoteci `sablon/4_regresija/X.npy` nalaze se podaci o visini 90 pacijenata dobijenih od lekara sa sistematskog pregleda, a u datoteci `sablon/4_regresija/y.npy` podaci o njihovim visinama.

Izvršiti regresiju nad podacima pri čemu za test skup treba uzeti  $\frac{1}{4}$  ukupnih podataka.

Na standardnom izlazu prikazati:

- Ukupan broj instanci
- Broj instanci u skupu za trening
- Broj instanci u skupu za testiranje
- Srednje kvadratnu grešku za trening i test skup
- Metriku  $R^2$  za trening i test skup
- Nacrtati podatke (kao na slici)

Ukupno instanci: 90

Ukupno instanci za trening: 67

Ukupno instanci za testiranje: 23

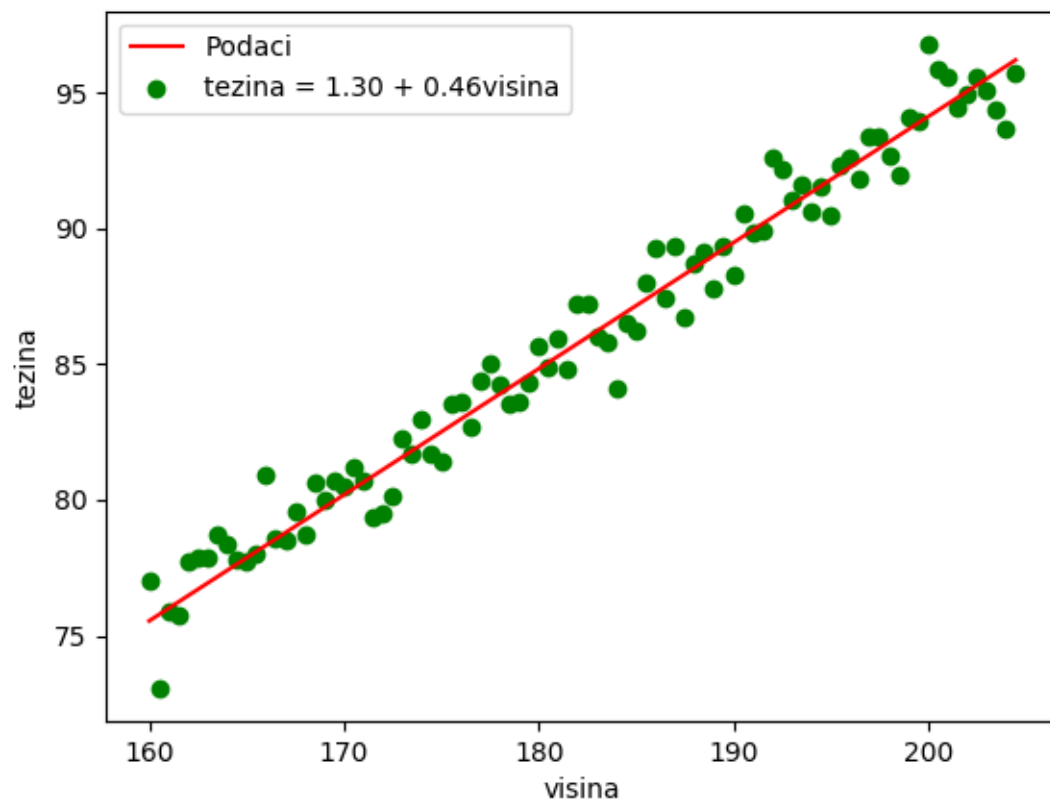
Dobijeni model:  $\text{tezina} = 1.30 + 0.46\text{visina}$

MSE trening = 1.0725029287659078

MSE test = 1.168714073249866

$r^2$  trening: 0.9716836991048453

$r^2$  test: 0.9617696016556625



Slika 1: Ilustracija dobijenog modela linearne regresije