Mašinsko učenje

Prvi domaći zadatak

27.3.2020.

Prvi domaći zadatak sastoji se od četiri nezavisna problema, opisana na narednim stranama. Uz svaki problem tj. njegov deo stoji odgovarajući broj poena, a ukupan broj poena je 100 (što se kasnije skalira na 25 predispitnih poena).

Pri izradi domaćeg zadatka moguća je saradnja studenata u grupama od najviše troje. Pritom, svaki student će biti nezavisno ocenjivan na osnovu odbrane pri kojoj se proverava razumevanje predatog rešenja i relevantnog gradiva. Iako je podela posla dozvoljena, ukoliko jedan član tima ne razume neke delove predatog rešenja, za te delove će mu biti dodeljeno 0 poena. Ukoliko postoji deo predatog rešenja koji niko iz tima ne razume, svim članovima tima će biti dodeljeno 0 poena na celom domaćem zadatku.

Domaći zadatak se izrađuje i predaje isključivo na sledeći način:

1. Downloadovanje arhive **ml_d1_x_y_z.zip** koja se nalazi uz ovaj dokument. U ovoj arhivi se nalaze svi potrebni podaci, a već su kreirani **.py** i ostali fajlovi koji čine rešenje.

2. Popunjavanje fajlova u skladu sa zahtevima datim u problemima, bez kreiranja dodatnih fajlova.

3. Zapakivanje foldera **ml_d1_x_y_z** u arhivu, pri čemu treba zameniti slova x/y/z brojevima indeksa (u formatu RN-br-god) članova tima.

4. Slanje fajla **ml_d1_indeks1_indeks2_indeks3.zip** na mejl adrese mmilunovic@raf.rs ili iciganovic@raf.rs pre isteka roka. Subject mejla mora biti u obliku “[ML D1] prezime1 prezime2 prezime3”. U tekstu mejla obavezno navesti članove tima sa brojevima indeksa.

Rok za slanje rešenja je utorak 10. april u 23:59.

Odbrana domaćeg zadatka će biti zakazana naknadno.

U slučaju nepoštovanja bilo kog od navedenih pravila (naslov mejla, ime arhive, rok za slanje) rad neće biti pregledan i svim članovima tima će biti dodeljeno 0 poena.

Na narednim stranama nalazi se opis problema sa jasnim smernicama koje fajlove treba popuniti i šta njihovo pokretanje treba da da kao izlaz. Svi problemi su uradivi korišćenjem znanja sa časova i uz malo samostalnog istraživanja. Naravno, dozvoljeno je koristiti kod sa vežbi (dokle god shvatate šta on zapravo radi), ali nije dozvoljeno koristiti kompletna rešenja direktno kopirana sa interneta. U slučaju da ima pitanja/nedoumica pošaljite mejl na mmilunovic@raf.rs ili iciganovic@raf.rs a možemo organizovati i konsultacije po dogovoru.

Problem 1: Istraživanje [15p]

Ovaj problem zahteva od vas da istražite i na osnovnom nivou razumete neke nove pojmove koji možda nisu pominjani na nastavi ali su u bliskoj vezi sa gradivom. Rešenja za svako od tri pitanja dati u nekoliko rečenica, u fajlu **1.txt**.

a. [5p] Koja je razlika između *k-fold*, *leave one out* i *random subsampling* cross validation algoritama?

b. [5p] Objasniti razliku između *Gaussian*, *Multinomial* i *Bernouli* Naive Bayes metoda.

c. [5p] Šta je "*linearna separabilnost*" (*linear separability*)? Da li su podaci iz skupa **iris.csv** linearno separabilni (objasniti šta se primećuje)?

Problem 2: Regresija [20p]

U arhivi se nalazi skup podataka **corona.csv** specifično kreiran za potrebe ovog problema.

a. [8p] U fajlu **2a.py** na ovom skupu podataka primeniti polinomijalnu regresiju, uz variranje stepena polinoma u intervalu [1, 6]. Pokretanje programa treba da proizvede dva grafika: jedan na kome su u 2D prikazani svi podaci iz skupa kao i svih 6 regresionih krivih, i drugi na kome je prikazana zavisnost finalne funkcije troška na celom skupu (ne u poslednjoj epohi treninga!) od stepena polinoma. Šta možemo primetiti? Diskutovati u komentaru ispod koda.

b. [7p] U fajlu **2b.py** trenirati polinomijalnu regresiju sa fiksnim stepenom polinoma 3, ali uz dodatu L2 regularizaciju. Za parametar *lambda* probati vrednosti iz skupa {0, 0.001, 0.01, 0.1, 1, 10, 100}. U redu je krenuti od kompletne kopije prethodnog fajla. Pokretanje programa treba da kreira dva grafika slična onima u prethodnom delu problema: grafik svih podataka sa 7 regresionih krivih (za različite vrednosti *lambda*) i grafik zavisnosti finalne funkcije troška na celom skupu od parametra *lambda*. Šta sada možemo primetiti? Diskutovati u komentaru ispod koda.

c. [5p] na mesto fajla **2c.png** sačuvati *tensorboard* prikaz grafa izračunavanja za model iz fajla **2b.py**. Imenovati promenljive kako bi bile prepoznatljive u grafu. Na samoj slici označiti deo koji računa funkciju troška.

Problem 3: k-NN [25p]

U arhivi se nalazi skup podataka **Prostate_Cancer.csv** koji se bavi predviđanjem maligniteta tumora na osnovu više parametara. Cilj je na osnovu osam takvih svojstva izvršiti klasifikaciju u jednu od 2 kategorije.

a. [10p] U fajlu **3 a.py** podeliti iris skup podataka na trening deo i test deo. Nakon toga primeniti netežinsku verziju k-NN algoritma uzimajući u obzir samo **prva četiri** feature-a, za k=3. Pokretanje programa treba da primeni k-NN i ispiše *accuracy* na test skupu. Takođe, treba da kreira i prikaže 2D grafik na kome su prikazani trening podaci, pri čemu su različite klase obojene različitom bojom. Na istom grafiku prikazati oblasti koje

bivaju klasifikovane u svaku od klasa (hint: vrlo sličan grafik je kreiran na časovima u sklopu softmax regresije).

b. [10p] U fajlu **3b.py** i dalje koristiti prva četiri feature-a ali za vrednost parametra *k* birati brojeve od 1 do 15. U redu je krenuti od kompletne kopije prethodnog fajla. Pokretanje programa treba da prikaže grafik zavisnosti *accuracy* metrike na test skupu u odnosu na vrednost parametra *k*. Koje *k* je najbolji izbor? Diskutovati u komentaru ispod koda.

c. [5p] U fajlu **3c.py** krenuti od kopije prethodnog fajla ali ovaj put uključiti svaki feature prisutan u originalnom fajlu. Pokretanje treba ponovo da prikaže isti grafik kao u prethodnom delu. Uporediti ta dva grafika u komentaru ispod koda.

Problem 4: Rad sa tekstom / Naive Bayes [40p]

U arhivi se nalazi datoteka **twitter.csv** koja sadrži skupa tvitova, čija je glavna primena klasifikacija tvita u dve klase: pozitivne i negativne. Za ovaj problem izdvojeno je 100.000 tvitova koji su označeni u 2 klase (pozitivnu i negativnu), i očekuje se da se koriste samo ovi podaci. Kompletno rešenje za ovaj problem (sva tri dela) treba uneti u fajl **4.py**. Pokretanje ovog fajla treba da izvrši sve pomenuto u nastavku problema i ispiše sve relevantne rezultate.

a. [30p] Očistiti skup podataka i zatim kreirati feature vektore metodama po izboru.

   Podeliti skup podataka na trening i test skup (po odnosu 80:20). Fitovati Multinomial Naive Bayes model. Neophodan je *accuracy* na test skupu od barem 75% (prosečan u tri uzastopna pokretanja programa).

b. [5p] Kreirati matricu konfuzije (matrica [[TN, FP], [FN, TP]]).

c. [5p] Pronaći 5 najčešće korišćenih reči u pozitivnim tvitovima. Isto uraditi i za negativne i prokomentarisati rezultate (u komentaru koda). Ako uvedemo metriku *LR(reč)* kao ***LR(reč) = br. poj. u poz. tvitovima (reč) / br. poj. u neg. tvitovima (reč)*** pronaći 5 reči sa najvećom i 5 reči sa najmanjom vrednošću ove metrike. Metrika se definiše samo za reči koje se barem 10 puta pojavljuju u pozitivnom, i 10 puta u negativnom korpusu, nakon čišćenja podataka. Prokomentarisati 10 ovako dobijenih reči, uporediti sa prethodnim rezultatima, i objasniti značenje metrike *LR* u komentaru ispod koda.

Hint: Obratite posebnu pažnju na čišćenje podataka. Evaluirajte dobijene “čiste” podatke dok ne dođete do dovoljno kvalitetne metode čišćenja za ovaj skup podataka.

Hint: Ukoliko koristite BoW pokušajte da limitirate vokabular na 10000 najčešće korišćenih reči u celom skupu podataka kako feature vektori ne bi bili previše dugački.