Mašinsko učenje

Prvi domaći zadatak

29.3.2022.

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​ agavric@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 nedelja 24. april u 23:59**​.

Odbrana domaćeg zadatka će biti održana u terminu prvih vežbi posle isteka roka. Ako bude potrebe biće zakazan dodatni termin.

U slučaju nepoštovanja bilo kog od navedenih pravila (rok za slanje, ime arhive i sl.) 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 agavric@raf.rs ili iciganovic@raf.rs​ a možemo organizovati i konsultacije po dogovoru.

# Problem 1: Istraživanje [15p]

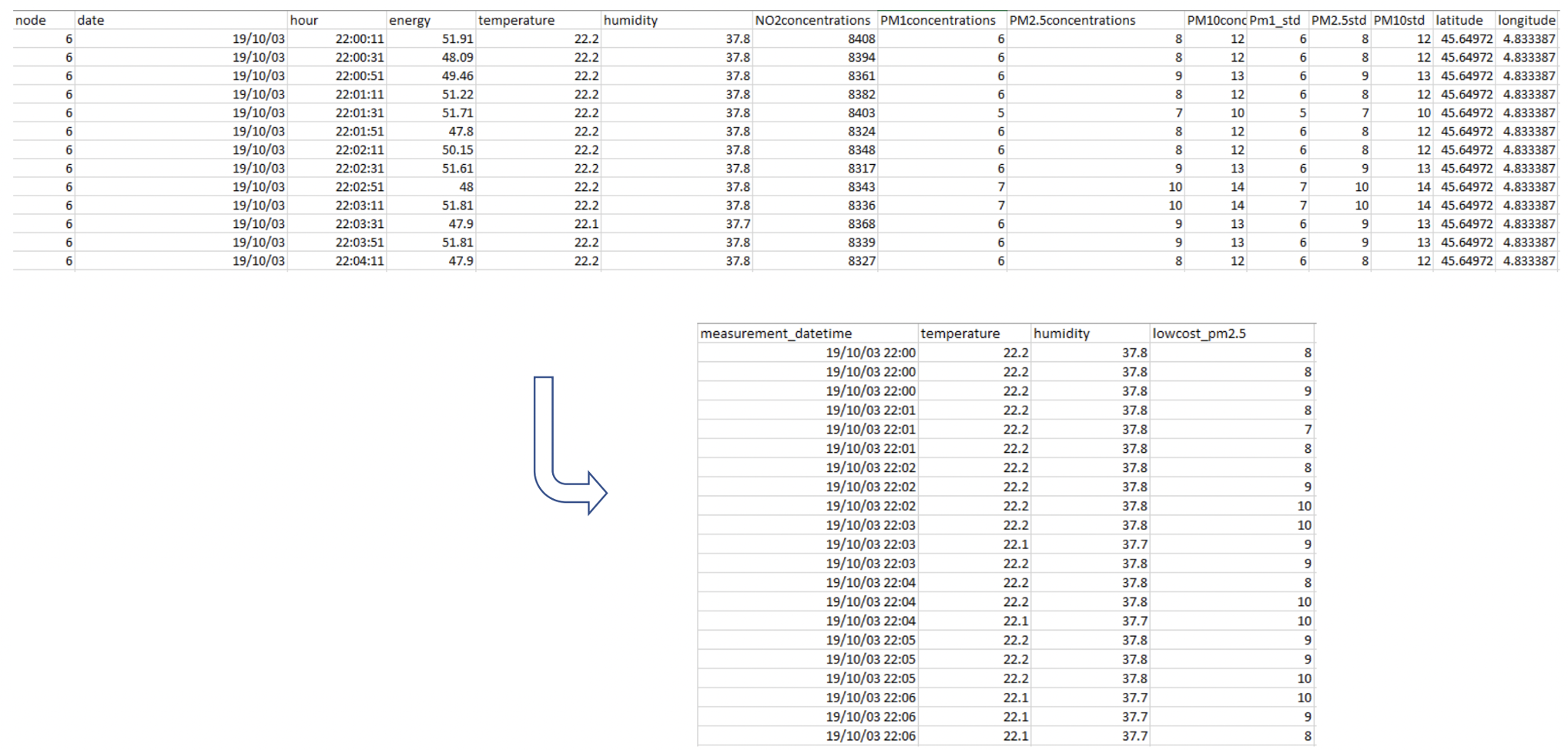
Ovaj problem zahteva od vas da istražite i na osnovnom nivou razumete neke nove pojmove koji možda nisu pominjani na nastavi ali su u bliskoj vezi sa gradivom. Rešenja za svako od tri pitanja dati u nekoliko rečenica, u fajlu ​**1.txt**​.

1. [5p] Šta su, kako rade i čemu služe *„normalizacija“* i *„regularizacija“*?
2. [5p] Objasniti razliku između *„Precision“* i *„Recall“* metrika i dati primer u kom slučaju nam koja više znači.
3. [5p] Navesti 3 primera u kojima izrazito želimo da imamo što manje FP čak i na račun povećanja FN i 3 primera kada važi obrnuto – želimo da imamo što manje FN čak i na račun povećanja broja FP.

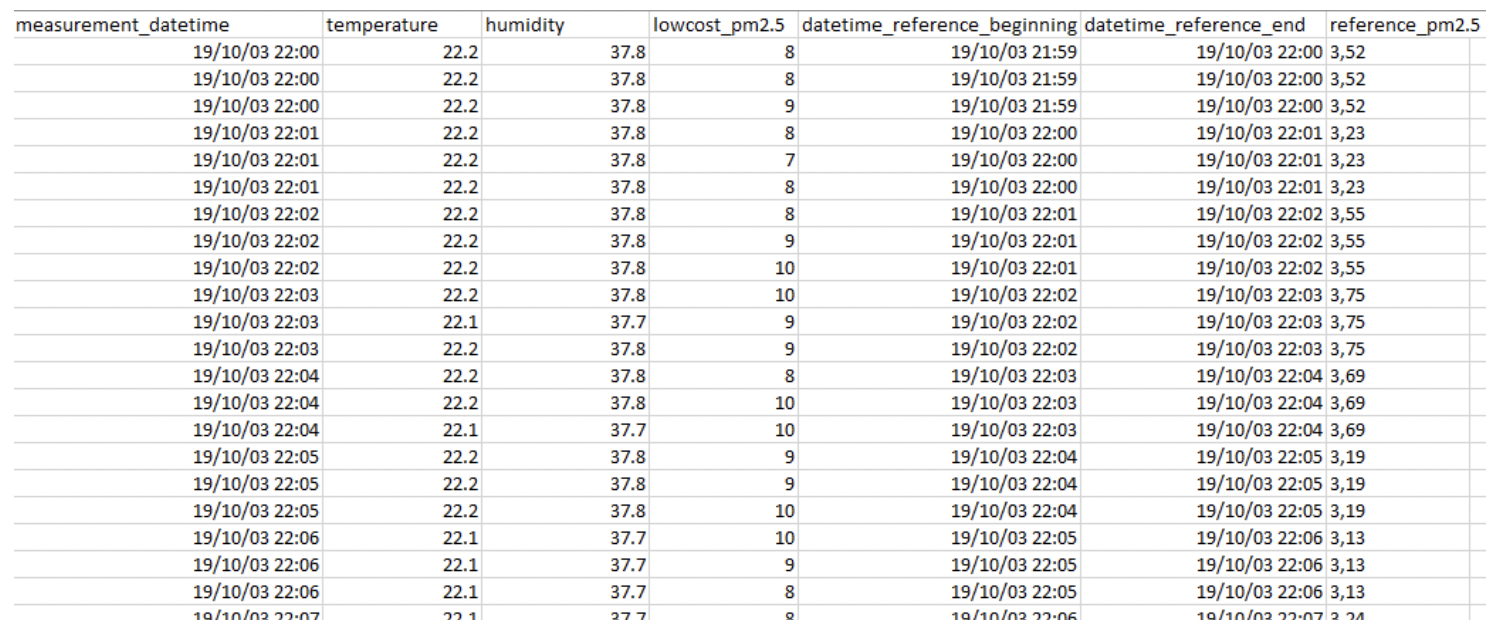
# Problem 2: Regresija [25p]

U arhivi se nalaze skupovi podataka ​**skupi\_senzori.xls** i **jeftini\_senzori.txt** koji sadrže merenja sa senzora za zagađenje vazduha.

1. [7p] Potrebno je izmeniti odnosno ispraviti oba fajla kako bi mogli da se učitaju kao *dataframe*-ovi korišćenjem biblioteke *„pandas“*. Kolone u fajlu jeftini\_senzori su redom: id senzora (treba ignorisati), datum\_merenja; vreme\_merenja; stanje baterije (treba ignorisati) ; temperatura ; vlažnost vazduha; NO2 koncentracija (treba ignorisati), PM1 koncentracija (treba ignorisati), PM2,5 koncentracija; PM10 koncentracija (treba ignorisati); Pm1\_std (treba ignorisati) ; PM2,5 std (treba ignorisati); PM10 std (treba ignorisati) , latituda (treba ignorisati), longituda (treba ignorisati). Primetite da je datum u ova dva fajla dat u različitim formatima koje treba dovesti do istog oblika a potom napraviti zajednički skup podataka koji će sadržati kolone koje ne ignorišemo iz oba fajla. Treba napraviti spoj redova tako da se **t = (datum\_merenja + vreme\_merenja)** iz jeftini\_senzori.txt poklopi sa svim redovima iz skupi\_senzori.xls za koje važi da je **t** između vrednosti b = (date beginning + time beginning) i e = (date end + time end). Vremena u time end i time beginning treba pomeriti u vremensku zonu GMT-2 (ako imate kolonu koja je po tipu *datetime* možete je prosto sabrati sa pd.DateOffset(hours=-2)). Postupak kreiranja dataseta napisati u fajlu ​**2a.py** a dataset priložiti na mestu fajla **2a.xls**. Ilustracije su date na Slici 2.1 i Slici 2.2.
2. [8p] U fajlu ​**2b.py**​ na dobijenom 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 sa po jednim 2D plotom prikazani podaci iz skupa za vlaznost, temperaturu i lowcost PM2.5 i njihove regresione krive sa kolonom PM2.5\_ambient na drugoj osi, i drugi na kome je prikazana zavisnost finalne funkcije troška na celom skupu (ne u poslednjoj epohi treninga!) od stepena polinoma. Treba predvideti PM2.5 merenja sa skupog senzora na osnovu temerature, vlažnosti vazduha, PM2.5 merenja sa jeftinog senzora. Šta možemo primetiti? Diskutovati u komentaru ispod koda.
3. [7p] U fajlu ​**2c.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.
4. [3p] Na mesto fajla ​**2d.png** ​sačuvati ​tensorboard​ prikaz grafa izračunavanja za model iz fajla ​**2c.py​.** Imenovati promenljive kako bi bile prepoznatljive u grafu. Na samoj slici označiti deo koji računa funkciju troška.



Slika 2.1. Primer transformacije skupa podataka Jeftini\_senzori.txt



Slika 2.2. Primer finalnog skupa podataka koji treba dobiti spajanjem početna dva

# Problem 3: k-NN [25p]

U arhivi se nalazi skup podataka **social\_network\_ads.csv** ​koji prikazuje parametre korisnika i deli ih u dve kategorije prema tome da li su kupili proizvod iz reklame na sajtu. Cilj je na osnovu tri takva parametra izvršiti klasifikaciju u jednu od te dve kategorije.

1. [10p] U fajlu **3**​ **a.py** podeliti skup podataka na trening deo i test deo. Nakon toga primeniti netežinsku verziju k-NN algoritma uzimajući u obzir samo ​**poslednja dva**​ 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).
2. [10p] U fajlu ​**3b.py** ​i dalje koristiti poslednja dva 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.
3. [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 [35p]

U arhivi se nalazi datoteka ​**fake\_news.csv**​ koja sadrži skup podataka artikala iz novina koji su podeljeni u dve klase: pouzdani i nepouzdani. 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.

1. [25p] 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).

1. [5p] Kreirati matricu konfuzije (matrica [[TN, FP], [FN, TP]]).
2. [5p] Pronaći 5 najčešće korišćenih reči u pouzdanim člancima. Isto uraditi i za nepouzdane i prokomentarisati rezultate (u komentaru koda). Ako uvedemo metriku *LR(reč)* ​kao ​***LR(reč) = br. poj. u pouzdanim (reč) / br. poj. u nepouzdanim (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 pouzdanom, i 10 puta u nepouzdanom 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.