

Mašinsko učenje

Drugi domaći zadatak

15.4.2025.

Drugi domaći zadatak sastoji se od tri 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_d2_x_y_z.zip** koja se nalazi uz ovaj dokument. U ovoj arhivi se nalaze svi potrebni podaci.
2. Rešenja sačuvati u odgovarajućim direktorijumima u skladu sa zahtevima datim u problemima. Dozvoljena je upotreba Google Colab platforme. U tom slučaju se umesto .py fileova šalje download-ovan colab notebook. Obavezno sačuvati izlaz izvršavanja programa (ili kao tekstualni file ili u colabu) i po potrebi treniran model (samo jedan koji predstavlja rešenje). Nemojte slati data set i slike generisane u simulatoru.
3. Zapakivanje foldera **ml_d2_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_d2_indeks1_indeks2_indeks3.zip** na mejl adresu iciganovic@raf.rs, pre isteka roka. Subject mejla mora biti u obliku "[ML D2] prezime1 prezime2 prezime3". U tekstu mejla obavezno navesti članove tima sa brojevima indeksa.

Rok za slanje rešenja je nedelja 4. maj u 23:59.

Odbrana domaćeg zadatka će biti zakazana naknadno.

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 urađivi 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 iciganovic@raf.rs a možemo organizovati i konsultacije po dogovoru.

Problem 1: Stabla odlučivanja [30p]

U arhivi se nalazi skup podataka **titanic** sa 2 filea **train.csv** i **test.csv** u kojima se nalaze trening i test podaci. Detaljan opis podataka i cilj istraživanja možete naći [ovde](#).

Potrebno je istrenirati model stabla odlučivanja da klasifikuje putnike prema tome da li su preživeli ili ne. Dozvoljeno je i očekivano po potrebi srediti podatke.

Problem 2: Neuralna Mreža [30p]

Na istom data setu iz prethodnog zadatka potrebno je istrenirati model neuralne mreže da klasifikuje putnike prema tome da li su preživeli ili ne. Neuralnu mrežu implementirati manuelno, u formi matrica, a ne koristeći Keras biblioteku. Dozvoljeno je i očekivano po potrebi srediti podatke.

Problem 3: Behavioral Cloning [40p]

U arhivi u direktorijumu **3** se nalazi simulator automobila. Postoje 3 verzije za različite operative sisteme. U simulatoru se vrši kreiranje (snimanje) data seta i testiranje treniranog modela. Pored simulatora nalaze se i 3 .py filea. U fileu **utils.py** se pored pomoćnih funkcija nalazi promenljiva sa veličinom ulazne slike (**INPUT_SHAPE**) i funkcija **batch_generator** koje treba koristiti. File **drive.py** se koristi za testiranje treniranog modela. Pokretanjem filea **model.py** se izvršava trening. U direktorijumu se takođe nalazi **requirements.txt** file koji sadrži sve potrebne biblioteke za izradu ovog zadatka. Preporuka je napraviti virtuelno okruženje i instalirati ove zahteve. Verzije moraju da ostanu kako su navedene u zahtevima inače neće raditi. Ako radite lokalno, a ne na Colabu, možete koristiti ovo okruženje i za ostale zadatke.

Potrebno je u fileu **model.py** popuniti kod koji fali u **build_model** i **train_model** funkcijama. Treba koristiti **Keras** model i **ModelCheckpoint** za snimanje modela u toku treninga. Model treba snimati na putanji **MODEL_SAVE_PATH** koja je zadata. Dozvoljeno je koristiti sve Keras layere koji postoje. Model kao ulaz prima sliku sa kamere vozila zadatih dimenzija, a kao izlaz treba da da jedan realan broj koji predstavlja ugao volana.

Nakon što napišete model i trening treba pokrenuti simulator. Izabrati levu stazu i kliknuti na **Training Mode**. Kada se otvori simulacija kliknuti na record u gornjem desnom uglu. Zatim izabrati **data** direktorijum kao putanju za snimanje data seta. Zatim ponovo kliknuti record dugme da bi snimanje počelo. Tada je potrebno ručno voziti auto dok traje snimanje. Kontrole pišu u simulatoru. Kada ste zadovoljni sa količinom snimljenih podataka ponovo kliknuti record dugme da se prekine snimanje i sačekati da se proces završi.

Tada treba pokrenuti **model.py** da se izvrši trening. U kodu možete videti command line argumente koje možete proslediti prilikom pokretanja i koji vam mogu biti korisni. Pokretanje ovog filea treba da snimi istreniran model na zadatoj putanji.

Kada je trening gotov pokrenuti **drive.py** file sa putanjom do modela koji želite da koristite prilikom testiranja. (npr. `python drive.py -m model/model-001.h5`) Kada se server pokrene treba pokrenuti simulator i ovaj put izabrati **Autonomous Mode**. Tada bi trebalo na konzoli da se ispiše da je server primio konekciju i auto bi trebalo da počne sam da vozi. Dovoljno je da ume voziti samo levu stazu. Ne mora obe.

Trening je moguće izvršiti na Colabu. Snimanje podataka i testiranje se mora izvršavati lokalno.