

Napomena: Na Desktop-u preimenovati direktorijum `mu.postavke` u oblik `mu.Prezime_Ime_Indeks` (npr. `mu.Peric_Pera_mi12082`). Unutar tog direktorijuma pokrenuti jupyter server komandom `jupyter lab` ili `jupyter notebook`.

1. (a) Učitati podatke iz datoteke `heart.csv` i prikazati prvih 5 instanci. Koliko ukupno podataka postoji? Koliko atributa postoji? Za ciljnu promenljivu smatrati atribut `target` (0 - pacijent nema srčanu bolest, 1 - pacijent ima srčanu bolest).
- (b) Nacrtati grafikon koji vizualizuje ciljnu promenljivu i pol. Na x osi umesto brojeva označiti vrednosti atributa (*zdrav* i *srcana bolest* za ciljnu promenljivu, i *musko*, *zensko* za pol).
- (c) Enkodirati kategoričke promenljive (`cp`, `thal`, `slope`). Razdvojiti podatke na podatke bez ciljne promenljive (`x`) i ciljnu promenljivu (`y`).
- (d) Izvršiti podelu podataka na trening i test podatke tako da je test skup 20% ukupnih podataka. Za `random_state` koristiti vrednost 0. Izvršiti stratifikaciju po ciljnoj promenljivoj. Izvršiti standardizaciju podataka.
- (e) Napraviti niz klasifikatora koji sadrži modele logističke regresije, kernelizovanog SVM-a i nasumičnih šuma odlučivanja. Napraviti niz rečnika koje predstavljaju skup parametara za pretragu i to:
 - Logistička regresija:
 - `C`: [0.01, 0.1, 1, 10, 100],
 - SVC:
 - `C`: [0.01, 0.1, 1, 10, 100],
 - `kernel`: [linear, poly, rbf, sigmoid]
 - Slučajnih šuma odlučivanja:
 - `n_estimators`: [2, 5, 10, 20, 100],
 - `criterion`: [gini, entropy]

Koristeći `GridSearchCV` odabrati najbolju konfiguraciju za svaki od klasifikatora i na standardnom izlazu napisati odabrane parametre.

- (f) Napraviti novi glasački klasifikator koji se sastoji od najboljih konfiguracija prethodno odabranih klasifikatora, obučiti ga na podacima za obučavanje i dati njegovu ocenu (klasifikacioni izveštaj i matricu konfuzije) za trening i test podatke.

NAPOMENE:

- `GridSearchCV` ne dozvoljava isprobavanje više različitih modela prilikom svojeg rada, dozvoljeno Vam je da više puta pokrenete `GridSearchCV` za različite modele.
 - Glasačkom klasifikator prihvata niz uređenih parova gde je prvi element u paru string koji predstavlja ime (oznaku) klasifikatora, a drugi element uređenog para je sam klasifikator. Odnosno `List[Tuple, Classifier]`, a ne `List[Classifier]`.
2. (a) Učitati `cifar10` podatke. Kao podatke za obučavanje uzeti prvih 1000 podataka iz učitanih podataka za obučavanje, a kao podatke za testiranje uzeti prvih 100 podataka iz podataka za testiranje.
 - (b) Odabrati nasumično 9 različitih instanci iz skupa za obučavanje i napraviti crtež dimenzija 3×3 na kojima ih je potrebno prikazati.
 - (c) Izvršiti normalizaciju piksela i enkodiranje ciljne promenljive.
 - (d) Napraviti konvolutivno neuronsku mrežu sa sledećom arhitekturom:
 - Konvolutivni sloj sa 32 filtera, ReLu aktivacijom i veličinom kernela 3×3
 - Konvolutivni sloj sa 32 filtera, ReLu aktivacijom i veličinom kernela 3×3
 - Agregirajući sloj sa veličinom prozora 2×2
 - Regularizacija izostavljanja (Dropout) sa parametrom 0.25
 - Konvolutivni sloj sa 64 filtera, ReLu aktivacijom i veličinom kernela 3×3
 - Konvolutivni sloj sa 64 filtera, ReLu aktivacijom i veličinom kernela 3×3
 - Agregirajući sloj sa veličinom prozora 2×2
 - Regularizacija izostavljanja (Dropout) sa parametrom 0.25
 - Sloj koji izravna dimenzije (Flatten)
 - Potpuno povezani sloj sa 512 neurona i ReLu aktivacijom
 - Potpuno povezani sloj sa brojem neurona koliko postoji klasa i aktivacijom mekog maksimuma
 - (e) Izvršiti obučavanje neuronske mreže. Kao broj paketa pri obučavanju postaviti 32, za broj epoha uzeti 10. Podatke za testiranje postaviti kao validacione podatke.
 - (f) Nacrtati grafik koji prikazuje kako se menja tačnost (accuracy) na podacima za obučavanje i testiranje tokom obučavanja mreže. Nacrtati grafik koji prikazuje kako se menja greška na oba skupa prilikom obučavanja mreže. Na *x* osi označiti epoha, a na *y* osi označiti odgovarajuću meru. Na oba grafika dodati legendu koja prikazuje šta koja funkcija označava (odnosno trening i test).