NSU: DN1

Na voljo sta dve nalogi. Izberite si eno od njiju, jo rešite in oddajte kodo ter kratko poročilo. Delali boste s podatkovjem `podatki.csv`, ki je bilo vzeto s strani `https://www.openml.org/`, nato pa malo spremenjeno:

1. Izbrisane so bile nekatere značilke, vrstni red preostalih pa je bil premešan.
2. Izbrisani so bili nekateri primeri, vrstni red preostalih pa je bil premešan.
3. Vrednosti stolpcev so bile spremenjene, tako da ne morejo bistveno vplivati na lastnosti podatkov.

Na koncu smo vse stolpce še preimenovali. Ciljna spremenljivka v podatkih nosi ime `y`.

1 Metaučenje

Vedite, da za število izbrisov velja naslednje:

- Izbrisali smo največ 100 značilk.
- Izbrisali smo največ 1000 primerov.

Izberite si primerne metaznačilke, s katerimi opišete podatkovja. V prostoru, ki ga te značilke razpenjajo, s pomočjo `openml.org` najdite tri podatkovja, ki so najbolj podobna temu iz `podatki.csv`. Za iskanje sosedov lahko uporabite algoritem k najbližjih sosedov ($k = 3$). Zgornji opombi vam lahko pomagata pri pripravi metapodatkovja.

Ko najdete sosede, ugotovite, kateri algoritem za klasifikacijo se najbolje odreže na njih. Tako dobite največ tri različne kandidate – za vsakega soseda enega.

Primer iskanja kandidatov. Uporabimo zavihek `Tasks` na `openml.org`. Če je eden od odkritih sosedov podatkovje `iris`, izvedemo poizvedbo `https://www.openml.org/search?q=tasktype.tt_id%3A1+iris&type=task&sort=runs&order=desc`, ki najde vse naloge klasifikacije na podatkovju iris in jih uredi po številu poganjanj (*run*-ov). Izberite enega od zadetkov z veliko *run*i in najdite najboljši algoritem. Za iris je to metoda podpornih vektorjev (SVC).

Izberite enega od treh kandidatnih algoritmov in ga poženite na `podatki.csv`. Njegovo zmogljivost ocenite tako, da podatke razbijete na učno (75%) in testno množico (25%).

Oddano poročilo naj vsebuje:

- metaznačilke, ki ste jih zgenerirali,
- imena treh najbližjih sosedov,
- najboljši algoritem za klasifikacijo za vsakega od treh sosedov,
- utemeljitev izbire končnega algoritma za klasifikacijo,
- klasifikacijsko točnost končnega algoritma na `podatki.csv`.

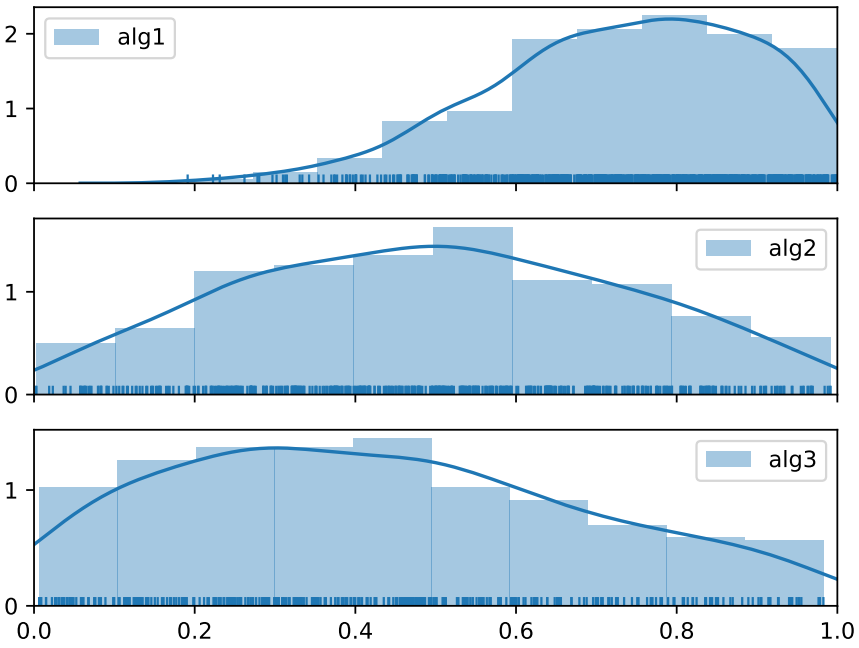
2 AutoML

Podatke razbijte na učno (75%) in testno množico (25%). Izberite vsaj tri klasifikacijske algoritme. Med njimi naj ne bo vrečenja dreves (ang. *bagging*a). Vsak od njih naj ima vsaj en parameter, katerega optimalno vrednost je treba najti. Vsaj en od algoritmov naj ima več kot le en parameter. Določi še smiselne kandidate (ali pa njihov prostor) za vrednosti parametrov in z uporabo **učnega dela** podatkov najdi najboljšo konfiguracijo.

Zmogljivost izbranega algoritma z optimalno konfiguracijo parametrov preizkusi na testni množici. Primerjaj jo z zmogljivostjo vrečenja dreves. Za oba algoritma izračunaj klasifikacijsko točnost.

Oddano poročilo naj vsebuje:

- natančen opis preiskovanega prostora konfiguracij,
- najboljšo konfiguracijo,
- graf porazdelitev klasifikacijskih točnosti (ki jih porodijo različne konfiguracije) za vsakega od algoritmov: pomagaš si lahko z `automl_primer.py`, s katerim je bila zgenerirana Slika 1. Grafe komentiraj.
- klasifikacijsko točnost najboljše konfiguracije in vrečenja na `podatki.csv`.



Slika 1: Prikaz porazdelitev točnosti za vsakega od treh algoritmov.