# NSU prva domača naloga

Tine Markočič

April 2023

## 1 Izbira metode in optimizacija hiperparametrov

### 1.1 Ročno

Podatke sem najprej pretvoril v pravi tip, tako da sem vrednosti spremenljivke $y$ nadomestil z 0 in 1. Nato sem podatke razdelil na učno in testno množico. Najprej sem poskusil z metodo najbližjih sosedov. Ker je ta priredila veliko napako, sem poskusil še z odločitvenim drevesom, ki je bolje delovalo. Z metodo grid search sem dobil najbolša parametra:

"max_depht" = 8, "min_samples_split" = 82.

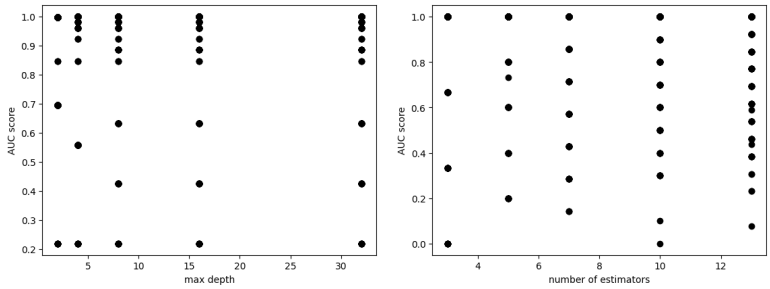
### 1.2 Avtomatizirano

Med seboj sem primerjal metode *naključnega gozda*, *odločitvenega drevesa* in *podpornih vektorjev*. Pri odločitvenem drevesu sem optimiziral enaka paramtra kot zgoraj ter dodal še parameter "min_samples_leaf", ki določa minimalno število vzorcev, ki jih mora drevo pustiti preden se loči v naslednjo vejo. Za nakljuni gozd sem izbral parametra "n_estimators", za število dreves v gozdu in čriterium", ki skrbi za kvaliteto delitve. Za podporne vektorje sem iskal najboljši parameter C, ki določa regularizacijo.

Po optimizaciji s knjižnico *hyperopt*, se je izkazal za najboljšega model naključnega gozda s parametroma:

"n_estimators" = 13, "criterium" = "entropy".

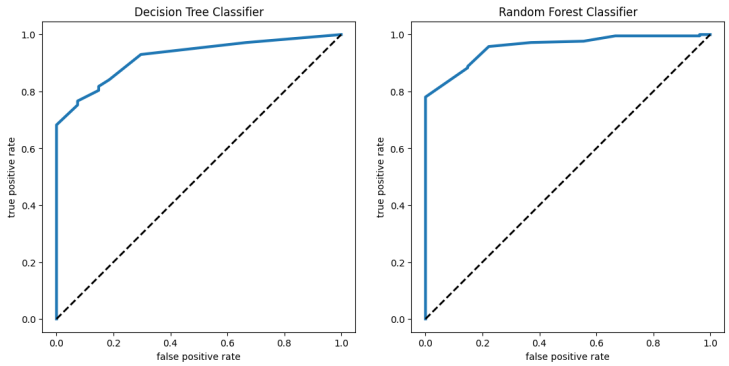
Iz porazdelitev zmogljivosti algoritmov odločitvenega drevesa in naključnega gozda vidimo, da prvi ni tako odisen od hiperparametra "max_depth", saj so je že pri vrednosti 4 dobra verjetnost, da bomo dobili dober model. Pri naključnem gozdu pa je število dreves ključen hiperparameter in večji kot bo, večja bo verjetnost, da dobimo dober model.




Slika 1: Vrednosti AUC za algoritma odločitveno drevo in naključni gozd

Tako ročni kot avtomatiziran model sta priredila dobri zmogljivosti, in sicer
ročni: 0.9212530287296643
avtomatiziran: 0.9564728279681552

Tu pa sta še njuni ROC krivulji.




**Opomba:** Za najzmogljivejšega se je izkazal algoritem naključnih gozdov z velikim številom dreves. Vendar pa je iskanje takega števila preveč časovno zahtevno, zato sem se zadovoljil z dobljeno zmogljivostjo.

# 2 Meta učenje

Iz pridobljenih podatkov sem obdržal le tiste, ki so primerni za klasifikacijo, nato pa sem z orodjem *pymfe* določil meta značilke tipa "general"in "info-theory". Vrednosti značilk sem določil še za podatke iz podatki.csv. Nato sem z algoritmom NearestNeighbors poiskal tri podatkovja, ki so najbolj podobna izhodiščnim podatkom.

To so

- ibm-employee-performance
- amazon-commerce-reviews_seed_0_nrows_2000_nclasses_10_ncols_100_stratify_True
- amazon-commerce-reviews_seed_3_nrows_2000_nclasses_10_ncols_100_stratify_True

Pri iskanju najboljših klasifikacijskih problemov za ta podatkovja sem naletel na zame nerešljive težave, zato mi tega dela naloge ni uspelo rešiti.