Poročilo - domača naloga 1

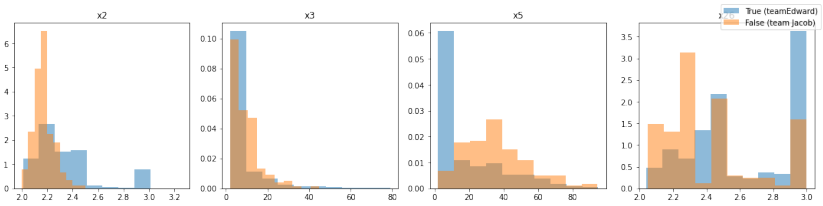
Urh Primožič

31. marec 2023

1 Izbira metode in optimizacija parametrov

1.1 Ročna izbira

Pri izbiri modela sem si pomagal z delnimi porazdelitvami značilk glede na vrenost ciljne spremenljivke.



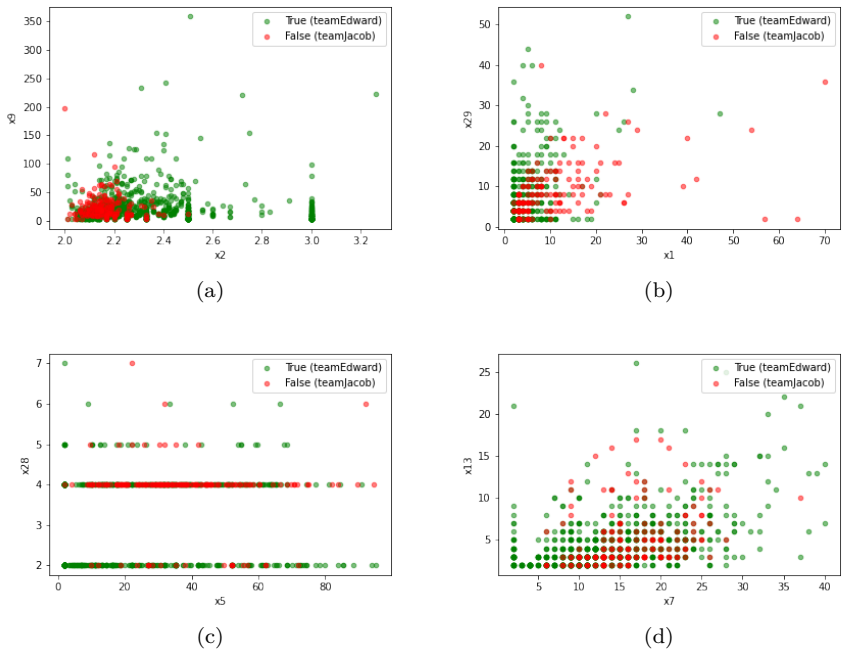
Slika 1: Nekatere porazdelitve posameznih značilk. Opazimo razlike v porazdelitvah razredov.

Iz histogramov porazdelitev posameznih spremenljivk in grafov raztrosa sem sklepal, da bi bila smiselna uporaba modelov, ki prostor geometrijsko razdelijo na odseke. Zato sem izbiral med metodo podpornih vektorjev in odločitvenimi drevesi.

Primerjal sem uspešnost obeh modelov s privzetimi parametri. Namesto odločitvenega drevesa sem rajši uporabil GBM drevesa, saj so običajno boljša od enega drevesa ali naključnih gozdov. Ploščina pod ROC podpornih vektorjev je bila 0.5, pod krivuljo napovedi dreves pa 0.9. Za model sem izbral gradient-boosting drevesa, njegove parametre pa sem izbral s prečnim preverjanjem in optimizacijo z naključnim izbiranjem parametrov. Ploščina pod ROC krivuljo na testni množici je bila 0.924.

1.2 Avtomatizirana izbira

Avtomatizirane izbire sem se lotil s knjižnico *ray.tune*, ki nudi hitro in učinkovito implementacijo optimiziranja parametrov. Za izbiranje parametrov med optimizacijo sem uporabil TPE in modul *hyperopt*.



Slika 2: Grafi raztrosa posameznih značilk. Vidne so geometrijske lastnosti posameznih razredov. K geometrijskim metodam pristopam z idejo, da se razreda *teamEdward* in *teamJacob* z večanjem dimenzije vse bolj delita.

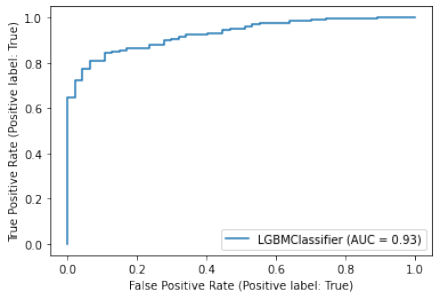
1.2.1 Preiskovalni prostor

Algoritem je izbiral med GBM, odločitvenim drevesom, najbljižjimi sosedi in SVC.

- podprostor parametrov pri drevesih GBM (`lightgbm.LGBMClassifier`):
  - možne vrednosti števila listov: $\{9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196\}$
  - prostor števila dreves: $\{9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361\}$
  - Najmanjše dovoljeno število primerov v listu: $\{5, 10, 15, 25, 50, 100, 200, 500\}$

parameter	vrednost
`model`	`lightgbm.LGBMClassifier`
`learning_rate`	0.0274
`min_child_samples`	10
`n_estimators`	81
`num_leaves`	67

Tabela 1: Vrednosti parametrov ročno izbranega modela.



Slika 3: ROC krivulja izbranega modela

  - hitrost učenja learning_rate $\sim$ Loguniform$(10^{-2}, 0.05)$

- prostor parametrov odločitvenega drevesa (`sklearn.tree.DecisionTreeClassifier`):
  - največja globina : $\{9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361\}$
  - minimalno število primerov, potrebnih za delitev vozlišča: $\{2, 5, 10, 15, 25, 50, 75, 100\}$
  - minimalno število potrebnih primerov za list: $\{2, 5, 10, 15, 25, 50, 75, 100\}$
- prostor parametrov najbližjih sosedov (`sklearn.neighbours.KNeigboursClassifier`):
  - število sosedov: $\{1, 2, 3, 4, 5, 10, 15, 25\}$
- Pri podpornih vektorjih (`sklearn.svm.SVC`) sem izbiral koeficient $C \sim$ Lognormal$(0, 1)$ in sledeča jedra:
  - linearno
  - rbg - v tem primeru je koeficient γ porazdeljen lognormalno $\gamma \sim$ Lognormal$(0, 1)$
  - polinomsko (stopnje 2)

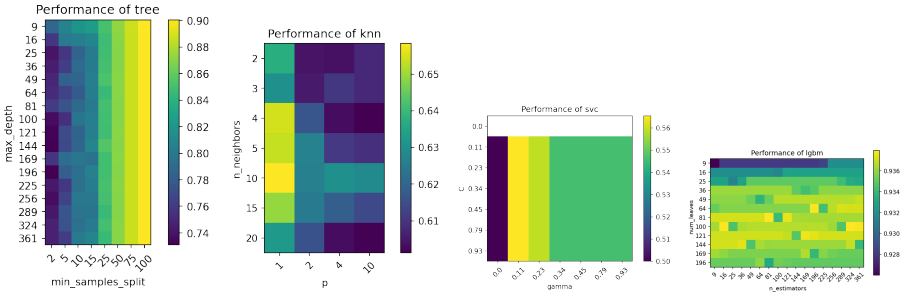
Vzorčil sem 250 različnih izbir parametrov.

1.2.2 Najboljša konfiguracija

Model z največjo ploščino pod ROC so bila GB-drevesa s sledečimi parametri

parameter	vrednost
`model`	`lightgbm.LGBMClassifier`
`learning_rate`	1.046
`min_child_samples`	50
`n_estimators`	324
`num_leaves`	49

Ploščina pod ROC krivuljo tega modela je bila enaka 0.929.



(a) (b) (c) (d)

Slika 4: Uspešnost modelov glede na izbire parametrov

1.3 Porazdelitev zmogljivosti

Grafi porazdelitev namigujejo na kompleksnost podatkov. SVC in KNN ne glede na parametre napovedujeta slabo, drevesu in LGBM pa s pravo izbiro lahko zmogljivost drastično izboljšamo. Globina drevesa in število dreves v GBM ne vplivata na zmogljivost. Bolj pomembno je število listov pri GBM in minimalna vrednost primerov v vozlišču pri drevesu. Ker so za oba parametra boljše nižje vrednosti, sklepam, da je prostor značilk razdeljen na manjše število klasifikacijskih regij.

1.4 Primerjava z ročno konfiguracijo

Rezultati ročne in avtomtizirane izbire so podobi. Model so v obeh primerih GB drevesa, s podobnimi nastavitvami. Avtomatizirana izbira bi verjetno dala boljše rezultate, če bi parametre vzorčil večkrat. Poleg tega so modeli z drevesnimi strukturami že s privzetimi parametri delovalo mnogo boljše kot SVC in KNN, zato bi bilo slednje smiselno odstraniti iz prostora konfiguracij, kar bi povečalo število vzorčenj parametrov dreves.

2 Meta učenje

Za meta značilke sem izbral vse značilke iz skupin general, statistical, info-theory in complexity. Nato sem odstranil vse z manjkajočimi vrednostmi na podatkovjih. Na koncu sem uporabil značilke `attr_conc.mean`, `attr_conc.sd`, `attr_ent.mean`, `attr_ent.sd`, `attr_to_inst`, `cat_to_num`, `class_conc.mean`, `class_conc.sd`, `class_ent`, `eq_num_attr`, `freq_class.mean`, `freq_class.sd`, `joint_ent.mean`, `joint_ent.sd`, `mut_inf.mean`, `mut_inf.sd`, `nr_attr`, `nr_cor_attr`, `nr_norm`, `nr_outliers`, `ns_ratio`, `sparsity.mean` in `sparsity.sd`.

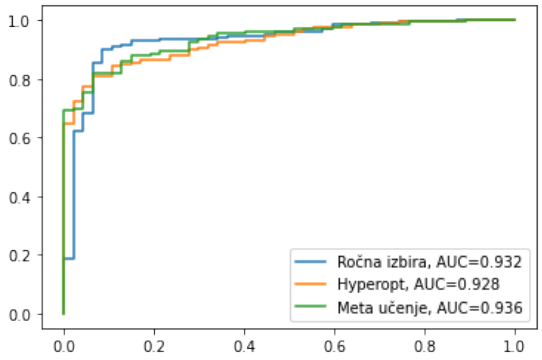
2.1 Najbližji sosedi

Sosede sem izbiral med podatkovji iz OpenML s številom značilk med 20 in 140 in številom primerov med 1103 in 2103. Na ta način sem zajel vse podatkovja s podobno obliko kot jo imajo naši podatki, tudi originalno podatkovje.

V prostoru meta značilk so trije najbližji sosedi naših podatkov podatkovja z identifikacijskimi številkami 1504, 44383 in 44386. Vsi trije algoritmi uporabljajo drevesno strukturo. Odločitveno drevo je bila ena od možnih konfiguracij avtomatiziranega modela, ki pa ga je prekosil GBM. Za test sem model, pridobljen z avtomatizirano izbiro, primerjal še z naključnim gozdom.

2.2 Model, pridobljen z meta učenjem

Parametre naključnih gozdov (`sklearn.ensemble.RandomForestClassifier`) sem izbral kot v 1.2. Ploščina pod ROC krivuljo pri izbranih parametrih je bila enaka 0.936. Rezultati vseh treh modelov so podobni, kar je pričakovano, saj



Slika 5: ROC krivulje vseh treh modelov.

vsi trije modeli temeljijo na drevesih. Zanimivo je, da so bili tokrat naključni gozdovi bolj uspešni od gradient-boosted dreves. Po meta učenju bi se odločil za drugačno izbiro konfiguracijskega prostora, in sicer bi izbiral le med naključnimi gozdovi, GBM in odločitvenimi drevesi.

ime	pomen
attr_conc.mean	koncentracijski koeficient
attr_conc.sd	koncentracijski koeficient
attr_ent.mean	shannonova entropija značilk
attr_ent.sd	shannonova entropija značilk
attr_to_inst	razmerje med števili atributov
cat_to_num	razmerje med številom diskretnih in numeričnih spremenljivk
class_conc.mean	koncentracijski koeficient med razredom in značilko
class_conc.sd	koncentracijski koeficient med razredom in značilko
class_ent	shannonova entropija ciljne spremenljivke
eq_num_attr	število ekvivalentnih atributov
freq_class.mean	relativna frekvenca razrediv
freq_class.sd	relativna frekvenca razrediv
joint_ent.mean	entropija med vsako značilko in razredom
joint_ent.sd	entropija med vsako značilko in razredom
mut_inf.mean	skupna informacija med značilkami in ciljnimi spremenljivkami
mut_inf.sd	skupna informacija med značilkami in ciljnimi spremenljivkami
nr_attr	število atributov
nr_cor_attr	število različnih parov atributov z veliko korelacijo
nr_norm	število normalno porazdeljenih atributov
nr_outliers	število atributov z izstopajočimi vrednostmi
ns_ratio	delež šuma
sparsity.mean	redkost podatkov

Tabela 2: Pomeni uporabljenih meta—atributov

id	najboljši model
1504	odločitveno drevo
44383	naključni gozd
44386	naključni gozd

Tabela 3: Caption