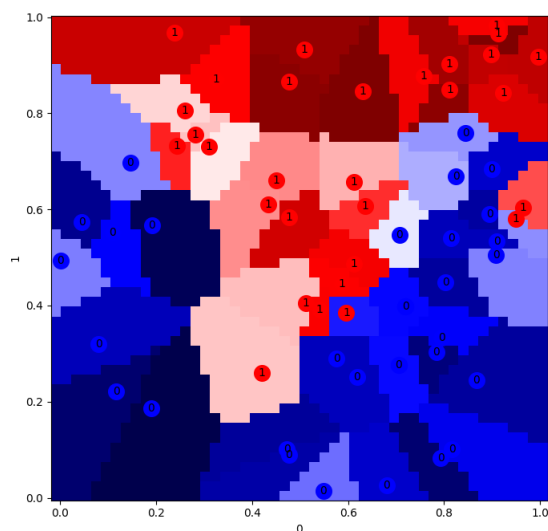


**Regularizacija.** Za regularizacije sem si izbral 3 različne vrednosti  $\lambda$ : 0.01, 0.001 in 0.0001. Najboljši rezultat dobim z vrednostjo 0.001. Tu se najbolje vidijo meje med razredoma.

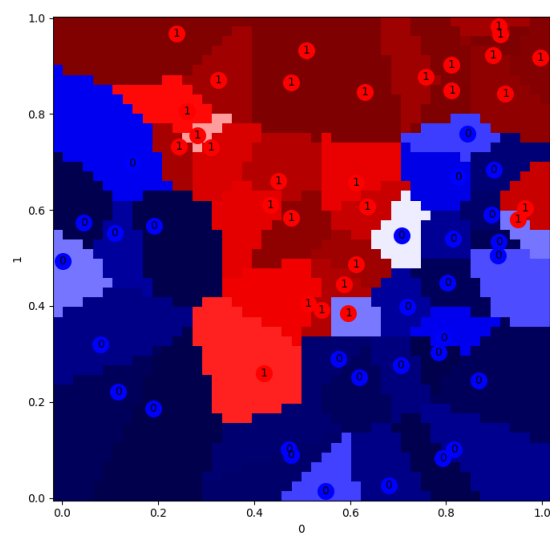
Namreč naše značilke imajo veliko več dimenzij (27), kot pa jih imamo mi na voljo za prikaz/vizualizacijo podatkov. To pomeni, da bomo v večdimenzionalnem prostoru lažje določili mejo med razredoma s hiperravnino, v 2D prostoru pa si lahko pomagamo na drug način.

V tem primeru lahko pri 0.001 vrednosti  $\lambda$  vidimo različne verjetnosti (odtenke barv) za primere na robu. Tako na nek način ohranjamo "moč" meje logistične regresije pri prikazu le-te v prostoru s precej manj dimenzijami.

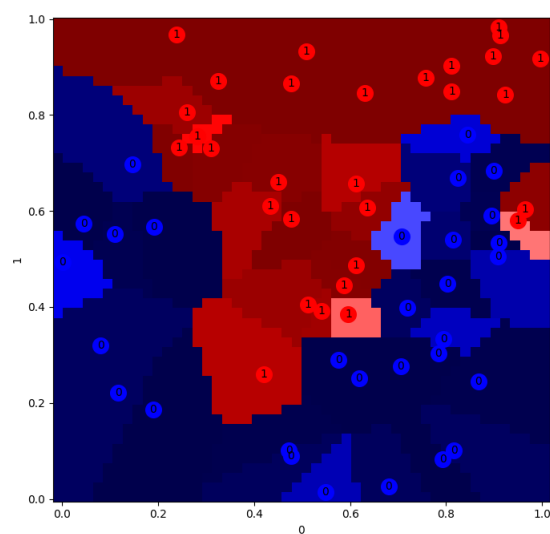
Rezultat je tako smiseln, saj pri  $\lambda=0$  ne vidimo nobenega prehoda med razredoma.



Slika 1: Napoved pri  $\lambda = 0.01$



Slika 2: Napoved pri  $\lambda = 0.001$



Slika 3: Napoved pri  $\lambda = 0.0001$

**Točnosti.** Koda je shranjena v funkciji `lambde()`.

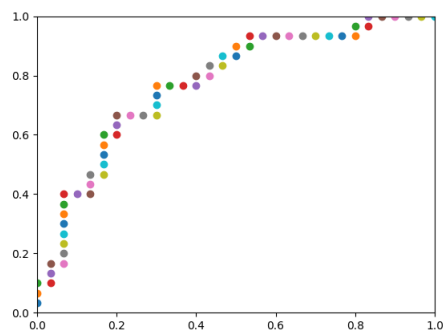
Najboljši rezultat je pri  $\lambda = 0.001$ , saj je tam dobra ocena CA, pa tudi točnost ocenjena s prečnim preverjanjem, ki je bolj odporna mera na pristranskost podatkov (in na overfitting testnih podatkov).

Tabela 1: Ocene točnosti CA in CV predikcij pri različnih vrednostih lambda.

lambda	točnost CA	točnost CV (cross validation/prečno preverjanje)
0.00001	1	0.667
0.0001	1	0.683
0.001	0.967	0.7
0.01	0.95	0.683
0.1	0.917	0.683
1	0.867	0.5
10	0.767	0.483
100	0.767	0.5
1000	0.75	0.5
10000	0.75	0.5

**Krivulja ROC.** AUC je implementiran vendar deluje z manjšo napako. Napaka nastane, ker sem podatke prefiltriral tako, da sem izpuštil primere z enako verjetnostjo klasifikacije v več kot 1 razred.

Pravilno bi bilo, da bi tudi te podatke upošteval in izračunal ploščino trikotnika, ki pri takih primerih nastane pri vizualizaciji ROC krivulje. Tako bi tudi test deloval pravilno, trenutno pa ne deluje, oz deluje z 0.001 natančnostjo.



Slika 4: ROC krivulja

**Izbor značiln.** /