5. domača naloga: Tekmovanje "DMC 2016"

Jernej Henigman (strežnik: fa0)

9. maj 2016

1 Uvod

Cilj domače naloge je bil pridelati, kar se da dobre napovedne vrednosti na prirejenih podatkih, ki jih ponuja spletna stran DMC 2016.

2 Metode

- **Obdelava podatkov**
  Podatkom odstranim vse nize. Pridobim numpy matriko, velikosti št. atributov x N. Kjer je N število primerov. Vrednosti, ki vsebujejo negativno vrednost ali N/A odstranim, ali pa nadomestim z ustreno zamenjavo. Učno in testno množico shranim v pickle objeckt, da je branje hitrejše.

- **Feature engineering**
  Poskušal sem z vključevanjem in izključevanjem različnih stolpcev (14 atributov). Ustvarim tudi dodaten atribut in sicer verjetnost, da posamezen kupec vrne artikel. To sem naredil tako, da sem pogledal razmerje med številom kupljenih/vrnjenih artiklov za posamezenga kupca (costumerID). Če kupca v testnih podatkih ni, nadomestim to vrednost, s povprečno vrednostjo verjetnosti vrnjenega artikla vseh kupcev. Za najboljši rezultat kombinacij atributov se izkažejo naslednji atributi [productGroup, price, rrp, costumerID, verjetnostDaVrne].

- **Strojno učne metode**
  Večino napovedi sem pridelal z metodo RandomForestClassifier iz knjižnice sklearn. Z metodo RandomForest pridobim tudi najboljši rezultat na tekmovalnem strežniku in sicer 0.36918. Poskušal sem še z naivnim baysesom, vendar se zaradi zamudne pretvorbe posameznih atributov v binarno-vektorsko obliko nisem resneje posvetil tej metodi. Poskušal sem tudi z metodo SVM, vendar je bila zelo neučinkovita in počasna.

3 Rezultati

Model	Št. atributov	Napaka
RF	14	0.44060
NB	1 - binariziran	0.42929
RF	1	0.40929
RF	2	0.38416
RF	5	0.37484
RF	5	0.37133
RF	5	0.36918
SVM	5	timeOut

Tabela 1.1. Napake na tekmovalnem strežniku. NB in SVM sta bila testirana lokalno.

Za učenje modela uporabimo med 20-40 odstotkov učnih primerov. Če jih uporabim več kot 80 odstotkov se rezultati ne izboljšajo (se celo poslabšajo). Ko sem popravil maksimlano globino gradnje drevesa, ter izbral najboljše atribute sem se spustil pod mejo 0.38. Da sem se spustil pod mejo 0.37 sem se še malo poigral z nastavitvijo parametrov pri naključnem gozdu. Da bi se spustil pod mejo 0.36 bi moral spremeniti kaj bolj fundamentalnega (nov atribut, dodatna ansambelska metoda, ...)

4 Poganjanje, testirnaje

Privzete poti do učnih in testnih podatkov data/train.txt ter data/test.txt, če sta datoteki poimenovani drugače je potrebno poti popraviti v programu in sicer pri klicih funkcij, parseTrainData("pot/imeDatoteke") parseTestData(pot/imeDatoteke"). Vse ostalo naredi program avtomatsko. Čas izvajanja programa na celotnih učnih in testnih podatkih je na procesorju i7-2679QM CPU 2.20GHz okrog 10min.