Fakultet organizacionih nauka, Beograd – Otkrivanje zakonitosti u podacima

**„Ansambl algoritmi“**

**Damir Pajaziti,**

04. April 2020.

# Uvod

U ovom radu, obrađene su različite ansambl tehnike. Neki od algoritama su detaljnije obrađeni, s obzirom da su se pokazali kao jedni od najboljih na poznatim *data science* takmičenjima. Jedan od takvih primera je i XGBoost algoritam. Nakon što je svaka tehnika objašnjena koracima algoritma i formulama, prikazana su i poređenja performansi na različitim skupovima podataka. Razlozi viših performansi ansambla u odnosnu na monolitne prediktore mogu se intuitivno objasniti *bernulijevim* procesom koji nije u okvirima naše time pa ga nećemo detaljnije obraditi.

# Bagging

U ovom radu, prvi algoritam koji ćemo obrađivati je Bagging. On je jedan od prvenaca i koristi osnovne tehnike koje će se i u novijim ansamblima koristiti.

Bagging metoda je jedna od ansambl tehnika objavljena od strane Leo Brajmana [Breiman (1996)]. Ova tehnika kreira više verzija jednog prediktora odnosno klasifikatora, čiji rezultati se na kraju sakupljaju i kreira se glasanje za konačnu odluku ansambla. Svaki prediktor dobija svoj set podataka koji je generisan tehnikom uzorkovanja sa zamenom, odnosno statističkom metodom *bootstraping* [Horowitz (2001)]. Algoritam je dobio ime kombinacijom reči ***b****oostraping i* ***agg****regat****ing*** (agregacija konačnih glasova pojedinačnih prediktora ansambla). Ova metoda spada u grupu koja ima mogućnost paralelizacije, jer učenje prediktora nije zavisno jedno od drugog. Korišćenjem uzorkovanja sa zamenom, u proseku svaki bootstrap koristi oko 63% slučajeva iz originalnog seta podataka, odnosno oko 37% slučajeva ostaje ne iskorišćeno. Ideja ove metode je da se koristi slab osnovni prediktor, i da u ukupnom klasanju dostigne veću preciznost.

Koraci algoritma:

1. Kreirati uniformnu distribuciju nad slučajevima u setu:
2. Pravljenje N boostrap uzoraka nad skupom podataka iz koraka 1 (trening skup) veličine K.
3. Za svaki od N uzoraka kreirati jedan model i trenirati ga koristeći osnovni algoritam: . Svaki prediktor daje svoje predviđanje atributa .
4. Računa se očekivana vrednost. Za kategoričke izlaze to će biti kategorija sa najviše glasova, dok će za numeričke biti srednja vrednost rezultata prediktora.

# Random Forest

Ansambl *Random Forest* je formulisao Leo *Brajman*, 2001. godine. Profesor *Brajman* u svojoj publikaciji navodi još autora koji su obrađivali sličnu tematiku i čiji su radovi uticali na razvijanje ideje o *Random Forest* ansamblu (*Dieterich[1998], Ho [1998], Amit and Geman [1997]*).

Ovaj ansambl je kombinacija prediktora stabla odlučivanja, koji međusobno zavise samo od nasumično kreiranih vektora atributa sa istom distribucijom. Ideja glasanja radi dobijanja konačnog izlaza je preuzeta od ranije pomenutog ansambla *Bagging*. Sam algoritam je lako paralelizovati zbog nezavisne izgradnje stabala.

Podaci su definisani:

Parametri algoritma su:

1. Osnovni algoritam -
2. Broj osnovnih modela u ansamblu – N
3. Broj atributa koji se selektuju – F

Koraci algoritma su:

1. Kreiranje raspodele verovatnoće nad atributima skupa podataka. Uglavnom se koristi uniformna raspodela.
2. Pravljenje vektora atributa veličine F za N modela u ansamblu. Atributi se biraju nasumično.
3. Za svaki model u ansamblu i njegov odgovarajući set podataka kome je dodeljen odgovarajući vektor atributa kreira se stablo odlučivanja osnovnim algoritmom. Svaki model je u stanju da napravi predviđanje nezavisno od ostalih modela.
4. Za predviđanje celog ansambla koristi se ista tehnika kao u četvrtom koraku prethodno opisanog *Bagging ansambla.*

# Boosting – Adaboost

Adaboost ansambl algoritam (*Freund and Schapire [1995]*) je formulisan, kako bi rešio praktične probleme prethodnih boosting algoritama. Početna ideja boosting algoritama je slična kao i kod bagging ansambla. Od slabih prediktora potrebno je napraviti konačni, bolji, koji će biti izglasan u ansamblu. Još jedna ideja boosting-a je da se iterativno treniraju modeli i da se penalizuju instance u setu nad kojima se napravi greška u predviđanju modela t. Svaki naredni model uči sa većim fokusom na greškama prethodnih modela. Ovo se postiže tako što osnovni algoritam mora da ima implementiranu funkcionalnost otežavanja zapisa u setu podataka. Kao i u stvarnom životu, postoje različiti nivoi eksperata koji mogu sa različitim procentima uspešnosti rešiti zadatke. Ovaj algoritam ima sistem procene eksperata (modela), i njih takođe penalizuje prema uspešnosti predviđanja. Što znači da će bolji eksperti imati jači glas u konačnoj odluci prilikom glasanja u ansamblu. Ovaj algoritam se prilikom treniranja ansambla izvršava sekvenciono, jer kalkulacije svakog modela (težine) zavise od rezultata predviđanja prethodnog modela, dok je predviđanje dosta brže od treniranja i može da se paralelizuje. Upravo zbog svih prilagođavanja težina instanci i modela ovaj algoritam je dobio naziv ***Ada****ptive* ***Boost****ing*.

Set podataka je definisan:

Parametri algoritma su:

1. Ukupan broj iteracija - T
2. Bazni algoritam -

Koraci algoritma:

1. Inicijalizuju se težine svih instanci iz seta L prema formuli:
2. Radi se predikcija za svaki klasifikator
3. Na osnovu greške modela

se računaju se težine prediktora (alfe):

1. U sledećem koraku se menjaju težine instanci:

1. U poslednjem koraku treniranja ansambla se računa konačna odluka sledećom jednačinom:

# Stacking

Ansambl Stacking je formulisao David Wolpert (*David H. Wolpert [1992]*). U literaturi algoritam se može naći i kao „Stacked generalization“ jer ga je Wolpert tako formalno nazvao. Ansambl je heterogen, odnosno sadrži više različitih osnovnih algoritama , čiji izlazi služe kao ulazni set podataka za meta algoritam (p - *eng. Parent*) algoritam koji daje konačan izlaz i on se u literaturi naziva generalizator (*eng. generalizer*). Cilj ovog ansambla je da postigne što je veću moguću generalizaciju konačnog modela. Za odabir najboljeg generalizatora, možemo napraviti testiranje i uporediti rezultate.

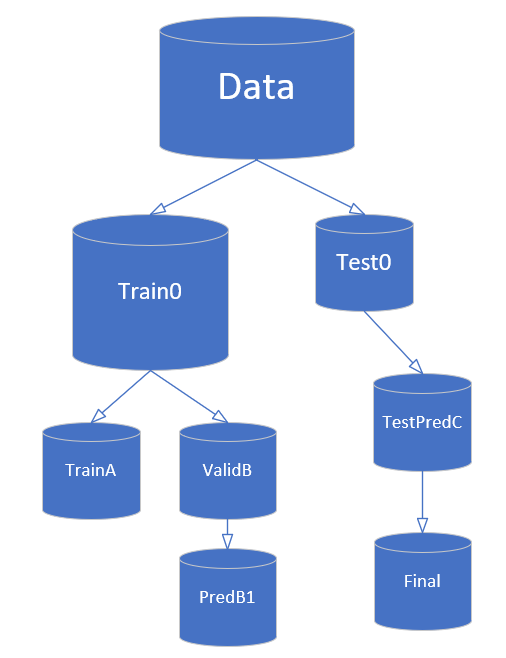
Podaci su definisani:

Parametri algoritma:

1. Osnovni algoritmi -
2. Algoritam generalizator -

Koraci algoritma:

1. Podaci se podele na 2 seta, i (test)
2. set se dalje deli na i .
3. Na setu treniraju se osnovni algoritmi .
4. Koristeći dobijene osnovne modele iz prethodnog koraka, napraviti predviđanja:
   1. Na validacionom setu i rezultate sačuvati u setu
   2. Na test setu i rezultate sačuvati u setu
5. Trenitati generalizator algoritam na setu
6. Predvideti izlaze modelom istreniranim u prethodnom koraku na setu



Slika – Stacking podela podataka prilikom izvršavanja algoritma

# Gradient boosting

# XGBoost

# Analiza

# Dalji rad