

## Lab 7: Data Mining.

### Metode de tip ansamblu

#### Analiza documentelor text (Text mining)

---

### 1. Metode de tip ansamblu

Sunt meta-modele care se obțin din câteva modele de bază antrenate pe același set sau pe seturi diferite de antrenare. Există mai multe variante de a construi modele de tip ansamblu:

- Utilizând modele bazate pe algoritmi diferiți antrenati pe același set de date (e.g. *bucket of models*)
- Utilizând modele bazate pe același tip de algoritm dar antrenate pe seturi diferite de date (e.g. *bagging* and *boosting*)
- Utilizând diferite modele și împărțind setul de date (e.g. *stacking*)

Motivația principală este de a reduce eroarea la predicție acționând asupra celor două componente: deplasarea (bias) și varianța (variance).

**Exercițiul 1/ Rattle.** Aplicați clasificatorul de tip Random Trees din Rattle pentru a clasifica datele din seturile folosite la Lab 3: (a) iris; (b) breast cancer (Wisconsin); (c) Titanic.

#### Exercițiul 2/R.

- a) Testați pachetul SuperLearner. Exemplu: [ExampleSuperLearner.r](#)
- b) Testați pachetul caretEnsemble. Exemple:  
<https://cran.r-project.org/web/packages/caretEnsemble/vignettes/caretEnsemble-intro.html>  
<https://rpubs.com/zxs107020/370699>

**Exercițiul 3/Weka.** Utilizand Weka Experimenter comparați performanța următoarelor metamodele: *Vote*, *Bagging*, *Random Forest*, *AdaBoost* și *Stacking* pt seturile de date: *iris.arff*, *glass.arff*

- a) Utilizați valorile implicite ale parametrilor
- b) Îmbunătățiți comportamentul pt *Vote*, *Bagging* și *AdaBoost* înlocuind clasificatorul de bază cu alt clasificator.

### 2. Text mining

Analiza textului (text mining) are ca scop extragerea de informații din documente (documentele sunt interpretate ca secvențe de cuvinte). Principalele tipuri de prelucrări sunt: clasificarea și gruparea documentelor pe baza conținutului lor. Cea mai simplă abordare se bazează pe aplicarea următoarelor etape:

- Pre-procesarea textului prin:
  - Eliminarea cuvintelor de legatură (*stop words*). Liste cu cuvinte de legatură pt diferite limbi pot fi găsite la <http://www.ranks.nl/stopwords>
  - Transformarea cuvintelor prin *stemming* (i.e. reducerea la rădăcina cuvintului). Cel mai popular algoritm de stemming este cel propus de Porter (vezi <http://tartarus.org/martin/PorterStemmer/>). Un serviciu web pt stemming este disponibil la <http://text-processing.com/demo/stem/>

- Construirea pt fiecare document a vectorului de frecvente( *frequency vector*): nr de aparitii ale fiecarui cuvânt din dictionar in cadrul documentului.

#### Exercitiul 4/Weka.

- a) Deschideți fișierul [movieReviews.arff](#) (conține recenzii ale unor filme clasificate în două categorii: pozitive și negative)
- b) Construiți setul de date cu ocurențele termenilor în colecția de recenzii folosind [Filters->Unsupervised->Attribute->StringToWordVector](#)
- c) Aplicați un clasificator (e.g. [Naïve Bayes](#)) asupra setului de date. Obs: e necesar ca primul atribut ( [@@class@@](#) ) să fie definit ca atribut de clasă (folosind [Edit](#), click dreapta pe [@@class@@](#) și selectând [Attribute as class](#))
- d) Analizați impactul utilizării etapei de stemming asupra calității clasificării.