ML Raport - Spam Email Filter

Istrate Valentin 3A1, Pavel Silviu-Mihail 3A1

12.01.2024

1 Alegerea algoritmului

Având în vedere că, setul de date conține clasa fiecărui e-mail - dacă este sau nu de tip spam - atunci, cea mai indicată metodă este să folosim un algoritm de învățare automată supervizată. Astfel, putem alege dintre următoarele:

- **arbori de decizie**: algoritmul ID3, algoritmul AdaBoost;
- **clasificare bayesiană**: algoritmul Bayes Naiv, algoritmul Bayes Optimal;
- **învățare bazată pe instanțe**: algoritmul k-NN (k-nearest neighbors);

În continuare, vom determina care algoritm este mai fezabil pentru implementarea modelului nostru (algoritmul Bayes Optimal este exclus din această comparație deoarece rezultatele lui sunt mai mult teoretice și destul de greu de aplicat în practică).

- **ID3:**
  - **Avantaje**: simplu de înțeles și interpretat, lucrează bine pe date categoriale;
  - **Dezavantaje**: predispus la overfitting și poate deveni bias către atributele cu mai multe nivele;
  - **Concluzie**: nu este cea mai buna varianta din cauza simplității și a problemelor de overfit;
- **AdaBoost:**
  - **Avantaje**: crește acuratețea clasificatorilor slabi și nu este la fel de predispus ca alți algoritmi la overfitting;
  - **Dezavantaje**: sensibil la zgomote si outlieri;
  - **Concluzie**: o variantă bună întrucât poate îmbunătăți performanța algoritmilor mai slabi și se poate adapta la natura evolutivă a spamului;
- **Bayes Naiv:**
  - **Avantaje**: se descurcă bine în clasificare pe mai multe clase, lucrează bine cu date textuale, eficient și ușor de implementat;
  - **Dezavantaje**: presupunerea de independență condiționată nu este neapărat respectată;

  - **Concluzie**: folosit de mult ori pentru clasificarea e-mailurilor spam din cauza eficienței în clasificare textuală;

- **k-NN:**
  - **Avantaje**: simplu și eficient în clasificare;
  - **Dezavantaje**: încet pe dataseturi mari, sensibil la atribute irelevante sau redundante și are nevoie de scalare;
  - **Concluzie**: nu este prima varinantă pentru clasificare textuală din cauza simplității și a naturii multi-dimensională a datelor textuale;

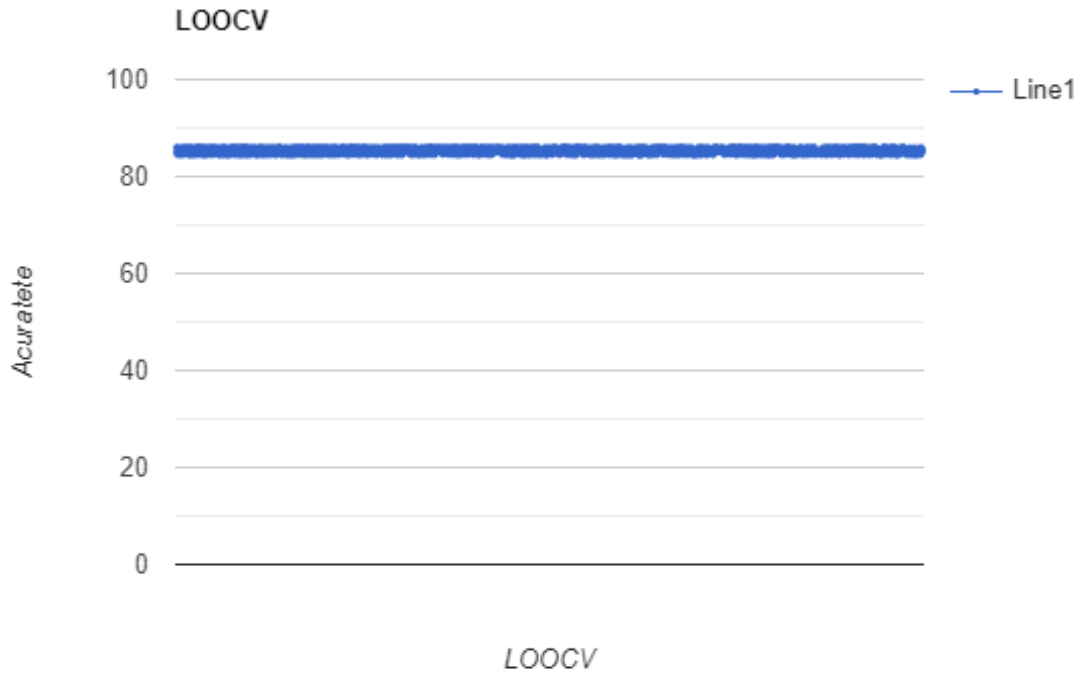
Astfel, avem de ales între Bayes Naiv și AdaBoost. Totuși, pentru a implementa AdaBoost avem nevoie de un alt clasificator mai slab pe care îl putem imbunatății. Deci, în următoarea implementare vom utiliza algoritmul Bayes Naiv deoarece este varianta cea mai folosită pentru a implementa un sistem de detecție spam și se cunoaște faptul că algoritmul este bun la clasificare textuală.

Algoritmul funcționează în felul următor:

1. se citesc datele din fișier cuvânt cu cuvânt;
2. pentru fiecare cuvânt se reține numărul de apariții într-un e-mail de tip spam și non-spam;
3. se calculează probabilitatea unui cuvând să fie spam și se salvează într-un fișier json;
4. pentru validare se folosesc probabilitățile calculate anterior și formula din algoritm (presupunând că independența condiționată este adevărată);
5. în cazul în care un cuvânt nu se găsește în json, atunci acesta va avea probabilitatea de 0.5;

2 Rezultatele stategiei de cross-validare Leave-One_Out

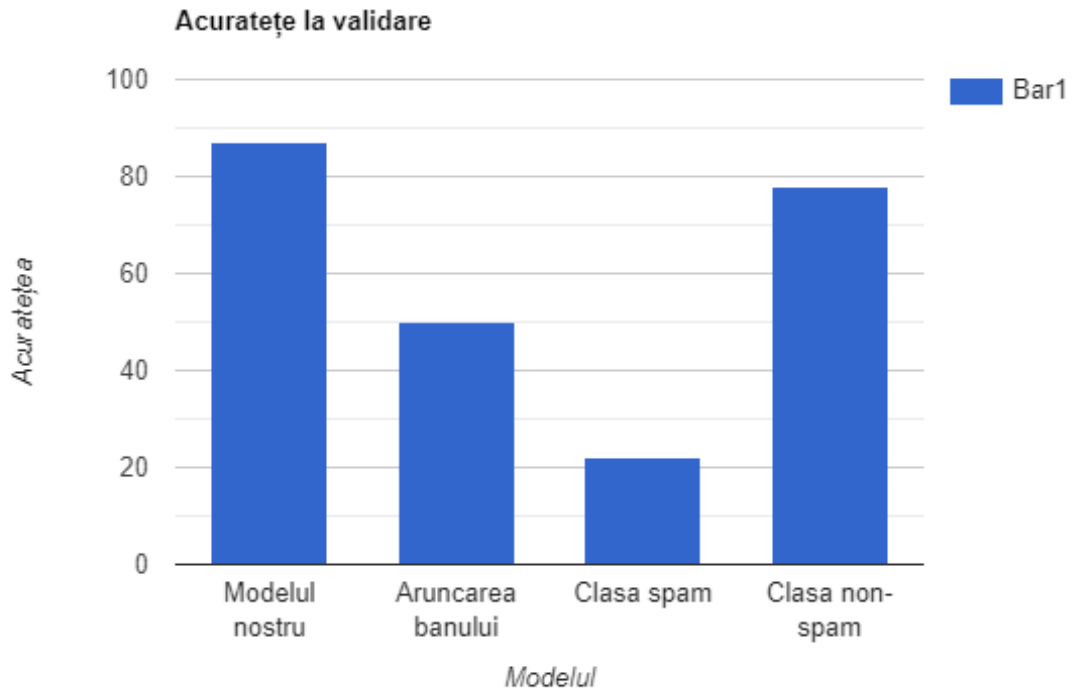
Mai jos se poate găsi un grafic ce reprezintă performanța LOOCV pentru setul de validare (part-10).



3 Performanta algoritmului

Modelul folosit pentru testarea acurateții la validare este cel obținut folosind toate datele din fișierele part1 - part9. În continuare, avem următoarele date:

- **acuratețea la validare**: 87% (0.87);
- **aruncarea unui ban**: 50% (0.5);
- **acuratețea dacă clasificam toate datele ca fiind spam**: 22% (0.22);
- **acuratețea dacă clasificam toate datele ca fiind non-spam**: 78% (0.78);



În concluzie, modelul creat este mai performant decât clasificarea cu o singură clasă sau cu o clasă random.