Temă practică - Învățare Automată

Studenți: Chirilă Gabriela-Valentina & Mistreanu Emanuela

12 Ianuarie 2024

1 Preprocesarea datelor

Setul de date pe care lucram se gaseste sub forma unui fisier text, de forma:

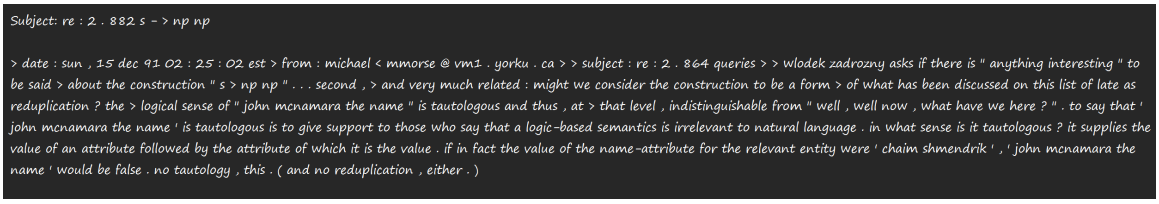

Figure 1: Structura unui fisier text din folder-ul cu date

Pentru a efectua o clasificare eficientă a email-urilor în categoriile de spam sau non-spam, am implementat o serie de pași de prelucrare a textului:

În primul rând, am eliminat toate caracterele speciale mai specific cele care nu sunt litere sau spații, incluzând semne de punctuație, cifre sau caractere non-literale.

Ulterior, după acest pas am descompus conținutul fiecărui email în cuvinte individuale.

Pentru a rafina mai mult setul de date primit ca input am ales să facem o selecție a cuvinte relevante, așa numitele "cuvinte cheie" pe care să putem ulterior aplica diferiți clasificatori. Mai exact am ales să eliminatinăm cuvintele supranumite "stop-words", adică prepoziții (de exemplu, "The", "in", "end"), pronume, substantive comune, articole etc.

O ultimă etapa de rafinare a setul de date o reprezină reducerea cuvintelor rezultate de până acum la forma lor de bază, exemplificată prin transformarea verbelor cum ar fi "are", "am" sau "is" în forma lor la infinitiv," be".

Într-un proces detaliat, am iterat prin fiecare email, de la "part1" până la "part9", pentru setul de antrenare și, respectiv "part10" pentru setul de testare, am curățat datele conform procedurilor menționate, reținând cuvintele cheie după curățare într-un fișier. Am organizat aceste cuvinte într-o coloană, iar într-o a doua coloană am adăugat etichetele corespunzătoare (1 pentru spam și 0 pentru non-spam). Etichetele au fost atribuite în funcție de numele fișierului asociat, conform artficiului precizat în cerință.

Această abordare meticuloasă a asigurat o pregătire adecvată a datelor pentru clasificare, evidențiind cuvintele semnificative și eliminând elementele redundante, facilitând astfel procesul de identificare a caracteristicilor distinctive ale email-urilor spam și non-spam.

text	label
Subject np np date sun dec est michael mmorse vm yorku ca subject query wlodek zadrozny asks anything interesting said construction np np second much related might consider construction form discussed list late reduplication logical sense john mcnamara name tautologous f	0
Subject np np discussion np np reminds year ago read source forgotten critique newsmagazines unique tendency writing style writer found overly cute one item tersely put follows time favorite colon lee hartman ga siucvmb bitnet department foreign language southern illinois u	0
Subject np np much restrictive np np np pro quite overrestriction	0
Subject gent conference listserv international conference second circular february literature analysis discourse special attention multicultural context tuesday september friday september gent university belgium writing reading literature oral literary tradition dialogic text nonlit	0
Subject query causatives korean could anyone point book article causative construction korean please send email directly thanks hiromi morikawa hiromi psych stanford edu	0
Subject l learning cultural empathy graduate student education approached colleague mine query linguist people may able help evaluation exchange program indonesia one object prepare high school teacher indonesia australia wondered anything written correlation degree a	0
Subject psycholinguistics teaching undergraduate course shortly teaching psycholinguistics would appreciate suggestion text instructor good experience also would indebted anyone offer specific reference work helen neville deaf alinguals acquisiton asl thanks klaiman klaima	0
Subject german corpus looking online corpus modern german information would appreciated ken beesley beesley parc xerox com	0

Figure 2: Structura datelor din fișierul .csv după preprocesare

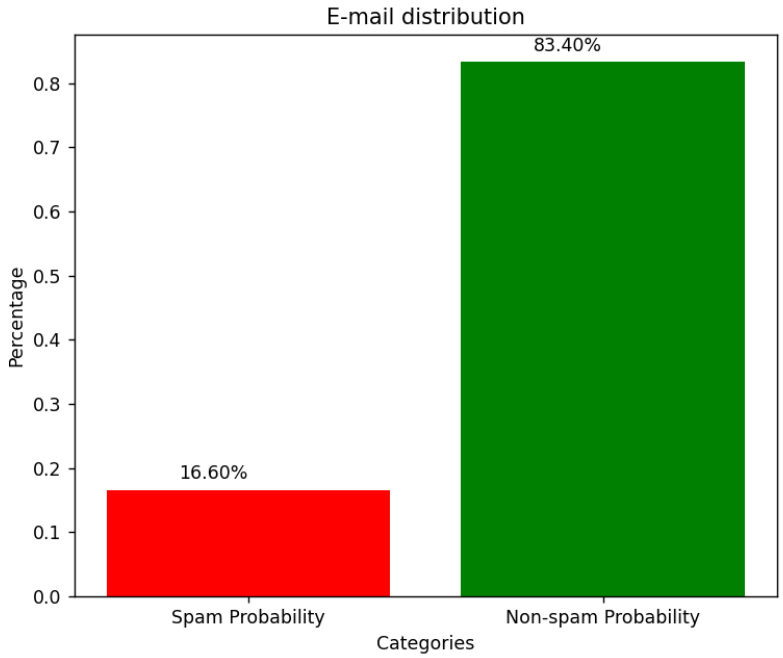

Figure 3: Distribuția email-urilor spam și non-spam

În graficul de mai sus, este ilustrată distribuția email-urilor spam și non-spam a datelor de antrenare.

2 Clasificatorul Bayes Naiv

Am optat pentru utilizarea acestui clasificator, considerându-l potrivit pentru soluționarea problemei date, deoarece:

- **Este robust la overfitting.**
- **Demonstrează o putere de generalizare considerabilă.**
- **Ia decizii rapid, adică este optimal din punct de vedere al timpului de clasificare.**
- **Furnizează un nivel înalt de interpretabilitate, folosind concepte simple bazate pe probabilități, fără a recurge la un model specific sau algoritmi complexi care ar necesita cunoștințe avansate pentru interpretare..**
- **Este capabil să gestioneze eficient atributele lipsă, bazându-se pe probabilități marginale.**
- **Manifestă o rezistență mai bună la zgomot (date puternic mixate), concentrându-se pe probabilități și presupunând independența între atribute.**
- **Este frecvent utilizat în clasificarea textelor, filtrarea spamului, analiza sentimentelor și în alte sarcini care implică calcule de probabilitate.**

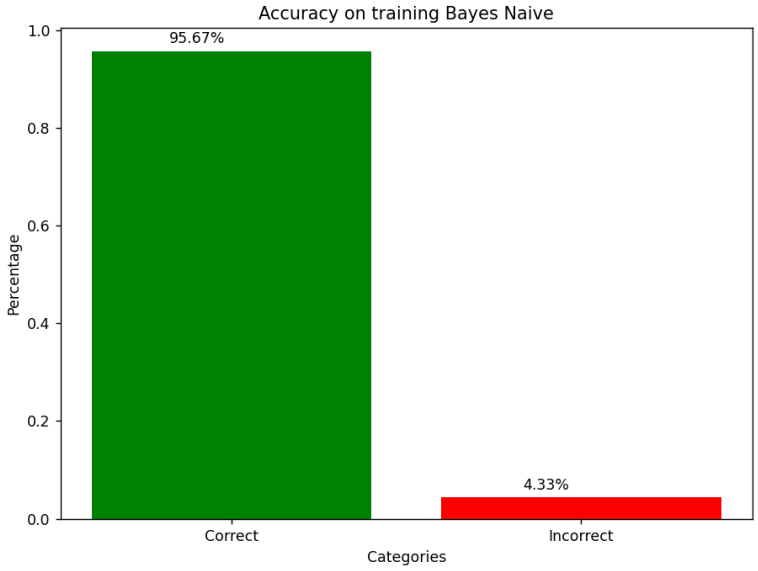

Figure 4: Acuratețea la antrenare pentru clasificatorul Bayes Naiv

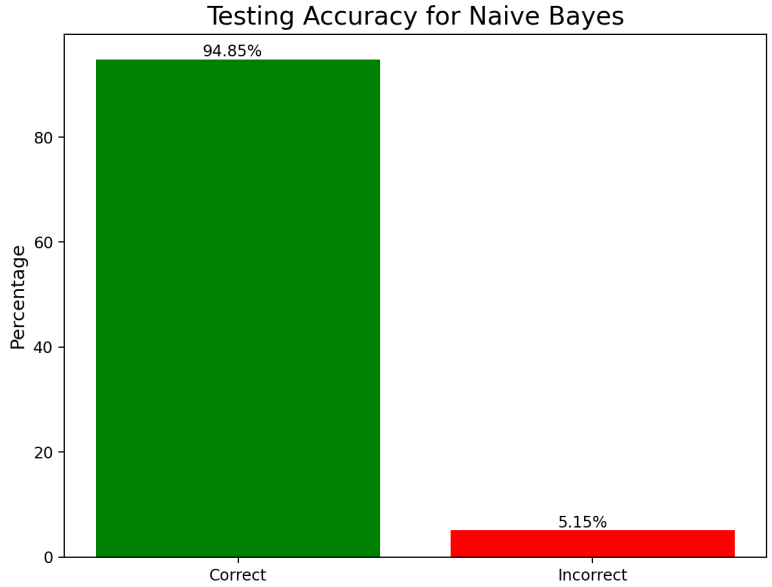

Figure 5: Acuratețea la testare pentru clasificatorul Bayes Naiv

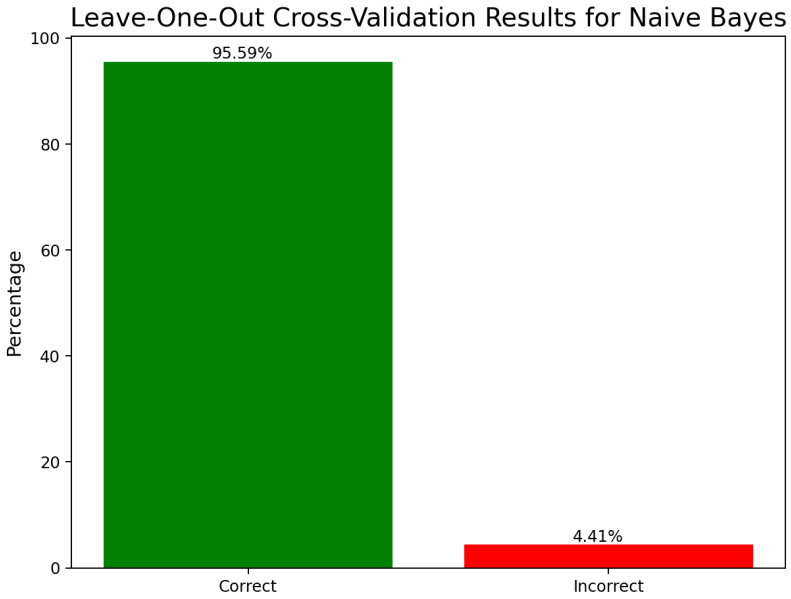

Figure 6: Cross-Validare Leave-One-Out pentru clasificatorul Bayes Naiv

În consecință, acest clasificator se evidențiază ca o alegere potrivită pentru sarcina noastră.

3 Bayes Naiv vs. ID3

Având în vedere problema clasificării textelor, este esențial ca algoritmul ales să se comporte bine în lipsa unor cuvinte din setul de date de antrenament. ID3 este sensibil la lipsa atributelor (cuvintelor) deoarece reprezintă o structură arborescentă și necesită ca secvența de cuvinte pe care încercăm să o clasificăm să se afle în nodurile arborelui. Pe de altă parte, Bayes Naiv se comportă bine în astfel de situații, putând lua în considerare regulile lui Laplace.

Este esențial să subliniem faptul că algoritmul ID3 nu reacționează la duplicarea atributelor; cu alte cuvinte, în setul nostru de date, acesta nu ia în considerare apariția repetată a aceluiași cuvânt în același e-mail. În contrast, Bayes Naiv este sensibil la această duplicare, adică ține cont de frecvența cu care apare un cuvânt, indiferent dacă se găsește în același e-mail sau nu. Atunci când încercăm să clasificăm e-mailurile pe baza apariției cuvintelor în diferite tipuri de e-mailuri, este esențial să luăm în considerare nu doar prezența sau absența cuvintelor, ci și frecvența acestora. Acest lucru este crucial pentru obținerea unei clasificări eficiente și precise.

Datorită complexității ridicate a setului de date și lucrului cu volume mari de date, ID3 poate fi sensibil la zgomot, deoarece încearcă să construiască un model perfect, în timp ce Bayes Naiv este mai robust, concentrându-se pe probabilități și vot majoritar.

ID3 poate produce ușor overfitting, dorind să construiască un model în concordanță perfectă cu datele, spre deosebire de Bayes Naiv, care nu este atât de predispus la overfitting.

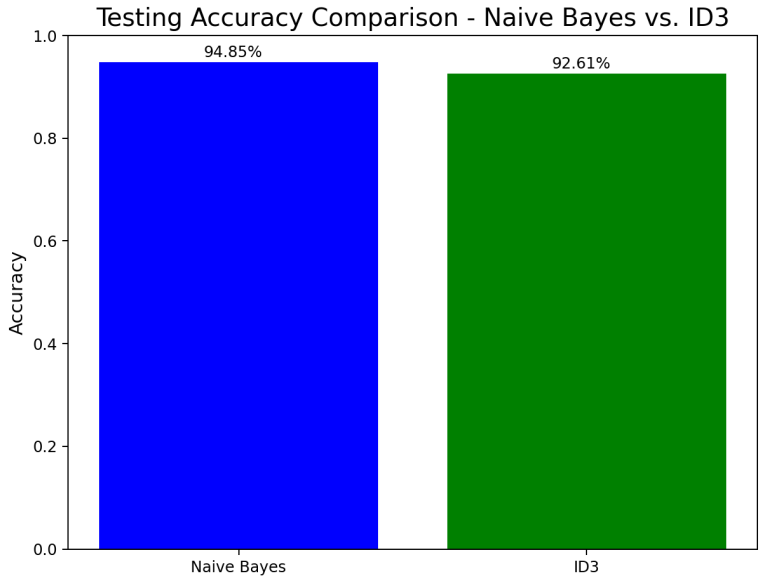

Figure 7: Comparație între Bayes Naiv Și ID3

4 Bayes Naiv vs. K-NN

K-NN este sensibil la outlieri și zgomote în date, deoarece acestea pot avea un impact semnificativ asupra calculului distanțelor, în timp ce Bayes Naiv este mai robust, concentrându-se pe votul majoritar.

K-NN se comportă bine pe seturi mici de date, în timp ce Bayes Naiv se adaptează bine atât la seturi mici, cât și la seturi mari de date.

În cadrul clasificatorului K-NN, performanța este influențată de două aspecte cheie: alegerea numărului de vecini (k) și tipul de măsură utilizată. În contrast, Bayes Naiv nu impune astfel de decizii critice. Optimalizarea valorii lui k depinde de caracteristicile setului de date, ceea ce aduce cu sine provocări de generalizare pe măsură ce se lucrează cu seturi de date variate. Aceste dificultăți de optimizare nu sunt întâlnite în mod similar în cadrul clasificatorului propus.
Din analiza graficelor prezentate mai jos, se evidențiază o diferență între clasificatorii Bayes Naiv și K-NN, mai ales în funcție de diferitele valori atribuite parametrului k. Este evident că acuratețea variază pentru K-nn variază, sugerând că performanța acestui algoritm este strâns legată de alegerea valorii optime a lui k, așa cum am menționat anterior.

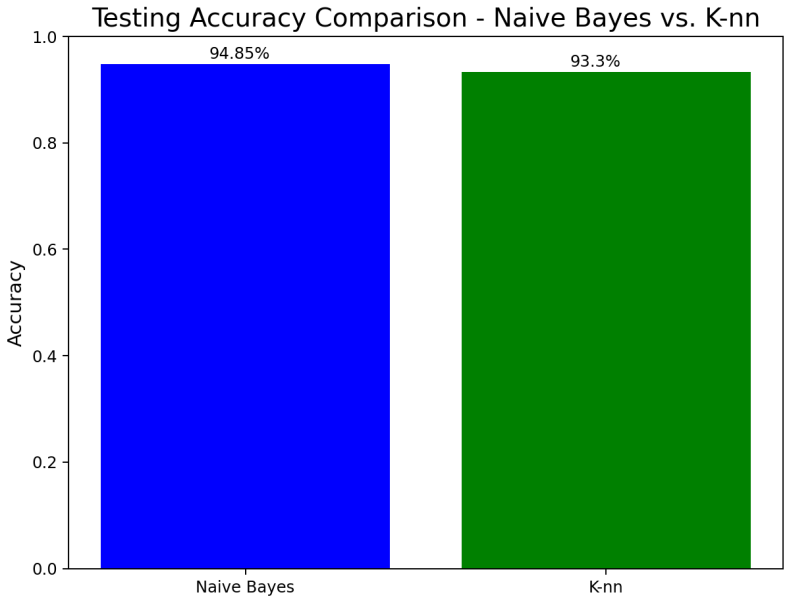

Figure 8: Comparație între Bayes Naiv Și 5-NN

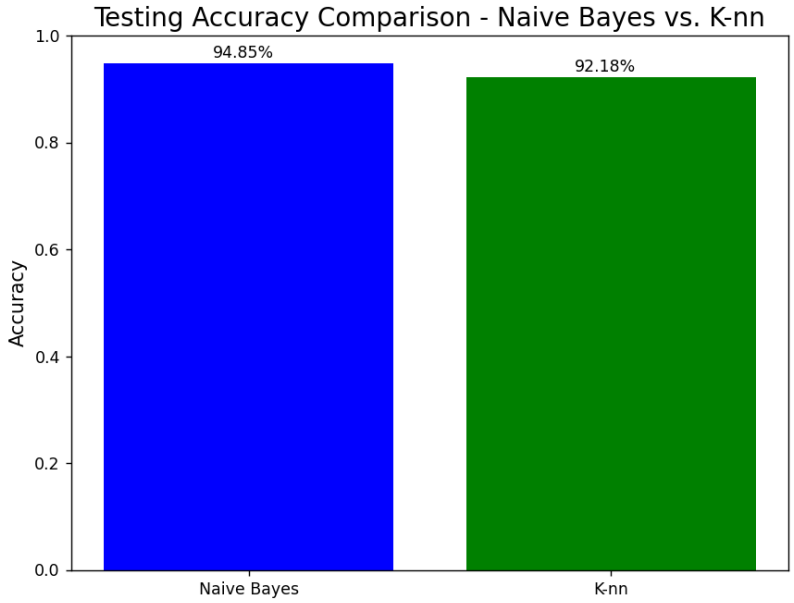

Figure 9: Comparație între Bayes Naiv Și 10-NN

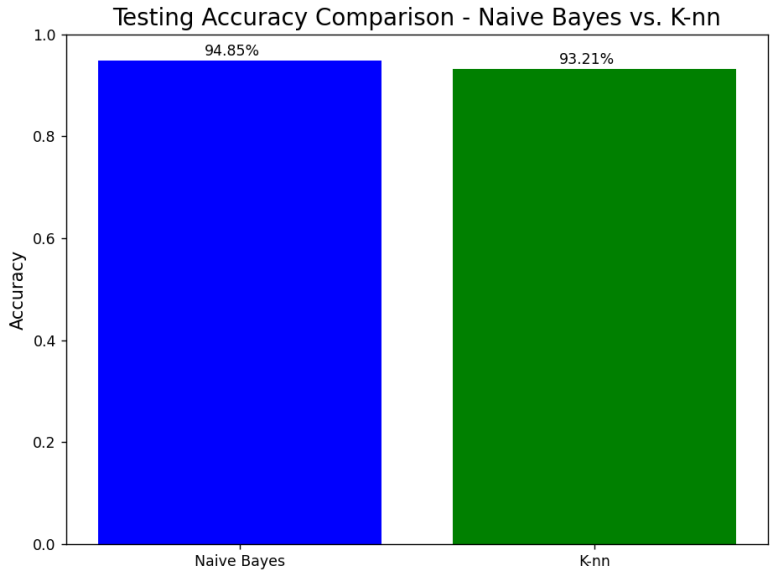

Figure 10: Comparație între Bayes Naiv Și 20-NN

5 Bayes Naiv vs. AdaBoost

AdaBoost este, de asemenea, sensibil la zgomote și prezintă o complexitate din punct de vedere a timpului mai mare, deoarece calculează noi distribuții la fiecare pas.

Adaboost și Bayes Naiv sunt amândoi clasificatori robusti împotriva overfittingului, fundamentându-se pe crearea unor clasificatori simpli bazate pe probabilități. Cu toate acestea, există o distincție semnificativă în abordarea lor. Bayes Naiv efectuează o singură iterație pentru calcularea probabilităților, în timp ce Adaboost parcurge mai multe iterații pentru a ajunge la o combinație optimă a clasificatorilor.
Faptul că Bayes Naiv se rezumă la o singură iterație sugerează o abordare mai simplă și mai directă în estimarea probabilităților. În schimb, Adaboost utilizează iterații multiple pentru a îmbunătăți performanța generală prin ponderarea succesivă a greșelilor clasificatorilor slabi.
Este important să menționăm că numărul optim de iterații în Adaboost este un aspect crucial. Un număr prea mic de iterații poate conduce la subestimarea capacității clasificatorului, rezultând într-un model suboptimal. Pe de altă parte, un număr prea mare de iterații poate crește complexitatea timpului de calcul și poate duce la suprainvatare sau chiar la overfitting în anumite cazuri.
Astfel, alegerea numărului adecvat de iterații în Adaboost este o decizie sensibilă, care trebuie să echilibreze îmbunătățirea performanței cu costurile asociate complexității.

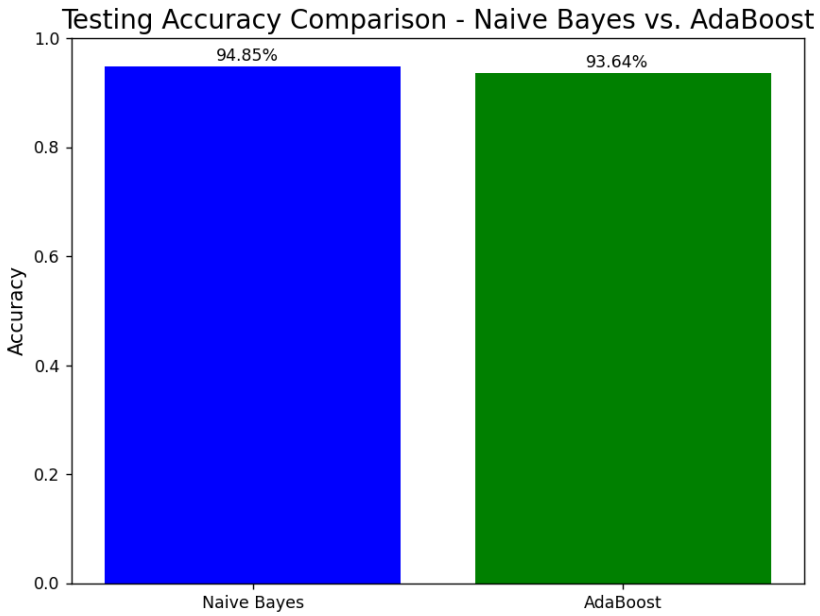

Figure 11: Comparație între Bayes Naiv Și AdaBoost (maxim 5 iteratii)

```
------------------------------
AdaBoost
------------------------------
Accuracy on Training Set: 0.9531129900076863
Accuracy on Test Set: 0.936426116838488
Timpul total de execuție: 5.754607439041138 secunde
```

Figure 12: Acuratețea precum timpul necesar rularii clasificatorului AdaBoost

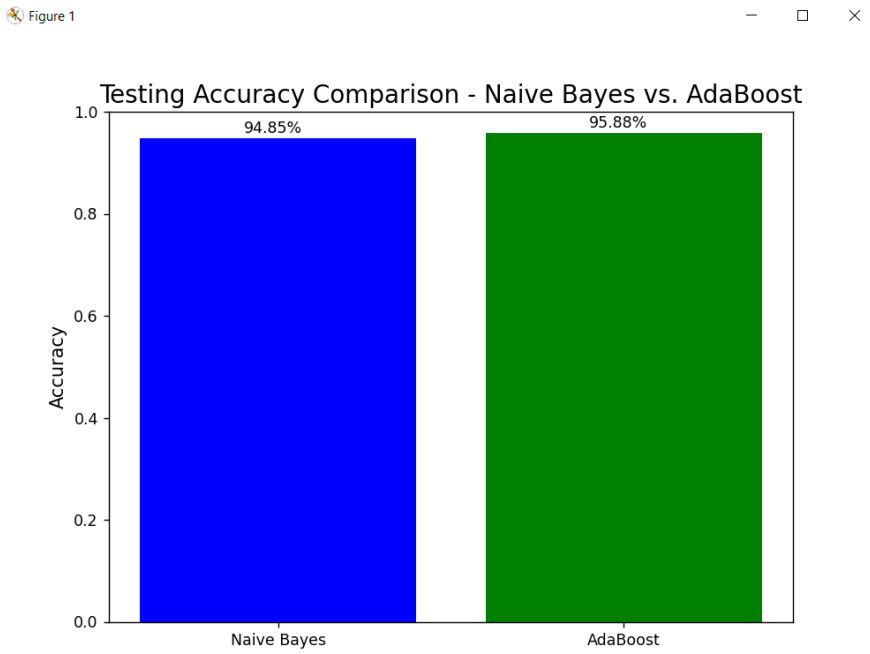

Figure 13: Comparație între Bayes Naiv Și AdaBoost (maxim 10 iteratii)

```
--------------------------------
AdaBoost
--------------------------------
Accuracy on Training Set: 0.9644504227517294
Accuracy on Test Set: 0.9587628865979382
Timpul total de execuție: 6.999921560287476 secunde
```

Figure 14: Acuratețea precum timpul necesar rularii clasificatorului AdaBoost

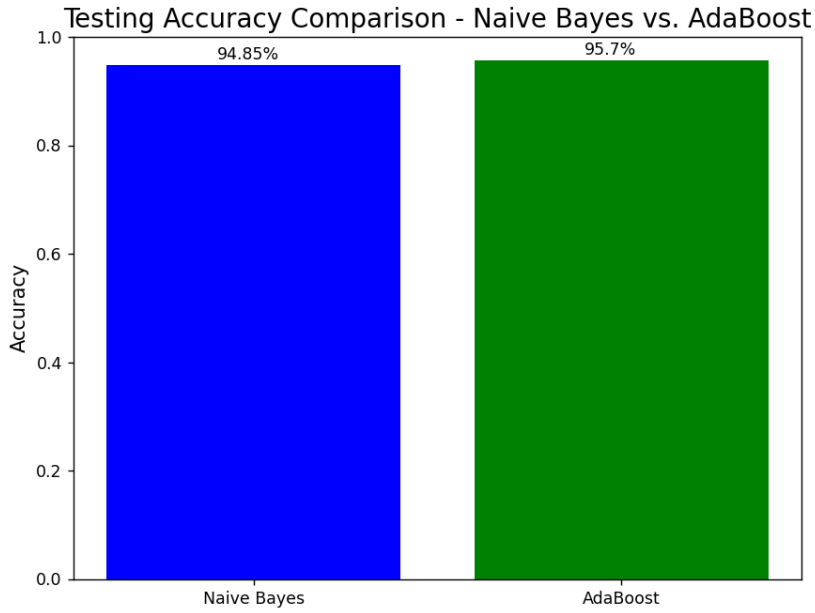

Figure 15: Comparație între Bayes Naiv Și AdaBoost (maxim 20 iteratii)

```
------------------------------
AdaBoost
------------------------------
Accuracy on Training Set: 0.9850115295926211
Accuracy on Test Set: 0.9570446735395189
Timpul total de execuție: 10.256410837173462 secunde
```

Figure 16: Acuratețea precum timpul necesar rularii clasificatorului AdaBoost

6 Concluzii

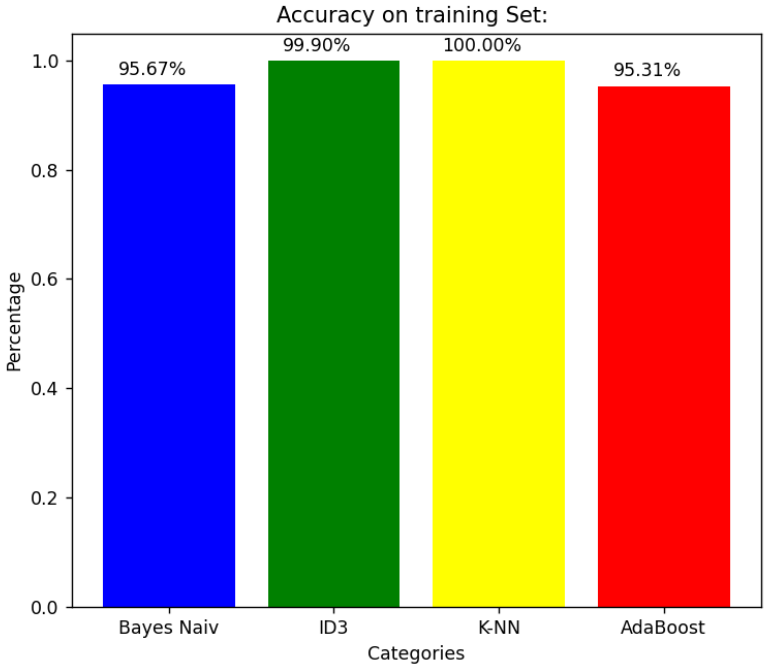

Figure 17: Acuratețea pe setul de antrenare

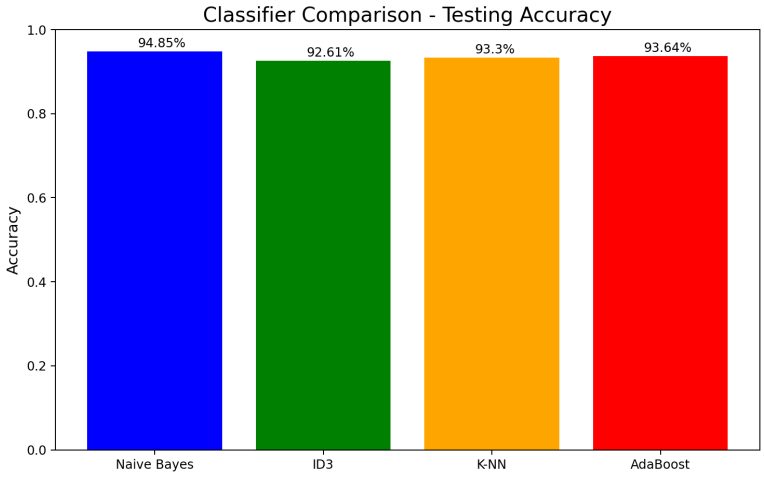

Figure 18: Acuratețea pe setul de testare

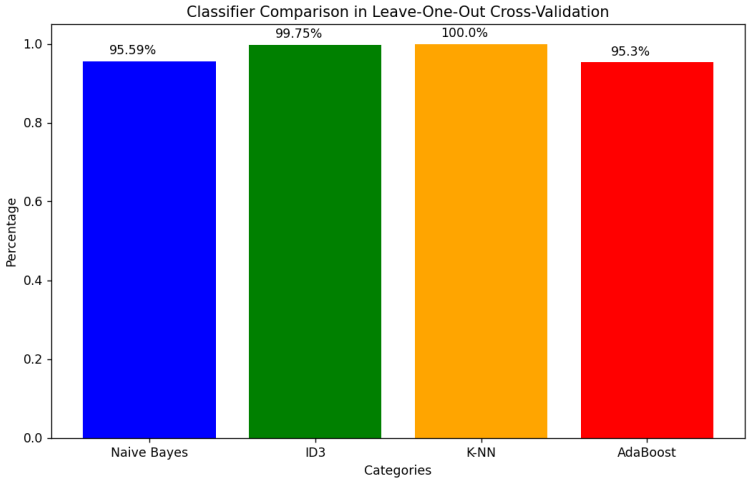

Figure 19: Cross-Validation Leave-One-Out

Analizând graficele, se observă că Bayes Naiv evidențiază cea mai mică discrepanță între erorile de antrenare și testare. Această observație indică faptul că algoritmul Bayes Naiv adoptă o abordare mai echilibrată în construirea modelului său, evitând supradaptarea la datele de antrenare și manifestându-se eficient în evaluarea pe setul de testare în comparație cu alți algoritmi.

În contrast, pentru ceilalți algoritmi testați, discrepanța este semnificativ mai mare, sugerând posibilitatea apariției fenomenului de overfitting. De asemenea, se observă că mulți dintre acești algoritmi depind de parametri optimi pentru a obține rezultate bune, iar acești parametri sunt strâns corelați cu caracteristicile specifice ale setului de date.
În ceea ce privește cele două caracteristici esențiale, eficiența (complexitatea în timp și spațiu a algoritmului) și puterea de generalizare, observăm că Bayes Naiv se dovedește a fi cel mai potrivit pentru clasificarea setului nostru de date în contextul problemei examinate.

Algorithm	**Training Accuracy**	**Test Accuracy**	**CVLOO Score**
Naive Bayes	95.67%	94.85%	95.59%
ID3	99.62%	92.44%	99.75%
K-NN	100%	93.30%	100%
AdaBoost	95.31%	93.64%	95.30%

Table 1: Comparison of Algorithm Performances