

## Task: Sarcina 02 - Validarea modelului pentru detectarea mesajelor spam

Curs: Introduction to Machine Learning using Python

Modul: Cât de bune sunt modelele noastre: Validare și evaluare

### Contextul sarcinii

După ce echipa ta a implementat primul model pentru detectarea automată a mesajelor spam, atmosfera din birou s-a schimbat: o parte dintre colegi sunt încântați („În sfârșit mai puține gunoaie în inbox!”), în timp ce alții observă mici neajunsuri („De ce mi-a ajuns un e-mail important în spam?”).

Acum apare o nouă întrebare – cât de mult putem, de fapt, să avem încredere în acest model? Și, mai important: am putea obține rezultate mai bune dacă am încerca și alte tehnici analitice sau modele?

Această etapă este un clasic „data challenge de echipă” – ține de tine să reevaluatezi performanța modelului actual, să-i descoperi punctele forte și punctele slabe și să deschizi drumul către o variantă mai precisă. O simulare reală de afaceri: nu mai este suficient să „funcționeze”, acum contează cât de bine funcționează și ce ar putea fi îmbunătățit.

Pe canalul Slack, comentariile curg: cineva propune să comparați câteva modele în plus, altcineva sugerează teste suplimentare, iar tu ai sarcina de a aduce datele care să ofere un răspuns clar întregii echipe.

### Scopul sarcinii

Scopul acestei sarcini este să îți formezi o imagine clară despre cât de bine funcționează modelul tău de detectare a mesajelor spam și să îneveți cum se măsoară, în practică, performanța diferitelor modele de clasificare.

Prin acest exercițiu îți dezvoltă capacitatea de a nu accepta o soluție „ca atare”, ci de a evalua critic fiecare model pe baza unor metrii relevante și, astfel, de a contribui la luarea unor decizii mai bune în echipă.

## **De ce este această sarcină importantă?**

Fiabilitatea unui model de detectare a mesajelor spam influențează direct activitatea zilnică, siguranța și încrederea utilizatorilor. Să înțelegi cum să evaluezi și să compari diferite modele nu este doar o abilitate tehnică – este ceea ce te diferențiază ca profesionist capabil să propună echipei cea mai bună soluție.

Această sarcină te ajută să îți formezi o gândire critică, să înveți să recunoști punctele forte și slabe ale fiecărui model și să iei decizii care pot avea un impact real asupra activității companiei.

## **Ce ai la dispoziție? - Toolbox-ul tău pentru validarea modelului**

Pentru această sarcină, lucrezi într-un mediu realist, exact ca parte a unei echipe de date adevărate. Tot ce îți trebuie pentru analizarea și compararea modelelor este deja la îndemâna ta, însă de data aceasta în poveste includem și notebook-urile Jupyter. Ai la dispoziție:

- Setul de date:

Folosești același set de date ca înainte – [`messages.csv`](#). Acesta conține două coloane:

- `message` – textul fiecărui mesaj;
- `category` – eticheta (spam sau ham), dar ține cont că există și valori lipsă/greșite.

(La fel ca într-un mediu real de lucru: datele nu sunt întotdeauna perfecte, iar tu decizi cum le vei pregăti pentru analiză!)

- Instrumente și biblioteci Python:

Total se face într-un notebook Jupyter (sau Colab), folosind:

- `pandas` pentru lucrul cu date;
- `scikit-learn` pentru crearea, antrenarea și evaluarea modelelor;

- `matplotlib` sau `seaborn` pentru vizualizarea matricilor de confuzie și a rezultatelor.

## **Sarcina ta - Cum testezi și alegi cel mai bun model pentru întreaga echipă**

Trebuie să creezi un notebook Jupyter cu numele `spam_model_evaluation.ipynb`, în care:

- analizezi și pregătești datele din setul `messages.csv` astfel încât să poți antrena și testa modele pentru detectarea mesajelor spam;
- creezi și compari mai multe modele de clasificare (de exemplu: Logistic Regression, Naive Bayes, Decision Tree, Linear SVC sau altul ales de tine);
- evaluatezi performanța fiecărui model folosind metriki cheie (acuratețe, precizie, recall, măsura F1, matrice de confuzie);
- prezintă rezultatele atât vizual, cât și în text, astfel încât să fie clare pentru orice membru al echipei.

La final, trebuie să iei și să argumentezi o decizie:

Care model este cel mai de încredere pentru echipă? Pe ce criterii ai ajuns la această concluzie?

Se așteaptă ca notebook-ul tău să fie suficient de clar, comentat și structurat astfel încât echipa să poată folosi concluziile tale ca bază pentru următorul pas în îmbunătățirea filtrului de spam.

## **Ce trebuie să conțină notebook-ul tău?**

Ca orice raport bun de echipă, notebook-ul tău Jupyter trebuie să „spună o poveste” – de la primii pași până la alegerea finală a celui mai bun model.

Gândește-l ca pe un mini raport analitic pe care echipa îl va folosi pentru decizii ulterioare!

Notebook-ul tău trebuie să conțină:

- introduceri și explicații în Markdown pentru fiecare parte cheie – scurte, clare, ca și cum ai explica unui coleg pentru prima dată ce faci și de ce;
- prezentarea informațiilor de bază despre setul de date – ca oricine să înțeleagă rapid cu ce lucrezi (număr de mesaje, raport spam/ham etc.);
- vizualizări ale matricilor de confuzie pentru fiecare model testat – prezentarea grafică ajută echipa să vadă unde greșește modelul și unde excelează;
- interpretarea și compararea rezultatelor – analizează rapoartele de clasificare, compară metricile (precizia, recall, F1 etc.) și descrie ce te-a surprins sau la ce ar trebui să fie atenții colegii;
- Concluzie clară la final – scrie ce model alegi și de ce, explicând care metrică a fost cea mai importantă în decizie (de exemplu: „F1 este mai important pentru mine decât acuratețea, deoarece vreau să minimizez rezultatele fals negative”).

### **La sfârșitul notebook-ului...**

Scrie o concluzie textuală succintă în care:

- indici care algoritm oferă cele mai bune rezultate;
- explică cum s-au comportat celelalte modele;
- evidențiezi care metrică a fost priorită în luarea deciziei.

Amintește-ți: Acest notebook nu este doar un „exercițiu tehnic” – este instrumentul tău analitic pentru echipă, un ghid pentru oricine vrea să înțeleagă rapid și să îmbunătățească modelul de detectare a spamului!

**Hint - Experimentează, întreabă și împărtășește - aici este adevărata valoare!**

Această sarcină este șansa ta să vezi cum este să lucrezi ca un analist de date într-o echipă: scopul nu este să ai imediat „cea mai bună” soluție, ci să arăți cum ajungi la concluzie, să testezi alternative și să împărtășești des observațiile tale.

Sfaturi practice:

- Dacă rămâi blocat – folosește documentația oficială (pandas, scikit-learn, matplotlib...). Nu trebuie să știi totul din memorie!
- Testează cel puțin două-trei modele, chiar dacă crezi că știi care este „câștigătorul”. Surprizele reale apar adesea când compari rezultatele!
- Vizualizările sunt prietenii tăi: matricea de confuzie și graficele ajută echipa să vadă unde greșesc modelele și unde sunt excelente.
- Fiecare greșală este o oportunitate de învățare. Notează „fail”, descrie ce ai încercat – poate acea parte ajută pe alțineva.
- Nu uita: cea mai valoroasă este disponibilitatea ta de a explica alegerea modelului și de a face soluția utilă și clară pentru colegi.

În această sarcină, se apreciază mai mult deschiderea și experimentarea decât codul perfect din prima încercare.

Arată procesul, împărtășește momentele de „aha” și întreabă curajos când te blochezi – aşa lucrează echipele reale de date!

## Ghid pentru rezolvarea sarcinii

Iată toți pașii necesari pentru a realiza această sarcină

### 1. Crearea unui nou notebook

- Folosește mediul local Jupyter sau unul pregătit, cum ar fi Colab.
- Rezultatul acestui pas trebuie să fie un fișier cu extensia `.ipynb`.

### 2. Încărcarea și analiza setului de date

- Încarcă fișierul `messages.csv` într-un `Pandas DataFrame`.

- Afisează primele câteva rânduri pentru a vedea cum arată datele.
- Afisează numărul de valori lipsă pe fiecare coloană.
- Afisează toate valorile unice care există în coloana category.

### 3.Curățarea și pregătirea datelor

- Elimină toate rândurile care conțin valori NaN în coloanele message și category.
- Transformă toate valorile din coloana category în litere mici (lower()).
- Standardizează valorile din coloana category, astfel încât în final să existe doar valorile spam și ham.

Explicație: Astfel elimini inconvenientele și te concentrezi doar pe clasificarea binară.

### 4.Împărțirea datelor

- Împarte datele în X (mesaje de intrare) și y (valori țintă – spam sau ham).
- Folosește `train_test_split` pentru a împărți datele în proporție 80:20, cu `random_state=42` și `stratify=y`.

Explicație: Stratificarea asigură că seturile de antrenament și de testare au o distribuție similară a mesajelor spam/ham.

### 5.Vectorizarea textului

- Folosește `TfidfVectorizer` pentru a transforma mesajele textuale în formă numerică, pe care algoritmii o pot înțelege.
- Aplică `fit_transform()` pe datele de antrenament și `transform()` pe datele de testare.

Explicație: TF-IDF transformă cuvintele în numere care exprimă importanța lor în documente.

### 6.Antrenarea mai multor modele

- Creează un dicționar de modele cu următorii algoritmi:

- LogisticRegression
- MultinomialNB
- DecisionTreeClassifier
- LinearSVC
- Pentru fiecare model:
  - Antrenează modelul pe datele de antrenament.
  - Realizează predicții pe setul de testare.

## 7. Evaluarea performanței

- Pentru fiecare model afișează:
  - Acuratețea (accuracy\_score).
  - Raportul de clasificare (classification\_report).
  - Matricea de confuzie (confusion\_matrix).
  - Optional: Vizualizează matricea de confuzie folosind seaborn.heatmap.

Explicație: Astfel măsori cât de bun este fiecare model și unde greșește cel mai des.

## 8. Concluzie

- Compara toate rezultatele și decide:
  - Care model are cea mai bună acuratețe?
  - Care model recunoaște cel mai bine mesajele spam?
  - Există un echilibru între precizie și recall?
- Scrie o concluzie clară în care alegi modelul cel mai bun și explică de ce.

## Predarea sarcinii

Când termini, verifică dacă notebook-ul tău (`(spam_model_evaluation.ipynb)`) prezintă clar întregul proces: de la procesarea datelor, compararea modelelor, până la alegerea argumentată a celei mai bune soluții.

Notebook-ul trebuie să fie suficient de clar și comentat, astfel încât orice

coleg să poată urmări logica ta și să folosească munca ta ca punct de plecare pentru următorul pas al echipei.

## Cum predai sarcina?

- Arhivează fișierul `.ipynb` într-o arhivă rar sau zip.
- Denumește fișierul clar (de exemplu, `Prenume_Nume_Task2_SpamValidation.zip`).
- Încarcă arhiva prin formularul prevăzut pe platformă.

## Verificare rapidă înainte de predare:

- Raportul tău „spune o poveste” – poate cineva din echipă să urmeze ușor analizele și deciziile tale?
- Există vreo parte unde cineva s-ar putea încurca? Adaugă un scurt comentariu sau exemplu.
- Codul rulează fără eroare dacă cineva îl pornește de la început?

## Criterii de evaluare

Lucrarea ta va fi evaluată ca un mini-proiect analitic real, cu accent pe valoarea pe care o aduci echipei – nu doar prin cod, ci și prin claritatea și capacitatea de a explica deciziile.

| Criteriu                                  | Ce înseamnă în practică?  | Procentaj din notă |
|---|---|--------------------|
| Curățarea și pregătirea corectă a datelor | Se demonstrează recunoașterea și rezolvarea „dezordinii” din date – baza fiecărui model fiabil. | 10%                |
| Utilizarea mai multor algoritmi           | Nu se concentrează doar pe o singură soluție – alternative explorate.                           | 20%                |
| Prezentarea și interpretarea metricilor   | Se arată clar unde modelul funcționează   | 30%                |

|  |   |     |
|--|---|-----|
|  | bine și unde greșește, folosind metricile corecte și vizualizări.                         |     |
| Argumentarea și concluzia asupra celui mai bun model | Soluția aleasă este susținută cu argumente pe care echipa le înțelege și le poate folosi. | 20% |
| Organizarea și claritatea notebook-ului              | Codul este clar, ordonat, comentariile sunt prezente și organizarea logică.               | 20% |

## Metoda de evaluare

Lucrarea ta va fi evaluată pe o scară de la 0 la 5 stele, în funcție de procentul de îndeplinire a criteriilor din secțiunea anterioară.

Intervale și semnificația notelor:

(0-59%): Insuficient.

Cerințele de bază nu sunt îndeplinite – codul este incomplet sau nu funcționează, comentariile și explicațiile sunt insuficiente, analiza nu aduce valoare echipei. Aceasta este o oportunitate de a face o nouă iterație și de a-ți consolida prezentarea.

(60-69%): Partial.

Pașii de bază sunt prezenți, dar lipsesc analizele esențiale sau comentariile sunt minime. Codul funcționează, dar povestea analitică nu este completă sau logica este greu de urmărit.

(70-79%): Satisfăcător.

Majoritatea cerințelor sunt îndeplinite. Modelele funcționează, există comentarii de bază, analiza este corectă – însă nu este încă „gata pentru echipă” fără explicații suplimentare sau ajustări.

**□□□** (80-89%): Bine.

Codul este ordonat, comentariile sunt clare, iar analiza ta demonstrează siguranță în lucrul cu datele. Soluția este utilă echipei și poate fi imediat partajată pentru discuție.

**□□□□** (90-95%): Foarte bine

Totul funcționează, explicațiile sunt precise, iar analiza ta ghidează clar echipa prin evaluarea modelului - demonstrezi maturitate și înțelegerea contextului mai larg.

**□□□□□** (96-100%): Excepțional!

Notebook-ul tău este o referință - educativ, clar explicat și ușor de aplicat. Colegii și echipa pot folosi imediat soluția ta ca exemplu de bună practică.

Această scară te ajută să îți autoevaluezi progresul, să recunoști punctele forte ale lucrării tale și să identifici unde mai poți îmbunătăți abordarea.