

# **Hate speech, offensive language identification, misogyny / stereotype detection**

**-Hate speech Analysis-**

Echipa 6

Februarie 2024

# Contents

|          |                                   |           |
|----------|-----------------------------------|-----------|
| <b>1</b> | <b>Introdúcere</b>                | <b>3</b>  |
| <b>2</b> | <b>Setul de date</b>              | <b>3</b>  |
| <b>3</b> | <b>Extragerea feature-urilor</b>  | <b>5</b>  |
| 3.1      | CBoW . . . . .                    | 5         |
| 3.2      | TF-IDF . . . . .                  | 5         |
| <b>4</b> | <b>Modele de clasificatori</b>    | <b>5</b>  |
| 4.1      | Multinomial Naive Bayes . . . . . | 5         |
| 4.2      | Ridge Classifier . . . . .        | 6         |
| 4.3      | SVC (kernel ‘rbf’) . . . . .      | 7         |
| 4.4      | Linear SVC . . . . .              | 8         |
| 4.5      | Gradient Boosting . . . . .       | 9         |
| 4.6      | MLP Classifier . . . . .          | 10        |
| 4.7      | XGBoost . . . . .                 | 11        |
| 4.8      | Deep Neural Network . . . . .     | 12        |
| 4.9      | Modele preantrenate . . . . .     | 13        |
| <b>5</b> | <b>Bibliografie</b>               | <b>15</b> |

# 1 Introducere

Echipa noastră (Carpineanu Alexandru, Cuciureanu Dragos-Adrian, Vasile George-Cristian) a ales tema ”#15 Hate speech, offensive language identification, misogyny / stereotype detection”.

Discursul ce instigă ură - denumit în continuare hate speech - este diferit de libera exprimare, deoarece nu este limitat la exprimarea unor opinii controversate, în schimb implică propagarea discriminării și a violenței. Hate speech-ul a fost întotdeauna o problemă cu care societatea a trebuit să se confrunte, însă, de la apariția internetului, acesta a fost propulsat în ochii tuturor oamenilor. Orice utilizator putea să întâlnească vorbe ofensive fără să vrea.

Problema hate speech-ului este prevăzută în legile multor țări, deși interpretarea acestora poate fi variată și identificarea sa mai dificilă. În special în mediul online, unde volumul la care sunt transmise informațiile, filtrarea canalelor de convorbiri și restricționarea accesului pentru entitățile ce vorbesc cu ură este o problemă ce necesită o soluție eficientă.

Majoritatea platformelor sociale folosesc deja inteligența artificială pentru a detecta hate speech-ul și a-l cenzura automat, însă mereu se pot aduce îmbunătățiri acestor modele pentru a ne asigura că cenzurarea se aplică doar asupra hate speech-ului, orice greșală a modelului putând să însemne o încălcare a dreptului la liberă exprimare a utilizatorilor.

## 2 Setul de date

Pentru alegerea setului de date am folosit platforma Hugging Face datorită numărului variat de opțiuni pe care îl oferă. Am ales setul de date “tweets\_hate\_speech\_detection”.

Am ales platforma Twitter deoarece postările care puteau fi făcute pe aceasta până recent erau limitate la 280 caractere, astfel tweet-urile fiind un mijloc simplu și bun pentru detectarea hate speech-ului. În contextul nostru, am definit problema clasificării unui tweet ca fiind unul de hate speech prin identificarea în conținutul unor texte ce au conotații rasiste și/sau sexiste.

Astfel, fiecare tweet din setul de date este etichetat cu 0 sau cu 1, eticheta 1 semnificând prezența rasismului/sexismului. Cele 31.962 de tweet-uri valabile sunt scrise majoritar în limba engleză. Dintre acestea doar 2.242 sunt clasificate cu 1, astfel tweeturile cu 0 fiind predominante. Pentru acest fapt, am

selectat doar 10.000 dintre acestea.

Din punct de vedere al împărțirii datelor, am selectat 60% pentru antrenare și 40% pentru testare (ponderile fiind ușor de modificat, dacă se dorește mărirea setului de antrenare și diminuarea celui de testare). Totodată, setul de date a fost preprocesat pentru a ajunge la o formă normalizată potrivită pentru clasificatori.

Pentru preprocesare am aplicat următoarele operații:

- Transpunerea textelor unicode în caractere cu cod ASCII.
- Transformarea literelor mari în litere mici, deoarece această caracteristică a lor nu este relevantă în procesul nostru.
- Înlăturarea hyperlinkurilor, ele fiind adrese către resurse web și neprezentând nicio semnificație lingvistică.
- Eliminarea numelor de utilizator, identificate prin caracterul “@” urmat de un cuvânt format doar din litere, numere și simbolul “\_”.
- Eliminarea hashtag-urilor. În postări de hate speech hashtag-urile pot fi utilizate pentru a organiza întâlniri bazate pe promovarea unor ideologii rasiste sau instigative la ură, poluând analiza conținutului propriu-zis al postării.
- Înlăturarea punctuației, neavând o influență în identificarea hate speech-ului.
- Înlăturarea caracterelor numerice.
- Normalizarea caracterelor repetate. Utilizatorii serviciilor de chat au tendința de a multiplica vocale sau uneori și consoane din diferite cuvinte pentru a întări anumite idei. În orice caz, aceasta formă a mesajelor nu este relevantă în identificarea hate speech-ului, de asemenea pot îngreuna procesul de analiză al modelelor prin nerecunoașterea anumitor cuvinte.
- Lematizarea, având rolul de a reduce varietatea lingvistică redundantă prin proiecția diferitelor forme ale cuvintelor (conjugări, declinări, articulații) în forma lor de baza.
- Eliminarea stopwords. Acestea reprezintă cele mai comune cuvinte dintr-o limbă și în general sunt prepozitii, conjunctii, articole, neavând sens de sine stătător.
- Eliminarea “caracterelor albe”, precum “space”, “tab”, “enter”.

## 3 Extragerea feature-urilor

Am ales două metode de a extrage feature-urile, **TF-IDF** (Term Frequency - Inverse Document Frequency) și **CBoW** (Continuous Bag of Words). Am antrenat și testat fiecare model pe ambele tipuri de feature.

### 3.1 CBoW

Continuous Bag of Words este o metodă de extragere a feature-urilor ce utilizează o rețea neuronală pre-antrenată cu 2 layere, care reconstruiește contextul cuvintelor primite ca input după vectorii creați în urma antrenării. Aceasta ajustează n-vectorii, astfel încât cuvintele găsite în contexte similare apar apropiate unele de altele în spațiul vectorial creat la antrenare.

### 3.2 TF-IDF

Term frequency - Inverse document frequency este un mod de extragere a feature-urilor ce evaluează importanța fiecărui cuvânt din corpus, în relație cu numărul său de apariții în fiecare document.

Spre exemplu, cuvinte precum 'the' sau 'is' au o importanță mult mai mică față de 'blacklives' într-un corpus de hate-speech.

TF-IDF nu ia în considerare contextul în care regăsim cuvintele în propoziție.

## 4 Modele de clasificatori

Am ales o gamă variată de modele, pentru a observa diferențele dintre ele. Mai mult, am folosit și modele pre-antrenate preluate tot de pe platforma HuggingFace pentru a avea un spațiu de comparație mai mare.

### 4.1 Multinomial Naive Bayes

Un algoritm probabilist care utilizează teorema lui Bayes ce presupune independența caracteristicilor pentru a decide clasa din care face parte input-ul.

Parametrii:

- **alfa**: Un parametru de netezire. Ajută la evitarea overfitting ului.

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.84      | 1.00   | 0.91     | 4021    |
| 1            | 0.99      | 0.12   | 0.22     | 876     |
| accuracy     |           |        | 0.84     | 4897    |
| macro avg    | 0.92      | 0.56   | 0.57     | 4897    |
| weighted avg | 0.87      | 0.84   | 0.79     | 4897    |

Figure 1: Tf-Idf

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.85      | 0.97   | 0.91     | 4021    |
| 1            | 0.64      | 0.22   | 0.32     | 876     |
| accuracy     |           |        | 0.84     | 4897    |
| macro avg    | 0.75      | 0.60   | 0.62     | 4897    |
| weighted avg | 0.81      | 0.84   | 0.80     | 4897    |

Figure 2: CBoW

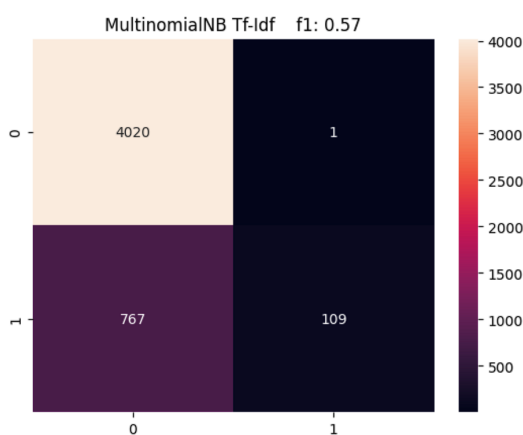


Figure 3: Tf-Idf

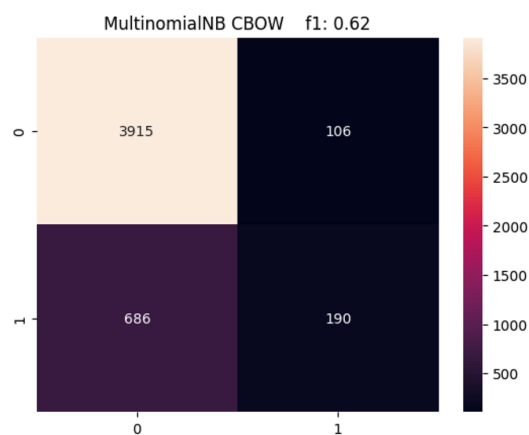


Figure 4: CBoW

## 4.2 Ridge Classifier

Clasificator binar care folosește regresia Ridge. Acesta penalizează complexitatea pentru a evita overfitting ul.

Parametrii:

- **alfa**: Utilizat pentru a controla puterea regularizarii. Reduce variația aproximărilor.
- **solver**: Algoritmul folosit pentru a rezolva problema de optimizare asociată modelului.

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.93      | 0.99   | 0.96     | 4021    |
| 1            | 0.91      | 0.67   | 0.77     | 876     |
| accuracy     |           |        | 0.93     | 4897    |
| macro avg    | 0.92      | 0.83   | 0.86     | 4897    |
| weighted avg | 0.93      | 0.93   | 0.92     | 4897    |

Figure 5: Tf-Idf

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.88      | 0.98   | 0.92     | 4021    |
| 1            | 0.77      | 0.36   | 0.49     | 876     |
| accuracy     |           |        | 0.87     | 4897    |
| macro avg    | 0.82      | 0.67   | 0.71     | 4897    |
| weighted avg | 0.86      | 0.87   | 0.85     | 4897    |

Figure 6: CBoW

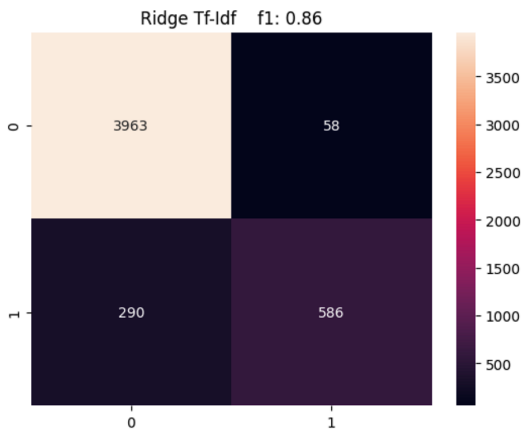


Figure 7: Tf-Idf

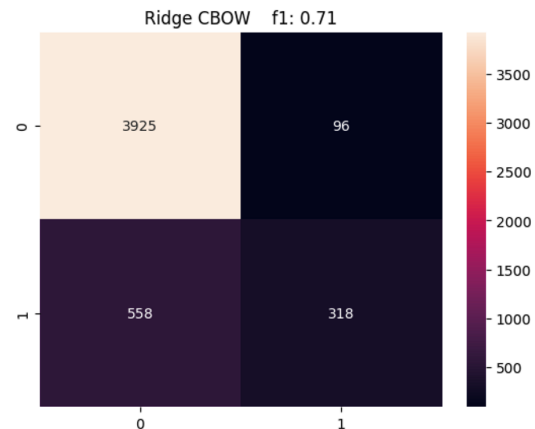


Figure 8: CBoW

### 4.3 SVC (kernel 'rbf')

Un algoritm de clasificare binară ce funcționează prin transformarea inputului într-un spațiu dimensional mai mare folosind funcții neliniare numite funcții radiale de baza (RBF) kernel. În acest spațiu mai mare, algoritmul încearcă să găsească un hyperplan optim care separă cele două clase de date maximizând separarea dintre clase (distanța minimă dintre un element al primei clase și un element al celei de-a doua).

Câteva parametrii cheie sunt:

- **C**: Parametrul de regularizare, ce determină balanța dintre a obține o margine mai mare și permiterea unor clasificări eronate. O valoare mai mică a lui C permite o margine mai mare, iar o valoare mai mică a sa implică o clasificare mai strictă.
- **gamma** (coeficientul kernel): Definese influența unui singur exemplu de antrenament asupra liniei de decizie. Un gamma mai mic înseamnă o influență mai largă și poate rezulta într-o linie de decizie mai uniformă, în timp ce un gamma mare înseamnă o influență mai restrânsă și poate duce la o linie de decizie mai complexă și mai ondulantă.
- **kernel**: Alegerea kernelului poate avea un impact semnificativ asupra performanței algoritmului.

RBF SVC utilizează implicit kernelul RBF, dar pot fi utilizate și alte kerneluri, cum ar fi cel liniar, polinomial sau sigmoidal. Fiecare kernel are proprii sai hyperparametri, cum ar fi gradul pentru kernelurile polinomiale sau coef0 pentru kernelurile sigmoidale, care controlează comportamentul acestora.

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.91      | 0.99   | 0.95     | 4021    |
| 1            | 0.93      | 0.56   | 0.70     | 876     |
| accuracy     |           |        | 0.91     | 4897    |
| macro avg    | 0.92      | 0.78   | 0.83     | 4897    |
| weighted avg | 0.92      | 0.91   | 0.91     | 4897    |

Figure 9: Tf-Idf

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.89      | 0.97   | 0.93     | 4021    |
| 1            | 0.79      | 0.44   | 0.56     | 876     |
| accuracy     |           |        | 0.88     | 4897    |
| macro avg    | 0.84      | 0.71   | 0.75     | 4897    |
| weighted avg | 0.87      | 0.88   | 0.86     | 4897    |

Figure 10: CBoW

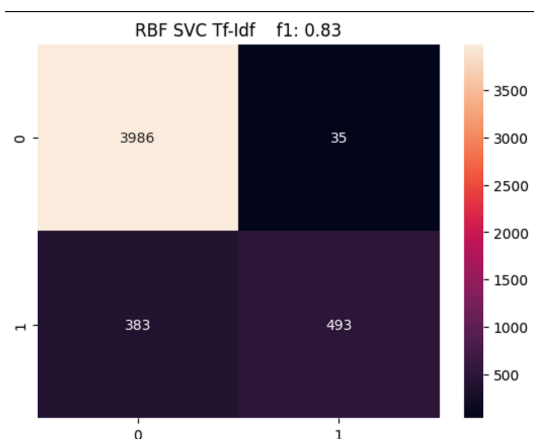


Figure 11: Tf-Idf

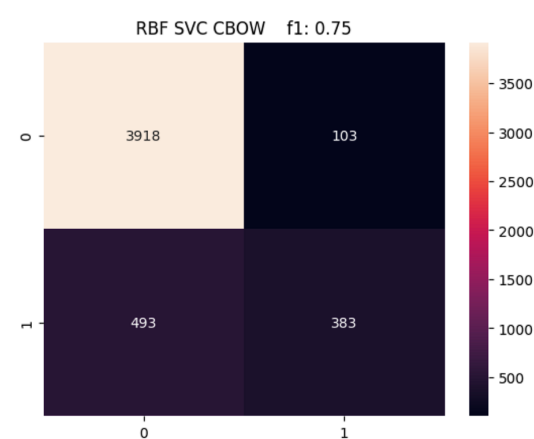


Figure 12: CBoW

## 4.4 Linear SVC

Algoritm de clasificare folosind masini cu vectori suport. Se ocupă cu căutarea hyperplanelor care separă clasele de date într-un spațiu dimensional mai mare. Este asemanator cu RBF SVC, însă folosește un kernel linear, iar hyperplanul rezultat este o linie dreaptă.

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.94      | 0.98   | 0.96     | 4021    |
| 1            | 0.89      | 0.71   | 0.79     | 876     |
| accuracy     |           |        | 0.93     | 4897    |
| macro avg    | 0.91      | 0.85   | 0.88     | 4897    |
| weighted avg | 0.93      | 0.93   | 0.93     | 4897    |

Figure 13: Tf-Idf

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.89      | 0.97   | 0.93     | 4021    |
| 1            | 0.76      | 0.47   | 0.58     | 876     |
| accuracy     |           |        | 0.88     | 4897    |
| macro avg    | 0.83      | 0.72   | 0.75     | 4897    |
| weighted avg | 0.87      | 0.88   | 0.87     | 4897    |

Figure 14: CBoW



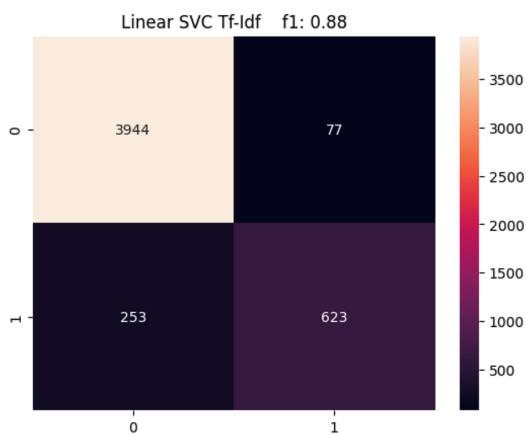


Figure 15: Tf-Idf

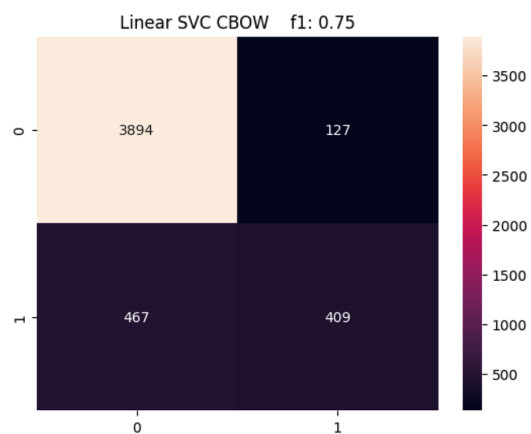


Figure 16: CBOW

## 4.5 Gradient Boosting

Acest algoritm construiește un model aditiv prin etape, permițând optimizarea funcțiilor de pierdere diferențiabile arbitrare.

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.90      | 0.98   | 0.94     | 4021    |
| 1            | 0.87      | 0.51   | 0.64     | 876     |
| accuracy     |           |        | 0.90     | 4897    |
| macro avg    | 0.88      | 0.74   | 0.79     | 4897    |
| weighted avg | 0.90      | 0.90   | 0.89     | 4897    |

Figure 17: Tf-Idf

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.89      | 0.97   | 0.93     | 4021    |
| 1            | 0.76      | 0.47   | 0.58     | 876     |
| accuracy     |           |        | 0.88     | 4897    |
| macro avg    | 0.83      | 0.72   | 0.76     | 4897    |
| weighted avg | 0.87      | 0.88   | 0.87     | 4897    |

Figure 18: CBOW

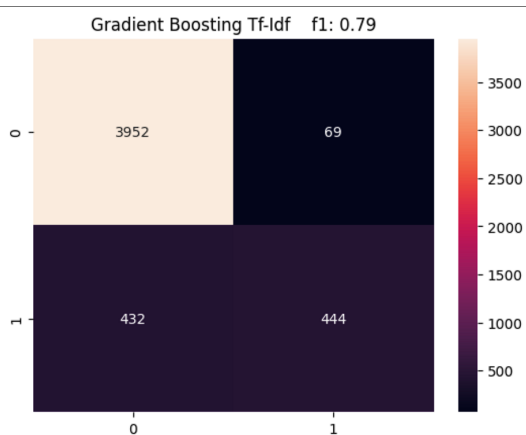


Figure 19: Tf-Idf

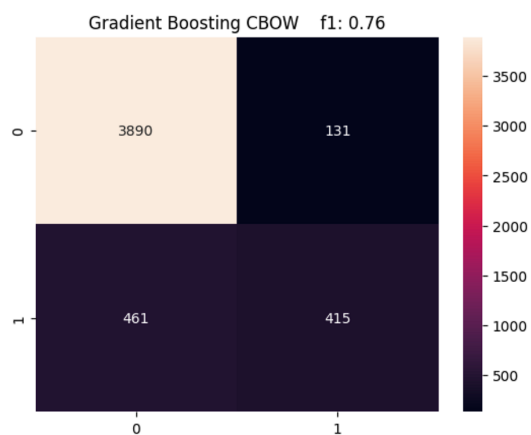


Figure 20: CBOW

## 4.6 MLP Classifier

Constă din mai multe straturi de neuroni artificiali. Orice rețea neuronală are un strat de intrare. Pe lângă acestea două, un MLP are un număr variat de straturi ascunse. Fiecare neuron primește input de la un alt neuron dintr-un strat precedent, îl modifică și îl trimite mai departe până la stratul de ieșire, unde se obține rezultatul final al rețelei.

Câțiva parametrii cheie sunt:

- **solver**: Algoritmul care rezolvă problema de optimizare.
- **alfa**: Puterea termenului de regresie Ridge.
- **hidden\_layer\_sizes**: Numărul de neuroni din straturile ascunse.
- **max\_iter**: Numărul maxim de iterații.
- **random\_state**: Determină generarea de numere aleatoare pentru inițializare.

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.95      | 0.97   | 0.96     | 4021    |
| 1            | 0.83      | 0.76   | 0.80     | 876     |
| accuracy     |           |        | 0.93     | 4897    |
| macro avg    | 0.89      | 0.86   | 0.88     | 4897    |
| weighted avg | 0.93      | 0.93   | 0.93     | 4897    |

Figure 21: Tf-Idf

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.91      | 0.90   | 0.91     | 4021    |
| 1            | 0.57      | 0.60   | 0.58     | 876     |
| accuracy     |           |        | 0.85     | 4897    |
| macro avg    | 0.74      | 0.75   | 0.75     | 4897    |
| weighted avg | 0.85      | 0.85   | 0.85     | 4897    |

Figure 22: CBoW

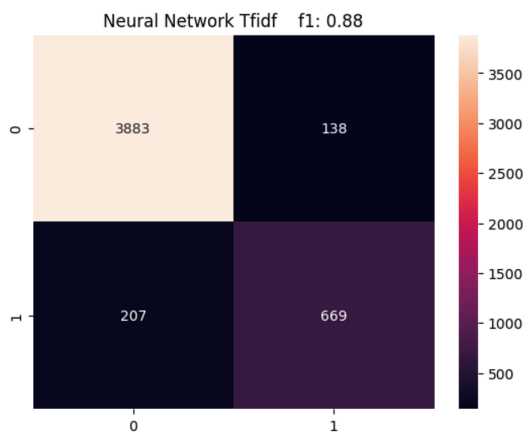


Figure 23: Tf-Idf

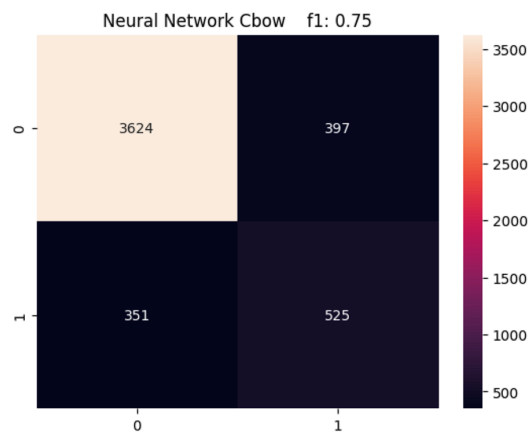


Figure 24: CBoW

## 4.7 XGBoost

Extreme Gradient Boosting este un algoritm proiectat pentru a optimiza performanța modelelor de gradient boosting ce folosește combinații de tehnici de boosting și regularizare. XGBoost îmbunătățește algoritmi tradiționali de gradient boosting prin introducerea mai multor inovații cheie: Optimizarea bazată pe gradient, regularizarea, tăierea arborilor, tratarea valorilor lipsa. Este eficient în rezolvarea problemelor de clasificare, regresie și clasificare a rangului.

Câțiva parametrii cheie sunt:

- **eta** (sau `learning_rate`): Are rolul de a preveni overfittingul.
- **gamma** (sau `min_split_loss`): Specifica reducerea minima a pierderii necesara pentru a produce o divizare.
- **max\_depth**: Adâncimea maximă a unui arbore. Este folosit în controlul overfitting-ului, intrucat adâncimea mai mare va permite modelului sa invete relații foarte specifice unui sample particular. Crescand aceasta valoare modelul devine mai complex și este foarte probabil sa produca overfitting.
- **subsample**: Rata de subsampling a instanțelor de învățare. Valori mai mici fac algoritmul mai conservator si previn overfittingul, inasa valori prea mici pot conduce la underfitting.
- **colsample\_bytree**: Rata de subsampling a coloanelor cand se construiește fiecare arbore. Subsampling-ul are loc odată pentru fiecare arbore construit.
- **n\_estimators**: Numărul de estimatori.

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.93      | 0.97   | 0.95     | 4021    |
| 1            | 0.83      | 0.64   | 0.73     | 876     |
| accuracy     |           |        | 0.91     | 4897    |
| macro avg    | 0.88      | 0.81   | 0.84     | 4897    |
| weighted avg | 0.91      | 0.91   | 0.91     | 4897    |

Figure 25:

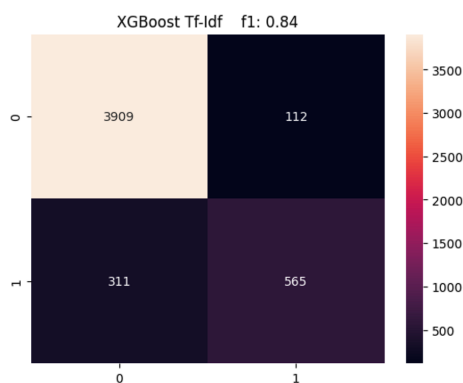


Figure 26:

## 4.8 Deep Neural Network

Rețelele Neuronale, cunoscute și ca Rețele Neuronale Artificiale sau Rețele Neuronale Simulate, sunt “inima” algoritmilor de învățare în profunzime (deep learning). Structura și numele lor sunt inspirate de creierul uman, imitând modul în care neuronii transmit semnale de la unul la altul.

În compoziția lor, rețelele neuronale artificiale au straturi de noduri (sau neuroni), dintre care unul este stratul de input, unul de output și celelalte sunt straturile ascunse. Fiecare nod este conectat la altul și are asociată o pondere și un prag. Dacă outputul unui neuron depășește valoarea pragului, neuronul este activat și trimite informația la următorul neuron din rețea, aflându-se în următorul strat.

Rețelele neuronale se bazează pe învățarea cu date de antrenare și își îmbunătățesc capacitățile treptat, însă odată rafinate, devin aparate puternice de mare viteză de clasificare și grupare în clustere a datelor.

Un model bazat pe învățarea în profunzime este Word2vec, recunoscut pentru utilizările sale în NLP. El învață asocierile dintre grupuri mari de cuvinte, utilizând un vector numeric pentru reținerea fiecărui cuvânt, redând similaritatea acestora. Printre algoritmi utilizati se află skip-gram și CBoW.

Câteva parametrii cheie sunt:

- **vector\_size**: Dimensiunea vectorilor de cuvinte.
- **feature\_size**: Dimensiunea vectorului de feature-uri.
- **workers**: Determină numărul de threaduri folosite pentru antrenarea modelului.
- **epochs**: Numărul de epoci în care se iterează textul.
- **total\_examples**: Numărul total de propoziții.

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.92      | 0.97   | 0.95     | 4021    |
| 1            | 0.81      | 0.64   | 0.71     | 876     |
| accuracy     |           |        | 0.91     | 4897    |
| macro avg    | 0.87      | 0.80   | 0.83     | 4897    |
| weighted avg | 0.90      | 0.91   | 0.90     | 4897    |

Figure 27: Raport de clasificare

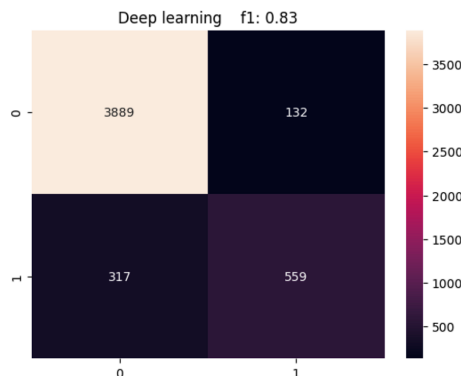


Figure 28: Matrice de confuzie

## 4.9 Modele preantrenate

Modele preantrenate folosite:

1. Obținut prin rafinarea modelului BERT(Bidirectional Encoder Representations from Transformers) cu datele de antrenament din acest set de date: Offensive Language Identification Dataset. [Link model.](#)

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.85      | 0.92   | 0.89     | 4021    |
| 1            | 0.43      | 0.26   | 0.32     | 876     |
| accuracy     |           |        | 0.80     | 4897    |
| macro avg    | 0.64      | 0.59   | 0.60     | 4897    |
| weighted avg | 0.78      | 0.80   | 0.79     | 4897    |

Figure 29: Raport de clasificare

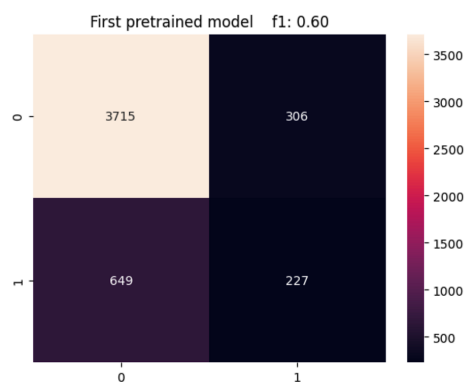


Figure 30: Matrice de confuzie

2. O alta rafinare a modelului BERT, pe acest dataset: MLMA, folosind Regularizarea atentiei bazata pe entropie. [Link model.](#)

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.91      | 0.49   | 0.63     | 4021    |
| 1            | 0.25      | 0.77   | 0.37     | 876     |
| accuracy     |           |        | 0.54     | 4897    |
| macro avg    | 0.58      | 0.63   | 0.50     | 4897    |
| weighted avg | 0.79      | 0.54   | 0.59     | 4897    |

Figure 31: Raport de clasificare

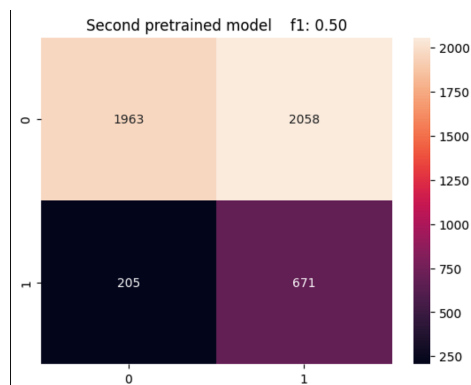


Figure 32: Matrice de confuzie

### 3. Alt model BERT preantrenat. Link model.

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.85      | 0.96   | 0.90     | 4021    |
| 1            | 0.53      | 0.19   | 0.28     | 876     |
| accuracy     |           |        | 0.82     | 4897    |
| macro avg    | 0.69      | 0.58   | 0.59     | 4897    |
| weighted avg | 0.79      | 0.82   | 0.79     | 4897    |

Figure 33: Raport de clasificare

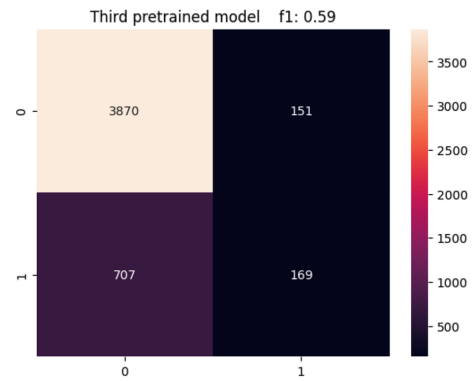


Figure 34: Matrice de confuzie

### 4. Forma de baza a modelului BERT. Link model.

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.83      | 0.98   | 0.90     | 4021    |
| 1            | 0.51      | 0.07   | 0.13     | 876     |
| accuracy     |           |        | 0.82     | 4897    |
| macro avg    | 0.67      | 0.53   | 0.51     | 4897    |
| weighted avg | 0.77      | 0.82   | 0.76     | 4897    |

Figure 35: Raport de clasificare

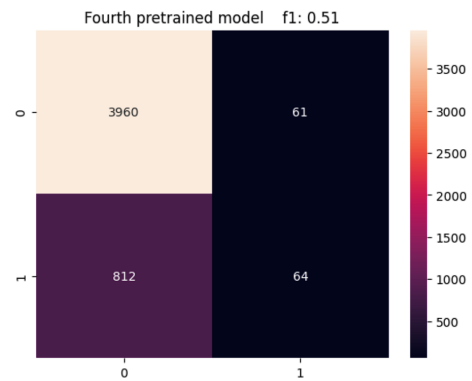


Figure 36: Matrice de confuzie

### 5. O alta rafinare pentru limba engleza a modelului BERT. Link model.

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.86      | 0.84   | 0.85     | 4021    |
| 1            | 0.34      | 0.38   | 0.36     | 876     |
| accuracy     |           |        | 0.76     | 4897    |
| macro avg    | 0.60      | 0.61   | 0.60     | 4897    |
| weighted avg | 0.77      | 0.76   | 0.76     | 4897    |

Figure 37: Raport de clasificare

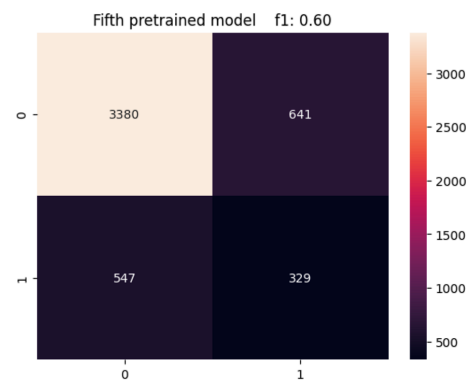


Figure 38: Matrice de confuzie

## Bibliografie

- [1] Attanasio, G., Nozza, D., Hovy, D., & Baralis, E. (2022). Entropy-based attention regularization frees unintended bias mitigation from lists. *Findings of the Association for Computational Linguistics: ACL 2022*. [Link](#)
- [2] Vidgen, B., Thrush, T., Waseem, Z., & Kiela, D. (2021). Learning from the worst: Dynamically generated datasets to improve online hate detection. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. [Link](#)
- [3] Mathew, B., Saha, P., Yimam, S. M., Biemann, C., Goyal, P., & Mukherjee, A. (2021). Hatexplain: A benchmark dataset for explainable hate speech detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17), 14867–14875. [Link](#)
- [4] Aluru, S. S. (2020). Deep Learning Models for Multilingual Hate Speech Detection. [Link](#)
- [5] Akhtar, S. (2021). Whose Opinions Matter? Perspective-aware Models to Identify Opinions of Hate Speech Victims in Abusive Language Detection. [Link](#)