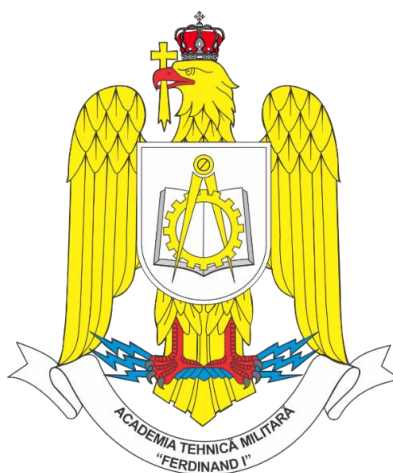


**ROMÂNIA**  
**MINISTERUL APĂRĂRII NAȚIONALE**  
**ACADEMIA TEHNICĂ MILITARĂ „FERDINAND I”**

**FACULTATEA DE SISTEME INFORMATICE ȘI SECURITATE**  
**CIBERNETICĂ**

*Calculatoare și Sisteme Informatice Pentru Apărare și Securitate Națională*



**Detecția afecțiunilor psihice**  
**folosind algoritmi de învățare automată**

**CONDUCĂTOR ȘTIINȚIFIC:**  
**Lect. dr. ing. Stelian SPÎNU**

**ABSOLVENT:**  
**Sd. sg. maj. Alexandra-Codruța NEAGU**

Conține: \_\_\_\_\_ file  
Inventariat sub nr. \_\_\_\_\_  
Poziția din indicator: \_\_\_\_\_  
Termen de păstrare: \_\_\_\_\_

**București**  
**2023**



NECLASIFICAT

NECLASIFICAT

NECLASIFICAT

NECLASIFICAT

NECLASIFICAT

NECLASIFICAT



NECLASIFICAT

NECLASIFICAT

## Abstract

This research aims to conduct a detailed study on introducing the use of neural network models in mobile applications to predict and detect mental disorders from text based on a person's thoughts.

The first part of the study consists in the analysis of several algorithms (Random Forest), Multinomial Naive Bayes, Long Short Term Memory, Convolutional Neural Network, BERT, Mental BERT, RoBERTa, Mental RoBERTa), using different processing methods (TF-IDF, Tokenization, Toenization and GloVe embeddings) and training techniques (using weights for classes, augmentation, various dropout values, data generator), with the intention of using the best model in the future. It started from the benchmark algorithms with scores around 60% to the best models with scores of about 75% for the f1 score. Two models were finally chosen, a binary one based on a LSTM and a multi-class one based on the BERT transformer, having the best results.

The second part is developing a mobile application with a simple interface, which can serve as a journal and detect possible mental disorders from the text input. The user has the possibility to add notes in this interface that will be sent to a server and the values of each disorder will be calculated using the best models from the first part of the study. The results returned will be in the form of probabilities for both binary and multi-class models. Furthermore, the application does a time analysis of the results in a simple, user-friendly way by creating graphical visualisations of each condition.

The combination of these parts proves the hypothesis of early detection and prevention of the spread of mental disorders, helping users to learn about them in a quick and compact way through a publicly accessible app.

Pagină albă

## Rezumat

Această lucrare își propune să realizeze un studiu detaliat despre introducerea utilizării modelelor de rețele neuronale în aplicațiile mobile, pentru a prezice și detecta tulburările mentale din text, pe baza gândurilor unei persoane.

Prima parte a studiului constă în analiza mai multor algoritmi (Random Forest), Multinomial Naive Bayes, Long Short Term Memory, Convolutional Neural Network, BERT, Mental BERT, RoBERTa, Mental RoBERTa), utilizând diferite tehnici de procesare (TF-IDF, Tokenizare, Tokenizare și încorporări GloVe) și de antrenare (utilizarea ponderilor pentru clase, augmentarea, valori de abandon diferite, generator de date), în ideea utilizării celui mai bun model în viitor. S-a pornit de la algoritmi de referință, cu scoruri de aproximativ 60% până la cele mai bune modele cu scoruri de 75% în cazul scorului f1. S-au ales în final 2 modele, unul binar pe baza LSTM și unul multi-clasă pe baza transformerului BERT, cu cele mai bune rezultate.

A doua parte constă în dezvoltarea unei aplicații mobile cu o interfață simplă, care poate servi drept jurnal și poate detecta posibilele tulburări psihice din textul introdus. Utilizatorul are posibilitatea de a adăuga notițe în această interfață ce vor fi trimise la un server și calculate valorile fiecărei afecțiuni cu ajutorul celor mai bune modele din prima parte a studiului. Rezultatele returnate vor fi sub forma unor probabilități atât în cazul modelului binar cât și în cazul celui multi-clasă. Mai mult, aplicația face o analiză în timp a rezultatelor într-un mod simplu, ușor de înțeles pentru un utilizator, prin crearea unor vizualizări grafice a fiecărei afecțiuni.

Combinația acestor părți demonstrează ipoteza detectării timpurii și a prevenirii răspândirii unor tulburări psihice, ajutând utilizatorii să afle despre acestea într-un mod rapid și compact, printr-o aplicație accesibilă publicului.

NECLASIFICAT

Pagină albă

# Cuprins

Listă de abrevieri .....	15
Listă de figuri .....	17
Listă de tabele.....	20
1. Introducere.....	21
1.1. Context .....	21
1.2. Importanța temei.....	21
1.3. Motivația și scopul proiectului .....	22
1.4. Etică.....	22
2. Noțiuni teoretice .....	23
2.1. Inteligență artificială.....	23
2.1.1. Învățare automată (Machine Learning) .....	23
2.1.2. Învățare profundă (Deep Learning).....	24
2.1.3. Învățare supervizată.....	25
2.1.4. Învățare nesupervizată.....	26
2.2. Noțiuni de învățare automată folosite .....	26
2.2.1. Random Forest .....	26
2.2.2. Mutimonial Naive Bayes.....	26
2.2.3. Long Short Term Memory Model (LSTM).....	27
2.2.4. Convolutional Neural Network Model (CNN) .....	27
2.2.5. Bidirectional Encoder Representations from Transformers (BERT) .....	27
2.2.6. Robustly Optimized BERT Approach (RoBERTa) .....	27
2.2.7. Framework-uri folosite.....	27
2.2.8. Metrici de evaluare ale modelelor .....	28
2.2.9. Tehnici de preprocesare a datelor .....	29
2.3. Noțiuni teoretice în dezvoltarea mobilă .....	30
2.3.1. React-Native .....	30
3. Studii anterioare și cercetări relevante .....	33
4. Pregătirea mediului de lucru.....	36
5. Dezvoltarea proiectului .....	37

5.1. Analiza, descrierea și procesare bazelor de date folosite .....	37
5.2. Pregătirea datelor .....	48
5.2.1. Tipuri de transformare de la propoziții la valori numerice. ....	48
5.2.2. Generator de date.....	50
5.3. Algoritmi folosiți pentru detectarea afecțiunilor psihice.....	51
5.3.1. Long Short Term Memory (LSTM) – clasificator binar .....	51
5.3.2. Random Forest .....	56
5.3.3. Multimonial Naive Bayes.....	65
5.3.4. Long Short Term Memory (LSTM) .....	68
5.3.5. Convolutional Neural Network (CNN) .....	86
5.3.6. Bidirectional Encoder Representations from Transformers (BERT) .....	92
5.3.7. Robustly Optimized BERT (RoBERTa).....	99
5.4. Rezultate și comparații .....	104
6. Mentol Cloud – Aplicație mobilă de integrare .....	106
6.1. Cerințe software .....	106
6.1.1. Cerințe funcționale .....	106
6.1.2. Cerințe nefuncționale .....	107
6.2. Diagrame Unified Modeling Language .....	107
6.3. Server.....	109
6.4. Client .....	109
7. Concluzii și direcții viitoare de cercetare .....	113
Bibliografie.....	115
Articole Științifice .....	115
Cărți .....	116



## Listă de abrevieri

Abreviere	Engleză	Română
URL	Uniform Resource Locator	Localizator uniform de resurse
API	Application Programming Interface	Interfața de programare a aplicațiilor
CPU	Central Processing Unit	Unitate centrală de procesare
GPU	Graphic Processing Unit	Unitate de procesare grafică
NLP	Natural Language Processing	Procesarea limbajului natural
LAN	Local Area Network	Rețea locală
REST	Representation State Transfer	Transfer de stare de reprezentare
ROC	Receiver Operating Characteristics	Caracteristici operaționale ale receptorului
RF	Random Forest	Pădure aleatorie
MNB	Multinomial Naive Bayes	Multinomial Naive Bayes
LSTM	Long Short Term Memory	Memorie pe termen lung și scurt
CNN	Convolutional Neural Network	Rețea neuronală convoluțională
BERT	Bidirectional Encoder Representations from Transformers	Reprezentări bidirecționale ale codicatorului din transformatoare
RoBERTa	Robustly Optimized BERT Approach	BERT optimizat în mod robust
Acc	Accuracy	Acuratețe
P	Precision	Precizie
R	Recall	Regăsire
F1	F1 Score	Scor F1
IDE	Integrated development environment	Mediu de dezvoltare

NECLASIFICAT

Pagină albă

## Listă de figuri

Figură 2.1.1-1 Componente inteligență artificială .....	23
Figură 2.1.2-1 O rețea neuronală simplă .....	25
Figură 2.2.8-1 Exemplu matrice de confuzie .....	28
Figură 2.3.1-1 Distribuția pe clase a bazei de date .....	38
Figură 2.3.1-2 Distribuția cuvintelor în baza de date .....	39
Figură 2.3.1-3 Nor de cuvinte .....	40
Figură 2.3.1-4 Distribuție baza de date binară .....	41
Figură 2.3.1-5 Frecvența cuvintelor în baza de date binară .....	42
Figură 2.3.1-6 Nor de cuvinte .....	43
Figură 2.3.1-7 Frecvență cuvinte clasa nonmentală.....	43
Figură 2.3.1-8 Frecvență cuvinte clasa mentală.....	44
Figură 2.3.1-9 Distribuția pe clase a bazei de date .....	45
Figură 2.3.1-10 Distribuția claselor în baza de date – plăcintă.....	45
Figură 2.3.1-11 Frecvența cuvintelor în propoziții.....	48
Figură 5.2.1-1 Exemplu word_index.....	49
Figură 5.2.2-1 Arhitectură clasificare.....	51
Figură 5.3.1-1 Arhitectură model binar.....	52
Figură 5.3.1-2 Arhitectură model binar .....	53
Figură 5.3.1-3 Curbe de învățare înainte de auto-corectare .....	53
Figură 5.3.1-4 Curbe de învățare după auto-corectare .....	54
Figură 5.3.1-5 Raport de clasificare înainte și după auto-corectare.....	54
Figură 5.3.1-6 Matrice de confuzie înainte și după auto-corectare.....	54
Figură 5.3.1-7 Raport de clasificare cu 300 de cuvinte.....	55
Figură 5.3.1-8 Matrice de confuzie cu 300 de caractere .....	55
Figură 5.3.2-1 Arbore de decizie .....	56
Figură 5.3.2-2 Matrice de confuzie baza de date nr.1 – TFIDF.....	57
Figură 5.3.2-3 Matrice de confuzie baza de date nr 1 – GloVe .....	58
Figură 5.3.2-4 Curba ROC baza de date nr 1 – TFIDF.....	58
Figură 5.3.2-5 Curba ROC baza de date nr 1 – GloVe.....	59
Figură 5.3.2-6 Matrice de confuzie baza de date nr.2 - TFIDF.....	60
Figură 5.3.2-7 Matricea de confuzie baza de date nr.2 - GloVe.....	61
Figură 5.3.2-8 Matricea de confuzie baza de date nr. 2 – TFIDF – dupa autocorectare {5}.....	62
Figură 5.3.2-9 Matrice de confuzie baza de date nr.2 – GloVe – dupa autocorectare {6} .....	63

Figură 5.3.2-10 Raport de clasificare baza de date nr. 2 - TF-IDF {5} .....	64
Figură 5.3.2-11 Raport de clasificare baza de date nr.2 – GloVe {6} .....	64
Figură 5.3.3-1 Matrice de confuzie baza de date nr. 1 – TFIDF .....	66
Figură 5.3.3-2 Curba ROC baza de date nr.1 – TFIDF .....	66
Figură 5.3.3-3 Matrice de confuzie baza de date nr.2 - TFIDF .....	67
Figură 5.3.3-4 Raport de clasificare bazade date nr. 2 cu TFIDF {5} .....	68
Figură 5.3.4-1 Vizulizare visualkeras model LSTM .....	69
Figură 5.3.4-2 Secvență de cod pentru modelul LSTM .....	70
Figură 5.3.4-3 Visualizare model LSTM .....	71
Figură 5.3.4-4 Curbe de învățare model LSTM - BD1 - Tozenizare .....	72
Figură 5.3.4-5 Raport de clasificare LSTM - BD1 – Tokenizare .....	72
Figură 5.3.4-6 Matrice de confuzie LSTM - BD1 - Tokenizare .....	73
Figură 5.3.4-7 Curbe de învățare model LSTM - BD2 - Tokenizare + GloVe .....	73
Figură 5.3.4-8 Raport de clasificare LSTM - BD2 – GloVe .....	74
Figură 5.3.4-9 Matrice de confuzie LSTM - BD1 – GloVe .....	74
Figură 5.3.4-10 Curbe de învățare LSTM Tokenizare cu 70 cuvinte {3} .....	75
Figură 5.3.4-11 Raport clasificare LSTM ?Tokenizare 70 cuvinte {3} .....	76
Figură 5.3.4-12 Curbe de învățare LSTM Tokenizare 300 cuvinte {4} .....	76
Figură 5.3.4-13 Raport clasificare LSTM Tokenizare 300 cuvinte {4} .....	77
Figură 5.3.4-14 Curbe de învățare LSTM Tokenizare 70 cuvinte - grreutăți varianta 1 {5} .....	77
Figură 5.3.4-15 Raport clasificare LSTM Tokenizare 70 cuvinte – ponderi varianta 1 {5} .....	78
Figură 5.3.4-16 Curbe de învățare LSTM Tokenizare 70 cuvinte - ponderi varianta 2 {6} .....	78
Figură 5.3.4-17 Raport clasificare LSTM Tokenizare 70 cuvinte – ponderi vaianta 2 {6} .....	79
Figură 5.3.4-18 Matrice de confuzie LSTM Tokenizare 300 cuvinte {7} .....	80
Figură 5.3.4-19 Curbe de învățare LSTM GloVe 70 cuvinte {9} .....	81
Figură 5.3.4-21 Raport clasificare LSTM GloVe 70 cuvinte {9} .....	81
Figură 5.3.4-20 Curbe de învățare LSTM GloVe 300 cuvinte {10} .....	81
Figură 5.3.4-22 Raport clasificare LSTM GloVe 300 cuvinte {10} .....	82
Figură 5.3.4-23 Curbe de învățare LSTM GloVe 300 cuvinte ponderi varianta 2 {12} .....	82
Figură 5.3.4-24 Raport de clasificare LSTM GloVe 300 cuvinte ponderi varianta 2 {12} .....	83
Figură 5.3.4-25 Matrice de confuzie LSTM GloVe 300 cuvinte {10} .....	84
Figură 5.3.5-1 Secvență de cod pentru modelul CNN .....	86
Figură 5.3.5-2 Vizualizare model CNN .....	87
Figură 5.3.5-3 Curbe de învățare CNN – Tokenizare {1} .....	88

Figură 5.3.5-4 Raport de clasificare CNN – Tokenizare {1} .....	88
Figură 5.3.5-5 Matrice de confuzie CNN – Tokenizare {1} .....	89
Figură 5.3.5-6 Curbe de învățare CNN GloVe {2} .....	90
Figură 5.3.5-7 Raport de clasificare CNN GloVe {2} .....	90
Figură 5.3.5-8 Matrice de confuzie CNN GloVe {2} .....	91
Figură 5.3.6-1 Sumar model BERT .....	92
Figură 5.3.6-2 Raport clasificare BERT - base-uncased {2} .....	93
Figură 5.3.6-3 Matrice de confuzie BERT base-uncased {2} .....	94
Figură 5.3.6-4 Curbe de învățare BERT - base-uncased {3} .....	95
Figură 5.3.6-5 Raport clasificare BERT - base-uncased {3} .....	95
Figură 5.3.6-6 Matrice de confuzie BERT - base-uncased {3} .....	96
Figură 5.3.6-7 Curbe de învățare Mental BERT {9} .....	97
Figură 5.3.6-8 Raport clasificare Mental BERT {9} .....	97
Figură 5.3.6-9 Matrice de confuzie Mental BERT {9} .....	98
Figură 5.3.7-1 Curbe de învățare RoBERTa {3} .....	100
Figură 5.3.7-2 Raport clasificare RoBERTa {3} .....	100
Figură 5.3.7-3 Matrice de confuzie RoBERTa {3} .....	101
Figură 5.3.7-4 Curbe de învățare Mental RoBERTa {5} .....	102
Figură 5.3.7-5 Raport de clasificare Mental RoBERTa {5} .....	102
Figură 5.3.7-6 Matrice de confuzie Mental RoBERTa {5} .....	103
Figură 5.3.7-1 Arhitectură aplicație .....	106
Figură 6.1.2-1 Diagrama cazurilor de utilizare .....	107
Figură 6.1.2-2 Diagrama de secvență .....	108
Figură 6.1.2-2 Ecran 1 .....	110
Figură 6.1.2-2 Ecran 2 .....	110
Figură 6.1.2-4 Ecran 5 .....	111
Figură 6.1.2-4 Ecran 6 .....	111
Figură 6.1.2-6 Ecran 3 .....	111
Figură 6.1.2-6 Ecran 4 .....	111
Figură 6.1.2-8 Ecran 7 .....	112
Figură 6.1.2-8 Ecran 8 .....	112

## Listă de tabele

Tabel 1 Random Forest.....	65
Tabel 2 Multinomial naive bayes .....	68
Tabel 3 LSTM.....	85
Tabel 4 CNN.....	91
Tabel 5 Bert .....	98
Tabel 6 RoBERTa .....	103
Tabel 7 Final .....	105

# 1. Introducere

## 1.1. Context

Inteligența artificială a apărut inițial în anii 1955, ca apoi să ajungă în sistemul medical la începutul anilor 1970, unde s-a folosit pentru început la identificarea infecțiilor la nivelul sângelui. Acest demers a evoluat, astăzi inteligența artificială fiind folosită în domenii precum radiologia, psihiatria, diagnosticarea bolilor, telemedicină și lista continuă [15].

În ultimii ani s-a început înțelegerea rolului pe care îl are sănătatea mentală în realizarea activităților de zi cu zi și cât de importantă este pentru populație. În zilele noastre depresia este una din principalele cauze de incapacitate întâlnită ce afectează realizarea obiectivelor zilnice. Iar în cadrul tinerilor, suicidul este a patra cauza principală de deces. Față de acum câteva decenii, persoanele mor prematur din cauza unor afecțiuni psihice ce pot fi prevenite și tratate [15].

Telefonul a ajuns să fie un obiect indispensabil în cadrul vieții de zi cu zi, un număr de 91% din întreaga populație deținând un astfel de dispozitiv<sup>1</sup>. Dintre aceștia un număr de 56.8%<sup>2</sup> dețin și sunt activi pe rețelele de socializare, pentru a se descărca, a cere sfaturi, sau a interacționa cu persoane din alte colțuri ale planetei. Astfel, aplicațiile de mobil pot ajuta foarte mult oamenii în a se elibera prin intermediul scrisului într-un mediu sigur și privat.

## 1.2. Importanța temei

Detectarea afecțiunilor psihice și prevenirea răspândirii acestora sunt pași esențiali în dezvoltarea societății într-o manieră sănătoasă și echilibrată. Lucrarea propusă vine în sprijinul comunității științifice prin analiza unei game de algoritmi de învățare automată pentru detectarea afecțiunilor psihice și realizarea unor comparații pe baza mai multor tehnici de procesare, antrenare și vizualizare a întregului proces.

Aplicația finală vine în ajutorul persoanelor ce vor să se descarce prin scris dar în același timp să-și monitorizeze sănătatea mintală pe parcursul zilelor, pentru a ajuta la îmbunătățirea acesteia dar și pentru prevenirea și tratarea simptomelor avute.

---

<sup>1</sup> <https://www.bankmycell.com/blog/how-many-phones-are-in-the-world>

<sup>2</sup> <https://backlinko.com/social-media-users>

### **1.3. Motivația și scopul proiectului**

Datorită preocupării din ce în ce mai mare la nivel global asupra sănătății mintale, se dorește ca detecția timpurie să fie accesibilă într-un mod rapid, ușor și sigur. Motivația acestei lucrări este realizarea unei aplicații ce va folosi cele mai bune metode în detecția afecțiunilor psihice.

Pentru realizarea lucrării se va urmări cercetarea mai multor algoritmi folosiți în clasificarea textului, compararea acestora și alegerea celor mai bune modele în final. Acest proces constă în studierea literaturii existente din domeniul detecției afecțiunilor psihice și a culegerii de date relevante studiului.

A doua parte a lucrării va avea motivația de a crea o metodă ca studiul să fie accesibil ușor și pentru publicul larg. Din acest motiv s-a ajuns la concluzia ca o aplicație mobilă de tip jurnal ar fi cea mai bună variantă de utilizare a modelelor de clasificare.

Scopul proiectului este realizarea cu succes a unei astfel de aplicații ce va reuși să identifice existența unei afecțiuni psihice din text. Chiar dacă în momentul actual o astfel de aplicație nu poate înlocui controlul specializat, această cercetare ar trebui să fie încă un pas în realizarea automată a acestui lucru.

### **1.4. Etică**

Această lucrare nu detectează 100% existență unor afecțiuni psihice, fiind doar o lucrare de cercetare ce are ca scop dovedirea creării unui astfel de mecanism. Datorită datelor culese de pe rețelele de socializare și nu de la centre specializate în acest domeniu, informațiile din aceste postări nu sunt sigure a avea o legătură cu afecțiunea de care se presupune că utilizatorul suferă. Astfel, nici o predicție în cauză din acest proiect nu înseamnă ca este adevărată sau sigură și se recomandă controlul specializat în domeniu la nevoie.

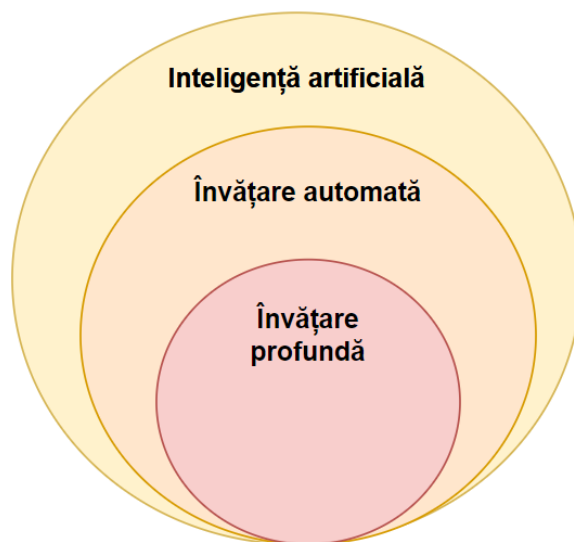


## 2. Noțiuni teoretice

În acest capitol se vor descrie conceptele teoretice folosite în dezvoltarea cercetării și aplicației mobile. Prima parte va vorbi despre inteligența artificială, noțiuni folosite, metode de clasificare și identificare și framework-uri. A doua parte descrie mecanismele folosite pentru implementarea aplicației mobile ce va pune în evidență importanța în societate a unor mecanisme asemănătoare.

### 2.1. Inteligență artificială

Inteligența artificială se referă la capacitatea sistemelor de a rezolva sarcini complexe și de a lua decizii autonome folosind diverse date și algoritmi. Acest domeniu este folosit în dorința de a automatiza și ușura sarcinile efectuate de oameni.



Figură 2.1.1-1 Componente inteligență artificială

#### 2.1.1. Învățare automată (Machine Learning)

Este ramura inteligenței automate care se ocupă de construirea unor modele ce pot „învăța” singure pe baza experiențelor. Se dorește construirea de algoritmi ce pot mima gândirea oamenilor pentru a-și îmbunătății pe parcurs acuratețea.

Procesul prin care algoritmi iau decizii se bazează pe preluarea datelor etichetate sau neetichetate și producerea unor estimări pentru tiparele descoperite. În timpul evaluării se folosește funcția de eroare ce se ocupă de evaluare predicției modelului și funcția de optimizare se ocupă de ajustarea parametrilor pentru eficientizarea procesului de predicție.

### 2.1.2. Învățare profundă (Deep Learning)

Învățarea profundă aparține de domeniul de învățare automată și se bazează pe suprapunerea mai multor straturi de neuroni conectate între ele și calcularea progresivă a parametrilor matematici numiți ponderi.

Neuronul este cel mai unitar și fundamental element în cadrul procesării. Un strat este format din mai mulți neuroni, fiecare neuron din stratul actual fiind conectat cu neuronii din stratul precedent și ascendent. Ponderile sunt valori matematice ce se schimbă continuu pentru o învățare cât mai bună, pe baza erorii rezultate.

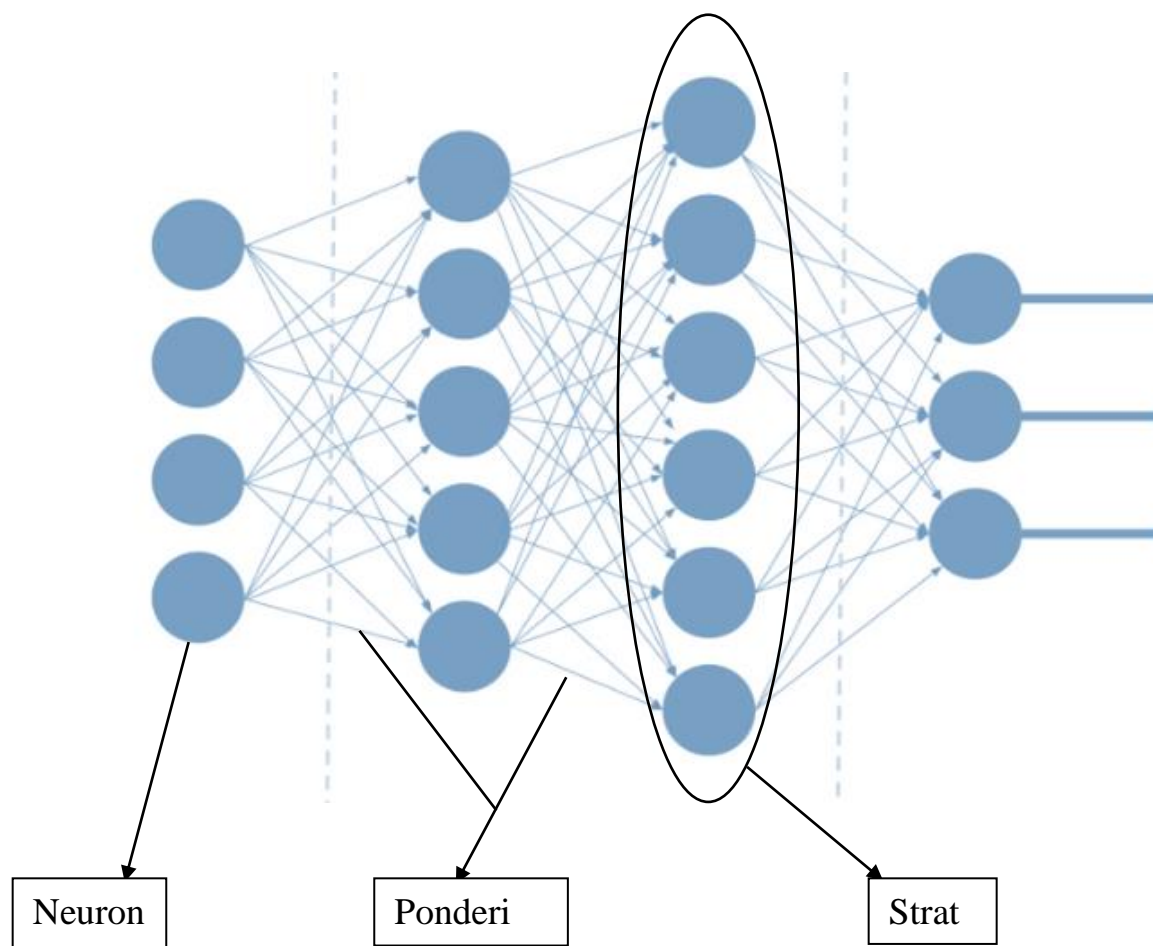
Complexitatea și profunzimea modelului este dată de numărul de straturi. Astfel primul strat servește preluării datelor de intrare iar ultimul corespunde ieșirilor. Între acestea există mai multe straturi ascunse, de mai multe tipuri:

- **Stratul de încorporare** (Embedding layer) – convertește intrările de tip text în reprezentări vectoriale.
- **Stratul recurent** – modelează informațiile secvențiale din text, captând atât datele din stratul actual cât și cele din stratul anterior.
- **Stratul convoluțional** – folosit pentru extragerea trăsăturilor din imagini și text.
- **Stratul de atenție** – se concentrează pe părțile relevante ale datelor de intrare în timp ce generează datele de ieșire.
- **Stratul dens** – este complet conectat, fiecare neuron din stratul X-1 este conectat la toți ceilalți neuroni din stratul X
- **Stratul de abandon** – folosit pentru a preveni supraînvățarea, dezactivează o proporție aleatoare de neuroni.
- **Max pooling** – folosit în calcularea valorii maxime dintr-o fereastră a filtrului, pentru a reduce dimensiunea.

Aceste straturi dispun de o funcție de activare. Aceasta definește ieșirea unui neuron având în vedere intrarea și poate fi de mai multe tipuri, ca de exemplu:

- **Treaptă unitate** –  $f(x) = \begin{cases} 0 & \text{pentru } x < 0 \\ 1 & \text{altfel} \end{cases}$ , cea mai simplă funcție.
- **Lineară** –  $f(x) = \{x\}$ , ieșirea este direct proporțională cu intrarea.
- **Sigmoid** –  $f(x) = \left\{ \frac{1}{1+e^{-x}} \right\}$ , numită și funcție logistică, ieșirea va fi între 0 și 1.
- **RELU** –  $f(x) = \{\max(0, x)\}$ , ieșirea este liniară și mai mare decât 0.

- **Softmax** –  $f(x) = \left\{ \frac{e^x}{\sum_k e^k} \right\}$ , calculează distribuția probabilităților, suma ieșirilor fiind 1.



Figură 2.1.2-1 O rețea neuronală simplă

### 2.1.3. Învățare supervizată

Învățarea supervizată este o ramură a învățării automate definită de utilizarea seturilor de date etichetate. Acestea sunt construite pentru a „supraveghea” algoritmi. Prin cunoașterea etichetelor modelul poate să își măsoare acuratețea și învața din timp.

Acest tip de învățare se împarte în două tipuri de probleme în cazul extragerii de date:

1. Clasificare
2. Regresie

Problemele de clasificare separă datele în categoriile specifice, pe baza etichetelor. Problemele de regresie sunt utile pentru a prezice valori numerice pe baza datelor de intrare. Cele mai populare metode de regresie sunt regresia liniară și polinomială.

### **2.1.4. Învățare nesupervizată**

Învățarea nesupervizată este ramura învățării automate definită de utilizarea seturilor de date neetichetate. Aceste modele grupează datele pe bază unor legături descoperite între date.

Aceste modele sunt utilizate pentru:

1. Clusterizare
2. Asocierea
3. Reducerea dimensionalității

Clusterizarea este o metodă de grupare a datelor pe baza diferențelor și asemănarilor dintre date.

Asocierea este o metoda folosită pentru a găsi relații între variabilele dintr-un set de date.

Reducerea dimensionalității micșorează numărul de attribute ale datelor fără a pierde integritatea acestora.

## **2.2. Noțiuni de învățare automată folosite**

### **2.2.1. Random Forest**

Random-Forest este un algoritm de învățare automată folosit frecvent în clasificare. Acesta este format din mai mulți arbori de decizie pentru a ajunge la un singur rezultat.

Un arbore de decizie este format din mai multe întrebări cu răspuns „DA” sau „NU”. Fiecare astfel de întrebare va face modelul să crească sub formatul unui arbore, urmând ca fiecare intrare să aparțină de una dintre nodurile acestuia.

Arborii de decizie sunt algoritmi de învățare automată, folosiți pentru clasificare.

### **2.2.2. Multinomial Naive Bayes**

Algoritmul Naive Bayes este o metodă de învățare probabilistică folosită în procesarea limbajului natural. Această metodă se bazează pe prezicerea de etichete

pentru un eșantion de date. Acest proces se realizează prin calcularea probabilității fiecărei etichete și returnarea etichetei cu cea mai mare valoare.

### **2.2.3. Long Short Term Memory Model (LSTM)**

Este un model ce se bazează pe stratul LSTM, un strat recurent. Diferența între LSTM și stratul recurent simplu este că stratul LSTM salvează informațiile pentru mai târziu, pentru a fi folosite sau respinse. Acesta utilizează celule de memorie pentru a reține informații actuale și anterioare, astfel permite modelului să înțeleagă și să țină cont de contextul temporal al informațiilor.

### **2.2.4. Convolutional Neural Network Model (CNN)**

Modelele convoluționale sunt folosite în principal pentru analiza și procesarea datelor sub formă de grilă, în principal a imaginilor sau semnalelor secvențiale, dar și a datelor sub formă de text, fiind considerate grile cu o dimensiune.

### **2.2.5. Bidirectional Encoder Representations from Transformers (BERT)**

BERT este o arhitectură de rețea neuronală dezvoltată de cercetătorii de la Google pentru prelucrarea limbajului. Acesta se bazează pe arhitectura Transformer, ce folosește straturi de atenție și straturi dense, este preantrenat pe un volum de date foarte mari într-un mod nesupervizat și folosește atât informații din trecut cât și viitorul unei propoziții. Rezultatele folosite de aceste modele sunt remarcabile în prelucrarea limbajului natural.

### **2.2.6. Robustly Optimized BERT Approach (RoBERTa)**

Este un model bazat pe BERT, dezvoltat de Facebook și a fost antrenat pe un set de date mai mare și pentru mai mult timp decât BERT, lucru ce a contribuit la îmbunătățirea performanței. Acest tip de model are rezultate remarcabile în sarcini de procesare a textului.

### **2.2.7. Framework-uri folosite**

#### **2.2.7.1. Tensorflow**

Este o bibliotecă software gratuită, open-source folosită în inteligența artificială, în special pentru rețelele neuronale.

#### **2.2.7.2. Keras**

Este o bibliotecă software, open-source folosită ca interfață pentru biblioteca TensorFlow.

### 2.2.7.3. Scikit-learn

Este o bibliotecă software gratuită, open-source folosită în inteligența artificială pentru mai multe tipuri de modele pre-implementate dar și pentru vizualizarea rezultatelor și a datelor.

### 2.2.8. Metrice de evaluare ale modelelor

Pentru verificarea funcționării a modelului folosim diverse metrice. Acestea se bazează pe matricea de confuzie.

		Clasa prezisă	
		P	N
Clasa actuală	P	TP	FN
	N	FP	TN

Figură 2.2.8-1 Exemplu matrice de confuzie

- **Acuratețea** – calculează numărul corect de predicții din numărul total de date.

$$\frac{TP + TN}{TP + FP + TN + FN}$$

( 1 )

- **Precizia** – Masoară proporția de date clasificate corect ca fiind pozitive din numărul total de preziceri.

$$\frac{TP}{TP + FP}$$

( 2 )

- **Regăsirea** – măsoară proporția de date clasificate corect ca fiind pozitive din totalul de date pozitive.

$$\frac{TP}{TP + FN}$$

( 3 )

- **F1-Score** – calculează scorul pe baza atât a preciziei cât și a regăsirii.

$$2 * \frac{\text{Precizie} * \text{Regăsire}}{\text{Precizie} + \text{Regăsire}}$$

( 4 )

- **Curba ROC – (Receiver Operating Characteristic)** este un instrument folosit în evaluarea modelelor. Aceasta este construită ca fiind Rata predicțiilor pozitive adevărate în funcție de rata predicțiilor negative. Cu cât aceasta descrie o arie mai mare, înseamnă că se aproprie de o clasificare perfectă.

Aceste metrice se calculează automat în procesul de testare și pot fi vizualizate utilizând matricea de confuzie, raportul de clasificare sau plotarea în funcție de epocă a valorilor de pierdere și de acuratețe. Fiecare astfel de metodă ne ajută să observăm capabilitățile modelului și îmbunătățirea acestuia în timp.

- **Supraînvățarea** – este o situație în procesul de învățare automată ce apare când modelul învață prea bine să realizeze clasificarea pentru datele de antrenare, dar în final acesta nu se comportă bine pe datele de test sau pe cele reale.
- **Subînvățarea** – este o situație în procesul de învățare automată ce apare atunci când modelul este prea simplu și nu reușește să capteze relațiile între date, în final acesta având rezultate proaste atât pe datele de antrenare cât și pe cele de test.

### 2.2.9. Tehnici de preprocesare a datelor

Deoarece datele în formatul lor brut nu pot fi folosite pentru antrenarea unui model în contextul NLP, trebuie aplicate mai multe tehnici de preprocesare pentru a pregăti datele în formatul adecvat. Printre acestea se numără tehnici de curățare a textului: eliminarea semnelor de punctuații, a URL – urilor; eliminarea cuvintelor de legătură ce nu aduc o îmbunătățire în procesul de clasificare; convertirea la litere mici a tuturor propozițiilor și se aplică tehnici de stemming și lematizare. Tehnica de stemming este procesul de eliminare a sufixelor și afixelor din cuvinte pentru a obține forma lor de bază. Tehnica de lematizare este procesul de reducere a cuvintelor la forma lor de bază, numită "lemă", prin luarea în considerare a contextului și a morfologie.

O altă tehnică de procesare a textului foarte importantă este tokenizarea ce constă în descompunerea textului în unități mai mici, cum ar fi grupuri de unu sau mai multe cuvinte.

Pentru ca datele să fie înțelese de mecanismele de învățare automata avem nevoie de tehnici de transformare a datelor într-un format numeric, cum ar fi codificarea one-hot sau încorporarea cuvintelor. Tehnica de bază este crearea unui dicționar unde fiecăruia grup de cuvinte/cuvânt îi este atribuit un număr.

Pentru codarea one-hot, fiecare cuvânt dintr-o listă de cuvinte va primi o valoare unică (mai multe coloane) formată din „0” și o singură valoare de „1”. Ca de exemplu în lista [ mere, pere, portocale ] valorile codate one-hot vor fi mere = [1,0,0], pere = [0,1,0] și portocale = [0,0,1].

O altă tehnică folosită pentru codarea cuvintelor este TF-IDF, formată din TF și IDF. TF provine de la frecvența termenilor într-un document. Iar IDF de la inversul frecvenței în corpul de documente. Cuvintelor din document li se atribuie scoruri în funcție de valoarea tf-idf calculată.

$$TF(t, d) = \frac{\text{numărul de cuvinte } t \text{ din document}}{\text{numărul total de cuvinte din document}}$$

( 5 )

$$IDF(t, d) = \ln \frac{N}{\text{numărul de apariții al cuvântului } t \text{ în corpul de documente} + 1}$$

( 6 )

$$TF - IDF = TF * IDF$$

( 7 )

GloVe este un algoritm de învățare nesupervizat ce extrage sensul propozițiilor cuvânt cu cuvânt și observă legături între acestea. Acestor cuvinte li se atribuie valori numerice păstrând relații semantice și sintactice.

TF-IDF și GloVe sunt astfel tehnici de procesare a cuvintelor ce realizează o transformare dintr-o valoare semantică în una numerică.

## 2.3. Noțiuni teoretice în dezvoltarea mobilă

### 2.3.1. React-Native

React-native este un framework open-source, cross-platform folosit pentru dezvoltarea de aplicații Android și iOS utilizând limbajul JavaScript. Acesta este bazat pe framework-ul React folosit pentru dezvoltarea web.

Dezvoltarea este rapidă și accesibilă datorită limbajului folosit și utilizării conceptului de componente reutilizabile.



Compilarea se face în cod nativ pentru fiecare platformă, astfel se oferă o performanță similară cu aplicațiile dezvoltate în totalitate pentru o anumită platformă, dar cu avantajul de scriere a unui singur cod.

Framework-ul este în continuă dezvoltare, astfel se facilitează dezvoltarea de aplicații complexe dar și extinderea rapidă a celor deja existente.

NECLASIFICAT

Pagină albă

### 3. Studii anterioare și cercetări relevante

Studiul condițiilor mentale este un subiect de interes pentru cercetători. Deoarece învățarea automată este din ce în ce mai folosită în viața de zi cu zi, aceasta a pus un accent și în detectarea condițiilor mentale, în special cu ajutorul rețelelor de socializare.

Astfel, sursa de inspirație pentru această lucrare constă în mai multe articole relevante pentru domeniul de studiu. Pentru fiecare dintre acesta s-au folosit date de în special de pe Reddit din cauza posibilității utilizării acestora liber în diverse studii pe mai multe domenii dar și de pe Twitter pentru studiile de început.

Inteligența automată în cadrul sănătății mentale este un subiect ce a fost dezvoltat și cercetat începând cu câteva decenii în urmă prin mai multe metode, după cum este specificat în articolul *Mental Health Predictions Using Machine Learning: Taxonomy, Applications, and Challenges* [8]. De-a lungul anilor s-au folosit mai multe metode de colectare a datelor, pornind de la interviuri audio sau scrise [11], [13], [7]. Bolile pentru care s-au realizat aceste articole au fost în general despre anxietate și depresie, schizofrenie, PTSD sau bipolaritate. S-au folosit metode precum Random Forest, Multinomial Naive Bayes, regresia logistică, cei mai apropiați K vecini, clasificarea Gaussiană, învățarea profundă sau modele neuronale. În această lucrare s-au ales câteva variante ce vor fi studiate, ca și Random Forest, Multinomial Naive Bayes și construirea de modele neuronale ce fac parte din învățarea profundă.

În ultimii ani, datorită activității online foarte mare și a utilizatorilor ce folosesc acest loc ca un spațiu sigur de a își scrie gândurile. Astfel, Twitter și în momentul actual Reddit sunt 2 rețele de socializare cu un corp imens de date, ce pun la îndemâna cercetătorilor foarte multe date, fără a avea access la datele utilizatorilor.

În cadrul articolului *Characterisation of mental health conditions in social media using Informed Deep Learning* [7] Reddit a fost platforma de pe care au fost preluate datele, pentru acesta s-a folosit un corp cu 12 etichete, incluzând-o pe cea fără legătură cu afecțiunile mentale. Datele au fost neechilibrate și s-au folosit două modele convoluționale, unul binar ce ar clasifica datele ca suferind de o afecțiune mentală sau nu. Al doilea model va clasifica datele în restul de 11 clase rămase. Astfel în această cercetare s-au folosit tot 2 modele, unul binar recurent folosind long short term memory și unul multi-clasă, pentru care s-au utilizat mai multe metode.

SMDH [9] este o bază de date cu un conținut de șase condiții mentale; utilizatorii sunt autodiagnosticați. Aceasta este tot o bază de date neechilibrată la care s-au folosit mai mulți algoritmi, cum ar fi un model convoluțional sau regresia logistică. Cele mai bune scoruri le-a avut modelul neuronal convoluțional, aspect ce a încurajat folosirea unei astfel de metode și în acest studiu. Din păcate, această bază de date nu este publică și nu a putut fi folosită.

În articolul *Sentiment Analysis of Public Social Media as a Tool for Health-Related Topics* [6] s-au descris mai multe nivele de înțelegere a sentimentelor pe baza textului. Acesta sunt nivelul pe document, pe propoziție pe aspect sau pe concept. Cele mai multe astfel de studii, asupra sentimentelor dintr-un text este utilizat la nivel de document sau de propoziție. Studiul urmărește și descrie o procedură standard de a extrage și evalua propoziții pe baza sentimentelor. Aceste proceduri se bazează pe colectarea datelor, identificarea opiniilor, extragerea necesarului, clasificarea opiniilor, rezumatul opiniilor și evaluarea performanței. Pentru fiecare dintre aceste proceduri sunt utilizate mai multe tehnici. Datele pot fi colectate din API-ul de la Reddit, identificarea opiniilor se bazează pe filtre la nivelul limbii, autorului sau timpului. Extragerea necesarului se referă la curățarea datelor înainte de utilizare. Clasificarea opiniilor sunt mecanismele de învățare automată. Iar rezumatul opiniilor și evaluarea performanțelor se realizează asupra metodelor folosite și a acestor rezultate.

Există multe metode de clasificare a datelor; în articolul *A deep learning model for detecting mental illness from user content on social media* [4], pentru o bază de date cu 6 clase, s-au folosit 6 clasificatoare binare ce apoi au fost combinate. O astfel de metodă poate să aducă un rezultat bun dacă datele sunt într-un raport bun una față de alta, clasificatorii binari rezultând în rezultate mai bune.

Din cauza interesului din ce în ce mai mare asupra sănătății mentale, în articolul *MentalBERT: Publicly Available Pretrained Language Models for Mental Healthcare* [10] s-a urmărit construirea unui model preantrenat pe bază transformerilor BERT și RoBERTa. Acesta a fost ajustat pe date de pe Reddit din spectrul mental, cum ar fi depresia, anxietatea, stresul sau bipolaritatea. Aceste tipuri de modele au avut rezultate mai bune în majoritatea clasificărilor față de utilizarea simplă a lui BERT sau RoBERTa. Aceste modele sunt publice pe *HuggingFace*<sup>3</sup> pentru viitoarele cercetări.

Articolul *Classification of mental illnesses on social media using RoBERTa* [3] folosește date colectate prin API-ul de la Reddit, de unde s-au ales cinci etichete. Modelele folosite în acesta au fost un model bazat pe bază de LSTM, unul

---

<sup>3</sup> <https://huggingface.co/>

BERT și unul RoBERTa. Clasificarea s-a efectuat asupra titlului postării, textului postării dar și asupra la ambele. Cele mai bune rezultate au fost pe cele combinate folosind RoBERTa. Pentru titlu s-a folosit un număr de 35 de cuvinte iar pentru textul postărilor un număr de 512 cuvinte, mult mai mult față de rezultatele noastre, unde am folosit mai 70 sau 300 de cuvinte pentru LSTM și 64 sau 100 pentru BERT.

Pe lângă acestea, s-au efectuat studii asupra datelor de pe Twitter sau Reddit în identificarea depresiei, a riscului de suicid, a stresului. [11], [12].

Un studiu pe baza căreia s-a folosit una din bazele de date din această cercetare este *Natural Language Processing Reveals Vulnerable Mental Health Support Groups and Heightened Health Anxiety on Reddit During COVID-19* [2] care pune accentul pe studiul afecțiunilor mentale în raport cu pandemia recentă și cum a fost influențată populația.

## 4. Pregătirea mediului de lucru

Mediul de lucru și pregătirea acestuia a constat în dezvoltarea unor jurnale în jupyter notebook folosind python pentru fiecare model folosit dar și pentru crearea bazei de date și vizualizarea acesteia.

Rularea acestora s-a făcut atât pe patru GPU – uri din gama Tesla K40m cât și pe CPU.

Modelele finale au fost salvate și încărcate în cadrul unui server python. Atât serverul cât și clientul au fost create folosind IDE-ul Visual Studio Code.

## 5. Dezvoltarea proiectului

Implementarea proiectului poate fi vizualizată și pe repositoryul de lucru<sup>4</sup>.

### 5.1. Analiza, descrierea și procesare bazelor de date folosite

Bazele de date folosite în această studiu sunt construite pe baza API-ului de la Reddit, etichetele fiind formate din subreddit-ul aferent fiecărei postări, acestea făcând parte din grupuri de suport și autodiagnosticare sau grupuri din diverse domenii.

Pentru această lucrare s-au găsit și analizat mai multe baze de date pentru detectarea afecțiunilor psihice din text: DAIC-WOZ [13], SMDH [9], Kaggle [1], Zenodo [2] dintre care au fost folosite la rulare cele mai relevante două: [1] și [2]. În cadrul modelelor de referință și a modelului LSTM s-au folosit ambele baze de date, pentru a crea o comparație a rezultatelor folosind același model cu diverse baze de date. În cazul celorlalte modele, CNN, BERT, RoBERTa s-a utilizat a doua bază de date [2], din cauza diversității domeniilor din care fac parte postărilor, creând posibilitatea de clasificare atât binară cât și multi-clasă, dar și datorită aplicabilității mai bune în aplicația finală.

În continuare se va face o analiză a celor două baze de date folosite, a metodelor de preprocesare și de construire ale acestora.

#### 1. Mental Disorders Identification (Reddit)

Prima bază de date [1] folosită la rulare a constatat în postări de pe Reddit neprocesate cu următoarele atribute:

- title
- selftext
- created\_utc
- over\_18
- subreddit

Din aceasta am păstrat doar coloanele care ne-au interesat, adică „selftext”, ce conține corpul postărilor și „subreddit”, ce conține clasa aferentă acestora.

S-a observat o distribuție neuniformă a bazei de date cu un număr total de 701787 rânduri înainte de preprocesare. S-au aplicat apoi tehnici de curățare și procesare a datelor:

---

<sup>4</sup> <https://github.com/Aleexuta/Mentol.git>

- S-au eliminat rândurile ce nu conțin postări valide
- Convertirea la litere mici
- Eliminarea URL-urilor din corpul postării
- Eliminarea semnelor de punctuație
- Eliminarea cuvintelor de legătură din dicționarul englez
- Tehnici de „steaming”
- Tehnici de lemmatizare

După aceste eliminări au rămas un număr de 580421 rânduri distribuite în șase clase:

BPD : 212615

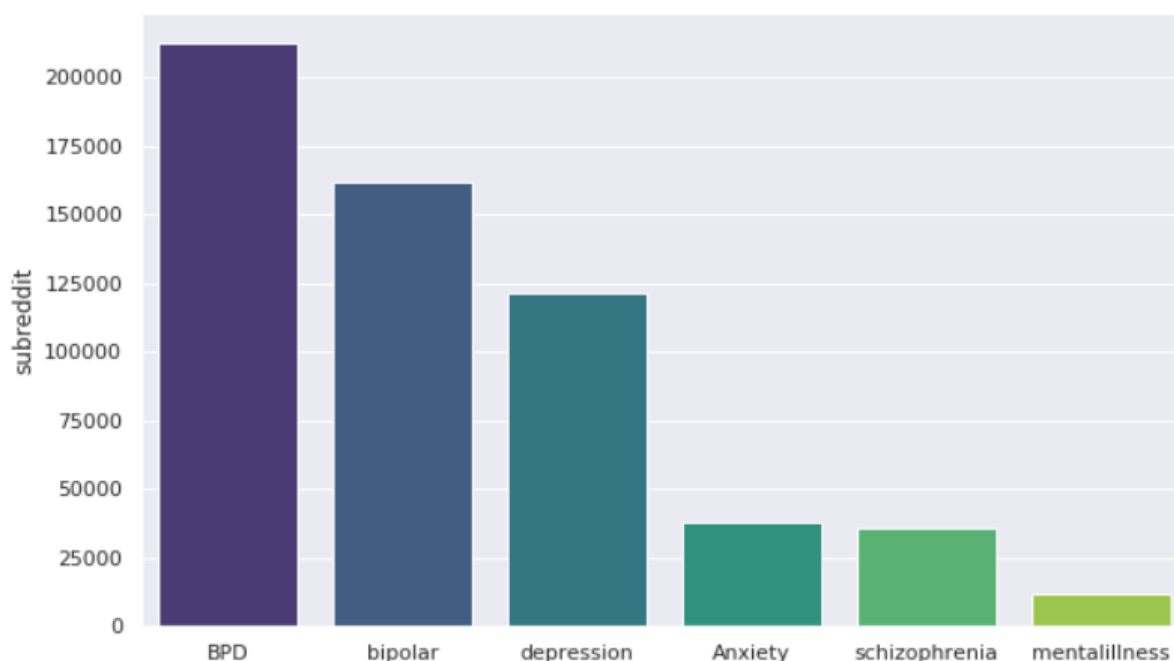
Anxiety : 161465

Depression : 121174

Mentalillness : 38021

Bipolar : 35535

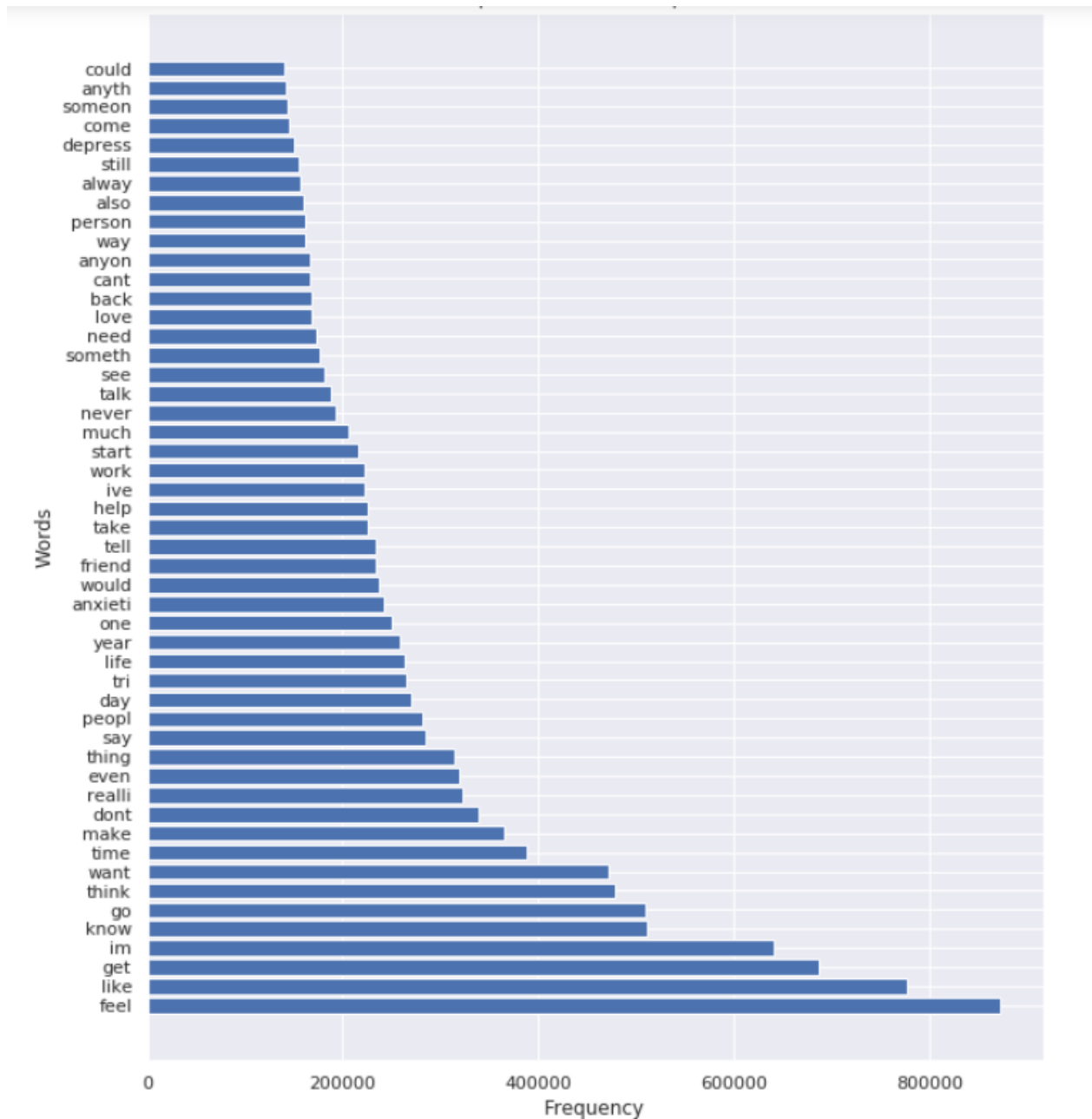
Schizophrenia : 11611



Figură 2.3.1-1 Distribuția pe clase a bazei de date

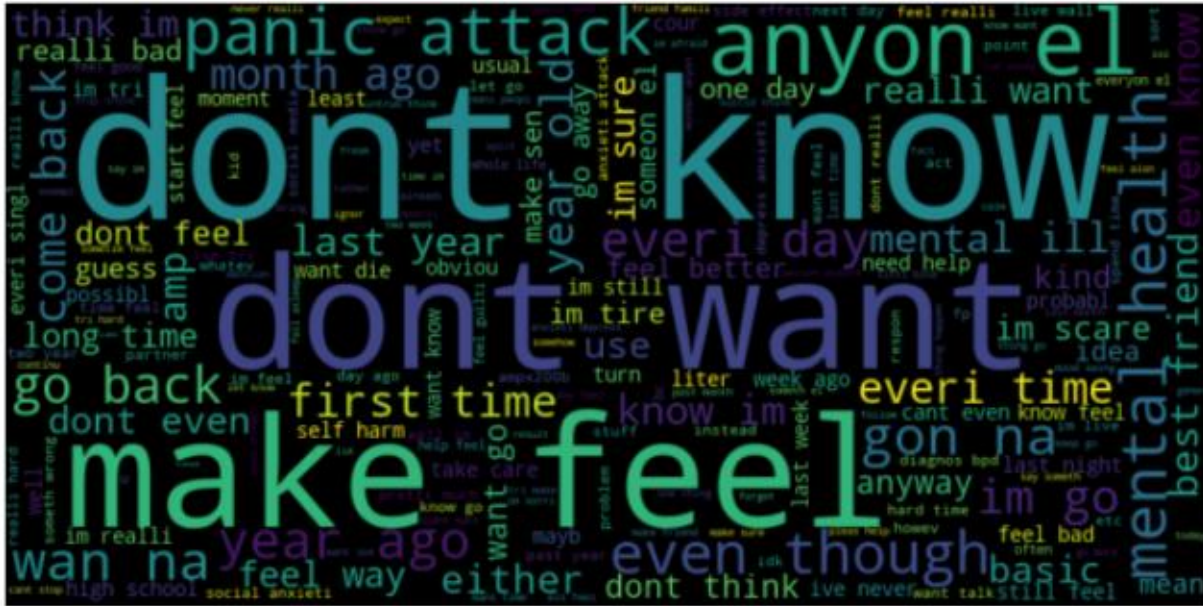
Se observă cum există mai câteva cuvinte foarte folosite, cum ar fi „feel” cu peste 800000 de apariții, „like”, „get”, și apoi o scădere uniformă pentru primele 50 de cuvinte.





Figură 2.3.1-2 Distribuția cuvintelor în baza de date

O altă vizualizare a datelor din baza de date este norul de cuvinte ce conține cele mai frecvente grupuri de cuvinte, afișându-le pe cele mai folosite cu fontul cel mai mare.



Figură 2.3.1-3 Nor de cuvinte

## 2. Reddit Mental Health Dataset

A doua bază de date [2] este construită pe baza unui studiu legat de influența asupra sănătății mentale a pandemiei din 2019.

Aceasta s-a manevrat într-un mod folositor studiului actual și s-au construit două subset-uri ce au fost folosite pentru două tipuri de clasificare, binară și multclasă.

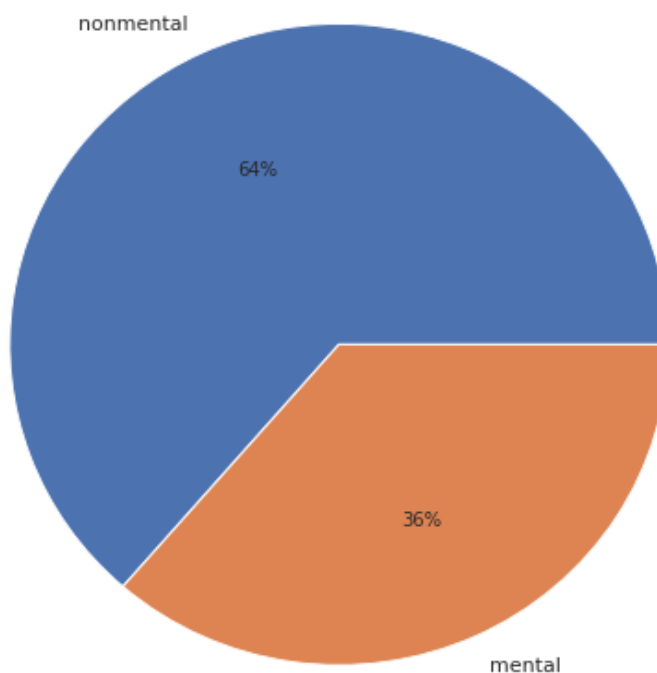
Aceasta este formată din mai multe etichete din spectrul sănătății mentale, dar și din domeniul de zi cu zi sau chiar la limita acestora. Acestea au fost alese pe baza mai multor intervale orare, ianuarie – aprilie 2020, ianuarie – aprilie 2019, ianuarie – aprilie 2018 și decembrie 2018 – decembrie 2019.

În cazul acestui studiu, subset-urile au fost formate fără a ține cont de intervalele orare, ci doar de etichetele postărilor.

Astfel, am descărcat fișierele necesare și am creat două subset-uri. Unul care conține 2 clase, prima cu toate datele având o referință la o boală mentală, și una cu datele fără legătură cu acest domeniu, cele de la limita lor nu au fost folosite. Al doilea subset are doar etichetele cu referință la o boală mintală, grupate în funcție de afecțiunea psihică.

Peste acestea am aplicat aceleași operații de curățare și procesare ca și în prima bază de date pentru a fi siguri de eliminarea oricărui zgomot, dar și o operație de corectare<sup>5</sup> a limbajului folosit din cauza greșelilor întâlnite.

Există un număr de 634486 de postări în spectrul non-mental și un număr de 362912 de postări în spectrul mental.

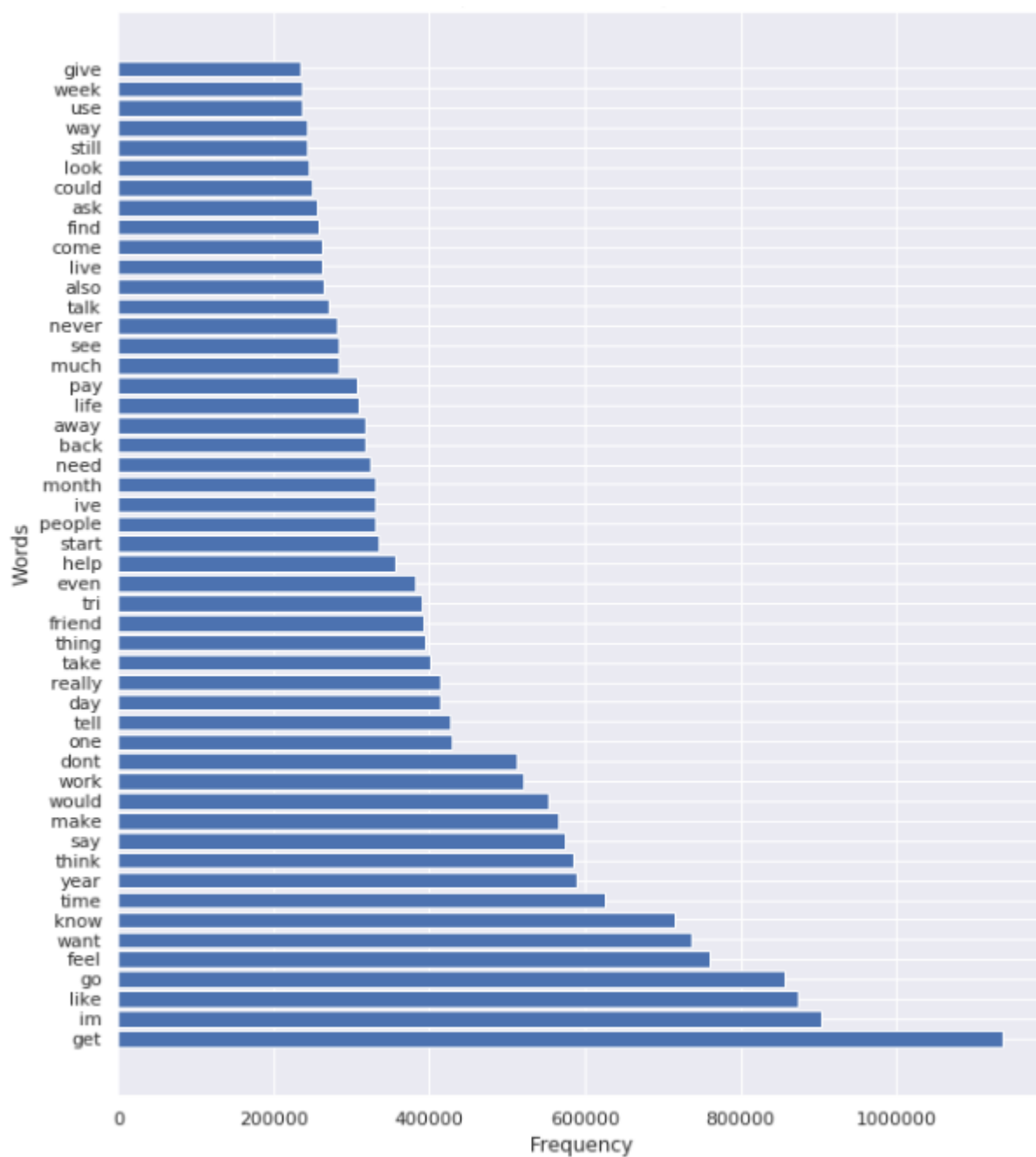


Figură 2.3.1-4 Distribuție baza de date binară

Se observă că în postări apare de 1000000 de ori cuvântul „get”, urmat de „im” și „like”.

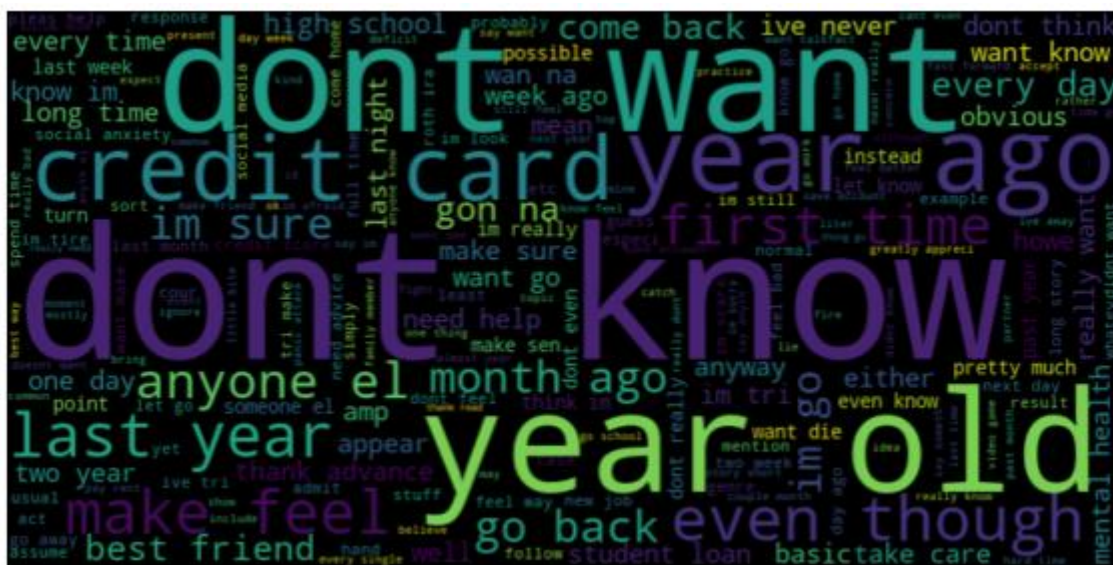
---

<sup>5</sup> <https://github.com/filyp/autocorrect>



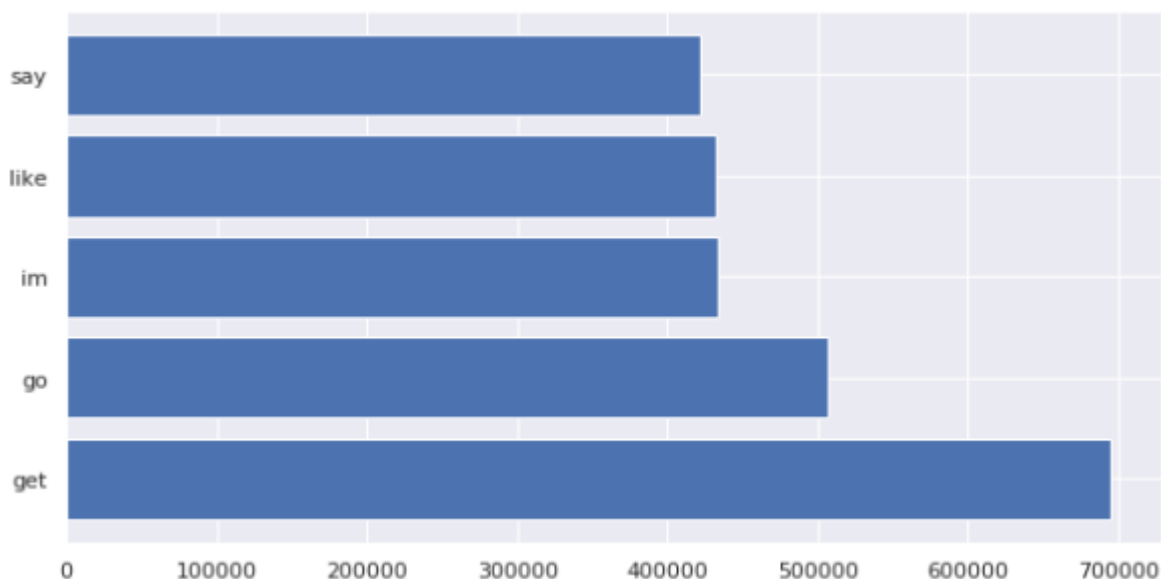
Figură 2.3.1-5 Frecvența cuvintelor în baza de date binară

Există asemănări în cazul grupurilor de cuvinte găsite în analiza ambelor baze de date, un mare procent repetându-se în ambele cazuri. De aici putem trage concluzia de integritate a sursei datelor.

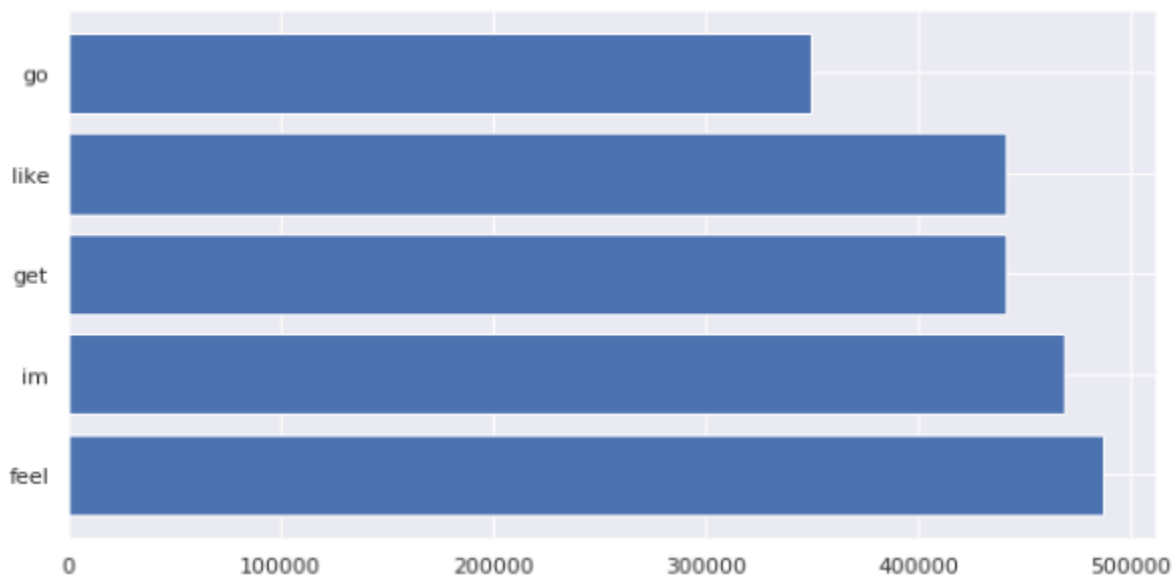


Figură 2.3.1-6 Nor de cuvinte

Se observă că cele mai folosite cuvinte diferă pentru cele două clase. Astfel pentru clasa nonmentală cel mai folosit cuvânt este „get”, iar pentru cea mentală este „feel”.



*Figură 2.3.1-7 Frecvență cuvinte clasa nonmentală*

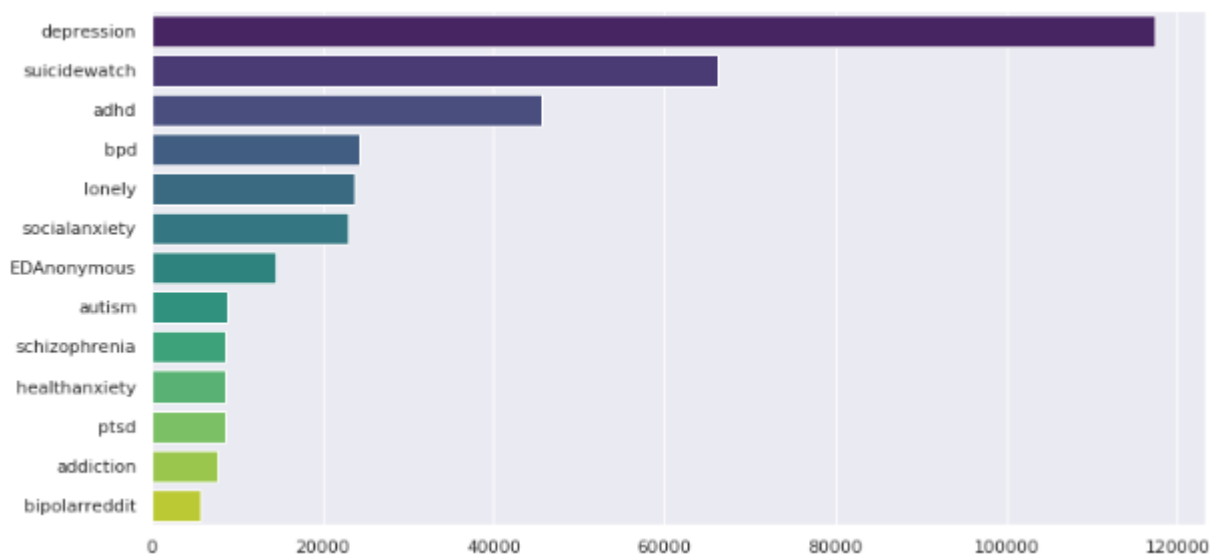


Figură 2.3.1-8 Frecvență cuvinte clasa mentală

Al doilea subset ce conține doar etichete mentale are o distribuție neuniformă și conține 13 (treisprezece) clase, fiecare cu numărul său de instanțe, astfel:

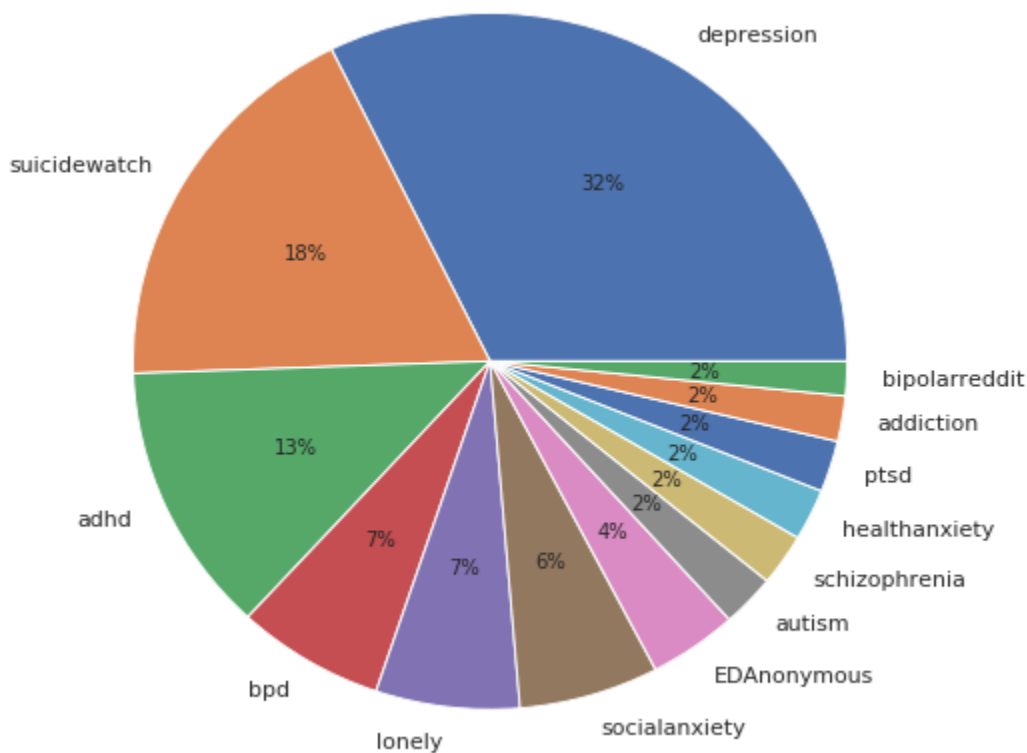
- Depression 117331
- Suicidewatch 66161
- Adhd 45631
- Bpd 24294
- Lonely 23635
- Socialanxiety 22996
- EDAnonymous 14577
- Autism 8869
- Schizophrenia 8712
- Healthanxiety 8648
- Ptsd 8643
- Addiction 7641
- Bipolarreddit 5780

Se observă cum eticheta „depression” are de aproape două ori mai multe intrări decât următoarea clasă, și ca numărul acestora scade exponențial. Acest lucru o să ne pună probleme în etapa clasificării.



Figură 2.3.1-9 Distribuția pe clase a bazei de date

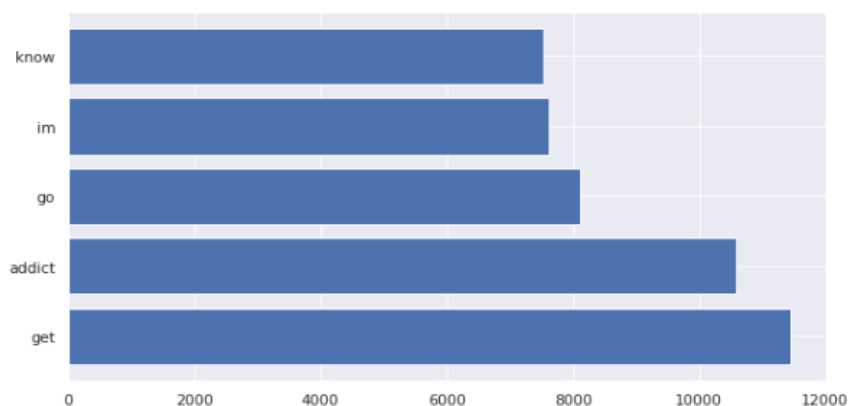
O bună vizualizare a diferențelor dintre numărul de clase se vede și în graficul de tip plăcintă.



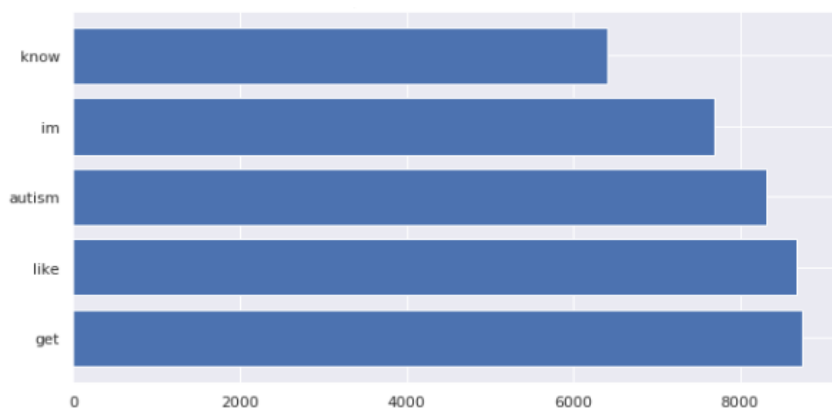
Figură 2.3.1-10 Distribuția claselor în baza de date – plăcintă

La o analiză mai amănunțită a celor mai utilizate cuvinte în cazul fiecărei clase se observă ceva interesant.

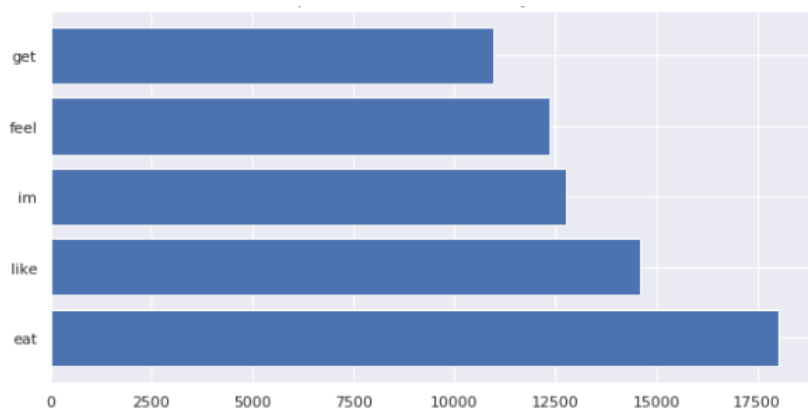
## NECLASIFICAT



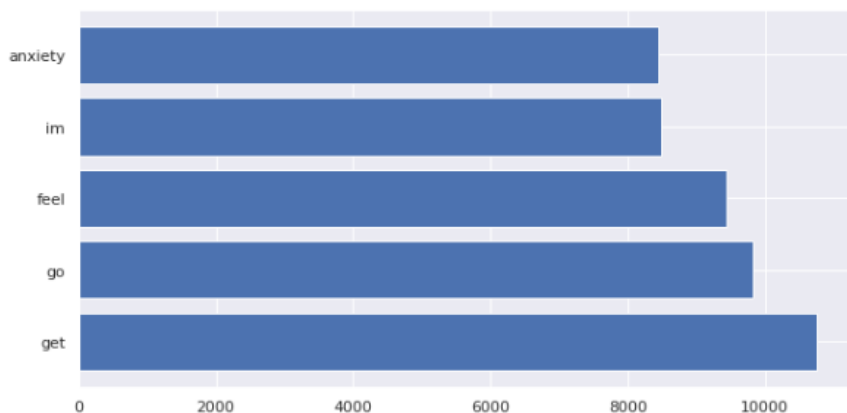
Pentru clasa „addiction”, unul din primele 5 cuvinte cele mai folosite este „addict” cu peste 10000 de apariții.



Pentru clasa „autism” al treilea cel mai utilizat cuvânt este „autism” cu puțin peste 8000 de apariții.



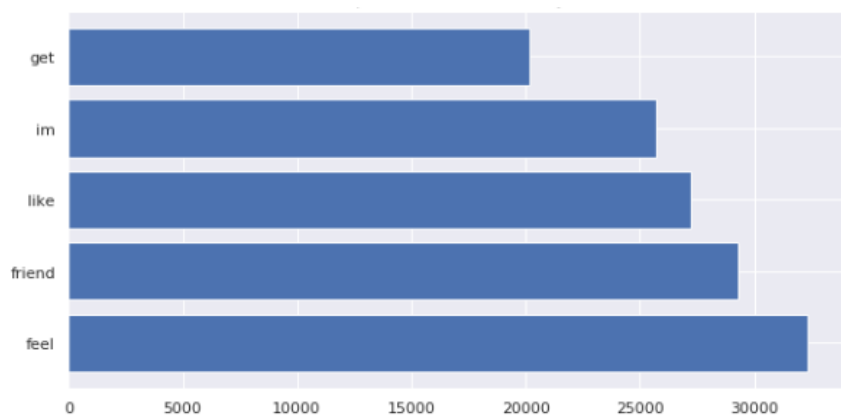
Clasa „EDAnonymous” are cel mai folosit cuvânt „eat” cu peste 17000 de apariții.



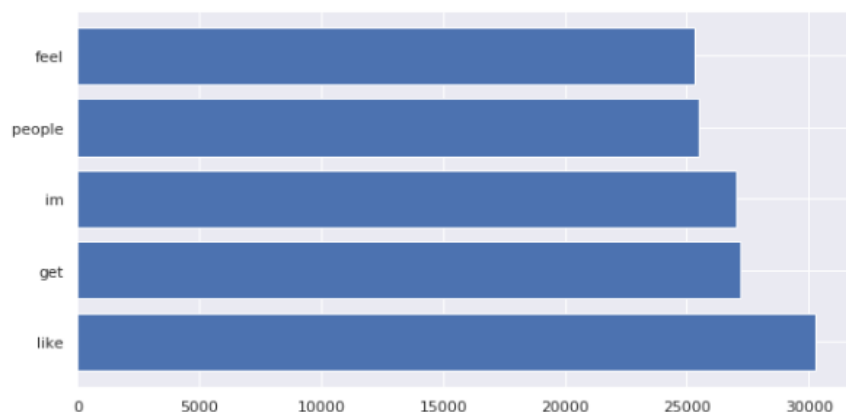
În clasa „healthanxiety”, unul din cuvintele cele mai folosite este „anxiety”.

## NECLASIFICAT





Clasa „lonely”  
folosește cu peste 25000  
intrări cuvântul „friend”.



Clasa „socialanxiety”  
vine tot cu un număr de  
25000 de intrări pentru  
cuvântul „people”.

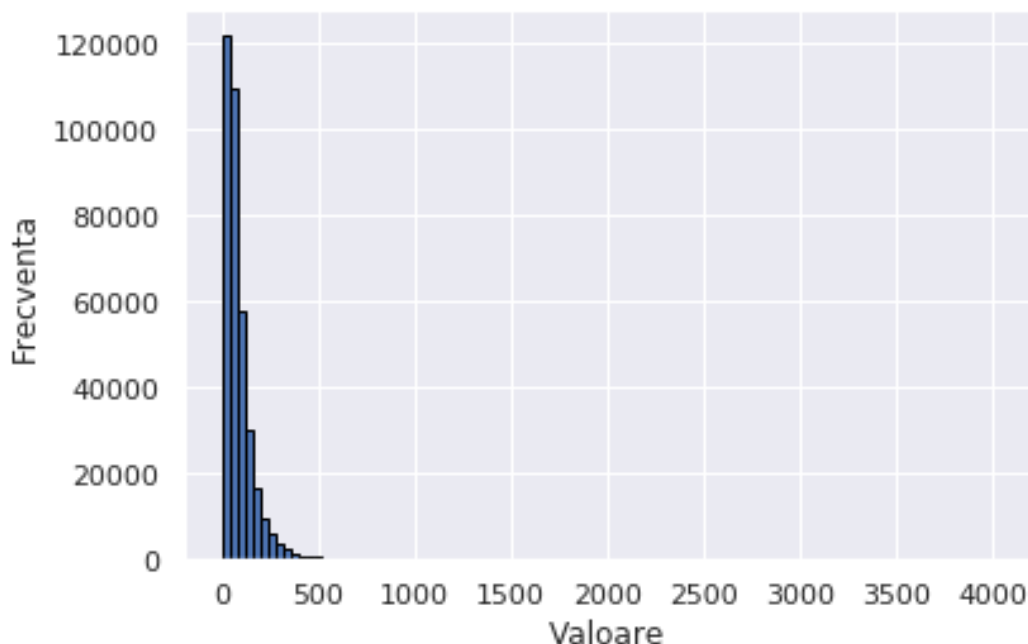
În aceste clase se observă cum unele din cele mai folosite cuvinte au o legătură directă cu boala de care aparțin, lucru ce ajută la asocierea postărilor cu eticheta corespunzătoare.

După această analiză amănunțită și înțelegere a datelor din baza de date, se poate trece la procesarea înainte de antrenare.

În fiecare caz, baza de date a fost împărțită în mai subset-ul de antrenare, validare și test astfel. Din totalul de rânduri s-au folosit 30% pentru test, iar restul de 70% s-a împărțit în 20% pentru validare și 80% pentru antrenare. Raportul final al bazei de date este

Antrenare	→	56%
Validare	→	14%
Testare	→	30%

Avem o dispersie medie a cuvintelor în postări / de aproximativ 83 de cuvinte pe rand.



Figură 2.3.1-II Frecvența cuvintelor în propoziții

Interval 0.0 - 400.1: 359124

Interval 400.1 - 800.2: 3460

Interval 800.2 - 1200.3: 278

Interval 1200.3 - 1600.4: 37

Interval 1600.4 - 2000.5: 1

Interval 2000.5 - 2400.6: 9

Interval 2400.6 - 2800.7: 0

Interval 2800.7 - 3200.8: 0

Interval 3200.8 - 3600.9: 1

Interval 3600.9 - 4001.0: 2

De aici putem înțelege ca există 359124 intrări ce au un număr între 0-400 de cuvinte, lucru ce ne va ajuta să hotărâm numărul de cuvinte folosit în procesul de tokenizare.

## 5.2. Pregătirea datelor

### 5.2.1. Tipuri de transformare de la propoziții la valori numerice.

Am pregătit datele în mai mai multe moduri pentru a putea observa care este varianta cea mai bună în cazul clasificatorului nostru.

## I. Tokenizare

Am creat un „tokenizer” folosind API-ul `keras` `tf.keras.preprocessing.text.Tokenizer`, antrenat pe vocabularul nostru folosind metoda `fit_on_texts()` și am transformat subset-urile în propoziții folosind metoda `sequences_to_texts()`. Am adăugat o umplere sau am făcut o trunchiere, în funcție de caz, folosind metoda `pad_sequences` din modulul `tf.keras.preprocessing.sequence` pentru a ajunge la aceeași dimensiune la toate propozițiile. Acești vectori de valori numerice au fost apoi folosite în procesul de antrenare.

## II. Vectorizare TFIDF

Am creat un „vectorizer” folosind din biblioteca `scikit-learn` clasa `sklearn.feature_extraction.text.TfidfVectorizer`. Am aplicat peste datele de antrenare metoda `fit_transform()` a noii instanțe pentru învățarea vocabularului și transformării acestuia în formă matriceală. Peste datele de test am aplicat metoda `transform()` pentru transformarea datelor în forma matriceală în funcție de vocabularul „vectorizer-ului”. Aceste date au fost apoi folosite în procesul de antrenare.

## III. Tokenizare și Încorporare GloVe

Am aplicat aceleași operați ca în cazul tokenizării simple. Am salvat dicționarul cu frecvențele cuvintelor folosit în procesul de tokenizare, ce este salvat în parametrul `word_index` al clasei `Tokenizer` folosite. Am folosit pentru formarea matricei de încorporare fișierul de la GloVe `glove.6B.300d.txt` de la care am luat valoarea fiecărui cuvânt precalculată pe care am adăugat-o în dicționarul nostru pe indicii necesari.

După tokenizare `word_index` este de forma: <sup>6</sup>

```
{'deep': 1, 'do': 5, 'i': 3, 'learning': 2, 'like': 7, 'love': 4, 'you': 6}
```

Figură 5.2.1-1 Exemplu `word_index`

Apoi, pentru fiecare cuvânt de aici i se va accesa vectorul asociat din fișierul `glove` ce va fi adăgat în matricea finală de încorporări folosită în stratul „Embedding”.

---

<sup>6</sup><https://medium.com/analytics-vidhya/understanding-nlp-keras-tokenizer-class-arguments-with-example-551c100f0cbd>

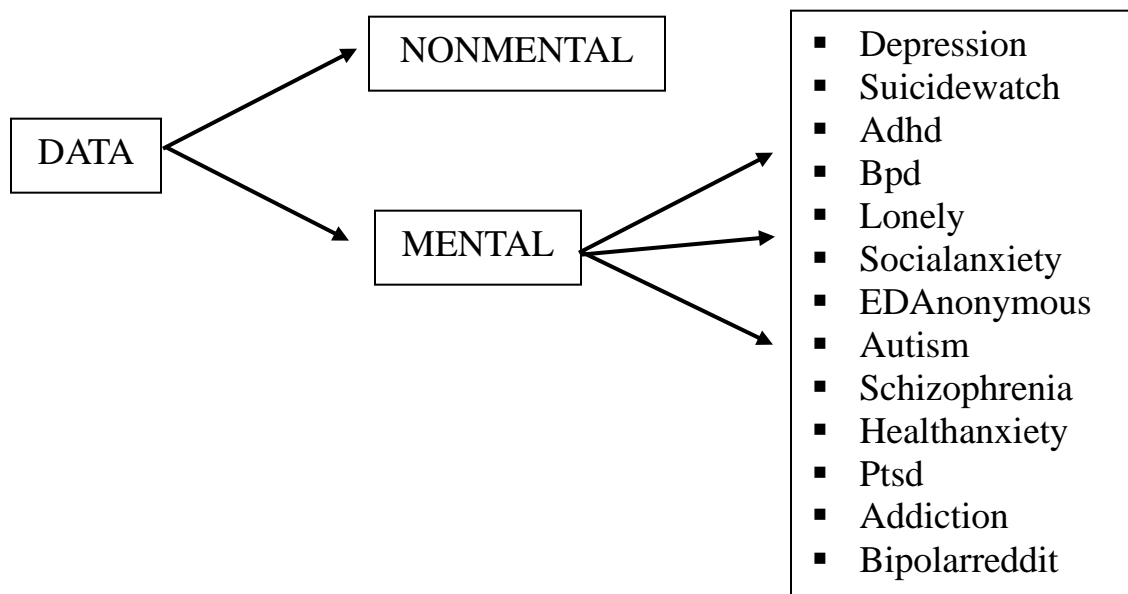
Cea mai recomandată (citate) și folosită variantă în acest studiu a fost încorporarea cu GloVe, din cauză asocierilor făcute între cuvinte și clasificarea mai bună a datelor.

### **5.2.2. Generator de date**

Din cauza volumului mare de date folosit în procesul de antrenare și imposibilității încărcării tuturor acestor informații în timpul antrenării, am creat un generator de date ce va primi pe rând informațiile ca apoi să le folosească. Acest generator a fost construit sub formă de metodă ce returnează porțiuni de date de o anumită mărime (32, 64, 128 de rânduri). În acest generator s-au putut aplica diverse operații pe date, cum ar fi augmentarea.

## 5.3. Algoritmi folosiți pentru detectarea afecțiunilor psihice

Am folosit mai mulți algoritmi și modele cu diverși parametrii pentru a ajunge la cea mai bună variantă a clasificatorului. După cum am specificat la analiza bazelor de date, cele două subseturi se vor folosi pentru antrenarea a două tipuri de modele, unul binar și unul multiclasă, ce vor fi folosite pentru detecția și clasificarea afecțiunilor psihice din text.



Figură 5.2.2-1 Arhitectură clasificare

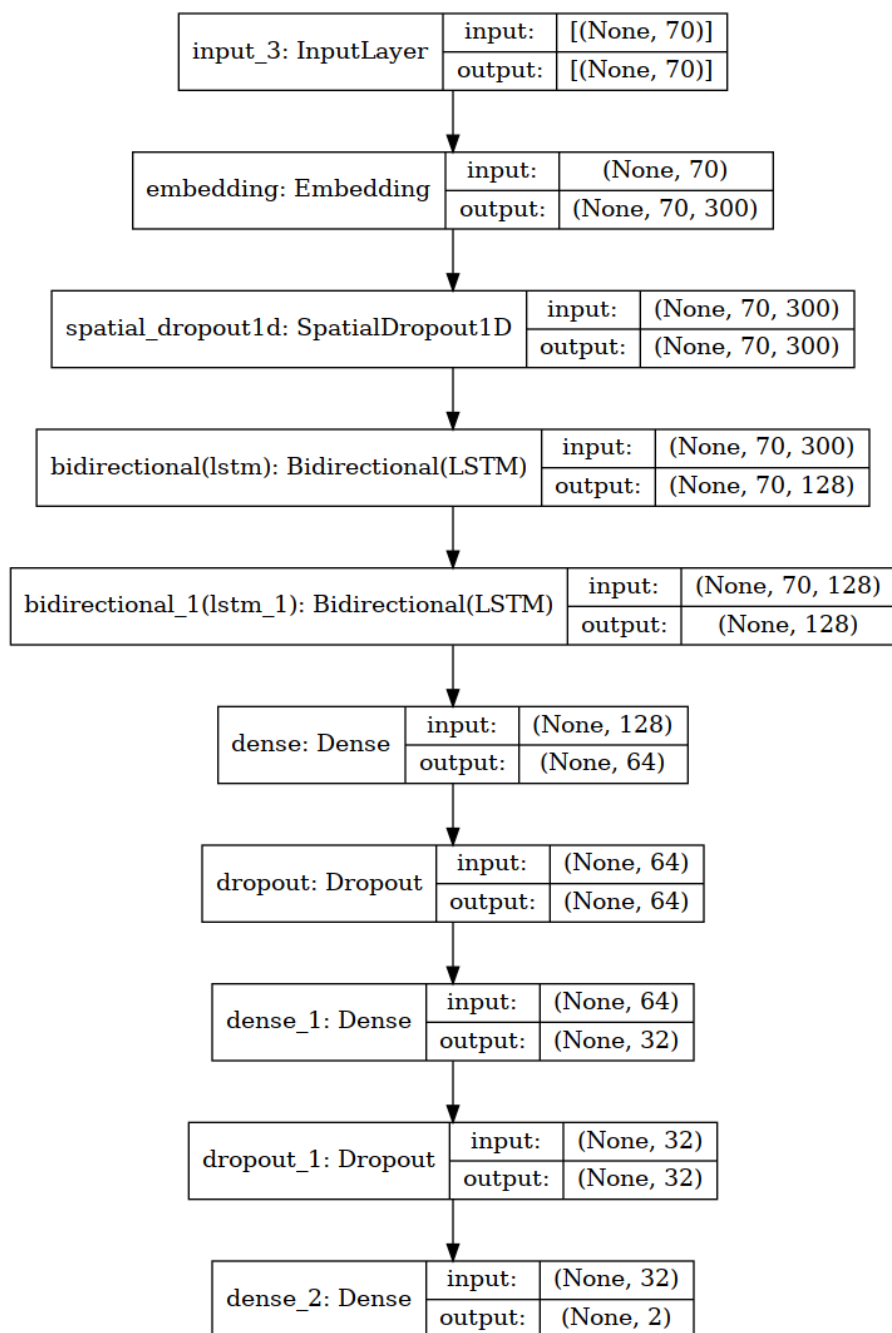
Am aplicat tehnici de tokenizare și vectorizare (TF-IDF sau GloVe) peste care am adăugat o umplutură sau o trunchiere a propozițiilor către dimensiunea aleasă.

### 5.3.1. Long Short Term Memory (LSTM) – clasificator binar

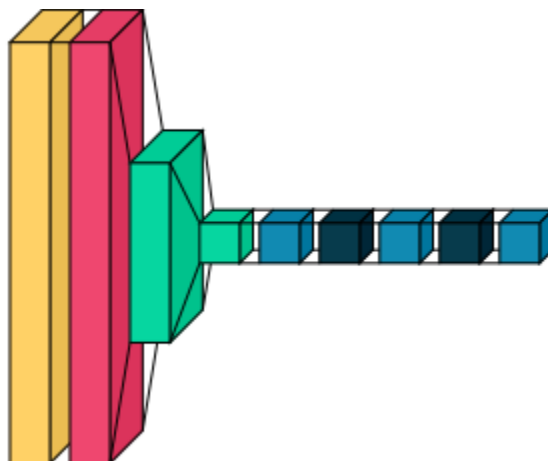
Pentru clasificarea binară am folosit un model neuronal bazat pe Long Short Term Memory. Am folosit un model secvențial ce primește datele în format de text, mai apoi preprocesate în vectori de valori numerice pe care le clasifică în două clase.

După cum se observă începem arhitectura modelului cu un strat de intrare de dimensiunea 70, un strat de încorporare, un strat de SpatialDropout ce ajută la păstrarea conectivității între caracteristici, două straturi de LSTM Bidirecțional, ce

procesează secvențele de intrare în atât anterior și posterior momentului actual de timp și mai multe straturi dense și de dropout. Avem aici un număr total de 81,929,398 parametrii, dintre care 296,098 de parametrii antrenabili și 81,633,300 neantrenabili.



Figură 5.3.1-1 Arhitectură model binar

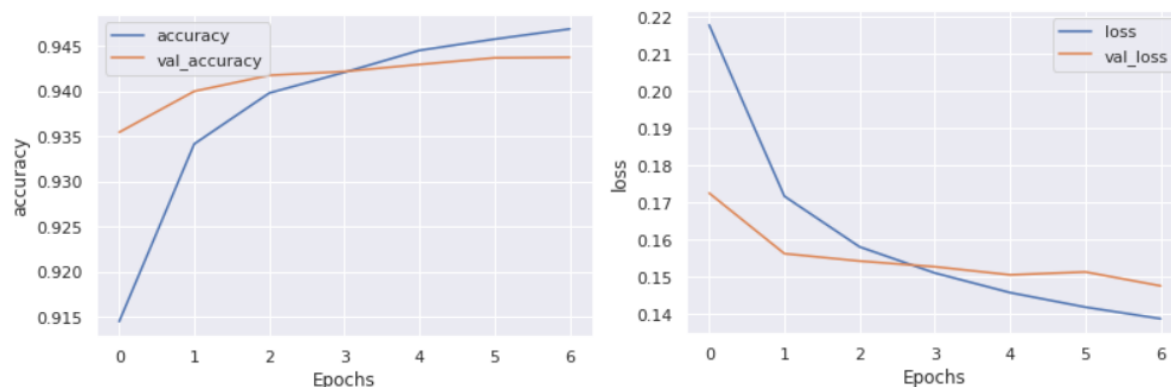


Figură 5.3.1-2 Arhitectură model binar

Pentru acesta, s-a folosit o lungime a propozițiilor de 70 cuvinte și lungimea încorporărilor de 300. S-a folosit optimizatorul Adam cu o rata de învățare de 0.001 și tokenizarea datelor împreună cu încorporarea GloVe.

Pentru a observa cât de mult ajută corectarea textului pe baza dicționarului englez s-a antrenat și testat modelul înainte și după această operație de procesare a textului.

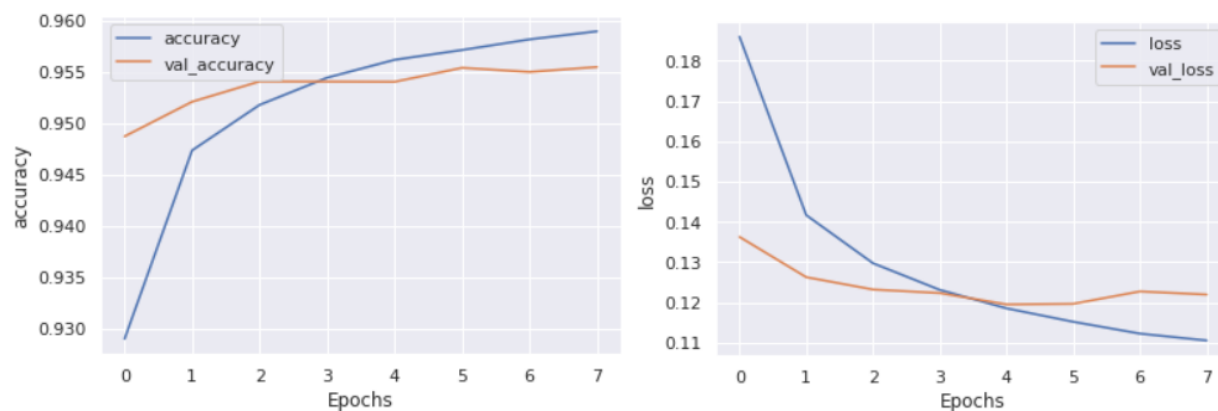
După cum s-a observat la analiza bazelor de date, raportul dintre clasa mentală și cea non-mentală este de aproximativ 1:2, lucru ce nu ne-a încurcat la obținerea unui scor bun.



Figură 5.3.1-3 Curbe de învățare înainte de auto-corectare

Se observă ca la procesul de antrenare acuratețea crește până la valoarea de 0.94-0.945 unde rămâne constantă și se oprește automat după 7 epoci. Funcția de pierdere scade până la valoarea de 0.14 înainte de auto-corectarea textului. După aceasta, acuratețea crește până la 0.955-0.96 unde rămâne constantă după 8 epoci,

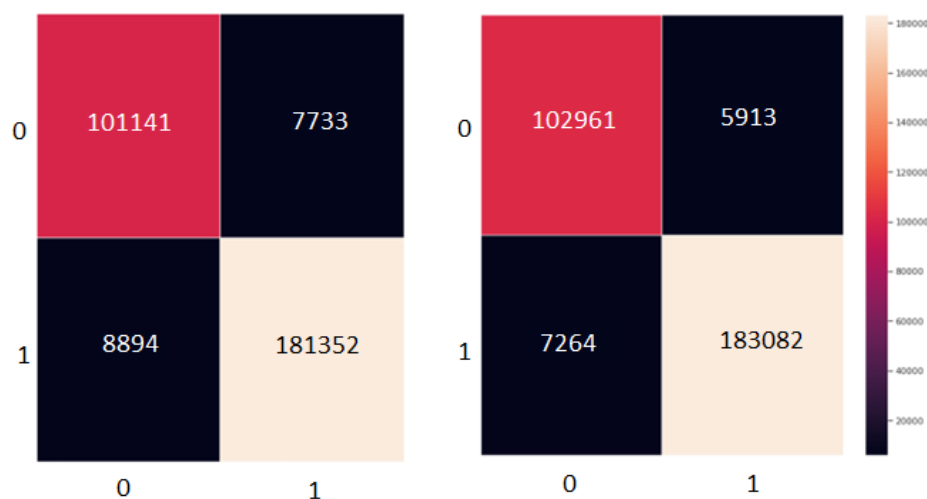
iar funcția de pierdere scade la 0.11. Un scor îmbunătățit destul de bine după un proces de pregătire a bazei de date ce nu pare așa important.



Figură 5.3.1-4 Curbe de învățare după auto-corectare

	precision	recall	f1-score		precision	recall	f1-score	support
mental	0.92	0.93	0.92		0.93	0.95	0.94	108874
nonmental	0.96	0.95	0.96		0.97	0.96	0.97	190346
accuracy			0.94				0.96	299220
macro avg	0.94	0.94	0.94		0.95	0.95	0.95	299220
weighted avg	0.94	0.94	0.94		0.96	0.96	0.96	299220

Figură 5.3.1-5 Raport de clasificare înainte și după auto-corectare



Figură 5.3.1-6 Metrice de confuzie înainte și după auto-corectare

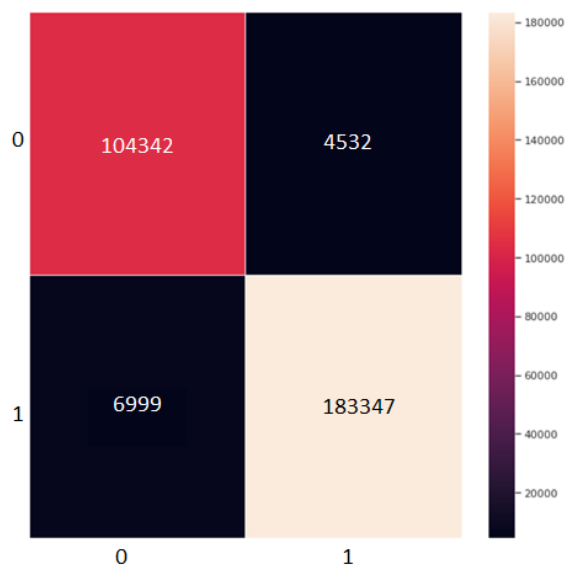
Folosind în loc de o lungime a intrărilor de maxim 70 de cuvinte, un număr de 300 am obținut scoruri asemănătoare, ambele având o acuratețe de 96%. Acest



lucru ne arată că se pot detecta rapid persoanele ce suferă de anumite afecțiuni psihice de cele ce nu suferă, fiind de ajuns doar câteva cuvinte.

	precision	recall	f1-score	support
mental	0.94	0.96	0.95	108874
nonmental	0.98	0.96	0.97	190346
accuracy			0.96	299220
macro avg	0.96	0.96	0.96	299220
weighted avg	0.96	0.96	0.96	299220

Figură 5.3.1-7 Raport de clasificare cu 300 de cuvinte



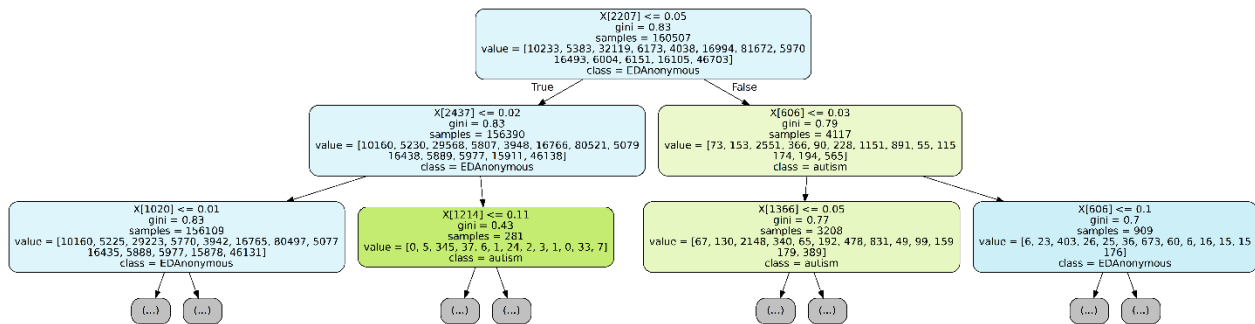
Figură 5.3.1-8 Matrice de confuzie cu 300 de caractere

Datorită folosirii a două baze de date, următoarele modele au fost folosite pentru a compara rezultatele cu diverși parametrii, tehnici de preprocesare și cu diverse seturi de date.

### 5.3.2. Random Forest

Algoritmul Random Forest se bazează pe o mulțime de arbori de decizie ce sunt folosiți pentru luarea de decizii.

Un exemplu de arbore de decizie folosit se vede în figura 5.3.2-1, unde este reprezentat începutul unui astfel de arbore în cadrul clasificării noastre. Se poate observa cum acesta va deveni foarte mare, fiind vizibile doar primele 3 nivele în figură.

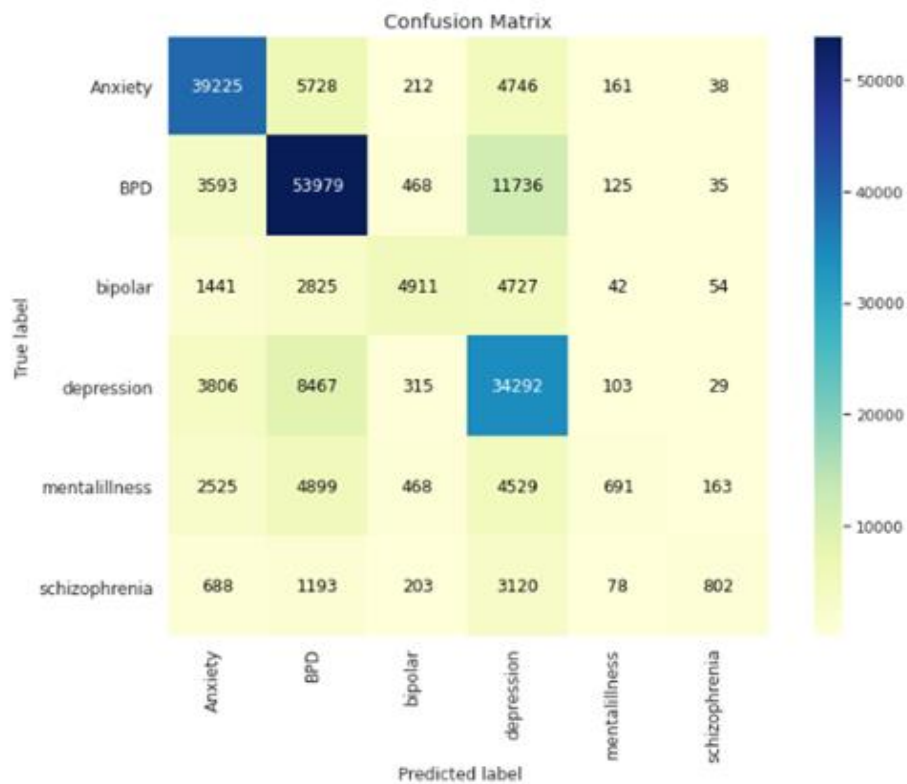


Figură 5.3.2-1 Arbore de decizie

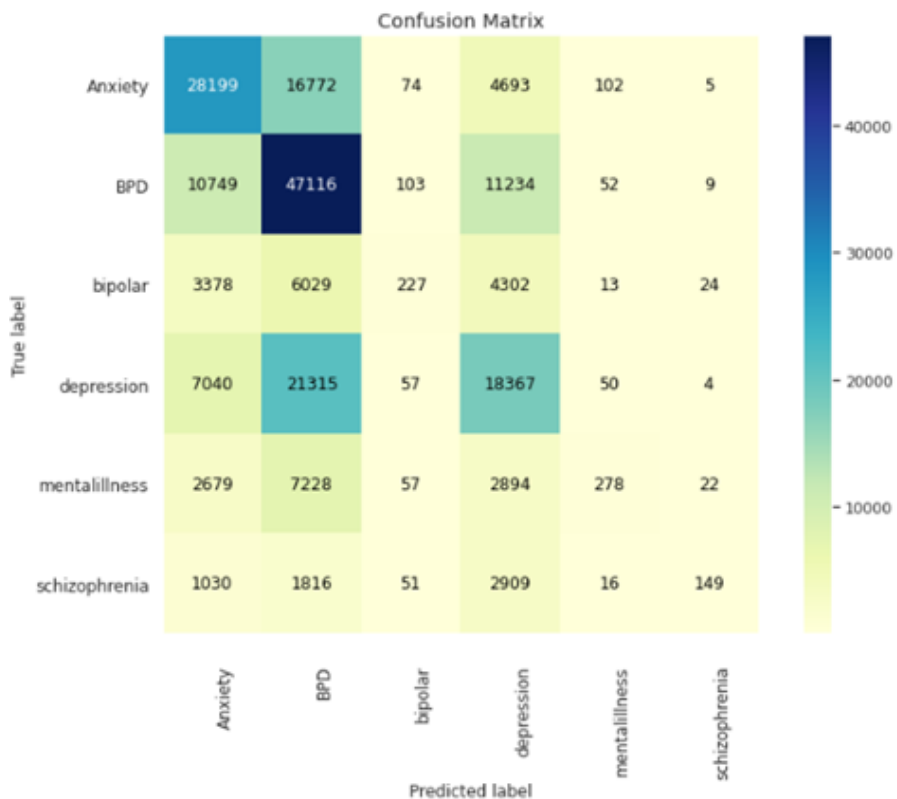
#### 5.3.2.1. Mental Disorders Identification (Reddit)

Pentru prima bază de date folosită [1], s-a realizat preprocesare textului folosind atât tehnica de vectorizare cu TF-IDF, dar și un vectorizator pe bază de GloVe.

Modelul folosit a fost *RandomForestClassifier* din librăria Scikit-Learn cu 100 de estimatori.



Figură 5.3.2-2 Matrice de confuzie baza de date nr.1 – TFIDF

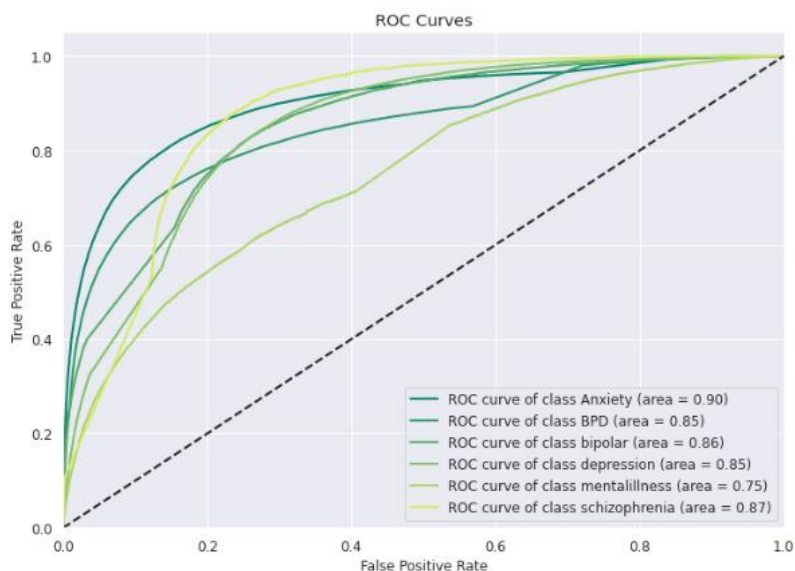


Figură 5.3.2-3 Matrice de confuzie baza de date nr 1 – GloVe

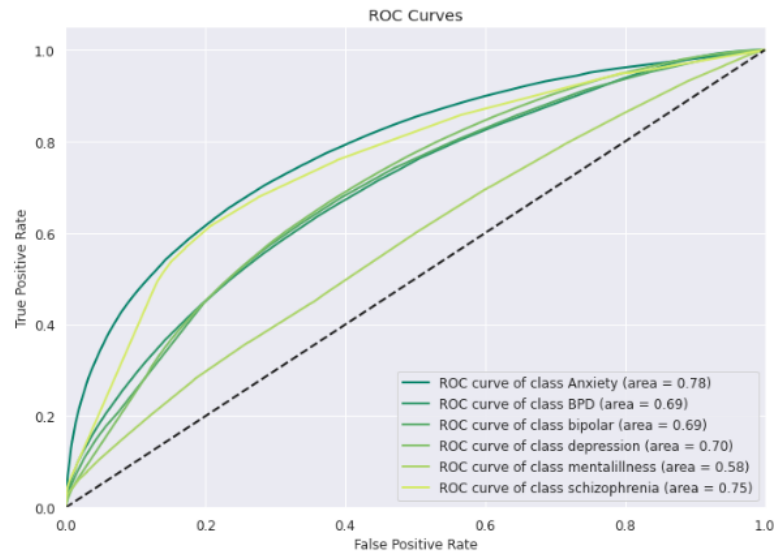
Se poate observa ca modelul nostru tratează mai bine cazul cu TF-IDF, neajutându-se de partea de legături între cuvinte ce o oferă GloVe. Acest lucru se observă din scorurile rezultate, TF-IDF având o acuratețe de 68% iar GloVe de doar 47%; dar și din matricile de confuzie (figura 5.3.2-2 și figură 5.3.2-3), diagonală variantei cu TF-IDF fiind vizibil mai închisă la culoare decât restul căsuțelor, lucru ce nu se poate spune despre varianta cu GloVe.

O altă interpretare rapidă a rezultatelor se află în curba ROC a modelului (figura 5.3.2-4 și figura 5.3.2-5), după cum se vede în varianta utilizând TF-IDF aria de sub grafic este mai mare, sau mai simplu, curba se aproprie vizual mai mult de laturile triunghiului dreptunghic decât de ipotenuza acestuia.

Rezultatele predicțiilor au fost bune, fiind vorba de un ansamblu de arbori de decizie și nu un model neuronal complex.



Figură 5.3.2-4 Curba ROC baza de date nr 1 – TFIDF



Figură 5.3.2-5 Curba ROC baza de date nr 1 – GloVe

### 5.3.2.2. Reddit Mental Health Dataset

Pentru a doua bază de date, s-au folosit aceleași metode de clasificare. Pentru aceasta comparăm rezultatele înainte și după auto-corectarea bazei de date.

		Confusion Matrix												
True label	EDAnonymous	3484	8	97	3	0	24	595	19	22	2	2	25	92
	addiction	37	1378	121	1	2	7	615	12	8	6	4	19	82
	adhd	83	18	11764	21	2	23	1511	13	27	5	5	110	107
	autism	17	3	209	1848	2	15	415	11	18	3	3	86	31
	bipolarreddit	23	8	192	4	508	41	806	12	9	13	26	33	59
	bpd	46	23	133	5	10	4543	2010	8	89	16	13	93	299
	depression	187	66	643	28	25	113	28963	61	663	70	23	308	4049
	healthanxiety	79	5	123	6	3	10	508	1721	7	2	17	66	47
	lonely	22	17	53	14	0	16	3449	2	3036	1	3	206	271
	ptsd	13	5	68	8	5	37	741	12	13	1569	4	30	88
	schizophrenia	33	8	178	5	11	16	691	17	20	10	1494	33	98
	socialanxiety	39	9	137	12	0	20	1861	25	162	10	7	4503	114
	suicidewatch	59	31	94	23	6	78	8724	18	162	33	17	59	10544
		EDAnonymous	addiction	adhd	autism	bipolarreddit	bpd	depression	healthanxiety	lonely	ptsd	schizophrenia	socialanxiety	suicidewatch
		Predicted label												

Figură 5.3.2-6 Matrice de confuzie baza de date nr.2 - TFIDF

Confusion Matrix

True label	EDAnonymous	712	1	564	9	0	5	2398	2	15	0	3	73	540
	addiction	15	429	563	0	1	3	1052	2	3	0	1	19	187
	adhd	52	1	6283	20	3	11	6324	11	15	1	6	118	883
	autism	12	0	419	529	0	7	1368	4	16	0	7	77	244
	bipolarreddit	8	0	276	3	264	10	971	1	8	1	5	36	143
	bpd	11	0	678	6	3	1113	4790	2	32	0	8	66	570
	depression	66	4	2219	13	2	26	28722	3	272	2	10	212	3609
	healthanxiety	22	1	454	6	1	5	1384	372	4	0	6	123	185
	lonely	14	0	224	3	0	2	4748	3	1291	0	0	79	786
	ptsd	6	1	416	5	1	6	1520	0	5	469	0	14	126
	schizophrenia	6	0	247	6	0	5	1510	8	10	2	494	68	298
	socialanxiety	26	1	408	7	1	6	3749	12	56	1	3	2118	630
	suicidewatch	40	3	1202	18	0	13	12627	1	145	5	6	88	5599
		EDAnonymous	addiction	adhd	autism	bipolarreddit	bpd	depression	healthanxiety	lonely	ptsd	schizophrenia	socialanxiety	suicidewatch
		Predicted label												

Figură 5.3.2-7 Matricea de confuzie baza de date nr.2 - GloVe

În această bază de date, scorurile înainte de corectat sunt relativ asemănătoare cu cele de la prima bază de date, cu o valoare a acurateței de 69% pentru modelul ce a folosit TF-IDF și 44% pentru cel ce a folosit GloVe. Se observă din matricea de confuzie de la TFIDF ca eticheta „suicidewatch” este foarte des încurcată cu eticheta de „depression”, iar invers mai mult decât cu alte etichete, făcând o legătură între cei cu gânduri sinucigașe și cei ce suferă de depresie. Majoritatea predicțiilor false se observă ca sunt etichetate ca fiind „depression”, această fiind clasa majoritară dar și una din cele mai răspândite boli în ultimii ani.

După corectarea bazei de date, scorurile sunt mai mici cu aproximativ 1%, lucru ce nu este neapărat influențat de baza de date în sine, ci de ce date au fost folosite în procesul de antrenare, datorat împărțirii random a acestora în date de antrenare, test și validare. Scorurile sunt de 68% și 44% pentru varianta cu TF-IDF și GloVe. Aceste diferențe se pot observa totuși și din matriciile de confuzie, la valorile acestora atât de pe diagonala principală cât și în cazul celorlalte căsuțe.

În ambele matrici de confuzie se observă totuși postări a căror etichetă este prezisă ca fiind „depression”, lucru ce va fi rezolvat în viitoarele modele.

Confusion Matrix

EDAnonymous	3404	9	106	3	0	35	645	19	24	7	1	30	90
addiction	35	1381	135	2	1	11	615	8	9	4	2	18	71
adhd	95	20	11694	25	2	58	1541	11	34	5	4	114	86
autism	25	4	225	1838	1	20	397	9	15	5	3	89	30
bipolarreddit	24	9	199	4	481	45	811	11	6	17	31	36	60
bpd	75	35	215	15	12	3437	2869	13	90	43	22	104	358
depression	197	68	690	40	21	141	28993	63	632	74	26	276	3978
healthanxiety	76	4	126	6	3	30	512	1693	5	4	16	68	51
lonely	18	12	70	17	0	33	3457	2	3016	5	3	184	273
ptsd	15	3	69	7	5	57	701	7	11	1594	3	30	91
schizophrenia	33	9	158	7	10	36	699	14	24	15	1490	28	91
socialanxiety	46	8	155	20	0	31	1882	20	159	12	7	4456	103
suicidewatch	62	29	103	20	5	45	8915	15	152	47	24	52	10379

Predicted label

Figură 5.3.2-8 Matricea de confuzie baza de date nr. 2 – TFIDF – după autocorectare [5]



Prin rapoartele de clasificare se observă metricile de evaluare complete pentru fiecare clasă a procesului de clasificare, cum ar fi precizia (precision), regăsirea (recall), scorul f1 (f1-score).

		Confusion Matrix												
True label	EDAnonymous	692	0	574	2	0	4	2422	1	19	0	4	71	533
	addiction	11	427	570	0	1	2	1049	3	3	0	1	15	193
	adhd	42	1	6461	16	3	5	6196	7	21	1	7	112	856
	autism	14	0	484	497	0	2	1383	6	9	0	7	57	224
	bipolarreddit	7	0	298	2	264	5	943	4	7	1	6	28	161
	bpd	8	0	739	2	3	1079	4746	5	41	1	7	78	570
	depression	61	4	2164	12	2	20	28780	6	264	2	10	198	3637
	healthanxiety	35	1	401	4	0	5	1474	362	5	0	2	97	177
	lonely	15	0	237	0	0	2	4789	3	1241	0	1	89	773
	ptsd	4	1	304	3	2	7	1612	3	6	469	0	29	129
	schizophrenia	13	0	234	4	0	2	1518	4	7	2	504	49	317
	socialanxiety	24	0	451	8	1	8	3845	10	61	1	2	1976	631
	suicidewatch	48	2	1205	11	0	6	12618	2	145	3	6	91	5610
		EDAnonymous	addiction	adhd	autism	bipolarreddit	bpd	depression	healthanxiety	lonely	ptsd	schizophrenia	socialanxiety	suicidewatch
		Predicted label												

Figură 5.3.2-9 Matrice de confuzie baza de date nr.2 – GloVe – după autocorectare [6]

	precision	recall	f1-score	support
EDAnonymous	0.83	0.78	0.80	4373
addiction	0.87	0.60	0.71	2292
adhd	0.84	0.85	0.85	13689
autism	0.92	0.69	0.79	2661
bipolarreddit	0.89	0.28	0.42	1734
bpd	0.86	0.47	0.61	7288
depression	0.56	0.82	0.66	35199
healthanxiety	0.90	0.65	0.76	2594
lonely	0.72	0.43	0.54	7090
ptsd	0.87	0.61	0.72	2593
schizophrenia	0.91	0.57	0.70	2614
socialanxiety	0.81	0.65	0.72	6899
suicidewatch	0.66	0.52	0.58	19848
accuracy			0.68	108874
macro avg	0.82	0.61	0.68	108874
weighted avg	0.71	0.68	0.67	108874

Figură 5.3.2-10 Raport de clasificare baza de date nr. 2 - TF-IDF {5}

	precision	recall	f1-score	support
EDAnonymous	0.71	0.16	0.26	4322
addiction	0.98	0.19	0.32	2275
adhd	0.46	0.47	0.46	13728
autism	0.89	0.19	0.31	2683
bipolarreddit	0.96	0.15	0.26	1726
bpd	0.94	0.15	0.26	7279
depression	0.40	0.82	0.54	35160
healthanxiety	0.87	0.14	0.24	2563
lonely	0.68	0.17	0.28	7150
ptsd	0.98	0.18	0.31	2569
schizophrenia	0.90	0.19	0.31	2654
socialanxiety	0.68	0.28	0.40	7018
suicidewatch	0.41	0.28	0.33	19747
accuracy			0.44	108874
macro avg	0.76	0.26	0.33	108874
weighted avg	0.56	0.44	0.40	108874

Figură 5.3.2-11 Raport de clasificare baza de date nr.2 – GloVe {6}

Se poate observa diferența dintre cele două tipuri de preprocesare a textului în special în cazul regăsirii în cadrul clasificării, unde diferențele sunt semnificative între cele două implementări, ceea ce aduce scoruri foarte diferite la final atât pentru fiecare clasă cât și la rezultatele finale.

Tabel 1 Random Forest

Nr.	Bază de date	Metodă preprocesare	Corectare	Acc	P	R	F1
1.	[1]	TF-IDF	NU	66%	69%	52%	54%
2.	[1]	GloVe	NU	47%	51%	28%	27%
3.	[2]	TF-IDF	NU	68%	81%	62%	68%
4.	[2]	GloVe	NU	44%	75%	28%	33%
<b>5.</b>	<b>[2]</b>	<b>TF-IDF</b>	<b>DA</b>	<b>68%</b>	<b>82%</b>	<b>61%</b>	<b>68%</b>
6.	[2]	GloVe	DA	44%	76%	26%	33%

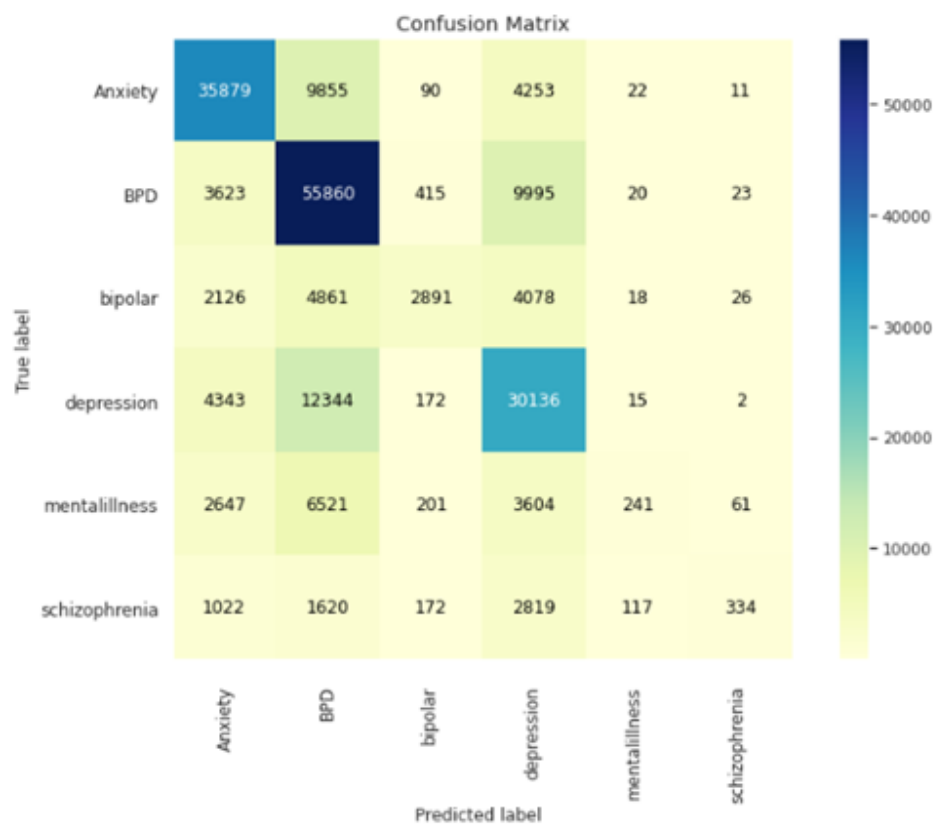
Se poate observa cel mai bun scor în cazul preprocesării cu TF-IDF. Ajungem astfel la concluzia, că Random Forest nu ține cont de legăturile între cuvinte așa cum le realizează GloVe, ba chiar este oarecum încurcat de acestea. Iar o tehnică așa simplă precum corectarea cuvintelor în cadrul dicționarului englez nu influențează așa mult deciziile realizate de algoritm în procesul de clasificare.

### 5.3.3. Multimonial Naive Bayes

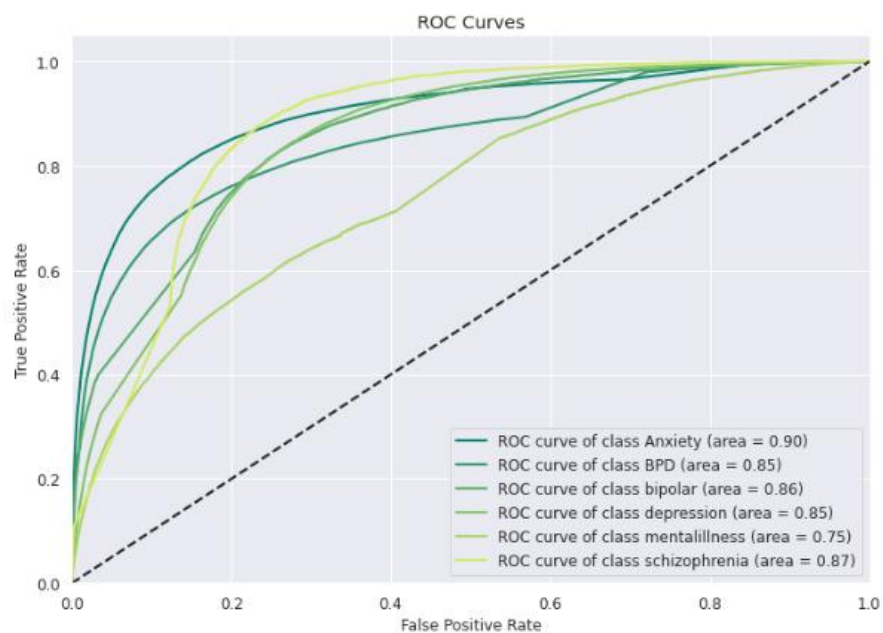
Și în cadrul algoritmului Multinomial Naive Bayes o să facem o clasificare pentru prima bază de date folosind una din metode, și pentru cea de-a doua bază de date folosind una din metode dar și observând diferența între baza de date corectată și necorectată. Pentru cea de a doua metodă, folosind încorporările GloVe, modelul de față nu este potrivit în privința asocierilor dintre intrări și ieșiri.

#### 5.3.3.1. Mental Disorders Identification (Reddit)

În prima bază avem rezultate de 62% folosind metoda TF-IDF, lucru ce poate observa că nu înseamnă o valoare neapărat bună, după cum se vede în matricea de confuzie.



Figură 5.3.3-1 Matrice de confuzie baza de date nr. 1 – TFIDF



Figură 5.3.3-2 Curba ROC baza de date nr.1 – TFIDF

### 5.3.3.2. Reddit Mental Health Dataset

În cea de-a doua bază de date avem rezultate de 58% folosind TF-IDF. Se poate observa că cu TFIDF algoritmul are rezultate bune finale ca și scor, dar nu ca și clasificare, după cum se observă în matricea de confuzie. Majoritatea intrărilor au fost clasificate ca și „depression”, rezultând un scor bun din cauza numărului mult mai mare de postări din această clasă.

Confusion Matrix

EDAnonymous	2977	5	112	6	0	12	1182	32	9	2	2	13	21
addiction	26	815	165	1	1	9	1198	10	7	3	1	3	53
adhd	88	18	10325	44	9	50	3016	25	8	7	4	80	15
autism	13	5	338	1323	0	14	852	5	16	4	5	70	16
bipolarreddit	18	5	280	4	359	57	924	7	9	10	31	3	27
bpd	30	8	257	12	40	2903	3717	9	66	21	13	43	169
depression	163	29	964	15	41	55	31330	48	302	25	19	105	2103
healthanxiety	43	3	142	7	2	6	726	1627	5	3	8	10	12
lonely	4	0	18	7	0	12	5242	1	1573	1	1	133	98
ptsd	7	2	109	9	5	59	1521	19	3	774	8	16	61
schizophrenia	17	9	187	6	26	23	1271	9	10	2	993	15	46
socialanxiety	32	6	251	12	1	13	3578	21	189	5	5	2770	16
suicidewatch	26	7	85	5	4	17	13014	15	70	11	10	9	6575

Predicted label

Figură 5.3.3-3 Matrice de confuzie baza de date nr.2 - TFIDF

În cazul variantei cu TF-IDF, chiar dacă acesta mai prezice unele intrări corect, nu prezice corect cele mai multe dintre intrările din fiecare clasă. De exemplu, în cazul clase „lonely” s-a prezis ca fiind din clasa „lonely” un număr de 1573 și din clasa „depression” un număr de 5242, mult mai mare decât prezirea adevărată. Datorită tipului de clasificator și a tipului de date necesare pentru acest algoritm, varianta cu GloVe nu este potrivită în combinație cu acesta.

După corectarea limbajului, scorurile nu se îmbunătățesc, cu o acuratețe de 59% și 32%, prezicerile rămân la fel în cazul ambelor metode.

	precision	recall	f1-score	support
EDAnonymous	0.86	0.68	0.76	4373
addiction	0.89	0.36	0.51	2292
adhd	0.78	0.75	0.77	13689
autism	0.91	0.50	0.64	2661
bipolarreddit	0.74	0.21	0.32	1734
bpd	0.90	0.40	0.55	7288
depression	0.46	0.89	0.61	35199
healthanxiety	0.89	0.63	0.74	2594
lonely	0.69	0.22	0.34	7090
ptsd	0.89	0.30	0.45	2593
schizophrenia	0.90	0.38	0.53	2614
socialanxiety	0.85	0.40	0.54	6899
suicidewatch	0.71	0.33	0.45	19848
accuracy			0.59	108874
macro avg	0.81	0.46	0.56	108874
weighted avg	0.69	0.59	0.57	108874

Figură 5.3.3-4 Raport de clasificare bazate date nr. 2 cu TFIDF [5]

Tabel 2 Multinomial naive bayes

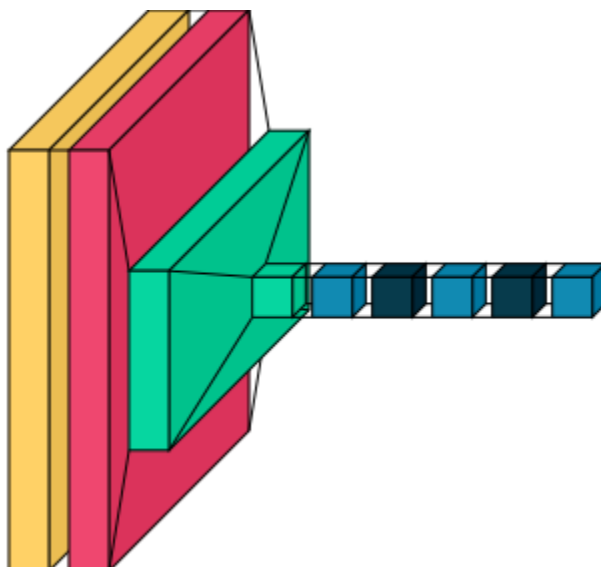
Nr.	Bază de date	Metodă preprocesare	Corectare	Acc	P	R	F1
1.	[1]	TF-IDF	NU	62%	66%	43%	45%
2.	[2]	TF-IDF	NU	58%	80%	45%	55%
3.	[2]	TF-IDF	DA	59%	81%	46%	56%

Cel mai bun scor l-a avut clasificatorul folosind metoda TF-IDF cu o valoare a acurateței de 59%.

### 5.3.4. Long Short Term Memory (LSTM)

Pentru acest tip de model am folosit API-ul de la keras și am construit un model secvențial cu un strat de încorporări la care am adăugat în una din variante matricea de încorporări de la GloVe pe care am descris-o în capitolele 2.2.8 și 5.2.1. Am avut două straturi bidirecționale cu LSTM pentru a ajuta la învățarea atât pe bază de timp anterior și posterior momentului de timp actual. Modelul dispune și de mai multe straturi de dense, pentru a conecta toți neuronii unul de altul, dar și de straturi de abandon (*Dropout*) pentru a evita supraînvățarea. Am folosit același model utilizând mai multe tipuri de parametrii, ca de exemplu am luat pentru partea de tokenizare un număr de 70 cuvinte pentru fiecare, dar și un număr de 300 de

cuvinte pentru a observa diferența scorurilor dacă ținem cont de cât mai multe date din fiecare postare, sau doar de începutul acestora S-a utilizat optimizatorul Adam cu o rată de învățare de 0.001 și ca funcții de callback: Early Stopping și ReduceLR. Ce ne ajută la modificarea ratei de învățare în timp și oprirea automată a antrenării în cazul în care scorurile nu se îmbunătățesc. La ambele am folosit o răbdare („*patience*”) de 5 epoci și un *min\_delta* de 0.01.



Figură 5.3.4-1 Vizualizare visulkeras model LSTM

În Figura 5.3.4-1 se observă o vizualizare a modelului folosind biblioteca *visulkeras*<sup>7</sup>, iar în figura o vizualizare mai detaliată, unde se pot observa mai bine dimensiunile de intrare și ieșire din fiecare strat. Se poate vedea cum dimensiunea de input a modelului este de 70, dimensiunea numărului maxim de cuvinte a datelor de antrenare. După ieșirea din stratul de încorporare (*Embedding*), se crează datele cu dimensiunea de 70, 300 ce se referă la 70 de cuvinte cu dimensiunea încorporărilor de 300. Stratul de abandon spațial elimină câteva legături între neuroni. Straturile de *LSTM* crează legăturile „inteligente” între caracteristicile textului, ajungând la o ieșire de 128 de neuroni la finalul ultimului strat. Straturile de dens și abandon următoare ne ajută să învățăm și în același timp să ajungem la dimensiuna dorită la finalul antrenării, și anume 13, numărul de clase din cea de-a doua bază de date folosită

<sup>7</sup> <https://github.com/paulgavrikov/visulkeras>

```

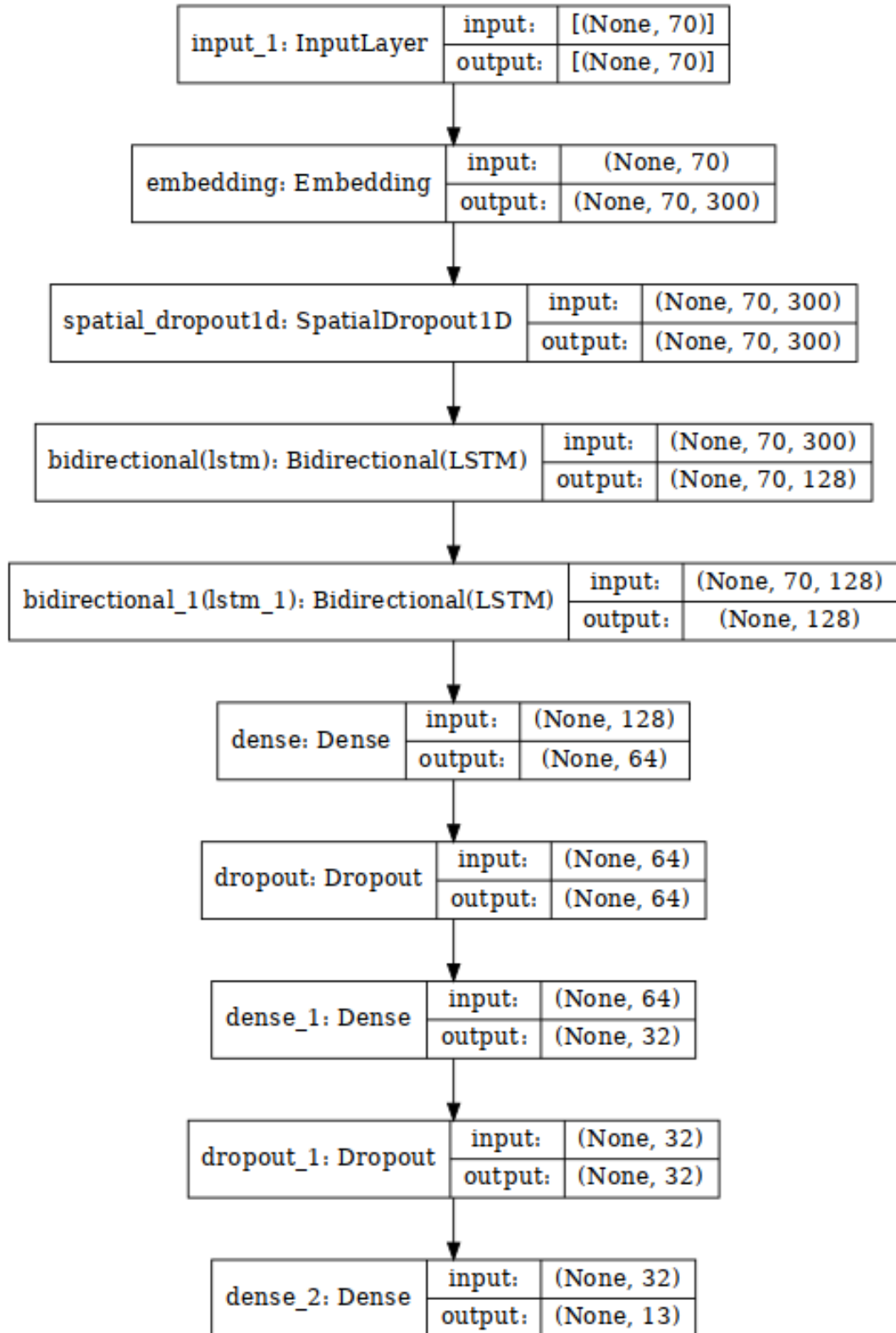
model1 = Sequential([
    Embedding(vocab_size, EMBEDDING_DIM, weights=[embedding_matrix],
              trainable=False, input_length=MAX_SEQUENCE_LENGTH),
    SpatialDropout1D(0.2),
    Bidirectional(LSTM(64, dropout=0.2, return_sequences=True)),
    Bidirectional(LSTM(64)),
    Dense(64, activation='relu'),
    Dropout(0.3),
    Dense(32, activation='relu'),
    Dropout(0.3),
    Dense(number_of_labels, activation='sigmoid')
])

```

*Figură 5.3.4-2 Secvență de cod pentru modelul LSTM*

Pentru a concepe un studiu cât mai amănunțit asupra celei mai bune variante a modelului, pe lângă tipul de preprocesare a textului, a numărului de cuvinte folosit la tokenizare, s-a utilizat și antrenarea cu ponderi ale claselor, un mecanism folosit în cazul bazelor de date neechilibrate pentru a se ține cont de importanța fiecărei clase în legătură cu numărul de intrări din fiecare și augmentarea bazei de date în timpul antrenării. S-a realizat și o comparație a modelului de bază LSTM între ambele baze de date pentru a vedea influență datelor, raportului dintre acestea și a diferențelor între numărul de etichete în cadrul rezultatelor finale.

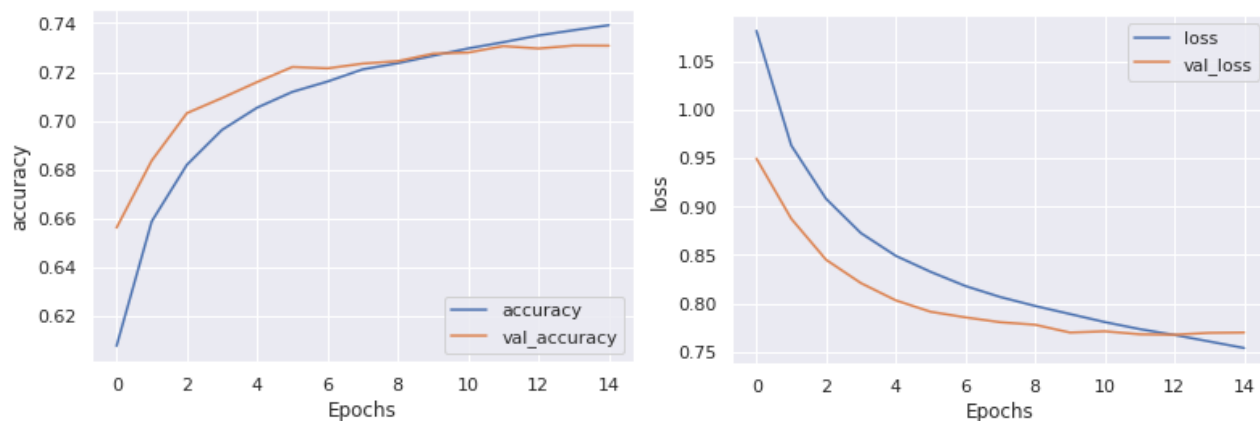




Figură 5.3.4-3 Visualizare model LSTM

### 5.3.4.1. Mental Disorders Identification (Reddit)

În cazul primei baze de date, avem rezultate ale acurateții de 73% în cazul tokenizării propozițiilor și de 70% când se folosesc și încorporările GloVe.



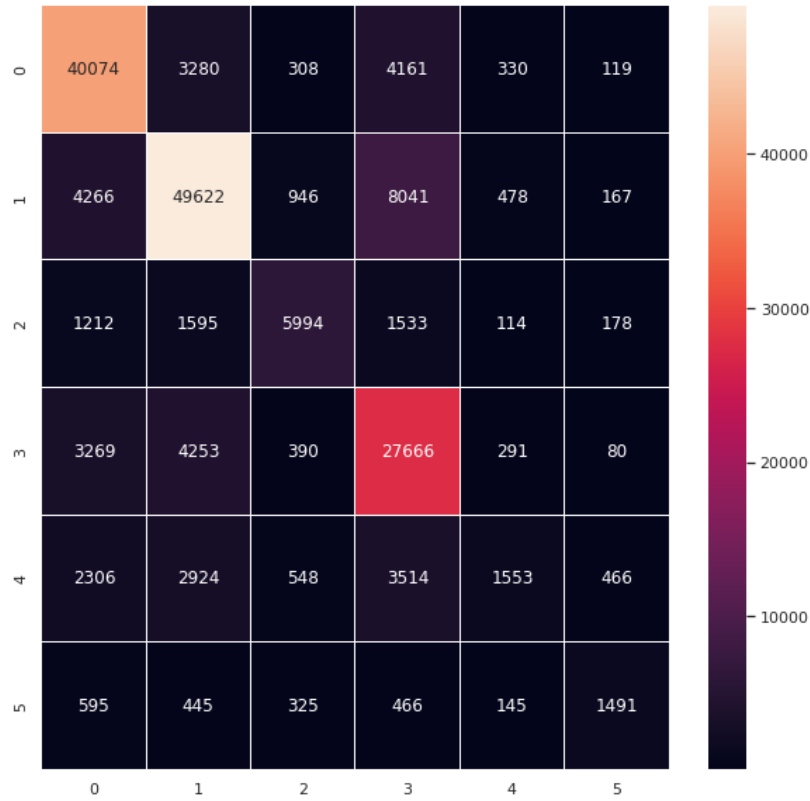
Figură 5.3.4-4 Curbe de învățare model LSTM - BDI - Tokenizare

Antrenarea în primul caz, unde s-a folosit tokenizarea ca simplă metodă durează 14 epoci, unde reușește să atingă epogeul de 73% în cazul datelor de validare. Se observă că la mai multe epoci, pe datele de antrenare s-ar fi produs un proces de supraînvățare.

	precision	recall	f1-score	support
Anxiety	0.77	0.83	0.80	48272
BPD	0.80	0.78	0.79	63520
bipolar	0.70	0.56	0.63	10626
depression	0.61	0.77	0.68	35949
mentalillness	0.53	0.14	0.22	11311
schizophrenia	0.60	0.43	0.50	3467
accuracy			0.73	173145
macro avg	0.67	0.59	0.60	173145
weighted avg	0.73	0.73	0.72	173145

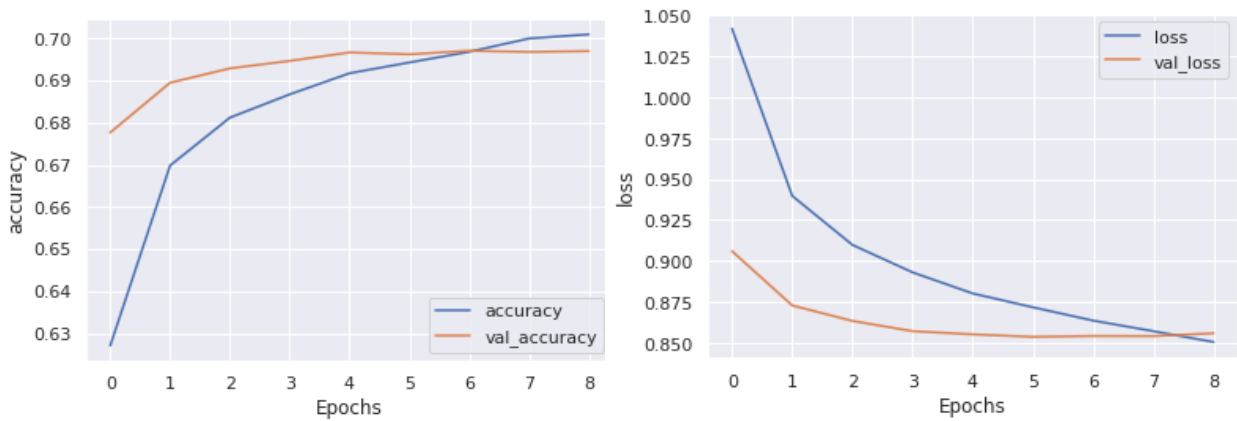
Figură 5.3.4-5 Raport de clasificare LSTM - BDI – Tokenizare

# NECLASIFICAT



Figură 5.3.4-6 Matrice de confuzie LSTM - BD1 - Tokenizare

În cazul utilizării de încorporări GloVe, apogeul este atins după un număr de 8 epoci înainte de a se aplatiza atât în cazul antrenării cât și în cel al validării și are o valoare de 70%.



Figură 5.3.4-7 Curbe de învățare model LSTM - BD2 - Tokenizare + GloVe

# NECLASIFICAT

	precision	recall	f1-score	support
Anxiety	0.70	0.76	0.73	48272
BPD	0.75	0.79	0.77	63520
bipolar	0.69	0.52	0.59	10626
depression	0.61	0.70	0.66	35949
mentalillness	0.53	0.12	0.20	11311
schizophrenia	0.51	0.34	0.41	3467
accuracy			0.70	173145
macro avg	0.63	0.54	0.56	173145
weighted avg	0.69	0.70	0.68	173145

Figură 5.3.4-8 Raport de clasificare LSTM - BD2 – GloVe



Figură 5.3.4-9 Matrice de confuzie LSTM - BD1 – GloVe

Rezultatele sunt mai bune față de Random Forest folosind simpla tokenizare a datelor de intrare în cazul acestei baze de date și un număr de 70 de cuvinte. Acest lucru se poate datora din cauza nenecesarului numărului de cuvinte potrivite pentru a forma relațiile dintre acestea pe baza încorporărilor.

### 5.3.4.1. Reddit Mental Health Dataset

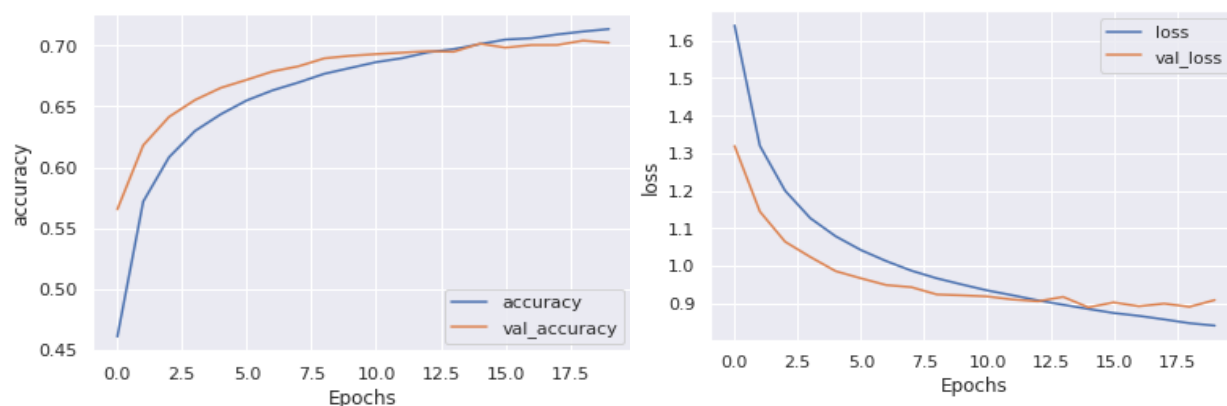
În cea de-a doua bază de date rezultatele folosind tokenizare sunt de 70% și folosind încorporările sunt de 69%, ambele cu un număr de 70 de cuvinte, acestea fiind valorile de pornire de la care am început pentru acest tip de model.

S-au utilizat mai multe tipuri de ponderi ale claselor. Acestea sunt folosite în prezicerea bazelor de date dezechilibrate și se calculează în mai multe moduri. Pentru primul tip se calculează raportul dintre numărul de intrări dintr-o anumită clasă și numărul total de intrări. O a doua metodă se calculează cu ajutorul logaritmului și a unei constante alese.

Un alt mecanism folosit în cazul antrenării unui clasificator este augmentarea textului. Acest mecanism este folosit pentru a mări baza de date prin dublarea unui procent ales de date și schimbarea unor informații din acestea păstrând sensul de bază. Există diverse metode de a face acest lucru, cum ar fi:

- Augmentarea cu sinonime, ce constă în înlocuirea cuvintelor aleator cu sinonimele lor.
- Augmentarea bazată pe caracter, unde se înlocuiește aleator un caracter cu unul asemănător pentru a crea zgomot în cadrul datelor.

În cazul tokenizării simple, corectarea textului pe baza dicționarului englez nu aduce o îmbunătățire a scorurilor, datorită limbajului folosit în baza de date și posibil a greșelilor nedetectate, după cum se observă în cazul intrării 3 și 4 din tabelul 3. Rularea ambelor modele se face în aproximativ același număr de epoci, modelul utilizând 300 de cuvinte și oprindu-se cu 2 epoci mai repede.

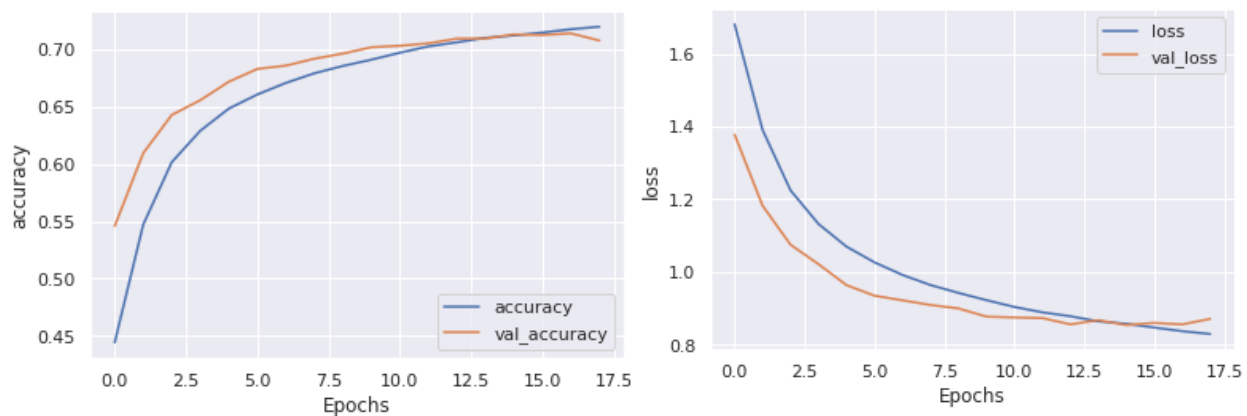


Figură 5.3.4-10 Curbe de învățare LSTM Tokenizare cu 70 cuvinte {3}

# NECLASIFICAT

	precision	recall	f1-score	support
EDAnonymous	0.86	0.81	0.84	4373
addiction	0.79	0.77	0.78	2292
adhd	0.85	0.85	0.85	13689
autism	0.91	0.72	0.80	2661
bipolarreddit	0.76	0.44	0.56	1734
bpd	0.86	0.52	0.65	7288
depression	0.62	0.75	0.68	35199
healthanxiety	0.84	0.82	0.83	2594
lonely	0.60	0.55	0.57	7090
ptsd	0.85	0.63	0.72	2593
schizophrenia	0.83	0.60	0.70	2614
socialanxiety	0.73	0.73	0.73	6899
suicidewatch	0.65	0.61	0.62	19848
accuracy			0.70	108874
macro avg	0.78	0.68	0.72	108874
weighted avg	0.71	0.70	0.70	108874

Figură 5.3.4-11 Raport clasificare LSTM ?Tokenizare 70 cuvinte {3}



Figură 5.3.4-12 Curbe de învățare LSTM Tokenizare 300 cuvinte {4}

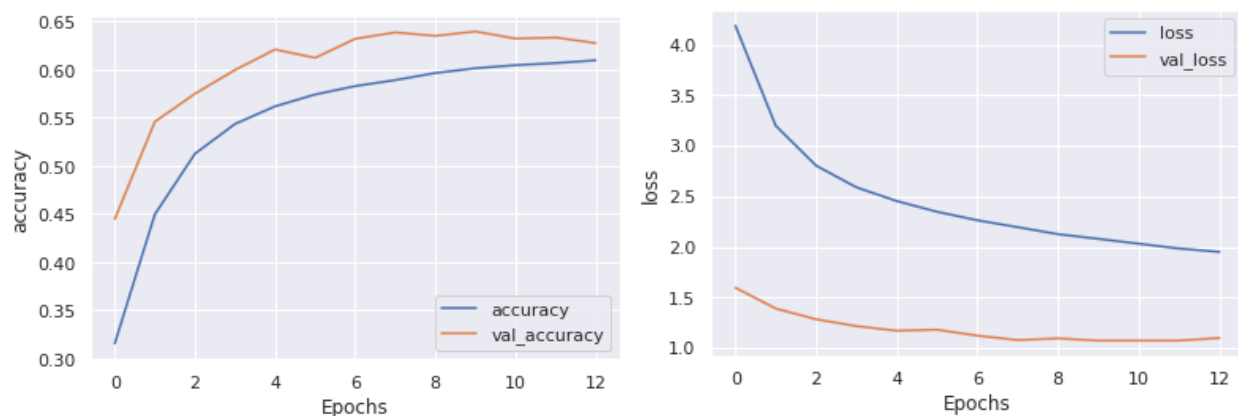
# NECLASIFICAT

	precision	recall	f1-score	support
EDAnonymous	0.86	0.82	0.84	4373
addiction	0.80	0.78	0.79	2292
adhd	0.87	0.86	0.86	13689
autism	0.89	0.72	0.80	2661
bipolarreddit	0.78	0.40	0.53	1734
bpd	0.83	0.57	0.67	7288
depression	0.64	0.74	0.69	35199
healthanxiety	0.82	0.84	0.83	2594
lonely	0.58	0.60	0.59	7090
ptsd	0.85	0.67	0.75	2593
schizophrenia	0.77	0.38	0.51	2614
socialanxiety	0.66	0.79	0.72	6899
suicidewatch	0.64	0.61	0.62	19848
accuracy			0.70	108874
macro avg	0.77	0.67	0.71	108874
weighted avg	0.71	0.70	0.70	108874

Figură 5.3.4-13 Raport clasificare LSTM Tokenizare 300 cuvinte {4}

În cazul utilizării unui număr mai mare de cuvinte în tokenizare, scorurile rămân identice, se observă astfel că nu are așa mare importanță numărul de informații rezultate în urma procesării, după cum se observă în cazul intrărilor 4 și 7 din tabelul 3.

În utilizarea ponderilor pentru un număr de 70 de cuvinte, prima variantă de ponderi folosite crește regăsirea cu care modelul clasifică datele, dar la final rezultatele sunt mai proaste, de 62% acuratețe, intrarea 5 din tabelul 3. Iar în cazul utilizării celei de a doua variante de ponderi, în intrarea 6 din tabelul 3, precizia cu care sunt clasificate datele scade cu 3% iar regăsirea crește cu 3%, în final având același scor f1 și de acuratețe.

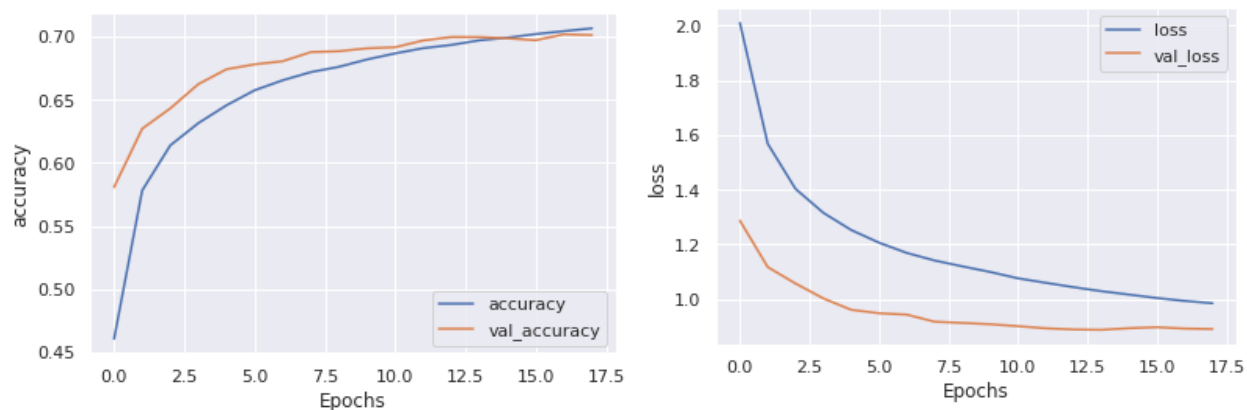


Figură 5.3.4-14 Curbe de învățare LSTM Tokenizare 70 cuvinte - greutatea varianta 1 {5}

# NECLASIFICAT

	precision	recall	f1-score	support
EDAnonymous	0.62	0.89	0.73	4373
addiction	0.32	0.91	0.47	2292
adhd	0.87	0.77	0.82	13689
autism	0.67	0.80	0.73	2661
bipolarreddit	0.35	0.63	0.45	1734
bpd	0.76	0.61	0.67	7288
depression	0.76	0.43	0.55	35199
healthanxiety	0.74	0.83	0.79	2594
lonely	0.36	0.78	0.49	7090
ptsd	0.63	0.74	0.68	2593
schizophrenia	0.64	0.66	0.65	2614
socialanxiety	0.66	0.74	0.69	6899
suicidewatch	0.60	0.61	0.61	19848
accuracy			0.62	108874
macro avg	0.62	0.72	0.64	108874
weighted avg	0.68	0.62	0.63	108874

Figură 5.3.4-15 Raport clasificare LSTM Tokenizare 70 cuvinte – ponderi varianta 1 {5}



Figură 5.3.4-16 Curbe de învățare LSTM Tokenizare 70 cuvinte - ponderi varianta 2 {6}

# NECLASIFICAT



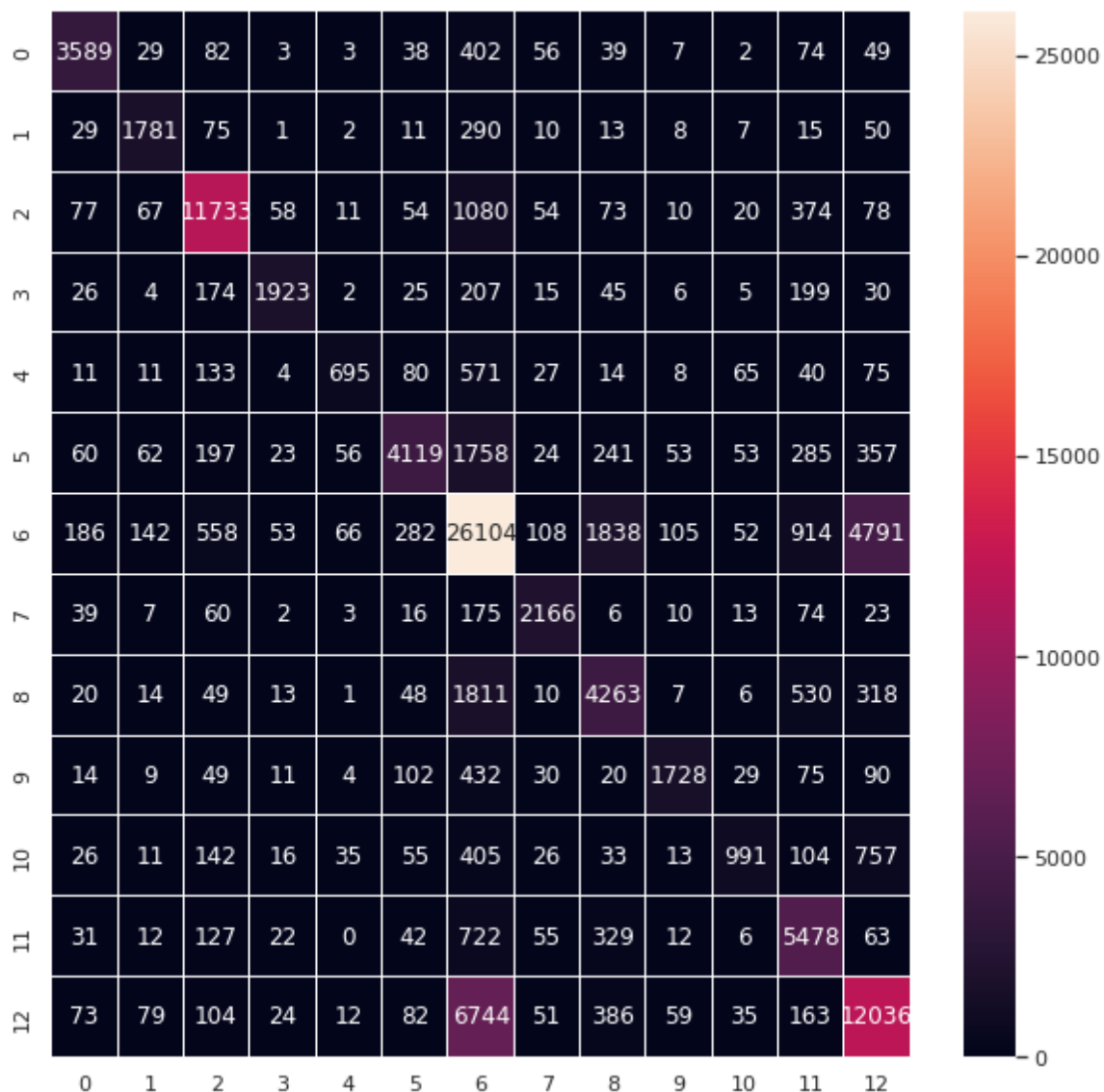
	precision	recall	f1-score	support
EDAnonymous	0.81	0.85	0.83	4373
addiction	0.63	0.85	0.72	2292
adhd	0.84	0.86	0.85	13689
autism	0.86	0.75	0.80	2661
bipolarreddit	0.63	0.56	0.59	1734
bpd	0.89	0.57	0.69	7288
depression	0.63	0.74	0.68	35199
healthanxiety	0.83	0.81	0.82	2594
lonely	0.55	0.60	0.58	7090
ptsd	0.82	0.66	0.73	2593
schizophrenia	0.82	0.63	0.71	2614
socialanxiety	0.76	0.71	0.73	6899
suicidewatch	0.66	0.56	0.61	19848
accuracy			0.70	108874
macro avg	0.75	0.71	0.72	108874
weighted avg	0.71	0.70	0.70	108874

Figură 5.3.4-17 Raport clasificare LSTM Tokenizare 70 cuvinte – ponderi vaiana 2 {6}

Nu putem spune care ar fi cea mai bună variantă în cazul utilizării doar a tokenizării, din cauza că scorurile sunt aproape identice în majoritatea cazurilor. Cel mai bun model ar putea fi astfel considerat cel ce ia cel mai puțin timp, are cele mai puține epoci, cel la care am folosit mai puține date sau după o ordine a metricilor, acest lucru rămâne la preferința utilizabilității acestuia.

În cadrul graficelor se observă că modelul se comportă bine, nu există tendința de supraînvățare, iar diferența dintre datele de antrenare și validare este mică în acest timp. Pentru datele de testare, se observă ca rezultatul acurateței este aproape identic cu cele din cazul antrenării.

În figura 5.3.4-18 este matricea de confuzie a modelului ce folosește 70 cuvinte fără ponderi ale claselor, cu o acuratețe de 70% și un scor f1 de 71%.

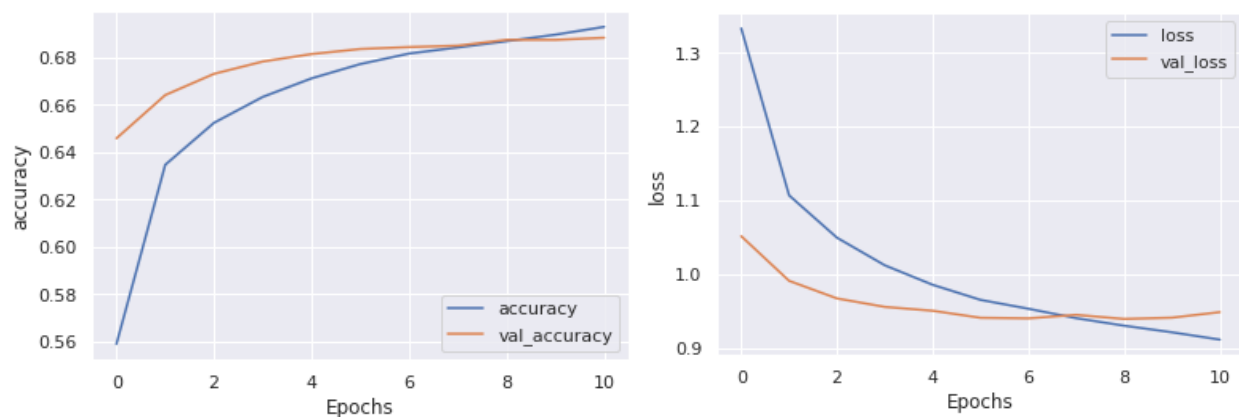


Figură 5.3.4-18 Matrice de confuzie LSTM Tokenizare 300 cuvinte {7}

În cazul utilizării tokenizării împreună cu încorporările GloVe, se observă o mai mare diferență între metodele utilizate. Corectarea textului nu a influențat nici aici rezultatele foarte multe, ambele având valori de 69% în cazul acurateței și de 69-70% la scorul f1.

În schimb, utilizarea unui număr mai mare de cuvinte în cadrul tokenizării a avut un impact asupra noilor rezultate, acuratețea crescând de la 69% la 72% iar scorul f1 de la 70% la 73%.

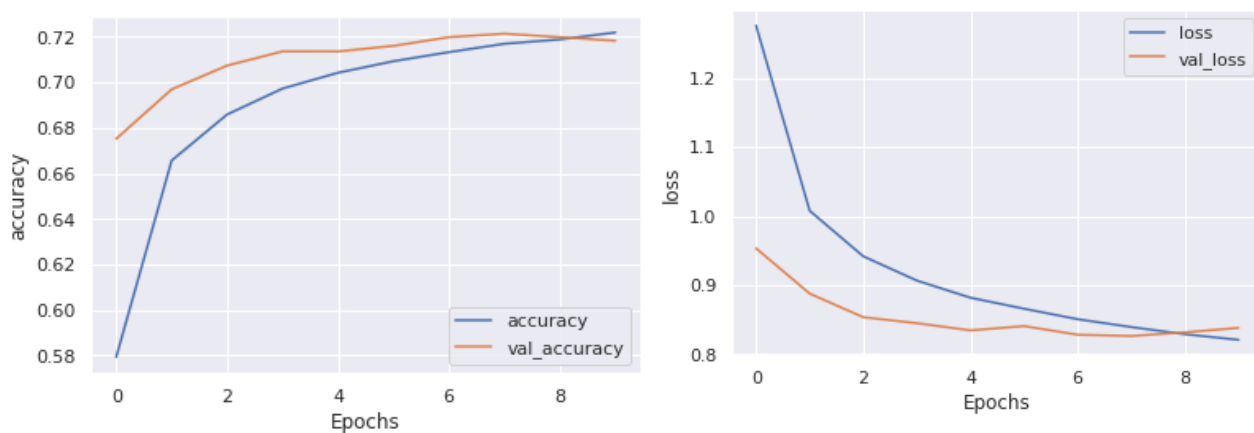
# NECLASIFICAT



Figură 5.3.4-19 Curbe de învățare LSTM GloVe 70 cuvinte {9}

	precision	recall	f1-score	support
EDAnonymous	0.85	0.82	0.84	4373
addiction	0.80	0.75	0.77	2292
adhd	0.82	0.84	0.83	13689
autism	0.83	0.68	0.75	2661
bipolarreddit	0.69	0.44	0.54	1734
bpd	0.85	0.60	0.70	7288
depression	0.62	0.73	0.67	35199
healthanxiety	0.86	0.75	0.80	2594
lonely	0.57	0.54	0.56	7090
ptsd	0.82	0.65	0.73	2593
schizophrenia	0.71	0.50	0.59	2614
socialanxiety	0.67	0.71	0.69	6899
suicidewatch	0.63	0.58	0.60	19848
accuracy			0.69	108874
macro avg	0.75	0.66	0.70	108874
weighted avg	0.69	0.69	0.68	108874

Figură 5.3.4-21 Raport clasificare LSTM GloVe 70 cuvinte {9}



Figură 5.3.4-20 Curbe de învățare LSTM GloVe 300 cuvinte {10}

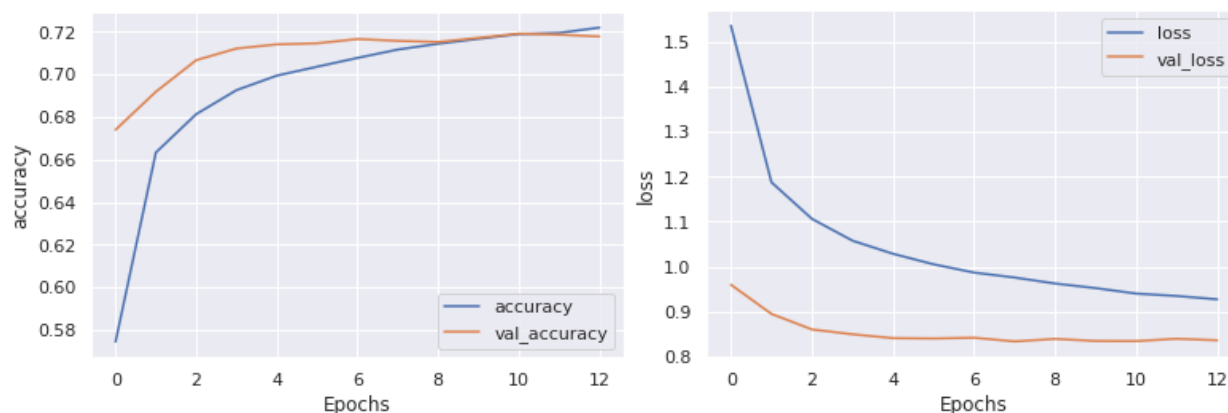
# NECLASIFICAT

	precision	recall	f1-score	support
EDAnonymous	0.87	0.85	0.86	4373
addiction	0.81	0.79	0.80	2292
adhd	0.85	0.87	0.86	13689
autism	0.84	0.78	0.81	2661
bipolarreddit	0.75	0.44	0.56	1734
bpd	0.82	0.60	0.69	7288
depression	0.64	0.77	0.70	35199
healthanxiety	0.87	0.82	0.84	2594
lonely	0.61	0.58	0.60	7090
ptsd	0.89	0.64	0.75	2593
schizophrenia	0.79	0.60	0.68	2614
socialanxiety	0.75	0.75	0.75	6899
suicidewatch	0.69	0.58	0.63	19848
accuracy			0.72	108874
macro avg	0.78	0.70	0.73	108874
weighted avg	0.72	0.72	0.71	108874

Figură 5.3.4-22 Raport clasificare LSTM GloVe 300 cuvinte {10}

S-au folosit cele două variante de ponderi pentru această metodă, rezultatele neîmbunătățindu-se semnificativ. În cazul folosirii primei variante de ponderi, regăsirea se îmbunătățește, însă precizia scade, modelul rezultat având un scor f1 de 66% și o acuratețe de 64%.

Iar în cazul celui de al doilea tip de ponderi ale claselor folosite, rezultatele sunt aproape identice cu cele fără ponderi, cea cu ponderi având o regăsire mai mare dar o precizie mai mică, lucru ce a adus la aceeași acuratețe de 72% și scor f1 de 73%.

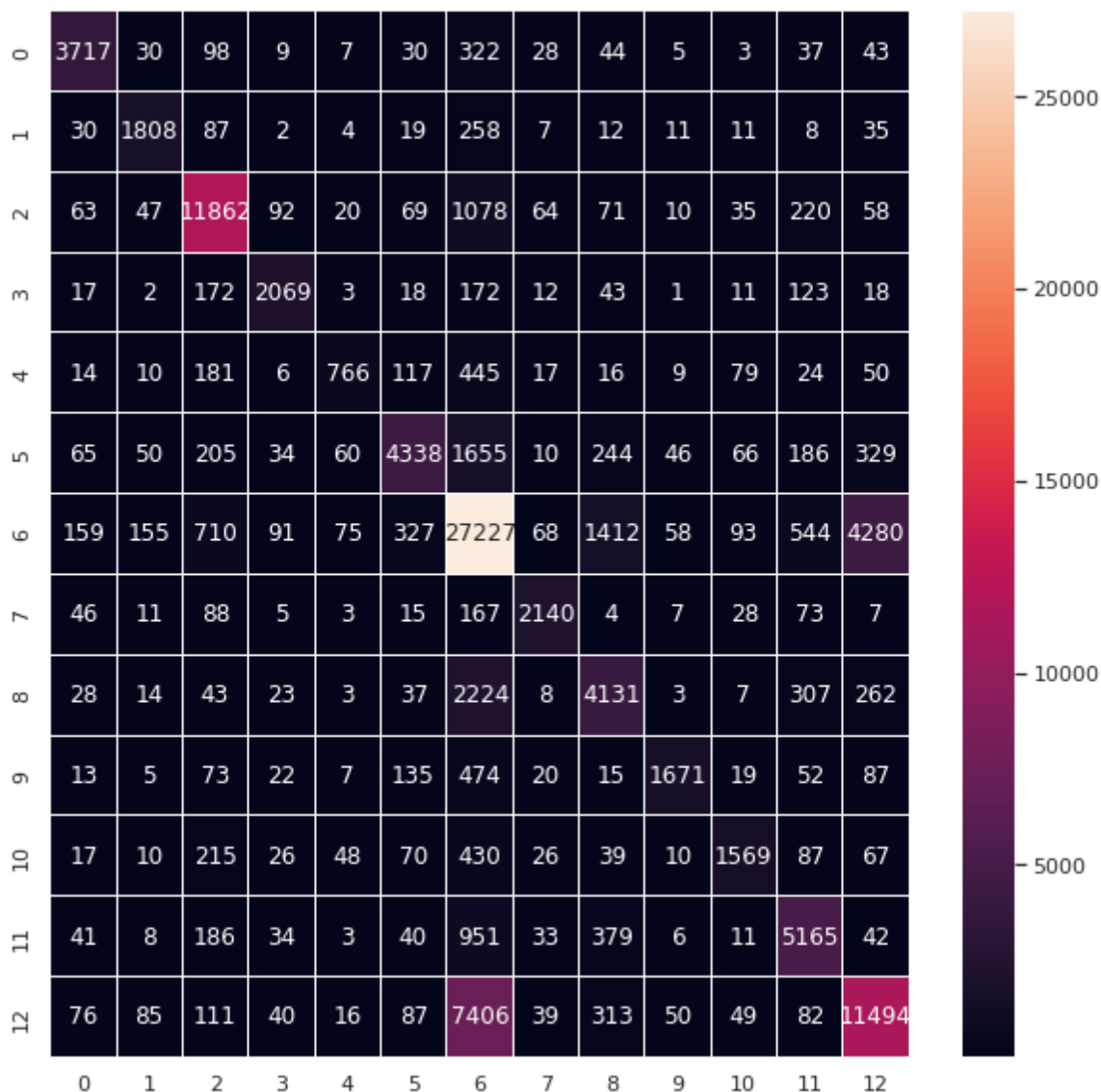


Figură 5.3.4-23 Curbe de învățare LSTM GloVe 300 cuvinte ponderi varianta 2 {12}

	precision	recall	f1-score	support
EDAnonymous	0.86	0.86	0.86	4373
addiction	0.76	0.83	0.80	2292
adhd	0.85	0.86	0.85	13689
autism	0.79	0.81	0.80	2661
bipolarreddit	0.65	0.52	0.58	1734
bpd	0.85	0.57	0.68	7288
depression	0.65	0.75	0.70	35199
healthanxiety	0.85	0.84	0.85	2594
lonely	0.62	0.58	0.60	7090
ptsd	0.82	0.71	0.76	2593
schizophrenia	0.68	0.67	0.68	2614
socialanxiety	0.74	0.75	0.74	6899
suicidewatch	0.67	0.60	0.63	19848
accuracy			0.72	108874
macro avg	0.75	0.72	0.73	108874
weighted avg	0.72	0.72	0.71	108874

Figură 5.3.4-24 Raport de clasificare LSTM GloVe 300 cuvinte ponderi varianta 2 {12}

Cel mai bun model în acest caz poate fi considerat cel ce folosește un număr de 300 de cuvinte ori fără ponderi ori cu ponderile din varianta 2. În figura 5.3.4-25 este matricea de confuzie a modelului ce utilizează 300 cuvinte fără ponderi, cu o acuratețe de 72% și un scor f1 de 73%



Figură 5.3.4-25 Matrice de confuzie LSTM GloVe 300 cuvinte {10}

S-a folosit o augmentare a cuvintelor cu ajutorul librării *nlpaug*<sup>8</sup> folosind mai multe tehnici. O prima variantă este cea folosind substituirea cuvintelor în funcție de sinonimele acestora. Rezultatele acesteia se pot vedea în intrarea 13 din tabelul 5, unde scorurile au scăzut de la o valoare a acurateței de 72% la una de 64%. O altă metodă este augmentarea caracterelor prin substituirea acestora cu caractere eronate în funcție de distanța acestora pe tastatură; însă nici această variantă nu a adus îmbunătățiri clasificării per total, având o acuratețe de 65% după cum se vede în intrarea 14 din tabelul 5.

<sup>8</sup> <https://github.com/makcedward/nlpaug>

Baza de date a influențat scorurile, acuratețea scăzând de la prima bază de date la a doua, însă scorul f1 este mult mai bun în cazul celei de a doua baze de date, acest lucru se datorează diferenței foarte mare între valorile regăsirii și preciziei.

Cel mai bun model rămâne intrarea 4 dacă dorim utilizarea la un număr de 70 cuvinte, sau intrarea 7 cu un număr de 300 cuvinte pentru varianta cu Tokenizarea simplă.

Tabel 3 LSTM

Nr.	Baza de date	Nr. cuv	Tehnica preprocesare	AC	Pond eri	Aug	Acc	P	R	F1
1.	[1]	70	Tokenizare	NU	NU	NU	73%	67%	59%	60%
2.	[1]	70	Tokenizare+ GloVe	NU	NU	NU	70%	63%	54%	56%
3.	[2]	70	Tokenizare	NU	NU	NU	70%	76%	71%	72%
4.	[2]	70	Tokenizare	DA	NU	NU	70%	78%	68%	72%
5.	[2]	70	Tokenizare	DA	DA – var1	NU	62%	62%	72%	64%
6.	[2]	70	Tokenizare	DA	DA – var2	NU	70%	75%	71%	72%
7.	[2]	300	Tokenizare	DA	NU	NU	70%	77%	68%	72%
8.	[2]	70	Tokenizare+ GloVe	NU	NU	NU	69%	74%	66%	69%
9.	[2]	70	Tokenizare+ GloVe	DA	NU	NU	69%	75%	66%	70%
10.	[2]	300	Tokenizare+ GloVe	DA	NU	NU	72%	78%	70%	73%
11.	[2]	300	Tokenizare+ GloVe	DA	DA – var 1	NU	64%	62%	75%	66%
12.	[2]	300	Tokenizare+ GloVe	DA	DA – var 2	NU	72%	75%	72%	73%
13.	[2]	300	Tokenizare+ GloVe	DA	NU	DA-sinonim	64%	62%	75%	66%
14.	[2]	300	Tokenizare+ GloVe	DA	NU	DA-caracter	65%	63%	75%	67%

### 5.3.5. Convolutional Neural Network (CNN)

Pentru construirea acestui model am folosit același API de la keras, și s-a folosit același tip de model secvențial la care s-a adăugat un strat de încorporări la care am adăugat în una din variante matricea de încorporări de la GloVe pe care am descris-o în capitolele 2.2.8 și 5.2.1. Față de modelul cu long short term memory, aici am utilizat un strat de *Conv1D* specific acestui tip de model CNN. S-a utilizat optimizatorul Adam cu o rată de învățare de 0.001 și ca funcții de callback: Early Stopping și ReduceLR. Ce ne ajută la modificarea ratei de învățare în timp și oprirea automată a antrenării în cazul în care scorurile nu se îmbunătățesc. La ambele am folosit o răbdare („*patience*”) de 5 epoci și un *min\_delta* de 0.01.

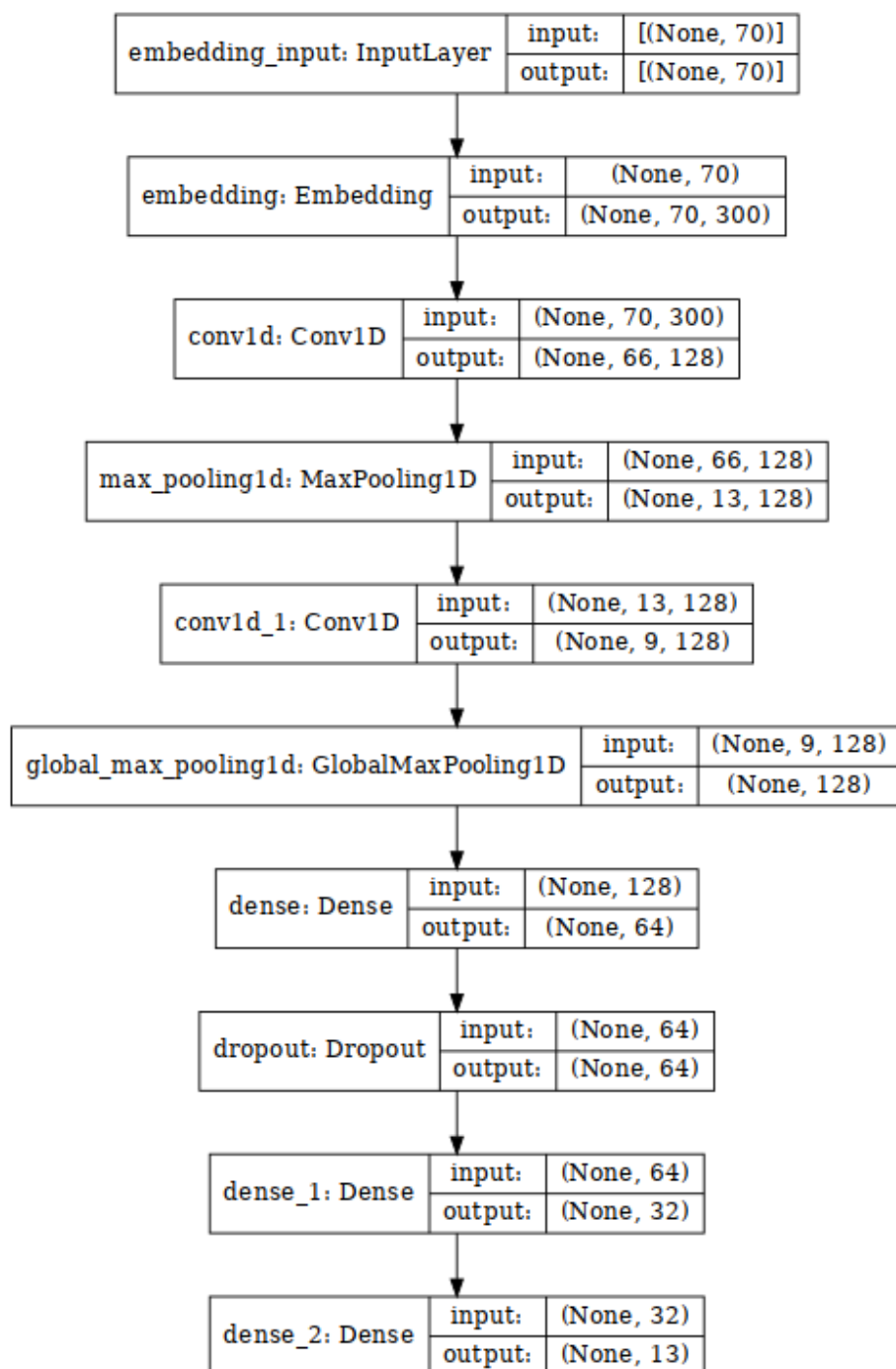
```
model1 = Sequential([
    Embedding(vocab_size, EMBEDDING_DIM, trainable=False, input_length=MAX_SEQUENCE_LENGTH),
    Conv1D(filters=128, kernel_size=5, activation='relu'),
    MaxPooling1D(5),
    Conv1D(128, 5, activation='relu'),
    GlobalMaxPool1D(),
    Dense(64, activation='relu'),
    Dropout(0.5),
    Dense(32, activation='relu'),
    Dense(number_of_labels, activation='softmax')
])
```

Figură 5.3.5-1 Secvență de cod pentru modelul CNN

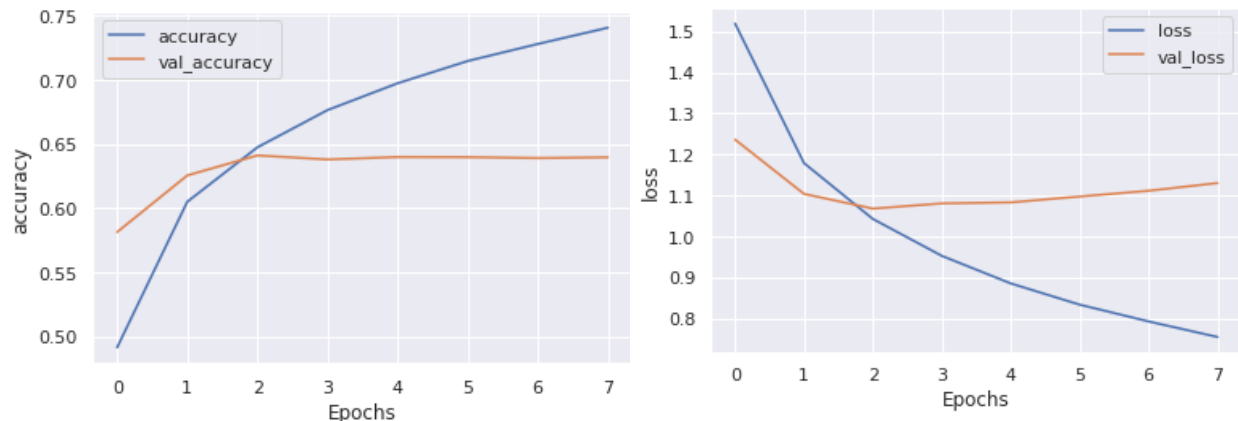
Stratul de încorporări primește ca intrare secvențe de 70 cuvinte ce au fost transformate în date numerice și întoarce o matrice de dimensiune 70, 300; unde 70 e numărul de elemente din vectorul primit și 300 e dimensiunea încorporărilor. Această ieșire este apoi introdusă în stratul convoluțional cu 128 de filtre, ce face calculele matematice necesare pentru a extrage caracteristicile relevante din text. Stratul de *maxpooling* cu dimensiunea ferestrei de 5 este adăugat pentru a reduce dimensiunea dar fără să piardă din caracteristicile necesare și este des folosit după straturile convoluționale. Un alt strat convoluțional este iar adăugat cu același număr de filtre. Stratul final ce transformă aceste caracteristici din (None, 9, 128) într-un număr de forma (None, x) este stratul *globalmaxpooling1d* ce ia valoarea maximă din întreaga secvență convoluțională, nu doar din fiecare fereastră. Următoarele straturi de dens și abandon următoare ne ajută să învățăm și în același timp să ajungem la dimensiunea dorită la finalul antrenării, și anume 13, numărul de clase.

Din cauza rezultatelor mai slabe față de modelul de LSTM, s-a realizat o comparație doar între cele două variante de procesare a textului, și anume varianta cu tokenizarea textului și varianta ce cuprinde și încorporările GloVe, fără de a ține cont de cele două baze de date.





Figură 5.3.5-2 Vizualizare model CNN

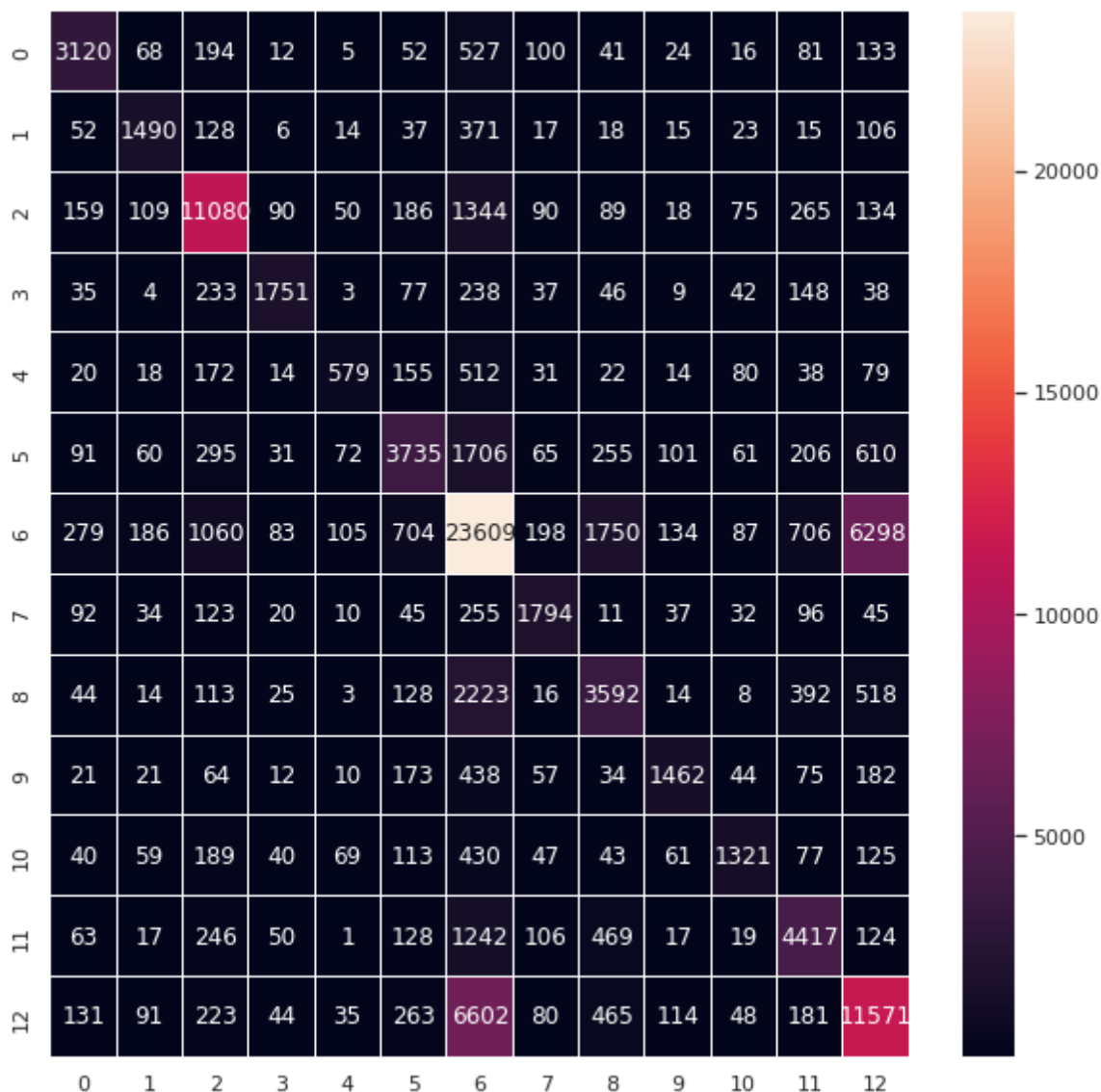


Figură 5.3.5-3 Curbe de învățare CNN – Tokenizare {1}

Se observă că modelul suferă de supraînvățare în cazul datelor de antrenare, dar nu și în cazul datelor de validare, scorurile acestora fiind constante de la epoca 2. În cazul datelor de test acuratețea se menține la 64% ca și la validare. Modelul nostru comportându-se bine în raport cu acest proces de antrenare-validare-testare.

	precision	recall	f1-score	support
EDAnonymous	0.75	0.71	0.73	4373
addiction	0.69	0.65	0.67	2292
adhd	0.78	0.81	0.80	13689
autism	0.80	0.66	0.72	2661
bipolarreddit	0.61	0.33	0.43	1734
bpd	0.64	0.51	0.57	7288
depression	0.60	0.67	0.63	35199
healthanxiety	0.68	0.69	0.69	2594
lonely	0.53	0.51	0.52	7090
ptsd	0.72	0.56	0.63	2593
schizophrenia	0.71	0.51	0.59	2614
socialanxiety	0.66	0.64	0.65	6899
suicidewatch	0.58	0.58	0.58	19848
accuracy			0.64	108874
macro avg	0.67	0.60	0.63	108874
weighted avg	0.64	0.64	0.64	108874

Figură 5.3.5-4 Raport de clasificare CNN – Tokenizare {1}

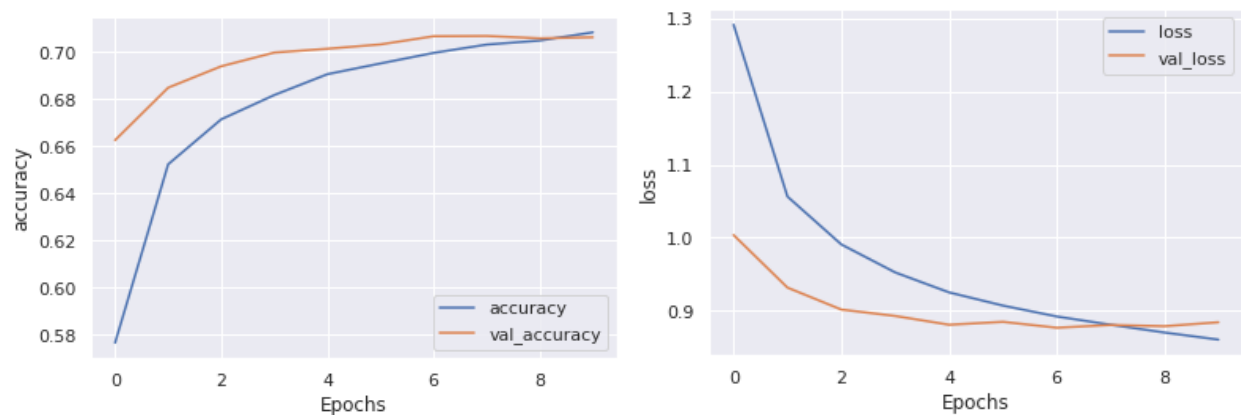


Figură 5.3.5-5 Matrice de confuzie CNN – Tokenizare {1}

Pentru varianta cu tokenizare simplă a cuvintelor s-a obținut o acuratețe de 64%, precizia cea mai mare a fost la calculul *macro*, având o valoare de 67%, regăsirea l-a avut calculul *weighted* cu valoarea de 64%, iar ca rezultat al lor, f1-score a fost mai mare la varianta *weighted* cu 64%. Acesta a avut o tendință de supraînvățare dar s-a oprit datorită optimizatorilor de oprire automată, și se observă o problemă la eticheta cu numărul 6, „depression”, unde un procent destul de mare din restul bolilor sunt confundate cu aceasta.

În cazul variantei cu încorporări, rezultatele sunt mai puțin bune decât în cazul utilizării de tokenizări simple, având o acuratețe de 70% și un scor f1 de 72%

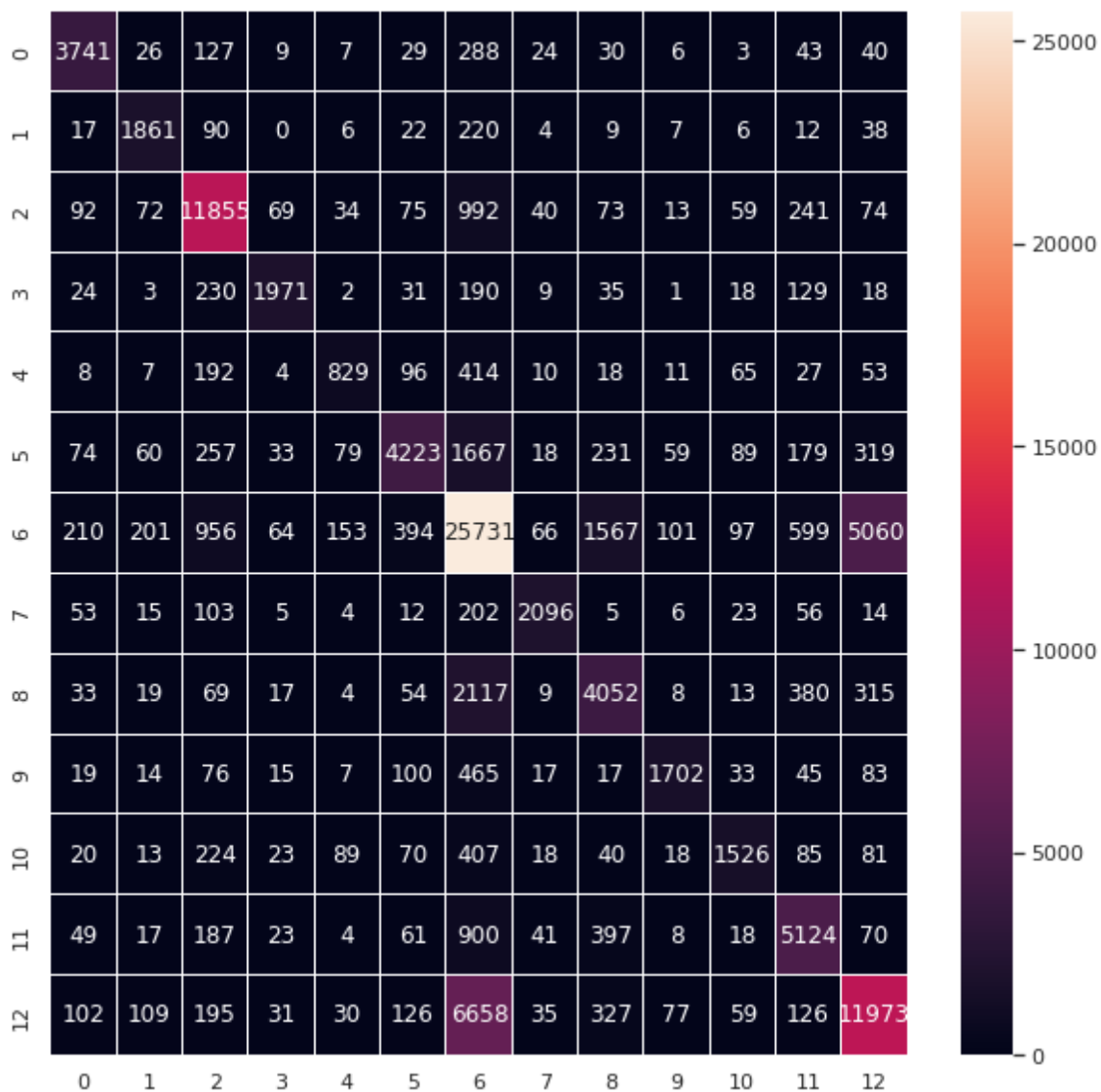
fără ponder. În cazul ponderilor, rezultatele scad la 69% în cazul acurateței și la 71% în cazul scorului f1.



Figură 5.3.5-6 Curbe de învățare CNN GloVe [2]

	precision	recall	f1-score	support
EDAnonymous	0.84	0.86	0.85	4373
addiction	0.77	0.81	0.79	2292
adhd	0.81	0.87	0.84	13689
autism	0.87	0.74	0.80	2661
bipolarreddit	0.66	0.48	0.56	1734
bpd	0.80	0.58	0.67	7288
depression	0.64	0.73	0.68	35199
healthanxiety	0.88	0.81	0.84	2594
lonely	0.60	0.57	0.58	7090
ptsd	0.84	0.66	0.74	2593
schizophrenia	0.76	0.58	0.66	2614
socialanxiety	0.73	0.74	0.73	6899
suicidewatch	0.66	0.60	0.63	19848
accuracy			0.70	108874
macro avg	0.76	0.69	0.72	108874
weighted avg	0.71	0.70	0.70	108874

Figură 5.3.5-7 Raport de clasificare CNN GloVe [2]



Figură 5.3.5-8 Matrice de confuzie CNN GloVe [2]

Tabel 4 CNN

Nr.	Baza de date	Nr. cuvinte tokenizare	Tehnica preprocesare	AC	Ponderi	Acc	P	R	F1
1.	[2]	70	Tokenizare	NU	NU	64%	67%	60%	63%
2.	[2]	70	Tokenizare+ GloVe	NU	NU	70%	76%	69%	72%
3.	[2]	70	Tokenizare+ GloVe	NU	DA – var 2	69%	74%	67%	70%

Cea mai bună variantă ce utilizează modelul convoluțional este cea cu încorporările GloVe ce are o valoare a acurateței de 70% și un scor f1 de 72%.

### 5.3.6. Bidirectional Encoder Representations from Transformers (BERT)

BERT este bazat pe transformeri, un model neuronal complex unde fiecare element de ieșire este conectat la fiecare element de intrare. Acesta a fost antrenat pe un corp foarte mare de date, făcând astfel preziceri mai bune în cazul clasificărilor decât alți clasificatori.

Model: "tf\_bert\_for\_sequence\_classification"

Layer (type)	Output Shape	Param #
bert (TFBertMainLayer)	multiple	109482240
dropout_37 (Dropout)	multiple	0
classifier (Dense)	multiple	9997
Total params: 109,492,237		
Trainable params: 109,492,237		
Non-trainable params: 0		

Figură 5.3.6-1 Sumar model BERT

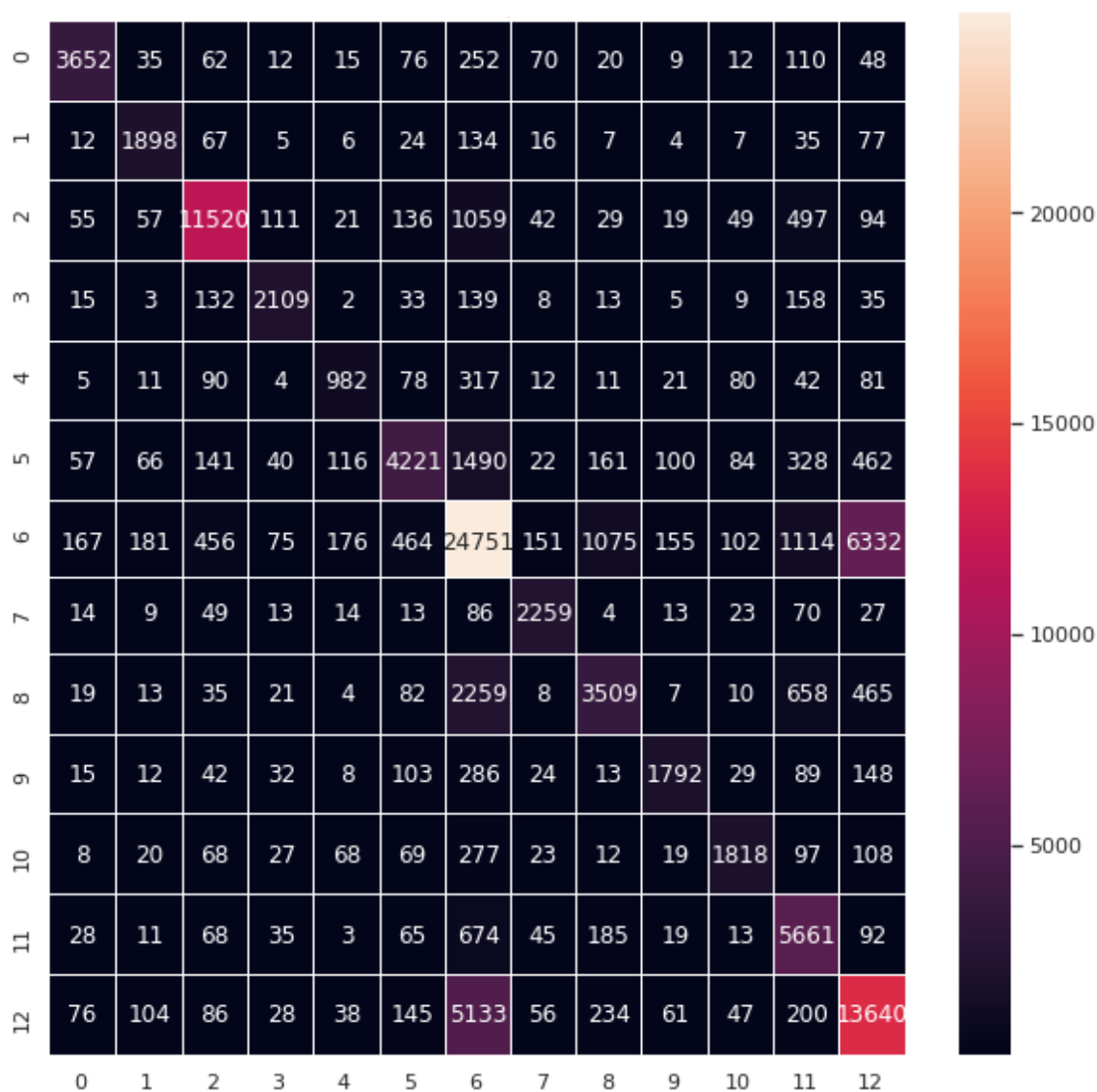
Corpusul default al modelului este de 512 de cuvinte per propoziție, noi totuși am folosit un nr de 64 de cuvinte pentru a reduce memoria utilizată și pentru o comparație mai bună față de modele cu 70 cuvinte. Scorurile sunt în jurul la o acuratețe de 71% atât înainte de corectarea textului cât și după, dar și în cazul utilizării de ponderi.

Față de articolele citate până acum, [3], BERT a avut un rezultat mai bun folosind 512 cuvinte în procesul de reantrenare față de modelul ce a utilizat LSTM, la noi însă, rezultatele cu 64 de cuvinte nu sunt așa de bune față de cel mai bun model al nostru ce utilizează LSTM, lucru datorat unui numărului mic de cuvinte.

	precision	recall	f1-score	support
EDAnonymous	0.89	0.84	0.86	4373
addiction	0.78	0.83	0.81	2292
adhd	0.90	0.84	0.87	13689
autism	0.84	0.79	0.82	2661
bipolarreddit	0.68	0.57	0.62	1734
bpd	0.77	0.58	0.66	7288
depression	0.67	0.70	0.69	35199
healthanxiety	0.83	0.87	0.85	2594
lonely	0.67	0.49	0.57	7090
ptsd	0.81	0.69	0.74	2593
schizophrenia	0.80	0.70	0.74	2614
socialanxiety	0.62	0.82	0.71	6899
suicidewatch	0.63	0.69	0.66	19848
accuracy			0.71	108874
macro avg	0.76	0.72	0.74	108874
weighted avg	0.72	0.71	0.71	108874

Figură 5.3.6-2 Raport clasificare BERT - base-uncased {2}

Se observă ca modelul încă încurcă clasele „depression” cu „suicidewatch” și viceversa sau „lonely” cu „depression”. Acest lucru este datorat similitudinii dintre aceste afecțiuni și ale simptomelor acestora.



Figură 5.3.6-3 Matrice de confuzie BERT base-uncased {2}

În cazul utilizării unui număr de 128 de cuvinte în procesul de tokenizare, avem rezultate ceva mai bune. Valoarea acurateței a crescut de la 71% la 72% iar scorul f1 de la 74% la 75%. Ajungem la concluzia că folosind un număr mai mare de cuvinte în procesul de tokenizare, rezultatele o să fie mai bune. Antrenarea se realizează în 10 epoci și se stabilizează după 5 epoci după cum se observă în figură.



# NECLASIFICAT

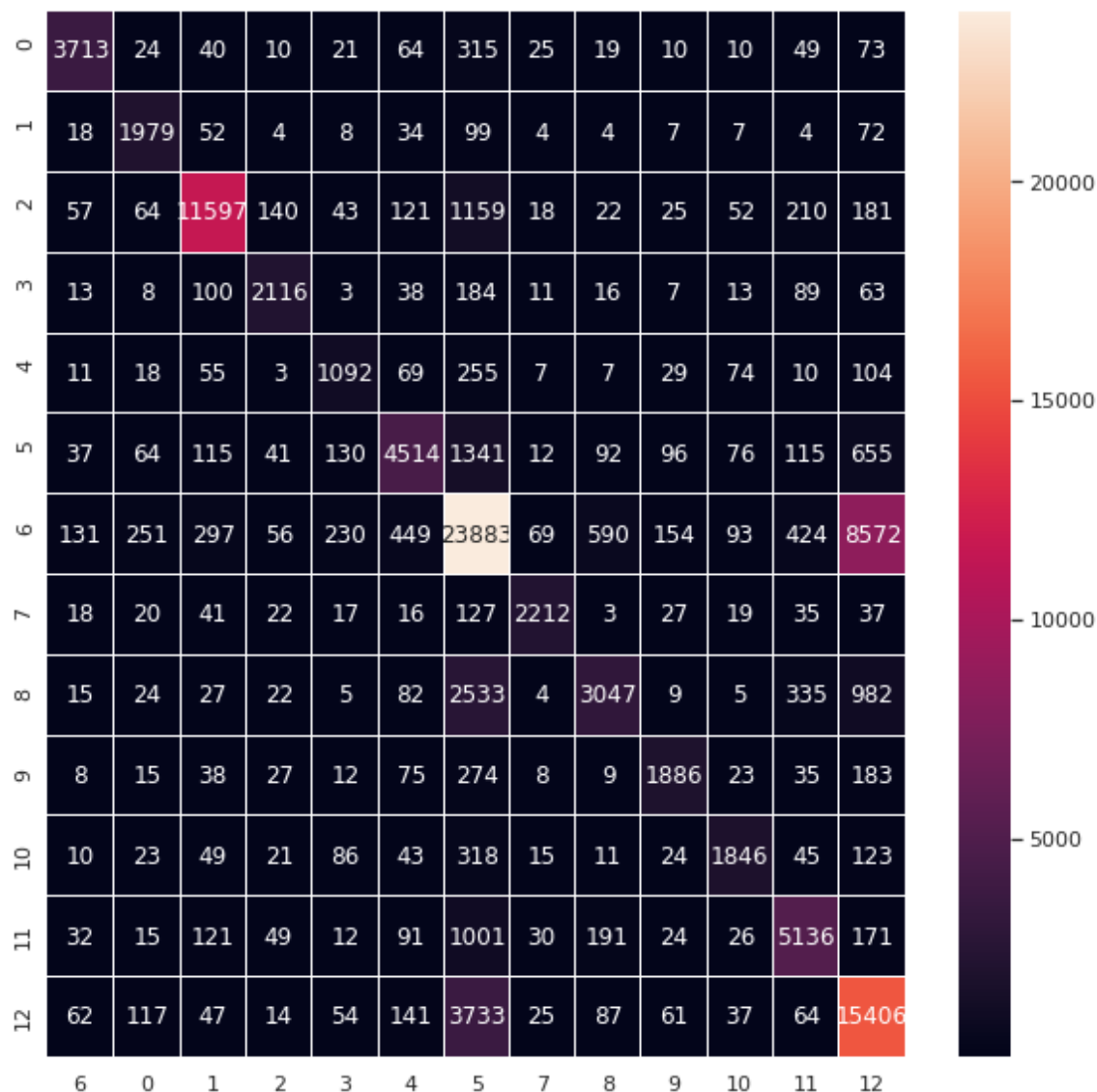


Figură 5.3.6-4 Curbe de învățare BERT - base-uncased {3}

	precision	recall	f1-score	support
EDAnonymous	0.90	0.85	0.87	4373
addiction	0.75	0.86	0.81	2292
adhd	0.92	0.85	0.88	13689
autism	0.84	0.80	0.82	2661
bipolarreddit	0.64	0.63	0.63	1734
bpd	0.79	0.62	0.69	7288
depression	0.68	0.68	0.68	35199
healthanxiety	0.91	0.85	0.88	2594
lonely	0.74	0.43	0.54	7090
ptsd	0.80	0.73	0.76	2593
schizophrenia	0.81	0.71	0.75	2614
socialanxiety	0.78	0.74	0.76	6899
suicidewatch	0.58	0.78	0.66	19848
accuracy			0.72	108874
macro avg	0.78	0.73	0.75	108874
weighted avg	0.73	0.72	0.72	108874

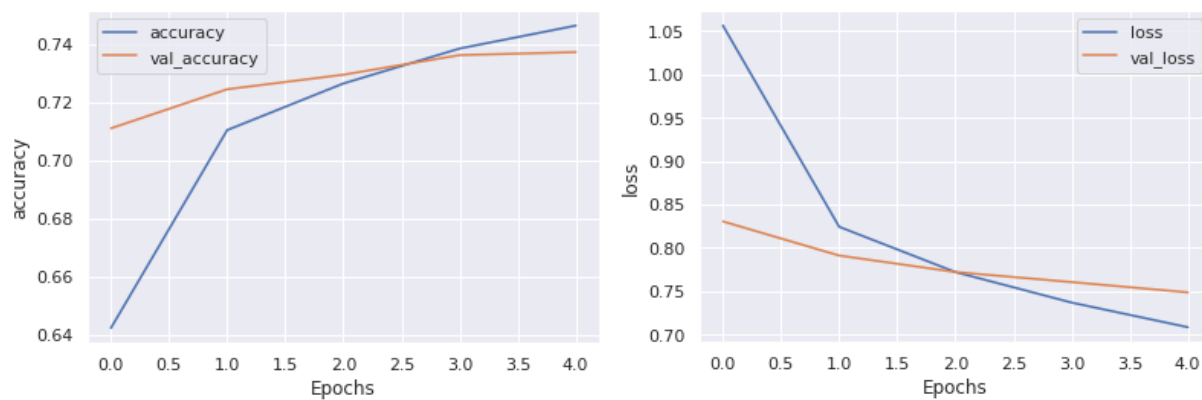
Figură 5.3.6-5 Raport clasificare BERT - base-uncased {3}

# NECLASIFICAT



Figură 5.3.6-6 Matrice de confuzie BERT - base-uncased {3}

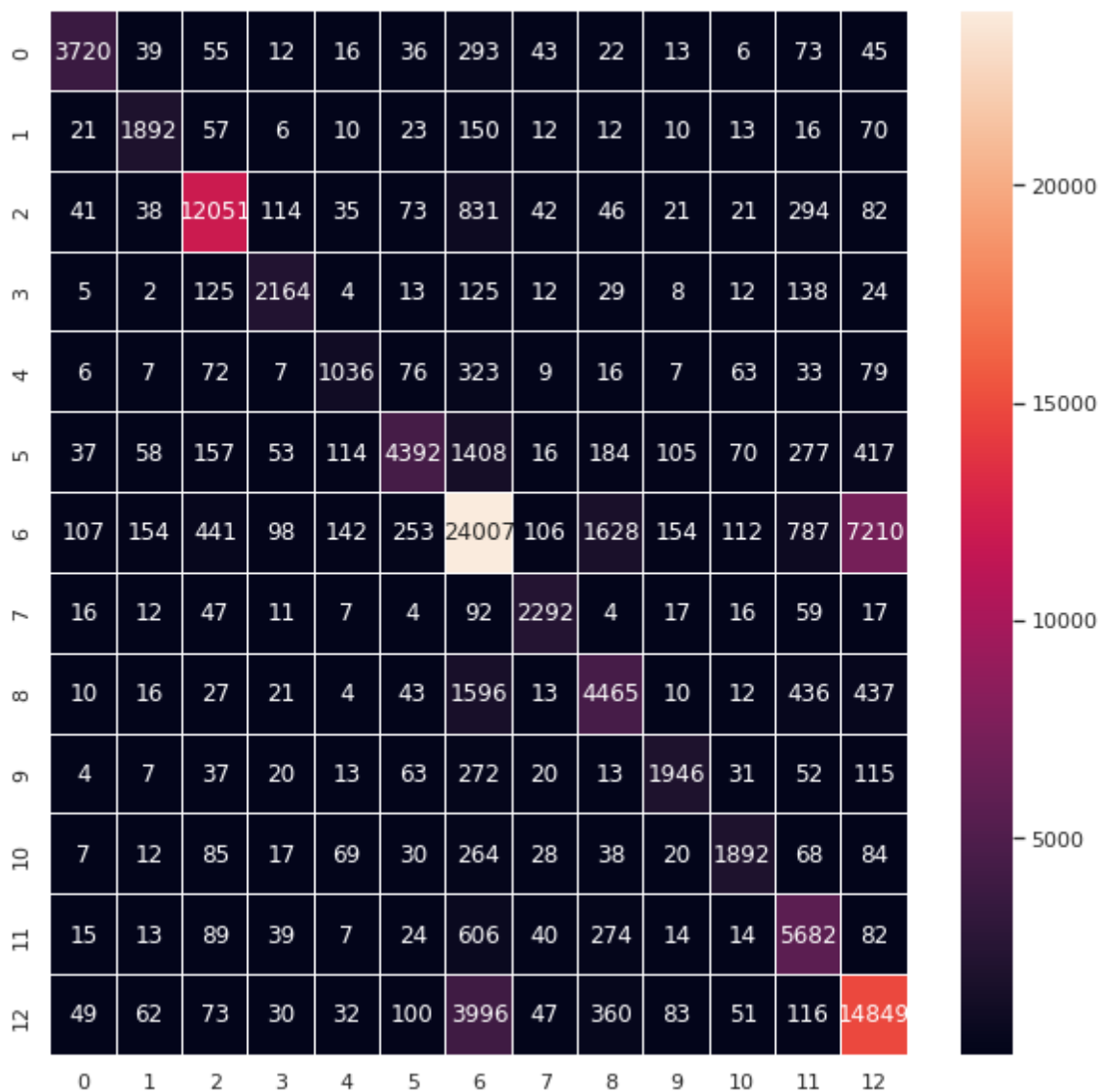
Mai există varianta de Mental BERT, un model BERT reatrenat pe date legate de afecțiuni mentale. Folosind un număr de 100 de cuvinte, acesta a avut un scor mai bun față de varianta obișnuită a BERT-ului, având o valoare a acurateței de 74% și un scor f1 de 76%. Rezultatele folosind un număr de 128 de cuvinte este aproape identic. Acest lucru se datorează reantrenării cu date specifice subiectului lucrării. Până acum acesta este cel mai bun model pentru clasificarea multi-clasă.



Figură 5.3.6-7 Curbe de învățare Mental BERT {9}

	precision	recall	f1-score	support
0	0.92	0.85	0.88	4373
1	0.82	0.83	0.82	2292
2	0.91	0.88	0.89	13689
3	0.83	0.81	0.82	2661
4	0.70	0.60	0.64	1734
5	0.86	0.60	0.71	7288
6	0.71	0.68	0.69	35199
7	0.86	0.88	0.87	2594
8	0.63	0.63	0.63	7090
9	0.81	0.75	0.78	2593
10	0.82	0.72	0.77	2614
11	0.71	0.82	0.76	6899
12	0.63	0.75	0.68	19848
accuracy			0.74	108874
macro avg	0.78	0.75	0.77	108874
weighted avg	0.75	0.74	0.74	108874

Figură 5.3.6-8 Raport clasificare Mental BERT {9}



Figură 5.3.6-9 Matrice de confuzie Mental BERT [9]

Tabel 5 Bert

Nr.	Model	Nr. cuvinte tokenizare	AC	Ponderi	Acc	P	R	F1
1.	bert-base-uncased	64	NU	NU	71%	75%	71%	73%
2.	bert-base-uncased	64	DA	NU	71%	76%	72%	74%
3.	bert-base-uncased	128	DA	NU	72%	78%	73%	75%

4.	bert-base-uncased	64	NU	DA – var 2	71%	74%	73%	73%
5.	bert-base-uncased	64	DA	DA – var 2	71%	73%	73%	72%
6.	bert-base-uncased	64	DA	DA – var 3	70%	72%	75%	73%
7.	mental /mental-bert-base-uncased	64	DA	NU	73%	77%	75%	76%
8.	mental /mental-bert-base-uncased	100	DA	NU	74%	79%	75%	76%
9.	<b>mental /mental-bert-base-uncased</b>	<b>128</b>	<b>DA</b>	<b>NU</b>	<b>74%</b>	<b>78%</b>	<b>75%</b>	<b>77%</b>

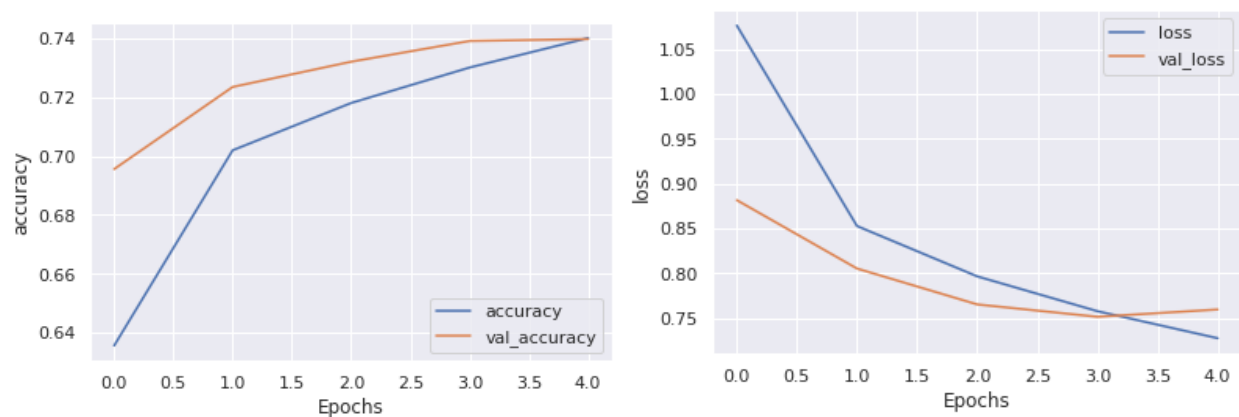
Cea mai bună variantă a transformerului BERT este cea folosind un număr de 128 de cuvinte, intrarea 3 având o valoare a acurateții de 72% și un scor f1 de 75%. Folosind metoda Mental BERT, tot cu un număr de 128 de cuvinte, s-a ajuns la o valoare a acurateții de 74% și de scor f1 de 77%.

### 5.3.7. Robustly Optimized BERT (RoBERTa)

RoBERTa este un model optimizat pe baza la BERT care aduce rezultate mai bune în majoritatea cazurilor. Acesta a fost antrenat pe o bază de date mult mai mare și s-au folosit diverse tehnici de antrenare mai eficiente. În prezenta lucrare s-a făcut un studiu pentru acest tip de model preantrenat, atât cu varianta de bază cât și cu varianta Mental RoBERTa. Mental RoBERTa este modelul de RoBERTa antrenat pe baza de date cu specific în acest domeniu. Rezultatele sunt asemănătoare cu cele date de modelul BERT simplu, dar cu o ușoară îmbunătățire. Cu un număr de 100 de cuvinte, se comportă aproape la fel de bine ca BERT cu 128 de cuvinte, astfel putem avea concluzia că și acest model s-ar comporta mult mai bine cu un număr de 512 cuvinte, cât este corpusul de la antrenarea inițială.

Se observă cum valoarea de abandon influențează rezultatele, fiind nevoie de mai multe rulări pentru a ajunge la rezultatele dorite. Utilizând un număr de 100 de cuvinte și o valoare de abandon de 0.4 am obținut o valoare a acurateții de 72% și

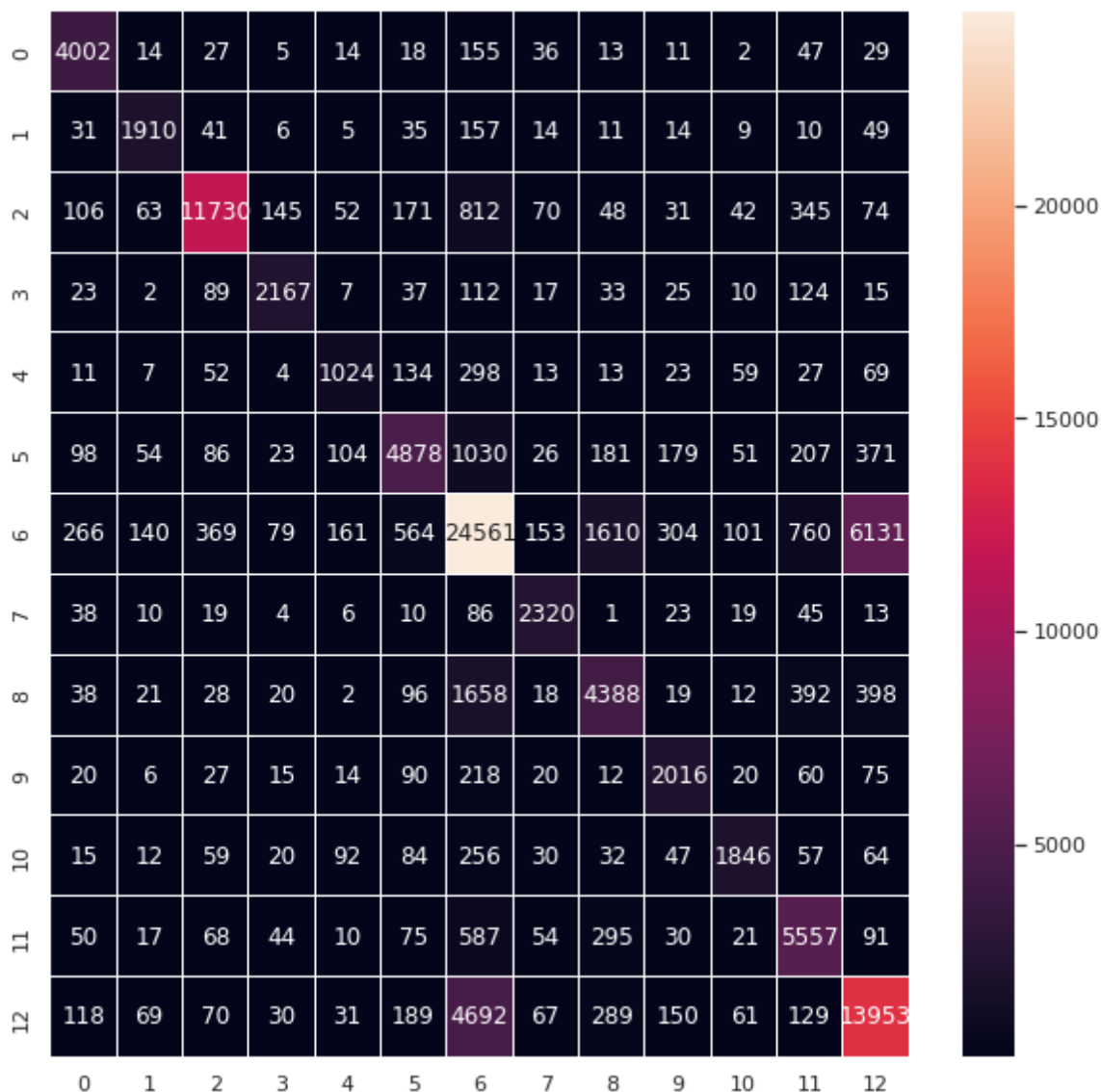
un scor f1 de 74%, aproape la fel de bun ca modelul BERT utilizând un număr de 128 de cuvinte. Schimbând numărul de cuvinte la 128 și valoarea de abandon la 0.3 am ajuns la o valoare de 74% și un scor f1 de 76%.



Figură 5.3.7-1 Curbe de învățare RoBERTa {3}

	precision	recall	f1-score	support
EDAnonymous	0.83	0.92	0.87	4373
addiction	0.82	0.83	0.83	2292
adhd	0.93	0.86	0.89	13689
autism	0.85	0.81	0.83	2661
bipolarreddit	0.67	0.59	0.63	1734
bpd	0.76	0.67	0.71	7288
depression	0.71	0.70	0.70	35199
healthanxiety	0.82	0.89	0.85	2594
lonely	0.63	0.62	0.63	7090
ptsd	0.70	0.78	0.74	2593
schizophrenia	0.82	0.71	0.76	2614
socialanxiety	0.72	0.81	0.76	6899
suicidewatch	0.65	0.70	0.68	19848
accuracy			0.74	108874
macro avg	0.76	0.76	0.76	108874
weighted avg	0.74	0.74	0.74	108874

Figură 5.3.7-2 Raport clasificare RoBERTa {3}



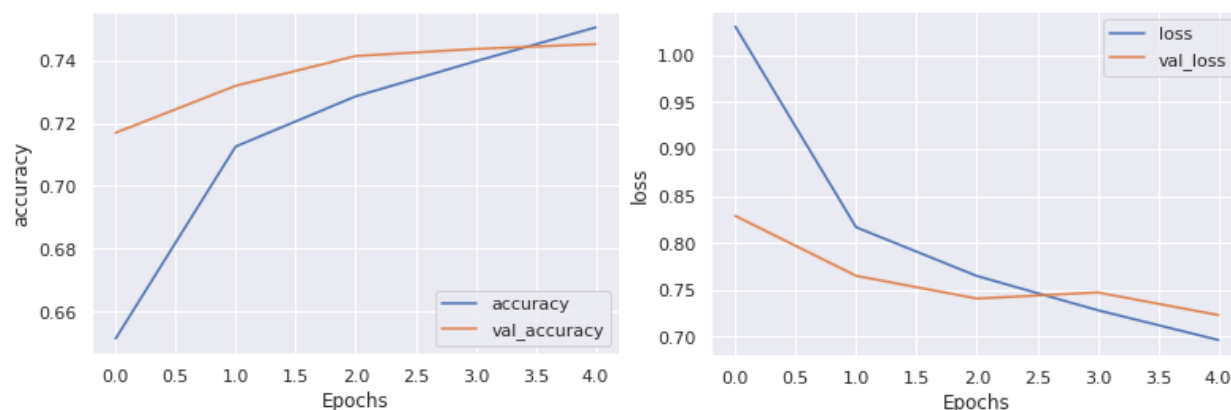
Figură 5.3.7-3 Matrice de confuzie RoBERTa {3}

Modelul încă încurcă etichete precum „*depression*” și „*suicidewatch*” sau „*depression*” cu „*lonely*” din aceeași cauză de asemănare a informațiilor și a simptomelor.

Folosind o valoare de abandon mai mare, de 0.5 se observă că scorurile scad destul de mult, la același număr de epoci, ajungând la concluzia că această valoare contează foarte mult în timpul procesului de antrenare. Astfel, s-au încercat mai multe valori pentru a ajunge la cea mai potrivită în cazul acestui tip de model.

Pentru utilizarea de Mental RoBERTa, construit pe același principiu ca Mental BERT, am folosit o valoare de abandon de 0.5 și un număr de 128 de cuvinte, cu rezultate de 71% în cadrul acurateței și de 73% în cazul scorului f1.

Comparând cu intrarea 2 din tabel, având aceași valoare de abandon și același număr de cuvinte, scorul s-a îmbunătățit cu 2%. Pentru a demonstra diferențele dintre RoBERTa și Mental RoBERTa, am folosit și o valoare de 0.3 la abandon, având rezultate de 74% la acuratețe și 77% la scorul f1, fiind cele mai bune rezultate de până acum.

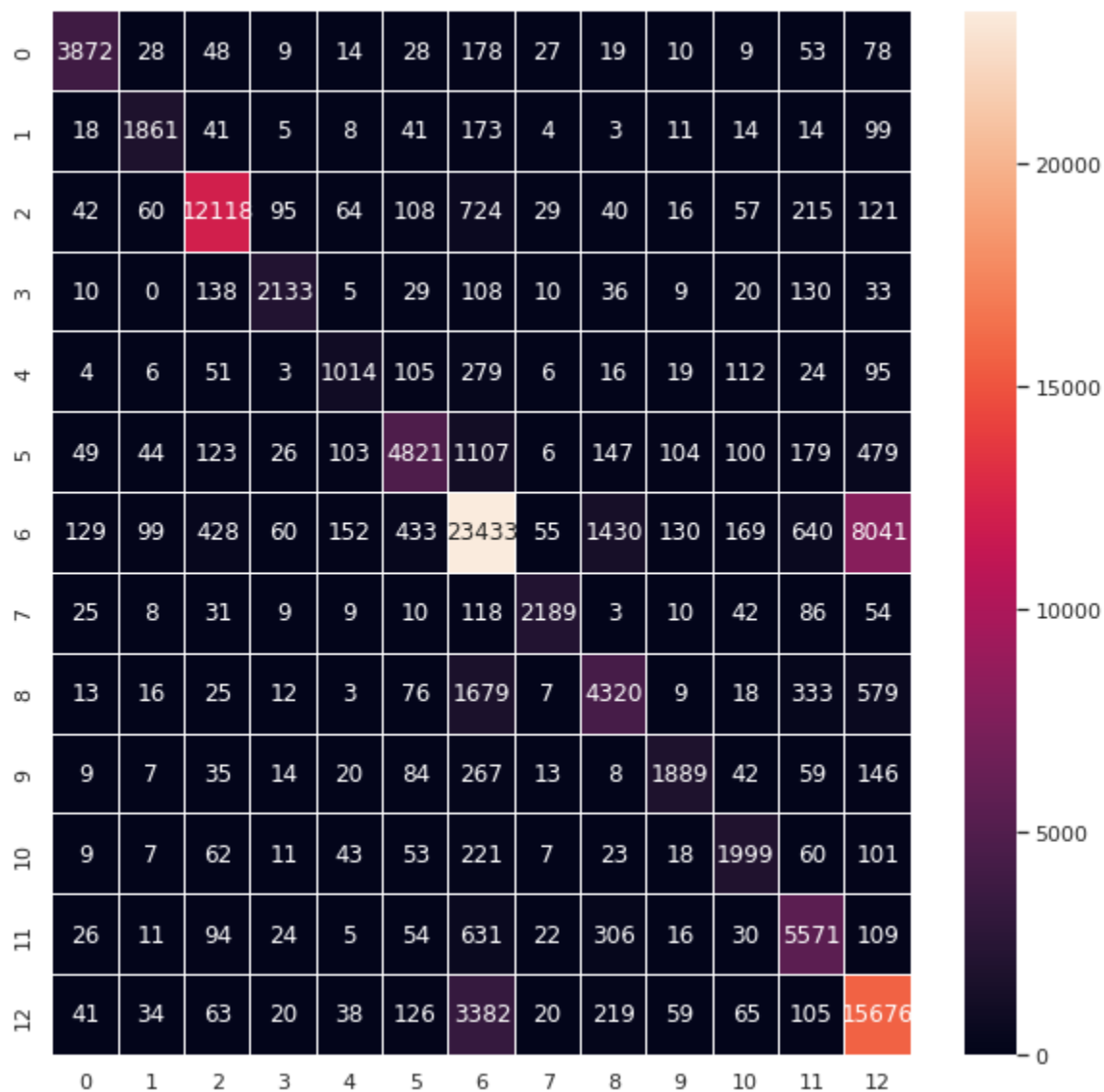


Figură 5.3.7-4 Curbe de învățare Mental RoBERTa {5}

	precision	recall	f1-score	support
EDAnonymous	0.91	0.89	0.90	4373
addiction	0.85	0.81	0.83	2292
adhd	0.91	0.89	0.90	13689
autism	0.88	0.80	0.84	2661
bipolarreddit	0.69	0.58	0.63	1734
bpd	0.81	0.66	0.73	7288
depression	0.73	0.67	0.69	35199
healthanxiety	0.91	0.84	0.88	2594
lonely	0.66	0.61	0.63	7090
ptsd	0.82	0.73	0.77	2593
schizophrenia	0.75	0.76	0.76	2614
socialanxiety	0.75	0.81	0.78	6899
suicidewatch	0.61	0.79	0.69	19848
accuracy			0.74	108874
macro avg	0.79	0.76	0.77	108874
weighted avg	0.75	0.74	0.74	108874

Figură 5.3.7-5 Raport de clasificare Mental RoBERTa {5}





Figură 5.3.7-6 Matrice de confuzie Mental RoBERTa [5]

Se observă în matricea de confuzie o îmbunătățire totală dar încă există o confundare între clasele „*depression*” și „*suicidewatch*” din cauza asocierii între cele două afecțiuni.

Tabel 6 RoBERTa

Nr.	Model	Nr. cuvinte tokenizare	Valoare Abandon	Acc	P	R	F1
1.	Roberta-base	100	0.4	72%	76%	73%	74%
2.	Roberta-base	128	0.5	69%	73%	70%	71%

<b>3.</b>	<b>Roberta-base</b>	<b>128</b>	<b>0.3</b>	<b>74%</b>	<b>76%</b>	<b>76%</b>	<b>76%</b>
4.	Mental- roberta-base	128	0.5	71%	76%	72%	73%
<b>5.</b>	<b>Mental- roberta-base</b>	<b>128</b>	<b>0.3</b>	<b>74%</b>	<b>79%</b>	<b>76%</b>	<b>77%</b>

## 5.4. Rezultate și comparații

Dintre toți algoritmi, toate modelele și variantele folosite în cadrul antrenării acestora, am selectat cea mai bună variantă de la fiecare pentru a realiza o comparație amănunțită a eficienței utilizării fiecăruia dintre ele.

Algoritmul Multinomial Naive Bayes a fost cel mai slab cu un scor f1 de 56%, acesta a reușit să identifice corect unele intrări în baza de date, dar cu un număr de date neechilibrat ca al nostru, aceste rezultate nu sunt promițătoare.

Rețeaua neuronală convoluțională s-a comportat decent folosind simpla tokenizare a cuvintelor, cu un scor f1 de 63%, însă performanțele sale în cadrul acestui tip de date nu a fost cel mai bun. Folosind și încorporările GloVe în cadrul antrenării și preprocesării datelor, acesta a avut un scor f1 de 72%, fiind la fel de bun ca modelul LSTM.

Algoritmul Random Forest, cu o foarte bună precizie de 82%, a avut un scor f1 de 68%, mai bun chiar decât rețeaua CNN anterioară.

Modelul Long Short Term Memory ajunge să fie cel mai detaliat din această lucrare. Scorurile sale bune în raport cu timpul de antrenare a făcut ca acest model să fie cel mai testat cu diverse tehnici, cum ar fi folosirea ponderilor, numărului de cuvinte de la tokenizare dar și utilizării augmentării. Cu simpla variantă de utilizare doar a tokenizării, acesta ajunge la un scor f1 de 71% iar utilizând și încorporările GloVe ajunge la scorul f1 de 73%.

Transformerul BERT ridică miza, având un scor f1 de 75%, acesta a fost testat utilizând un număr diferit de cuvinte dar și de ponderi, nici o variantă nereușind creșterea mai mare a rezultatelor.

Mental BERT a ajuns la un scor f1 de 77% , depășind puțin astfel modelul BERT simplu.

RoBERTa se descurcă aproape la fel de bine ca și Mental BERT, cu o valoare a acurateței de 74% și un scor f1 de 76%.

Mental RoBERTa are rezultatele cele mai bune, cu o acuratețe de 74% și un scor f1 de 77%.

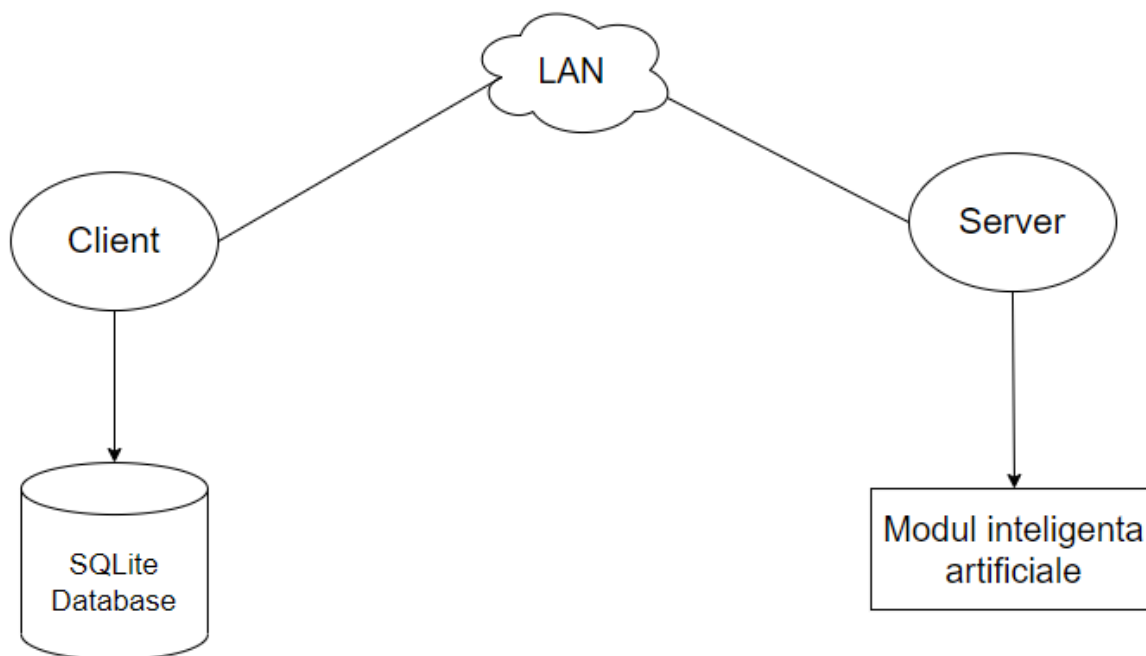
În final, modelul RoBERTa este cel cu cele mai bune rezultate în clasificarea multi-clasă.

*Tabel 7 Final*

<b>Model</b>	<b>Tehnică de procesare</b>	<b>Acc</b>	<b>P</b>	<b>R</b>	<b>F1</b>
Multinomial Naive Bayes	TFIDF	59%	81%	46%	56%
CNN	Tokenizare	64%	67%	60%	63%
Random Forest	TFIDF	68%	82%	61%	68%
LSTM	Tokenizare	70%	77%	67%	71%
CNN	Tokenizare+ GloVe	70%	76%	69%	72%
LSTM	Tokenizare+ GloVe	72%	78%	70%	73%
BERT	BERT Tokenizer	72%	78%	73%	75%
Mental BERT	BERT Tokenizer	74%	78%	75%	77%
RoBERTa	RoBERTa Tokenizer	74%	76%	76%	76%
<b>Mental RoBERTa</b>	<b>RoBERTa Tokenizer</b>	<b>74%</b>	<b>79%</b>	<b>76%</b>	<b>77%</b>

## 6. Mentol Cloud – Aplicație mobilă de integrare

Pentru aplicația de integrare se va folosi o arhitectură client-server în LAN, cu o bază de date SQLite locală folosind REST și cu modulul de inteligență artificială în server. O vizualizare se poate observa în figura alăturată.



Figură 5.3.7-1 Arhitectură aplicație

### 6.1. Cerințe software

#### 6.1.1. Cerințe funcționale

- Aplicația va consta într-o aplicație pe mobil, simplă și intuitivă.
- Nu va exista înregistrare, aplicație fiind offline.
- Introducerea textului se va face în limba engleză.
- Datele primite de server de la client o să fie textul, iar acesta va returna rezultatul.
- Datele se vor stoca într-o bază de date cu un singur tabel.
- Datele stocate vor fi textul, rezultatul de la server și marcajul temporal.
- Modelele neuronale vor avea cea mai mare acuratețe ce a putut fi obținută.
- Predicția datelor se va realiza cu un model de tip clasificare binară ce va depista dacă utilizatorul suferă sau nu de o boală mintală, și un clasificator multiclăsă ce va clasifica probabilitatea fiecărei boli.

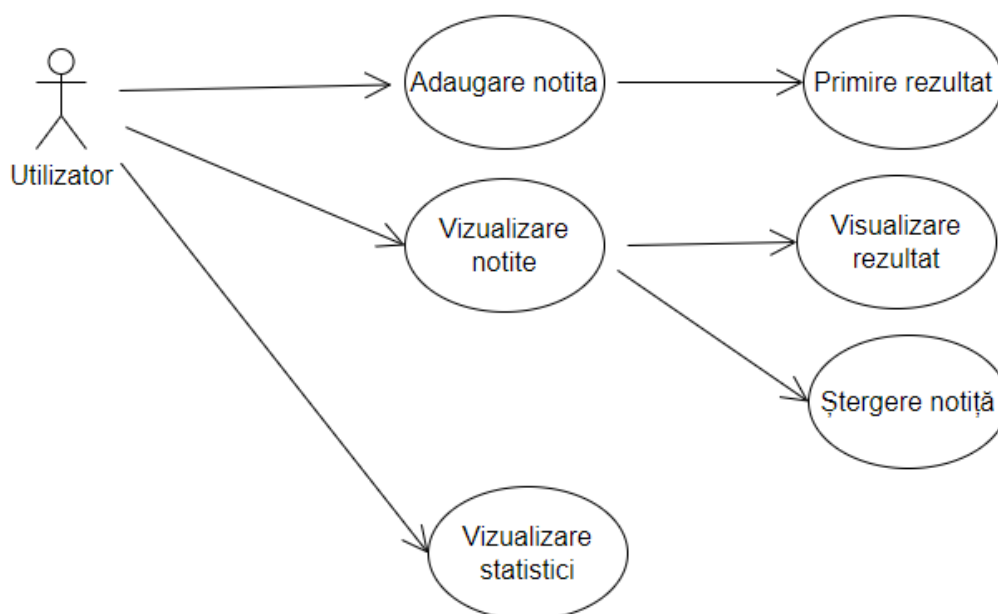
- Textul introdus trebuie sa fie destul de lung încât datele să poată fi interpretate ușor de catre modelul neuronal.
- Vor exista statistici în legătură cu fiecare notiță introdusă și grafice față de timp.
- Va exista un raport pentru fiecare notiță și rezultat.

### 6.1.2. Cerințe nefuncționale

- **Performanțe bune ale modelelor antrenate.** Modele vor respecta regulile generate impuse de comunitate (tipurile de straturi, transformarea text – valori numerice)
- **Programare în Python a modelului neuronal**, acest limbaj permite o dezvoltare facilă a acestor modele de învățare automată. Aceste module vor fi adăugate în aplicația mobilă.
- **Programare în JavaScript – React Native a aplicației mobile**
- **Interfața mobilă.** Aceasta va fi simplă, fără prea multe informații irelevante, cu instrucțiuni clare ale butoanelor sau ecranelor.

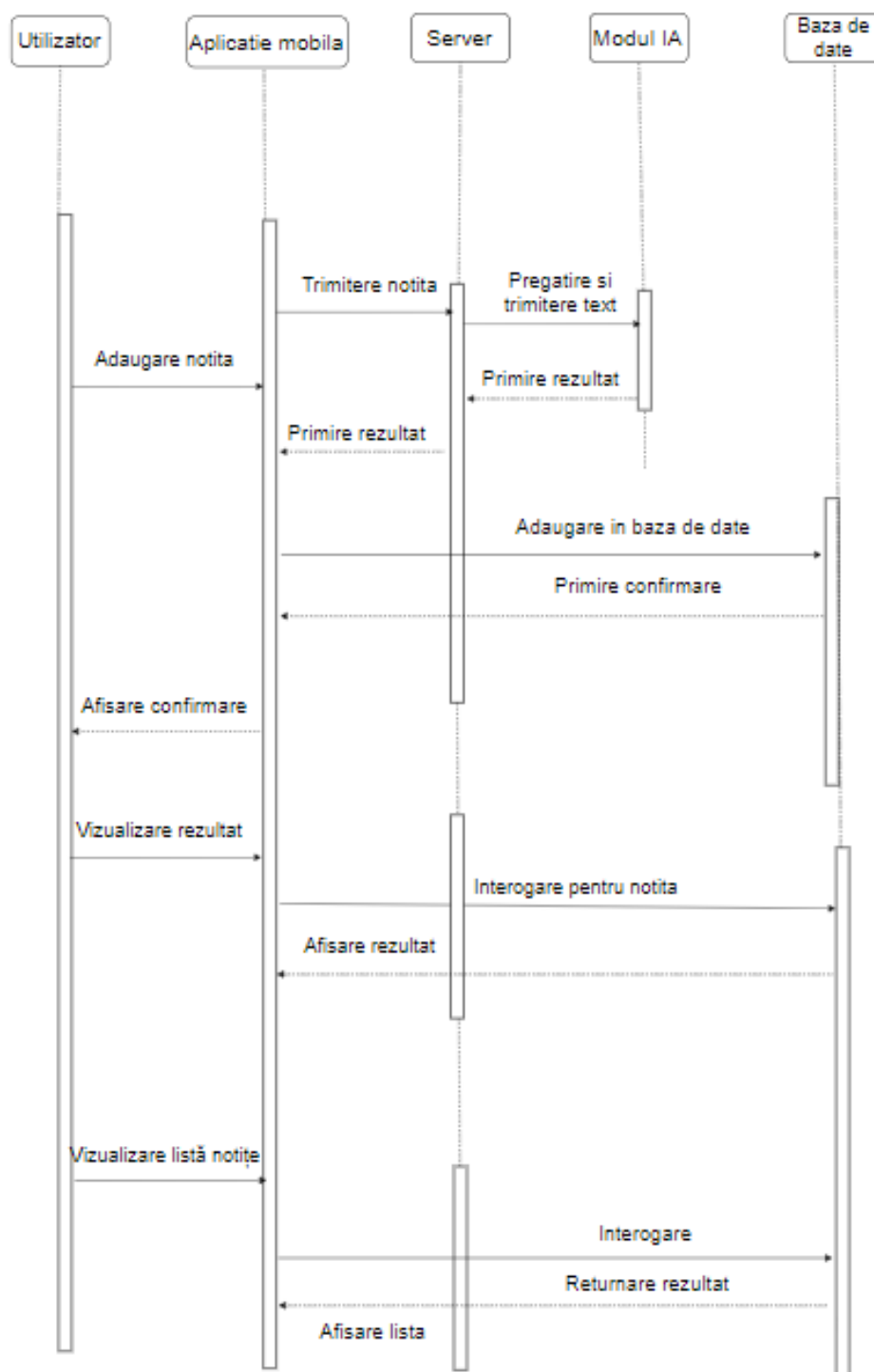
## 6.2. Diagrame Unified Modeling Language

Fluxul aplicației poate fi rezumat în cateva diagrame. Una dintre ele este diagrama cazurilor de utilizare care urmărește interacțiunea dintre utilizator și funcționalitățile aplicației.



Figură 6.1.2-1 Diagrama cazurilor de utilizare

O altă diagramă ce este folosită pentru înțelegerea acestor cazuri de utilizare este diagrama de secvență.



Figură 6.1.2-2 Diagrama de secvență

## 6.3. Server

Serverul a fost realizat în python, cu ajutorul librăriei *flask*<sup>9</sup> și a rulat în rețeaua locală. Acesta a integrat cele mai bune modele din fiecare tip de clasificare. Cel binar folosind Long Short Term Memory cu acuratețea de 96% a fost utilizat pentru detectarea existenței afecțiunilor psihice din text. Cel mai bun model folosit pentru clasificarea unei afecțiuni psihice din text a fost Mental RoBERTa ce clasifică tipul de afecțiune psihică cu o acuratețe de 74% și un scor f1 de 77%.

Serverul primește pe ruta acestuia un text ce conține textul introdus de utilizator. Se trece printr-un proces de curățare de date și preprocesare identic cu cel din cazul antrenării. Se identifică existența unei afecțiuni psihice folosind primul model, iar în caz că aceasta identifică un rezultat pozitiv în timpul prezicerii, se intră cu textul în cel de-al doilea model pentru clasificarea tipului de afecțiune. Serverul returnează aceste rezultate sub formă de probabilități înapoi la client unde sunt introduse în baza de date locală.

## 6.4. Client

Clientul proiectului constă în aplicația mobilă ce dă posibilitatea unui utilizator de a introduce notițe cărora li se va prezice nivelul afecțiunilor psihice din conținut. Se vor afișa rezultatele individuale dar și în grafice în funcție de timp pentru urmărirea unei evoluții a stării mentale. Clientul constă dintr-o aplicație mobilă cu o interfață simplă.

Se pot adăuga, edita și șterge notițe, dar și vizualiza detalii despre acestea, cum ar fi rezultatele din punctul de vedere al primului model și când e cazul din punctul de vedere și a celui de al doilea. Rezultatele sunt afișate într-o formă grafică, simplă și ușor de interpretat. Se poate observa evoluția în timp a fiecărei clase printr-un grafic colorat ce urmărește probabilitatea fiecărei afecțiuni din fiecare notiță introdusă.

Aplicația se deschide în ecranul principal, unde se află toate notițele existente deja (fig 6.1.2-1) . De aici se poate adăuga o notiță nouă (fig 6.1.2-2). După introducerea se așteaptă un răspuns cu rezultatele, acesta poate dura câteva secunde pentru a fi gata. La finalizare, se apasă pe notița nou creată (fig 6.1.2-3) și se deschid detaliile din meniu. Se afișează fereastra ce conține rezultatele notiței. Aceasta constă în două grafice, primul care spune probabilitatea de a suferi sau nu de o afecțiune psihică (fig 6.1.2-4). Și cel de-al doilea grafic, afișat doar în cazul în care probabilitatea ca utilizatorul să sufere de o afecțiune psihică

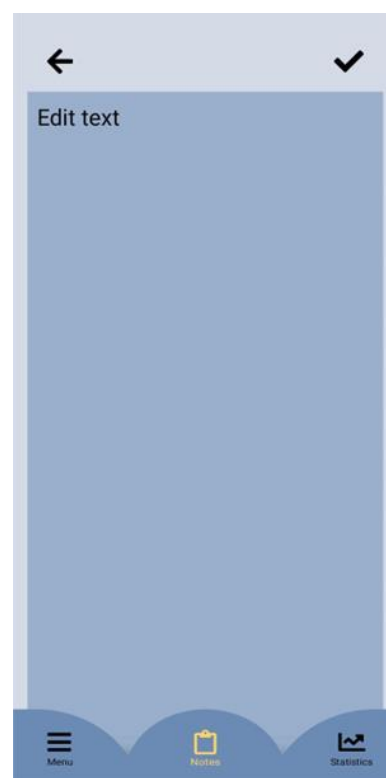
<sup>9</sup> <https://flask.palletsprojects.com/en/2.3.x/>

este mai mare de 50%, si constă în probabilitățile fiecărei afecțiuni extrasă din text. (fig 6.1.2-5). Pe lângă această funcționalitate de bază, există și ecranul de statistici ce poate fi apelat din bara de navigație, acesta deschide statisticile privind evoluția afecțiunilor mentale în timp. (fig 6.1.2-5).

O ultimă funcționalitate este cea din ecranul de meniu, apelat din bara de navigare, aceasta conține o intrare spre mai multe numere de telefon de urgență ce utilizatorul le poate folosi la nevoie.



Figură 6.1.2-2 Ecran 1

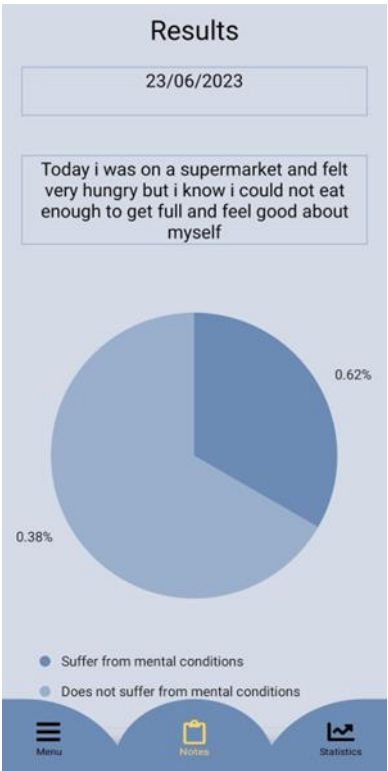


Figură 6.1.2-2 Ecran 2

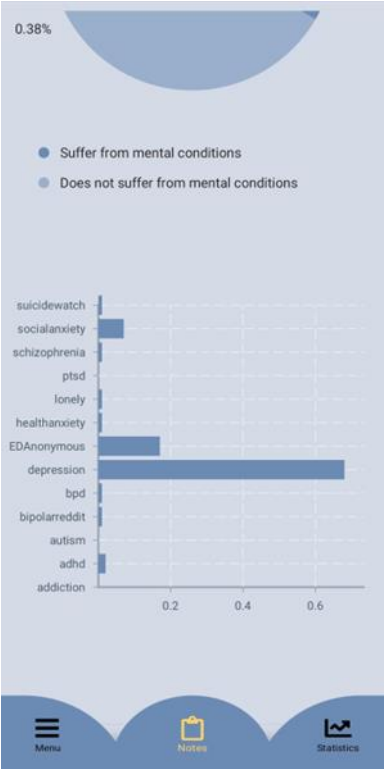




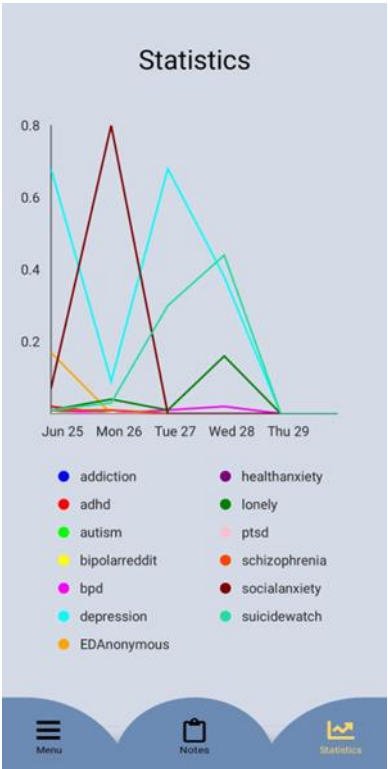
Figură 6.1.2-6 Ecran 3



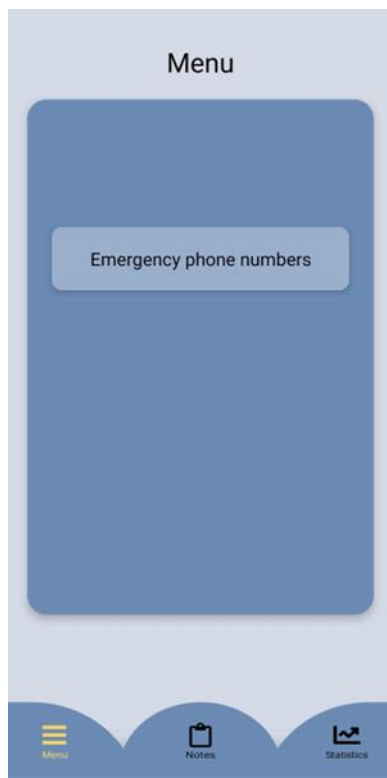
Figură 6.1.2-6 Ecran 4



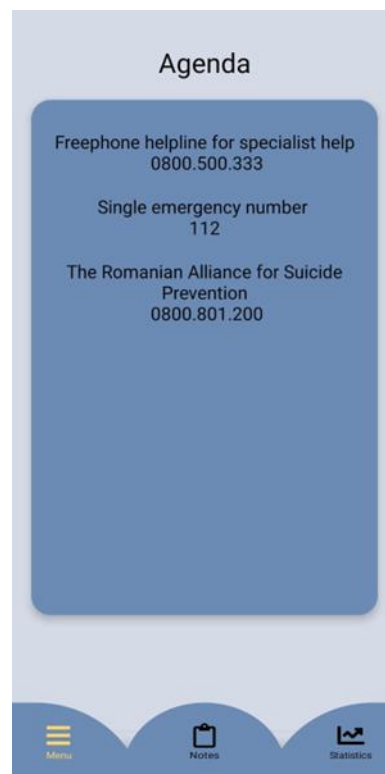
Figură 6.1.2-4 Ecran 5



Figură 6.1.2-4 Ecran 6



Figură 6.1.2-8 Ecran 7



Figură 6.1.2-8 Ecran 8

## 7. Concluzii și direcții viitoare de cercetare

Acest studiu a dorit să demonstreze posibilitatea detectării și clasificării cu ajutorul mecanismelor de învățare automată a afecțiunilor psihice întâlnite în viața de zi cu zi, dar și crearea unei aplicații accesibile publicului larg ce ar ajuta la detectarea timpurie a existenței unor astfel de afecțiuni. S-au cercetat următoarele tehnici și metode de clasificare în cazul unei astfel de probleme: Random Forest, Multinomial Naive Bayes, Long Short Term Memory (LSTM), Convolutional Neural Network (CNN), Bidirectional Encoder Representations from Transformers (BERT) și Robustly Optimized Bidirectional Encoder Representations from Transformers (RoBERTa), și s-a ajuns la concluzia că Random Forest se descurcă decent în cazul clasificării unor baze de date neechilibrate, Multinomial Naive Bayes este modelul cel mai puțin bun în această problemă. LSTM se descurcă bine pe parcursul antrenării simple, cu un număr mai mare de cuvinte, însă scorurile nu se îmbunătățesc în cazul folosirii ponderilor la clase în procesul de antrenare sau a augmentării datelor prin mai multe variante. BERT se descurcă bine, însă nu a reușit să concureze cu variantele folosite în cazul altor tipuri de clasificare și alte baze de date, unde diferențele erau mai mari față de cel mai bun model la momentul respectiv. RoBERTa are scoruri mai bune cu puțin față de BERT. Modelul cel mai bun rămâne Mental RoBERTa, care a ajuns la o valoare a acurateței de 74% și un scor f1 de 77%.

Pornind de la modelele de referință, Multinomial Naive Bayes și RandomForest cu o valoare a scorului f1 de 59% și 68% s-a reușit alegerea, antrenarea și construirea unor modele cu valori de la 72% la 77% în cazul Mental RoBERTa.

Pentru aplicația mobilă, a fost folosit cel mai bun model de clasificare cu mai multe clase și cel mai bun model de clasificare binară. Acestea au fost introduse în serverul creat și au ajutat la prezicerea în timp real a unor posibile afecțiuni ce au fost mai apoi afișate în interfață existentă.

Prezenta lucrare are ca dorința de viitor utilizarea unei baze de date corectate și verificate de un personal autorizat; dezvoltarea unui model ce va putea primi ca date de intrare text din mai multe limbi și prezicerea mai multor astfel de afecțiuni, dar și să se creeze o colaborare cu centre medicale autorizate cu care să se creeze o monitorizare mai atentă a pacienților în caz de nevoie.

NECLASIFICAT

Pagină albă

# Bibliografie

## Articole Științifice

1. Kamarul Adha. (2022). <i>Mental Disorders Identification (Reddit)</i> [Data set]. Kaggle. <https://doi.org/10.34740/KAGGLE/DSV/4579285>
2. Low, D. M., Rumker, L., Torous, J., Cecchi, G., Ghosh, S. S., & Talkar, T. (2020). Natural Language Processing Reveals Vulnerable Mental Health Support Groups and Heightened Health Anxiety on Reddit During COVID-19: Observational Study. *Journal of medical Internet research*, 22(10), e22635.
3. Murarka, A., Radhakrishnan, B., & Ravichandran, S. (2021). Classification of mental illnesses on social media using RoBERTa. In Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis (pp. 59-68). Association for Computational Linguistics. Retrieved from <https://aclanthology.org/2021.louhi-1.7>.
4. Kim, J., Lee, J., Park, E. et al. A deep learning model for detecting mental illness from user content on social media. *Sci Rep* 10, 11846 (2020). <https://doi.org/10.1038/s41598-020-68764-y>
5. Liu, W., Chen, W., Tang, J., Li, J., & Guo, Z. (2021). GRAN: Granular Knowledge Aware Network for Explainable and Generalizable Recommendation. Retrieved from arXiv:2110.15621.
6. F. Arias, M. Zambrano Núñez, A. Guerra-Adames, N. Tejedor-Flores and M. Vargas-Lombardo, "Sentiment Analysis of Public Social Media as a Tool for Health-Related Topics," in *IEEE Access*, vol. 10, pp. 74850-74872, 2022, doi: 10.1109/ACCESS.2022.3187406.
7. Gkotsis, G., Oellrich, A., Velupillai, S., Liakata, M., Hubbard, T. J., Dobson, R. J., & Dutta, R. (2017). Characterisation of mental health conditions in social media using Informed Deep Learning. *Scientific reports*, 7, 45141. <https://doi.org/10.1038/srep45141>
8. Jetli Chung, Jason Teo, "Mental Health Prediction Using Machine Learning: Taxonomy, Applications, and Challenges", *Applied Computational Intelligence and Soft Computing*, vol. 2022, Article ID 9970363, 19 pages, 2022. <https://doi.org/10.1155/2022/9970363>
9. Cohan, A., Desmet, B., Yates, A., Soldaini, L., MacAvaney, S., & Goharian, N. (2018). {SMHD}: A Large-Scale Resource for Exploring Online Language Usage for Multiple Mental Health Conditions. In Proceedings of the 27th International Conference on Computational Linguistics (pp. 1485-1497). Santa Fe, New Mexico, USA: Association for

- Computational Linguistics. Retrieved from <https://aclanthology.org/C18-1126>.
10. Ji, S., Zhang, T., Ansari, L., Fu, J., Tiwari, P., & Cambria, E. (2022). {M}ental{BERT}: Publicly Available Pretrained Language Models for Mental Healthcare. In Proceedings of the Thirteenth Language Resources and Evaluation Conference (pp. 7184-7190). Marseille, France: European Language Resources Association. Retrieved from <https://aclanthology.org/2022.lrec-1.778>.
  11. Cite as: Al Hanai, T., Ghassemi, M., Glass, J. (2018) Detecting Depression with Audio/Text Sequence Modeling of Interviews. Proc. Interspeech 2018, 1716-1720, doi: 10.21437/Interspeech.2018-2522
  12. Hussein Orabi, A., Buddhitha, P., Hussein Orabi, M., & Inkpen, D. (2018). Deep Learning for Depression Detection of {T}witter Users. In Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic (pp. 88-97). New Orleans, LA: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/W18-0609>
  13. Dham, S., Sharma, A., & Dhall, A. (2017). Depression Scale Recognition from Audio, Visual and Text Analysis. Retrieved from arXiv preprint arXiv:1709.05865.
  14. Zhang, T., Schoene, A. M., Ji, S., & Ananiadou, S. (2022). Natural Language Processing Applied to Mental Illness Detection: A Narrative Review. npj Digital Medicine, 5(46). <https://doi.org/10.1038/s41746-022-00589-7>

## Cărți

15. World Health Organization. (2014). Handbook for guideline development. Retrieved from <https://www.who.int/publications/i/item/9789240049338>
16. Chollet, F. (2021). Deep Learning with Python, 2nd Edition. Shelter Island, NY: Manning Publications.
17. Vajjala, S., Majumder, B., Gupta, A., & Surana, H. (2020). Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems. Sebastopol, CA: O'Reilly Media.

GitHub: <https://github.com/Aleexuta/Mentol.git>