



Electronic, Telecommunications and Informational Technology

TEZĂ DE DOCTORAT
- REZUMAT -

**Îmbunătățirea algoritmilor de recunoaștere a emoțiilor roboților
folosind semnale bazate pe vorbire**

Student-doctorand:
Toma TELEMBICI

Conducător științific:
Prof. Dr. Ing. Corneliu RUSU

Comisia de evaluare a tezei de doctorat:

Președinte: Prof. Dr. Ing. **Sorin Hintea** - Universitatea Tehnică din Cluj-Napoca;

Conducător științific: Prof. Dr. Ing. **Corneliu Rusu** - Universitatea Tehnică din Cluj-Napoca;

Referenți:

- Prof. Dr. Ing. **Corneliu Burileanu** – Universitatea Nationala de Stiinta si Tehnologie Politehnica Bucuresti;

- Prof. Dr. Ing. **Corina Naforniță** - Universitatea Politehnica din Timisoara;

- Prof. Asoc. Dr. Ing. **Lăcrimioara Grama** - Universitatea Tehnica din Cluj-Napoca.

- Cluj-Napoca -
2024

CUPRINS

ABREVIERI.....	Error! Bookmark not defined.
1. Introducere.....	Error! Bookmark not defined.
2. Roboți.....	Error! Bookmark not defined.
2.1. Introducere.....	Error! Bookmark not defined.
2.2. Roboți de servicii comerciale	Error! Bookmark not defined.
2.2.1. Educaționali.....	Error! Bookmark not defined.
2.2.2. Divertisment	Error! Bookmark not defined.
2.2.3. Din interiorul locuinței	Error! Bookmark not defined.
2.2.4. Sociali.....	Error! Bookmark not defined.
2.2.5. Jocuri	Error! Bookmark not defined.
2.3. Roboți de servicii pentru cercetare.....	Error! Bookmark not defined.
2.3.1. Securitate	Error! Bookmark not defined.
2.3.2. Divertisment	Error! Bookmark not defined.
2.3.3. Platforme de dezvoltare	Error! Bookmark not defined.
2.3.4. Reabilitare - Gospodărie.....	Error! Bookmark not defined.
2.4. Direcția de dezvoltare	Error! Bookmark not defined.
2.4.1. Recunoaștere audio.....	Error! Bookmark not defined.
2.4.2. Recunoaștere audio-video.....	Error! Bookmark not defined.
2.4.3. Recunoașterea emoțiilor	Error! Bookmark not defined.
2.5. Concluzia capitolului.....	Error! Bookmark not defined.
3. Baze de date Emoționale	Error! Bookmark not defined.
3.1. Ce este o bază de date emoțională.....	Error! Bookmark not defined.
3.1.1. Grupările emoționale ale lui Parrott.....	Error! Bookmark not defined.
3.2. Care este scopul creării unei baze de date?.....	Error! Bookmark not defined.
3.3. Baze de date utilizate.....	Error! Bookmark not defined.
3.3.1. CREMA.....	Error! Bookmark not defined.
3.3.2. RAVDESS.....	Error! Bookmark not defined.
3.3.3. SAVEE	Error! Bookmark not defined.
3.3.4. Emo-DB	Error! Bookmark not defined.
3.4. Caracteristici extrase	Error! Bookmark not defined.
3.4.1. Frecvența fundamentală.....	Error! Bookmark not defined.
3.4.2. Cepstrum și MFCC.....	Error! Bookmark not defined.
3.4.3. Linear Predictive Coding.....	Error! Bookmark not defined.

- 3.4.4. Magnitude-Based Spectral Root Cepstral Coefficients (MSRCC) **Error! Bookmark not defined.**
- 3.4.5. Normalized gammachirp cepstral coefficients (NGCC)**Error! Bookmark not defined.**
- 3.4.6. Gammatone frequency cepstral coefficients (GFCC)**Error! Bookmark not defined.**
- 3.4.7. Extragerea caracteristicii Chroma**Error! Bookmark not defined.**
- 3.4.8. Extracția caracteristică bazată pe spectrograma Mel**Error! Bookmark not defined.**
- 3.5. Clasificatori folosiți..... **Error! Bookmark not defined.**
 - 3.5.1. K-nearest neighbours (*K*-NN)**Error! Bookmark not defined.**
 - 3.5.2. SVM..... **Error! Bookmark not defined.**
 - 3.5.3. RF..... **Error! Bookmark not defined.**
 - 3.5.4. Multimodal Logistic Regression**Error! Bookmark not defined.**
- 4. Crearea bazei de date..... **Error! Bookmark not defined.**
 - 4.1. Conținutul bazei de date..... **Error! Bookmark not defined.**
 - 4.2. Analiza bazei de date **Error! Bookmark not defined.**
 - 4.3. Concluzii **Error! Bookmark not defined.**
- 5. Implementare și Rezultate folosind doar clasificatorul k-NN**Error! Bookmark not defined.**
 - 5.1. Implementare **Error! Bookmark not defined.**
 - 5.1.1. Python..... **Error! Bookmark not defined.**
 - 5.1.2. De ce folosim Python pentru AI și ML?**Error! Bookmark not defined.**
 - 5.1.3. Extragerea de Caracteristici**Error! Bookmark not defined.**
 - 5.2. Rezultate folosind doar clasificatorul k-NN**Error! Bookmark not defined.**
 - 5.2.1. CREMA rezultate..... **Error! Bookmark not defined.**
 - 5.2.2. RAVDESS rezultate **Error! Bookmark not defined.**
 - 5.2.3. Emo-DB rezultate **Error! Bookmark not defined.**
 - 5.2.4. SAVEE rezultate **Error! Bookmark not defined.**
 - 5.2.5. Rezultate pentru baza de date din limba română**Error! Bookmark not defined.**
 - 5.2.6. Alte rezultate..... **Error! Bookmark not defined.**
 - 5.3. Concluzii **Error! Bookmark not defined.**
- 6. Implementarea clasificatorilor Support Vector Classifier, Logistic Regression și Random Forest și rezultatele obținute **Error! Bookmark not defined.**
 - 6.1. Implementarea clasificatorilor rămași**Error! Bookmark not defined.**
 - 6.1.1. Support vector classifier (SVC)**Error! Bookmark not defined.**
 - 6.1.2. Logistic regression (LR) **Error! Bookmark not defined.**
 - 6.1.3. Random forest classifier..... **Error! Bookmark not defined.**

6.2. Rezultate.....	Error! Bookmark not defined.
6.2.1. CREMA rezultate.....	Error! Bookmark not defined.
6.2.2. RAVDESS rezultate	Error! Bookmark not defined.
6.2.3. SAVEE rezultate	Error! Bookmark not defined.
6.2.4. Emo-DB rezultate.....	Error! Bookmark not defined.
6.2.5. Rezultate pentru baza de date din limba română	Error! Bookmark not defined.
6.2.6. Alte rezultate.....	Error! Bookmark not defined.
6.3. Concluzii	Error! Bookmark not defined.
7. Concluzii generale	Error! Bookmark not defined.
7.1. Comparație cu rezultatele obținute de alți cercetători	Error! Bookmark not defined.
7.2. Originalitatea tezei de doctorat	Error! Bookmark not defined.
7.3. Contribuții.....	Error! Bookmark not defined.
REFERINȚE.....	Error! Bookmark not defined.
LISTA DE FIGURI	Error! Bookmark not defined.
LISTA DE TABELE	Error! Bookmark not defined.
LISTA DE PUBLICAȚII	Error! Bookmark not defined.

Introducere

Pentru că oamenii sunt ființe emoționale, tot ceea ce intră în contact cu noi pare lipsit de viață și rece dacă nu există emoție. Având în vedere că roboții au un impact major asupra vieții noastre de zi cu zi, există o necesitate vizibilă de a-i face să semene mai mult cu oamenii. Pe măsură ce domeniul recunoașterii vocii și al recunoașterii vorbitorilor atinge apogeul, următorul domeniu care, în mod firesc, va fi cercetat este cel al recunoașterii emoțiilor. Acest domeniu este relativ nou și se depun eforturi pentru a crește calitatea rezultatelor și a resurselor disponibile în acest domeniu.

După cum știm, există numeroase țări care promovează dezvoltarea domeniului recunoașterii emoțiilor. În prezent, există algoritmi puternici de recunoaștere a emoțiilor pe baza recunoașterii imaginilor, care sunt utilizați în domeniul securității, în special în SUA și în China. Domeniul recunoașterii emoțiilor pe baza modelelor vocale este încă subdezvoltat, dar există numeroase țări care dezvoltă baze de date și caută algoritmi practici.

Titlul tezei de doctorat este "Îmbunătățirea algoritmilor de recunoaștere a emoțiilor de către roboți folosind semnale bazate pe voce". După cum am menționat deja, emoția joacă un aspect important în modul în care oamenii percep diferite întâlniri cu roboți sau lucruri. Există categorii de persoane care au o interacțiune limitată cu alți oameni din motive de siguranță, de sănătate sau din alte motive. Pentru aceștia, o interacțiune cu roboți emoționali poate avea un impact mare asupra sănătății lor mintale și poate avea o oscilație în starea lor emoțională actuală. Obiectivul acestei cercetări este de a crea o bază de date emoționale în limba română pentru a include persoanele dintr-o zonă demografică în categoria potențialilor beneficiari ai roboților emoționali. Un alt obiectiv este de a găsi un algoritm care să aibă rezultate bune pe bazele de date deja existente și de a testa noua bază de date pentru a vedea dacă poate fi inclusă alături de celelalte din partea de cercetare.

Pentru a ne atinge toate obiectivele, am încercat în primul rând să ne punem la curent cu toate cercetările noi care au fost efectuate în domeniu. Apoi am încercat să vedem ce sunt emoțiile, cum sunt produse, care sunt familiile de emoții și care sunt granițele emoțiilor. Pentru a crea propria noastră bază de date de emoții, a trebuit să determinăm ce emoții sunt prezente de obicei în viața noastră de zi cu zi și ce propoziții pot avea înțelesuri diferite atunci când sunt rostite în cadrul unor emoții diferite. Pentru a crea un algoritm care să fie performant în domeniul vorbirii emoționale, a trebuit să cercetăm ce extractori de caracteristici și ce clasificatori sunt utilizați în acest domeniu. Am ales un clasificator și apoi am testat diferiți extractori de caracteristici pentru a vedea rezultatul obținut prin aceste combinații, utilizând toate bazele de date de testare și noua bază de date pe care am creat-o. După ce am eliminat cei mai slabi extractori de caracteristici, am adăugat mai mulți clasificatori pentru a vedea ce combinații pot oferi un algoritm puternic. În final, am creat un model care a fost testat pe baza de date proaspăt creată.

În stadiul actual al părții de cunoaștere a tezei, capitolul de început este capitolul "Roboți", în care sunt prezentați diferiți roboți comerciali sau de serviciu pentru a vedea cu ce avem de lucrat și care este direcția de dezvoltare. Următorul capitol al tezei se intitulează "Baze de date emoționale", unde se definește ce este o bază de date emoționale, care sunt emoțiile de bază, care sunt emoțiile de bază contrastante și ce grupuri de emoții există. În acest capitol spunem, de asemenea, care este importanța creării unei noi baze de date. În capitolul următor vom aborda ce baze de date sunt folosite ca test în această lucrare și cum sunt compuse. În continuare, vorbim despre diferitele tipuri de caracteristici extrase și ce clasificatori sunt utilizați.

În partea de contribuție personală a tezei, mai întâi vorbim despre cum am creat noua bază de date, care este conținutul bazei de date și câteva analize efectuate pe baza

conținutului acesteia. Următorul capitol prezintă implementarea și rezultatele obținute folosind un singur clasificator selectat. Teza continuă cu implementarea celorlalți clasificatori și cu rezultatele acestora. După aceea, analizăm alte rezultate obținute în același domeniu și le comparăm cu rezultatele obținute prin cercetarea noastră. La final avem concluziile noastre finale bazate pe munca depusă până acum, urmate de lista de referințe, de lista de figuri și de lista de tabele.

Crearea bazei de date

Baza de date emoțională a fost gândită și concepută în așa fel încât fiecare propoziție să poată avea diferite înțelesuri în funcție de emoția transmisă prin acea propoziție. Propozițiile sunt propuse pentru a fi înțelese și folosite de către un robot de servicii utilizat într-un mediu interior. Propozițiile propuse sunt următoarele:

“A venit rezultatul testului.”

“Nu mai am medicamente.”

“Testul este pozitiv.”

“Interpretarea a fost greșită.”

“A crescut foarte mult.”

“A scăzut foarte mult.”

“Îi închisă ușa.”

“Îi stins becul.”

“Sunt foarte bine.”

“Cheile sunt în ușa.”

Pentru a înțelege mai bine fiecare propoziție în parte, vom analiza situațiile în care propozițiile ar putea fi folosite și prin diferite emoții cum ar putea fi interpretate.

Prima propoziție, “A venit rezultatul testului.”, spusă pe un ton fericit, poate indica o scăpare din boală, un test de sarcină așteptat, sau un răspuns pozitiv al unui examen. Acest răspuns poate veni și ca o surpriză pentru vorbitor. Prima propoziție rostită pe un ton neutru poate indica că acel răspuns primit, nu este o surpriză, ci mai mult vine ca o validare către vorbitor. Când această propoziție este rostită cu o emoție tristă, poate însemna primirea unei vești proaste de către vorbitor, ca și descoperirea sau agravarea unei boli, sau un rezultat negativ a unui examen. Prin cunoașterea acestor emoții și decriptarea mesajului ascuns, robotul de servicii poate avea o înțelegere mai amplă asupra evenimentelor care au avut loc. Robotul de servicii poate acționa cu un răspuns mult mai așteptat și evident persoanelor umane.

A doua propoziție, “Nu mai am medicamente”, primită de către robot având o emoție veselă, poate susține faptul că vorbitorul și-a terminat medicația prescrisă care ar presupune și sfârșitul bolii în care se afla. Pe un ton neutru, vorbitorul poate realiza că medicamentele s-au terminat, în continuare are nevoie de ele, dar nu îi împiedică sănătatea dacă ar merge după altele mai târziu. Pe o emoție tristă, propoziția reflectă nevoia urgentă a vorbitorului pentru medicamente, care îi poate pune în pericol chiar și viața.

A treia propoziție, “Testul este pozitiv”, poate fi descifrată doar din context sau din emoția transmisă. Pe o emoție pozitivă, propoziția transmite feririrea vorbitorului și că un eveniment pozitiv s-a întâmplat în viața vorbitorului, de exemplu un test de sarcină. Propoziția pe un ton neutru rezultă doar că vorbitorul se aștepta la acest rezultat. Nu putem spune concret că este un eveniment pozitiv sau negativ în viața vorbitorului, din context putem să ne dăm seama mai bine la ce se referă. O emoție negativă imprigmată asupra acestei propoziții poate rezulta

doar că vorbitorul a primit o veste proastă și că evenimentul la care se referă vorbitorul este unul negativ, de exemplu un test SarsCov-2.

Cea de a patra propoziție: "Interpretarea a fost greșită", ne duce cu gândul la un diagnostic pus greșit de către un medic, apoi acesta să fie infirmat de către un altul. Emoția acestei propoziții poate schimba total sensul acesteia. Dacă avem o emoție fericită, adevăratul diagnostic este unul care îl ajută pe pacient. Poate fi că acesta este complet sănătos sau doar că boala pe care o are nu este la fel de gravă ca boala cu care fusese diagnosticat inițial. Emoția de tristețe care acaparează această propoziție este exact inversul propoziției cu emoție fericită. Putem să ne dăm seama că diagnosticul pus ulterior este unul sever care poate pune în pericol viața vorbitorului sau poate are nevoie de ajutor imediat.

Cea de a cincea propoziție: "A crescut foarte mult", fără context sau fără emoție poate fi aproape imposibil de descifrat de către persoanele umane, rezultând că roboții dacă vor încerca să descifreze această propoziție este echivalentul unui rezultat a unei funcții random. În lipsa contextului ne putem baza pe emoții. Pentru o emoție de fericire, de exemplu vorbitorul se poate bucura de creșterea în înălțime a unui copil. Pentru o emoție neutră, nu putem să apreciem dacă este un eveniment fericit sau trist în viața vorbitorului, doar putem presupune că evenimentul este unul la care vorbitorul se aștepta să se întâmple. O emoție negativă rezultă că în viața vorbitorului un eveniment negativ, chiar și așteptat a avut loc. De exemplu, o creștere a colesterolului la analize sau o creștere a unui hematom de la ultima verificare.

Cea de a șasea propoziție: "A scăzut foarte mult.", este extrem de asemănătoare cu predecesoarea ei, această propoziție fiind introdusă atât pentru ca vorbitorul să aibă mai multe variante de exprimare, dar și pentru a introduce dificultate în recunoașterea propoziției corecte.

Cea de a șaptea propoziție: "Îi închisă ușa.", introduce un limbaj mai colocvial pe care robotul este probabil să îl audă. Această propoziție poate activa și funcțiile locomotorii ale robotului de servicii, asta depinzând de emoția transmisă. Dacă vorbitorul transmite o emoție pozitivă prin această propoziție, acesta dorește să exprime că ușa este închisă și ar trebui să rămână așa. Dacă robotul de servicii detectează o emoție neutră, mesajul pe care acesta ar trebui să îl recepționeze, este că ușa este închisă, dar în orice moment poate primi un mesaj să caute cheile sau doar să mențină ușa închisă. Dacă emoția transmisă prin acest mesaj este una negativă, robotul ar trebui să înțeleagă că ușa este închisă și ar trebui să caute cheile dacă ușa este încuiată și să o deschidă.

Cea de a opta propoziție este similară cu predecesoarea ei, având sensuri similare dar scopuri diferite. Aici robotul de servicii, poate ajuta la aprinderea sau stingerea becului într-o anumită cameră. Această propoziție fiind foarte utilă pentru persoane cu handicap locomotor.

Cea de a noua propoziție, "Sunt foarte bine.", poate expune una dintre problemele secolului, și anume depresia. Dacă robotul de servicii poate interveni și distinge depresia încă dintr-o stare incipientă a acesteia, poate ajuta în mod inimaginabil de mult acea persoană. Acea persoană poate lua acțiuni în înlăturarea depresiei până când nu este într-o formă avansată. Dacă emoția transmisă prin această propoziție este una pozitivă, înseamnă că mesajul paraverbal este același cu mesajul verbal. Dacă propoziția primită de robot are un mesaj neutru, asta nu înseamnă că vorbitorul vrea să înșele robotul prin mesajul transmis, vorbitorul poate fi doar obosit. Cu toate acestea robotul ar trebui să rămână vigilent în următoarele mesaje și emoții transmise de către vorbitor. Dacă emoția transmisă este una negativă, robotul ar trebui să înțeleagă că vorbitorul nu se simte bine și ar trebui să ofere confort emoțional sau să cheme ajutoare dacă este vorba de un accident fizic.

Ultima propoziție: "Cheile sunt în ușa", poate avea mai multe înțelesuri paraverbale, chiar dacă mesajul verbal este destul de clar. Dacă emoția care însoțește propoziția este una

pozitivă, poate avea ca un mesaj, bucuria de a găsi cheile după o căutare îndelungată, sau doar bucuria că acele chei se află unde trebuie. Dacă emoția este una neutră, mesajul poate fi ca robotul să aducă acele chei vorbitorului sau doar un mesaj ca să se știe de locația acelor chei. Dacă emoția este una negativă, robotul ar trebui să scoată acele chei din ușa și să le aducă vorbitorului.

Aceste propoziții pot face ca roboții să vină mai aproape în ajutorul oamenilor și să înțeleagă atât limbajul discret cât și mesajul transmis paraverbal. Prin aceste 10 propoziții putem da un exemplu de folosință a unei baze de date emoționale cât și importanța acesteia. Există posibilitatea ca un robot de servicii care nu înțelege limbajul paraverbal să nu poată decoda mesajul transmis sau chiar decoda greșit și să acționeze într-un mod eronat.

Baza de date are un total de 4713.391 secunde sau 78.5 minute. Este formată din 10 propoziții explicate anterior, fiecare propoziție având 3 tipuri de emoții explicate. Toți cei 7 vorbitori au repetat fiecare propoziție pe fiecare emoție de 10 ori. Fiecare propoziție, în medie are o lungime de 2.244 secunde. Baza de date are un total de 2100 de propoziții, din care 700 sunt pe emoție veselă, 700 au o emoție neutră, iar 700 au o emoție tristă.

Baza de date este structurată în următorul mod:

- Fiecare folder reprezintă clasa, iar fiecare dintre ele este numerotată în felul următor: "*NumelePersoanei_pk_emoție*", unde *k* reprezintă numărul propoziției și emoția poate fi fericit, neutru sau trist.
- Fiecare clasă are 10 instanțe numite în următorul fel: "*NumelePersoanei_pk_emoție_n.wav*", unde *n* merge de la 01 la 10 și reprezintă numărul instanței.
- P1 reprezintă "A venit rezultatul testului.", P2 reprezintă "Nu mai am medicamente.", P3 reprezintă "Testul este pozitiv.", P4 reprezintă "Interpretarea a fost gresita.", P5 reprezintă "A crescut foarte mult.", P6 reprezintă "A scăzut foarte mult.", P7 reprezintă "Ii inchisa usa.", P8 reprezintă "Ii stins becul.", P9 reprezintă "Sunt foarte bine." Și în final P10 reprezintă "Cheile sunt in usa."
- Persoanele 1, 3, 4 și 7 sunt de sex masculine, iar persoanele 2, 5 și 6 sunt de sex feminine.

În Fig. 4.1 Este reprezentată distribuția instanțelor pentru 1 din totalul de 210 clase.

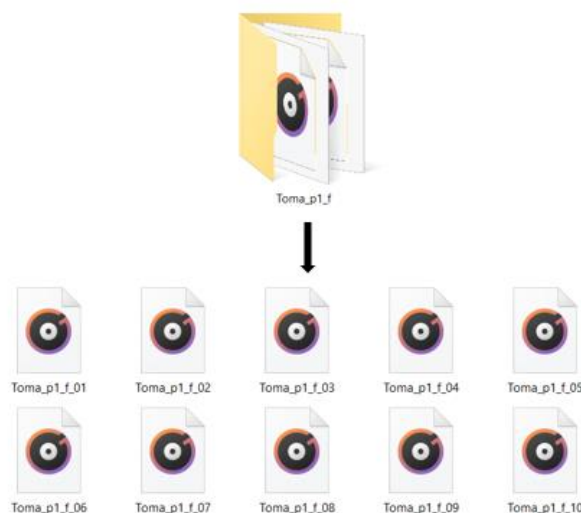


Fig 4.1 Distribuția înregistrărilor

Class	No.	Person1 [s]	Person2 [s]	Person3 [s]	Person4 [s]	Person5 [s]	Person6 [s]	Person7 [s]	Total [s]
P1 - f	1	2.560	2.731	3.157	2.901	2.731	2.816	2.816	19.712
	2	2.731	2.987	3.243	3.157	2.731	3.072	2.560	20.481
	3	2.645	3.072	2.475	2.987	2.475	3.072	2.816	19.542
	4	2.475	2.987	2.731	3.157	2.560	2.901	2.901	19.712
	5	2.645	2.560	2.987	2.816	2.389	2.816	2.901	19.114
	6	2.645	3.413	2.816	2.901	2.475	2.901	2.731	19.882
	7	2.645	2.816	3.072	2.731	2.389	3.243	2.645	19.541
	8	2.645	3.840	3.072	2.731	2.389	2.560	2.645	19.882
	9	2.645	2.901	2.645	2.645	2.475	2.731	2.645	18.687
	10	2.816	3.072	2.901	2.987	2.389	2.731	2.475	19.371
P1 - n	1	2.645	3.072	2.731	3.072	2.645	2.816	2.219	19.200
	2	2.560	3.328	3.157	3.072	2.219	2.731	2.560	19.627
	3	2.389	3.328	2.560	2.731	2.475	2.816	2.048	18.347
	4	2.475	3.413	2.901	2.987	2.389	2.475	2.304	18.944
	5	2.475	3.755	2.731	2.816	2.219	2.560	2.219	18.775
	6	2.475	3.413	2.987	2.901	2.389	2.816	2.219	19.200
	7	2.645	3.328	2.645	2.987	2.133	2.645	2.219	18.602
	8	2.475	3.072	2.901	2.731	2.219	2.987	2.304	18.689
	9	2.475	3.328	2.560	2.816	2.389	2.560	2.219	18.347
	10	2.731	3.072	2.731	2.731	2.304	2.560	2.219	18.348
P1 - t	1	3.072	3.413	2.987	2.645	2.560	2.816	2.645	20.138
	2	2.987	3.413	3.072	2.304	2.560	2.560	2.645	19.541
	3	3.072	3.243	2.987	2.389	2.475	2.645	2.560	19.371
	4	3.157	3.243	2.645	2.475	2.475	2.901	2.475	19.371
	5	3.157	3.328	2.560	2.475	2.560	2.560	2.389	19.029
	6	3.072	3.243	2.816	2.304	2.475	2.560	2.304	18.774
	7	3.072	3.072	2.645	2.389	2.304	2.645	2.475	18.602
	8	3.157	3.499	2.731	2.389	2.389	2.731	2.389	19.285
	9	3.072	3.157	2.731	2.304	2.645	2.560	2.645	19.114
	10	2.816	3.243	2.560	2.219	2.304	2.901	2.731	18.774
P2 - f	1	1.877	2.560	2.645	2.304	2.475	2.731	2.304	16.896
	2	2.219	2.731	2.475	2.475	2.133	2.560	2.389	16.982
	3	2.219	2.475	2.389	2.475	2.048	2.560	2.475	16.641
	4	2.304	2.560	2.560	2.560	1.963	2.389	2.219	16.555
	5	2.560	2.304	2.304	2.304	1.963	2.219	2.048	15.702
	6	2.048	3.072	2.560	2.389	2.133	2.304	2.560	17.066
	7	1.877	2.645	2.560	2.389	1.963	2.304	2.219	15.957
	8	2.133	2.645	2.219	2.645	2.133	2.304	2.389	16.468
	9	2.133	2.731	2.475	2.219	1.963	2.560	2.475	16.556
	10	2.304	2.901	2.219	2.816	2.048	2.645	2.219	17.152
P2 - n	1	2.133	2.645	2.731	2.304	2.219	2.731	1.707	16.470
	2	2.219	3.157	2.304	2.219	2.048	2.475	1.963	16.385
	3	2.048	2.731	2.560	2.304	1.963	2.219	1.707	15.532
	4	1.877	2.987	2.475	2.389	1.963	1.877	1.877	15.445
	5	2.219	2.987	2.475	2.475	2.048	2.219	1.877	16.300
	6	2.048	2.901	2.560	2.133	1.792	2.133	1.963	15.53

	7	2.560	2.901	2.475	2.389	1.877	2.219	1.963	16.384
	8	2.389	3.669	2.560	2.389	1.877	2.304	1.877	17.065
	9	1.877	2.304	2.475	2.219	2.048	2.219	1.707	14.849
	10	2.048	2.475	2.560	2.219	2.048	2.389	2.219	15.958
P2 - t	1	2.219	3.499	2.475	1.877	2.219	1.963	2.133	16.385
	2	2.304	2.731	2.560	2.219	2.304	2.389	2.219	16.726
	3	2.389	2.816	2.219	1.963	2.389	2.645	2.475	16.896
	4	2.133	3.328	2.304	2.048	2.219	2.389	2.048	16.469
	5	2.475	3.072	2.304	1.963	2.304	2.475	2.048	16.641
	6	2.219	3.243	2.304	1.963	2.219	2.475	1.877	16.300
	7	2.304	2.901	2.219	2.133	2.133	2.475	2.389	16.554
	8	2.475	2.901	2.304	2.133	1.963	2.304	2.219	16.299
	9	2.389	2.816	2.219	2.133	2.048	2.389	2.133	16.127
	10	2.389	3.072	2.048	1.963	2.133	2.389	2.133	16.127
P3 - f	1	2.133	2.389	2.219	2.133	2.219	2.389	2.219	15.701
	2	2.475	2.901	2.389	2.304	2.304	2.304	2.133	16.810
	3	2.304	2.731	2.304	1.963	2.048	2.219	2.304	15.873
	4	2.304	2.731	2.304	2.304	2.048	2.389	2.219	16.299
	5	2.304	2.304	2.304	2.304	2.048	2.133	2.219	15.616
	6	1.963	2.816	2.901	2.048	2.133	2.389	2.304	16.554
	7	2.133	2.475	2.645	2.304	2.048	2.133	2.133	15.871
	8	2.048	3.243	2.475	2.048	2.048	2.048	2.048	15.958
	9	2.133	2.901	2.304	2.133	2.048	2.304	2.219	16.042
	10	2.304	2.901	2.389	1.963	2.048	2.133	2.133	15.871
P3 - n	1	1.792	2.560	2.389	3.072	2.133	2.475	2.048	16.469
	2	1.963	2.645	2.645	2.389	2.133	2.219	2.133	16.127
	3	1.877	2.389	2.645	2.048	1.877	2.133	2.475	15.444
	4	1.963	2.645	2.475	2.475	1.877	2.219	1.877	15.531
	5	1.963	2.901	2.645	2.048	1.877	2.219	1.707	15.360
	6	1.792	2.304	2.389	2.219	1.877	2.389	1.877	14.847
	7	1.963	2.645	2.475	2.304	1.707	2.133	1.707	14.934
	8	1.877	2.560	2.133	2.304	1.877	2.389	1.621	14.761
	9	1.963	2.560	2.219	2.133	1.877	2.304	1.707	14.763
	10	2.048	2.389	2.304	2.048	1.877	2.219	1.792	14.677
P3 - t	1	2.133	2.987	2.219	2.133	2.219	2.219	2.133	16.043
	2	2.304	3.072	2.389	2.133	2.048	2.560	2.389	16.895
	3	2.048	3.072	2.304	1.877	1.963	2.304	2.389	15.957
	4	2.304	3.072	2.219	1.877	2.048	2.389	2.389	16.298
	5	2.219	3.157	2.133	1.792	1.963	2.475	2.133	15.872
	6	2.219	2.987	2.304	1.792	2.133	2.475	1.963	15.873
	7	2.304	2.901	2.133	1.877	2.133	1.963	2.133	15.444
	8	2.133	3.157	2.389	1.963	2.048	2.219	1.877	15.786
	9	2.048	2.987	2.219	2.389	2.048	2.731	1.792	16.214
	10	2.048	2.816	2.304	2.048	1.963	2.304	1.963	15.446
P4 - f	1	2.475	2.731	2.731	2.219	2.389	2.304	2.816	17.665
	2	2.304	2.987	2.816	2.133	2.133	2.731	2.304	17.408
	3	2.475	3.243	2.816	2.133	2.133	2.304	2.304	17.408
	4	2.475	2.816	3.243	2.731	2.304	2.816	2.560	18.945
	5	2.475	3.499	3.157	2.048	2.219	2.816	2.219	18.433
	6	2.304	3.243	2.901	2.133	2.304	2.645	2.219	17.749

	7	2.304	2.816	2.731	1.963	2.048	2.475	2.048	16.385
	8	2.219	3.243	2.645	2.048	2.219	2.645	2.048	17.067
	9	2.475	2.816	2.901	2.133	2.304	2.645	2.133	17.407
	10	2.219	3.328	2.816	2.389	2.219	2.731	2.304	18.006
P4 - n	1	2.048	3.072	2.901	2.219	2.219	2.645	2.304	17.408
	2	2.389	2.731	2.816	2.645	2.304	2.731	2.048	17.664
	3	2.219	2.901	2.901	2.389	1.963	2.816	2.389	17.578
	4	2.389	2.901	2.645	2.304	2.133	2.901	2.304	17.577
	5	2.560	2.901	2.816	2.389	2.133	2.389	2.133	17.321
	6	2.219	2.731	2.560	2.304	1.963	2.475	2.133	16.385
	7	2.219	2.987	2.645	2.219	1.877	2.731	1.877	16.555
	8	2.219	2.987	2.731	2.133	1.963	2.645	2.048	16.726
	9	2.133	3.243	2.816	2.389	2.304	2.816	1.877	17.578
	10	2.219	2.987	2.560	2.304	1.963	2.901	2.133	17.067
P4 - t	1	2.560	3.328	2.645	2.304	2.304	2.731	2.475	18.347
	2	2.475	3.328	2.731	2.304	2.560	2.731	2.560	18.689
	3	2.816	3.499	2.560	2.133	2.133	2.901	2.475	18.517
	4	2.731	3.243	2.731	1.963	2.048	2.987	2.475	18.178
	5	2.475	3.328	2.731	1.877	2.219	2.816	2.389	17.835
	6	2.475	3.328	2.560	2.048	2.219	2.645	2.304	17.579
	7	2.560	3.584	2.731	2.048	2.219	2.645	2.304	18.091
	8	2.645	3.072	2.645	1.877	2.219	2.901	2.389	17.748
	9	2.731	3.413	2.304	2.048	2.133	2.901	2.475	18.005
	10	2.560	3.413	2.645	1.877	2.304	2.816	2.219	17.834
P5 - f	1	2.048	2.645	2.048	2.219	1.877	2.219	2.219	15.275
	2	2.304	2.731	2.219	1.963	2.133	1.963	2.219	15.532
	3	2.133	2.560	2.219	2.048	2.219	2.133	2.219	15.531
	4	2.219	3.157	2.219	1.963	1.963	1.963	2.133	15.617
	5	2.133	2.816	2.475	1.963	2.133	2.048	2.304	15.872
	6	2.304	2.901	2.304	1.877	2.219	2.133	2.133	15.871
	7	2.219	3.157	2.304	2.048	2.133	2.133	2.048	16.042
	8	2.133	3.499	2.475	1.792	2.219	2.133	2.048	16.299
	9	2.304	3.328	2.475	1.877	2.048	2.048	1.792	15.872
	10	2.133	3.413	2.389	2.048	2.133	2.048	1.877	16.041
P5 - n	1	1.877	2.816	2.560	2.048	1.877	2.133	1.621	14.932
	2	1.792	2.560	2.560	2.048	1.877	2.304	1.621	14.762
	3	1.963	2.645	2.304	2.048	1.877	2.048	1.451	14.336
	4	1.877	2.560	2.389	1.877	1.707	2.133	1.365	13.908
	5	1.877	2.560	2.133	1.963	1.792	2.048	1.792	14.165
	6	2.048	2.560	2.219	1.707	1.963	2.304	1.536	14.337
	7	2.048	2.560	2.219	1.792	1.877	2.219	2.304	15.019
	8	1.707	2.731	2.560	1.792	1.877	2.219	1.621	14.507
	9	1.707	2.731	2.133	1.792	1.707	2.219	1.621	13.910
	10	1.963	2.901	2.219	1.877	1.792	2.389	1.707	14.848
P5 - t	1	2.133	3.072	2.645	2.475	2.048	2.389	2.560	17.322
	2	2.389	3.328	2.816	1.963	2.048	2.133	2.304	16.981
	3	2.475	3.072	2.389	2.048	2.048	2.304	2.560	16.896
	4	2.219	2.901	2.304	2.048	1.877	2.219	2.219	15.787
	5	2.304	2.816	2.219	1.877	1.963	2.389	2.048	15.616
	6	2.475	2.901	2.560	1.792	2.048	2.389	2.048	16.213

	7	2.389	3.157	2.389	1.792	2.048	2.475	2.048	16.298
	8	2.389	3.328	2.645	1.792	2.048	2.475	2.133	16.810
	9	2.389	3.072	2.560	1.963	1.963	2.304	2.048	16.299
	10	2.560	3.243	3.157	1.792	2.048	2.560	2.133	17.493
P6 - f	1	2.133	3.499	2.304	2.219	2.219	2.475	1.877	16.726
	2	2.389	3.499	2.219	2.133	2.219	2.048	1.877	16.384
	3	2.389	3.157	2.389	1.877	2.389	2.133	1.963	16.297
	4	1.792	2.901	2.304	1.963	2.133	2.048	2.048	15.189
	5	2.133	3.243	2.133	1.792	2.048	2.048	2.048	15.445
	6	2.219	3.157	2.389	1.963	2.133	1.963	2.219	16.043
	7	2.133	3.243	2.133	2.133	2.219	1.963	1.877	15.701
	8	2.304	2.901	2.304	2.133	2.048	2.048	2.048	15.786
	9	2.133	3.328	2.219	2.048	2.219	2.133	2.133	16.213
	10	2.219	3.072	2.133	1.963	2.133	1.707	1.877	15.104
P6 - n	1	1.963	2.560	1.963	1.707	1.792	2.219	1.792	13.996
	2	1.877	2.901	2.645	2.133	1.792	2.048	2.133	15.529
	3	1.963	2.389	2.389	1.963	1.792	2.304	1.707	14.507
	4	1.792	2.645	2.475	1.707	1.877	2.475	1.877	14.848
	5	1.536	2.901	2.645	1.707	1.877	2.304	1.621	14.591
	6	1.621	2.645	2.731	1.707	1.963	2.133	1.792	14.592
	7	1.792	2.560	2.731	1.707	1.792	2.304	1.707	14.593
	8	1.792	2.645	2.304	1.707	1.877	2.389	1.792	14.506
	9	2.048	2.645	2.133	1.707	1.877	2.219	1.707	14.336
	10	1.792	2.816	2.304	2.048	1.792	2.475	1.621	14.848
P6 - t	1	2.560	3.243	2.560	1.963	2.048	2.304	2.475	17.153
	2	2.475	2.816	2.475	1.963	2.133	2.048	2.048	15.958
	3	2.219	3.243	2.645	1.963	1.877	2.389	2.048	16.384
	4	2.219	3.157	2.475	1.877	1.877	2.048	2.219	15.872
	5	2.389	3.157	2.475	1.792	1.877	2.304	2.219	16.213
	6	2.219	3.157	2.731	1.963	1.963	2.304	2.048	16.385
	7	2.389	3.157	2.560	2.048	1.877	2.389	2.048	16.468
	8	2.389	2.987	2.645	2.389	1.877	1.963	2.133	16.383
	9	2.731	3.243	2.731	1.707	1.792	2.560	2.304	17.068
	10	2.219	3.072	2.816	1.877	1.792	2.475	2.304	16.555
P7 - f	1	1.963	2.475	1.877	1.877	1.963	1.792	2.731	14.678
	2	2.048	2.645	1.792	1.792	2.048	2.133	1.536	13.994
	3	2.219	2.560	1.792	1.877	2.133	1.707	1.707	13.995
	4	2.304	2.645	2.048	1.707	1.963	1.621	1.365	13.653
	5	2.048	2.816	1.877	1.621	1.707	1.963	1.792	13.824
	6	1.963	2.901	1.792	1.707	1.963	1.877	1.792	13.995
	7	2.219	2.475	1.707	1.707	1.877	1.792	1.792	13.569
	8	2.048	3.328	1.707	2.219	1.621	1.963	2.219	15.105
	9	2.219	2.645	1.877	1.792	1.707	2.048	1.877	14.165
	10	2.219	2.816	1.877	1.707	1.792	2.304	1.792	14.507
P7 - n	1	1.621	2.304	2.048	1.621	1.792	1.792	1.451	12.629
	2	1.877	2.645	2.048	1.536	1.877	1.877	1.621	13.481
	3	1.792	2.645	2.133	1.877	1.707	1.792	1.451	13.397
	4	1.963	2.475	1.877	1.792	1.792	2.133	1.536	13.568
	5	1.707	2.816	1.877	1.877	1.707	1.963	1.792	13.739
	6	1.877	2.304	2.048	1.792	1.792	2.304	1.877	13.994

	7	1.877	2.645	2.048	1.621	1.792	2.048	1.707	13.738
	8	1.877	2.389	2.048	1.451	1.621	2.133	1.536	13.055
	9	2.048	2.389	2.048	1.707	1.707	2.219	1.365	13.483
	10	1.707	2.645	1.877	1.792	1.963	2.219	1.451	13.654
P7 - t	1	1.792	2.560	2.219	1.707	1.536	2.304	1.877	13.995
	2	1.877	2.987	2.389	1.792	1.707	2.219	1.877	14.848
	3	1.792	2.901	2.219	1.451	1.707	2.133	2.133	14.336
	4	2.133	3.243	2.048	1.707	1.707	1.963	1.963	14.764
	5	1.877	2.731	2.389	1.451	1.792	1.963	2.048	14.251
	6	2.219	2.560	2.048	1.536	1.707	2.048	1.792	13.910
	7	2.389	2.731	2.304	1.365	1.621	1.963	1.963	14.336
	8	2.133	2.987	2.475	1.707	1.877	2.048	1.707	14.934
	9	2.048	2.645	2.219	1.877	1.877	1.963	1.707	14.336
	10	1.963	2.645	2.560	1.707	1.707	2.048	1.707	14.337
P8 - f	1	2.133	2.389	2.048	1.792	1.707	1.963	2.048	14.080
	2	1.963	2.389	2.048	1.877	1.792	1.963	1.963	13.995
	3	1.707	2.645	1.963	1.621	1.536	1.877	1.707	13.056
	4	2.048	2.560	1.963	1.877	1.792	1.451	1.792	13.483
	5	1.707	2.219	2.048	2.048	1.707	1.621	1.877	13.227
	6	1.707	2.475	1.792	2.048	1.707	1.707	2.048	13.484
	7	1.877	2.389	1.792	1.621	1.792	1.707	1.877	13.055
	8	1.707	2.389	1.963	1.621	1.621	1.963	1.877	13.141
	9	1.963	2.475	2.219	1.451	1.707	1.963	1.792	13.57
	10	1.877	2.816	2.219	1.536	1.792	1.963	1.621	13.824
P8 - n	1	1.792	2.475	2.048	1.621	1.877	1.963	1.707	13.483
	2	1.707	2.389	2.219	1.536	1.707	1.707	1.451	12.716
	3	1.621	2.560	2.133	1.621	1.877	1.963	1.451	13.226
	4	1.792	2.560	2.048	1.621	1.707	2.219	1.365	13.312
	5	1.877	2.475	2.048	1.621	1.536	2.048	1.536	13.141
	6	1.877	2.475	2.219	1.707	1.536	2.048	1.792	13.654
	7	1.536	2.645	1.792	1.707	1.707	1.963	1.536	12.886
	8	1.707	2.560	2.048	1.536	1.792	2.048	1.621	13.312
	9	1.621	2.645	2.133	1.621	1.536	2.475	1.621	13.652
	10	1.792	2.645	1.792	1.707	1.707	2.133	1.451	13.227
P8 - t	1	1.963	2.816	2.389	2.133	1.877	2.133	1.621	14.932
	2	2.048	2.901	2.389	1.707	1.792	2.133	1.536	14.506
	3	2.133	2.987	2.475	1.792	1.792	2.048	1.877	15.104
	4	2.219	2.987	2.219	1.621	1.707	1.963	1.877	14.593
	5	1.963	2.987	2.560	1.792	1.877	2.133	1.792	15.104
	6	1.877	2.816	2.475	1.707	1.792	2.219	1.792	14.678
	7	1.963	2.987	2.475	1.877	1.963	2.304	1.451	15.02
	8	2.048	2.901	2.560	1.536	1.963	2.219	1.792	15.019
	9	2.048	2.987	2.645	1.707	1.963	2.048	1.707	15.105
	10	2.048	2.987	2.389	1.621	1.707	1.877	1.792	14.421
P9 - f	1	1.451	2.560	1.792	1.621	1.621	1.621	2.048	12.714
	2	2.048	2.645	1.963	1.792	1.707	1.621	1.963	13.739
	3	1.792	2.304	1.792	1.707	1.792	1.792	1.707	12.886
	4	1.451	2.133	1.792	2.048	1.792	1.877	1.621	12.714
	5	2.048	2.219	1.621	1.877	1.707	1.963	1.877	13.312
	6	2.048	2.389	1.877	1.792	1.621	2.048	1.963	13.738

	7	2.048	2.389	1.792	1.792	1.792	2.133	1.792	13.738
	8	1.536	2.389	1.877	1.792	1.707	1.536	2.048	12.885
	9	1.877	2.389	1.963	1.707	1.451	1.877	1.963	13.227
	10	1.707	2.560	1.877	1.536	1.707	1.707	2.048	13.142
P9 - n	1	1.707	2.645	1.877	1.536	1.707	2.048	1.621	13.141
	2	1.707	2.731	1.963	1.621	1.707	2.048	1.451	13.228
	3	1.621	2.475	1.877	1.536	1.877	1.536	1.536	12.458
	4	1.621	2.560	1.792	2.048	1.707	1.621	1.536	12.885
	5	1.536	2.304	1.792	1.621	1.707	1.963	1.707	12.630
	6	1.877	2.389	1.877	2.133	1.707	1.707	2.133	13.823
	7	1.792	2.475	1.792	1.621	1.877	1.963	1.792	13.312
	8	1.451	2.560	1.792	1.536	2.133	1.877	1.707	13.056
	9	1.792	2.731	1.877	1.365	1.451	1.963	1.621	12.800
	10	1.707	2.731	1.792	1.792	1.536	1.877	1.707	13.142
P9 - t	1	1.963	2.901	2.816	1.963	1.792	1.536	1.792	14.763
	2	2.389	3.328	2.901	1.792	1.792	1.707	1.707	15.616
	3	2.133	2.645	2.816	2.133	1.707	1.621	1.621	14.676
	4	1.963	2.560	2.645	1.707	1.877	1.707	2.048	14.507
	5	1.963	2.560	2.475	1.536	1.707	2.048	1.792	14.081
	6	2.048	2.901	2.901	1.621	1.621	1.963	1.792	14.847
	7	1.877	2.731	2.645	1.451	1.621	1.877	2.048	14.25
	8	1.877	2.987	2.475	1.536	1.621	1.536	1.792	13.824
	9	1.792	2.987	2.560	1.621	1.621	2.304	1.963	14.848
	10	2.133	3.072	2.901	1.707	1.707	2.048	1.792	15.360
P10 - f	1	2.048	2.816	2.048	2.304	2.133	2.048	1.963	15.360
	2	2.133	2.560	2.133	1.963	2.133	2.133	2.048	15.103
	3	2.304	2.389	1.963	2.389	2.048	2.133	1.877	15.103
	4	2.219	2.731	1.963	2.475	1.963	2.048	1.877	15.276
	5	2.048	2.645	1.792	2.475	1.877	1.792	2.048	14.677
	6	2.475	2.816	1.877	2.219	1.792	2.048	1.877	15.104
	7	2.304	2.731	1.877	2.560	1.963	2.389	1.877	15.701
	8	2.475	2.560	1.877	2.645	2.048	1.707	1.963	15.275
	9	2.389	2.901	2.048	2.731	1.963	2.133	1.963	16.128
	10	2.219	2.987	1.877	2.304	1.963	2.133	1.963	15.446
P10 - n	1	1.963	2.475	1.963	2.133	1.963	2.475	2.133	15.105
	2	2.048	2.816	2.048	2.048	1.963	2.133	2.048	15.104
	3	1.877	2.816	1.963	1.963	1.963	2.048	2.048	14.678
	4	2.048	2.645	1.963	1.707	1.877	1.963	2.219	14.422
	5	1.963	2.901	2.048	2.389	1.877	2.048	2.133	15.359
	6	2.133	2.816	1.877	1.621	1.792	2.219	1.963	14.421
	7	2.133	2.816	1.963	1.707	1.707	2.048	2.048	14.422
	8	1.963	3.157	1.963	1.621	1.877	2.133	1.877	14.591
	9	1.877	2.731	1.877	1.536	1.877	2.048	1.877	13.823
	10	2.133	3.157	1.963	1.963	1.877	2.048	1.792	14.933
P10 - t	1	2.219	3.755	2.389	1.792	2.048	2.219	1.963	16.385
	2	2.304	3.328	2.389	2.219	1.707	2.304	1.877	16.128
	3	2.304	3.243	2.560	1.963	1.877	1.792	2.133	15.872
	4	2.133	3.413	2.475	2.133	1.877	2.304	1.877	16.212
	5	2.304	3.157	2.560	2.048	1.877	2.219	2.133	16.298
	6	2.304	3.157	2.475	1.963	1.963	1.963	1.963	15.788

7	2.048	3.072	2.475	2.304	1.792	2.133	1.963	15.787
8	2.133	2.901	2.560	2.048	1.963	2.048	2.133	15.786
9	2.389	3.584	2.475	1.963	1.792	2.219	1.877	16.299
10	2.304	2.987	2.645	2.133	1.877	2.304	1.963	16.213
Total duration [s]								4713.3 91

Tabelul 4.1 – Durata fiecărei instanțe

După cum putem observa în tabel, propozițiile au fost alese în asemenea fel încât toate să aibă lungime asemănătoare. Putem observa că în medie persoana cu numărul 2 a avut cele mai lungi propoziții din punct de vedere al timpului, iar persoana cu numărul 7 a avut cele mai scurte propoziții din punct de vedere al timpului. Acest lucru se poate observa și în Fig. 4.2 de mai jos.

Instance time with respect to sentence

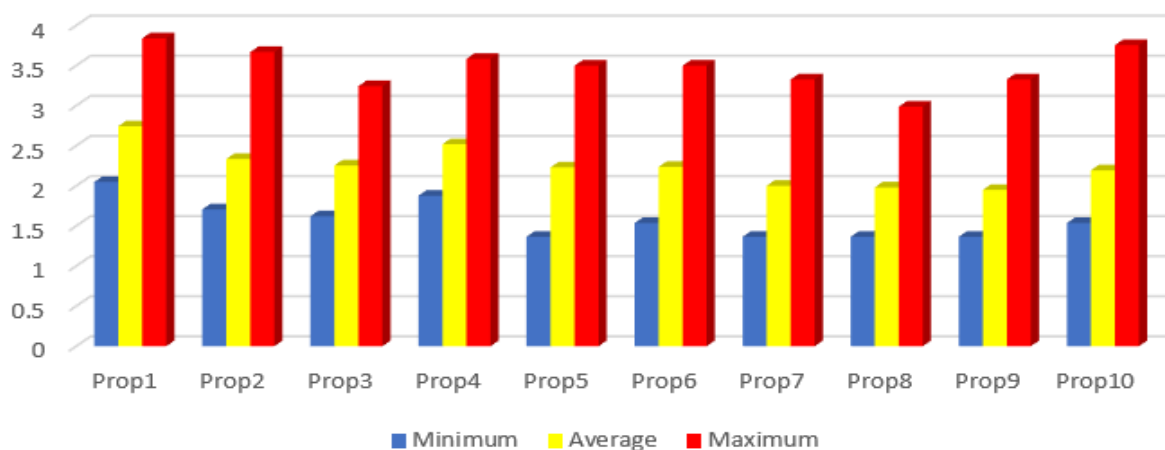


Fig. 4.2 – Valorile minime, medii și maxime a instanțelor de timp în funcție de propoziție

Minimul este reprezentat cu albastru, media este reprezentată cu galben, iar maximul este reprezentat cu roșu. Se poate observa că propoziția numărul 1(A venit rezultatul testului) are cea mai mare valoare minimă de 2.048 secunde, fiind și singura care are o valoare minimă de peste 2 secunde. Cea mai mică valoare minimă o au propozițiile 5(A crescut foarte mult), 7(Li inchisa usa), 8(Li stins becul) și 9(Sunt foarte bine), cu o valoare de 1.365 secunde. Cea mai mare valoare medie îi aparține tot propoziției numărul 1, cu o valoare de 2.743 secunde. Singurele 2 propoziții care au durata medie de sun 2 secunde sunt propoziția 8 și propoziția 9. Este probabil ca propozițiile 8 și 9 să fie destul de uzuale pentru vorbitori și de aceea să aibă o medie a duratei de sub 2 secunde. Propoziția numărul 1 are și cel mai mare maxim al duratei, acesta fiind de 3.84 secunde. Cea mai mică și singura valoare maximă care se află la sub 3 secunde este propoziția numărul 8(). Acest lucru poate fi dat și de exprimarea colocvială a acesteia.

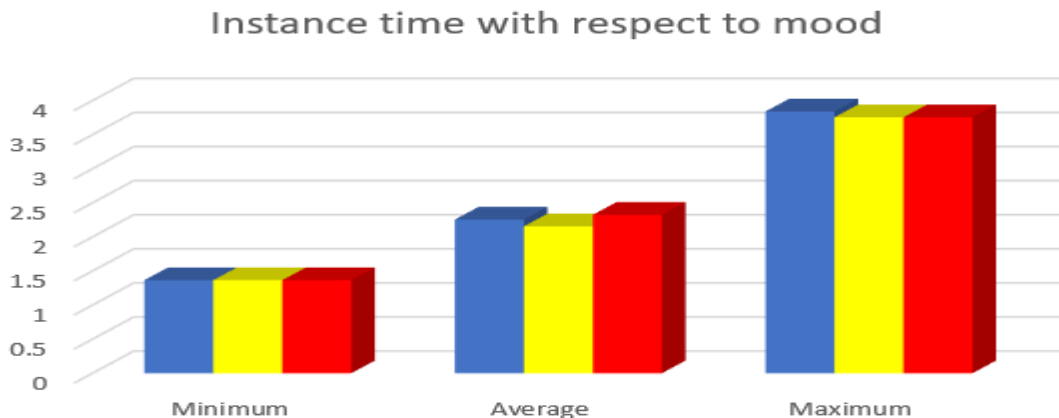


Fig. 4.3 – Valorile minime, medii și maxime a instanțelor de timp în funcție de emoția transmisă.

În Fig. 4.3 de mai sus putem vedea lungimile propozițiilor în funcție de frază. Emoția fericită este reprezentată cu albastru, emoția neutră este reprezentată cu galben, iar emoția tristă este reprezentată cu roșu. Chiar dacă ne așteptam ca propozițiile triste să fie mult mai lungi ca cele fericite sau neutre, putem observa că acestea au o lungime aproximativ egală. Dacă luăm în considerare minimul, toate cele 3 emoții sunt la egalitate cu o lungime a propozițiilor de 1.365 secunde. Când luăm în calcul lungimea medie, emoția tristă este cea mai lungă, cu o durată de 2.324 secunde urmată de emoția fericită cu 2.254 secunde și în final cea neutră cu 2.155 secunde. Cea mai lungă valoare maximă aparține emoției fericite, cu o lungime de 3.84 secunde, fiind urmată de cele 2 emoții rămase neutru și trist cu 3.755 secunde. După această analiză, putem spune că nu ne putem lua după durata unei propoziții ca să o categorisim după emoție

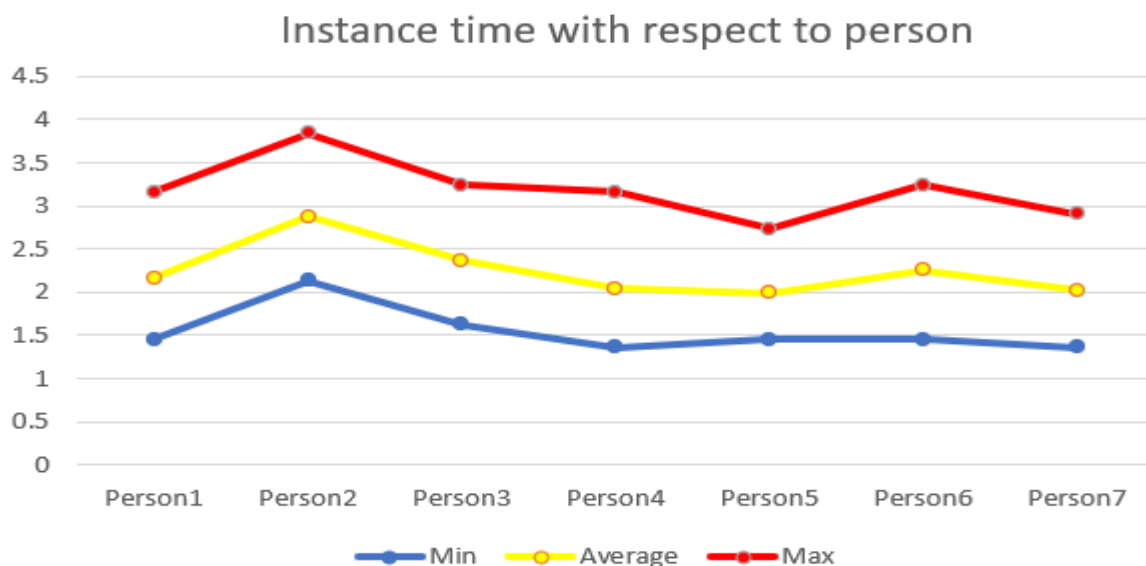


Fig. 4.4 – Valorile minime, medii și maxime a instanțelor de timp în funcție de persoană/vorbitor

Din Fig. 4.4 de mai sus putem analiza în funcție de persoane minimul, media și maximul. Minimul este reprezentat cu albastru, media este reprezentată cu galben, iar maximul este reprezentat cu roșu. Putem observa că atât valoarea minimă, cât și cea medie și cea maximă au valorile cele mai ridicate pentru persoana numărul 2 cu valorile corespunzând minimului mediei și maximului: 2.133 secunde, 2.876 secunde și 3.84 secunde. Valoarea cea mai mică pentru durata minimă a unei propoziții o are persoana 4 și persoana 7 cu 1.365 secunde. Valoarea cea mai mică și singura care coboară sub 2 secunde a valorii medii, aparține

persoanei cu numărul 5. Minimul duratei maxime aparține tot persoanei cu numărul 5, cu o valoare de 2.731 secunde.

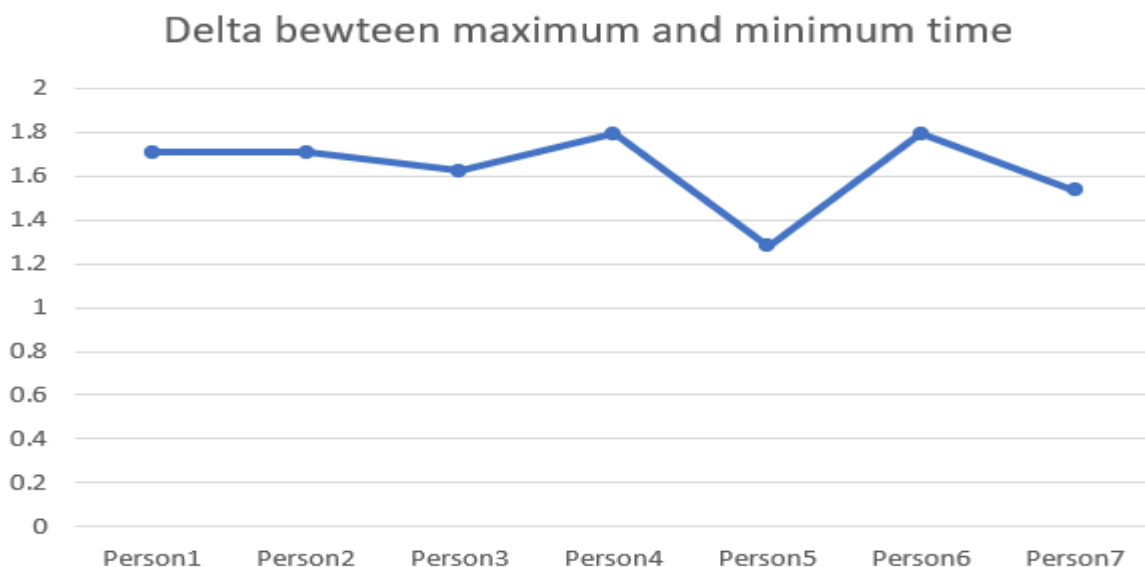


Fig. 4.5 – Diferența dintre valoarea maximă și minimă a instanțelor de timp în funcție de persoană/vorbitor

În Fig. 4.5 de mai sus este explicată diferența dintre valoarea maximă și cea minimă a duratei propozițiilor pentru toate persoanele. Există o diferență destul de mare deoarece propoziția numărul 1 este mai lungă decât propoziția numărul 9 care în medie este cea mai scurtă. Putem vedea că pentru persoanele 1, 2, 3, 4 și 6 diferența de timp este în marja de 1.6 secunde și 1.8 secunde. Persoana numărul 7 are un delta de 1.536 secunde, dar persoana cu cel mai mic delta este persoana cu numărul 5, aceasta având o deviație de doar 1.28 secunde.

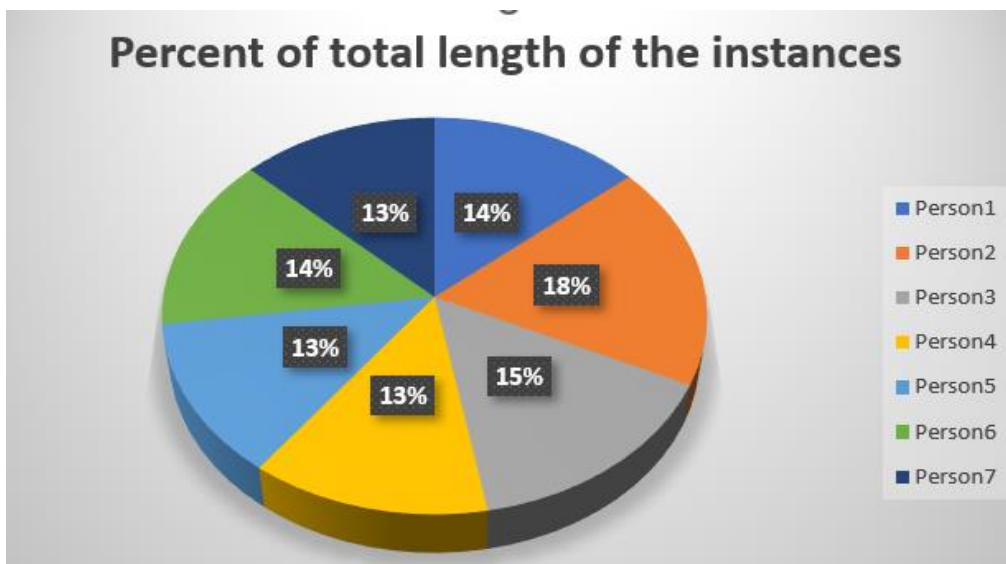


Fig. 4.6 – Timpul total al propozițiilor în funcție de persoane reprezentat în procente

În Fig. 4.6 reprezentată mai sus, avem reprezentat totalul de secunde pentru fiecare persoană reprezentat în procente. Putem observa că procentele sunt aproximativ egale pentru toate persoanele cu excepția persoanei cu numărul 2, care are cu 3 procente mai mult decât următoarea persoană.

Implementare

Python este un limbaj de programare dinamic multi-paradigmă, creat în 1989 de programatorul olandez Guido van Rossum. Van Rossum este și în ziua de astăzi un lider al comunității de dezvoltatori de software care lucrează la perfecționarea limbajului Python și implementarea de bază a acestuia, CPython, scrisă în C. Python este un limbaj multifuncțional folosit de exemplu de către companii ca Google sau Yahoo! pentru programarea aplicațiilor web, însă există și o serie de aplicații științifice sau de divertisment programate parțial sau în întregime în Python. Popularitatea în creștere, dar și puterea limbajului de programare Python au dus la adoptarea sa ca limbaj principal de dezvoltare de către programatori specializați și chiar și la predarea limbajului în unele medii universitare. Din aceleași motive, multe sisteme bazate pe Unix, inclusiv Linux, BSD și Mac OS X includ din start interpretatorul CPython.

Python pune accentul pe curățenia și simplitatea codului, iar sintaxa sa le permite dezvoltatorilor să exprime unele idei programatice într-o manieră mai clară și mai concisă decât în alte limbaje de programare ca C. În ceea ce privește paradigma de programare, Python poate servi ca limbaj pentru software de tipul object-oriented, dar permite și programarea imperativă, funcțională sau procedurală. Sistemul de tipizare este dinamic iar administrarea memoriei decurge automat prin intermediul unui serviciu „gunoier” (garbage collector). Alt avantaj al limbajului este existența unei ample biblioteci standard de metode. Implementarea de referință a Python este scrisă în C și poartă deci numele de CPython. Această implementare este software liber și este administrată de fundația Python Software Foundation [1].

Pentru a extrage caracteristicilor trebuie să stabilim care sunt fișierele audio pentru care dorim să le extragem. De exemplu pentru baza de date Crema s-a folosit următorul cod:

```
directory = "D:/Doctorat_Toma/Crema/"
directory2= "D:/Doctorat_Toma/Crema/MFCC/"
# put your own directory here
# directory to put our results in, you can change the name if
you like
for it in os.scandir(directory):
    if it.is_dir():
        directoryName = it.path
        #resultsDirectory = directoryName + "/MPEG"

        # make a new folder in this directory to save our
results in
        #if not os.path.exists(resultsDirectory):
            #os.makedirs(resultsDirectory)

        # MFCCs for every .wav file in our specified directory
.csv SAVE
        for filename in os.listdir(directoryName):
            if filename.endswith('.wav'): # only get MFCCs
from .wavs
                # read in our file
```

```

        (rate, sig) = wav.read(directoryName + "/" +
filename)
        directoryName_final = directoryName + "/" +
filename

```

unde `directory` definește unde este definite baza de date iar `directory2` definește fișierul unde caracteristicile vor fi extrase. Pentru a identifica toate fișierele de tip “.wav”, vedem că folosim funcția `.endswith` ca să putem căuta după tipul fișierului necesar.

Pentru MFCC folosim pachetul Python numit “python_speech_features” de unde de unde alegem funcția `mfcc`. Această funcție are următorii parametrii:

`sig` – care este semnalul folosit pentru extragerea caracteristicilor

`Fs` – frecvența dorită în Hz

`Wl` – sau window length, este lungimea ferestrei de analiză măsurată în secunde

`Ws` – sau window step, este pasul între ferestre succesive măsurate în secunde.

`L` – este numărul de coeficienți MFCC

`M` – este numărul de filterbanks,

`N_fft` = este mărimea transformatei Fourier rapide.

`LF` – este frecvența minimă

`HF` – este frecvența maximă

`alpha` – este coeficientul de preaccentuare aplicat filtrului de preaccentuare

`Lf` – numărul de parametrii cepstrali

Astfel că funcția MFCC va arăta ca și în următorul exemplu:

```

python_speech_features.base.mfcc(sig, Fs, winlen=Wl, winstep=Ws,
numcep=L, nfilt=M, nfft=N_fft, lowfreq=LF, highfreq=HF,
preemph=alpha, ceplifter=Lf, appendEnergy=True,
winfunc=numpy.hamming)

```

Pentru a extrage coeficienții bazei de date Crema, trebuie văzut cum este alcătuită aceasta și trebuie să grupăm emoțiile în clase diferite. De exemplu, în fiecare nume al propozițiilor din baza de date Crema de la poziția 39 până la poziția 42 avem o grupare de 3 litere care pot fi ‘ANG’, ‘DIS’, ‘FEA’, ‘HAP’, ‘NEU’ sau ‘SAD’ care corespund următoarelor emoții: nervozitate, dezgust, frică, fericire, neutru și tristețe. Următorul cod a fost folosit pentru a gruparea emoțiilor în clasele corespunzătoare:

```

if directoryName_final[39:42] == 'ANG':
    classNo = 1
    if directoryName_final[39:42] == 'DIS':
        classNo = 2
    if directoryName_final[39:42] == 'FEA':
        classNo = 3
    if directoryName_final[39:42] == 'HAP':
        classNo = 4
    if directoryName_final[39:42] == 'NEU':
        classNo = 5
    if directoryName_final[39:42] == 'SAD':
        classNo = 6

```

Pentru alte baze de date, abordări asemănătoare au fost folosite, de exemplu pentru baza de date Savee, de la poziția 33 până la poziția 35 avem o grupare de 1 sau 2 litere care pot fi ‘a’, ‘d’, ‘f’, ‘h’, ‘n’, ‘sa’ sau ‘su’ care corespund următoarelor emoții: nervozitate, dezgust, frică, fericire, neutru, tristețe și surprindere.

```

if directoryName_final[33:34] == 'a':

```

```

        classNo = 1
    if directoryName_final[33:34] == 'd':
        classNo = 2
    if directoryName_final[33:34] == 'f':
        classNo = 3
    if directoryName_final[33:34] == 'h':
        classNo = 4
    if directoryName_final[33:34] == 'n':
        classNo = 5
    if directoryName_final[33:35] == 'sa':
        classNo = 6
    if directoryName_final[33:35] == 'su':
        classNo = 7

```

Pentru baza de date Emo-DB, de la poziția 33 până la poziția 34 avem câte 1 literă care poate fi 'F', 'W', 'L', 'E', 'A', 'T' sau 'N' care sunt din limba germană și corespund următoarelor emoții: fericire, furie, plictiseală, dezgust, anxietate/teamă, tristețe și neutru.

```

if directoryName_final[33:34] == 'F':
    classNo = 1
    if directoryName_final[33:34] == 'W':
        classNo = 2
    if directoryName_final[33:34] == 'L':
        classNo = 3
    if directoryName_final[33:34] == 'E':
        classNo = 4
    if directoryName_final[33:34] == 'A':
        classNo = 5
    if directoryName_final[33:34] == 'T':
        classNo = 6
    if directoryName_final[33:34] == 'N':
        classNo = 7

```

Pentru baza de date Ravdess, numărul clasei este în titlul sunetului și se găsește de la poziția 57 până la 59. Aceasta este tradus în cod în următorul fel:

```
classNo = int(directoryName_final[57:59])
```

După cum știm, baza de date în limba română are doar 3 emoții: fericire, neutru și tristețe. Pentru a crea clasele pentru aceste 3 emoții, vom folosi tot numele fișierelor unde emoția se găsește de la poziția 39 la 40 unde 'f' reprezintă emoția de fericire, 'n' reprezintă emoția neutră, 't' reprezintă emoția de tristețe. Codul pentru acestea va fi:

```

if directoryName_final[39:40] == 'f':
    classNo = 1
    if directoryName_final[39:40] == 'n':
        classNo = 2
    if directoryName_final[39:40] == 't':
        classNo = 3

```

Acum că am văzut cum sunt organizate toate bazele de date, să ne mutăm atenția asupra extractorilor de caracteristici rămași. Pentru a folosi LPC am importat pachetul „spafe”. Coeficienții necesari pentru LPC sunt asemănători cu cei folosiți de MFCC. Coeficienții sunt: sig, Fs, Wl, Ws, L unde acum este ordinul modelului LPC și alpha. Codul în python va fi:

```
lpc_coef = lpc(sig, Fs, order=L, win_len=Wl, win_hop=Ws,
pre_emph_coeff=alpha, win_type=numpy.hamming)
```

Pentru a folosi GFCC am importat pachetul „spafe”. Coeficienții necesari pentru GFCC sunt asemănători cu cei folosiți de MFCC. Coeficienții sunt: sig, Fs, Wl, Ws, L unde acum este ordinul modelului GFCC, M, N_fft, LF, HF, Lf, alpha și D care este tipul de transformare cosinus discretă este folosită. Codul în python va fi:

```
gfcc_coef = gfcc(sig, Fs, win_len=Wl, win_hop=Ws, num_ceps=L,
nfilters=M, nfft=N_fft, low_freq=LF, high_freq=HF,
pre_emph_coeff=alpha, lifter=Lf, use_energy=True,
win_type=numpy.hamming, dct_type=D)
```

Pentru a folosi MSRCC am importat pachetul „spafe”. Coeficienții necesari pentru MSRCC sunt asemănători cu cei folosiți de GFCC. Coeficienții sunt: sig, Fs, Wl, Ws, L unde acum este ordinul modelului MSRCC M, N_fft, LF, HF, Lf, alpha și D care este tipul de transformare cosinus discretă este folosită. Codul în python va fi:

```
msrcc_coef = msrcc(sig, Fs, win_len=Wl, win_hop=Ws, num_ceps=L,
nfilters=M, nfft=N_fft, low_freq=LF, high_freq=HF,
pre_emph_coeff=alpha, lifter=Lf, use_energy=True,
win_type=numpy.hamming, dct_type=D)
```

Pentru a folosi NGCC am importat pachetul „spafe”. Coeficienții necesari pentru NGCC sunt asemănători cu cei folosiți de GFCC și MSRCC. Coeficienții sunt: sig, Fs, Wl, Ws, L unde acum este ordinul modelului NGCC, M, N_fft, LF, HF, Lf, alpha și D care este tipul de transformare cosinus discretă este folosită. Codul în python va fi:

```
ngcc_coef = ngcc(sig, Fs, win_len=Wl, win_hop=Ws, num_ceps=L,
nfilters=M, nfft=N_fft, low_freq=LF, high_freq=HF,
pre_emph_coeff=alpha, lifter=Lf, use_energy=True,
win_type=numpy.hamming, dct_type=D)
```

Dacă ne uităm la caracteristica mel, am folosit pachetul “librosa” de unde am folosit funcția melspectrogram care are următorii coeficienți: sig, Fs, N_fft, hop_length care este numărul de sample între 2 cadre, Wl, Wt fiind tipul ferestrei folosite, pad_mode fiind tipul de atenuare a sunetului pe marginile cadrului, power fiind exponentul magnitudinii spectogramei mel și n_mels fiind numărul de sample care sunt extrase pentru a crea caracteristica spectogramei mel. Codul în python va fi:

```
mel_coef = melspectrogram(y=sig, sr=Fs, S=None, n_fft=2048,
hop_length=512, win_length=None, window='hann', center=True,
pad_mode='constant', power=2.0, n_mels=128)
```

Pentru a ajunge la caracteristica chroma, am folosit pachetul “librosa” de unde am folosit funcția chroma_stft care are următorii coeficienți: sig, Fs, N_fft, hop_length, Wl, Wt, pad_mode, și n_chroma fiind numărul de puncte care sunt extrase pentru a crea caracteristica spectogramei chroma. Codul în python va fi:

```
chroma_coef = chroma_stft(y=sig, sr=Fs, S=None, norm=numpy.inf,
n_fft=2048, hop_length=512, win_length=None, window='hann',
center=True, pad_mode='constant', tuning=None, n_chroma=12)
```

Pentru a calcula la caracteristica frecvenței fundamentale F0, am folosit pachetul “pandas” de unde am folosit funcția yin care are următorii coeficienți: wav_data fiind numele fișierului, sr fiind frecvența de eșantionare, fmin și fmax, fiind frecvența minimă și frecvența maximă. Codul în python va fi:

```
f0 = librosa.yin(wav_data, sr=48000, fmin=50, fmax=2100)
```

Pentru F0, pentru fiecare bază de date s-a făcut o diferențiere pentru sexul vorbitorului, pentru a vedea mai bine diferențele între vorbitorii de gen masculin și de gen feminin și diferențele dintre cele două categorii. De exemplu pentru baza de date Emo-DB codul va fi:

```
if filename[0:2] == '03':
    gender = 2
    if filename[0:2] == '08':
        gender = 1
    if filename[0:2] == '09':
        gender = 1
    if filename[0:2] == '10':
        gender = 2
    if filename[0:2] == '11':
        gender = 2
    if filename[0:2] == '12':
        gender = 2
    if filename[0:2] == '13':
        gender = 1
```

Până acum am vazut toți extractorii de caracteristici transpuși în codul python. Pentru a avea un rezultat este nevoie și de o clasificare. Pentru asta trebuie să vedem cum clasificatorul *k*NN este implementat. A fost nevoie de pachetul “sklearn” de unde am folosit funcția KNeighborsClassifier cu următorii coeficienți: n_neighbors care reprezintă numărul de vecini, weights, care reprezintă ponderea fiecărui vecin. Leaf_size reprezintă dimensiunea frunzelor. Acest lucru poate afecta viteza de construcție și interogare, precum și memoria necesară pentru a păstra arborele. Valoarea optimă depinde de natura problemei. P reprezintă parametrul de putere pentru metrica Minkowski. Când p = 1, aceasta este echivalentă cu utilizarea distanței manhattan (l1), iar distanța euclidiană (l2) pentru p = 2. Pentru p arbitrar, se folosește distanța minkowski. Metrica se referă la tipul de distanță calculată. Distanța Minkowski este o metrică într-un spațiu vectorial normat, care poate fi considerată o generalizare atât a distanței euclidiene cât și a distanței Manhattan. Este numit după matematicianul german Hermann Minkowski. Codul pentru *k*NN ar arăta în felul următor:

```
KNeighborsClassifier(n_neighbors=5, weights='uniform',
algorithm='auto', leaf_size=30, p=2, metric='minkowski',
metric_params=None, n_jobs=None)
```

Pentru a integra și validarea încrucișată, am folosit funcția cross_val_score pe care am setat-o să facă de 10 ori validarea încrucișată, împărțită în câte 10 eșantioane. Codul pentru aceasta va fi:

```
result_rkf = cross_val_score(estimator=pipe, X=X, y=y,
scoring='accuracy', cv=RepeatedKfold(n_splits=10, n_repeats=10))
```

```
SVC(C=100, kernel='rbf', degree=3, gamma=0.1, coef0=0.0, shrinking=True,
probability=False,
```

```
tol=0.1, cache_size=200, class_weight=None, verbose=False, max_iter=-1,
decision_function_shape='ovr', break_ties=False, random_state=None)
```

Implementarea se bazează pe libsvm. Timpul de potrivire se adaptează cel puțin pătratic cu numărul de eșantioane și poate fi nepractic dincolo de zeci de mii de eșantioane. Suportul multiclasă este gestionat conform unei scheme unu-vs-unu.

Mașinile vector support sunt eficiente în spații de dimensiuni mari. Utilizează un subset de puncte de antrenament în funcția de decizie (numiții vectori de suport), deci este și eficient în memorie. Pot fi specificate diferite funcții Kernel pentru funcția de decizie. Sunt furnizate nuclee comune, dar este și posibil să specificăm nuclee personalizate.

Când antrenăm un SVM cu nucleul Funcției de bază radială (RBF), trebuie luați în considerare doi parametri: C și gamma. Parametrul C, comun tuturor nucleelor SVM, schimbă clasificarea greșită a exemplurilor de antrenament cu simplitatea suprafeței de decizie. Un C scăzut face ca suprafața de decizie să fie netedă, în timp ce un C ridicat vizează clasificarea corectă a tuturor exemplurilor de antrenament. gamma definește cât de multă influență are un singur exemplu de antrenament. Cu cât gama este mai mare, cu atât celelalte exemple trebuie să fie mai apropiate pentru a fi afectate.

Al doilea parametru al funcției specifică tipul de nucleu care va fi utilizat în algoritm 'linear', 'poly', 'rbf', 'sigmoid' sau 'precomputed'. Gradul funcției nucleului polynomial este luat în considerare doar de tipul 'poly'. Coef0 este un termen independent în funcția nucleului, fiind semnificativ doar în tipul 'poly' și 'sigmoid'. Probabilitatea este setată pe 'False' deoarece nu se dorește activarea estimărilor de probabilitate. Criteriul de toleranță la oprire este considerat 0.1. Mărimea memoriei cache a nucleului are un impact puternic asupra timpilor de rulare pentru probleme mai mari. 'class_weight' setează parametrul C al clasei i la $class_weight[i]*C$ pentru SVC (clasificator vector suport). Dacă nu primește valoare, toate clasele ar trebui să aibă greutatea unu. Modul „echilibrat” folosește valorile lui y pentru a ajusta automat ponderile invers proporționale cu frecvențele clasei din datele de intrare ca $n_samples / (n_classes * np.bincount(y))$. Dacă ieșirea verboasă este activată, este posibil ca aceasta să nu funcționeze corect într-un context cu mai multe clase. Funcția de decizie depinde de un anumit subset al datelor de antrenament, numite vectori suport [2].

```
LogisticRegression(C=50, multi_class = 'multinomial', penalty= 'l1', solver = 'saga',
tol=0.1)
```

Este posibil ca unele penalități să nu funcționeze cu unii rezolvatori. Norma pedepsei alese este L1. Regresia lasso (sau regularizarea L1) este o tehnică de regularizare care penalizează coeficienții de valoare mare, corelați. Introduce un termen de regularizare (numit și termen de penalizare) în funcția de pierdere sumei erorilor pătrate (SSE) a modelului. Acest termen de penalizare este valoarea absolută a sumei coeficienților.

Algoritmul utilizat în problema de optimizare este „lbfgs”. Pentru a alege soluția optimă se ia în considerare predicția de tip multiclass și posibilitatea de a intercepta pierderile de tip multinomial, alegându-se astfel 'saga'.

Aceștia sunt câțiva dintre parametrii esențiali pe care îi putem întâlni atunci când utilizăm regresia logistică în scikit-learn. Parametrii reali pot varia ușor în funcție de versiunea specifică a scikit-learn pe care o utilizăm. Pentru a obține cele mai precise și actualizate informații, este întotdeauna o idee bună să consultăm documentația oficială pentru versiunea în care lucrăm [3].

```
O RandomForestClassifier(n_estimators=100,criterion='gini',
max_depth=None, min_samples_split=2,
min_samples_leaf=1, min_weight_fraction_leaf=0.00,
```

```
max_features='auto', max_leaf_nodes=None,
min_impurity_decrease=0.0,
bootstrap=True, oob_score=False,
n_jobs=None, random_state=None, verbose=0, warm_start=False,
class_weight=None, ccp_alpha=0.0, max_samples=None)
```

Random forest este un metaestimator care se potrivește unui număr de clasificatori ai arborelui de decizie pe diferite sub-eșantioane ale setului de date și utilizează media pentru a îmbunătăți acuratețea predictivă și pentru a controla supraajustarea. Arborii pădurii folosesc cea mai bună strategie de împărțire, adică echivalentă cu trecerea splitter="best" la DecisionTreeRegressor de bază. Dimensiunea sub-eșantionului este controlată cu parametrul max_samples dacă bootstrap=True (implicit), în caz contrar, întregul set de date este folosit pentru a construi fiecare arbore.

Numărul de arbori ai pădurii este considerat 100. Funcția de măsurare a calității unei împărțiri a datelor, "criterion" are criteriul 'gini', pentru impuritatea Gini pentru castigul de informații Shannon. Acest parametru este specific arborelui.

Numărul minim de mostre necesare pentru a împărți un nod intern este ales ca fiind 2, iar numărul minim de probe necesare pentru a fi la un nod frunză este 1. Un punct de împărțire la orice adâncime va fi luat în considerare numai dacă lasă cel puțin probe de antrenament min_samples_leaf în fiecare dintre ramurile din stânga și din dreapta. Acest lucru poate avea ca efect netezirea modelului, mai ales în regresie.

Al 6-lea parametru reprezintă fracția minimă ponderată a sumei totale a greutateților (din toate eșantioanele de intrare) necesară pentru a fi la un nod frunză. Mostrele au o greutate egală atunci când parametrul sample_weight nu este furnizat. Numărul de caracteristici de luat în considerare atunci când căutăm cea mai bună împărțire este considerat auto. Cele mai bune noduri sunt definite ca o reducere relativă a impurităților. Este considerat 'None', atunci un număr nelimitat de noduri Frunze sunt disponibile. Un nod va fi divizat dacă această scindare induce o scădere a impurității mai mare sau egală cu valoarea următorului parametru. Mostre bootstrap sunt folosite la construirea arborilor. Numărul de joburi de rulat în paralel este ales 1. Dacă acest parametru 'random_state' este setat, acesta asigură reproductibilitatea prin fixarea semințelor aleatorii pentru generarea numerelor aleatorii.

Acești parametri pot fi ajustați pentru a regla performanța modelului Random Forest pe baza caracteristicilor specifice ale setului de date și a problemei în cauză.

Rezultate

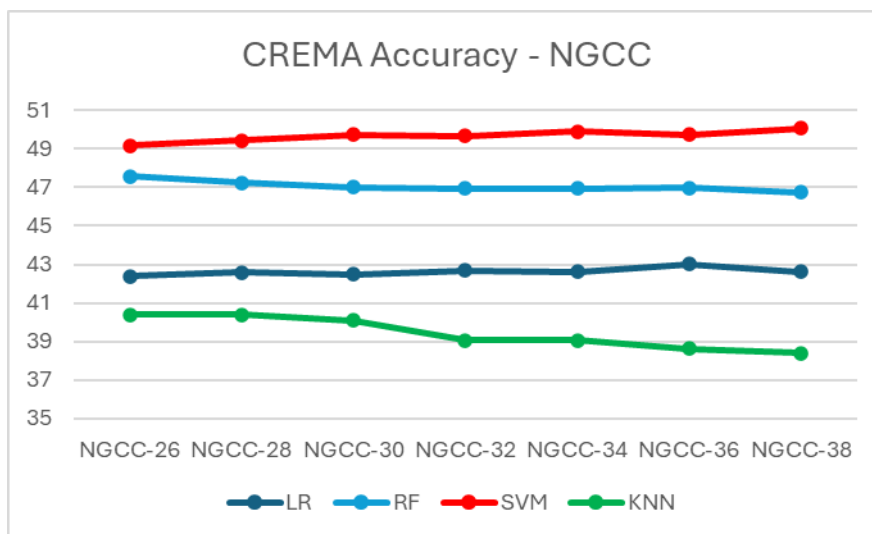


Figura 6.4 – Rata de clasificare corectă pentru baza de date Crema folosind extractorul de caracteristici NGCC.

Din figura 6.4 putem extrage că *kNN* are cea mai mică rată de clasificare corectă cu un maxim de 40.4% pentru 26 și 28 de caracteristici. Minimul pentru *kNN* este de 38.39% pentru 38 de caracteristici. *KNN* reprezintă minimul pentru fiecare număr de caracteristici. *LR* are o pantă aproximativ ascendentă plecând de la un minim de 42.38% pentru 26 de caracteristici și crescând apoi până la un maxim de 43.02% pentru 36 de caracteristici. *LR* are al doilea cel mai slab rezultat pentru fiecare număr de caracteristici. *RF* prezintă un salt de aproximativ 4 procente față de *LR* și ocupă locul 2 în această statistică la cea mai bună rată de clasificare corectă. Rezultatele pentru *LR* reprezintă o pantă descendentă dar cu valori nu foarte depărtate de la o caracteristică la alta, având un maxim pentru 26 de caracteristici cu o valoare de 47.59% și o valoare minimă de 46.73% pentru 38 de caracteristici. *SVM* are cea mai bună acuratețe pentru fiecare număr de caracteristici, prezentând o pantă ascendentă, fiind și singurul care a reușit o rată de clasificare corectă de peste 50%. Maximul pentru *SVM* este de 50.08% pentru 38 de caracteristici și un minim de 49.18 pentru 26 de caracteristici.

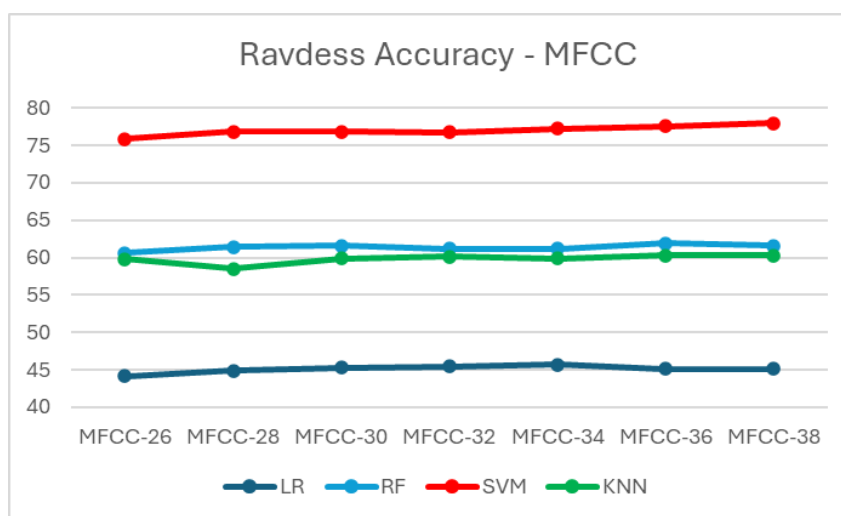


Figura 6.13 – Rata de clasificare corectă pentru baza de date Ravdess folosind extractorul de caracteristici MFCC.

Trecând la o altă bază de date, și anume Ravdess, vom vedea o schimbare destul de puternică în rata de clasificare corectă. După cum putem observa cel mai slab clasificator din punct de vedere al acurateții devine LR, care rămâne aproximativ constant în jurul valorii de 45%, cu o valoare minimă de 44.11% pentru un număr de 26 de caracteristici și o valoare maximă de 45,47% pentru un număr de 32 de caracteristici. KNN a reușit să depășească LR dar să se și apropie de RF, acesta având o valoare minimă de 58.53% pentru 28 de caracteristici, dar și o valoare maximă de 60.28% pentru 36 de caracteristici. Clasificatorul RF ocupă poziția a 2-a din punct de vedere a acurateții, rezultatele pastrându-se aproximativ egale pentru toate numerele de caracteristici. RF a avut o minimă de 60.6% pentru un număr de 26 de caracteristici, dar și un maxim de 61.91% pentru 36 de caracteristici. Cel mai bun clasificator din punct de vedere al ratei de clasificare corecte și singurul clasificator care a reușit să treacă peste o acuratețe de 70% este SVM. Rezultatele clasificatorului SVM descriu o pantă ascendentă pornind de la 26 de caracteristici cu o acuratețe de 75.9% și ajungând la 39 de caracteristici cu o acuratețe de 77.94%.

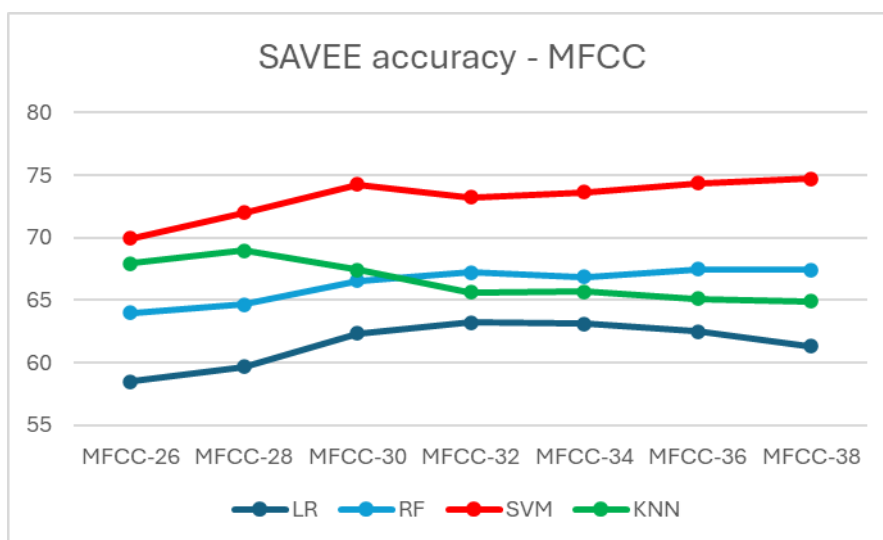


Figura 6.23 – Rata de clasificare corectă pentru baza de date SAVEE folosind extractorul de caracteristici MFCC.

În figura 6.23 este reprezentată acuratețea bazei de date SAVEE folosind extractorul de caracteristici MFCC pentru clasificatorii LR, RF, SVM și kNN. Pentru clasificatorul LR, se poate observa o îmbunătățire față de bazele de date Crema și Ravdess, unde a avut o rată de clasificare corectă sub 50%. LR are o valoare minimă a acurateții de 58.5% pentru 26 de caracteristici și o valoare maximă de 63.2% pentru un număr de 32 de caracteristici. RF descrie o pantă ascendentă, plecând de la o valoare minimă de 64% pentru 26 de caracteristici și ajungând la o valoare maximă de 67.48% pentru 36 de caracteristici. Pe de cealaltă parte se află clasificatorul kNN care descrie o pantă descendentă, Plecând de la o valoare maximă de 68.99% pentru 28 de caracteristici, coborând până la o valoare minimă de 64.89%. Cel mai bun clasificator din punct de vedere al acurateții rămâne SVM, având valorile maxime pentru fiecare număr de caracteristici. SVM are un o valoare minimă a ratei de clasificare corecte de 69.96% pentru un număr de 26 de caracteristici și o valoare maximă de 74.71% pentru un număr de 38 de caracteristici.

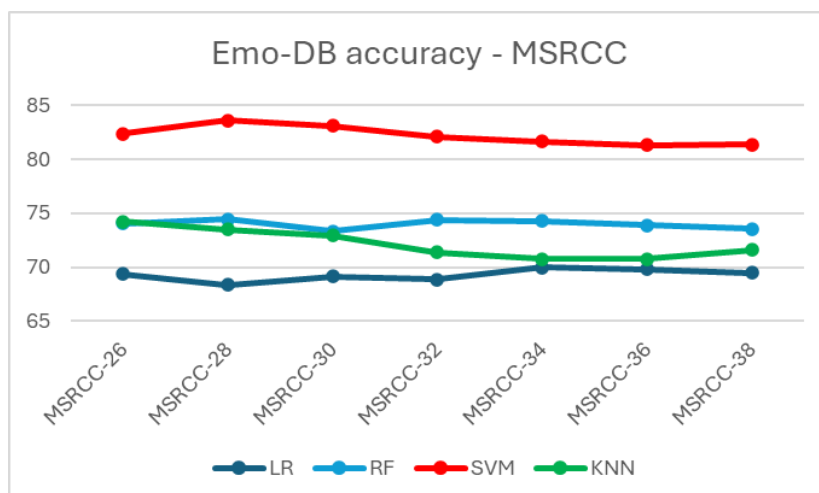


Figura 6.39 – Rata de clasificare corectă pentru baza de date Emo-DB folosind extractorul de caracteristici MSRCC.

În figura 6.39, ne este prezentată acuratețea pentru baza de date Emo-DB, folosind extractorul de caracteristici MSRCC și toți clasificatorii folosiți până în acest moment. Putem observa o creștere pentru clasificatorul kNN și o menținere pentru ceilalți trei clasificatori față de experimentele anterioare pentru aceeași bază de date. Clasificatorul cu cele mai slabe rezultate din punct de vedere al acurateții este LR, având o valoare minimă de 68.37% pentru 28 de caracteristici și o maximă de 69.96% pentru 34 de caracteristici. Următorul clasificator din punct de vedere al ratei de clasificare corecte este KNN, care deși a avut o creștere destul de mare față de rezultatele anterioare se clasează doar pe poziția a 3-a. Acest clasificator are o minimă a acurateții de 70.76% pentru 36 de caracteristici și o valoare maximă de 74.2% pentru un număr de 26 de caracteristici. Pe locul secund din punct de vedere al acurateții se află clasificatorul RF cu o valoare minimă de 73.34% pentru un număr de 30 de caracteristici și o valoare maximă de 74.42% pentru un număr de 32 de caracteristici. Cel mai bun clasificator din punct de vedere al acurateții este SVM, fiind singurul clasificator care are o rată de clasificare corectă de peste 80%. Cea mai mică valoare a acurateții pentru SVM este de 81.29% pentru 36 de caracteristici, însă cea mai mare valoare a acurateții este dată pentru 28 de caracteristici cu valoare de 83.6%.

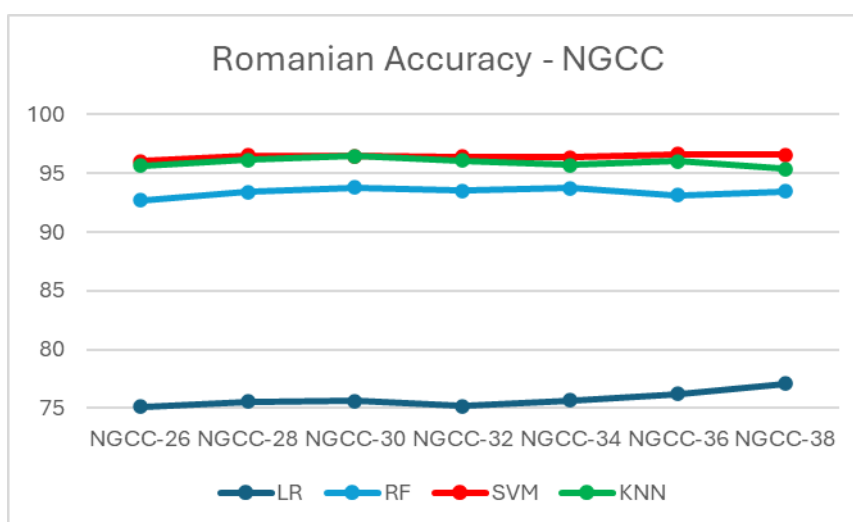


Figura 6.46 – Rata de clasificare corectă pentru baza de date în limba Română folosind extractorul de caracteristici NGCC.

În figura 6.46 ne este prezentată rata de clasificare corectă pentru baza de date în limba română, folosind extractorul de caracteristici NGCC pentru toți clasificatorii de până acum. Față de experimentul precedent, putem observa că toate ratele de clasificare corecte au fost crescute, ordinea clasificatorilor se schimbă doar între primii 2 clasificatori față de testul precedent. Cea mai slabă rată de clasificare corectă aparține clasificatorului LR pentru toate numerele de clasificatori. Se poate observa și că rezultatele pentru LR prezintă o pantă ascendentă. Valoarea minimă pentru rata de clasificare corectă a clasificatorului LR este de 75.13% pentru 26 de caracteristici, și valoarea maximă pentru LR este de 77.11% pentru un număr de 38 de caracteristici. Următorul clasificator din punct de vedere a acurateții este tot clasificatorul RF cu o rată minimă de clasificare de 92.7% pentru 26 de clasificatori și o valoare maximă de 93.77% pentru 30 de clasificatori. Următorul clasificator este k NN, care are o rată de clasificare minimă de 93.34% pentru 38 de clasificatori și o rată de clasificare maximă de 96.46%. Cel mai bun clasificator din punct de vedere al acurateții este SVM, cu o rată minimă de clasificare de 96.04% pentru 26 de caracteristici, și o valoare maximă de 96.63% pentru 36 de caracteristici.

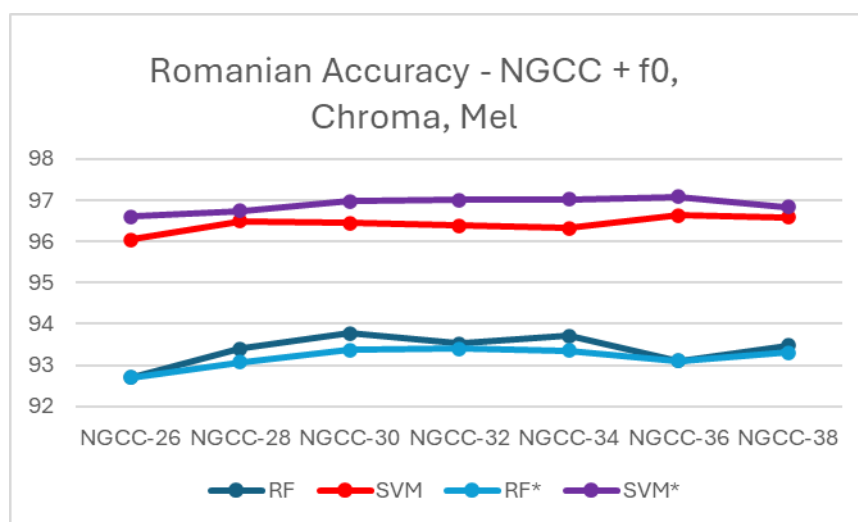


Figura 6.52 – Rata de clasificare corectă pentru baza de date în limba română folosind extractorul de caracteristici NGCC cât și caracteristicile F0, Chroma și Mel.

Figura 6.52 ne prezintă cazul în care schimbăm extractorul de caracteristici MFCC cu NGCC. O dată cu această schimbare putem observa atât o diminuare a diferenței dintre SVM* și SVM, cât și o schimbare în combinația RF și RF* unde RF are cea mai bună rată de clasificare corectă în 6 din 7 cazuri, în celălalt caz ambele au aceeași valoare a acurateții. Dacă analizăm cazul în care SVM* are valori mai mari decât SVM, putem observa că SVM* are o valoare minimă de 96.61% pentru 26 de caracteristici, dar și o valoare maximă care depășește pragul de 97% și ajunge la o valoare de 97.08% pentru 36 de caracteristici.

Pentru baza de date în limba Română, combinația cea mai optimă între extractor și clasificator din punct de vedere al ratei de clasificare corecte este între NGCC și SVM pentru 36 de caracteristici, dacă luăm în considerare și caracteristicile F0, Chroma și Mel. Dacă luăm în considerare și timpul, combinația cea mai optimă ar fi NGCC cu k NN pentru 30 de caracteristici.

Din toate rezultatele de până acum am decis să creem modele pentru extractorii de caracteristici NGCC cu 30 de caracteristici, NGCC cu 36 de caracteristici, MFCC cu 36 de caracteristici și clasificatorii k NN și SVM. În total voi fi create 6 modele. Pentru aceste modele s-au folosit 80% din date pentru antrenare și 20 % pentru testare. Pentru testare s-au ales din fiecare clasă sunetele 3 și 8, iar pentru antrenare restul.

```
For NGCC-30 + kNN
Train accuracy is: 82.440 %
Test accuracy is: 73.810 %

For NGCC-36 + kNN
Train accuracy is: 82.500 %
Test accuracy is: 74.762 %

For MFCC-36 + kNN
Train accuracy is: 91.190 %
Test accuracy is: 90.952 %

For NGCC-30 + SVM
Train accuracy is: 98.690 %
Test accuracy is: 91.667 %

For NGCC-36 + SVM
Train accuracy is: 98.929 %
Test accuracy is: 91.667 %

For MFCC-36 + SVM
Train accuracy is: 100.000 %
Test accuracy is: 95.000 %
```

Figura 6.53 – Rezultatele pentru fiecare model

Din Figura 6.53 reies ratele de clasificare corente pentru fiecare model. Putem observa că pentru antrenare, rata de clasificare corectă este sun 100% pentru toate cazurile, exceptând cazul în care folosim MFCC cu 36 de caracteristici și clasificatorul SVM. Cea mai mică rată de clasificare corectă pentru antrenare aparține modelului NGCC cu 30 de caracteristici și *k*NN cu o acuratețe de 82.44%. Pentru acest model este și cea mai mică rată de clasificare corectă pentru testare cu o acuratețe de 73.81%. Cea mai mare rată de clasificare corectă aparține modelului ce folosește ca și extractor de caracteristici MFCC cu 36 de caracteristici și SVM ca și clasificator.

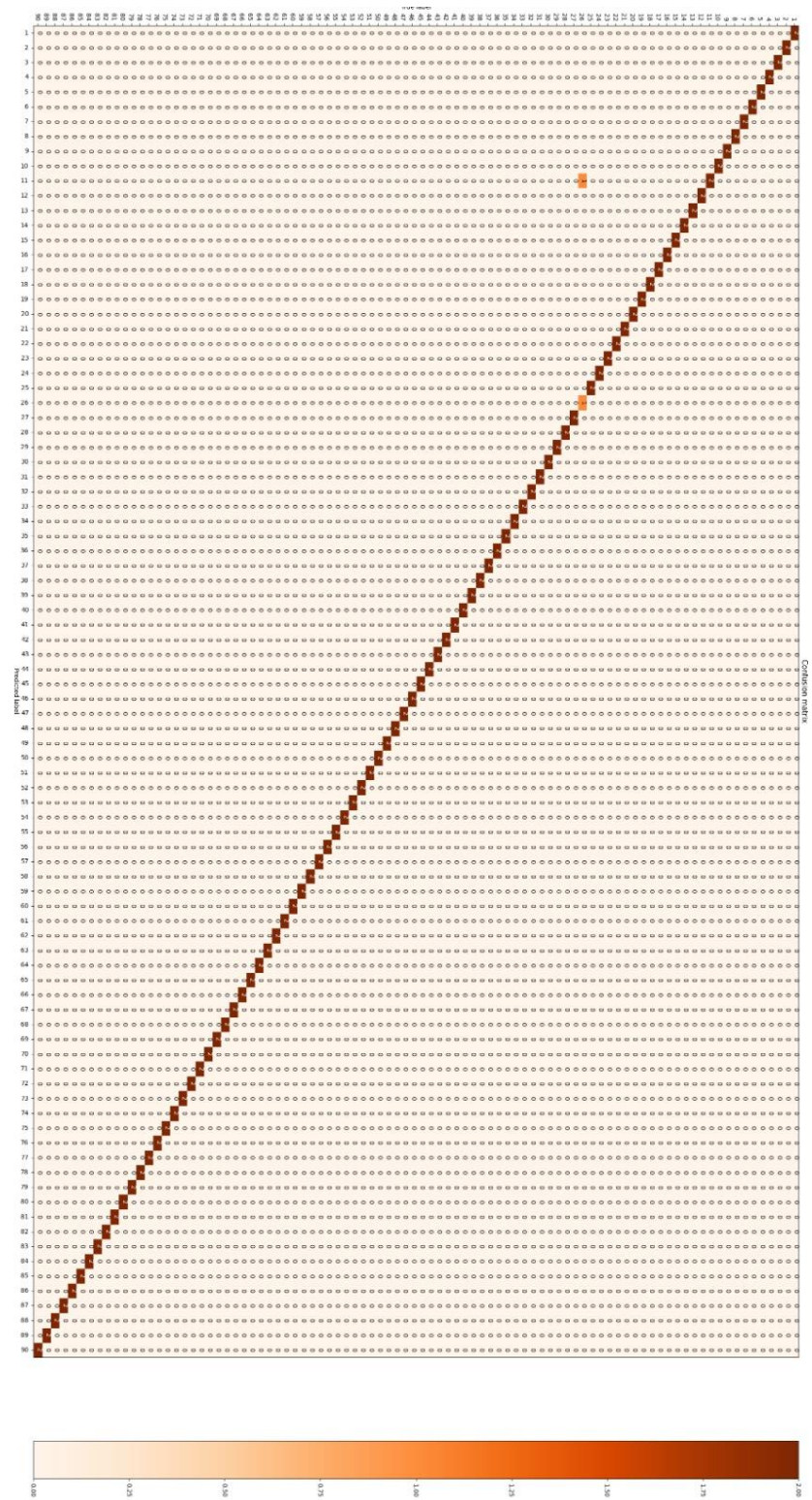


Figura 6.54 – Matricea de confuzie pentru MFCC-36 și SVM, pentru primii 3 vorbitori

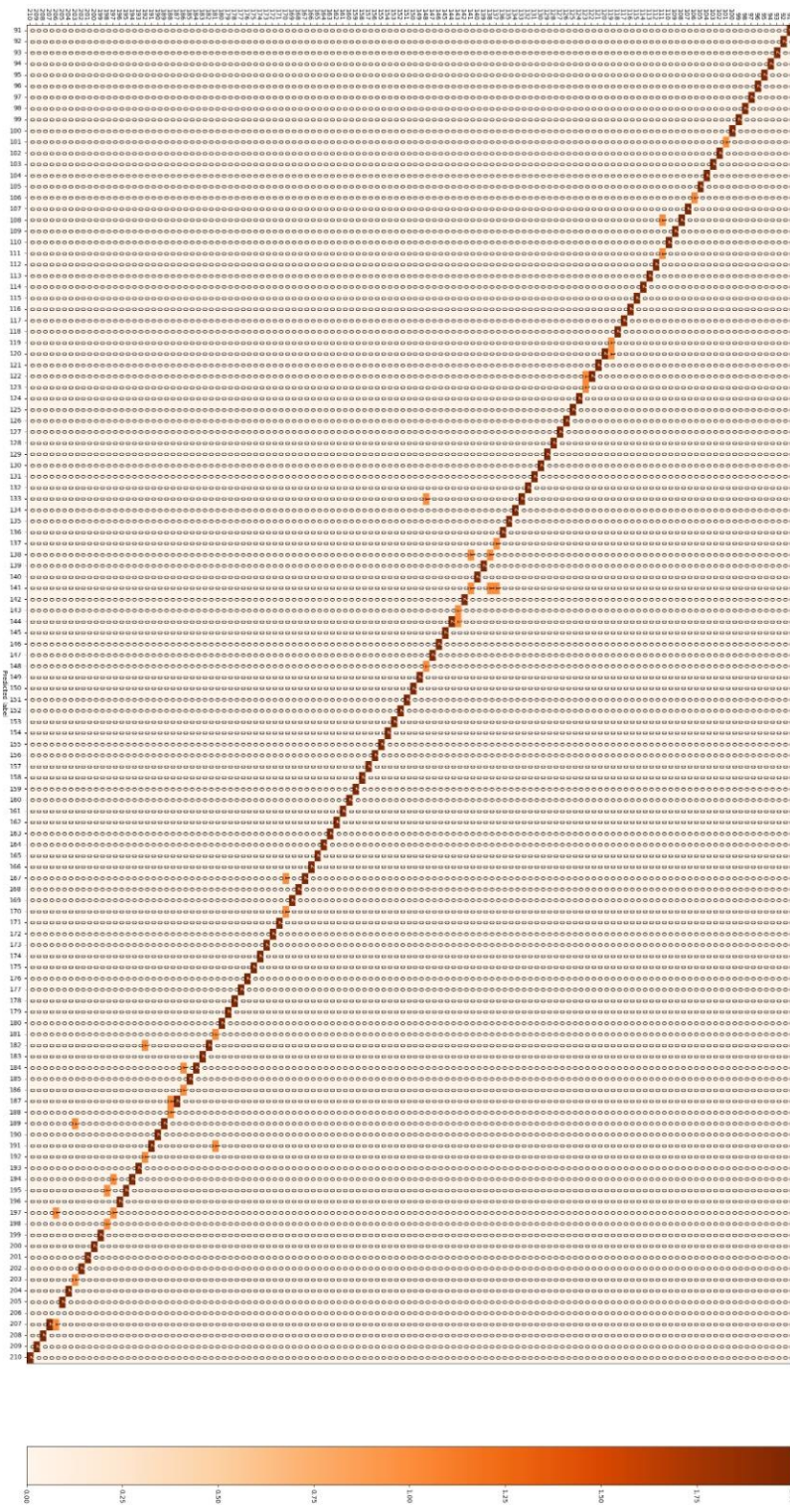


Figura 6.55 – Matricea de confuzie pentru MFCC-36 și SVM, pentru ultimii 4 vorbitori

Pentru că modelul MFCC cu 36 de caracteristici și SVM a avut cele mai bune rezultate, am creat matricea de confuzie pentru rezultatele din testare în figurile 6.54 și 6.55. Am ales să împărțim matricea de confuzie în 2 pentru a ajuta la lizibilitate. Se poate observa că doi vorbitori au avut 100% clasificare corectă, alți 2 vorbitori, unul de gen masculin și unul de gen feminin, a avut doar o propoziție identificată greșit. Din punct de vedere al clasificării, ultimii 2 vorbitori au avut 6 propoziții identificate greșit. Un vorbitor de gen masculin a fost clasificat greșit ca un vorbitor de gen feminin pentru 2 instanțe de propoziții. Din ce am observat, genul

unei persoane nu influențează rata de clasificare corectă. Vorbitorul de gen feminin a fost clasificat greșit pentru propoziția “Sunt foarte bine.”, spusă pe un ton vesel, care a fost confundată cu propoziția “Interpretarea a fost greșită.”, spusă pe un ton neutru. Vorbitorul de gen masculin care a avut doar o propoziție clasificată greșit a fost clasificat greșit pentru propoziția “E închisă ușa.” Rostită pe un ton neutru, care a fost clasificată ca și “A scăzut foarte mult”, rostită pe același ton. Cea mai des întâlnită clasificare greșită a emoției a fost de la neutru la trist. Cea mai puțin întâlnită clasificare greșită a emoției a fost de la fericit la trist.

Concluzii

Dezvoltarea de roboți este foarte solicitată din punct de vedere al comenzilor. Roboții și mașinile sunt integrate într-un număr mare de domenii și au nevoie de anumite competențe pentru a funcționa. Atunci când vorbim despre roboți care au o interacțiune cu oamenii, este de preferat ca aceștia să aibă caracteristici asemănătoare cu cele umane. Pentru a consolida interacțiunea om-robot ne-am gândit că o caracteristică pe care o au oamenii sunt emoțiile. De obicei, roboții sunt descriși ca fiind reci, deoarece nu au empatie sau sentimente. Până în prezent ne este imposibil să integrăm sentimente într-un robot, dar putem să îl facem să înțeleagă emoțiile. Pentru ca roboții să nu înțeleagă doar cuvintele pe care le rostim, folosind recunoașterea vorbirii, sau cine vorbește, folosind recunoașterea vorbitorului, încercăm să îi facem să vadă dincolo de primul strat al informațiilor transmise. Făcându-i pe roboți să înțeleagă emoțiile îi poate face să ia măsuri adecvate pentru a se simți mai emoționali sau empatici.

Atunci când vorbim despre emoții din punct de vedere audio, fiecare limbă are propriul set de intervale emoționale care pot avea sau nu similitudini cu alte limbi. Este foarte important să existe o bază de date emoțională în limba principală a unei țări, în cazul în care un robot este implementat în țara respectivă. Când vorbim despre emoțiile dintr-o bază de date, cu cât sunt mai multe emoții, cu atât mai bine. Pentru a simula multe emoții, este nevoie de ajutorul actorilor. Aceștia pot selecta fiecare emoție necesară din întregul spectru de emoții.

Ca rezultat al muncii noastre, am reușit să creăm o nouă bază de date care poate fi utilizată pentru cercetare în domeniul recunoașterii emoțiilor. Baza de date vine ca un ajutor, deoarece este prima de acest gen înregistrată în limba română. Baza de date emoționale are 3 emoții diferite: fericită, neutră și tristă. Există 10 propoziții care pot avea sensuri diferite în funcție de emoția cu care este folosită alături. Propozițiile sunt: “A venit rezultatul testului.”, “Nu mai am medicamente.”, “Testul este pozitiv.”, “Interpretarea a fost greșită.”, “A crescut foarte mult.”, “A scăzut foarte mult.”, “Îi închisă ușa.”, “Îi stins becul.”, “Sunt foarte bine.”, “Cheile sunt în ușa.”. Persoanele care au înregistrat propozițiile au fost 4 bărbați și 3 femei. Vârsta subiecților era cuprinsă între 24 și 28 de ani în momentul înregistrării.

Fiecare dintre propozițiile prezentate are un înțeles diferit pentru fiecare emoție care este încorporată cu fiecare propoziție. Cu acele propoziții înțelese de un robot, inclusiv toate semnificațiile lor, roboții pot veni ca un ajutor fizic și mental pentru oameni.

După ce am făcut această bază de date, o altă barieră între oameni și roboți este înlăturată, deoarece roboții pot începe acum să ne înțeleagă emoțiile și pot acționa în consecință. Fără trăsătura emoțională, roboții pot fi percepuți ca fiind reci și distanți. Uneori, roboții iau decizii greșite chiar dacă au înțeles propoziția transmisă. Acest lucru se întâmplă de obicei atunci când mesajul este transmis prin limbajul paraverbal care include emoții și limbajul corpului.

Aici vine o limitare a bazei de date. Oamenii erau foarte apropiați ca vârstă și pentru o bază de date mai fiabilă ar trebui să existe mai mulți oameni implicați care ar putea extinde

diferența de vârstă. În această bază de date ar trebui să fie înscrși și copii și cetățeni în vârstă. Întrucât România este o țară diversă din punct de vedere al diversității și etnografic, oameni din toată țara ar trebui incluși în baza de date pentru a varia accentele și dialectele.

O altă limitare a bazei de date vine în numărul de emoții înregistrate. Putem vedea că sunt folosite doar 3 emoții. Din studiile emoțiilor am văzut că există numeroase emoții pe care o ființă umană le poate experimenta. Având mai multe emoții ar ajuta roboții și mașinile să înțeleagă mai mult comportamentul nostru.

Această teză aduce contribuții în domeniul recunoașterii emoțiilor. În primul rând, baza de date nou creată în limba română deschide noi oportunități de cercetare. A avea o nouă bază de date într-o limbă care nu a fost inclusă în domeniul recunoașterii emoțiilor este o mare realizare. Se pot face studii pentru a vedea cât de asemănătoare poate fi o țară cu alta în ceea ce privește emoțiile. De asemenea, se poate studia ușurința tranziției de la o emoție la alta și pentru a vedea cum oamenii dintr-o anumită țară își pot schimba emoțiile.

În al doilea rând, o mare contribuție care vine cu această teză este studiul realizat în comparație între bazele de date emoționale în alte limbi și baza de date nou creată în limba română. Este posibil să observăm că pentru diferite limbi, diferite extractoare de caracteristici dau cele mai bune rezultate. Este interesant de văzut cât de mult fluctuează și atunci când emoțiile se schimbă.

În continuare, pe lângă analiza efectuată, am adăugat și caracteristicile pentru F_0 , Mel și Chroma pentru a vedea dacă vor îmbunătăți rata de clasificare corectă. Este fascinant de observat că în unele cazuri rata de clasificare corectă a scăzut după adăugarea caracteristicilor menționate în analiză, dar pentru noua noastră bază de date rata de clasificare corectă s-a îmbunătățit. De asemenea, a fost interesant de observat că folosind doar caracteristicile menționate anterior, pentru baza de date în limba română, rata de clasificare corectă a fost între 35% pentru Chroma și 50% pentru F_0 .

Dacă vorbim despre cea mai mare rată de clasificare corectă pentru baza de date în limba română, am obținut un maximum de 97,08% folosind NGCC ca și extractor de caracteristici și SVM ca și clasificator. Pentru a obține acest rezultat, am adăugat și următoarele caracteristici pentru a ajuta la clasificare: F_0 , Mel și Chroma.

Pe scurt, principalele contribuții ale acestei teze pot fi sintetizate sub următoarele puncte:

- S-au adunat voluntari și au reușit să-și schimbe emoțiile pentru a crea o bază de date emoțională.
- Crearea unei baze de date emoționale în limba română care poate fi folosită pentru cercetări în domeniul roboticii.
- Baza de date emoționale creată are 3 emoții: fericit, neutru și trist, care sunt cele mai comune emoții din viața noastră de zi cu zi.
- Am testat mai multe baze de date în diferite limbi pentru a descoperi că pentru diferite limbi, cel mai bun algoritm diferă.
- Algoritmii creați sunt compuși din diferiți extractori de caracteristici și clasificatori, care la rândul lor pot fi ajustați și cu ajutorul diferiților parametri specifici.
- Crearea unui algoritm optimizat, pe baza testelor efectuate pe bazele de date preexistente, pentru baza de date în limba română care are o acuratețe comparabilă cu acuratețea pentru recunoașterea vorbirii.
- S-a creat o legătură mai puternică între oameni și roboți, deoarece roboții pot depăși acum datele aleatorii atunci când mesajul audio transmis este făcut prin emoții.
- Roboții pot veni acum ca un ajutor atunci când se vorbește despre monitorizarea sănătății mintale și interacțiunile sociale.

Referințe

- [1] "Python programming language" En.wikipedia.org, 2023, [Online], Available: [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)) [Accessed: 05- Nov- 2023].
- [2] F. Gao, C. -H. Chen, J. -G. Hsieh and J. -H. Jeng, "Support Vector Classifier Trained by Gradient Descent," 2021 International Conference on Sensing, Measurement & Data Analytics in the era of Artificial Intelligence (ICSMD), Nanjing, China, 2021, pp. 1-5, doi: 10.1109/ICSMD53520.2021.9670839.
- [3] Defazio, Aaron, Francis Bach, and Simon Lacoste-Julien. "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives." Advances in neural information processing systems 27 (2014).