UNIVERZITET U NIŠU  
ELEKTRONSKI FAKULTET

Katedra za računarstvo

Uticaj primene metoda augmentacije teksualnih podataka na detekciju govora mržnje

Seminarski rad

Predmet: Prikupljanje i predobrada podataka za mašinsko učenje

Student:

Katarina Stanojković, br. ind. 1773

Mentor:

Prof. dr Aleksandar Stanimirović

Niš, 2024. godina

Sadržaj

[1. Uvod 3](#_Toc184225408)

[2. Predobrada tekstualnih podataka 4](#_Toc184225409)

[2.1. Čišćenje teksta 4](#_Toc184225410)

[2.2. Normalizacija 4](#_Toc184225411)

[2.3. Tokenizacija 5](#_Toc184225412)

[2.4. Lematizacija i stemovanje 5](#_Toc184225413)

[2.5. Uklanjanje stop-reči 6](#_Toc184225414)

[2.6. Spajanje kontrakcija 6](#_Toc184225415)

[2.7. Uklanjanje duplikata 6](#_Toc184225416)

[3. Pregled tehnika augmentacije tekstualnih podataka 7](#_Toc184225417)

[3.1. Data Space 8](#_Toc184225418)

[3.1.1. Na nivou karaktera 8](#_Toc184225419)

[3.1.2. Na nivou reči 9](#_Toc184225420)

[3.1.3. Na nivou fraza i rečenica 10](#_Toc184225421)

[3.1.4. Na nivou dokumenta 12](#_Toc184225422)

[3.2. Feature Space 15](#_Toc184225423)

[3.2.1. Indukcija šuma 15](#_Toc184225424)

[3.2.2. Interpolacione metode 15](#_Toc184225425)

[4. Praktični deo rada 16](#_Toc184225426)

[4.1. Opis dataset-a 16](#_Toc184225427)

[4.2. Preprocesiranje podataka 17](#_Toc184225428)

[4.3. Primena modela nad osnovnim dataset-om 18](#_Toc184225429)

[4.4. Metode za augmentaciju podataka 20](#_Toc184225430)

[4.5. Primena modela nad augmentiranim dataset-ovima 21](#_Toc184225431)

[4.6. Tabelarni prikaz i analiza rezultata 24](#_Toc184225432)

[5. Zaključak 26](#_Toc184225433)

[6. Reference 27](#_Toc184225434)

# Uvod

U savremenom razvoju sistema mašinskog učenja, podaci predstavljaju osnovu za izgradnju modela koji mogu pouzdano obraditi i analizirati kompleksne zadatke. Kvalitet i kvantitet tih podataka direktno utiču na performanse modela, ali prikupljanje velike količine relevantnih podataka često predstavlja izazov, posebno kada je reč o tekstualnim podacima. Ovi podaci mogu biti ograničeni u obimu, raznovrsnosti i kvalitetu, što može dovesti do lošije generalizacije modela i problema sa overfitting-om.

Kako bi se prevazišli ovi izazovi, metode augmentacije tekstualnih podataka su se pokazale kao ključne tehnike za unapređenje raznovrsnosti i kvaliteta skupa podataka. Augmentacija tekstualnih podataka obuhvata generisanje novih primera na osnovu postojećih, pri čemu se čuvaju ključne semantičke informacije. Ove metode omogućavaju poboljšanje performansi modela i njihovu otpornost na varijacije u realnim podacima, dok istovremeno smanjuju rizik od prekomernog prilagođavanja (overfitting).

Cilj ovog rada je pružiti pregled različitih tehnika augmentacije tekstualnih podataka i analizirati njihov uticaj na rešavanje praktičnih problema u obradi prirodnog jezika (NLP). Poseban akcenat stavljen je na primenu ovih metoda u kontekstu detekcije govora mržnje, jedne od ključnih oblasti primene u savremenom NLP-u.

Rad uključuje i praktičan deo koji demonstrira implementaciju metoda augmentacije nad specifičnim skupom podataka, analizu njihovog uticaja na performanse različitih modela mašinskog učenja i uporednu evaluaciju rezultata. Ovaj pristup omogućava sticanje uvida u efikasnost različitih tehnika i pruža preporuke za njihovu primenu u stvarnim NLP zadacima.

# Predobrada tekstualnih podataka

## 2.1. Čišćenje teksta

Čišćenje teksta je prvi korak u pripremi podataka za obradu, jer pomaže u uklanjanju nepotrebnih elemenata koji mogu ometati modele u tačnom prepoznavanju obrazaca u tekstu. Proces obično uključuje sledeće:

* **Uklanjanje znakova interpunkcije**: Interpunkcija poput tačaka, zareza, upitnika i uzvičnika često se uklanja jer obično ne doprinosi semantičkom značenju teksta u NLP zadacima.
* **Uklanjanje brojeva**: Brojevi se obično uklanjaju, osim u slučajevima kada su važni za analizu, poput analiza koje se bave numeričkim podacima.
* **Uklanjanje emotikona i specijalnih karaktera**: Emotikoni i drugi simboli kao što su "$", "&" ili "@", koji nisu deo semantičkog značenja, takođe se uklanjaju.
* **Uklanjanje URL-ova i email adresa**: Linkovi i email adrese često nisu korisni za analizu i mogu ometati tok obrade.
* **Uklanjanje HTML tagova**: U tekstovima preuzetim sa interneta ili iz HTML izvora, često se uklanjaju HTML tagovi, koji mogu dodavati šum.

Ove korake je potrebno prilagoditi zavisno od zadatka, jer se ponekad neki elementi mogu zadržati ako su relevantni za specifičnu analizu.

## 2.2. Normalizacija

Normalizacija teksta osigurava doslednost tako što transformiše tekst u standardizovani oblik. Glavni ciljevi normalizacije uključuju:

* **Pretvaranje velikih slova u mala slova**: Ovaj proces pomaže da se izbegne razlikovanje između reči poput "Pas" i "pas", čime se smanjuje broj varijacija koje model mora da nauči.
* **Uklanjanje dijakritika**: U jezicima poput srpskog, francuskog ili španskog, dijakritički znakovi (npr. č, ć, š, é) mogu biti uklonjeni kako bi se smanjile varijacije u rečima ("čovek" -> "covek").
* **Standardizacija teksta**: Uključuje zamenu različitih oblika reči njihovim doslednim varijantama (npr. "color" i "colour" postaju "color").

Normalizacija je važna za stvaranje doslednog skupa podataka koji će omogućiti bolju generalizaciju modela.

## 2.3. Tokenizacija

Tokenizacija je ključan proces u obradi prirodnog jezika, jer razbija tekst na manje jedinice (tokene) koje model može lakše obrađivati. Postoji nekoliko vrsta tokenizacije:

* **Tokenizacija na nivou reči**: Tekst se deli na pojedinačne reči. Ova vrsta tokenizacije se često koristi u zadacima poput klasifikacije teksta, analize sentimenta i pretraživanja informacija, gde je razumevanje pojedinačnih reči ključno. Na primer, za rečenicu „NLP is fun!“, tokenizacija na nivou reči bi dala tokene [„NLP“, „is“, „fun“, „!“].
* **Tokenizacija na nivou rečenica**: Tekst se deli na rečenice. Ova vrsta tokenizacije je korisna u zadacima poput sažimanja teksta i mašinskog prevođenja, gde je fokus na razumevanju i obradi celih rečenica. Na primer, za tekst „ NLP is fun. Let’s learning it together.“, tokenizacija na nivou rečenica bi dala tokene [„NLP is fun.“, „Let’s learning it together.“].
* **Tokenizacija na nivou karaktera**: Tekst se deli na pojedinačne karaktere. Tokenizacija na nivou karaktera se koristi u zadacima poput modelovanja jezika za jezike sa složenim pismima, generisanja teksta na nivou karaktera i rukovanja rečima koje nisu u vokabularu modela. Na primer, za reč „NLP“, tokenizacija na nivou karaktera bi dala tokene [„N“, „L“, „P“].
* **Tokenizacija na nivou subreči**: Tekst se deli na jedinice subreči, često korišćene u jezicima sa bogatom morfologijom. Ova vrsta tokenizacije je efikasna u neuronskom mašinskom prevođenju i rukovanju jezicima sa složenim oblicima reči, gde reči mogu imati više prefiksa i sufiksa. Na primer, za reč „unhapiness“, tokenizacija na nivou subreči bi dala tokene [„un“, „hapiness“].

Tokenizacija je ključna jer omogućava modelima da obrađuju tekst u delovima.

## 2.4. Lematizacija i stemovanje

Lematizacija i stemovanje su tehnike za redukciju reči na njihov osnovni oblik. Ove tehnike se koriste kako bi se smanjila varijacija u rečima koje predstavljaju iste koncepte.

* **Lematizacija**: Pretvara reč u njen osnovni oblik na temelju njenog značenja i konteksta. Na primer, "running", "ran" i "runs" bi se sve vratile na osnovni oblik "run". Lematizacija zahteva lingvističke informacije o reči, što je čini preciznijom od stemovanja.
* **Stemovanje**: Skraćuje reči uklanjanjem završetaka, ali bez uzimanja u obzir značenja reči. Na primer, "running" postaje "run", ali isto tako i reč “runner” postaje “run”. Iako je brže i jednostavnije, stemovanje je manje precizno jer ne uzima u obzir gramatiku i kontekst.

Obe tehnike su korisne za smanjenje redundantnih oblika reči u tekstu, što olakšava modelima da uče efikasnije.

## 2.5. Uklanjanje stop-reči

Stop-reči su uobičajene reči kao što su "the", "is", "in", "at" koje se često pojavljuju u tekstu, ali ne nose mnogo informacija o značenju. Uklanjanje stop-reči omogućava modelima da se fokusiraju na reči koje nose veću informativnu vrednost.

* **Unapred definisane liste stop-reči**: Većina NLP biblioteka dolazi sa unapred definisanim listama stop-reči, ali se ove liste mogu prilagoditi potrebama zadatka.
* **Prilagođavanje stop-reči**: Ponekad je potrebno kreirati prilagođene liste stop-reči za specifične domene. Na primer, u medicinskim tekstovima reč "patient" može biti relevantna i ne bi trebala biti uklonjena.

Ovaj korak smanjuje šum u podacima i poboljšava performanse modela.

## 2.6. Spajanje kontrakcija

Spajanje kontrakcija podrazumeva proširivanje skraćenih oblika reči u njihov pun oblik. Kontrakcije poput "can't", "won't" ili "I'm" često zbunjuju modele za obradu teksta jer predstavljaju više reči spojenih u jednu.

* **Proširenje kontrakcija**: "can't" postaje "cannot", "I'm" postaje "I am", itd. Ovo olakšava precizniju tokenizaciju i bolji rad modela.
* **Poboljšana analiza teksta**: Kada se kontrakcije prošire, modeli za obradu teksta imaju tačniji prikaz onoga što tekst znači, što poboljšava sve daljnje korake u obradi.

Ovo je posebno važno u tekstovima koji sadrže puno kolokvijalnog jezika ili neformalnih izraza.

## 2.7. Uklanjanje duplikata

Uklanjanje duplikata je proces kojim se iz skupa podataka uklanjaju tekstovi koji se pojavljuju više puta. Ovaj korak je važan kako bi se izbegla redundantnost koja može dovesti do prekomernog prilagođavanja modela (overfitting).

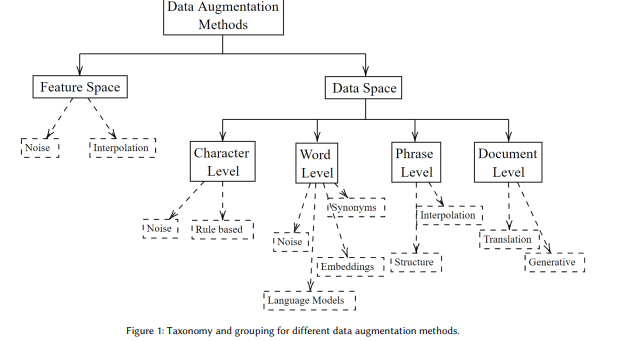
* **Smanjenje redundancije**: Duplikati često stvaraju problem kada model uči iz skupa podataka, jer isti podaci više puta utiču na rezultate.
* **Povećana efikasnost**: Kada se duplikati uklone, dataset postaje manji i brži za obradu, što poboljšava ukupne performanse modela.

# Pregled tehnika augmentacije tekstualnih podataka

Augmentacija teksta obuhvata širok spektar metoda, koje se mogu primeniti na različitim nivoima tekstualnih podataka, od reči i fraza, do celih dokumenata. Kroz ovaj rad, fokusiraćemo se na pregled različitih tehnika augmentacije tekstualnih podataka, sa naglaskom na njihov uticaj na detekciju govora mržnje.

U ovom kontekstu, tehnike augmentacije se mogu podeliti u dve glavne kategorije: one koje se primenjuju na nivou karakteristika (Feature Space) i one koje direktno manipulišu tekstualnim instancama (Data Space). Pregled ovih metoda obuhvata jednostavne pristupe, kao što su sinonimna zamena i back-translation, kao i napredne metode koje koriste jezičke modele i generativne algoritme. Ovaj pregled pruža uvid u ključne prednosti i ograničenja svake od ovih tehnika, istovremeno naglašavajući specifičnosti njihove primene u realnim zadacima obrade teksta.

Kroz poređenje ovih tehnika, rad nudi smernice o tome koja metoda augmentacije je najpogodnija za određene situacije, osvetljavajući ključne aspekte koji utiču na performanse modela u različitim kontekstima.



## Data Space

### Na nivou karaktera

Augmentacija tekstualnih podataka na nivou karaktera podrazumeva različite transformacije koje se primenjuju na najmanjoj jedinici teksta – karakterima. Metode augmentacije na nivou karaktera često koriste pravila ili uvode šum kako bi simulirale greške koje se javljaju prilikom kucanja, skeniranja ili prepoznavanja teksta.

#### ****Simulacija pravopisnih grešaka****

Ova tehnika koristi unapred definisana pravila kako bi stvorila greške koje podsećaju na uobičajene pravopisne pogreške. Ova metoda omogućava modelu da nauči da prepoznaje varijante reči sa pravopisnim greškama, čime se povećava njegova sposobnost da pravilno interpretira tekst sa greškama.

**"The fox jumps over the lazy dog" → "The fox jumps over the lazi dog"**

#### ****Simulacija grešaka u kucanju (Keyboard Augmenter)****

Ova metoda simulira greške u kucanju na osnovu blizine tastera na tastaturi, što imitira uobičajene tipografske greške koje se dešavaju pri brzom kucanju. Ova augmentacija pomaže modelu da postane otporniji na greške koje korisnici prave prilikom unosa teksta.

**"The fox jumps over the lazy dog" → "The fox jumps over the lazu dog"**

#### ****Optičko prepoznavanje karaktera (OCR simulacija)****

Ova tehnika imitira greške koje nastaju prilikom korišćenja sistema za optičko prepoznavanje karaktera (OCR), gde slični karakteri često bivaju pogrešno interpretirani. Ova metoda je korisna za treniranje modela da prepoznaje tekst iz različitih skeniranih dokumenata sa greškama u prepoznavanju.

**"The fox jumps over the lazy dog" → "The fox jumps 0ver the 1azy dog"**

#### ****Random augmentacija karaktera****

Ova tehnika uvodi nasumične varijacije u tekstualne podatke. Postoje 4 osnovne operacije:

* Insert – Ova operacija dodaje karaktere na nasumičnim mestima u tekstu

**"Boat" → "Boaat"**

* Substitute – Ova operacija vrši slučajnu zamenu postojećeg karaktera drugim

**"Boat" → "Beat"**

* Swap – Ova operacija slučajno menja mesta dva karaktera

**"Boat" → "Baot"**

* Delete – Ova operacija slučajno briše postojeći karakter

**"Boat" → "Bat"**

### Na nivou reči

Augmentacija na nivou reči obuhvata tehnike koje modifikuju pojedinačne reči u rečenici kako bi generisale varijacije teksta, a istovremeno očuvale osnovni smisao. Ove metode često koriste zamene sinonima, indukciju šuma ili naprednije modele kao što su ugneždena predstavljanja reči (embedding). Cilj je povećanje raznovrsnosti podataka za treniranje modela, što doprinosi boljoj generalizaciji i robusnosti modela u realnim uslovima.

#### Zamena sinonima

Sinonimna zamena koristi tezaurus kao što je WordNet za pronalaženje semantički sličnih reči. Ova metoda koristi verovatnoću zamene na osnovu distribucije, gde se retki sinonimi preferiraju, što doprinosi boljem učenju, posebno kod zadataka s malim resursima.

**"Happy" → "Joyful"**

#### Zamena antonima:

Zamena antonima menja reči sa njihovim suprotnim značenjima, kao što su "happy" u "sad". Ova metoda uvodi kontrast u tekst i može biti korisna za generisanje raznovrsnijih konteksta ili za testiranje sposobnosti modela da prepozna promene u sentimentu.

**"Happy" → "Sad"**

#### Umetanje reči (Random Insertion):

Nasumično umetanje reči dodaje dodatne ili semantički slične reči u rečenicu, bez promene njenog osnovnog značenja. Ova tehnika dodaje detalje tekstu, povećavajući njegovu bogatost i raznovrsnost.

**"He enjoys reading books" → "He really enjoys reading interesting books"**

#### Nasumična zamena (Random Swap):

Nasumična zamena menja pozicije reči unutar rečenice, zadržavajući njihov semantički smisao. Ova metoda unosi varijacije u strukturu rečenice, što pomaže modelu da postane fleksibilniji u prepoznavanju različitih redosleda reči.

**"The quick brown fox jumps over the lazy dog"  
→ "The brown quick fox jumps over the lazy dog"**

#### Nasumično brisanje (Random Deletion):

Nasumično brisanje uklanja pojedine reči iz rečenice, ali se zadržava njen osnovni smisao. Ova tehnika može skratiti rečenice i pomoći modelu da nauči da radi sa nepotpunim podacima.

**"Artificial intelligence is transforming the world"   
→ "Artificial intelligence is transforming world"**

#### Podela reči (Split Augmentation):

Podela reči nasumično deli reči na manje delove, menjajući strukturu reči i stvarajući nove tekstualne varijacije. Ova metoda pomaže u povećanju varijabilnosti u trening podacima.

**"Darkness" → "Dark ness"**

#### Zamena reči putem ugneždenih predstavljanja (Embedding):

Umesto zamene reči sinonimima, ova metoda koristi rečničke vektore (Word2Vec, GloVe, FastText) kako bi zamenila reči onima koje su najbliže u vektorskom prostoru.

**"Happy" → "Joyful"**

#### Contextual Embeddings:

Contextual Embeddings koriste modele kao što su BERT ili RoBERTa kako bi zamenila reči sličnim rečima u kontekstu rečenice. Ova tehnika uzima u obzir širi kontekst rečenice, omogućavajući preciznije zamene.

**"The storm was strong" → "The storm was intense"**

#### Rezervisane reči (Reserved Word Augmentation):

Ova tehnika omogućava definisanje reči koje se ne smeju menjati tokom augmentacije. Na primer, ukoliko je važno da reč "lion" ostane nepromenjena, može se koristiti lista rezervisanih reči kako bi se izbegla njena zamena tokom procesa augmentacije.

**"The lion roared loudly" → "The lion roared extremely loudly"**

(ako je "lion" rezervisana reč, ostaje nepromenjena).

#### TF-IDF Augmentacija:

Ova metoda koristi termine sa niskim TF-IDF skorom, koje se mogu zameniti drugim rečima sa sličnim skorom. TF-IDF (Term Frequency-Inverse Document Frequency) izračunava važnost reči u tekstu, omogućavajući zamenu manje značajnih reči bez promene osnovnog značenja.

**"The cat sat on the mat" → "The cat sat on the rug"**

### Na nivou fraza i rečenica

Augmentacija na nivou fraza i rečenica fokusira se na modifikaciju celokupnih sintaktičkih jedinica kako bi se stvorile nove varijante teksta uz očuvanje semantičkog značenja. Ove tehnike omogućavaju generisanje varijacija u tekstu koje pomažu u diversifikaciji skupa podataka i unapređuju sposobnost modela da generalizuje na realnim primerima.

#### Kropljenje i rotacija (Cropping and Rotation)

Kropljenje i rotacija su tehnike inspirisane obradom slike, primenjene na strukturu rečenice. Kropljenje podrazumeva skraćivanje rečenice fokusiranjem na ključne komponente, kao što su subjekti ili objekti.

**"The cat sat on the mat and looked at the dog" → "The cat sat on the mat"**

Rotacija podrazumeva premeštanje delova rečenice oko ključnog korena, stvarajući različite varijante.

**"The cat sat on the mat" → "On the mat sat the cat"**

Ove tehnike su korisne za zadatke kao što je označavanje delova govora, ali u nekim scenarijima donose minimalne dobitke u performansama.

#### Parafraziranje

Parafraziranje je tehnika preformulisanja teksta dok se zadržava originalno značenje. Korišćenjem parafrazera, može se stvoriti sintetički tekst koji uvodi leksičku raznolikost. Ključan resurs za parafraziranje je PPDB (Paraphrase Database), koji sadrži milione leksičkih, frazalnih i sintaktičkih parafraza generisanih metodom pivotiranja između više jezika. Parafraziranje omogućava generisanje različitih verzija rečenica, što je korisno za zadatke kao što su sažimanje teksta i odgovaranje na pitanja.

**"The cat sat on the mat" → "The feline rested on the rug"**

#### Zamena fraza pomoću zavisnih stabala

Ova metoda koristi zavisna stabla kako bi se identifikovale fraze koje mogu biti zamenjene sličnim frazama iz drugih rečenica. Zamena fraza omogućava generisanje novih sintetičkih tekstova koji zadržavaju gramatiku i strukturu originalne rečenice, ali uvode semantičku raznolikost.

**"The quick brown fox jumps over the lazy dog"   
→ "The quick brown fox jumps over the sleepy cat"**

#### Apstraktno sažimanje

Apstraktno sažimanje generiše kratke sažetke dugih tekstova, kreirajući nove rečenice koje možda ne postoje u izvornom tekstu. Ova tehnika unosi raznovrsnost u strukturu fraza i pomaže modelima da obrade kraće verzije teksta zadržavajući ključne informacije, čime se poboljšava efikasnost i tačnost modela.

**"It was a dark and stormy night. I was alone at home when I saw a lion's face followed by a scary thunderous roar at the windows"   
→ "A lion appeared during a stormy night"**

#### Contextual Word Embeddings

Contextual Word Embeddings koriste pretrenirane modele kao što su GPT-2, XLNet ili DistilGPT-2 kako bi generisali nove rečenice zasnovane na semantičkom kontekstu. Ova metoda može dodati nove rečenice koje proširuju postojeći tekst ili zameniti postojeće fraze novim, zadržavajući kontekst i smisao rečenice.

**"The storm was strong" → "The storm was intense"**

#### LAMBADA augmentacija

LAMBADA (Language Model Based Data Augmentation) tehnika koristi unapred trenirane modele jezika, poput GPT ili BERT, kako bi generisala alternativne verzije rečenica predviđanjem nedostajućih reči na osnovu konteksta. Ova metoda doprinosi raznolikosti struktura rečenica u skupu podataka, povećavajući bogatstvo i varijabilnost trening primera.

**"The cat was chasing the \_\_\_" →"The cat was chasing the mouse"**

#### Nasumična augmentacija rečenica (Random Sentence Augmentation)

Nasumična augmentacija na nivou rečenica uključuje različite nasumične operacije, kao što su brisanje, premeštanje ili zamena rečenica unutar većeg teksta. Ove varijacije stvaraju nove verzije teksta koje zadržavaju osnovni smisao, ali menjaju redosled ili strukturu rečenica, što doprinosi diversifikaciji trening skupa podataka.

**"It was a dark and stormy night. I was alone at home when I saw a lion's face followed by a scary thunderous roar at the windows"   
→ " I was alone at home when I saw a lion's face followed by a scary thunderous roar at the windows. It was a dark and stormy night"**

### Na nivou dokumenta

Augmentacija na nivou dokumenta obuhvata tehnike koje se primenjuju na celu jedinicu teksta, odnosno na kompletne dokumente. Ove metode omogućavaju generisanje novih dokumenata na osnovu originalnih, čime se povećava raznovrsnost i količina podataka za treniranje modela. Tehnike na nivou dokumenta često uključuju prevođenje, korišćenje generativnih modela i druge napredne metode koje stvaraju sintetičke dokumente sa očuvanim semantičkim značenjem.

#### ****Back-Translation (BT)****

Back-Translation je metoda koja koristi prevod sa jednog jezika na drugi, a zatim vraćanje teksta na originalni jezik kako bi se dobila nova sintetička verzija. Ova tehnika generiše parafrazirani tekst koji zadržava osnovno značenje originala, ali uvodi leksičke i sintaktičke varijacije.

**"The cat sat on the mat"  
→ "Le chat s'est assis sur le tapis"  
→ "The cat rested on the rug"**

Back-Translation je naročito korisna u zadacima mašinskog prevođenja, ali je pokazala i uspeh u drugim NLP zadacima poput analize sentimenta i odgovaranja na pitanja, gde doprinosi povećanju količine i raznolikosti trening podataka.

#### ****Iterative Back-Translation (IterativeBT)****:

Iterative Back-Translation predstavlja unapređenu varijantu osnovne Back-Translation metode, gde se proces prevođenja i vraćanja teksta ponavlja više puta. Svaka iteracija koristi unapređene prevode generisane prethodnim modelom, čime se model kontinuirano poboljšava u generisanju sintetičkog teksta.

**"The cat sat on the mat"  
→ "El gato se sentó en la alfombra"  
→ "The cat sat on the doormat"  
  
→ "Die Katze saß auf der Matte"  
→ "The cat was sitting on the mat"**

Ovaj pristup je posebno koristan u okruženjima sa malim resursima, jer omogućava generisanje većih količina podataka za treniranje modela bez potrebe za dodatnim ručnim unosom.

#### ****Noised Back-Translation (NoisedBT)****:

Noised Back-Translation kombinuje osnovnu Back-Translation tehniku sa dodavanjem šuma u tekst tokom procesa prevođenja. Šum može uključivati nasumično brisanje, zamenu ili premeštanje reči, čime se dodatno povećava raznovrsnost sintetičkog teksta.

**"The cat sat on the mat"  
→ "The cat on sat the mat"  
→ "Il gatto era seduto sul tappetino"  
  
→ "The cat was sitting on the carpet"**

Dodavanje šuma čini model otpornijim na prirodne varijacije u tekstu, omogućavajući mu da bolje prepoznaje i interpretira različite strukture rečenica.

#### ****Generativne metode (GPT-based models)****:

Generativni modeli, poput LAMBADA algoritma zasnovanog na GPT-2, koriste unapred trenirane neuronske mreže za generisanje novih sintetičkih podataka. U ovoj metodi, model jezika (GPT-2) se prilagođava na osnovu originalnog skupa podataka i koristi se za generisanje novih fraza.

Na primer, model može generisati nove rečenice koje su semantički slične originalu, a zatim se koristi klasifikator za filtriranje kvalitetnih sintetičkih podataka. Ova metoda omogućava kreiranje bogatog sintetičkog skupa podataka koji pomaže u smanjenju prekomernog prilagođavanja modela (overfitting).

**"Artificial intelligence is transforming the world"  
→ "AI is revolutionizing global industries"**

#### ****Tagged Back-Translation (TaggedBT)****:

Tagged Back-Translation je varijanta Back-Translation tehnike koja koristi specijalne tagove za označavanje sintetički generisanog teksta. Umesto dodavanja šuma, ovaj pristup označava generisane podatke kako bi model naučio da razlikuje originalni i sintetički tekst.

**"The cat sat on the mat"→ "[SYN] The cat rested on the rug"**

Ova tehnika omogućava modelu da ispravno koristi sintetičke podatke, čime se poboljšava tačnost i robusnost modela bez narušavanja semantičkog značenja originalnog teksta.

#### Back Transliteration

**Back Transliteration** je metoda augmentacije podataka koja se koristi za generisanje rečenica ili fraza koje zvuče fonetski slično izvornom jeziku, ali su napisane u drugom pismu. Ova tehnika je posebno korisna za generisanje trening podataka za klasifikacione zadatke koji uključuju lokalizovane ili bi-jezične fraze, gde je ciljni jezik jezik sa malim resursima, odnosno ima manje dostupnih izvora podataka.

**"Machine learning is a subset of AI"→ "Машине лернинг ис а сабсет оф АИ"**

## Feature Space

Augmentacija podataka u **feature space** (prostoru karakteristika) znači da se ne rade promene direktno na tekstu, već na njegovim numeričkim reprezentacijama, tj. na vektorskim prikazima rečenica ili reči (tzv. embeddings). Ove metode omogućavaju da se kreiraju nove varijacije podataka bez promene originalnog teksta, čime se model trenira da bude robusniji i otporniji na različite varijacije.

### Indukcija šuma

Indukcija šuma u prostoru karakteristika podrazumeva dodavanje malih, nasumičnih promena na vektorskim prikazima teksta. Ove promene pomažu modelu da uči iz raznih varijacija i da postane bolji u prepoznavanju sličnih podataka.

Umesto da se menja stvarni tekst, dodaju se male nasumične promene u vektorskim prikazima rečenica. Na primer, ako se rečenica "The cat is on the mat" predstavi vektorima, šum može dodati male promene u te brojeve, ali će osnovno značenje ostati isto. Model se trenira na takvim promenjenim podacima kako bi postao otporniji na greške ili manipulacije.

### Interpolacione metode

Interpolacione metode prave nove podatke kombinovanjem dve ili više rečenica, ali ne direktno, već koristeći njihove numeričke prikaze (embeddings). Ovo pomaže modelima da bolje generalizuju i da budu otporniji na prekomerno prilagođavanje (overfitting).

#### SMOTE Interpolacija

**SMOTE** je metoda koja pomaže da se balansiraju podaci u zadacima klasifikacije. Umesto da jednostavno kopira postojeće podatke, SMOTE kreira nove instance tako što kombinuje slične instance iz iste klase.  
**Primer**: Ako imamo dve slične rečenice iz iste klase, SMOTE će stvoriti novu rečenicu kombinujući njihove vektorske prikaze. To pomaže u balansiranju skupa podataka.

#### Mixup Interpolacija

**Mixup** je metoda koja kombinuje dve različite rečenice i njihove klase kako bi stvorila novu rečenicu koja predstavlja neku vrstu "mešavine" obe. Ova metoda pomaže modelu da uči iz različitih klasa i da bolje generalizuje.  
**Primer**: Ako imamo rečenicu iz klase 0 ("The cat is on the mat") i rečenicu iz klase 1 ("The dog is barking"), mixup tehnika će stvoriti novu rečenicu koja je kombinacija obe, a i klasa će biti mešavina oba originalna labela.  
Mixup se često koristi u dubokim modelima (kao što su BERT ili RoBERTa), gde se ove kombinacije primenjuju na različitim slojevima mreže, čime se poboljšava razumevanje složenih obrazaca u tekstu.

## Praktični deo rada

### Opis dataset-a

Ovaj rad se fokusira na analizu i detekciju govora mržnje i uvredljivog jezika na Twitter-u korišćenjem posebnog skupa podataka nazvanog *hate\_speech\_offensive*. Dataset je pažljivo kreirana kolekcija anotiranih tvitova na engleskom jeziku, namenjena treniranju mašinskih modela za automatsko prepoznavanje govora mržnje i uvredljivog sadržaja. Skup podataka je dostupan u formatu CSV datoteke pod nazivom *train.csv* i nije podeljen na više delova—dostupan je samo trening set.

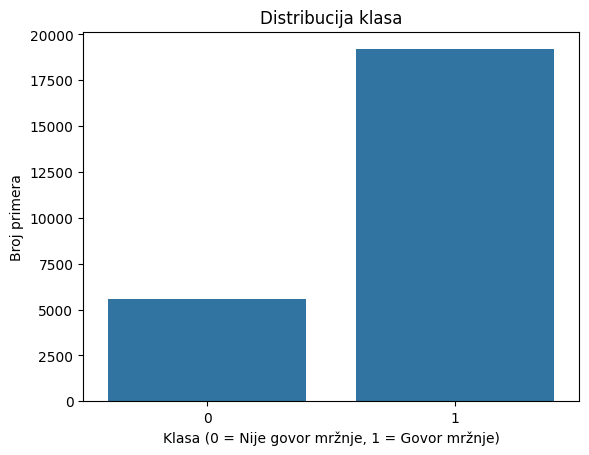
**Pregled skupa podataka**

Dataset sadrži nekoliko ključnih kolona koje pružaju detaljne informacije o klasifikaciji svakog tvita:

* **count**: Ukupan broj anotacija za svaki tvit.
* **hate\_speech\_count**: Broj anotacija koje klasifikuju tvit kao govor mržnje.
* **offensive\_language\_count**: Broj anotacija koje označavaju tvit kao uvredljiv jezik.
* **neither\_count**: Broj anotacija koje identifikuju tvit kao ni govor mržnje ni uvredljiv jezik.
* **class**: Konačna vrednost klase, 0 – ne pripada ni govoru mržnje ni uvredljivom govoru, 1 – pripada govoru mržnje, 2 – pripada uvredljivom govoru.

Podaci su prikupljeni putem javnog Twitter API-ja korišćenjem specifičnih ključnih reči povezanih sa govorom mržnje i uvredljivim jezikom. Nakon prikupljanja, tvitove je ručno anotiralo više anotatora koji su ih klasifikovali u odgovarajuće kategorije.

Radi jednostavnosti, u ovom projektu je fokus na detekciji govora mržnje, stoga su klase 0 i 2 objedinjene u jednu – nije govor mržnje.



Možemo primetiti da je dataset u početnom trenutku nebalansiran, pa ćemo primenjivati metode za augmentaciju podataka na klasi 0 (nije govor mržnje), kako bismo povećali broj instance te klase i samim tim dobili balansiran dataset.

### Preprocesiranje podataka

U cilju efikasnog treniranja modela za detekciju govora mržnje i uvredljivog jezika, neophodno je sprovesti detaljno preprocesiranje tekstualnih podataka. Predobrada teksta omogućava uklanjanje nepotrebnih informacija i normalizaciju podataka, što poboljšava performanse mašinskih algoritama.

**Koraci preprocesiranja:**

1. **Uklanjanje URL-ova i email adresa:** URL-ovi i email adrese ne doprinose semantičkom sadržaju tvita u kontekstu detekcije govora mržnje i mogu uneti šum u model. Korišćenjem regularnih izraza uklanjaju se sve instance URL-ova (počinju sa http, www) i email adresa.
2. **Uklanjanje HTML tagova:** HTML tagovi su tehnički elementi koji nisu relevantni za analizu teksta i mogu ometati procesiranje. Biblioteka **BeautifulSoup** koristi se za parsiranje i uklanjanje bilo kakvih HTML tagova koji mogu biti prisutni u tekstu.
3. **Uklanjanje emotikona i specijalnih karaktera:** Iako emotikoni mogu nositi emocionalni ton, često otežavaju procesiranje zbog kodiranja i mogu biti izvor šuma. Definiše se funkcija remove\_emojis koja koristi regularne izraze za identifikaciju i uklanjanje emotikona i specijalnih simbola.
4. **Uklanjanje specijalnih karaktera i interpunkcije:** Cilj je zadržati samo korisne tekstualne informacije i eliminisati potencijalno nerelevantne simbole. Uklanjaju se svi karakteri koji nisu slova (uključujući slova sa dijakritičkim znacima) i razmaci.
5. **Pretvaranje u mala slova:** Uniformisanje teksta olakšava procesiranje i osigurava da se reči poput "Govor" i "govor" tretiraju isto. Cela tekstualna kolona se konvertuje u mala slova.
6. **Uklanjanje dijakritika:** Normalizacija teksta za potrebe modela koji možda ne prepoznaju dijakritike, čime se smanjuje kompleksnost vokabulara. Korišćenjem biblioteke **unicodedata**, uklanjaju se dijakritički znakovi iz slova.
7. **Uklanjanje višestrukih razmaka:** Čisti tekst od nepotrebnih praznina koje mogu uticati na tokenizaciju. Višestruki razmaci se zamenjuju jednim razmakom, a vodeći i prateći razmaci se uklanjaju.
8. **Tokenizacija:** Tokenizacija je neophodan korak za mnoge NLP procese, omogućava individualnu obradu svake reči. Tekst se deli na individualne reči (tokene) korišćenjem biblioteke **nltk**.
9. **Uklanjanje stop-reči:** Stop-reči mogu zamagliti značajne obrasce u podacima; njihovo uklanjanje fokusira analizu na informativne reči. Iz tokena se uklanjaju uobičajene reči koje ne doprinose značenju.
10. **Lematizacija:** Lematizacija pomaže u smanjenju varijacija reči, što omogućava modelu da prepozna iste reči u različitim oblicima. Reči se svode na njihov osnovni ili korenski oblik korišćenjem **WordNetLemmatizer**.

### Primena modela nad osnovnim dataset-om

Korišćene metode uključuju klasične algoritme mašinskog učenja kao što su logistička regresija, mašina sa podrškom vektora (SVM) i Naive Bayes, kao i duboke neuronske mreže poput LSTM i CNN.

1. **Logistička regresija**

Logistička regresija je statistički model koji se koristi za binarnu klasifikaciju. Ona procenjuje verovatnoću da određeni ulaz pripada jednoj od dve moguće kategorije. U ovom slučaju, koristi se za predviđanje da li tvit sadrži govor mržnje (klasa 1) ili ne (klasa 0). Ovaj model je jednostavan za implementaciju i interpretaciju, što ga čini čestim izborom u NLP zadacima.

1. **SVM**

SVM je nadgledani algoritam za mašinsko učenje koji se koristi za klasifikaciju i regresiju. Cilj SVM-a je pronaći optimalnu hiper-ravan koja najbolje razdvaja klase u visokodimenzionalnom prostoru. U tekstualnoj klasifikaciji, SVM je efikasan zbog svoje sposobnosti da radi sa velikim brojem karakteristika koje proizilaze iz vektorizacije teksta.

1. **Naive Bayes klasifikator**

Naive Bayes je probabilistički klasifikator zasnovan na Bajesovoj teoremi, uz pretpostavku da su sve karakteristike međusobno nezavisne. Ova pretpostavka pojednostavljuje model i čini ga efikasnim za velike skupove podataka. Često se koristi u NLP zbog svoje brzine i efikasnosti, posebno kod problema klasifikacije teksta.

1. **Long Short-Term Memory (LSTM) neuronske mreže**

LSTM je tip rekurentne neuronske mreže (RNN) koja je dizajnirana da prevaziđe problem kratkoročne memorije u RNN-ovima. LSTM ima unutrašnje mehanizme, poput ćelijskih stanja i vrata, koji omogućavaju čuvanje informacija kroz duge sekvence. Ovo je posebno korisno u obradi teksta, gde kontekst i sekvencijalne informacije igraju ključnu ulogu.

1. **Konvolucione neuronske mreže (CNN)**

CNN su prvobitno razvijene za obradu slike, ali su uspešno primenjene i u NLP zadacima. U tekstualnoj klasifikaciji, CNN može da identifikuje lokalne obrasce u podacima, kao što su fraze ili n-grami, koji su značajni za klasifikaciju. CNN koristi konvolucione slojeve za ekstrakciju karakteristika i često je efikasniji od tradicionalnih RNN-ova u pogledu brzine treniranja.

**Analiza i poređenje rezultata**

Logistička regresija pruža solidne performanse sa ukupnom tačnošću od 90%. Klasa 1 se detektuje veoma dobro, sa visokim preciznošću (92%) i recall-om (95%), dok je balans za klasu 0 nešto slabiji, što ukazuje na povećan broj lažno pozitivnih predikcija za ovu klasu. Ovaj model se dobro pokazuje u zadacima gde je balans između obe klase manje važan, a prioritet se daje detekciji klase 1.

SVM model postiže istu tačnost kao logistička regresija, ali donosi bolji balans između preciznosti i recall-a za obe klase. Konkretno, recall za klasu 0 je viši (75%), što ukazuje na manji broj lažno pozitivnih predikcija. Takođe, performanse za klasu 1 ostaju na visokom nivou, sa preciznošću od 93% i recall-om od 95%. SVM pokazuje stabilnost i pouzdanost u scenarijima gde je važno obezbediti ravnotežu između metrika za obe klase.

Naive Bayes model se ističe visokim recall-om za klasu 1 (98%), što znači da gotovo svi primeri ove klase bivaju ispravno klasifikovani. Međutim, to dolazi po cenu veoma niskog recall-a za klasu 0 (37%), što ukazuje na značajnu pristrasnost prema klasi 1. Ovaj model bi mogao biti koristan u situacijama gde je ključno minimizovati propuštanje pozitivnih primera, ali nije pogodan za scenarije gde su greške u klasi 0 kritične.

LSTM model postiže tačnost od 89.3% i pokazuje odlične performanse za klasu 1, sa balansom između preciznosti i recall-a. Ipak, recall za klasu 0 je nešto niži (73%), što ukazuje na slabiju sposobnost modela da ispravno klasifikuje primere ove klase. Iako LSTM može da uhvati kompleksne obrasce u podacima, njegova složenost i zahtevi za resursima čine ga manje efikasnim za manje složene zadatke.

CNN model, s druge strane, pokazuje nešto slabiju tačnost (87%) u poređenju sa ostalim modelima. Njegov recall za klasu 0 je najniži među analiziranim modelima (68%), što ukazuje na teškoće u detekciji ove klase. Performanse za klasu 1 su i dalje dobre, ali slabiji balans između klasa čini ovaj model manje pogodnim za primene gde je jednaka tačnost za sve klase ključna.

Zaključno, SVM se izdvaja kao najstabilniji model sa dobrim balansom performansi za obe klase. Naive Bayes je optimalan izbor u scenarijima gde je recall za klasu 1 ključan, dok su LSTM i CNN modeli prikladni za zadatke gde se očekuje složenost u podacima, iako zahtevaju dodatnu optimizaciju za bolju generalizaciju. Logistička regresija pruža konzistentne i jednostavne rezultate, ali sa slabijim balansom između klasa u odnosu na SVM.

### Metode za augmentaciju podataka

U cilju poboljšanja performansi modela i povećanja raznolikosti skupa podataka, definisane su sledeće metode augmentacije tekstualnih podataka:

1. **Simulacija pravopisnih grešaka (simulate\_spelling\_errors)**: Umeće pravopisne greške u tekst sa određenom verovatnoćom kako bi model bio otporniji na greške u pisanju.
2. **Simulacija grešaka u kucanju (keyboard\_augmenter)**: Simulira tipične greške koje nastaju zbog blizine tastera na tastaturi, oponašajući ljudske greške pri kucanju.
3. **Simulacija OCR grešaka (ocr\_simulation)**: Umeće greške karakteristične za optičko prepoznavanje karaktera, povećavajući robusnost modela na takve nesavršenosti.
4. **Nasumična augmentacija karaktera (random\_character\_augmentation)**: Uključuje brisanje, zamenu, permutaciju ili umetanje karaktera u tekst kako bi se generisale različite varijante.
5. **Zamena sinonimima (synonym\_replacement)**: Zamenjuje određeni broj reči njihovim sinonimima, obogaćujući vokabular i uvodeći semantičku raznolikost.
6. **Zamena antonimima (antonym\_replacement)**: Zamenjuje reči njihovim antonimima, što testira sposobnost modela da razume promene u značenju.
7. **Nasumično umetanje reči (random\_insertion)**: Umeće sinonime nasumično u tekst, povećavajući dužinu i složenost rečenica.
8. **Nasumična zamena reči (random\_swap)**: Menja mesta dvema rečima u tekstu, stvarajući sintaksičke varijacije.
9. **Nasumično brisanje reči (random\_deletion)**: Briše reči iz teksta sa određenom verovatnoćom, pomažući modelu da se nosi sa nepotpunim informacijama.
10. **Podela reči (split\_augmentation)**: Deli reči na manje delove, simulirajući greške u pisanju ili kucanju.
11. **Augmentacija pravopisnim greškama (spelling\_augmentation)**: Umeće pravopisne greške direktno u reči, povećavajući robusnost modela na pravopisne varijacije.
12. **Zamena na osnovu embeddinga (embedding\_replacement)**: Zamenjuje reči sličnim rečima koristeći pretrenirane rečničke vektore, uvodeći semantičku sličnost.
13. **Augmentacija kontekstualnim embeddingom (contextual\_embedding\_augmentation)**: Koristi kontekstualne modele poput BERT-a za zamenu reči u skladu sa kontekstom, čuvajući koherentnost teksta.
14. **Parafraziranje teksta (paraphrase\_text)**: Generiše parafraze originalnog teksta koristeći modele poput T5, obogaćujući skup podataka različitim izrazima istog značenja.
15. **Back-Translation (back\_translation)**: Prevođenjem teksta na drugi jezik i nazad dobija se parafraziran tekst koji zadržava originalno značenje.
16. **Nasumična augmentacija rečenica (random\_sentence\_augmentation)**: Menja redosled rečenica u tekstu, uvodeći varijacije u strukturi i toku misli.
17. **Generisanje teksta GPT-2 modelom (generate\_text\_gpt2)**: Generiše novi tekst na osnovu početnog prompta koristeći GPT-2 model, kreirajući potpuno nove instance podataka.

Radi jednostavnosti, samo neke od ovih metoda će biti primenjene u svrhu augmentacije seta podataka, čime će se povećati njegova raznolikost i poboljšati generalizacija modela za detekciju govora mržnje i uvredljivog jezika.

### Primena modela nad augmentiranim dataset-ovima

Nakon primene metoda za augmentaciju podataka na klasu 0 (nije govor mržnje), kreirana su dva augmentirana skupa podataka kako bi se dodatno poboljšala raznovrsnost i balansiranost skupa. Na svakom od ovih skupa podataka primenjene su iste mašinske metode kako bi se analizirali efekti različitih tehnika augmentacije na performanse modela. Sledeća analiza obuhvata rezultate dobijene na oba augmentirana skupa podataka.

#### 4.5.1. Primena modela nad prvim augmentiranim dataset-om

Prvi augmentirani dataset je kreiran korišćenjem sledećih metoda: simulacija pravopisnih grešaka (simulate\_spelling\_errors), zamena sinonima (synonym\_replacement), nasumična augmentacija rečenica (random\_character\_augmentation) i splitovanje (split\_augmentation). Nakon augmentacije, skup podataka je balansiran sa 19,189 instanci klase 0 i 19,190 instanci klase 1.

Augmentacija je doprinela povećanju raznovrsnosti podataka, što se ogleda u poboljšanoj tačnosti, boljoj ravnoteži između klasa i poboljšanim metrikama performansi u poređenju sa prethodnim rezultatima bez augmentacije.

**Logistička regresija** beleži tačnost od 91%, uz F1-score vrednosti od 91 za obe klase. Klasa 0 ima visoki recall (93%), što znači da model uspešno prepoznaje većinu negativnih primera, dok klasa 1 ima nešto niži recall (89%), ukazujući na mali broj propuštenih pozitivnih primera. Augmentacija je značajno doprla do ravnoteže između klasa, što je očigledno u poboljšanim vrednostima F1-score za klasu 0 u poređenju sa osnovnim datasetom.

**SVM** pokazuje najbolje performanse među svim modelima sa tačnošću od 93%. Klasa 0 ima F1-score od 93%, dok klasa 1 takođe postiže F1-score od 93%. Recall za klasu 0 je izuzetno visok (95%), što ukazuje na sposobnost modela da minimizira lažno pozitivne predikcije. Napredne metode augmentacije dodatno su poboljšale stabilnost modela i njegovu sposobnost da balansira između klasa.

**Naive Bayes** beleži ukupnu tačnost od 86%, uz značajno poboljšanje u F1-score za klasu 0 (85%) u poređenju sa osnovnim datasetom. Klasa 1 održava visok recall (90%), ali model i dalje pokazuje pristrasnost prema klasi 1, što se ogleda u većem broju lažno pozitivnih predikcija za klasu 0. Napredne metode augmentacije smanjuju ovu pristrasnost, ali se Naive Bayes i dalje suočava sa inherentnim ograničenjima u ravnoteži između klasa.

**LSTM** postiže tačnost od 92%, sa gotovo savršenim balansom između klasa. Klasa 0 beleži F1-score od 93%, dok klasa 1 ima F1-score od 92%. Augmentacija podataka omogućila je modelu da bolje generalizuje obrasce, smanjujući broj grešaka u predikciji obe klase. Ovo potvrđuje da LSTM efikasno koristi dodatnu raznovrsnost podataka kako bi poboljšao performanse.

**CNN** dostiže tačnost od 91%, uz F1-score od 92 za klasu 0 i 91 za klasu 1. Model ima visoki recall za klasu 0 (93%), što ukazuje na njegovu sposobnost da tačno klasifikuje negativne primere, ali se suočava sa nešto nižim recall-om za klasu 1 (89%), što rezultira blago povećanim brojem lažno negativnih predikcija. Napredne metode augmentacije značajno su poboljšale balans performansi između klasa, iako CNN pokazuje manje stabilnosti u poređenju sa SVM-om i LSTM-om.

Zaključno, primena metoda augmentacije podataka očigledno je doprinela poboljšanju svih modela, ali su SVM i LSTM pokazali najbolje rezultate, sa visokom tačnošću i izuzetnim balansom između klasa. Logistička regresija ostaje stabilna i konzistentna, dok su CNN i Naive Bayes ostvarili značajan napredak, iako još uvek zaostaju za vodećim metodama. Augmentacija podataka se pokazala ključnom za povećanje generalizacije modela i njihovu sposobnost da se nose sa raznovrsnim obrascima.

#### 4.5.2. Primena modela nad drugim augmentiranim dataset-om

Drugi augmentirani dataset je kreiran korišćenjem naprednijih metoda augmentacije podataka putem **TextAttack** biblioteke, uključujući **CLAREAugmenter** i **EmbeddingAugmenter**. Ove metode omogućavaju kontekstualno prilagođavanje i raznovrsnije transformacije tekstualnih podataka, što doprinosi većoj raznolikosti i kvalitetu augmentisanih podataka.

Nakon augmentacije, skup podataka je balansiran sa 18,593 instanci klase 0 i 19,190 instanci klase 1, čime se eliminiše neravnoteža između klasa.

Analiza rezultata ukazuje na značajno poboljšanje performansi modela usled primene naprednih metoda augmentacije. Ove tehnike su omogućile modelima da efikasnije generalizuju i preciznije klasifikuju podatke, što se ogleda u povećanju tačnosti, ravnoteži metrika između klasa, i smanjenju grešaka u predikcijama.

Logistička regresija ostvaruje tačnost od 91%, sa ravnomernim performansama za obe klase. Klasa 0 ima preciznost od 90% i recall od 92%, dok klasa 1 beleži obrnuti balans sa preciznošću od 92% i recall-om od 90%. Augmentacija je ovde očigledno doprinela poboljšanju sposobnosti modela da prepoznaje različite obrasce, smanjujući greške u predikciji i obezbeđujući stabilan učinak.

SVM model pruža najbolje rezultate sa tačnošću od 93%, uz impresivan balans između klasa. Klasa 0 postiže preciznost od 92% i recall od 95%, dok klasa 1 dostiže preciznost od 95% i recall od 92%. U poređenju sa prethodnim eksperimentima, napredne metode augmentacije dodatno su pojačale efikasnost ovog modela, čineći ga izuzetno robusnim za klasifikaciju u oba razreda.

Naive Bayes beleži značajan napredak sa tačnošću od 86%, ali i dalje zaostaje za ostalim modelima. Klasa 0 ima preciznost od 90% i recall od 81%, dok klasa 1 dostiže preciznost od 83% i recall od 92%. Iako napredne metode augmentacije pomažu u povećanju recall-a za klasu 1, model se i dalje suočava sa pristrasnošću ka klasi 1, što rezultira većim brojem lažno pozitivnih predikcija za klasu 0.

LSTM model se ističe svojom tačnošću od 92% i gotovo savršenim balansom između klasa. Klasa 0 ima preciznost i recall od 92%, dok klasa 1 beleži identične metrike. Ovo ukazuje na to da LSTM uspešno koristi informacije iz augmentovanih podataka za prepoznavanje kompleksnih obrazaca. Napredne metode augmentacije dodatno su povećale njegovu generalizaciju i sposobnost prepoznavanja svih klasa sa visokim nivoom tačnosti.

CNN model takođe pokazuje impresivne rezultate, sa tačnošću od 92%. Klasa 0 postiže preciznost od 91% i recall od 93%, dok klasa 1 ima preciznost od 93% i recall od 91%. CNN koristi napredne metode augmentacije kako bi se bolje prilagodio raznovrsnim podacima, čime se smanjuju greške u predikciji i obezbeđuju uravnotežene performanse za obe klase.

Ovi rezultati pokazuju da su naprednije metode augmentacije, poput onih iz **TextAttack** biblioteke, doprinele boljoj generalizaciji modela i višim performansama u detekciji govora mržnje. Klasični modeli, posebno SVM i Logistička regresija, nastavljaju da pokazuju odlične performanse, dok su duboke neuronske mreže takođe pokazale značajno poboljšanje, što sugeriše da dodatna augmentacija podataka može doprineti njihovoj superiornosti u složenijim zadacima klasifikacije.

## Tabelarni prikaz i analiza rezultata

**Logistička regresija:**

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | Tačnost (%) | F1 score – klasa 0 | F1 score – klasa 1 |
| Osnovni | 90 | 76 | 93 |
| Prvi augmentirani | 91 | 91 | 91 |
| Drugi augmentirani | 91 | 91 | 91 |

**SVM:**

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | Tačnost (%) | F1 score – klasa 0 | F1 score – klasa 1 |
| Osnovni | 90 | 78 | 94 |
| Prvi augmentirani | 93 | 93 | 92 |
| Drugi augmentirani | 93 | 93 | 93 |

**Naive Bayes:**

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | Tačnost (%) | F1 score – klasa 0 | F1 score – klasa 1 |
| Osnovni | 84 | 52 | 91 |
| Prvi augmentirani | 86 | 85 | 86 |
| Drugi augmentirani | 86 | 85 | 87 |

**LSTM:**

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | Tačnost (%) | F1 score – klasa 0 | F1 score – klasa 1 |
| Osnovni | 89 | 76 | 93 |
| Prvi augmentirani | 92 | 93 | 92 |
| Drugi augmentirani | 92 | 92 | 92 |

**CNN:**

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | Tačnost (%) | F1 score – klasa 0 | F1 score – klasa 1 |
| Osnovni | 87 | 70 | 92 |
| Prvi augmentirani | 91 | 91 | 91 |
| Drugi augmentirani | 92 | 92 | 92 |

# Zaključak

Ovaj rad je pružio sveobuhvatan pregled metoda za augmentaciju tekstualnih podataka u kontekstu mašinskog učenja. Počeli smo sa uvodom u značaj kvalitetnih i raznovrsnih podataka za izgradnju efikasnih modela, posebno fokusirajući se na izazove povezane sa prikupljanjem i obradom tekstualnih podataka.

U drugom delu rada detaljno smo obradili korake predobrade tekstualnih podataka, uključujući čišćenje teksta, normalizaciju, tokenizaciju, lematizaciju, uklanjanje stop-reči, spajanje kontrakcija i uklanjanje duplikata. Ovi koraci su ključni za pripremu podataka, osiguravajući njihovu doslednost i relevantnost za dalju analizu.

Treći deo rada posvetili smo pregledavanju različitih tehnika augmentacije tekstualnih podataka, koje smo podelili u dve glavne kategorije: Data Space i Feature Space. Unutar Data Space, istražili smo metode na nivou karaktera, reči, fraza i rečenica, kao i na nivou dokumenta, uključujući tehnike poput back-translation i generativnih modela. U okviru Feature Space, fokusirali smo se na indukciju šuma i interpolacione metode, koje manipulišu numeričkim reprezentacijama teksta kako bi stvorile nove varijacije podataka.

Praktični deo rada obuhvatio je analizu specifičnog dataset-a za detekciju govora mržnje na Twitter-u. Kroz opis dataset-a, primenu predobrade podataka, implementaciju različitih modela mašinskog učenja, te primenu različitih metoda augmentacije, demonstrirali smo kako ove tehnike mogu unaprediti balansiranost i raznovrsnost skupa podataka.

Zaključno, ovaj rad je istakao važnost kombinovanja različitih metoda augmentacije kako bi se poboljšala performansa modela mašinskog učenja u obradi tekstualnih podataka. Primenom raznovrsnih tehnika, moguće je značajno povećati kvalitet i kvantitet dostupnih podataka, čime se modeli bolje prilagođavaju realnim uslovima i složenim zadacima klasifikacije.

# Reference

Agrawal, R. (2021, June 14). *Must Known Techniques for text preprocessing in NLP*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/06/must-known-techniques-for-text-preprocessing-in-nlp/

Arslan, E. (2024, June 25). *Natural language Processing: Deep dive into text preprocessing and tokenization— step 2*. Medium. https://medium.com/@erhan\_arslan/natural-language-processing-deep-dive-into-text-preprocessing-and-tokenization-step-2-b5dcf9520266

Aydin, A. (2023, October 4). *1 — text preprocessing techniques for NLP*. Medium. https://ayselaydin.medium.com/1-text-preprocessing-techniques-for-nlp-37544483c007

Bolle, M. (2023, June 3). *Text augmentation in python with NLPAUG - Marc Bolle*. Medium. https://medium.com/@marc.bolle/text-augmentation-in-python-with-nlpaug-48c3eebacf46

Chiusano, F. (2022, April 4). *Two minutes NLP — A taxonomy of data augmentation for text classification*. NLPlanet. https://medium.com/nlplanet/two-minutes-nlp-a-taxonomy-of-data-augmentation-for-text-classification-52c96f332bad

Claude, C. (2018). Text data augmentation made simple by leveraging NLP Cloud APIs. In *arXiv [cs.CL]*. http://arxiv.org/abs/1812.04718

*Getting started with text preprocessing*. (2019, August 20). Kaggle.com; Kaggle. https://www.kaggle.com/code/sudalairajkumar/getting-started-with-text-preprocessing

Li, B., Hou, Y., & Che, W. (2022). Data augmentation approaches in natural language processing: A survey. *AI Open*, *3*, 71–90. https://doi.org/10.1016/j.aiopen.2022.03.001

Ma, E. (2019, April 20). *Data Augmentation library for text*. Towards Data Science. https://towardsdatascience.com/data-augmentation-library-for-text-9661736b13ff

Markus, B., Marc-André, K., & Christian, R. (2021). A Survey on Data Augmentation for Text Classification. In *arXiv [cs.CL]*. http://arxiv.org/abs/2107.03158

Pellicer, L. F. A. O., Ferreira, T. M., & Costa, A. H. R. (2023). Data augmentation techniques in natural language processing. *Applied Soft Computing*, *132*(109803), 109803. https://doi.org/10.1016/j.asoc.2022.109803

*Text augmentation techniques in NLP*. (2023, April 17). GeeksforGeeks. https://www.geeksforgeeks.org/text-augmentation-techniques-in-nlp/

Van Otten, N. (2023, January 25). *How to use text normalization techniques in NLP with python [9 ways]*. Spot Intelligence. https://spotintelligence.com/2023/01/25/text-normalization-techniques-nlp/