УНИВЕРЗИТЕТ У БЕОГРАДУ

ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ

СОФТВЕРСКО ИНЖЕЊЕРСТВО ВЕЛИКИХ БАЗА ПОДАТАКА

Семинарски рад

Професор:

проф. др. Мирослав Бојовић

Студент:

Андријана Миковић

2023/3042

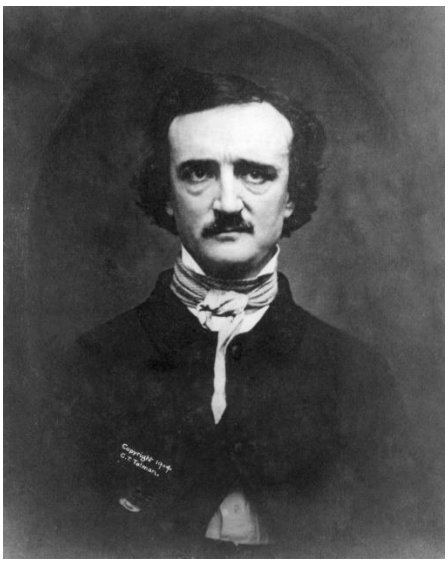
Београд, јануар 2024.

САДРЖАЈ

1. Увод

У домену књижевности, анализа ауторског стила представља фасцинантан изазов, којем се може приступити помоћу техника машинског учења. У овом семинарском раду је решаван проблем идентификације аутора хорор жанра (енгл. *Spooky Author identification*) [1]. Који се заснива на препознавању стила сваког од аутора, и идентификацији ко је написао одређени текст. Овај проблем ставља акценат на три аутора која су оставила велики траг у историји хорор књижевности, Едгар Алан По, Хаурд Филипс Лафкрафт и Мери Шели.

Едгар Алан По, приказан на слици 1.1 , је један од најпознатијих америчких писаца 19. века, био је песник, књижевни критичар и један од пионира хорор жанра. Стекао је славу кроз своје мрачне и мистериозне приче, често истраживајући дубине људске психе. Познат је по својој способности стварања језивих атмосфера и неочекиваних заплета. Његови ликови често су опсесивни, а приче препуне елемената туге, губитка и лудила.



Слика 1.1. Едгар Алан По

Хаурд Филипс Лафкрафт, кога можемо видети на слици 1.2, био је пионир у доношењу космичке хорор димензије у књижевности. Његове приче често истражују несхватљиве и страшне силе које постоје изван људске перцепције. Познат је по стварању богатог митолошког света са божанствима попут *Cthulhu-a* , чије присуство изазива дубоки страх од

непознатог. Његова дела, као што су *“The Call of Cthulhu”* и *“At the Mountains of Madness”* , проширила су границе маште за даљи развој хорор-а.

Слика 1.2. Хаурд Филипс Лафкрафт

Мери Шели, слика 1.3, ауторка класичног романа *“Frankenstein”,* кроз који истражује етичке и моралне дилеме везане за стварање живота. Оставила је неизбрисив траг у хорор жанру, а њена дела су и даље релевантна. Често је сматрају једном од пионира жанра хорора и научне фантастике.

Слика 1.3. Мери Шели

У наредном поглављу извршена је анализа самог проблема идентификације аутора хорор жанра [1]. База података, која се састоји од текстова три аутора, обрађена је у трећем поглављу. У четвртом поглављу приказани су потребни кораци трансформације текста, да би се он касније користио као улаз за алгоритме класификације као и неуралне мреже. Пето поглавље приказује алгоритме машинског учења коришћене у класификацији, а шесто се бави моделима неуралних мрежа коришћених за решавање проблема. У седмом поглављу су приказане коришћене метрике као и добијени резултати коришћењем различитих алгоритама и модела. На крају је дат закључак резултата семинарског рада.

2. АНАЛИЗА ПРОБЛЕМА

У овом поглављу, биће анализиран проблем идентификације аутора хорор жанра (енгл. *Spooky Author identification*) [1]. Који представља проблем проналажења личног печата аутора текста и његове накнадне идентификације, креирањем модела машинског учења. Текстови припадају једном од три аутора, Едгар Алан По, Хаурд Филипс Лафкрафт и Мери Шели.

2.1. Улазни фајлови

Постоје три доступна фајла, од којих један представља пример излаза потребног за такмичење, а друга два представљају тренинг и тест скуп података. Фајл *train.csv* садржи следеће информације:

- *id* – јединствени идентификатор сваке реченице
- *text* – неки текст написан од стране једног од аутора
- *autor* – информацију ко је аутор одрађене реченице

Поље аутор је кодирано на следећи начин

- *EAP* = Едгар Алан По
- *HPL* = Хаурд Филипс Лафкрафт
- *MWS* = Мери Шели

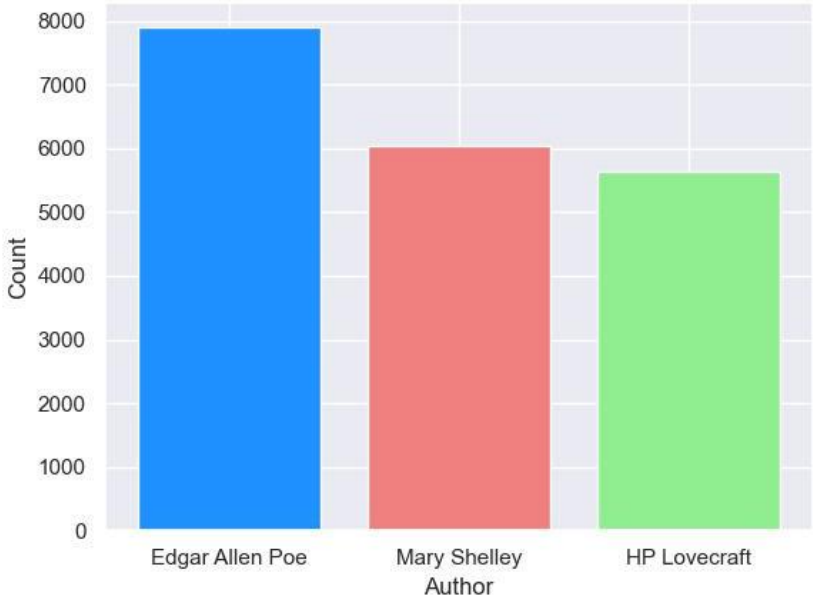
Док у *test.csv* недостаје последња колона, која предвиђа аутора датог текста. Циљ такмичења је био да се ти подаци класификују уз помоћ алгоритама, и решење представи по структури *sample_submission.csv* фајла, који се преко *id* повезује са тест скупом података. У овом семинарском вршена је подела самог тренинг сет података, на два подскупа и на тај начин је тестирана ефикасност алгоритама.

3. БАЗА ПОДАТАКА

У овом поглављу биће дат опис базе података коришћене у овом истраживању. Анализа коришћених речи од стране одговарајућих аутора.

3.1. Опште информације

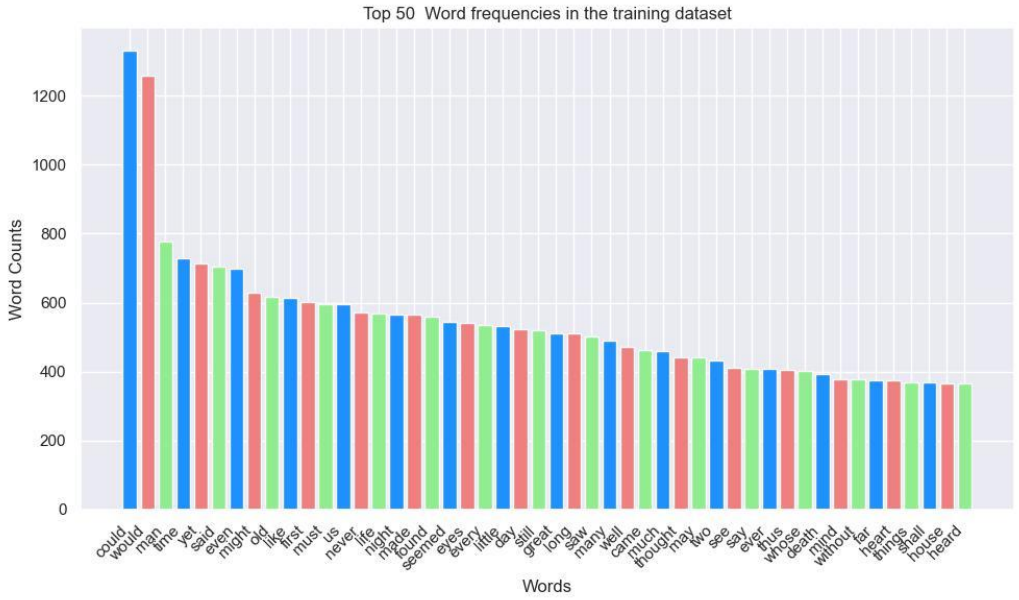
Део базе предвиђен за тренирање се састоји од 19579 редова, чија дистрибуција броја података по ауторима је приказана на слици 3.1.1. Примећујемо да база добро покрива разноврсности аутора, великом количином текста који припада сваком од њих, иако је број редова чији је аутор Поe већи од осталих.



Слика 1.3. Мери Шели

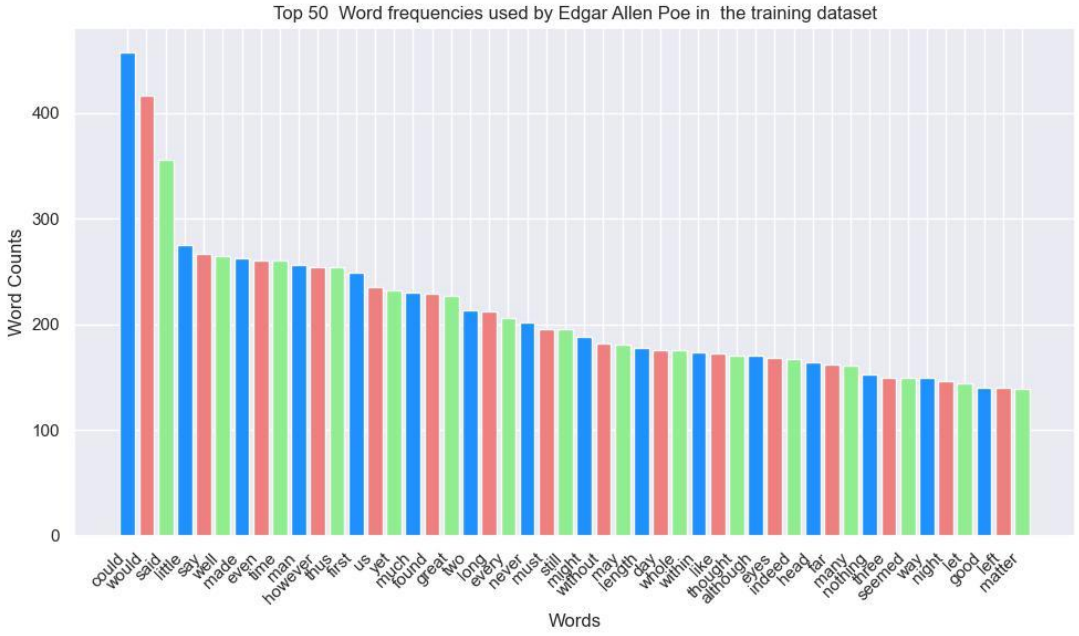
3.2. Најчешће коришћене речи

Различити аутори имају различите стилове писања и преференце речника. Анализирајући најчешће коришћене речи можемо утврдити специфичне језичке изборе за сваког од аутора. На слици 3.2.1. видимо најчешће коришћене речи на целој бази података, након отклањана *stopwords* из енглеског језика, о чему ће бити више реч у наредном поглављу, а које представљају често коришћене речи, које не носе значајне сематичке вредности.



Слика 3.2.1. 50 Најчешће коришћених речи на целој бази података

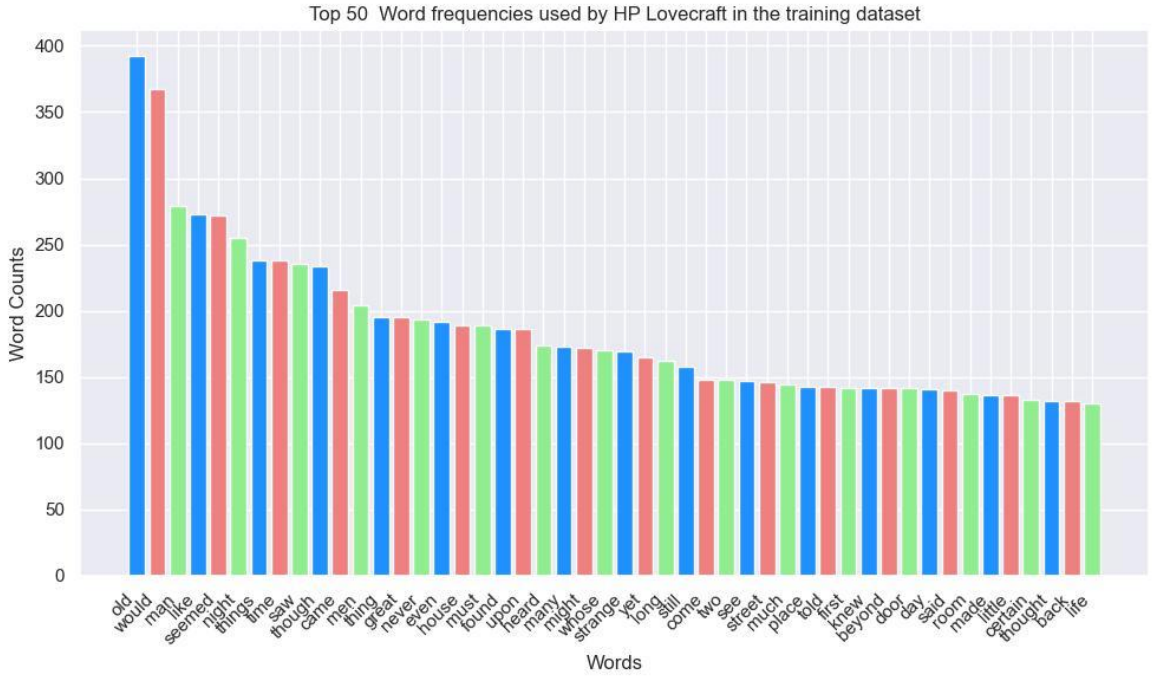
Сада можемо погледати како се ова дистрибуција разликује за сваког појединачног аутора. На слици 3.2.2. налазе се најчешће коришћене речи од стране Едгарда Алана Поа.



Слика 3.2.2. 50 Најчешће коришћених речи Едгар Алан По

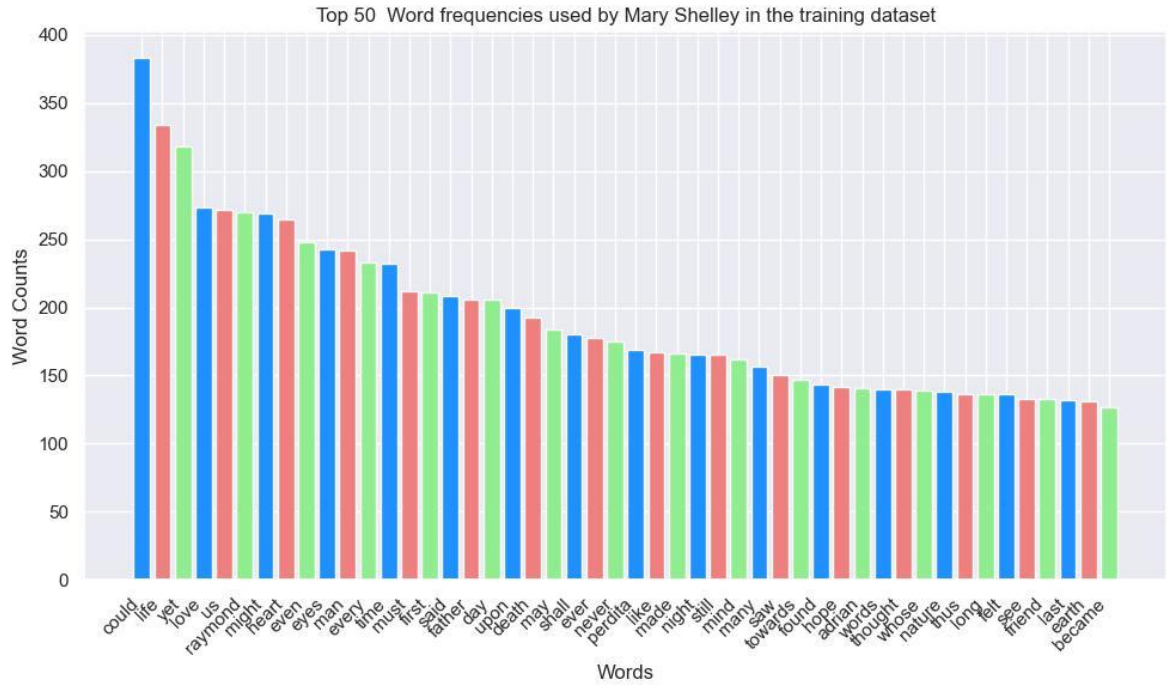
Већ од треће речи примећујемо разлику у односу на целокупну базу података, поклапање прве две речи може бити због малог дизбаланса базе, и већинског текста овог аутора. Док се код

Х.П. Лафкарта најчешће коришћена реч, слика 3.2.3., разликује и не налази се уопште у првих 50 најчешће коришћених речи од стране Поа.



Слика 3.2.3. 50 Најчешће коришћених речи Хаурд Филипс Лафкрафт

На слици 3.2.4. видимо 50 најчешће коришћених речи од стране Мери Шели, и да она доприноси највећој фреквенцији речи *could* , заједно са Пом.



Слика 3.2.4. 50 Најчешће коришћених речи Мери Шели

4. ТРАНСФОРМАЦИЈА ТЕКСТА

Пре успешне анализе и класификације улазног текста, потребно га је прилагодити. Извршене су следеће манипулације текста:

1. Претварање текста у мала слова
2. Процес лематизације
3. Отклањање *stopwords* енглеског језика
4. Отклањање знакова интерпукција и осталих карактера који се не налазе у скупу представљеним формулом 1

$$[a - zA - Z0 - 9\_] \quad (1)$$

4.1 . Лематизација

Процес лематизације је лингвистичка и техничка обрада природног језика, која подразумева смањење речи на основни облик, познат као лема. Лема је канонски, облик речи, који се не мора нужно поклапати са облицима које видимо у стварној употреби. Потребно је да лема постоји као реч у речнику језика на коме је текст писан, у овом случају енглеском језику.

Овај процес је рађен користећи *WordNetLemmatizer* класу из *nltk* библиотеке. На слици 4.1. приказан је код за процес лематизације, у којем се такође врши и трансформација текста у мала слова.

```
#Lemmatization
lemm = WordNetLemmatizer()
Andrijana Mikovic
def lemamatization_process(text):
    words = word_tokenize(text)
    lemmatized_words = [lemm.lemmatize(word.lower()) for word in words]
    lematized_text = " ".join(lemmatized_words)
    return lematized_text
```

Слика 4.1.1. Процес лематизације

4.2 . Отклањање *stopwords*

Stopwords представља скуп речи које су често коришћене у посматраном језику, али се сматрају да имају малу информациону вредност и значење. Па самим тим не помажу у идентификацији аутора датог текста. Коришћена је листа ових речи из *nltk* библиотеке, направљена за енглески језик. Код за отклањање ових речи као и знакова интерпукције се налази на слици 4.2.1.

```
def clean_text(text):
    first_text_list = word_tokenize(text)
    stop_words = nltk.corpus.stopwords.words('english')
    first_text_list_cleaned = ' '.join(word for word in first_text_list if word.lower() not in stop_words)
    first_text_list_cleaned = re.sub(r'\W', ' ', first_text_list_cleaned)
    return first_text_list_cleaned
```

Слика 4.2.1. Процес отклањања *stopwords*

4.3 . Преглед трансформисаног текста

Након извршених потребних трансформација, можемо погледати њихов резултат на првих 5 редова, базе података. На слици 4.3.1. приказан је резултат ове трансформације, где *text* колона представља оригинални текст, а *cleaned_text* колона је текст након извршавања свих корака трансформације.

	text	cleaned_text
0	This process, however, afforded me no means of...	process however afforded mean ascertaining...
1	It never once occurred to me that the fumbling...	never occurred fumbling might mere mistake
2	In his left hand was a gold snuff box, from wh...	left hand wa gold snuff box capered hill ...
3	How lovely is spring As we looked from Windsor...	lovely spring looked windsor terrace sixteen f...
4	Finding nothing else, not even gold, the Super...	finding nothing else even gold superintend...

Слика 4.3.1. Излаз након трансформације текста

5. МОДЕЛИ МАШИНСКОГ УЧЕЊА

У овом поглављу биће представљена четири алгоритма машинског учења коришћена за класификацију датог текста. Након трансформације текста извршена је подела на одлике (енгл. *features*) и мете (енгл. *targets*). Одлике представља колоне *cleaned_text,* док мете података представља колона аутор. Затим је извршена подела на податке за тренирање (80%) и податке за тестирање (20%), помоћу *train_test_split* методе из *sckit-learn* библиотеке.

5.1. Векторизација

Подаци за тренирање који представљају одлике се налазе у текстуалном формату, који није погодан за алгоритме класификације. Потребно га је претворити у нумерички облик. Ова трансформација је извршена помоћу *TfidfVectorizer (Term Frequency-Inverse Document Frequeny Vectorizer)* класе из *scikit-learn* библиотеке. Наведена класа има методу *fit_transform* која речима из улазног текста додељује одговарајуће тежине, чиме се приказује колику важност нека реч има у датом текстуалном документу.

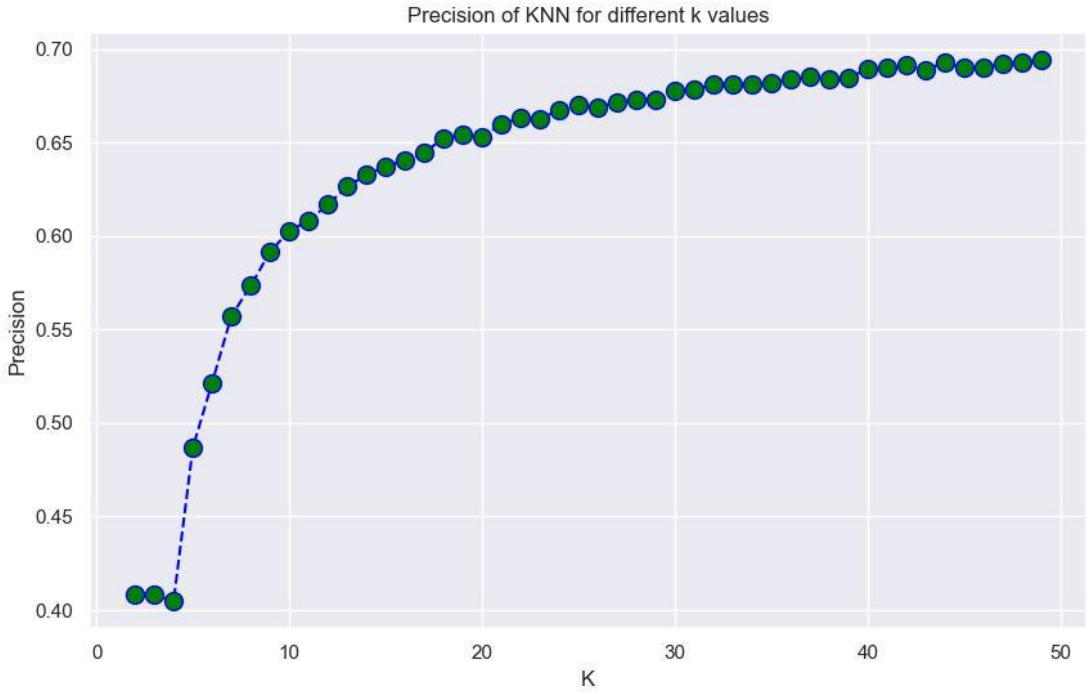
5.2. Логистичка регресија

Логистичка регресија представља математички модел којим се описује веза између атрибута и категоричке зависне променљиве. Она враћа вероватноћу да дати скуп атрибута, са специфичним вредностима, припада свакој од категорија. У оквиру овог истраживања, примењена је бинарна логистичка регресија, јер је потребно утврдити да ли особа пати од дијабетес или не. Овај модел се ослања на употребу логистичке функције како би повезао улазне атрибуте и пружио вероватноћу припадности одређеној категорији.

5.3. КНН – к најближих комшија

Алгоритам к најближих суседа скр. KNN (енгл. *k-nearest neighbors*) представља непараметарски метод, надгледног учења. Циљ КНН класификације је додељивање нових узорака одређеним класама на основу сличности са већ познатим тренинг скупом података. Користи метрику удаљености како би мерио сличности између узорака у простору атрибута.

Ова метрика се користи за проналажење *к* најближих суседа за сваки нови узорак. Параметар *к* представља број најближих суседа који ће бити узети у обзир приликом класификације новог узорка. Промену прецизности алгоритма у зависности од вредности параметра можемо видети на слици 5.3.1.



Слика 5.3.1. Прецизност КНН алгоритма за различите вредности параметра

5.4. Стабло одлучивања

Стабло одлучивања представља алгоритам машинског учења који се базира на концепту доношења одлука у складу са условима и акцијама. Карактерише га јасна графичка структура, у облику стабла. У стаблу одлучивања, корен представља почетни услов који одређује избор параметра на основу кога се врши раздвајање података. Гране стабла представљају различите услове који се примењују на податке, а листови садрже финалне одлуке, тј. излазне резултате алгоритма. Процес раздвајања на гране захтева меру информационог добитка. Информациони добитак је колико раздвајање на одређени начин помаже у смањењу неодлучности. За сваки могући услов врши се рачунање информационог добитка и одабира се најбољи пут за раздвајање, тј. стабло се изграђује тако да минимизује неодлучност и максимизује информациони добитак.

5.5. Мултиномијални наивни Бајесов класификатор

Представља један од алгоритама за класификацију у области машинског учења, који се заснива на Бајесовој теореми [2]. Термин наивни потиче од претпоставке да су сви атрибути независни, што често није случај. Мултиномијални се користи када су подаци представљени као бројност (фреквенција), као што је често случај у анализи текста.

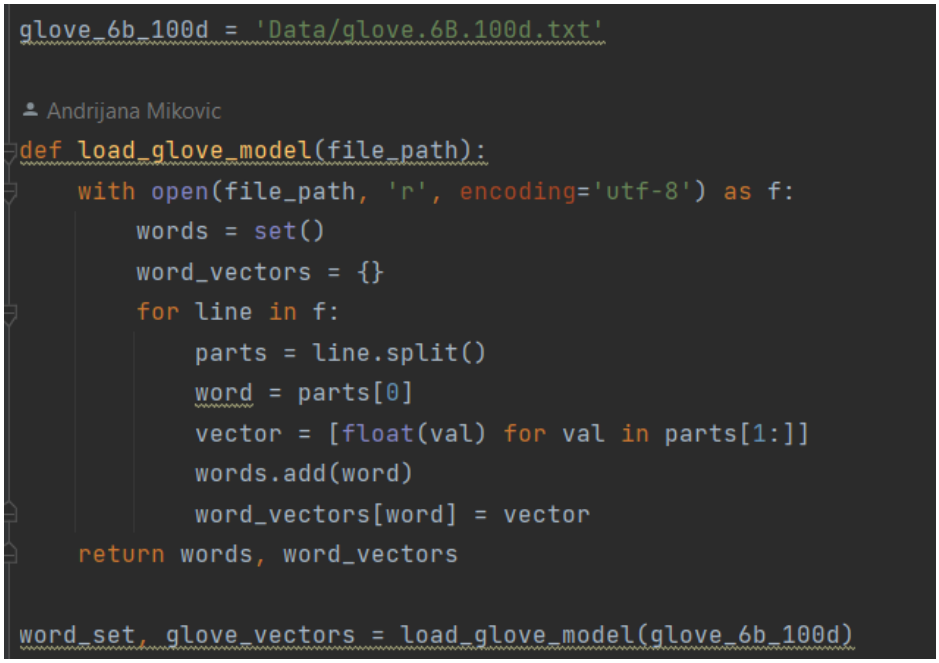
6. МОДЕЛИ НЕУРАЛНИХ МРЕЖА

У овом поглављу биће представљена три модела неуралних мрежа коришћена за класификацију датог текста. Као и њихови резултати. Након трансформације текста извршена је подела на одлике (енгл. *features*) и мете (енгл. *targets*). Одлике представља колоне *cleaned_text,* док мете података представља колона аутор. Затим је извршена подела на податке за тренирање (80%) и податке за тестирање (20%), помоћу *train_test_split* методе из *sckit-learn* библиотеке.

6.1. Токенизација и *GloVe*

Пре самог коришћена модела неуралних мрежа, потребно је извршити додатне припреме текста. Прво је извршена токенизација коришћењем класе *Tokenizer* из *keras* библиотеке, на овај начин су одређене јединствене речи које постоје у текстовима, и њима је додељен одговарајући индекс. Ови индекси су корисни за представљање текста као низа бројева. За даљу обраду нам је потребно да сви вектори добијени токенизацијом имају једнаку дужину. Извршено је додавање нула, (енгл. *Padding)*, код вектора са мањом дужином од потребне.

Јединствене речи, добијене токенизацијом, преставићемо матрицом која садржи информације о семантичким везама између речи, базираној на статистици заједничког понављања у великим текстовима. То је урађено користећи већ креирани фајл *GloVe*, који има 100 димензија. *GloVe (Global Vectors for Word Representation)*, је алгоритам не надзорног учења за добијање векторске репрезентације речи. У овом семинарском учитан је фајл са резултатима већ истренираног алгоритма, на великој бази података. Врши се проналазак сваког од токена из базе података, у овом фајлу и ако он постоји мења се одговарајућом матрицом, у супротном замењује се матрицом попуњеном нулама, примена овог алгоритма види се на слици 6.1.1.



Слика 6.1.1. Учитавање и примена *GloVe* фајла

За разлику од неких других метода, где се фокусирају на локални контекст, нпр. *Word2Vec skip-gram, GloVe* узима у обзор глобални контекст речи и њихова заједничка појављивања. На овај начин ухвати шире семантичке везе, и речи које имају слична значења или употребе такође имају сличне матричне репрезентације.

6.2. LSTM

LSTM (Long Short-Term Memory) је рекурентна неуронска мрежа (РНН), направљена да превазиђе ограничења традиционалне РНН мреже у чувању дуготрајних зависности секвенци података и настајању градијента. У стању су да обрађују секвенце података, као што су временске серије, текст и говор. Користе меморијске ћелије и капије за контролу протока информација, на овај начин селективно задрже или одбаце информације по потреби и тако избегавају проблем настајања градијента.[3]

6.3. BiLSTM

BiLSTM (Bidirectiona Long Short-Term Memory) је унапређење традиционалне *LSTM* архитектуре. Кључна разлика је у томе како се информације обрађују унутар мреже. Код стандардне *LSTM* се унализни низ обрађује у једном смеру, и чува зависности само од

претходних информација, док код *BiLSTM* се то врши у оба смера, и чува се зависност и од претходних и будућих информација.

6.4. CNN

CNN (Convolution Neural Network) конволуциона неуронска мрежа, која кроз своје слојеве примењује конволуционе операције на улазне податке. Конволуција представља прелаз филтером преко улазних података, да би се извукле карактеристике. Ови филтери науче да детектују потребне карактеристике за решавање одређеног проблема. Најчешћа примена је у обради слика и снимака, али овакав тип мреже налази своју примену и у обради природног језика. Дизајнирана је да обрађује структурине матричне податке, па као улаз може да прими матрице које представљају одређене токене извучене из текста, добијене нашом обрадом.

6.5. Креирање слојева код модела неуралних мрежа

Креирање слојева неуралних мрежа извршено је коришћењем *Sequential* класе из *keras* библиотеке. Код које је могуће једноставно додавање слојева користећи *add* методу.

Први слој код сваког модела, јесте *Embedding layer* из *keras* библиотеке. Примарна улога овог слоја је да мапира бројевну репрезентацију речи у вектор густине фиксне величине. Као улазне параметре прихвата величину речника тј. број јединствених речи, затим величину вектора помоћу кога је представљена свака реч, тежине одређене помоћу *GloVe* фајла, и на крају величину најдуже речи. Додавање овог слоја можемо видети на слици 6.5.1.

```
model = Sequential()
model.add(Embedding(len(word_index) + 1,
                    dim_glove,
                    weights=[word_vectorization_matrix],
                    input_length=maxlen_,
                    trainable=False))
```

Слика 6.5.1. Додавање *Embedding layer*

Други слој представља изабрани модел и служи за процесирање речи добијених од претходног слоја. За креирање сваког од слоја коришћена је одговарајућа класа из *keras* библиотеке. Након тога су додата два скривена слоја, праћена са *dropout layer* за регулацију, и на крају излазни слој за класификацију у једну од три класе. *Dropout* спречава *overfitting* током тренирања, одбацивањем дела података који су дошли од претходног слоја.

7. РЕЗУЛТАТИ

У овом поглављу биће представљени кључни резултати семинарског. Анализираћемо перформансе различитих модела класификације на описаном скупу података у поглављу 3, као и перформансе неуронских мрежа. Приказати њихове резултате коришћењем одговарајућих метрика.

7.1. Метрика

За евалуацију резултата класификације коришћене су различите метрике:

- Тачност
- Прецизност
- Одзив
- Ф1-мера
- Матрица конфузије
- *Multi-class Log Loss*

Свака метрика пружа јединствени увид у перформансе класификационих модела, омогућавајући дубље разумевање њихових способности и недостатака.

7.1.1. Тачност

Тачност даје информацију о генералној успешности предикције модел, кроз све класе. Рачуна се по следећој формули:

$$\mathrm{A} = \frac{Tcp}{Tcp+Fcp} \qquad (6.1.2.)$$

где Tcp представља број тачних предикција, а $Tcp + Fcp$, укупан број предикција.

7.1.2. Прецизност

Прецизност је метрика која процењује тачност позитивних предвиђања модела. Даје информацију о томе колико је модел склон тачном класификовању позитивних примера. Рачуна се по следећој формули:

$$\mathrm{P} = \frac{Tp}{Tp+Fp} \qquad (7.1.2.)$$

где Tp представља број правих позитивних вредности, тј. узорака које је модел класификације означио као позитивне, и они то заиста и јесу, а Fp представља број лажно позитивних, модел је негативне вредности погрешно класификовао.

7.1.3. Одзив

Одзив је метрика која процењује способност модела да тачно идентификује све стварно позитивне примере. Пружа информацију о томе колико је модел ефикасан у откривању позитивних примера. Рачуна се по следећој формули:

$$O = \frac{Tp}{Tp+Fn} \tag{7.1.3}$$

где Tp представља број правих позитивних вредности и Fn представља број лажно негативних вредности.

7.1.4. Ф1-мера

Ф1-мера комбинује прецизност и одзив модела у једну вредност, и представља њихову хармоничну средину. Користи се када је потребно узети у обзир равнотежу између тачности позитивних предвиђања и способности модела да идентификује све позитивне примере. Рачуна се по формули 6.1.4. где P представља вредност прецизности, а O одзива.

$$F1 = 2 * \frac{P*O}{P+O} \tag{7.1.4.}$$

7.1.5. Матрица конфузије

Табела која сумира број тачно позитивних, тачно негативних, лажно позитивних и лажно негативних предикција за сваку класу. Даје детаљан пресек перформанси модела за сваку класу.

7.1.6. Multi-class Log Loss

Процењује перформансе класификације модела, чији излаз представља вероватноћу између 0 и 1. У поставци проблема за такмичење, захтевана је евалуација решења коришћењем ове метрике. Добијене по следећој формули:

$$\text{logloss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij}) \tag{7.1.6.}$$

где је N број посматрања у тест подскупу, M број класа, у овом случају 3, y да ли тренутни улаз одговара датој класи или не, узима вредност 0 или 1, и p представља вероватноћу да дати улаз одговара посматраној класи. Коришћена је за процену перформанси само неуронских

мрежа, зато што оне директно враћају вероватноћу. Могуће је прилагодити и алгоритме машинског учења тако да враћају вероватноћу припадања одређеној класи, али ова вредност није значајна у процесу класификације датих алгоритама.

7.2. Резултати добијени алгоритмима машинског учења

Резултате класификације алгоритама машинског учења можемо видети у табели 7.2.1.

Најбоље се показао Мултиномијални наивни Бајесов класификатор, по свим параметрима, одмах после њега је Логистичка регресија. Док стабло одлучивања има доста слабе перформансе.

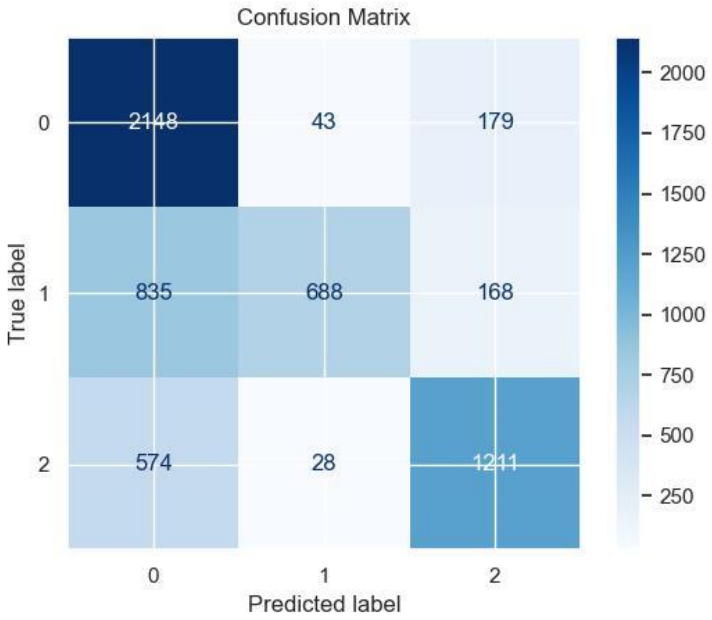
Табела 7.2.1. Резултати класификације алгоритама машинског учења

Алгоритми	Тачност [%]	Прецизност [%]	Одзив [%]	Ф1-мера [%]
КНН	68.9	74.45	68.897	67.588
Логистичка регресија	80.4	80.568	80.388	80.352
Стабло одлучивања	58.5	58.412	58.495	58.421
МНБ	81.3	82.203	81.324	81.246

У матрицама конфузије датим у наставку, лабеле су кодиране на следећи начин:

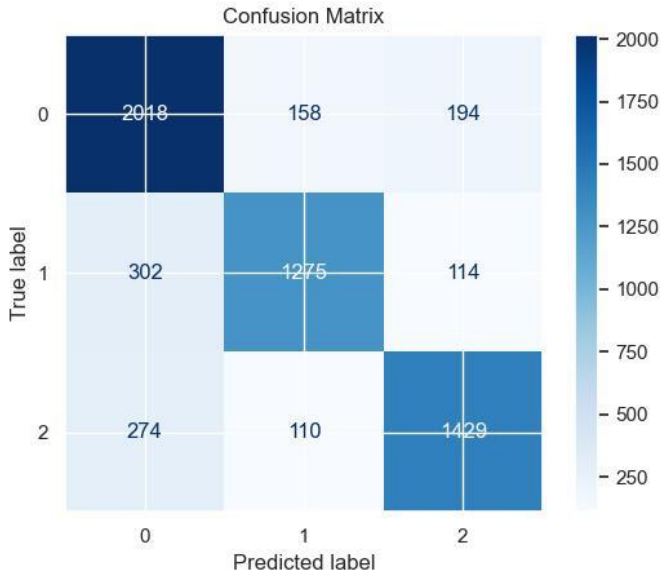
- 0 = Едгар Алан По
- 1 = Хаурд Филипс Лафкрафт
- 2 = Мери Шели

На слици 7.2.1. налази се матрица конфузије КНН алгоритма, можемо приметити да он има велике вредности дуж дијагонале, што се поклапа са перформансама алгоритма. Такође видимо да лажно детектује 0 класу, што значи да преписује По-у дела која нису његова, до овога долази због благог дизбаланса класа, где постоји више текстова овог аутора у односу на друга два.



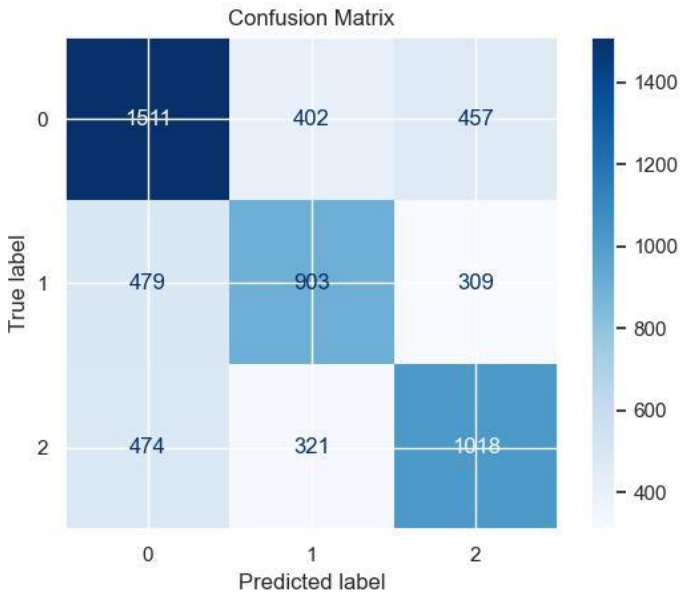
Слика 7.2.1. Матрица конфузије КНН алгоритма

На слици 7.2.3. налази се матрица конфузије логистичке регресије, на њој примећујемо доста мање вредности лажних позитива у односу на КНН алгоритам.



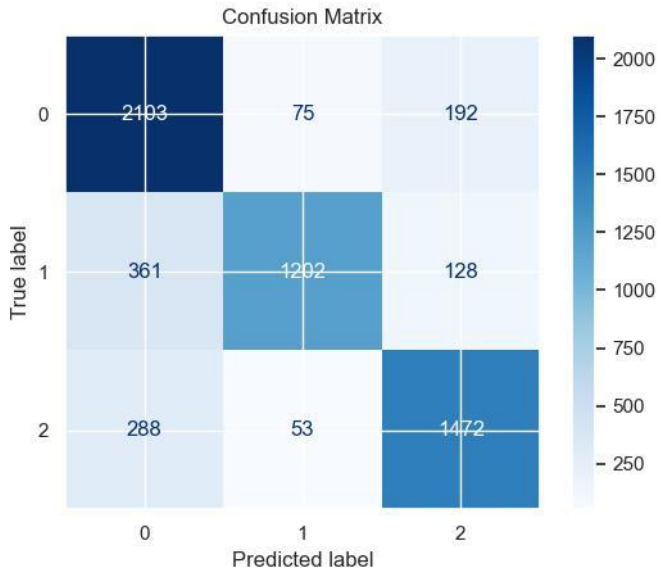
Слика 7.2.2. Матрица конфузије Логистичке регресије

Матрица конфузије стабла одлучивања приказана је на слици 7.2.3., на њој видимо знатно више вредности лажних позитива, као и ниже вредности на дијагонали. Овакви резултати одговарају перформансама алгоритма из табеле, који се показао као најлошији избор за решавање датог проблема.



Слика 7.2.3. Матрица конфузије стабла одлучивања

На слици 7.2.4. приказана је матрица конфузије мултиномијалног наивни Бајесов класификатор, који се показао најбоље у класификације. Али има јако сличне перформансе са логистичком регресијом, што се види и са упоређивањем њихових матрица конфузије.



Слика 7.2.4. Матрица конфузије МултиномијалногНБ

7.3. Резултати добијени неуронским мрежама

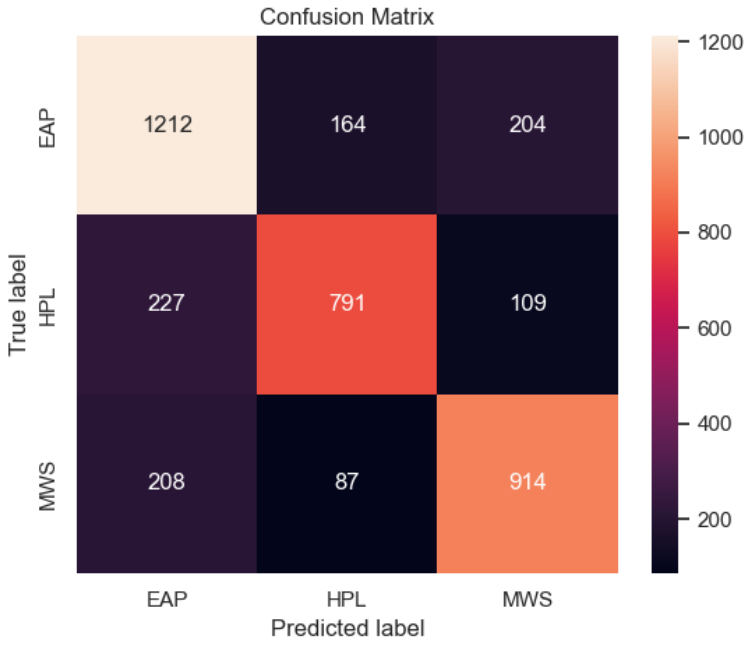
Резултате класификације неуронским мрежама можемо видети у табели 7.3.1.

Најбоље се показао Мултиномијални наивни Бајесов класификатор, по свим параметрима, одмах после њега је Логистичка регресија. Док стабло одлучивања има доста слабе перформансе.

Табела 7.3.1. Резултати класификације неуронским мрежама

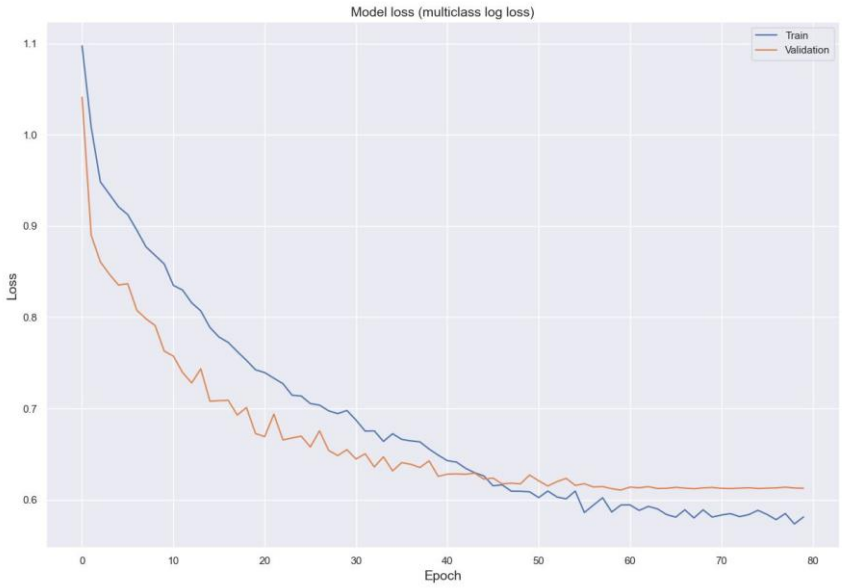
Алгоритми	Тачност [%]	Прецизност [%]	Одзив [%]	Ф1-мера [%]
LSTM	74.41	74.46	74.41	74.38
BiLSTM	74.95	74.95	74.95	74.94
CNN	69.87	70.11	69.87	69.76

На слици 7.3.1. налази се матрица конфузије *LSTM* модела, примећујемо високе вредности дуж дијагонале, али не занемарив број лажних позитива.



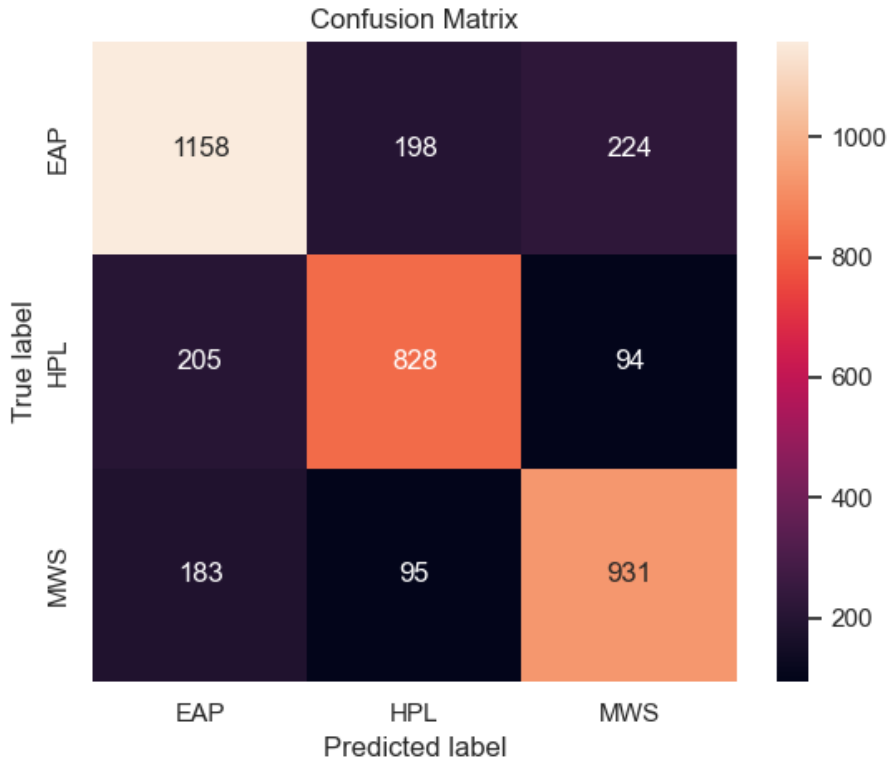
Слика 7.3.1. Матрица конфузије *LSTM*

На слици 7.3.2. приказани су графици *Log-Loss* функције, на тренинг и валидационом сету, можемо приметити да обе функције опадају што представља индикацију да не постоји *overfitting.*



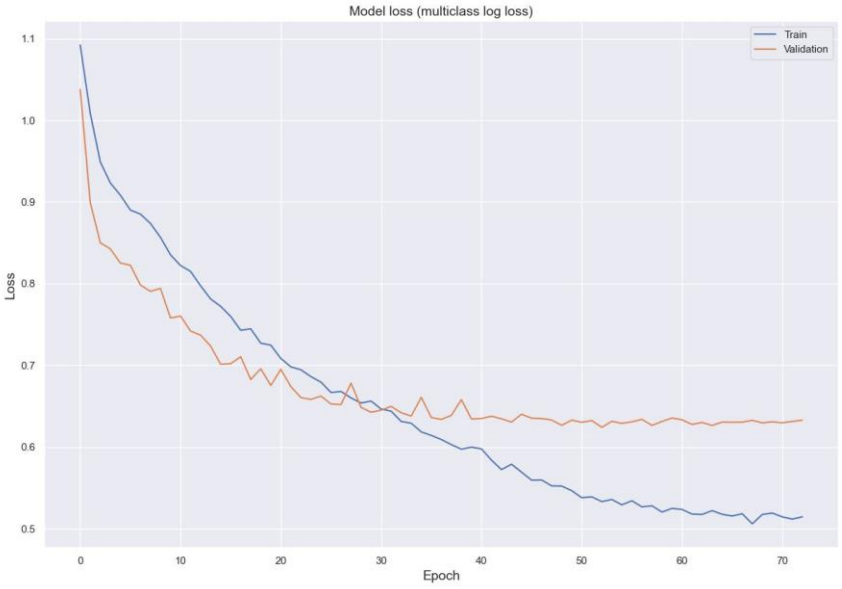
Слика 7.3.2. *Log-Loss* фунција *LSTM*

На слици 7.3.3. налази се матрица конфузије *BiLSTM* модела, који има сличне вредности као и *LSTM* модел, тако да бидирекционални приступ не доводи до успешније класификације.



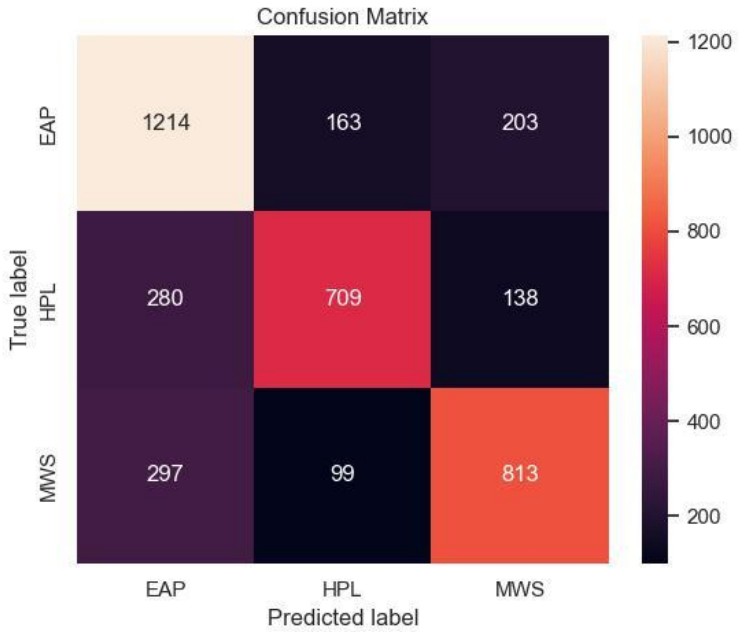
Слика 7.3.3. Матрица конфузије *BiLSM*

На слици 7.3.4. видимо *Log-Loss* функције за овај модел, које су опадајуће. Такође примећујемо више вредности валидационе Log-Loss функције што указује на веће грешке и већи проблем *overfitting-a* у односу на *LSM* модел.



Слика 7.3.4. *Log-Loss* функција *BiLSTM*

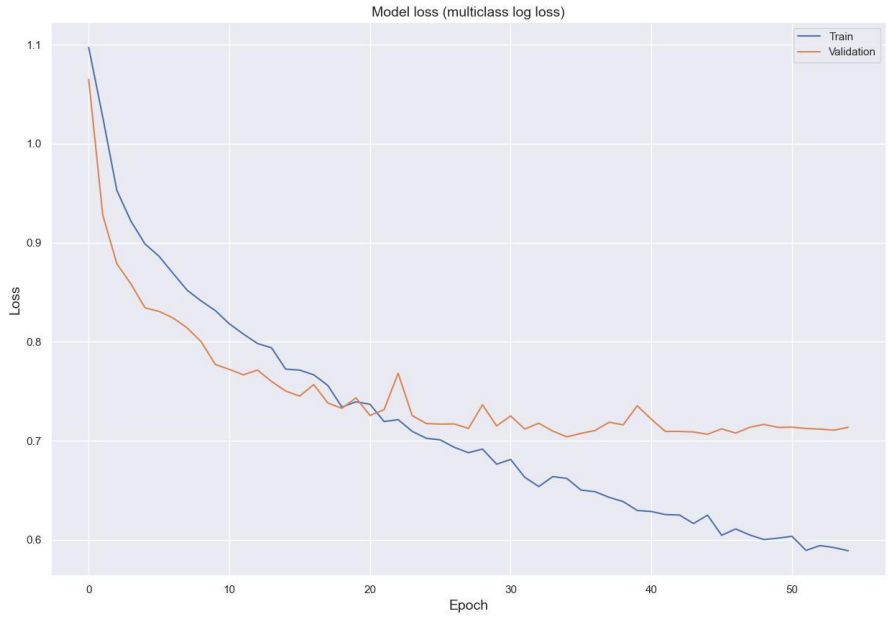
Матрица конфузије *CNN* мреже приказана је на слици 7.3.5., на њој видимо ниже вредности на дијагонали за не доминанту класу у односу на претходна два алгоритма.



Слика 7.3.5. Матрица конфузије *CNN* мреже

Слика 7.3.3. Матрица конфузије *BiLSM*

На слици 7.3.6. видимо *Log-Loss* функције за овај модел, које су опадајуће. Такође примећујемо више вредности валидационе Log-Loss функције што указује на веће грешке и већи проблем *overfitting-a*, али чињеница да ова функција има вредност мању од 0.5 указује на успешност алгоритма.



Слика 7.3.6. *LogLoss* фунција *CNN*

8. Закључак

На основу резултата приказаних у претходном поглављу, примећујемо да најбоље перформансе има мултиномијалног наивни Бајесов класификатор код кога све метрике су у распону 81.2% - 82.2%, након њега следи Логистичка регресија чије метрике достижу 80%. Након ова два алгоритма машинског учења, следе неуралне мреже, *LSTM I BiLSTM* дају јако сличне резултате, мало испод 75% али када погледамо график log-loss функције код *BiLSTM* модела примећујемо већи *overfitting*. Након њих следе *CNN* модел и КНН алгоритам, неурална мрежа има нижу прецизност, док се по осталим параметрима показала за нијансу боља од КНН алгоритма. Стабло одлучивања има најлошије резултате класификације.

Литература

[1] Spooky Author Identification, *https://www.kaggle.com/competitions/spooky-author-identification/overview (16.01.2024.)*

[2] Бајесова теорема, https://sr.wikipedia.org/sr-ec/Бајесова-теорема (17.01.2024.)

[3] Introduction to Long Short-Term Memory(LSTM), https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/lstm (17.01.2024.)

СПИСАК СЛИКА

СПИСАК ТАБЕЛА